ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DOTTORATO DI RICERCA IN

COMPUTER SCIENCE AND ENGINEERING

Ciclo 37

Settore Concorsuale: 09/H1 - SISTEMI DI ELABORAZIONE DELLE INFORMAZIONI

Settore Scientifico Disciplinare: ING-INF/05 - SISTEMI DI ELABORAZIONE DELLE INFORMAZIONI

DEEP REINFORCEMENT LEARNING AND CREATIVITY

Presentata da: Giorgio Franceschelli

Coordinatore Dottorato

Ilaria Bartolini

Supervisore

Mirco Musolesi

Co-supervisore

Paolo Torroni

Esame finale anno 2025

Abstract

Generative artificial intelligence (AI) is one of the most exciting developments in computer science over the last decade. Its impact has been tremendous: generative models, such as large language models, are revolutionizing several areas, including the arts, journalism, advertising, and scientific research, to name a few. In these fields, generative modeling is not only complementing but also replacing the creative abilities that were once solely in the hands of humans. However, current generative models are limited by their self-supervised learning schemes, which merely aim to imitate training data as accurately as possible. To develop more creativity-oriented models, new learning schemes should be considered. Among them, reinforcement learning (RL) represents a promising direction. RL is an inherently learning-by-acting approach. Moreover, since it is based on non-differentiable objectives, it is able to capture a greater variety of target behaviors. For these reasons, it is ideal for modeling how humans learn to behave creatively.

We claim that studying RL together with creativity can be of crucial importance for both fields. Based on these considerations, this thesis explores whether creativity can be used to enhance the design of RL algorithms and, vice versa, whether RL can help develop more creative generative models. In particular, we study if dreaming can help RL agents better generalize, as recently suggested for humans. Specifically, we leverage generative augmentations to transform standard, predicted trajectories into more dream-like experiences for training the agent, and we evaluate generalization capabilities in different low-resource scenarios. Then, we develop a new creativity score that quantifies both the originality and value of artifacts. We use this score as the basis of the reward structure in an RL framework, and we propose using it to fine-tune pre-trained generative models toward more creative solutions. We validate our proposed method in two different domains: poetry generation and mathematical problem resolution. In addition, we present new sampling schemes that can better simulate the human creative process by working at the response generation and validation levels.

Finally, we conclude with a deep analysis of three main social and practical issues related to AI creativity: whether current foundation models are creative and their main social implications; whether current foundation models can be entitled to agency and what can happen to human agency when collaborating with them; and how current copyright laws can manage the complexity of generative AI in terms of protecting human- and machine-generated artworks.

Contents

1 Introduction

1.1 A Historical Perspective

It was the year 1843 when Ada Lovelace, an English mathematician and writer recognized by many as the first computer programmer, wrote that the Analytical Engine [20] *"has no pretensions to originate anything. It can do whatever we know how to order it to perform"* [433]. This statement was then defined as "Lovelace's objection" by Alan Turing, who also provided an alternative formulation: a machine can never "take us by *surprise*" [662]. This was just the beginning of an ongoing philosophical discussion, which has often included psychological elements, around human creativity [29, 42, 76, 461, 624], as well as computational creativity [415, 691, 324, 55, 421, 131].

In general, computer scientists have always been fascinated by the possibility of building machines that are able to generate something "new". Among several applications of artificial intelligence (AI) to various artistic fields, it is worth mentioning the AARON Project by Harold Cohen, a program designed to draw images [128]; the Computerized Haiku by Margaret Masterman [124]; the storyteller TALE-SPIN [432]; RACTER and its poems' book [510]; MEXICA and its short narratives [501]; the artificial composer of David Cope [133]; BACON to simulate human thought processes and discover scientific laws [360]; the COPYCAT Project to discover insightful analogies [283]; and many others [443]. Different AI techniques have been explored, from planning [537] and case-based reasoning [663] to evolutionary strategies [425]. Some approaches combine all of them [215]. This growing interest has contributed to the emergence of a specialized field in computer science, namely computational creativity [90], which concerns with the study of the relationship between creativity and artificial systems [131, 691].

In this context, the adoption of deep learning (DL) techniques has led to substantial breakthroughs in recent years. Vast computational power and very large amounts of available data are at the basis of the increasing success of deep generative models (i.e., generative models based on DL [194]). Indeed, generative deep learning technologies have been used to write newspaper articles [229], generate human faces [333] and voices [388], write and publicly perform poems [192], design drugs and proteins [325], and even create artworks sold for hundred thousand dollars [119].

While close in terms of application scenarios, computational creativity and generative DL are profoundly different. The former aims at building AI models that adhere to human theories of creativity; in other words, it works through a *top-down* approach, i.e., by first defining the final goal and then trying to develop appropriate components to achieve it. The latter learns to generate artifacts regardless of such theories and is only used for creative purposes *a posteriori*; in other words, it works through a *bottom-up* approach (first developing the best possible components and then assembling them to reach the goal).

However, the analysis of both solutions revolves around a fundamental question: *What is creativity, and what does it mean to be creative?* The next section considers the main theories of creativity, and how they are related to current - and future - AI systems.

1.2 A Preliminary Analysis of Creativity and Artificial Intelligence

At first sight, the question of what creativity is appears to be straightforward, if not rhetorical in nature. Anyone possesses some knowledge about creativity and can recognize it at first sight. However, in practice, defining it is incredibly hard. This is because, using Minsky's characterization, creativity is one of those *suitcaselike words* we use to conceal the complexity of large ranges of different things [445]. As beautifully depicted by Prentky [508],

> "what creativity is, and what it is not, hangs as the mythical albatross around the neck of scientific research on creativity."

In fact, it has been estimated that more than one hundred definitions of creativity have been proposed [8, 657], and perhaps the count is still growing. Among them, a prominent work is that of Rhodes [535], who was arguably the first to provide a unique framework that covers most of the various definitions. According to Rhodes, creativity has to be considered under four perspectives: *product*, *process*, *press*, and *person*. In the remainder of this section, we (as well as others in the context of computational creativity [323, 356]) follow this categorization and independently discuss each of them, first theoretically, and then in the light of the most recent AI developments.

1.2.1 Product

Products are artifacts of thoughts, ideas embodied into a tangible form by the means of words, paint, fabric, or other materials [535]. The idea-expression dichotomy is a central point in the conversation, not only from a practical perspective (see for example copyright, which only protects the expression and not the idea [553]), but also from a theoretical one. A line of thought assumes that the creative act is completed when the idea of the work is formed [204]; the thinking up of the work *is* the creative act, and its physical realization is a non-creative activity [653]. Excluding the materialization of the creative idea in a tangible form from the analysis of creativity is anyway incorrect. Indeed, a creative idea is typically refined during and after its implementation, therefore requiring the act of making; even more importantly, only the tangible expression of the idea can be communicated, perceived, and evaluated as a creative product. For this reason, we can generally say that focusing on the product means directing our attention to the material result of the creative act. Although it is not straightforward to assume that machines can have thoughts or ideas (and therefore that they are able to act creatively), it is apparent that they can produce artifacts. According to this perspective, a machine can be considered creative if its productions can be defined as creative.

Boden [53] defines creativity as the ability to generate ideas or artifacts that are *new*, *surprising*, and *valuable*. A large number of researchers agrees on the centrality of value, also referred to as quality [538], and novelty (first proposed in 1959 [461]) in assessing a computer's creativity [421, 672, 691]. Commonly, value and novelty are combined with surprise [414], but sometimes they are also combined with other dimensions such as typicality [538], transformativity [230], rarity and recreational effort [377], among others.

Novelty is generally defined as the dissimilarity between the produced item and other examples in its class [538]; we can see it as the property of not having been experienced yet, either by the producer or by anyone in history [53]. While it has typically been considered for automatic evaluation methods [172, 453, 672], there have been few examples of its adoption for generating products; we will introduce them in Section 3.1.7.

Surprise is about how much a stimulus disagrees with expectations [31, 44]. Boden identifies three categories of surprise that lead to three different forms of creativity, ordered by increasing rarity and produced surprise. *Combinatorial* creativity is about making unfamiliar combinations of familiar ideas, e.g., analogies in textual forms or collages in the visual arts. *Exploratory* creativity involves the exploration of the conceptual space defined by the cultural context considered, e.g., inventing a new type of cut for fries.

Transformational creativity involves changing that space in a way that allows new and previously inconceivable thoughts to become possible, as it has been for free verse in poetry or abstract painting in art. From a computational perspective, surprise can be computed as the surprisal [658], which is the unexpectedness of an event (i.e., the prediction error), or as the Bayesian surprise [306], that is the difference between posterior and prior beliefs about the world (before and after experiencing that event). Interestingly, it has been found that a limit exists when trying to maximize both Bayesian surprise and quality of artificial productions to obtain combinatorial creativity and that creative products should lay around that boundary [671]. Several computational methods to calculate surprise have been used [85, 230, 328]; however, only one attempt has been made to produce something surprising (see Section 3.1.7).

Value has been considered as a measure of how the artifact compares to others in its class in terms of utility, performance, and attractiveness; it is the reflection of its acceptance by society [421]. Value is the property of an artifact of being a good, qualitative contribution to its field. In these terms, it is a fundamental property of AI systems. For instance, this is the goal of loss functions used in deep learning: by optimizing a certain function, more valuable results can be achieved. An example is the so-called Generative Adversarial Network [226] in which two networks are used to generate and discriminate outputs that are deemed valuable in a given field (such as visual arts) or specific application (such as design of furniture).

Since its inception, the focus of generative DL has been on the design of effective loss functions (and in some cases architectures), as we will see in Sections 2.1 and 3.1. The most relevant attempts to directly induce the model to produce creative results are based on creativity-oriented loss functions, as we will discuss in Section 3.1.7. Unfortunately, these are generally considered unsatisfactory. In fact, value, novelty, and surprise are three concepts difficult to quantify in practice, especially in a differentiable form as required by self-supervised learning techniques. Alternatives include evolutionary algorithms for achieving novelty [376] and surprise [232], which can also be used in combination with classic evolutionary search (where the fitness function can represent the quality of a solution) [233]. These approaches have been recently integrated with neural networks as well [132, 307, 392, 589, 632].

1.2.2 Process

Focusing on the product alone is not sufficient. The resulting product is fundamental, but how such a result is achieved is important as well: a switch from creativity in the product to creativity in the process may be taken into consideration to develop the creativity of an AI agent.

Process is about motivation, perception, learning, thinking, and communication [535]. Its focus is on how we arrive at a certain result, how we create the prerequisites to obtain that result, and how we finally perceive and communicate it. In this context, it is important to keep the analysis as general as possible and not to refer specifically to arts; although we can refer to artistic processes, the same considerations can be applied to other kinds of activities. In general, thinking and behaving creatively does not require a person to be an artist. One can be creative when cooking, when approaching a new maths problem at school, and in general in any task that is heuristic rather than algorithmic [10], i.e., without a clear and straightforward path to the solution [272].

Focusing on creativity in products allows for non-creative processes to be adopted. For example, the backpropagation and inference algorithms used by the loss-based approaches introduced above cannot be seen as a form of a creative process per se (even if they are a highly creative result of a human creative process [545]). Instead, reinforcement learning (RL) [639] appears to be a promising technique since it is based on the idea of learning by doing, mirroring certain human processes [596].

More specifically, RL consists in learning how to act in order to maximize a numerical signal, i.e., the *reward*, over time [639]. At each time step, the agent receives the current state of the environment and performs an action, observing its consequences, i.e., the reward and entering a new state. Through multiple interactions with the environment, the agent learns to maximize the cumulative reward.

An interesting direction is the use of intrinsic motivation. This refers to exploratory and playful behaviors observed in animals [690], which are performed not only because they might lead to rewards but since they are inherently interesting or enjoyable [146]; these are typically activities that have the appeal of novelty, challenge, or aesthetic value for an individual [549]. It has been demonstrated that intrinsic motivation plays a crucial role in creativity [11]. Indeed, motivation is the first step of the typical human creative process, which also involves a preparation step (where the necessary information is acquired or regained), a response generation step (thanks to creativity-relevant skills), and a response validation step (thanks to domain-relevant skills), which can lead to repeat one or more of these steps in the case the evaluation is not passed successfully [10]. Motivation, particularly intrinsic motivation, has been studied in RL [32, 604], where it has become a prominent alternative to the standard, extrinsic reward provided by the environment. Most of the time, it has been modeled through curiosity.

In fact, curiosity is an intense, intrinsically driven appetite for information and knowledge [406], a motivational prerequisite for exploratory behavior [42]. Curiosity can be characterized according to the kind of (uncomfortable) states that contribute to its emergence: *under-stimulation*, which leads to *diversive* curiosity, e.g., when someone is bored and wants to do something different or exciting; and *over-stimulation*, which leads to *specific* curiosity, e.g., when someone has run into something different and arousing and wants to know more [43]. This overlaps with two possible underlying driving forces: information gap [406], i.e., one is curious to fill a knowledge gap about a certain, known context, which is close to specific curiosity; learning progress [565], i.e., one is curious to learn more [228], which is close to diversive curiosity. In general, we might say that curiosity is a driving power for deeper (i.e., specific) and further (i.e., diversive) exploration.

Many RL approaches use curiosity as an intrinsic reward, with or without relying on an extrinsic one. Some of them are based on acquiring skills [63], others on trying to discover as many states as possible [51, 627], also by maximizing uncertainty (typically defined in terms of self-disagreement [494, 577]). However, the most adopted approaches are based on the error in predicting the consequences of actions or the difference between the posterior distribution and the prior distribution (see Section 3.2.2 for more details).

Essentially, they are based on the notions of surprisal and Bayesian surprise, respectively. Indeed, surprise can be both the cause and the consequence of curiosity, as formally theorized in [231]. However, it is important to make a distinction: while creativity *in the product* is related to an extrinsic surprise (which should be experienced by whoever judges the creativity of the result), creativity *in the process* is related to an intrinsic surprise (which should be experienced as a motivation to explore and learn more). A similar consideration also applies to novelty which is less used in RL [338, 562, 598] but is equally important in guiding curiosity in experiencing and learning new things.

Even though a distinction between novelty and surprise in process and product exists, without experiencing or having experienced intrinsic novelty and surprise, it seems illogical that someone can create a result that the entire world would consider novel and surprising. Exercising curiosity is therefore necessary to move from a creative process to a creative product. As far as value is concerned, it may be easily translated into expertise acquisition. One must learn how to properly solve a task; skills are required as well as intrinsic motivation to perform creatively [10]. Skills are required to materialize the original idea and therefore communicate it to others [653]; in addition, they are required to formulate an idea and assess its quality [204], i.e., in the response validation and generation steps [11]. Again, RL appears to be suitable for expertise acquisition: by acting repeatedly, the agent can learn the consequences of its actions, acquiring both knowledge about the task, and skills needed to solve it in the best possible way.

Going back to the definition of process, we need to consider an additional dimension: imagination. It is the ability to form concepts not perceived by the senses [302]; it allows us to reflect on the consequences of potential future actions and to think in advance about different possible alternatives. Intuitively, a creative process takes advantage of imagination by thinking in advance about diverse (novel and surprising) ways of achieving a certain objective. As Gaut [203] puts it, imagination is the vehicle for one's creative explorations.

Imagination has become a topic of interest in RL as well [259]. In this context, it refers to the possibility of using an internal world model to generate entire imagined episodes, without the need to collect them [252]. These episodes are used either to augment the current state and better guide the policy in its action choice [509] or the replay memory on which the agent is trained; we will analyze them in Section 3.2.1.

Closely related to imagination is dreaming. Although it is not yet clear how and why humans dream, one of the most prominent theories suggests dreams to be imagined, fictional scenes whose function is to help the brain learn better [281]. Specifically, dreams may be used to generalize and avoid overfitting by providing experiences different from the daily ones. However, current imagination-based RL research has been focusing on imagining experiences as close as possible to those accumulated during standard exploration. This is in contrast with human dreams that allow one to explore the experiential state space in ways that deviate from waking life [280]. According to [280], the fiction the brain produces at night is of the same kind of fiction produced by fabulists, or surrealist artists; in a sense, the fiction the artists are in the business of producing is nothing different from consumable, portable, durable, and in the end superior artificial dreams.

To summarize, expertise, curiosity, imagination, and dreaming are (part of) what creative processes require. Nonetheless, we still miss why such creative processes lead to creative products only occasionally. The next section will cover this gap.

1.2.3 Press

While a process can be considered creative only by focusing on the individual who has performed it, for the assessment of creativity in products an individual point of view is not sufficient. It is necessary to switch to a societal perspective, the so-called *press* [535]. The term press refers to the relationship between human beings and their environments, where creative products are influenced by certain kinds of forces played upon individuals as they grow up and as they function [535].

Particularly relevant in this direction is the work of Csikszentmihalyi [140], who stated it is not possible to study creativity by isolating individuals and their works from the social and historical milieu in which their actions are carried out. On the contrary, what is defined as creative is the product of three main shaping forces: the field (a set of persons, e.g., critics, historians, or peer groups of creators, which select those that are worth preserving from the variations produced by individuals); the domain (that preserves and transmits the selected variations to the following generations); and, finally, the individual (who produces variations that the field can consider as creative). Creativity can exist only thanks to the interaction of all these three components since each one affects and is affected by the other two. The person is still important, but only as part of a system of mutual influences and information [140]. Previously, we discussed how a product is creative if it is novel, surprising, and valuable. This section provides the answer to questions like: with respect to what is it possible to consider the novelty? The answer, now, is straightforward: the domain. And who can judge the value of, and be surprised by, a product? The field. This means that creativity depends not only on the mere attributes of a product, on the process followed, or on the person who generated it; it also depends on the persons assessing its creativity and on the attributes of the environment, which influences the source of evaluation as well as the source of stimulation and inspiration [33]. In general, the creative product is strictly dependent on the sociocultural context in which it has been thought, produced, distributed, and accepted [409].

To summarize, the systems view is a sort of continuous evolutionary process: at first, the generation of novel products by the individuals; then, the selection of the most creative products by the field; and finally, the transmission of the selected products through the domain, through which individuals can generate novel products [140]. This helps us understand how the press can be simulated in AI. Multi-agent systems, especially multi-agent RL systems, seem suitable, potentially leading to the so-called computational social creativity [560]. Generative agents can play the role of individuals; collected examples (i.e., a training set) and rules (e.g., a knowledge base) governing generation and evaluation can represent the domain; and other generative or discriminative agents can simulate the field in evaluating other agents' productions. The three phases of generation, selection (evaluation), and transmission (extension of domain with new products) can then be iterated through several epochs. For instance, the Digital Clockwork Muse [561] implements the theories of Martindale [428] by means of a collection of curious agents playing the role of both creators (through a genetic algorithm) and evaluators (by computing a personal degree of novelty through a self-organizing map). RL can also be used to train a collection of agents to exercise both self-critic (by proposing variations that achieve a sufficient degree of intrinsic novelty) and voting (by evaluating the novelty of others' productions) [399]. Though remarkable, these attempts are just the beginning, and it is possible to envision future systems that also consider value and surprise in the product and a more creative process.

1.2.4 Person

We now consider the fourth of the four dimensions of creativity listed at the beginning. Even in the same environment and with the same training, two persons will arrive at different creative products. For example, two students of the same class tasked to complete the same creative homework through the same process (e.g., write a new poem from Dickinson's *Hope* by substituting only the word 'hope' with another one) will produce two different compositions. The person in themselves influences the results and the degree of creativity. The term *person* covers information about personality, intellect, temperament, habits, attitude as well as value systems and defense mechanisms [535]. Many theories have been proposed to explain which properties of the personality are more connected with creativity [187]. For instance, Rogers [541] suggests openness to experience, an internal locus of evaluation, and an ability to toy with elements and concepts as inner conditions for creativity; while Miller [443] lists the need for introspection, the knowledge of your own strengths, not being afraid of making mistakes, and having different experiences and suffering among the hallmarks of creativity. Gaut [205] suggests that a creative agent must exhibit flair (for that the product depends on its purposes as well as on its understanding and judgment). In general, the term *person* refers to all traits of personality, emotions, intentions, and experiences that can influence a creative result.

Framing the person perspective in the context of computational creativity is not straightforward. Note that it is not about defining an artificial agent through human attributes, which is another fundamental and perhaps unsolvable problem [95], nor it is only a matter of terminology, for it is sufficient to replace *person* with a less anthropomorphic word such as *producer* [323]. The problem is how an artificial agent can be truly entitled to the personal characteristics, purposes, and behaviors we typically attribute to the author. Navigating through the list of information covered by the person, intellect is the easiest to assume in AI. In addition, the agent can be entitled to a value system, if a component such as the discriminative part of a GAN is accepted as a way to measure value. In theory, it might also have habits, if habits merely mean following unconscious patterns and repeating actions. On the other hand, attitude is by definition related to feelings and opinions; temperament is about moods and behaviors; defense mechanisms require feelings too; and personality is the hardest of all, assuming as its definition "the set of emotional qualities, ways of behaving, etc., that makes a person different from other people".

One may argue certain qualities make a creative agent different from others. Sometimes, these qualities might even be linked with opinions, emotions, and preferences. A neural network producing classical music is different from another one generating jazz variations. But is it a reflection of its personality? The output of the model only depends on the training set used during training, its architecture and hyper-parameters, and even the random vector passed as input or the stochastic sampling of its output. Now, one may argue that the very same kind of influence exists for humans too: if our teacher teaches us to write about real feelings and to use everyday English, it is more probable to end with a short story in the style of Raymond Carver, than that of a Brothers Grimm's fairy tale. And if we are born in a Western country, it is more probable we will paint like Monet or Picasso rather than like Hokusai and Hiroshige. But there is a substantial difference: we are free to visit an art gallery, fall in love with ukyio-e, and decide to abandon our previous path. Or we can be inspired by them and try to merge that style with our habits. A neural network does not have the freedom of choice over its source of inspiration: in a sense, it lacks both liberty (independence from controlling principles) and agency (capacity for intentional action) [305], which are crucial for self-determination and, then, for creativity [550]. It does not have the chance of falling in love. It does not have the opportunity of going out of its habits. It does not have the possibility of experiencing something, being dramatically upset by it, and then deciding to write a poem on such an emotion. The only emotions it can write about are those it has come across during training or the ones it has been asked to compose on. In other words, all the emotional characteristics an AI can express are artificial in comparison to human ones.

It is the developer's personality that influences most of the creative aspects of the artificial agent. Whatever creative process the agent follows, whatever creative product the agent returns, whatever multi-agent system influences its directions, the decisions of the human creator remain a central part of the design of these AI systems, unless one day machines become real social agents [270]. Asking if a machine can be *truly*, *genuinely* creative (i.e., creative upon all the four P's), is not different from asking if a machine can be human [30].

1.3 Research Questions and Structure

In the previous section, we briefly discussed the current state-of-the-art at the intersection between creativity and deep learning, and the potential role of reinforcement learning. Starting from this technological scenario, this dissertation explores two main research questions. Is it possible to use creativity to enhance RL research, and is it possible to develop RL methods to train more creative generative models? More specifically, we aim to study if dreaming can help RL agents better generalize, as recently suggested for humans. At the same time, we develop RL-based strategies to push generative models toward more creative outputs.

This thesis is structured as follows:

- Chapter 2 introduces all the fundamental concepts and building blocks we will rely upon in the remainder of the thesis. In particular, we describe the main generative deep learning models by focusing on their learning and inference schemes; we introduce the RL framework, together with practical algorithms to train neural network-based agents and the foundations of imagination-based RL; and we examine how RL can be applied to generative modeling.

- Chapter 3 surveys the literature around the main topics of this dissertation. In particular, we focus on generative DL and its relation with creativity; reinforcement learning, with a specific focus on imagination-based RL, curiosity-driven RL, and generalization; reinforcement learning for generative modeling, discussing the advantages and limitations of its adoption; and current research involving generative AI and society and some of its main issues such as AI anthropomorphization, its use for creativity tasks, and the main legal problems related to the adoption of these technologies.

- Chapter 4 addresses the first research question, i.e., whether "dreaming" can help RL agents generalize better. Leveraging creative augmentations, we transform standard, predicted trajectories typical of imagination-based RL into more dream-like experiences for training the agent. We evaluate the obtained agent and its generalization capabilities on ProcGen environments [126] with limited-resource scenarios.

- Chapter 5 presents the approach developed to answer the second research question, i.e., whether reinforcement learning can help generative models produce more creative outputs. We develop a new information-theoretic creativity score that acknowledges both the originality and value of artifacts. We use this score as the reward in an RL framework to fine-tune pre-trained generative models toward more creative solutions. We validate our new method in two different domains: poetry generation and mathematical problem resolution.

- Chapter 6 presents other strategies, not related to RL, to increase the creativity of machine-generated products. By focusing on the response generation and validation steps we propose two new sampling schemes that can better simulate the human creative process.
- Chapter 7 discusses three main social and practical issues arising from the use of generative AI for creative purposes: whether current foundation models are creative and their main social implications; whether current foundation models can be entitled to agency and what can happen to human agency when collaborating with them; and how current copyright laws can manage the complexity of generative AI in terms of protecting human- and machine-generated artworks.
- Chapter 8 concludes the thesis, summarizing the main results and discussing potential future directions for generative AI, RL, and (computational) creativity.

2 Preliminaries

2.1 Generative Deep Learning

A generative model can be defined as follows: given a dataset of observations X, and assuming that X has been generated according to an unknown distribution p_{data}, a generative model p_{model} is a model able to mimic p_{data}. By sampling from p_{model}, observations that appear to have been drawn from p_{data} can be generated [194]. Generative deep learning is just the application of deep learning techniques to form p_{model}.

At first glance, this definition appears to be incompatible with the main requirements of creativity as presented in Section 1.2. Indeed, mimicry is the opposite of novelty. However, what a generative model should aim at mimicking is the underlying distribution representative of the artifacts, and not the specific artifacts themselves; in other words, it should aim at learning the conceptual space defined by the context considered. While generative models can be described in terms of different dimensions, we focus on the two most relevant for creativity: how the space of solutions is learned from real data; and how the model samples a new solution from that space.

In this section, we introduce the main classes of existing generative deep learning models. In particular, we analyze how the models learn their spaces of solutions and how the observations are generated from them. Figure 2.1 provides a summary of the six generative classes considered in this section.

2.1.1 Variational Autoencoders

A Variational Autoencoder (VAE) [339, 534] is a learning architecture composed of two models: an encoder (or recognition model) and a decoder (or generative model). The former compresses high-dimensional input data into a latent space, i.e., a lower-dimensional space whose features are not directly observable, yet provide a meaningful representation. The latter decompresses the representation vector back to the original domain [194]. Classic autoencoders directly learn to represent each input in a latent representation vector.

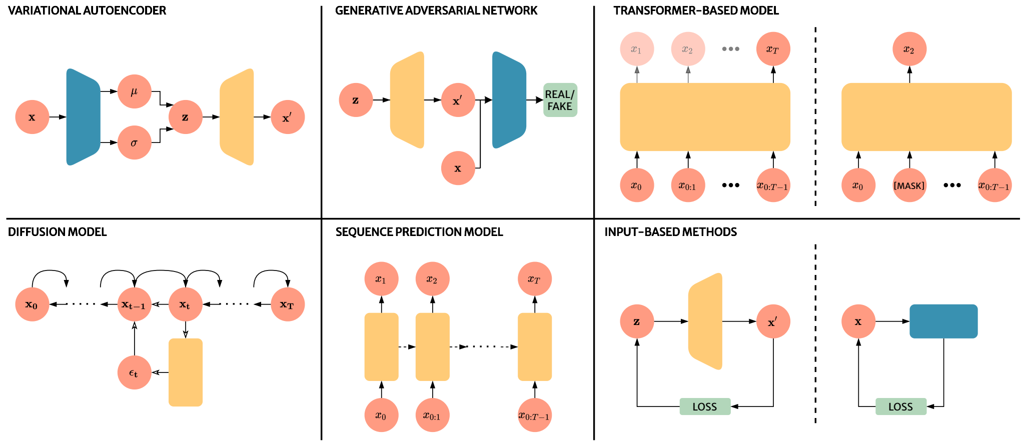


Figure 2.1: A schematic view of the six classes of generative learning methods presented in this section. Top, left to right: Variational Autoencoder (2.1.1), with a decoder generating $\mathbf{x}'$ given a latent vector $\mathbf{z}$, and an encoder representing $\mathbf{x}$ into a latent distribution; Generative Adversarial Network (2.1.2), with a generator to produce $\mathbf{x}'$, and a discriminator to distinguish between real $\mathbf{x}$ and synthetic $\mathbf{x}'$; Transformer-based model (2.1.4), with a Transformer outputting $\mathbf{x}$ one token after the other given in input previous tokens, or a masked version of $\mathbf{x}$. Bottom, left to right: Diffusion model (2.1.5), with a model to learn an error $\boldsymbol{\epsilon}$, which is used to incrementally reconstruct $\mathbf{x_0}$; Sequence prediction model (2.1.3), with a generator to output $\mathbf{x}$ one token after the other given in input previous tokens; Input-based methods (2.1.6), with an input optimized by a given loss. The input can be a vector $\mathbf{z}$ given to a generative model to obtain the desired output, or directly a product $\mathbf{x}$ becoming the desired output.

Conversely, VAEs learn a (Gaussian) distribution over the possible values of the latent representation, i.e., the encoder learns the mean and the (log of the) variance of the distribution.

VAEs are trained by optimizing two losses: the reconstruction loss and the regularization loss. The former is the log-likelihood of the real data $\mathbf{x}$ from the decoder given their latent vectors $\mathbf{z}$, i.e., it is the error of the decoder in reconstructing $\mathbf{x}$. The latter is the Kullback-Leibler (KL) divergence between the distribution learned by the encoder and a prior distribution, e.g., a Gaussian. The overall loss function is reported in Equation 2.1.

$$L_{\boldsymbol{\theta},\boldsymbol{\phi}}(X) = \mathbb{E}_{\mathbf{x}\sim X}\left[\mathbb{E}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\cdot|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) - D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \;||\, \mathcal{N}(\mathbf{0},\mathbf{I}))\right]\right], \quad (2.1)$$

with:

$$\mathbf{z} = \boldsymbol{\mu}_{\mathbf{x}} + \boldsymbol{\sigma}_{\mathbf{x}}\epsilon, \epsilon \sim \mathcal{N}(0, I)\,, \quad (2.2)$$

where $q_{\boldsymbol{\phi}}$ is the encoder returning the mean $\boldsymbol{\mu}$ and the standard deviation $\boldsymbol{\sigma}$ given the input $\mathbf{x}$ while $p_{\boldsymbol{\theta}}$ is the decoder. Notably, the latent vector $\mathbf{z}$ in input to the decoder is obtained by means of the so-called *reparameterization trick* (Equation 2.2), i.e., instead of directly sampling from the distribution defined by the mean and the variance, we separate the deterministic and stochastic parts of the operation by combining them with an ϵ sampled from a multivariate normal distribution. Without it, sampling would induce noise in the gradients required for learning [340]. To generate a new output $\mathbf{x}'$, we only need to sample a random latent vector from the multivariate normal distribution and pass it to the generator:

$$\mathbf{x}' \sim p_{\boldsymbol{\theta}}(\mathbf{x}'|\mathbf{z}')\,, \mathbf{z}' \sim \mathcal{N}(\mathbf{0}, \mathbf{I})\,. \tag{2.3}$$

The mathematical derivation of the whole loss has its roots in variational inference [322]. Indeed, VAEs can be seen as an efficient and stochastic variational inference method, in which neural networks (NNs) and stochastic gradient descent are used to learn an approximation (i.e., the encoder) of the true posterior [198]. In VAEs, similar high-dimensional data are mapped to close distributions. This makes it possible to sample a random point $\mathbf{z}$ from the latent space, and still obtain a comprehensible reconstruction [194]. On the other hand, VAE tends to produce blurred images [733]. It may also happen that high-density regions under the prior have a low density under the approximate posterior, i.e., these regions are not decoded to data-like samples [14]. Finally, the objective can lead to overly simplified representations without using the entire capacity, obtaining only a sub-optimal generative model [81].

2.1.2 Generative Adversarial Networks

A Generative Adversarial Network (GAN) [226] is an architecture composed by two networks: a generative model and a discriminative model. The latter learns to distinguish between real samples and samples generated by the former. In parallel, the former learns to produce samples from random noise vectors such that they are recognized as real by the latter. This competition drives both models to improve their methods until the generated samples are indistinguishable from the original ones. Equation 2.4 reports the overall objective function:

$$\min_{G_{\boldsymbol{\theta}}} \max_{D_{\boldsymbol{\phi}}} \mathbb{E}_{\mathbf{x} \sim X} \left[\log D_{\boldsymbol{\phi}}(\mathbf{x})\right] + \mathbb{E}_{\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[\log(1 - D_{\boldsymbol{\phi}}(G_{\boldsymbol{\theta}}(\mathbf{z})))\right], \tag{2.4}$$

where $G_{\boldsymbol{\theta}}$ is the generator network and $D_{\boldsymbol{\phi}}$ is the discriminator network.

The adversarial training allows the generator to learn to produce seemingly real samples from random noise without being exposed to data. Then, to generate a new output $\mathbf{x}'$, we just need to sample a random latent vector from the multivariate normal distribution and pass it to the generator:

$$\mathbf{x}' \sim G_{\boldsymbol{\theta}}(\mathbf{x}'|\mathbf{z}')\,, \mathbf{z}' \sim \mathcal{N}(\mathbf{0}, \mathbf{I})\,. \tag{2.5}$$

The simplicity of the idea and the quality of results are the basis of the success of GANs. However, few limitations exist. For instance, GAN can suffer from mode collapse, where the generator only learns to produce a small subset of the real samples [437]. In addition, the latent space of random inputs is typically not disentangled and it is necessary to introduce constraints in order to learn an interpretable representation [335].

2.1.3 Sequence Prediction Models

A sequence prediction model is a generative model that considers generation as a sequential process. It works in an autoregressive fashion: it predicts the future outcome of the sequence (i.e., the next token[1]) from the previously observed outcomes of that sequence, usually by means of an internal state that encodes information from the past. It is trained to maximize the log-probability of each token in the dataset as per Equation 2.6:

$$\max \mathbb{E}_{\mathbf{x},\mathbf{c} \sim X} \left[\sum_{t=1}^{T} \log p_{\boldsymbol{\theta}}(x_t | x_{t-1} \dots x_1, \mathbf{c}) \right], \tag{2.6}$$

where $p_{\boldsymbol{\theta}}$ is the prediction network which returns the probability distribution of the next token, each data sample is a sequence $\mathbf{x} = (x_1 \dots x_T)$, and $\mathbf{c}$ is an optional input to condition the generation (e.g., a desired class or style). At inference time, this simple yet effective approach only requires sampling one token after the other, feeding back to the model what has been produced so far [329] and potentially a conditional input $\mathbf{c}'$ randomly sampled or passed by the user:

$$\mathbf{x}' = (x'_1 \dots x'_T), x'_t \sim p_{\boldsymbol{\theta}}\big(x'_t | x'_{t-1} \dots x'_1, \mathbf{c}'\big)\ \forall t \in [1, T]. \tag{2.7}$$

In other words, it learns dependencies between tokens in real data so that the same dependencies can be exploited when generating synthetic data.

[1] We use the term "token" to refer to any discrete element an unstructured data point can be broken into, independently the data source is in the form of text (e.g., [513]), music (e.g., [294]), image (e.g., [521]) and so on.

However, this causes the generation to be highly dependent on real data, e.g., there is the risk of potentially reproducing portions of the training set.

Historically, sequence prediction models have been typically implemented through Recurrent Neural Networks (RNNs), and especially through Long Short-Term Memory (LSTM) [279] or Gated Recurrent Units (GRU) [122]. The reason is that RNNs use internal states based on previous computation: inputs received at earlier time steps can affect the response to the current input, i.e., the prediction of x_t depends on the current hidden state h_t, which is in turn computed from the current input and the previous hidden state h_{t-1}. However, RNNs tend to perform worse with longer sequences [40]. LSTM is a specific RNN architecture that addresses the problem of long-term dependencies through the use of additional gates determining what to remember and what to forget at each step. GRU then simplifies the inner structure by unifying two internal gates.

2.1.4 Transformer-Based Models

Transformer-based models are neural networks based on the Transformer architecture [673]. They represent the main example of foundation models [59], because of the leading role they have been assuming in language, vision, and robotics. A Transformer is an architecture for sequential modeling that does not require recurrent or convolutional layers. Instead, it only relies on a self-attention mechanism [22] that models long-distance context without a sequential dependency. Each layer consists of multi-head attention (i.e., several self-attention mechanisms running in parallel), a feed-forward network, and residual connections. Since self-attention is agnostic to token order, a technique called positional embedding is used to capture the ordering [673].

In principle, a Transformer is nothing more than an autoregressive model: it works by predicting the current token given the previous ones (see Section 3.1.3). However, few fundamental differences exist. There is no hidden state to encode information from past inputs; the output of the Transformer only depends on its current input. As an alternative to Equation 2.6, a Transformer can also be trained by means of masked modeling: some of the input tokens are randomly masked, and the model has to learn how to reconstruct them from the entire context, and not only from the previous portions [156]. The possibility of dealing with very long sequences allows for prompting. By providing a natural language prompt in input, the model can generate the desired output, e.g., the answer to a question, classification of a given text, or a poem in a particular style [74]. This is done by simply passing the prompt in input as a text, and then leveraging the model to predict what comes next (e.g., the answer to a question):

$$\mathbf{x}' = (x'_1 \dots x'_T), x'_t \sim p_{\boldsymbol{\theta}}\big(x'_t | x'_{t-1} \dots x'_1, p_{T'} \dots p_1\big) \,\forall t \in [1, T], \tag{2.8}$$

where $p_{\boldsymbol{\theta}}$ is the Transformers network and $\mathbf{p} = (p_1 \dots p_{T'})$ is the provided prompt.

The sampling of the token from the probability distribution learned by the Transformers can happen in several ways. The greedy strategy considers sampling the highest-probable token every time. However, this can lead to a lack of diversity and repetitions. The probability distribution from which sampling can also be transformed through a *temperature* parameter. The temperature scales the differences among the various probabilities such as a temperature lower than 1 will increase the probability of the most-probable tokens (a zero temperature degenerates to greedy strategy) while a temperature higher than 1 will increase the probability of the least-probable tokens, allowing for more diversity in generation [497]. However, this might lead to the selection of tokens that are not (syntactically) appropriate for the current input. Top-k and top-p strategies [285] can reduce the token space to the k most probable ones (or to the ones that together have a probability greater than p). To get more natural and coherent solutions, contrastive search [631] uses top-k and a degeneration penalty that encourages selected tokens to be different from already generated ones. Still, all these solutions work at the token level: they cannot generate highly probable sequences if they start with low-probable tokens. To address this, Beam Search [480] maintains several hypotheses (known as the beam budget B) at each time step and eventually chooses the hypothesis with the overall highest probability. This approach, rather than focusing on single tokens (which can lead to sub-optimal or even degenerated solutions), considers the likelihood of the entire sequence [88]. However, Beam Search often focuses on a single highly valued beam, resulting in final candidates that are merely minor variations of a single sequence. Diverse Beam Search [675] proposes to overcome this issue by dividing the beam budget into G groups. It enforces diversity between different groups by penalizing candidates that share tokens with other beams. This guarantees increased diversity in the final solutions. Other variants of Beam Search have been proposed as well, to enforce a certain constraint over the output [284] or to substitute the likelihood with a self-evaluation scheme [709].

The flexibility of the sampling strategies, together with the very large amount of data available, the increasing computational power, and the parallelism induced by their architecture, has contributed to the popularity of Transformer-based architectures, as evidenced by a large number of applications in many fields. Nevertheless, it is worth noting that the computational cost of the architecture from [673] grows quadratically with the input size, imposing a clear trade-off in terms of required resources.

2.1.5 Diffusion Models

Diffusion models are a family of methods able to generate samples by gradually removing noise from a signal [615]. The most representative approach is the Denoising Diffusion Probabilistic Model (DDPM) [276]. An input $\mathbf{x_0}$ is corrupted by gradually adding noise until obtaining an output $\mathbf{x_T}$ from a predefined distribution; the model then has to reverse the process. Specifically, the forward diffusion process is describing by the following:

$$q(\mathbf{x_t}|\mathbf{x_{t-1}}) = \mathcal{N}\Big(\sqrt{1-\beta_t}\mathbf{x_{t-1}}, \beta_t \mathbf{I}\Big)\,, \tag{2.9}$$

where q is the function that adds a Gaussian noise with variance β_t. Notably, $\mathbf{x_0}$ must be normalized to a zero mean and unit variance; this, together with the scaling of the input $\mathbf{x_{t-1}}$, ensures that all $\mathbf{x_t}$ will have zero mean and unit variance, including the last $\mathbf{x_T}$, which will approximate a standard Gaussian distribution.

In other words, each timestep t corresponds to a certain noise level; $\mathbf{x_t}$ can be seen as a mixture of $\mathbf{x_0}$ with some noise $\boldsymbol{\epsilon}$ whose ratio is determined by t, thus we can directly derive any noised version $\mathbf{x_t}$ from $\mathbf{x_0}$ as follows:

$$q(\mathbf{x_t}|\mathbf{x_0}) = \mathcal{N}\big(\sqrt{\bar{\alpha}_t}\mathbf{x_0}, (1-\bar{\alpha}_t)\,\mathbf{I}\big)\,, \tag{2.10}$$

where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$ and $\alpha_t = 1-\beta_t$. Thanks to the reparameterization trick, this forward, one-shot transformation can also be seen as $\mathbf{x_t} = \sqrt{\bar{\alpha}_t}\mathbf{x_0} + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$, with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

To reverse this process, the model learns a function ϵ_θ to predict the noise component of $\mathbf{x_t}$ by minimizing its mean-squared error:

$$\min ||\boldsymbol{\epsilon} - \epsilon_\theta\big(\sqrt{\bar{\alpha}_t}\mathbf{x_0} + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t\big)||^2, \tag{2.11}$$

where $\mathbf{x_{t-1}}$ is then obtained from a diagonal Gaussian with mean as a function of $\epsilon_\theta(\mathbf{x_t}, t)$, and with a fixed [276] or learned [470] variance $\boldsymbol{\sigma_t}$. In other words, it learns to associate points from a predefined random distribution with real data through iterative denoising. At inference time, a diffusion model can iteratively generate a new sample by starting from pure random noise $\mathbf{x'_T} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$:

$$\mathbf{x'_{t-1}} = \sqrt{\bar{\alpha}_{t-1}}\frac{\mathbf{x'_t} - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(\mathbf{x_t}, t)}{\sqrt{\bar{\alpha}_t}} + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\epsilon_\theta(\mathbf{x_t}, t) + \sigma_t\boldsymbol{\epsilon_t}, \tag{2.12}$$

where the first component is the predicted $\mathbf{x_0}$, the second reapplies the predicted noise until $t-1$, and the third introduces additional Gaussian random noise through $\boldsymbol{\epsilon_t} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ [616]. While in theory we might use the first part alone and directly generate $\mathbf{x_0}$ from pure noise, mimicking the forward process allows the model to adjust its predictions, leading to better results.

The generation can also be conditioned by simply modifying the noise perturbation so that it depends on the conditional information. However, this iterative sampling process might potentially lead to slow generation; a proposed solution is to induce self-consistency, i.e., ensuring that points on the same trajectory map to the same initial ones [620]. In this way, the output can be obtained in a single step.

The aforementioned diffusion process is similar to that followed by score-based generative models [617, 618]. Instead of noise, here a model is trained to learn the score, i.e., the gradient of the log probability density with respect to real data. The samples are then obtained using Langevin dynamics [689]. Despite the differences, both of them can be seen as specific, discrete cases of Stochastic Differential Equations [619].

2.1.6 Input-Based Methods

Finally, we introduce two approaches to sample results from (pre-trained) DL models. The first is about carefully selecting or optimizing the input to a generative model (e.g., the latent vector or the text prompt) so that to obtain the desired output. Assuming the desired output is $\mathbf{x}' \sim p_{\boldsymbol{\theta}}(\mathbf{z_T})$, the target input $\mathbf{z_T}$ is obtained starting from $\mathbf{z_0}$ after T steps of gradient ascent according to Equation 2.13:

$$\mathbf{z_t} = \mathbf{z_{t-1}} + \eta \nabla_{\mathbf{z_{t-1}}} g(p_{\boldsymbol{\theta}}(\mathbf{z_{t-1}}))\,, \tag{2.13}$$

with $g(\cdot)$ as an objective function that depends on the current version of the output and some other neural networks. One possible solution is represented by VQGAN-CLIP [139], which considers a function g to estimate how close the CLIP [514] embedding of the generated image is to the embedding of the user-provided description.

The second approach is about optimizing the input so that it directly approximates the desired output. Such methods rely on losses that are usually based on features learned by neural networks. A notable example is Deepdream [451]: given an input $\mathbf{x_0}$ and a neural network $q_{\boldsymbol{\phi}}(\mathbf{x})$, the final output $\mathbf{x_T}$ is obtained after T steps of gradient ascent according to Equation 2.14:

$$\mathbf{x_t} = \mathbf{x_{t-1}} + \eta \nabla_{\mathbf{x_{t-1}}} \frac{\sum_{i=1}^{D} q_{\boldsymbol{\phi}}^{LC}(\mathbf{x_{t-1}})_i}{D}, \tag{2.14}$$

where q_ϕ^L is the activation of layer(s) L with total length D and η is the step size of the gradient ascent.

While the two approaches are technically different, both of them aim at obtaining better outputs by exploiting the knowledge of a pre-trained model through the optimization of the inputs.

2.2 Reinforcement Learning

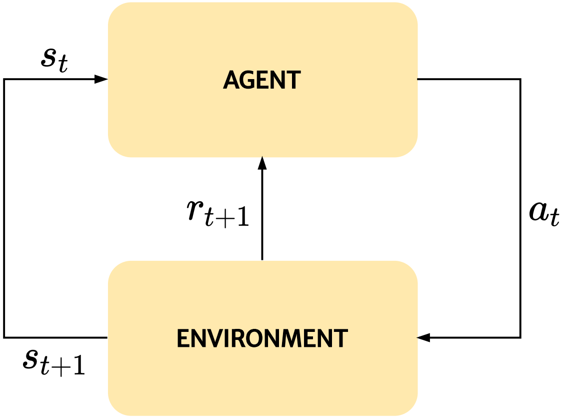


Figure 2.2: The canonical reinforcement learning framework: at each timestep t, the agent performs an action a_t based on the current state s_t, which is a representation of the environment. Upon the execution of the action, the agent finds itself in a new state s_{t+1}, and receives a reward r_{t+1}.

Reinforcement learning is a machine learning paradigm that consists of learning an action based on a current representation of the environment in order to maximize a numerical signal, i.e., the *reward* over time [639]. More formally, at each time step t, an *agent* receives the current *state* s_t from the *environment*, then it performs an *action* a_t and observes the reward r_{t+1} and the new state s_{t+1}. Figure 2.2 summarizes the process. The learning process aims to teach the agent to act in order to maximize the *cumulative return* $G_t = \sum_{i=1}^{T-t} \gamma^{i-t-1} r_{i+t}$, i.e., a discounted sum of future rewards. Deep learning is also used to learn and approximate a *policy* π, i.e., the mapping from states to action probabilities, or a *value function*, i.e., the mapping from states (or state-action pairs) to expected cumulative rewards. In this case, we refer to it as deep reinforcement learning.

Algorithms that aim to learn a value function are called *value function approximation methods*. Given the actual state-value function $v_\pi(s)$ and the actual action-value function $q_\pi(s, a)$ under the current policy π, the goal of such methods is to learn either $\hat{v}(s, \mathbf{w})$ or $\hat{q}(s, a, \mathbf{w})$, where $\mathbf{w}$ are the weights of the neural network used as the approximator, whose cardinality should be much smaller than the number of possible states. These neural networks are trained to minimize the prediction error. In the case of a Monte Carlo method, i.e., with the value estimation happening at the completion of the episode, the target is G_t. In the case of a temporal difference method, i.e., with the value estimation happening after N steps, the target bootstraps the value function of the N-state. A notable case is TD(0), where the target for the state-value function approximator becomes $r_{t+1} + \gamma\hat{v}(s_{t+1}, \mathbf{w})$ or $r_{t+1} + \gamma\hat{q}(s_{t+1}, a_{t+1}, \mathbf{w})$.

While one might be interested in simply solving the prediction problem, i.e., the estimation of the value function given a fixed policy, SARSA [639] leverages this method to address the more complex control problem, i.e., the derivation of an optimal policy π_*. SARSA induces its current policy from the action-value function approximator $\hat{q}(s, a, \mathbf{w})$ by computing the optimal action according to it:

$$a_t^* = \underset{a}{\texttt{argmax}}\, \hat{q}(s_t, a, \mathbf{w})\,. \tag{2.15}$$

Always choosing the optimal action according to the current value function is called *greedy* policy. In order to better balance exploitation and exploration, the ϵ-greedy policy is usually adopted, where the greedy action is chosen with $1 - \epsilon$ probability and a random action is chosen with ϵ probability. Since the approximated action-value function is updated based on the action taken by the agent, SARSA is an *on-policy* method.

However, on-policy methods are sensitive to the exploration strategy and are bound to the actions taken, leading to slower convergence. On the contrary, *off-policy* methods separate the behavior policy (the one used to act in the environment) from the target policy (the one used to update action-value functions), allowing for different exploration strategies and faster convergence. The most important off-policy method is Q-learning [683]. Instead of relying on the action taken by the agent to compute the action-value update, it considers the best action according to its current policy: the target then becomes $r_{t+1} + \gamma \max_a \hat{q}(s_{t+1}, a, \mathbf{w})$. In this way, the update is completely independent of the policy followed during exploration, enabling early convergence proofs. Q-learning is at the basis of one of the most successful deep RL algorithms, Deep Q-Network (DQN) [449]. DQN uses a deep Convolutional Neural Network (CNN) to represent the Q-function $\hat{q}(s, a, \mathbf{w})$, and trains it to minimize the prediction error of the Q-learning target. In addition, DQN introduces other technical novelties such as reward clipping, where each positive reward is clipped to $+1$ and each negative reward is clipped to -1; error clipping, where the prediction error is clipped in the interval $[-1, +1]$; and experience replay, where the agent's experience is stored in a memory, and training updates are performed over random samples of experiences (which is made possible by the off-policy nature of DQN). In this way, successive updates are uncorrelated, reducing their high variance, and each experience can be used multiple times, making the overall learning more efficient.

Several variants of DQN have been proposed over the years. For example, double DQN [670] considers a copy of the Q-network for the action selection in the target update whose parameters are slowly updated to match those of the Q-network. This helps reduce the maximum value overestimation. Another possibility is to modify the experience replay sampling scheme by prioritizing experience based on the temporal-difference error (since this should lead to faster learning) [564].

While effective, value function approximation methods have some limitations. If the optimal policy is deterministic but requires high exploration to find it, strategies like ϵ-greedy cannot approach it. On the contrary, if the optimal policy is not deterministic, there is no natural way to learn it. Finally, small changes in the estimated value function can lead to large, discontinuous changes in the policy, making the learning process unstable. To deal with this issue, it is possible to let the neural network directly learn a policy, thus making it return the action probability distribution given the current state. We refer to Section 2.2.1 for a more in-depth overview of policy gradient methods.

The RL community has developed a variety of solutions to address the specific theoretical and practical problems emerging from its simple formulation. For example, if the reward signal is not known, inverse reinforcement learning (IRL) [464] is used to learn it from observed experience. Intrinsic motivation [604, 398], e.g., curiosity [493] can be used to deal with sparse rewards and encourage the agent to explore further. Imagination-based RL (detailed in Section 2.2.2) is a solution that allows to train an agent, reducing at the same time the need for interaction with the environment. Hierarchical RL [492] allows to manage more complex problems by decomposing them into sub-tasks and working at different levels of abstraction. RL is not only used for training a single agent, but also in multi-agent scenarios [725]. Finally, generalization in RL [345] is currently an area of great interest for the community; in Section 3.2.3 we survey its current state of the art.

2.2.1 Policy Gradient Methods

Policy gradient methods learn a parameterized policy $\pi(a|s, \boldsymbol{\theta})$ that directly predicts actions without consulting a value function. This helps overcome the limitations of action-value function methods, providing a more straightforward and intuitive way to model a reinforcement learning agent. However, there is no direct target to approximate as in action-value function methods. Ideally, we ought to have some scalar performance measure $J(\boldsymbol{\theta})$ with respect to the policy parameter that estimates the performance of our current policy. If so, we could update the parameters via gradient ascent. While we might not have $J(\boldsymbol{\theta})$, what we really need is its gradient $\nabla J(\boldsymbol{\theta})$. The policy gradient theorem [640] provides an analytic expression for the gradient of performance with respect to the policy parameter:

$$\nabla J(\boldsymbol{\theta}) \propto \sum_{s} \mu(s) \sum_{a} q_{\pi}(s,a)\, \nabla \pi(a|s,\boldsymbol{\theta})\,, \tag{2.16}$$

where $\mu(s)$ is the on-policy distribution under π. We refer to the original paper for the entire derivation.

REINFORCE [692] is the simplest policy gradient method. Starting from the policy gradient theorem, it is possible to derive its update rule presented in Equation 2.17.

$$\boldsymbol{\theta_{t+1}} = \boldsymbol{\theta_t} + \eta G_t \nabla \ln \pi(a_t|s_t, \boldsymbol{\theta_t}), \tag{2.17}$$

where η is the step size of the gradient ascent update. Since G_t is considered, REINFORCE is a Monte Carlo method.

However, REINFORCE tends to suffer from high variance, which might lead to slow learning. To reduce the variance, REINFORCE with baseline introduces a comparison of the action value (in this case, G_t) with an arbitrary function independent from a_t: in particular, an estimate of the state value $\hat{v}(s_t, \mathbf{w})$ can be used as a baseline to reduce the action value to the actual advantage of choosing a_t in state s_t:

$$\boldsymbol{\theta_{t+1}} = \boldsymbol{\theta_t} + \eta\,(G_t - \hat{v}(s_t, \mathbf{w}))\,\nabla \ln \pi(a_t|s_t, \boldsymbol{\theta_t}). \tag{2.18}$$

Closely resembling the value function methods introduced above, it is also possible to replace the cumulative return G_t with the TD(0) target $r_{t+1} + \gamma\hat{v}(s_{t+1}, \mathbf{w})$, to gain the usual advantages of temporal-difference methods over Monte Carlo methods. The policy gradient algorithm that leverages this change is called one-step actor-critic since it performs updates after one step and the state-value approximator is used to assess the actions, leading to the name *critic*. The overall update rule thus becomes:

$$\boldsymbol{\theta_{t+1}} = \boldsymbol{\theta_t} + \eta\,(r_{t+1} + \gamma\hat{v}(s_{t+1}, \mathbf{w}) - \hat{v}(s_t, \mathbf{w}))\,\nabla \ln \pi(a_t|s_t, \boldsymbol{\theta_t}). \tag{2.19}$$

It is of course possible to generalize this to n-step methods.

Due to their advantages, several other actor-critic methods have been proposed in recent years. For example, Asynchronous Advantage Actor-Critic (A3C) [450] considers parallel actor-learners updated asynchronously to stabilize training. Deep Deterministic Policy Gradient (DDPG) [395] concurrently learns a Q-function and a policy by using off-policy data to learn the Q-function and the Q-function to learn the policy. Soft Actor-Critic (SAC) [253] leverages entropy regularization to optimize stochastic, continuous policies in an off-policy way. Finally, Trust Region Policy Optimization (TRPO) [571] and Proximal Policy Optimization (PPO) [573] update policies by taking the largest step possible to improve performance without stepping too far from the old policies.

In particular, PPO has gained great popularity thanks to its theoretical guarantees and adoption by reinforcement learning from human feedback (see Section 2.3). TRPO tries to prevent the new policy from moving too far from its old version by introducing a constraint about the KL divergence between them. However, its second-order method makes the optimization complex and not always feasible. Instead, Schulman et al. [573] propose to substitute the hard constraint with a more tractable objective based on a clipped surrogate objective.

The PPO overall loss is defined as follows:

$$L(\theta) = \hat{\mathbb{E}}_t \big[L_t^{CLIP}(\theta) - c_v L_t^{VF}(\theta) + c_e S[\pi_\theta](s_t) \big] \,, \tag{2.20}$$

with

$$L_t^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\min \left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \hat{A}_t, clip\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}, 1-\epsilon, 1+\epsilon \right) \hat{A}_t \right) \right] \tag{2.21}$$

the clipped surrogate objective that modifies the policy in the right direction while preventing too large changes, and

$$L_t^{VF}(\theta) = \hat{\mathbb{E}}_t \Big[\big(V_\theta(s_t) - V_t^{target} \big)^2 \Big] \,. \tag{2.22}$$

V_θ is the function used to estimate the value of the current state. $S[\pi_\theta]$ is an entropy bonus that prevents the policy from collapsing over one or few actions, while c_v and c_e are tunable coefficients that scale down the relative losses. Finally, the advantage can be estimated in multiple ways. The two most common choices are the one-step temporal difference error [147]:

$$\delta_t = r_{t+1} + \gamma V_\theta(s_{t+1}) - V_\theta(s_t) \,, \tag{2.23}$$

and the Generalized Advantage Estimation (GAE) [571]:

$$\delta_t = \sum_{i=0}^{T-t} (\gamma\lambda)^i \left(r_{t+i+1} + \gamma V_\theta(s_{t+i+1}) - V_\theta(s_{t+i})\right). \tag{2.24}$$

Finally, the value function is trained to approximate

$$V_t^{target} = \delta_t + V_\theta(s_t). \tag{2.25}$$

2.2.2 Imagination-Based Reinforcement Learning

Usually, RL requires a very large amount of collected experience, especially compared to the one required by humans [660], limiting its applications to real-world tasks. Model-based RL [639] constitutes a promising direction toward sample efficiency. It requires learning a world model capable of predicting the next states and rewards conditioned on actions. This allows the agent to plan [569] or build additional training trajectories [252] by predicting the consequences of its actions. In particular, recent imagination-based methods [261, 440] have shown remarkable performance simply by learning from imagined episodes within a learned latent space, limiting the interaction with the real environment to that necessary to train a working world model.

World models represent a compact and learned version of the environment capable of predicting imagined future trajectories [638]. When the inputs are high-dimensional observations $\mathbf{o_t}$ (i.e., images), Dreamer [259, 260, 261] represents the current state of the art due to its ability to learn compact latent states $\mathbf{z_t}$. In general, Dreamer world model consists of the following components:

$$\begin{array}{ll}
\text{Recurrent model:} & \mathbf{h_t} = f_{\boldsymbol{\theta}}(\mathbf{h_{t-1}}, \mathbf{z_{t-1}}, a_{t-1}) \\
\text{Encoder model:} & \mathbf{z_t} \sim q_{\boldsymbol{\theta}}(\mathbf{z_t} | \mathbf{h_t}, \mathbf{o_t}) \\
\text{Transition predictor:} & \mathbf{\hat{z}_t} \sim p_{\boldsymbol{\theta}}(\mathbf{\hat{z}_t} | \mathbf{h_t}) \\
\text{Reward predictor:} & \hat{r}_t \sim p_{\boldsymbol{\theta}}(\hat{r}_t | \mathbf{h_t}, \mathbf{z_t}) \\
\text{Continue predictor:} & \hat{c}_t \sim p_{\boldsymbol{\theta}}(\hat{c}_t | \mathbf{h_t}, \mathbf{z_t}) \\
\text{Decoder model:} & \mathbf{\hat{o}_t} \sim p_{\boldsymbol{\theta}}(\mathbf{\hat{o}_t} | \mathbf{h_t}, \mathbf{z_t})
\end{array}$$

The deterministic recurrent state $\mathbf{h_t}$ is predicted by a Gated Recurrent Unit [114], while the encoder and decoder models use convolutional neural networks for visual observations. Overall, the Recurrent State-Space Model [258], an architecture that contains recurrent layers, encoders, and transition components, learns to predict the next state only from the current one and the action, while also allowing for correct reward, continuation bit, and image reconstructions.

While $\mathbf{z_t}$ was originally parameterized through a multivariate normal distribution, more recent works [260, 261] consider a discrete latent state. In particular, they use a vector of C one-hot encoded categorical variables (i.e., a very sparse binary vector). The authors of [261] parameterize this categorical distribution as a mixture of 1% uniform and 99% neural network output. Moreover, instead of regressing the rewards via squared error, they propose a learning scheme based on two transformations: first, the rewards are symlog-transformed [685]; then, they are two-hot encoded, i.e., converted into a vector of K values where $K-2$ are 0, and the remaining, consecutive two are positive weights whose sum is 1. The K values correspond to equally spaced buckets, so we reconstruct the original reward by multiplying the vector with the bucket values. This solution facilitates the learning process, especially in environments with very sparse rewards.

Overall, given a sequence of inputs $\{\mathbf{o_{0:T-1}}, a_{0:T-1}, r_{1:T}, c_{1:T}\}$, the world model is trained to minimize the following loss:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{q_{\boldsymbol{\theta}}}\left[\sum_{t=1}^{T}(\mathcal{L}_{pred}(\boldsymbol{\theta}) + \beta_1 \mathcal{L}_{dyn}(\boldsymbol{\theta}) + \beta_2 \mathcal{L}_{rep}(\boldsymbol{\theta}))\right] \tag{2.26}$$

where $\mathcal{L}_{pred}$ trains the decoder model via mean squared error loss, the reward predictor via categorical cross-entropy loss, and the continue predictor via binary cross-entropy loss; while $\mathcal{L}_{dyn}$ and $\mathcal{L}_{rep}$ consider the same Kullback-Leibler (KL) divergence between $q_{\boldsymbol{\theta}}(\mathbf{z_t}|\mathbf{h_t}, \mathbf{o_t})$ and $p_{\boldsymbol{\theta}}(\mathbf{\hat{z}_t}|\mathbf{h_t})$, but using the stop-gradient operator on the former for the first loss and on the latter for the second loss. Moreover, free bits [342] are employed to clip the KL divergence below the value of 1 nat. Finally, β_1 and β_2 are scaling factors necessary to encourage learning an accurate prior over increasing posterior entropy [260].

Leveraging the world model detailed above, a policy $\pi_{\boldsymbol{\phi}}(a_t|\mathbf{s_t})$ can be learned by acting only in the latent space of imagination: given a compact latent state $\mathbf{\hat{s}_t^{im}} = \left(\mathbf{h_t^{im}}, \mathbf{\hat{z}_t^{im}}\right)$, the agent selects an action a_t, returns it to the world model, and receives $\hat{r}_{t+1}$, $\hat{c}_{t+1}$, and $\mathbf{\hat{s}_{t+1}^{im}}$. Furthermore, a critic $v_{\boldsymbol{\phi}}(v_t|\mathbf{s_t})$ is simultaneously learned to predict the state-value function v_t. This process is repeated until a fixed imagination horizon is reached and the policy can be learned from the imagined experience as it would have done by acting in the real environment. The agent can be trained on the collected trajectories either by direct reward optimization (leveraging the differentiability of the trajectory construction and back-propagating through the reward model [259]) or by using a model-free policy gradient method, e.g., REINFORCE (see Section 2.2.1).

2.3 Reinforcement Learning for Generative Modeling

Due to its adherence to the formal framework of Markov decision processes [639], RL can be used as a solution to the generative modeling problem in the case of sequential tasks [21], e.g., text generation or stroke painting. The generative model plays the role of the agent, returning "generative" actions a_t. For example, such actions can be single tokens, e.g., words or notes, or image layers to be superimposed. The current version of the generated output $\mathbf{x_t}$ represents the state $\mathbf{s_t}$ (potentially with additional information), and it is obtained as a function of all the previous actions:

$$\mathbf{x_t} = f(a_{t-1} \dots a_1) \,, \tag{2.27}$$

where $\mathbf{x_T}$ represents the final output produced by the generative model, and $f(\cdot)$ is a function that "composes" the actions together, e.g., by appending the actions one after the other to form a text or by iteratively applying changes to a picture. Note that $f(\cdot)$ can also be an identity function if the generative agent returns a completed output in one shot. Finally, the reward r_{t+1} measures the "quality" of the current output. Figure 2.3 summarizes the entire process.

It is possible to identify three fundamental design aspects: the implementation of the agent itself, e.g., diffusion model or Transformer; the definition of the system's dynamics, i.e., the transition between one state to another through function $f(\cdot)$; and the choice of the reward structure. The first two depend on the task to be solved, e.g., music generation with LSTM composing one note after the other or painting with CNN superimposing subsequent strokes, and the final sampling scheme to generate new outcomes depends on the chosen architecture. The third one is instead responsible for the actual learning, together with the choice of the RL algorithm.

The main advantage of this formulation is that rewards in reinforcement learning can be non-differentiable. While self-supervised learning requires the objective function to be fully differentiable for each generative model's output, limiting its expressiveness, the reward function can be any function. This allows us to design ad-hoc rewards such as sets of rules to satisfy, testing-time metrics, or domain-specific properties. In addition, it also makes it possible to have an objective for the entire $\mathbf{x} = f(a_T \dots a_1)$ and backpropagate it to each single generative step a_t.

The most famous example of an approach that leverages this property is Reinforcement Learning from Human Feedback (RLHF). RLHF is a training scheme that aims to make the RL agent maximize human preferences. While

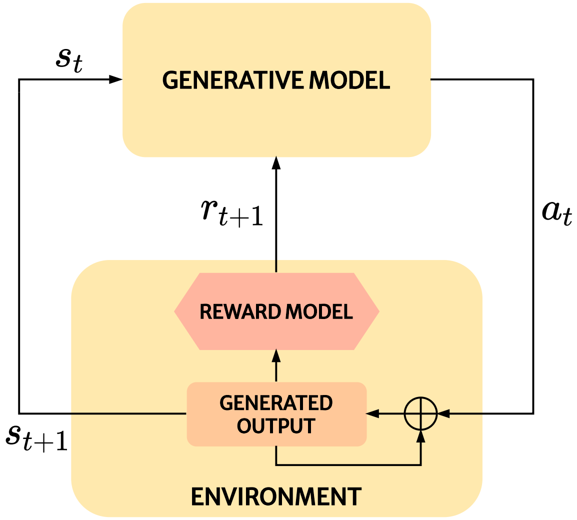


Figure 2.3: The reinforcement learning framework for generative modeling: at each timestep t, the generative model (i.e., the agent) generates an action a_t based on the current description of the generated output (i.e., current state) s_t, which updates the current description of the generated output to s_{t+1}, and receives a reward r_{t+1} related to it.

firstly proposed for game-like environments [118], now it is widely adopted as the last training stage in a vast number of popular large language models.

Given a pre-trained language model, RLHF is articulated in different steps. First, a preliminary step where demonstration data are collected from human labelers and the language model is fine-tuned with self-supervised learning is usually present [482]. Then, three steps are repeated multiple times: human feedback collection; reward model training; and language model training with reinforcement learning [626]. The human feedback collection requires sampling a prompt $\mathbf{p}$ from the training set and letting the language model produce multiple outputs $\mathbf{x_i}, i = 1 \ldots K$. A human labeler is entitled to rank such outputs from best to worst. Then, the collected data are used to train the reward model. Since the reward model must return a numerical score $r_\phi(\mathbf{x}, \mathbf{p})$ for each input, we need a way to teach the reward model to assign higher scores for best-ranked outputs and lower scores for worst-ranked outputs. Thus, the reward model is trained to optimize the following objective:

$$\min -\frac{1}{\binom{K}{2}} \mathbb{E}_{(\mathbf{p},\mathbf{x_w},\mathbf{x_l})\sim X} \left[\log \sigma(r_\phi(\mathbf{x_w}, \mathbf{p}) - r_\phi(\mathbf{x_l}, \mathbf{p}))\right], \tag{2.28}$$

with $\mathbf{x_w}$ preferred over $\mathbf{x_l}$, and $\sigma(\cdot)$ as the sigmoid function.

Finally, the third step consists of sampling a new prompt $\mathbf{p}$ from the training set, making the language model generate an output $\mathbf{x}$, getting the output score $r_\phi(\mathbf{x}, \mathbf{p})$ from the reward model, and training the language model with PPO. To avoid moving too far away from the original language model, a per-token KL penalty is commonly added to the predicted reward:

$$r(\mathbf{x}, \mathbf{p}) = r_\phi(\mathbf{x}, \mathbf{p}) - \beta D_{KL}\left(\pi_\theta(\mathbf{x}|\mathbf{p}) \mid\mid \pi_{\theta^{SFT}}(\mathbf{x}|\mathbf{p})\right), \tag{2.29}$$

where $\pi_{\theta^{SFT}}$ is the language model after the (self-) supervised fine tuning, and β is the penalty scaling factor. Once the final rewards are computed, PPO (see Section 2.2.1) is used with some minor implementation changes [297].

While highly effective, RLHF suffers from several open problems [96], e.g., how we can get correct, unbiased, and well-representative human feedback and how we can make this process resource-efficient. Moreover, RLHF tends to be complex and unstable due to the need to train the reward model concurrently and the fact it is based on reinforcement learning. To address these two issues, Direct Preference Optimization (DPO) [515] directly optimizes the language model without explicit reward modeling or reinforcement learning. DPO implicitly performs reward maximization with a KL-divergence penalty through the following loss:

$$\min -\mathbb{E}_{(\mathbf{p},\mathbf{x_w},\mathbf{x_l})\sim X}\left[\log \sigma\left(\beta \log \frac{\pi_\theta(\mathbf{x_w}|\mathbf{p})}{\pi_{\theta^{SFT}}(\mathbf{x_w}|\mathbf{p})} - \beta \log \frac{\pi_\theta(\mathbf{x_l}|\mathbf{p})}{\pi_{\theta^{SFT}}(\mathbf{x_l}|\mathbf{p})}\right)\right]. \tag{2.30}$$

In other words, DPO fits an implicit reward whose optimal policy is the language model itself.

3 Related Work

In this chapter we discuss the related work in the fields that are relevant for this thesis. Section 3.1 reviews the generative deep learning models introduced in Section 2.1, with a specific focus on how they relate to creativity theories. Then, Section 3.2 discusses the state of the art in RL. Section 3.3 explores the current research that involves reinforcement learning for generative AI, highlighting the main opportunities that arise from their combination. Finally, Section 3.4 introduces the literature at the interface of creativity and generative deep learning, also providing an overview of its societal implications.

3.1 Generative Deep Learning

This section aims to present and critically discuss the state of the art in generative deep learning from the point of view of machine creativity. In particular, we focus on the product dimension of creativity. We underpin our analysis on Boden's three criteria (i.e., value, novelty, and surprise) since they have been widely adopted, together with the three forms of creativity (i.e., combinatorial, exploratory, and transformational), as introduced in Section 1.2. For each of the generative modeling families from Section 2.1, we present some relevant *examples of models*; potential *applications*; and a *critical discussion* evaluating the level of machine creativity considering the definitions above. Finally, we conclude by analyzing potential directions in order to make those models more *creativity-oriented*.

As a final remark, it is worth noting that we limit our examples to the Arts (e.g., poems, music, or paintings). Indeed, generative learning can be applied to design [211, 422]; game content generation (see [401] for a comprehensive survey); recipes [453, 672]; scientific discovery [130, 566]; and in general to any activity, which has a non-trivial solution [74].

3.1.1 Variational Autoencoders

Examples of Models. Several models based on VAEs have been proposed [340] in recent years. Our focus is on those relevant to our discussion on machine creativity. In β-VAE [271], a parameter β is used to scale the magnitude of the regularization loss, which allows a better disentanglement of the latent space [83]. Another example is VAE-GAN [361], which merges VAE and GAN (see Section 2.1.2). This is done by treating the decoder as the generator of the GAN, thus training it through the GAN loss function. This leads to the generation of substantially less blurred images. Similarly, Adversarially Learned Inference (ALI) [170] merges VAE and GAN by asking the discriminator to distinguish between pairs of real data (and their latent representations) and pairs of sampled representations and synthetic data. Adversarial Autoencoders (AAE) [424] substitute the regularization loss with a discriminative signal, where the discriminator has to distinguish between random latent samples and encoded latent vectors. Another way to address the problem of "sample blurriness" is with PixelVAE [244], where the autoregressive PixelCNN [667, 668] is used as the decoder. In [64], the encoder learns to produce a latent sentence representation to deal with sequential data such as texts, where generation requires more steps. In contrast, the recurrent neural network RNN-based decoder learns to reproduce it word after word. However, VAE can also generate text through convolution and deconvolution [579]. To solve the problem of low-density regions, the authors of [14] propose an energy-based model called noise contrastive prior (NCP), trained by contrasting samples from the aggregate posterior to samples from a base prior. Finally, another interesting model is Vector Quantised-VAE (VQ-VAE) [669]; in this case, the encoder outputs discrete, rather than continuous, codes, and the prior is learned rather than static.

Applications. VAEs can be used for semi-supervised classification to provide an auxiliary objective, improving the data efficiency [341, 413]; to perform iterative reasoning about objects in a scene [177]; to model the latent dynamics of an environment [684]. Of course, VAEs have also been used to generate synthetic data, possibly with conditional generation. For example, a layered foreground-background generative model can be used to generate images based on both the latent representation and a representation of the attributes [713]. In [251] the latent space of a VAE is trained on chemical structures by means of gradient-based optimization toward certain properties (see Section 3.1.6). AAEs have also been applied to the same problem [326]. Finally, another interesting application of VAE is Deep Recurrent Attentive Writer (DRAW) [234]. DRAW constructs scenes iteratively by accumulating changes emitted by the decoder (then given to the encoder in input). This allows for iterative self-corrections and a more natural form of image construction. RNNs and the attention mechanism are used to consider previous generations and to decide at each time step where to focus attention, respectively.

Critical Discussion. Models based on VAEs can be considered as an example of exploratory creativity. The latent space is learned with the goal of representing data in the most accurate way. The random sampling performed during generation is therefore an exploration of that space: regions not seen during the training can be reached as well, even though they can lead to poor generation [14] and some more complex variants may be needed, as discussed. On the other hand, there is no guarantee that the results will be valuable, novel, or surprising. There is no guarantee that the generation from random sampling is of good quality or diverse from training data. Indeed, given their characteristics, VAEs discourage novelty in a sense. In particular, diversity could be achieved in theory using VAEs and gradient-based optimization techniques, such as those presented in [251], with novelty and surprise as target properties. We will discuss these aspects in Section 3.1.7.

3.1.2 Generative Adversarial Networks

Examples of Models. Several variants of GANs have been proposed, and the number is still growing. An in-depth survey is [241]. Indeed, several refinements have been proposed in the past years, such as using deep convolutional networks [512] or self-attention [724], incrementally growing the networks [330], or scaling the model parameters [70]. In the following, we present examples that are relevant to the issue of machine creativity.

The problem of non-meaningful representation has been addressed in different ways. For instance, InfoGAN [109] adds a latent code $\mathbf{c}$ to $\mathbf{z}$. An auxiliary model learns to predict $\mathbf{c}$ given the sample generated by means of it. In this way, it can learn disentangled representations in a completely unsupervised manner. Another possibility is Bidirectional GAN (BiGAN) [162]. In order to include an inverse mapping from data to latent representation, an encoder is added to the architecture. The discriminator is then trained to distinguish between pairs of random noise and synthetic data and pairs of real data and latent encoding. It is possible to condition the generation through a target content [474], a text [526], or even an image [304]. To do so, it is sufficient to use the conditional information as an input for both generator and discriminator [447]. Similarly, image-to-image translation is also possible without paired datasets. CycleGAN [740] trains two generators (from one domain to another, and vice versa) so that each of them produces images both from the target domain and correctly reconstructed by the counterpart.

In StyleGAN [331, 332], the generator architecture is re-designed to control the image synthesis process. At each layer, the style of the image is adjusted based on the latent code (the specific intermediate code to control each layer is provided by a non-linear mapping network). This allows for the automatic separation of high-level attributes from stochastic variations in the generated images. It also allows for mixing regularization, where two latent codes are used alternatively to guide the generator. StyleGAN-V [606] builds on top of it to learn to produce videos by only using a few frames of it. To generate longer and more realistic motions, a two-stage approach can be used as well: first, a low-resolution generator is adversarially trained on long sequences; then, a high-resolution generator transforms a portion of the produced, low-resolution video in a high-resolution one [71].

Finally, it is also worth mentioning variants that adapt GANs to sequential tasks (e.g., text generation). Since GANs require the generator to be differentiable, they cannot generate discrete data [225]. However, several techniques have been proposed to avoid this problem. One possibility is to transform the discrete generation into a continuous one. Music can be processed like an image by considering its waveform (as in WaveGAN [161] and GANSynth [176]) or its musical score composed of tracks and bars (as in MuseGAN [163]). Music in a desired style can be obtained through conditional inputs. Another possibility is to consider a soft-argmax function as an approximation of the inference for each step [729]. TextGAN [730] uses it together with feature matching to learn the production of sentences. In place of the discriminative signal, it uses the difference between the latent feature distributions of real and synthetic sentences learned by the discriminator. Another solution is to transform the GAN into a reinforcement learning framework; we explore this solution in Section 3.3.1. Finally, Gumbel-softmax relaxation [309, 419] can also be used, as in Relational GAN (RelGAN) [471]. Controlled TExt generation Relational Memory GAN (CTERM-GAN) [47] builds on the latter by also conditioning the generator on an external embedding input. In addition, it uses both a syntactic discriminator to predict whether a sentence is correct and a semantic discriminator to infer if a sentence is coherent with the external input.

Applications. GANs have been applied to a variety of practical problems in several application scenarios. They have been widely used for semi-supervised learning [473]; for generating adversarial examples [707] to better train image classifiers [420]; and, in general, in computer vision (see [682] for a detailed discussion). The generative power of GANs has also found its place in recommender systems [149] to generate fashion items; in science and chemistry [434, 454]. Of course, its ability to generate high-quality samples has been exploited in many other areas, from anime design [320] and 3D object modeling [699] to photo-realistic consequences of climate change [567]. Conditional inputs also allow the production of artistic works by controlling stylistic properties such as genre [644] or influencer [120]. Finally, the most famous example of the artistic power of GAN is the collection of paintings by Obvious, a French art collective [674]; one of their works has been sold to more than 400,000 dollars[1].

Critical Discussion. GANs are difficult to evaluate from a machine creativity perspective. The generator does not receive the original works as input, so it samples from a conceptual space that is built only indirectly from them. In rare cases, this can also lead to a different conceptual space (with respect to the original one) and so to transformational creativity, but it typically leads to exploratory creativity. Since the goal is to learn to generate seemingly real artifacts from a latent distribution, it will likely approximate the real one. Still, it is possible to identify potential creative solutions among those generated by the model.

An advantage of GANs is the presence of a *recognition* network, i.e., the discriminator, trained to recognize real (valuable) works. This is important for two reasons. It suffices for being able to define GANs *appreciative* [130], which is a central sub-task of creativity [10, 203]. In addition, it allows us to consider their products as valuable, as it is in a sense their intrinsic objective. However, there is no guarantee that they will also be new and surprising. Nevertheless, it seems possible to extend a GAN objective to include such properties as well (see Section 3.1.7 for a discussion).

3.1.3 Sequence Prediction Models

Examples of Models. RNNs can be used to model joint probabilities of characters (Char-RNN) [329]; words [506]; phonemes [288]; syllables [742]; and even tokens from transcriptions of folk music (Folk-RNN) [628]. They can also receive conditional inputs like the encoding of the previous lines [728]. Richer architectures that combine models focusing on different properties can be used to generate more complex text, e.g., poetry based on pentameter and rhymes [363]. Finally, sequence modeling can also be combined with reinforcement learning, as we will see in Section 3.3.

[1] Fun fact: the sold painting is called *Portrait of Edmond De Belamy* because Belamy sounds like *bel ami*, a sort of French translation of... *Goodfellow.*

Due to the difficulties in working with long sequences, results in tasks like narrative generation are affected by a lack of coherence [540]. Many approaches have been proposed to address this problem. For instance, stories can be generated in terms of events [427] (i.e., tuples with subject, verb, object, and an additional *wildcard*) by an encoder-decoder RNN (also known as Sequence-to-Sequence, see [636]); events are modeled by another encoder-decoder RNN. Instead of events, it is also possible to focus on entities (i.e., vectors representing characters) [123].

Sequence prediction models are also used for domains not commonly modeled as sequences, like images. Image modeling can be defined as a discrete problem through a joint distribution of pixels: the model learns to predict the next pixel given all the previously generated ones. It starts at the top left pixel and then proceeds towards the bottom right. The two seminal architectures for sequence prediction of images are PixelRNN and PixelCNN [667]. The former is a two-dimensional RNN (based on rows or diagonals). The latter is a convolutional neural network (CNN) with an additional fixed dependency range (i.e., the convolution filters are masked to only use information about pixels above and to the left of the current one). To obtain better results, gated activation units can be used in place of rectified linear units between the masked convolutions; conditional inputs encoding high-level image descriptions can be used as well [668]. Notably, the Gated PixelCNN architecture can also be used for other types of data: WaveNet [666] implements it to generate audio based on the waveform, possibly guiding the generation with conditional inputs.

While intuitive in terms of architecture, RNNs are limited by the vanishing gradient problem and non-parallelizability in the time dimension [366]. Very recent works explore solutions to tackle these issues by means of structured state spaces [237] and a combination of RNNs and Transformers [499].

Applications. As discussed, sequence prediction models have been used to learn to write poems or stories (by predicting a character, syllable, or word after the other); to compose music (by predicting a note or a waveform after the other); to draw images (by predicting a pixel after the other). In general, they can be used for any kind of time series forecasting [396]. They can also be used for co-creativity, as in Creative Help [540]. Despite their simplicity, sequence prediction models have been one of the most successful generative techniques. An interesting example is *Sunspring*. It might be considered the first AI-scripted movie: it was generated by a Char-RNN trained on thousands of sci-fi scripts [443]. The quality of the result is demonstrated by the fact that it was able to reach the top ten at the annual Sci-Fi London

Film Festival in its 48-Hour Film Challenge[2].

Critical Discussion. Sequence prediction models generate outputs that exhibit characteristics of both exploratory and combinatorial creativity. These models are based on probabilistic predictions, allowing them to generate new outputs in the induced space. They can also combine sequences of tokens from different works. However, there is no guarantee that the results will be valuable or novel, and classic methods such as RNNs lack surprise [80]. It is worth noting that the use of conditional inputs and the ability to work at different levels of abstraction might indirectly lead to creative outputs. In such cases, creativity should be attributed to the higher-level component (or human if the input is provided by the user) that guides the generation toward specific elements and characteristics of the result.

3.1.4 Transformer-Based Models

Examples of Models. Several Transformer-based approaches have been proposed in recent years. The design of specific Transformers for a variety of applications is presented in several surveys (e.g., [59, 336]) and books (e.g., [661]).

The domain mostly influenced by Transformers is natural language processing. Bidirectional Encoder Representations from Transformers (BERT) [156] is a Transformer-based encoder trained for both predicting the next sentence (in an autoregressive fashion) and reconstructing masked tokens from the context. Several enhanced variations of the original model have been proposed, such as, for instance, solutions that remove the next-sentence pre-training objective [405], use inter-sentence coherence as an additional loss [359], or employ distillation [274] to train a smaller model [557]. The other main approach is that used by the Generative Pre-trained Transformer (GPT) family [511, 513, 74]. Here, a Transformer-based decoder is trained in an autoregressive way by additional conditioning on the task of interest. After training, it can be used to perform a wide range of tasks by providing a description or a few demonstrations of the task. The effectiveness of this text-to-text generative approach has then been explored by T5 [516]. Many other large language models [594, 655, 726] have been proposed to achieve better results by means of more parameters and computation [610], or more qualitative data [245]. Mixture of Experts [588] can be used as well in place of

[2] Quite interestingly, the AI system that wrote *Sunspring* declared that its name was Benjamin, probably in honor of Walter Benjamin, the German philosopher who, already in 1935 [41], understood that new mechanical techniques related to art can radically change the public attitude to arts and artists.

the feed-forward network to train a larger but lighter model (since only portions of it are used per task), as done by Generalist Language Model (GLaM) [167]. Finally, Bidirectional and Auto-Regressive Transformer (BART) [383] ideally merges a BERT-encoder (trained by corrupting text with an arbitrary noising function) and a GPT-decoder (trained to reconstruct the original text autoregressively). Such an encoder-decoder architecture achieved state-of-the-art results in machine translation and in other text-to-text tasks.

Transformer-based models have been used in domains different from language modeling. Few have been proposed for music generation. One of the first examples was Music Transformer [295], which can generate one-minute music in Bach's style with internal consistency; another remarkable one is Musenet [496], which is able to produce 4-minute musical composition with a GPT-2 architecture; and, finally, it is worth mentioning Jukebox [158], which can generate multiple minutes of music from raw audio by training a Sparse Transformer [112] (i.e., a Transformer with sparse factorization of the attention matrix to reduce from quadratic to linear scaling) over the low-dimensional discrete space induced by a VQ-VAE. Conditioning is always considered by means of genre, author, or instruments. MusicLM [5] additionally allows to generate music from text descriptions by aligning text and audio representation from different state-of-the-art models [62, 296]. Another important application domain is video-making. Video Vision Transformer (ViViT) [16] generates videos using classic Transformer architectures; Video Transformer (VidTr) [731] achieves state-of-the-art performance thanks to the standard deviation-based pooling method; and VideoGPT [712] does so by learning discrete latent representations of raw video with VQ-VAE, and then training a GPT autoregressively.

Transformers have been highly influential in computer vision too. The first model was Image Transformer [489]. It restricts the self-attention mechanism to attend to local neighborhoods, so larger images can be processed. Class-conditioned generation is also supported, by passing the embedding of the relative class in input. To avoid restricting self-attention to local neighborhoods, Vision Transformer [165] divides an image into fixed-size patches, linearly embeds each of them, adds position embeddings, and then feeds the resulting sequence of vectors to a standard Transformer encoder. Masked Autoencoders (MAE) [265] instead uses an encoder-decoder architecture based on Transformers trained with masked image modeling (i.e., to reconstruct randomly masked pixels). A BERT adaptation to images called Bidirectional Encoder representation from Image Transformers (BEiT) [28] has also been proposed. Masked image modeling has also been used together with classic autoregressive loss [106]. Conversely, Vector Quantised-GAN (VQ-GAN) [179] allows a Transformer to be based on vector quantization. A

GAN learns an effective codebook of image constituents. To do so, the generator is implemented as an autoencoder; vector quantization is applied over the latent representation returned by the encoder. It is then possible to efficiently encode an image in a sequence corresponding to the codebook indices of their embeddings. The Transformer is finally trained on that sequence to learn long-range interactions. These changes also allow us to avoid quadratic scaling, which is intractable for high-resolution images. Finally, DALL-E [514] takes advantage of a discrete VAE. To generate images based on an input text, it learns a discrete image encoding; it concatenates the input text embedding with the image encoding; it learns autoregressively on them. CogView implements a similar architecture [160].

Finally, Transformer-based models have also been used in multimodal settings, in which data sources are of different types. A survey can be found in [641]. The first examples of these systems consider text and images as the output of the Transformer architecture. By aligning their latent representations, images and texts can be generated by Transformer-based decoders given a multimodal representation. For instance, Contrastive Language-Image Pre-training (CLIP) [514] has an image encoder pre-trained together with a text encoder to generate a caption for an image. A Large-scale ImaGe and Noisy-text embedding (ALIGN) [316], based on similar mechanisms, can achieve remarkable performance through training based on a noisier dataset. In [659] the authors propose a frozen language model for multimodal few-shot learning: a vision encoder is trained to represent each image as a sequence of continuous embeddings so that the frozen language model prompted with this embedding can generate the appropriate caption. In [186] the authors present Bridging-Vision-and-Language (BriVL), which performs multimodal tasks by learning from weak semantic correlation data. Finally, there is a trend toward even more complex multimodal models. For example, Video-Audio-Text Transformer (VATT) [7] learns to extract multimodal representations from video, audio, and text; instead, Gato [527] serializes all data (e.g., text, images, games, other RL-related tasks) into a flat sequence of tokens that is then embedded and passed to a standard large-scale language model. Similarly, Gemini [209] achieves state-of-the-art performance in multimodal tasks by working on interleaved sequences of text, image, audio, and video as inputs; [210] extends it to a mixture of experts setting. Finally, NExT-GPT [702] handles any combination of four modalities (text, audio, image, and video) by connecting a language model with multimodal adaptors and diffusion decoders.

Applications. Transformer-based large language models can be used for almost any NLP task, including text summarization, generation, and interaction. In order to do so, the model can be used as frozen (i.e., to provide latent representations in input to other models); can be fine-tuned for the specific objective; can be exploited with zero-shot, one-shot or few-shot setting by prompting the task or few demonstrations in input. Transfer learning can instead be used to perform image classification by means of Transformer-based models trained on images. Other domain-specific techniques can be used as well: for instance, PlotMachines [525] learns to write narrative paragraphs not by receiving prompts, but by receiving plot outlines and representations of previous paragraphs. From a generative learning perspective, Transformers have shown impressive performance in producing long sequences of texts and music or speech [678], as well as in generating images based on input text. Their application has not been limited to these data sources. For instance, AlphaFold uses a Transformer architecture to predict protein structure [325]; RecipeGPT employs it to generate recipes [371]; and GitHub Copilot relies on it to support code development [107].

Critical Discussion. Considering that Transformers can be seen as an evolution of sequence prediction models, the observations made for that class of models (see Section 3.1.3) still hold. However, the inherent characteristics of their architecture allow for larger models and higher-quality outputs, leading to the capture of a variety of text dependencies across data sources. More in general, a broader conceptual space is induced. This means that domain-specific tasks might be addressed through solutions outside or at the boundary of the sub-space linked with that domain. Moreover, possibly also through careful use of inputs (see Section 3.1.6), their adoption might lead to transformational creativity. As far as Boden's criteria are concerned, there is no guarantee that the output of the Transformer architecture would be valuable, novel, or surprising, even though current state-of-the-art Large Language Models (LLMs) achieve almost human-like performance in creative tests [625, 734]. Finally, LLMs have proven capable of evaluating their own outputs, making them potentially *appreciative*: the so-called LLM-as-a-Judge approach [111, 736] returns evaluations that align with those from human experts. However, these evaluations are still naive: for example, they suffer from positional bias, i.e., altering the order of candidate responses can affect their quality ranking [681].

3.1.5 Diffusion Models

Examples of Models. Diffusion models have been primarily used for image generation. In order to produce higher-quality images and allow text-to-image generation, a variety of effective conditioning methods have been proposed. A possibility is to use classifier guidance [157]: the diffusion score (i.e., the added noise) includes the gradient of the log-likelihood of an auxiliary classifier model. An alternative is classifier-free guidance [275]: to avoid learning an additional model, a single neural network is used to parameterize two diffusion models, one conditional and one unconditional; the two models are then jointly trained by randomly setting the class for the unconditional model. Finally, the sampling uses a linear combination of conditional and unconditional score estimates. Guided Language to Image Diffusion for Generation and Editing (GLIDE) [468] demonstrates how classifier-free guidance can be effectively used to generate text-conditional images. In addition, it shows how diffusion models can be used for image editing by fine-tuning to reconstruct masked regions. Performance improvement can be obtained through a cascade of multiple diffusion models performing conditioning augmentation [277]. Notably, the diffusion model can operate on latent vectors instead of real images. Stable Diffusion [542] employs a diffusion model in the latent space of a pre-trained autoencoder. Similarly, DALL-E 2 [522] generates images by conditioning with image representations. At first, it learns a prior diffusion model to generate possible CLIP image embeddings from a given text caption, i.e., conditioned by its CLIP text embedding. Then, a diffusion decoder produces images conditioned by the image embedding. The generation quality can be further improved by means of generated captions for the images in the training set [46]. Imagen [552] uses instead a cascaded diffusion decoder, together with a frozen language model as a text encoder to increase the quality of output.

Although the approach is particularly suitable for images, applications to other data sources have been developed as well. DiffWave [349] and WaveGrad [108] use diffusion models to generate audio. They overcome the continuous-discrete dichotomy by working on the waveform. Another possibility is to use an autoencoder like MusicVAE [539] to transform the sequence into a set of continuous latent vectors, on which training a diffusion model [448]. Resembling image generators, Contrastive Language-Audio Pretraining (CLAP) embeddings [174] can be used to generate audio by conditioning on text descriptions [400]. Diffusion-LM [390] employs diffusion models to write text by denoising a sequence of Gaussian vectors into continuous word vectors (then converted into discrete words by a rounding step); DiffuSeq [224] performs sequence-to-sequence generation tasks by embedding source and target sequences in the same embedding space through a Transformer architecture. Diffusion models have been used for 3D generation as well [469]. Finally, diffusion models for video have also been proposed, based on gradient-based conditioning [278], and on processing latent spacetime patches. In particular, with respect to the latter, Sora [72] first turns videos into sequences of patches and then uses a diffusion Transformer to predict the original patches from random noise (and conditioning inputs like text prompts), improving sample quality and flexibility.

Applications. Despite their recent introduction, diffusion models have been used to generate audio, music, and video, as well as to create and edit images conditioned on input text, e.g., with in-painting [410] or subject-driven generation [544]; we refer to [715] for a comprehensive survey of this area. Indeed, they lead to higher-quality outputs than the previous state-of-the-art models. In particular, DALL-E 2 and Stable Diffusion have been able to produce images from textual instructions with superior fidelity and variety.

Critical Discussion. Diffusion models learn a mapping between real images and a Gaussian latent space. Because of this, they are an example of exploratory creativity: they randomly sample from that space, and then they possibly navigate it in the direction imposed by conditional inputs. There is no guarantee that the results will be valuable, novel, or surprising, even though these approaches are able to generate outputs characterized by a high variety. As already argued, novelty and surprise may only arise due to the conditioning input (for example, a human describing a novel combination of elements), i.e., the model is not imaginative on its own.

3.1.6 Input-Based Methods

Examples of Models. As detailed in Section 2.1.6, input-based methods can be divided into two approaches. The first one consists of carefully modifying the input of a generative model until the output matches the desired properties. The main example is VQGAN-CLIP [139]. Given a text description, VQGAN produces a candidate image from a random latent vector; the vector is then optimized by minimizing the distance between the embeddings of the description and the candidate image. Both embeddings are computed using CLIP [514]. Variants can be implemented as in Wav2CLIP [698], where an audio encoder is learned to match the CLIP encoders so that VQGAN-CLIP can be used from raw audio; or as in music2video [310], where videos are generated from audios a frame after the other by both minimizing the distance between subsequent frames, and the distance between image and music segment embedded by Wav2CLIP. In addition to the random latent vector, the text or audio description can be optimized as well. This can be performed by the users through many iterations of careful adjustments or an automated procedure. The latter is commonly known as prompt tuning. Prompt tuning is about producing prompts via backpropagation; the optimized prompts can condition frozen language models to perform specific tasks without fine-tuning them [381]. An additional model can also be trained to output the desired prompt [382]. Finally, image generators such as VQGAN can also be exploited in other ways, i.e., with a binary-tournament genetic algorithm [190] or more complex evolution strategies [647]. Another possibility is to optimize the input so that the generated output maximizes a target neuron of an image classifier [465]. This helps generate what that neuron has learned. The desired latent vector can also be produced by an additional model [466].

The second approach is to optimize the inputs to transform them into the desired outputs. DeepDream [451] generates "hallucinated" images by modifying the input to maximize the activation of a certain layer from a pre-trained classifier. Artistic style transfer is based on the same idea. Given an input image and a target image, the former is modified using both style and content losses thanks to a pre-trained classifier. The content loss is minimized if the current and the original input images generate the same outputs from the hidden layers. The style loss is minimized if the current and target images have the same correlation pattern between feature maps in the hidden layers [200]. Control over results can be improved by considering additional losses about color, space, and scale [201].

Applications. Input-based methods can be used with any generative model to produce the desired output. With language models, they can exploit their generality in several specific tasks without fine-tuning them. For instance, prompt tuning can be used by writers for co-creation [100] or to force LLMs to *brainstorm* [634]. With image generators, they can obtain drawings adherent to given descriptions, or high-quality but yet peculiar paintings like colorist [190], abstract [647], or alien [611] artworks. We believe applications to other domains are yet to come. Both types of input-based methods can be used not only to produce desired outputs or transfer styles; they can also be adopted to better analyze what is inside the network [465, 475].

Critical Discussion. Since input-based methods are applied to pre-trained generative models, the space of solutions in which they work is the one induced by those models, i.e., the common spaces we can derive from real data. Nonetheless, some techniques may be able to lead to productions that are outside that space or at its boundaries, i.e., to cause transformational creativity. This might happen if the model is general, and the output for a specific task is not only sampled from the sub-space of solutions for that task (e.g., with prompt tuning over a language model). Input-based methods are also valuable: the optimization itself is typically guided by some sort of qualitative loss. On the other hand, they are not explicitly novel or surprising (although the results might seem so). However, nothing prevents optimizing the loss in such directions (see Section 3.1.7).

3.1.7 Practical Assessment of Creativity-Oriented Methods

We conclude this analysis of generative models with a discussion of how they might increase their *creativity* according to Boden's definition. We have discussed how the presence of a recognition model (e.g., a discriminative model or a reward model) helps ensure the value of the products. In the same way, novelty and surprise can be fostered by the integration of other components. A straightforward way to obtain novel and surprising outputs is to train a generative model by means of novelty and surprise objectives. This is the core idea behind Creative Adversarial Network (CAN) [173, 563]. In addition to the classic discriminative signal, i.e., a value loss, the generator is also trained to optimize a novelty loss. This is defined as the deviation from style norms, i.e., the error related to the prediction of the style of the generated image. The sum of the two training signals helps the model learn to produce artworks that are different (in style) from the training data. The same approach has been used to develop a creative StyleGAN, i.e., StyleCAN [315]. Another, very simple way to augment the training signal of a generative model with *creativity-oriented* objectives is through RL-based methods (see Section 3.3). The choice of the reward structure is the fundamental element in the design of effective generative reinforcement learning systems. Rewards should teach the model to generate an output with a high level of novelty and surprise. An example is ORGAN [243], where appropriate reward functions can be used. For instance, statistical measures (e.g., Chi-squared) or metrics of distance between distributions (e.g., KL divergence) might be used to ground ideas of novelty and surprise.

Another possibility is the development of an input-based method where the input is optimized to obtain a product that is valuable, novel, and surprising. This may be achieved either by forcing a further exploration of the latent space (e.g., using evolutionary search [188]), or by defining appropriate loss functions to perform gradient descent over the input. All these methodologies are also called *active divergence* [45] since they aim to generate in ways that do not simply reproduce training data. A survey on active divergence can be found in [69]. A different output can also be obtained by carefully altering the probability distribution of the model, e.g., by scaling its probabilities with learned functions to maximize target properties [142, 612, 714].

An alternative approach is followed by the Composer-Audience architecture [80]. Two models are considered: the Audience, a simple sequence prediction model trained on a given dataset; and the Composer, another sequence prediction model trained on a different dataset. In addition, the Composer also receives the next-token expectations from the Audience, and it learns when to follow its guidance and when to diverge from expectations, i.e., when to be surprising. For instance, it can learn to produce jokes by considering non-humorous texts to train the Audience and humorous texts to train the Composer. Even though this approach is useful for learning how to generate valuable and surprising output, it is only applicable when paired datasets are available.

Generative family	Type of creativity	Boden's criteria	Creative suggestions
VAE	Exploratory	∼ Value ∼ Novelty ∼ Surprise	Creativity-oriented input-based methods
GAN	Exploratory	✓ Value ∼ Novelty ∼ Surprise	CAN; Creativity-oriented input-based methods
Sequence prediction model	Combinatorial, Exploratory	∼ Value ∼ Novelty × Surprise	Composer-Audience; Creativity-oriented RL-based methods
Transformer-based models	Combinatorial, Exploratory, Transformational	∼ Value ∼ Novelty × Surprise	Creativity-oriented prompt tuning or RL-based methods
Diffusion models	Exploratory	∼ Value ∼ Novelty ∼ Surprise	Creativity-oriented input-based methods
Input-based methods	Exploratory, Transformational	✓ Value ∼ Novelty ∼ Surprise	Evolutionary search; Novelty-based optimization

Table 3.1: Summary of all the methods explained so far, considering their type of creativity as discussed in the corresponding subsections; the possible presence of Boden's criteria (✓ if induced by the training process; ∼ if not considered; × if excluded); and some practical suggestions to achieve a higher degree of creativity.

As far as the type of creativity is concerned, there can be ways to achieve a better exploration or even transformation of the space of solutions. For example, since CAN novelty loss is used during training, it learns to diverge from the distribution of real data. The same is true for RL-based methods with novelty and surprise rewards (especially if the training happens from scratch). Finally, increased exploration and transformation may be achieved using RL-based methods driven by curiosity [82]: an agent can learn to be creative and discover new patterns thanks to intrinsic rewards to measure novelty, interestingness, and surprise (see Section 3.2.2). This can be done by training a predictive model of the growing data history and using its learning progress as the reward. In this way, the agent is motivated to explore in order to discover things that the predictor does not already know. If an external qualitative reward is considered as well, the agent should in theory learn to do things that are new, but still valuable [565]. The same idea can also be applied to other techniques like evolutionary strategies [416]. Deep Learning Novelty Explorer (DeLeNoX) [392] uses a denoising autoencoder to learn low-dimensional representations of the last generated artifacts. Then, a population of candidate artifacts (in terms of their representation) is evolved through a feasible-infeasible novelty search [393] to maximize the distances between them, i.e., to increase their novelty, while still considering qualitative constraints. Other evolutionary strategies might be embraced as well to search the space of artifacts for novel [376] and surprising [232] results. Instead of relying on manually crafted metrics, Quality Diversity through Human Feedback (QDHF) [159] uses human feedback for computing quality and distance in learned latent projection for computing diversity. Quality-Diversity through AI Feedback (QDAIF) [67] makes the model more independent in searching and innovating by completely relying on its own feedback for both quality and diversity.

Table 3.1 summarizes all the generative approaches discussed in this section, highlighting their characteristics from a machine creativity perspective.

3.2 Reinforcement Learning

Several reinforcement learning sub-fields and research directions have been explored in recent years. In the following, we will cover the ones crucial for our future discussion.

3.2.1 Imagination-Based Reinforcement Learning

Model-based RL requires learning a model of the environment dynamics, i.e., a transition model to predict the next state given the current state and action, and a reward model to predict the reward associated with that transition. By using them, the action selection can be based on the next states and rewards expectations, e.g., by building a Monte Carlo Tree Search [134] to find the optimal policy [600]. Indeed, model-based RL is commonly used together with planning [121, 267, 569] since Dyna [638]. However, another possibility is to use such a dynamics model to construct imaginary trajectories in the latent space induced by an encoder model [684]. Such trajectories can be used to guide model-free agent decisions as in Imagination-Augmented Agents (I2A) [509, 79], or directly train the agent reducing the need for interaction with the real environment.

The latter approach is the most popular. A few main examples can be identified in the literature. The first one is World Models [252]. It makes use of a VAE [339] to encode real observations into a latent space, and of an RNN to learn transitions between latent space and to predict associated rewards based on the current state-action pair and belief (i.e., the RNN state). This RNN is trained to minimize the negative log-likelihood of the next state and the mean squared error of reward. Then, an actor is obtained through a policy gradient method only by acting in the defined dynamics model, i.e., returning actions to the world model, from which receiving the next (latent) state and related rewards. In [300], the world model is also trained on the RL agent objective. The environment's dynamics can also be modeled differently. In [456], instead of predicting the entire new state, it only learns the difference between consecutive states. Then, imagined trajectories can be generated from a starting state by summing the subsequent, predicted differences. These generated trajectories are finally used to pre-train an agent (then fine-tuned in a model-free fashion). Stochastic Latent Actor-Critic (SLAC) [368] learns a compact latent representation of the environment through three models: a recognition model (i.e., a posterior model returning the current latent state given the current observation, previous state, and previous action); a generative model (reconstructing the current observation given its latent state); and a dynamics model (i.e., a prior model to predict the encoded state given the previous state-action pair). The entire world model is trained to minimize the reconstruction error and the KL divergence between posterior and prior. After that, an actor-critic algorithm is directly trained in such a latent space. Finally, Dreamer [259] combines the recurrent nature of [252] with the prior-posterior dichotomy of [368]. The entire dynamics (presented in Section 2.2.2) is characterized by the use of a Recurrent State-Space Model (RSSM), as proposed in PlaNet [258] for planning in latent space, to model the temporary dependencies between states. As in [368], the entire model is trained through the reconstruction errors and the KL divergence between prior and posterior. The resulting dynamics model is used to construct imagined trajectories on which training the agent, which happens in parallel with the dynamics training. DreamerV2 [260] and DreamerV3 [261] extend it to discrete latent spaces. Notably, Transformers can be used in place of recurrent neural networks [104]. Other variants have been proposed as well: Plan2Explore [577] also uses an intrinsic reward to guide imagined exploration and to collect episodes on which training the dynamics model. Such intrinsic reward is computed as the variance of predictions of next-state features made by an ensemble of models. The BrIdging Reality and Dream (BIRD) algorithm [739] uses latent overshooting (again introduced in PlaNet [258]) to train the dynamics-agent pair together. Finally, Imagining with Derived Memory (IDM) [455] proposes to use an imagined trajectory built not only starting from real states but also from a derived memory of states whose features are randomly modified.

3.2.2 Curiosity-Driven Reinforcement Learning

In Section 1.2, we briefly discussed the theoretical foundations of intrinsic motivation and curiosity. In Section 3.1.7, we saw how intrinsic motivation can be conducive to creativity-oriented generative methods. In addition, curiosity-driven RL has some practical benefits, such as solving the exploration-exploitation trade-off and dealing with environments with sparse non-zero rewards.

Due to these factors, intrinsic rewards have been widely explored in the RL community. From a practical point of view, it is possible to identify a few common approaches: surprisal, Bayesian surprise, novelty, and uncertainty. Surprisal [658] is the difference between an expected event and the actual one. In the RL framework, it is commonly defined as the difference between the real next state and the expected one, as computed by a transition model trained to predict the next state given the current state-action pair. This difference can be a simple mean squared error for deterministic state prediction [493, 82, 529], or a KL divergence for probabilistic state prediction (i.e., with a model returning the mean and the variance of the probability distribution) [3]. Conversely, Bayesian surprise [306] is the difference between the posterior distribution (i.e., after experiencing the new state) and the prior distribution. It can be computed as the KL divergence between predictions before and after being trained on the new transition [3, 196, 710], or directly as the transition model's gradient score, as it effectively measures the degree of variation caused by the new state [290, 255]. Another possibility is to consider two different models for prior and posterior, and then compute the KL divergence between their predictions (which incidentally is the loss score used to perform model updates) [430]. Novelty has been instead defined in different ways: by means of an information theory-based approach, i.e., by measuring the KL divergence between an encoder and a fixed, prior distribution [338]; or with reachability, by counting the number of steps necessary to reach the current state from the closest state in memory [562]. Finally, uncertainty can be seen as the disagreement between different internal transition models [494, 577], or as the difficulty of recognizing something as real (i.e., the inverse of a GAN discriminative signal when judging a single state [286] or an entire trajectory [78]).

3.2.3 Generalization in Reinforcement Learning

One of the main problems of reinforcement learning is that of generalization. The most used benchmarks (e.g., Atari [36]) use the very same environment for both training and testing. This can lead to strong evaluation overfitting, and it is in contrast with the open-ended and constant changing of reality. Generalization in deep reinforcement learning then refers to producing RL algorithms whose policies can correctly generalize to unseen situations at inference time (a deep survey can be found in [345]). Among the different benchmarks proposed to work with generalization, ProcGen [126] is probably the most successful. It is a suite of 16 procedurally generated game-like environments. To benchmark generalization capabilities, only a small subset of the distribution of levels is used to train the agent; the full distribution is then used to test it on unseen levels. To prevent overfitting and approach generalization, different techniques have been proposed. The first strategy is to adopt some of the techniques used in supervised learning to avoid overfitting, e.g., dropout, batch normalization, and specific convolutional architectures [125, 301]. Another strategy is to improve the agent's architecture, the training process, or even the sampling technique for experience replay [318]. For instance, Raileanu and Fergus [517] propose to decouple the policy and value functions, and also train the value function with an auxiliary loss that encourages the model to be invariant to task-irrelevant properties of the environment. Also, post-training distillation may help improve generalization to new data [412], as well as learning an embedding in which states are close when the optimal policies in these states are similar [4]. Finally, a third strategy is to use data augmentation to increase the size and variability of training data [125, 362, 372, 718]. The kind of augmentation technique can even be learned and not selected *a priori* [518].

3.3 Reinforcement Learning for Generative AI

In this section, we will discuss the state of the art in RL for generative learning considering three classes of solutions, which are summarized in Table 3.2:

RL as an alternative solution for output generation to approximate outputs from a given domain of interest with high fidelity; RL as a way for generating output while maximizing an objective function which captures (additional) quantifiable properties or indicators at the same time; and, finally, RL as a way of embedding additional desired characteristics, such as value alignment, which cannot easily be captured through an objective function into the generative process.

Goal	Reward	Advantages	Limitations
Mere generation	- GAN's discriminative signal; - Log-likelihood of real or predicted targets; - Constraint satisfaction.	- Models domains defined by non-differentiable objectives; - Adapts GAN to sequential tasks; - Can implement RL strategies, e.g., hierarchical RL.	- Learning without supervision is hard; - Pre-training can prevent an appropriate exploration.
Objective maximization	- Test-time metrics; - Countable desired or undesired characteristics; - Distance-based measures; - Quantifiable properties; - Output of ML algorithms.	- Satisfies quantifiable requirements; - Optimizes a generator from a specific domain towards desirable sub-domains; - Reduces the gap between training and evaluation.	- Not every desirable property is quantifiable or easy to define; - Goodhart's law.
Improving not easily quantifiable characteristics	- Output of a model trained to reproduce human or AI feedback about non-quantifiable properties (e.g., helpfulness, appropriateness, creativity).	- Satisfies non-quantifiable requirements (for example, the alignment problem); - Requires preferences between candidates instead of defining a mathematical measure of the desired property.	- Getting user preferences is expensive; - Users might misbehave, disagree, or be biased; - Reward modeling is difficult; - Prone to jailbreaks out of alignment.

Table 3.2: Summary of the three purposes for using RL with generative AI, considering the used rewards, their advantages, and their limitations.

3.3.1 Reinforcement Learning for Mere Generation

Overview. The simplest approach is RL for *mere* generation, i.e., to train a generative model to approximate outputs from a given domain of interest as best as possible. Essentially, the objective function replicates the behavior of the self-supervised learning loss used in traditional generative learning approaches, such as the adversarial one.

The first example we consider is SeqGAN [721]. In their original formulation, GANs cannot be used for sequential tasks (see Section 3.1.2). SeqGAN circumvents this problem by using RL, which allows us to learn from rewards received further in the future. Indeed, SeqGAN exploits the discriminative signal as the actual reward received at the end of the episode, i.e., when the sequence is completed. The approach is based on REINFORCE [692]. A similar method is also used in MaskGAN [184], where the generator learns with in-filling (i.e., by masking out a certain amount of words and then using the generator to predict them) through actor-critic learning [637]. Notably, hierarchical RL can also be used: for example, LeakGAN [246] relies on a generator composed of a manager, which receives *leaked* information from the discriminator, and a worker, which relies on a goal vector as a conditional input from the manager. Since SeqGAN might produce very sparse rewards, alternative strategies have been proposed. The discriminator can be replaced with a reward model learned with inverse reinforcement learning on state-action pairs so that the reward is available at each timestep (together with an entropy regularization term) [590]. A more complex state composed of a context embedding can also be used [388]. Instead, [387] is based on a variation of SeqGAN: it uses Monte Carlo tree methods to get rewards at each timestep. In addition, the authors also suggest alternating RL with a "teacher", i.e., the classic supervised training. This helps deal with tasks like text generation where the action space (i.e., the set of possible words or sub-words) is too large to be consistently explored using RL alone. Another solution to this problem is Natural Language Policy Optimization (NLPO) [520], which is a parameterized-masked extension of PPO [573] that restricts the action space via top-p sampling. The authors of [426] use top-p sampling as well; however, they restrict the action space through a pre-trained task-agnostic model *before* applying policy gradient with PPO. Similarly, ColdGAN [575] forces the sampling of a SeqGAN-like generator to be close to the distribution modes by selecting actions with top-p sampling and low temperature [285] and training the generator via importance sampling [507]. Finally, the top-p sampling strategy can be replaced by a cooperative one based on a Monte Carlo Tree Search structure evaluated by the discriminator [358]; again, the generator is trained via importance sampling.

Another reason to use RL is to take advantage of its inherent properties. For example, Generation by Off-policy Learning from Demonstrations (GOLD) [485] is an algorithm that substitutes self-supervised learning with off-policy RL and importance sampling. It uses real demonstrations, which are stored in a replay buffer; the reward corresponds to either the sum or the product of the action probabilities over the sampled trajectories, i.e., of every single real token according to the model. While it can be considered close to a self-supervised approach, off-policy RL with importance sampling allows up-weighting actions with high (cumulative) returns and actions preferred by the current policy, encouraging to focus on in-distribution examples.

RL is also an effective solution for learning in domains where a differentiable objective is difficult or impossible to define. RL-Duet [319] is an algorithm for online accompaniment generation. Learning how to produce musical notes according to a given context is a complex task: RL-Duet first learns a reward model that considers both inter-part (i.e., with counterpart) and intra-part (i.e., on its own) harmonization. Such a model is made of an ensemble of networks trained to predict different portions of music sheets (with or without the human counterpart, and with or without machine context). Then, the generative system is trained to maximize this reward through an actor-critic architecture with GAE [572]. CodeRL [365] performs code generation through a pre-trained model and RL. In particular, the model is fine-tuned with policy gradient to maximize the probability of passing unit tests: it receives a (sparse) reward quantifying if (and how) the generated code has passed the test for the assigned task. In addition, a critic learns a (dense) signal to predict the compiler output. The model is then trained to maximize both signals considering a baseline obtained with a greedy decoding strategy. To obtain a denser and more informative reward, PPOCoder [595] also considers three additional signals: a syntactic matching score based on the Abstract Syntax Tree of the generated code; a semantic matching score based on the data-flow graph; and a KL penalty to prevent the model from deviating considerably from its pre-trained version. The sum of these four signals is then optimized via PPO.

Another interesting application area is painting. The author of [708] suggests modeling stroke painting as a Markov Decision Process, where the state is the canvas, and the actions are the brushstrokes performed by the agent. Rewards calculated considering the location and inclination of the strokes are then used to train the agent. For instance, Doodle-SDQ [738] fine-tunes a pre-trained sketcher with Double DQN [670] and a reward that is calculated by evaluating how well a sketch reproduces a target image at pixel, movement, and color levels. In [298], a discriminator is trained to recognize real canvas-target image pairs to derive a corresponding reward. Instead, in

[602] a painting policy operates at two different levels: foreground and background. Each of them uses a discriminator; in addition, the authors adopt a focus reward measuring the degree of indistinguishability of two object features. On the other hand, Intelli-Paint [603] is based on four different types of rewards, which are used to learn a painting policy with DDPG [395] based on a discriminator signal on canvas-image pairs, two penalties for the color and position of consecutive strokes, and the same semantic guidance from [602]. Finally, RL has also been used for collage artwork. The authors of [369] propose an RL-based method to compose different elements (such as newspaper or texture cuts) to obtain an output that resembles a target picture. The state is composed of the canvas, the target image, and a randomly (or value-based) sampled material; the action determines which region of the material to cut and where to paste it on the current canvas; and the reward is the amount of similarity change between consecutive timesteps (where a WGAN-GP discriminator [244] trained in parallel to discriminate between target-target and target-canvas pairs computes the similarity between the canvas and the target image). A model-based soft actor-critic [253] is then used to optimize the reward minus a penalty for each timestep to teach the agent to complete the tasks with the minimum number of actions.

Discussion. RL can represent an alternative method for deriving generative models, especially if the target loss is non-differentiable. It allows for the adaptation of known generative strategies, e.g., GANs, to tasks for which traditional techniques are not suitable, e.g., in text generation. In addition, it can be applied to domains in which feasibility and correctness (e.g., running code as above) are essential dimensions to consider. In other words, RL can train a generative model to produce observations that appear to have been drawn from the domain of interest even when such a domain cannot be modeled through generative functions and corresponding differentiable losses. RL can also be used to derive more complex generative strategies (e.g., through hierarchical RL) and to reduce the model dependence on training data, which might have an impact on copyright issues (see Section 7.3).

It is possible to identify some limitations of the proposed solution. Learning without supervision is hard, especially when the reward is sparse. This is likely to happen for sequence generation, such as (long) text or music, where the reward is available only at the last timestep. In addition to the aforementioned techniques for obtaining a denser reward, a potential solution might consist of considering an intrinsic reward [19] as an additional learning signal, to encourage exploration as well. Moreover, the action space can be very large (potentially orders of magnitude larger than those of standard RL problems [12]), especially for text generation. Ensuring a sufficient exploration of all possible actions while still exploiting the most promising ones to collect higher rewards is one of the key problems in RL. Starting with some prior knowledge about the possible best actions for different situations might be necessary for fast convergence. For this reason, pre-trained generative models are selected for this task. This can cause the agent to initially focus on highly probable tokens, increasing their associated probabilities and, because of that, failing to explore different solutions (i.e., by only moving the probability mass of the already most probable tokens) [117]. These problems can be avoided through variance reduction techniques (e.g., incorporating baselines and critics) and exploration strategies [337].

3.3.2 Reinforcement Learning for Objective Maximization

Overview. Since RL allows us to use any non-differentiable function for modeling the rewards, it could be the case that simply replicating the behavior of a self-supervised learning loss is not the optimal solution. For example, the authors of [524] point out the mismatch between how deep learning models are trained (i.e., on differentiable losses) and how they are commonly evaluated (i.e., on non-differentiable metrics): an emerging line of research is focusing on the use of non-differentiable metrics as reward functions for generative learning capturing a variety of requirements and constraints.

RL for quantity maximization has been mainly adopted in text generation, especially for dialogue and translation. In addition to exposure bias mitigation, it allows for replacing classic likelihood-based losses with metrics used at inference time. A pioneering work is [524], where RL is adopted to directly maximize BLEU [486] and ROUGE [397] scores. To deal with the size of the action space, the authors introduce MIXER, a variant of REINFORCE algorithm that uses incremental learning (i.e., an algorithm based on an optimal pre-trained model according to ground truth sequences) and combines reward maximization with classic cross-entropy loss by means of an annealing schedule. In this way, the model starts with preexisting knowledge, which is preserved through the classic loss, while aiming at exploring alternative but still probable solutions, which should increase the score at test time. A similar approach is also used by Google's neural machine translation system [705]. BLEU score is used as the reward while fine-tuning a pre-trained neural translator with a mixed maximum likelihood and expected reward objective. In [23], an actor-critic algorithm is considered for machine translation, with the critic conditioned on the target text, and the pre-trained actor fine-tuned with BLEU as the reward. The authors of [495] suggest learning to perform text summarization by using self-critical policy training [530], where the reward associated with the action that would have been chosen at inference time is used as a baseline. ROUGE score is the reward and is linearly mixed with teacher forcing [693], i.e., classic supervised learning. Scores alternative to ROUGE have been proposed as well, e.g., ROUGESal and Entail both described in [491]. The former up-weights the salient sentences or words detected via a key-phrase classifier. The latter rewards logically-entailed summaries through an entailment classifier. They are then used alternatively in subsequent mini-batches to train a Seq2Seq model [636] by means of REINFORCE. Finally, BLEU score can be employed to train a dialogue system on top of collected human interactions with offline RL [737]. An additional dialogue-level reward function (measuring the number of proposed API calls) is also used. Recently, the RL4LM library [520] started offering many of these metrics as rewards, thus facilitating their use for LM training or fine-tuning. Different families of solutions are considered, i.e., n-grams overlapping such as ROUGE, BLEU, SacreBLEU [505] or METEOR [364]; model-based methods such as BertScore [727] or BLEURT [578]; task-specific metrics; and perplexity. Notably, RL4LM also allows balancing such metrics with a KL-divergence minimization with respect to a pre-trained model.

Test-time metrics are not the only quantities that can be maximized through RL. For example, the count of 4-gram repetitions in the generated text can be considered to reduce the likelihood of undesirable results [355]. The combination of these techniques and classic self-supervised learning helps learn both *how to write* and *how not to write*. In [386], a Seq2Seq model for dialogue is trained by rewarding conversations that are informative (i.e., which avoid repetitions), interactive (i.e., which reduce the probability of answers like “I don’t have any idea” that do not encourage further interactions), and coherent (i.e., which are characterized by high mutual information with respect to previous parts of the conversation). Sentence-level cohesion (i.e, compatibility of each pair of consecutive sentences) and paragraph-level coherence (i.e., compatibility among all sentences in a paragraph) can be achieved by maximizing the cosine similarity between the encoded version of the relative text, with the encoders trained so that the entire discriminative models can distinguish between real and generated pairs [115]. A distance-based reward can instead guide a plot generator toward reaching desired goals. The authors of [642] train an agent working at the event level (i.e., a tuple with the encoding of a verb, a subject, an object, and a fourth possible noun) with REINFORCE to minimize the distance between the generated verb and the goal verb. Other domain-specific rewards are used by [719], where two distinct generative models produce poetry by maximizing fluency (i.e., MLE on a fixed language model), coherence (i.e., mutual information), meaningfulness (i.e., TF-IDF), and overall quality from a learned classifier. In addition, the two models also learn from each other: the worst performing can be trained on the output produced by the other, or its distribution can be modified to better approximate the other's.

Another popular technique is hierarchical RL: it allows the optimization of quantifiable objectives even when they work at a different level of abstraction with respect to the generative model. For example, the authors of [498] design a dialogue system able to perform composite tasks, i.e., sets of subtasks that need to be performed collectively. A high-level policy, trained to maximize an extrinsic reward directly provided by the user after each interaction, selects the sub-tasks. Then, "primitive" actions to complete the given sub-task are chosen according to a lower-level policy. A global state tracker on cross-subtask constraints provides the RL model with an intrinsic reward measuring how likely a particular subtask will be completed. Finally, a fixed LLM can be perturbed through a learned state-action and a state-value function, rather than directly fine-tuning the model itself [612]. This allows us to preserve the capabilities of the given pre-trained language model, while still maximizing a specific utility function.

While text generation is one of the areas that have attracted most of the attention of researchers and practitioners in the past years, RL with quantity maximization has been applied to other sequential tasks as well. An important line of research [311, 313, 312] consists of fine-tuning a pre-trained sequence predictor with imposed reward functions, while preserving the learned properties from data. For instance, a pre-trained note-based RNN can represent the starting point for the Q-network in DQN. A reward given by the probability of the chosen token according to the original model (or based on the inverse of the KL divergence) and one based on music theory rules (e.g., that all notes must belong to the same key) are used to fine-tune the model. Another possibility is to extend SeqGAN to domain-specific reward maximization, as in Objective-Reinforced GAN (ORGAN) [243]. ORGAN linearly combines the discriminative signal with desired objectives, also dividing the reward by the number of repetitions made, to increase diversity in the result. Music generation can then be performed by considering tonality and ratio of steps as rewards; solubility, synthesizability, and drug-likenesses are instead adopted to perform molecule generation as a sequential task, i.e., by considering a string-based representation of molecules (by means of SMILES language [688]). While the original work considered RNN-based models, Transformer architectures can be used as well [384].

Molecular generation is indeed one of the most explored tasks at the intersection between RL and generative AI. While MolGAN [145] adapts

ORGAN to graph-based generative models [391] to directly produce molecular structures, the majority of research focuses on simplified molecular-input line-entry system (SMILES) textual notation [688] to leverage the recent advancements in text generation. Reinforcement Learning for Structural Evolution (ReLeaSe) [504] fine-tunes a pre-trained generator to maximize physical, biological, or chemical properties (learned by a reward model). In [476] a pre-trained generator is fine-tuned with REINFORCE to maximize a linear combination of a prior likelihood (to avoid catastrophic forgetting) and a user-defined scoring function (e.g., to match a provided query structure or to have a predicted biological activity). REINVENT [50] also avoids generating molecules the model already produced through a memory that keeps track of the good scaffoldings generated so far. REINVENT is then adapted for the graph-based deep generative model GRAPHINVENT [435] to directly obtain molecules that maximize desired properties, e.g., pharmacological activity [18]. Instead, the authors of [735] generate kinase inhibitors relying on a variational autoencoder to reduce molecules to continuous latent vectors. Then REINFORCE is used to teach the decoder how to maximize three properties learned through self-organizing maps: activity of compounds against kinases; closeness to neurons associated with DDR1 inhibitors within the whole kinase map; and novelty of chemical structures. The average reward for the produced batch is assumed as a baseline to reduce variance. Notably, RL is used here for single-step generation (i.e., through a contextual bandit). The authors of [202] propose to generate molecules maximizing their partition coefficient without any pre-training by working with a simplified language [352]; its solution space can also be better explored with intrinsic rewards [646]. Graph Convolutional Policy Network (GCPN) [720] trains a graph-CNN to optimize domain-specific rewards and an adversarial loss (from a GAN-like discriminator) through PPO. Other tasks have been investigated as well. The authors of [467] merge GAN and actor-critic to obtain a generator capable of producing 3D material microstructures with desired properties. DDPG can train an agent to design buildings (in terms of shape and position) to maximize several signals related to the performance and aesthetics of the generated block, e.g., solar exposure, collision, and number of buildings [263].

Finally, techniques based on objective maximization can also be effective for image generation. Denoising Diffusion Policy Optimization (DDPO) [49] can train or fine-tune a denoising diffusion model to maximize a given reward. It considers the iterative denoising procedure as a Markov Decision Process of fixed length. The state contains the conditional context, the timestep, and the current image; each action represents a denoising step; and the reward is only available for the termination state when the final, denoised image is obtained. DDPO has therefore been used to learn how to generate images that are more compressed or uncompressed, by minimizing or maximizing JPEG compression; more aesthetically pleasing, by maximizing LAION score [570]; or more prompt-aligned, by maximizing the similarity between the embeddings of prompts and generated image descriptions. Improving the aesthetics of the image while preserving the text-image alignment has also been done at the prompt level [264]. A language model that given human input provides an optimized prompt can be trained with PPO to maximize both an aesthetic score (from an aesthetic predictor) and a relevance score (as the CLIP embedding similarity) of the image generated from the given prompt.

Discussion. Reinforcement learning for objective maximization opens up several new possibilities: generators can be adapted for particular domains or specific problems; they can be built for tasks difficult to model through differentiable functions; and pre-trained models can be fine-tuned according to given requirements and specifications. Essentially, RL is not used only for mere generation, since it also allows more specific, goal-oriented generative modeling: instead of training a generator to produce *correct, reasonable examples* for the domain of interest, the goal is to derive *the best possible examples* according to some specific target functions. Any desired and quantifiable property can now be set as a reward function, thus in a sense "teaching" a model how to achieve it. While research has focused on sequential tasks like text or music generation, other domains might be considered as well. As shown by [735], tasks not requiring multiple generative steps can be performed simply by reducing the RL problem to a contextual bandit one. In this way, RL can be considered as a technique for specific sub-domains, in a manner similar to neural style transfer [200] or prompt engineering [404].

We can identify possible drawbacks as well. Reinforcement learning has typically a very high computational cost [98], due to the number of iterations required to converge. In addition, certain desired properties (e.g., harmlessness or appropriateness) can be difficult to quantify, or the related measures can be expensive to compute, especially at run-time. This can lead to excessive computational time for training. While offline RL might alleviate this problem, it would require a collection of evaluated examples, thus eliminating the advantage of not needing a dataset and increasing the risk of exposure bias. Finally, a fundamental issue arises from using test-time metrics as objective functions: how should we evaluate the model we derive? In fact, according to the empirical Goodhart's Law [227], "when a measure becomes a target, it ceases to be a good measure". New metrics are therefore required, and a gap between training objectives and test scores might be inevitable.

3.3.3 Reinforcement Learning for Improving Not Easily Quantifiable Characteristics

Overview. While test-time metrics as objectives reduce the gap between training and evaluation, they do not always correlate with human judgment [99]. In these cases, using such metrics would not help obtain the desired generative model. Moreover, certain qualities might not have a correspondent metric because they are subjective, difficult to define, or, simply, not quantifiable. Typically, users only have an implicit understanding of the task objective, and, therefore, a suitable reward function is almost impossible to design: this problem is commonly referred to as the *agent alignment problem* [378].

One of the most promising directions is reward modeling, i.e., learning the reward function from interaction with the user and then optimizing the agent through RL over such a function. In particular, RLHF (see Section 2.3) allows us to use human feedback to guide policy gradient methods. The authors of [741] apply RLHF to text continuation, e.g., to write positive continuations of text summaries. Similarly, RLHF can be used to perform text summarization from sampled Reddit posts [626]. The authors of [700] propose to summarize entire books with RLHF by means of recursive task decomposition, i.e., by first learning to summarize small sections of a book, then summarizing those summaries into higher-level summaries, and so on. In this way, the size of the texts to be summarized is smaller. This is more efficient in terms of generative modeling and human evaluation since the samples to be judged are shorter. InstructGPT [482] fine-tunes GPT-3 [74] with RLHF so that it can follow written instructions. With respect to [626], demonstrations of desired behavior are first collected from humans and used to fine-tune GPT-3 before actually performing RLHF. In particular, this procedure is adopted in ChatGPT and GPT-4 [478], which are fine-tuned to be aligned with human judgment.

Although all these methods consider human feedback regarding the “best” output for a given input (with “best” generally meaning appropriate, factual, respectful, or qualitative), more specific or different criteria are also used. In [25] human preferences for helpfulness and harmlessness are considered. Sparrow [221] is trained to be helpful, correct, and harmless, with the three criteria judged separately so that three more efficient rule-conditional reward models are learned. In addition, the model is trained to query the web for evidence supporting provided facts; and again RLHF is used to obtain human feedback about the appropriateness of the collected evidence. Finally, RLHF can fine-tune GPT-2 to learn how to write *haikus* maximizing the relevance to the provided topic, self-consistency, creativity, form, and avoiding toxic content through human feedback [487]. In addition to text, RLHF has been used to better align text-to-image generation with human preferences. After collecting user feedback about text-image alignment, a reward model is learned to approximate such feedback, and its output is used to weight the classic loss function of denoising diffusion models [373]. On the contrary, Diffusion Policy Optimization with KL regularization (DPOK) [180] directly applies online reinforcement learning for fine-tuning text-to-image diffusion models, which are optimized using a learned reward model from human feedback [711] and a KL regularization with respect to the pre-trained model.

While very effective, RLHF is not the only existing approach. When human ratings are available in advance for each piece of text, a reward model can be trained offline and then used to fine-tune an LLM [57]. Such a reward model can also be combined with classic MLE to effectively train a language model [354] or used to pre-pend reward tokens to generated text, forming a replay buffer suitable for online, off-policy algorithms to unlearn undesirable properties [408]. Alternatively, Advantage-Leftover Lunch (A-LoL) [24] adopts offline policy gradient with a single-action step assumption (i.e., the entire sequence is a single action) to optimize for pre-trained, sequence-level reward models; to improve learning efficiency, it filters out data points with negative advantages, with the critic based on a frozen reference LLM. Since human ratings might be inaccurate, they can be simulated by applying perturbations on automatically generated scores [466]. Alternatively, the provided dataset of scored text allows for batch (i.e., offline) policy gradient methods to train a chatbot [327]. A very similar approach is also followed by [314], where offline RL is used to train a dialogue system on collected conversations (with relative ratings) filtered to avoid learning misbehavior. Other strategies can be implemented as well. The authors of [199] rely on a learned reward model from human-provided judgment as the other systems discussed above; however, such a reward model is used to optimize a policy directly at inference time for the provided text. Instead of training a policy over multiple inputs and then exploiting it at inference time, they train a different policy for each required input.

Another possibility is to use AI feedback instead of, or in addition to, a human one. Constitutional AI [26] is a method to train a non-evasive and "harmless" AI assistant without any human feedback, only relying on a *constitution* of principles to follow[3]. In a first supervised stage, a pre-trained

[3] While the selection of precepts to be included in the original "constitution" is defined by the researchers at Anthropic, a follow-up project called Collective Constitutional AI [15] involves the participation of humans for crowd-sourcing the underlying principles by means of Polis [608], which is a platform for running online deliberative processes augmented by machine learning algorithms.

LLM is used to generate responses to prompts, and then to iteratively correct them to satisfy a set of principles; once the response is deemed acceptable, it is used to fine-tune the model. Then, RLHF is performed, with the only difference being that feedback is provided by the model itself and not by humans. Reinforcement Learning from AI Feedback (RLAIF) [370] completely replaces human preferences with those from an off-the-shelf LLM for text summarization. The desired overall behavior is induced by careful prompting. RL can fine-tune a Seq2Seq model to generate knowledge useful for a generic question-answer model [402]. This is first re-trained on knowledge generated with GPT-3 (which is prompted by asking to provide the knowledge required to answer a certain question). Then, RL is used to fine-tune the model to maximize an accuracy score using knowledge generated by the model itself as a prompt. To avoid catastrophic forgetting, a KL penalty (with respect to the initial model) is introduced. Reinforced Neural Extractive Summarization (RNES) [704] is instead a method to train an extractive summarizer (i.e., a component that selects which sentences of a given text should be included in its summary) using a reward based on coherence. A model is trained to identify the appropriate next sentence composing a coherent sentence pair; then, such a signal is used to obtain immediate rewards while training the agent (with the ROUGE score as the reward for the final composition). Finally, the authors of [630] propose to limit requests for human feedback to cases in which the learned reward model is uncertain.

Discussion. Reward modeling introduces a greater level of flexibility in RL for generative AI. Generative models can be trained to produce content that humans consider appropriate and of sufficient quality, by aligning them with their preferences. This is useful and in many situations essential: a quantifiable measure might not exist or information to derive it might be hard to obtain. This methodology has already shown its intrinsic value in obtaining accurate, helpful, and useful text. In the same way, these techniques can be applied to other domains in which desired qualities are difficult to quantify or hard to express in a mathematical form, e.g., aesthetically pleasant or personalized (multimodal) content or creative artifacts. A summary of the applications discussed in this paper is reported in Table 3.3.

RLHF has proven to be a highly effective approach. However, it suffers from several open problems [96]. For example, collecting user feedback can be very expensive. Moreover, users might misbehave, whether on purpose or not, be biased, or disagree with each other [189]. Also, they might not correctly represent the population of end users or marginalized categories and comparison-based feedback may not correlate with the desirability of responses [96]. For these reasons, other techniques for modeling preferences might be considered. If human ratings are available in advance, a reward model can be derived from them and used offline. Using AI itself to provide feedback is also an option; notably, AI-based feedback is also used outside the RL paradigm, e.g., to guide DPO [723], to provide verbal feedback to be appended to prompts [593], or to support collaboration with other LLMs at inference time [164, 168]. In addition, other techniques, such as IRL or cooperative IRL [256], can be applied to induce a reward model from human demonstrations.

Reward modeling can be problematic as well. Reducing the diversity of society to a single reward function might cause the majority views to disproportionately prevail [185]. In addition, seemingly well-performing preference-based reward models might fail to generalize to out-of-distribution states [649], thus being prone to reward hacking, i.e., optimizing an imperfect proxy reward function that leads to poor performance according to the true reward function [605]. For these reasons, recent work has focused on eliminating the need for a reward model at all [515, 620].

Finally, the authors of [694] show that, even in cases in which they are aligned, LLMs can still be prompted in ways that lead to undesired behavior. In particular, “jailbreaks” out of alignment can be obtained via single prompts, especially when asking the model to simulate malicious personas [153]. This is more likely to happen in the case of aligned models rather than of non-aligned ones, because of the so-called *waluigi effect*: by learning to behave in a certain way, the model also learns its exact opposite [458]. More advanced approaches are required to mitigate this problem and completely prevent certain undesired behaviors.

Application	Reward	RL Type	Example
Chemistry			
Molecule (graph)	Discriminator + chemical properties	P	[145]
	Pharmacological activity + prior likelihood	P	[18]
	Adversarial loss + desired properties	P	[720]
	Novelty + utility of inhibitors	CB	[735]
Molecule (text)	Discriminator + chemical properties	P	[243]
	Learned desired properties	P	[504]
	Desired property + prior likelihood	P	[476]
	As above + penalty for repetitions	P	[50]
	Partition coefficient	TD	[202]
	Desired property + intrinsic reward	P	[646]
Computer Vision			
Collage	Discriminator on canvas-target pairs + length penalty	P	[369]
Image	Compression or aesthetic or prompt alignment	P	[49]
Stroke painting	Pixel, movement, color reproduction	TD	[738]
	Discriminator on canvas-target pairs	P	[298]
	Background vs foreground + focus	P	[602]
	Two above + adjacent color/position	P	[603]
Text-to-image	RLHF on text-image alignment	RWCE	[373]
	Learned reward model from human feedback	P	[180]
Design			
Building	Performance and aesthetic metrics	P	[263]
Microstructure	Adversarial loss + target properties	P	[467]
Music			
Accompaniment	Log-likelihood for pre-trained models	P	[319]
Music	Discriminator signal	P	[721]
	Music theory rules + log-likelihood for original model	TD	[311]
	Discriminator signal + tonality + ratio of steps	P	[243]
Natural language			
Chatbot	Discriminator signal at each t through MC methods	P	[387]
	Discriminator signal at each t through IRL	P	[590]
	Repetitive or useless answer penalty + mutual information	P	[386]
	Reward from user + likelihood of sub-task completion	HP	[498]
	BLEU + number of proposed API calls	OffP	[737]
	RLHF	P	[482]
	RLHF on helpfulness and harmlessness	P	[25]
	RLHF on helpfulness, harmlessness and correctness	P	[221]
	AI feedback based on a constitution of principles	P	[26]
	Collected human ratings	OffP	[327]
	Learned reward model of human ratings	TD	[630]
	Learned sequence-level reward model of human preferences	OffP	[24]
Extractive summarization	Reward model from human ratings	TD	[199]
	Coherence ratings + ROUGE	P	[704]
Generic text	Discriminator signal	P	[184]
	Sum or product of log-likelihood of tokens from target text	OffP	[485]
	4gram repetition penalty + log-likelihood of target output	P	[355]
	Discriminator signals on coherence and cohesion	P	[115]
	Specific utility function to maximize at inference time	TD	[612]
Knowledge	Accuracy score + kl penalty	P	[402]
Machine translation	BLEU + log-likelihood of target output	P	[524]
	BLEU	P	[23]
	Implicit task-based feedback from users	P	[354]
	Perturbed predicted human ratings	CB	[466]
Plot	Generated vs target verbs distance	P	[642]
Prompt optimization	Aesthetic score + CLIP similarity	P	[264]
Poetry	Discriminator signal	P	[721]
	Fluency + coherence + meaningfulness + quality	P	[719]
	RLHF on relevance, consistency, creativity, form, toxicity	P	[487]
Text continuation	RLHF	P	[741]
Text summarization	ROUGE + log-likelihood of target output	P	[495]
	ROUGESal + Entail	P	[491]
	RLHF	P	[626]
	Reward model trained on human ratings	TD	[57]
	RLAIF	P	[370]
Programming			
Code	Result of unit tests	P	[365]

Table 3.3: Summary of all the applications covered by past research in RL for generative AI, with the considered rewards and the relative references. Type of algorithms used: On-**P**olicy; **Off-P**olicy; **T**emporal-**D**ifference; **C**ontextual **B**andit; **H**ierarchical **P**olicy; **R**eward-**W**eighted **C**ross-**E**ntropy.

3.4 Generative AI and Society

Despite their relative novelty, different studies have already considered generative AI, and in particular foundation models [59] such as LLMs, under the lens of human-related concepts and their potential impact on society. Section 3.4.1 analyzes how human-centered definitions have been applied to LLMs and the risk of anthropomorphizing them due to their human-level performances [77]. Then, we move closer to our focus, introducing how these models have been studied together with creativity (Section 3.4.2). Finally, Section 3.4.3 moves from philosophical to practical issues, introducing the main open questions on generative AI and intellectual property.

3.4.1 AI Anthropomorphization

AI anthropomorphization is now a highly debated topic. The authors of [154] examine its legal and social risks and claim responsible use of AI. Bender and Koller [38] discuss the lack of realistic meaning in models only trained on text because they do not have reference; Piantadosi and Hill [502] oppose this view since we humans as well do not always need a reference. In [39], it is suggested that the autoregressive training nature makes LLMs mere “stochastic parrots”, unable to have any communicative intent. Starting from a similar consideration, Shanahan [585] debates on the lack of beliefs, knowledge, and reasoning in LLMs. The authors of [87] analyze LLMs under various theories of consciousness, finding that current models appear not to be conscious. Agüera y Arcas [6] discusses personhood and understanding in humans and LLMs by drawing several appealing parallelisms; Browning [75] replies that LLMs cannot be entitled to personhood because they lack real (social) agency. The authors of [334, 423] show how LLMs possess formal competence but not functional competence (e.g., formal reasoning, world knowledge, or situation modeling), while those of [559, 665] study how they are limited in theory-of-mind tasks and social intelligence in general. Serapio-García et al. [582] demonstrate LLMs can simulate human personality traits through psychometric methods, also discussing the benefits and concerns of AI anthropomorphization. The idea of using human-intended tests to evaluate the psychological skills of LLMs is achieving high attention now [48, 350, 357, 625]. Similarly, in Section 7.1, we will discuss how LLMs are typically not creative following the main theories of human creativity.

3.4.2 LLMs and Creativity

The potential impact of LLMs on creative fields has been evident since the advent of GPT models [74, 478] and their competitors, e.g., [656]. Research has been conducted to determine whether LLMs can pass human creativity tests, such as the Alternative Uses Test [242], finding that humans are still usually better than LLMs [254, 625]; the authors of [223, 634] explore ways to improve their results and find that both specific prompts and brainstorming steps can enhance their performances. However, their intrinsic lack of intentionality and consciousness should prevent them from being truly creative, as we will discuss in Chapter 7. Wang et al. [679] theorize that generative models should be trained and evaluated by considering the creative abilities of a specific, hypothetical human (or group of humans); however, it is not clear how to practically operationalize this. Commonly, the creative tasks in which AI excels are different from the ones in which humans excel [648]. For this reason, another well-studied path is that of human-AI co-creativity [249]. Human-decision making can be improved by using AI models that diverge from our strategies, increasing the novelty of solutions [592]. Similarly, LLMs can be a powerful helper in the hands of a skillful writer [586], especially for translation and reviewing [101].

3.4.3 Generative AI and Copyright

There has been a long-standing interest in the copyright issues around generative AI [86, 554]. Different legal issues are at play when considering the entire generative-AI supply chain [374]. Whether training neural networks on protected data is lawful has been highly debated across different national legislations [379], e.g., U.S. fair use [268, 613], EU text and data mining exceptions [208, 614], and others [240, 556]. The other main debate has focused on whether a machine-generated work is protected by copyright or not [136, 214, 235] and on the question of who might own its ownership in the future [60]. However, other topics have also been considered, such as whether the model output can infringe the reproduction right [218, 676] or how the trained model can be protected [479, 607]. Section 7.3 will cover the main aspects of all these issues.

4 Creativity for Reinforcement Learning

Deep RL has emerged as a very effective mechanism for dealing with complex and intractable AI tasks of different nature. Model-free methods that essentially learn by trial and error have solved challenging games [449], performed simulated physics tasks [395], and aligned large language models with human values [482]. However, RL commonly requires a very large amount of collected experience, especially compared to the one required by humans [660], limiting its applications to real-world tasks.

Model-based RL [639] represents a promising direction toward sample efficiency. Learning a world model capable of predicting the next states and rewards conditioned on actions allows the agent to plan [569] or build additional training trajectories [252]. In particular, recent imagination-based methods [258, 259, 260, 261, 440] have shown remarkable performance simply by learning from imagined episodes within a learned latent space. As introduced in Section 1.2, such imagined trajectories are commonly mentioned as *dreams* since also the human brain simulates actions and their consequences during sleep [581]. However, these dreams are nothing like *human* dreams, as they essentially try to mimic reality as best as possible. According to the *Overfitted Brain* hypothesis [281], dreams happen to allow generalization in the human brain. In particular, it is by providing hallucinatory and corrupted content [280] that are far from the limited daily experiences (i.e., the training set) that dreaming helps prevent overfitting. In this chapter, we build on this intuition and ask: *can more creative, human-like "dreams" help RL agents generalize better when dealing with limited experience?*

To answer this, we explore whether this type of experience augmentation based on dream-like generated trajectories helps generalization and, consequently, improves learning. In particular, we consider the situation in which only a limited amount of real experience (analogously to "daylight activities" for humans) is available, and we question whether building a world model upon it and leveraging it to generate dream-like experiences improves the agent's generalization capabilities. To simulate the hallucinatory and corrupted nature of dreams, in Section 4.1 we propose to transform the classic imagined trajectories with *generative augmentations*, i.e., through interpolation with random noise [680], DeepDream [451], or critic's return optimization (similar to class visualization [601]). These three alterations should provide divergent but also meaningful and valuable experiences for the agent learning. Then, Section 4.2 evaluates them under different scenarios on four ProcGen environments [126], a standard suite for generalization in RL [345].

4.1 Improving Generalization through Generative Learning

By starting from the models presented in Section 2.2.2, our method proposes: to learn a latent world model from real experience; to augment the imagined trajectories to resemble human dreams (Section 4.1.1); and to exploit such new trajectories to learn policies that are more keen to generalize (Section 4.1.2).

4.1.1 Generating Human-Like Dreams

Given a trained world model, we can use it to construct imagined trajectories as detailed in Section 2.2.2. Notably, instead of starting each trajectory from a real collected state (as is commonly done in the literature), we start from randomly generated states: $\mathbf{\hat{s}_0^{im}} = \left(\mathbf{h_0^{im}}, \mathbf{\hat{z}_0^{im}}\right)$ with

$$\begin{aligned}
&\mathbf{h_{init}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})\,,\\
&\mathbf{\hat{z}_{init}} = \texttt{one\_hot}(u_{1:C})\,, u_c \sim \mathcal{U}(0, J-1) \;\text{ for } c = 1 \dots C,\\
&\mathbf{h_0^{im}} = f_{\boldsymbol{\theta}}(\mathbf{h_{init}}, \mathbf{\hat{z}_{init}}, a_{init})\,,\\
&\mathbf{\hat{z}_0^{im}} \sim p_{\boldsymbol{\theta}}\big(\mathbf{h_0^{im}}\big)\,,
\end{aligned} \tag{4.1}$$

where J is the number of classes each of the C categorical variables can assume, $\texttt{one\_hot}(\cdot)$ transforms a list of categorical variables into a vector of one-hot encoded vectors, and a_{init} is a zero vector.

To obtain more human-like dreams, we leverage the world model to propose three perturbation strategies (see Figure 4.1 for a summary of the process):

- **Random jump**, i.e., interpolation between the current state $\mathbf{\hat{s}_t^{im}} = \left(\mathbf{h_t^{im}}, \mathbf{\hat{z}_t^{im}}\right)$ and a random noise state (similar to [680]). In particular,

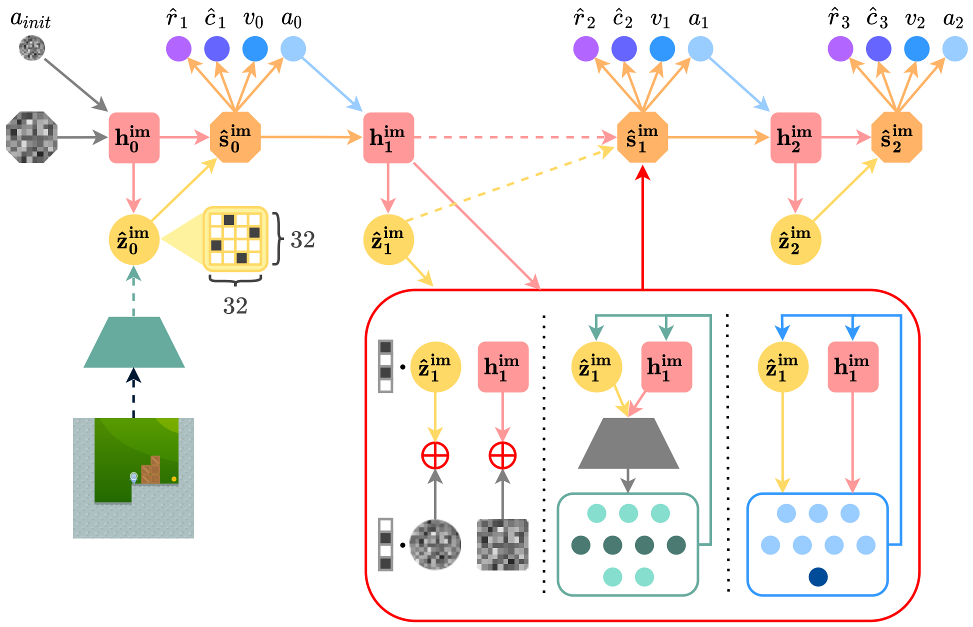

a_{init}
$\hat{r}_1$ $\hat{c}_1$ v_0 a_0
$\hat{r}_2$ $\hat{c}_2$ v_1 a_1
$\hat{r}_3$ $\hat{c}_3$ v_2 a_2
$\mathbf{h}_0^{\mathbf{im}}$
$\hat{\mathbf{s}}_0^{\mathbf{im}}$
$\mathbf{h}_1^{\mathbf{im}}$
$\hat{\mathbf{s}}_1^{\mathbf{im}}$
$\mathbf{h}_2^{\mathbf{im}}$
$\hat{\mathbf{s}}_2^{\mathbf{im}}$
$\hat{\mathbf{z}}_0^{\mathbf{im}}$
32
32
$\hat{\mathbf{z}}_1^{\mathbf{im}}$
$\hat{\mathbf{z}}_2^{\mathbf{im}}$
$\hat{\mathbf{z}}_1^{\mathbf{im}}$ $\mathbf{h}_1^{\mathbf{im}}$
$\hat{\mathbf{z}}_1^{\mathbf{im}}$ $\mathbf{h}_1^{\mathbf{im}}$
$\hat{\mathbf{z}}_1^{\mathbf{im}}$ $\mathbf{h}_1^{\mathbf{im}}$

Figure 4.1: At imagination time, we start from a **random latent state** or as an alternative (dashed lines) an **encoded image**. Then, we only leverage the predicting capabilities of our world model to obtain future **latent states** (the concatenation of a **discrete latent vector** and a **recurrent hidden state**), **rewards** and **termination bits** given the **actions** from the agent. To introduce a **dream-like transformation**, we modify the current **latent state** with a small probability by doing one of three operations: *interpolate* it with **random noise**; *DeepDream* its corresponding observation from the **decoder** by maximizing the activation of the **encoder** last convolution layer; *optimize* it to maximize the divergence of the **critic output**.

we perturb the hidden state $\mathbf{h}_\mathbf{t}^{\mathbf{im}}$ by adding a random vector $\mathbf{h}_{\mathbf{rand}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Instead, our transformation over the latent state $\hat{\mathbf{z}}_\mathbf{t}^{\mathbf{im}}$ can be formalized as:

$$\begin{aligned}
&\hat{\mathbf{z}}_\mathbf{t}^{\mathbf{im}} = \texttt{one\_hot}\big(\lambda \cdot \texttt{reverse\_one\_hot}\big(\hat{\mathbf{z}}_\mathbf{t}^{\mathbf{im}}\big) + (1-\lambda)\cdot u_{1:C}\big)\,,\\
&\lambda \sim \texttt{Bin}(C, p_{jump})\,,\\
&u_c \sim \mathcal{U}(0, J-1) \;\; \text{for } c = 1 \ldots C
\end{aligned} \tag{4.2}$$

where $\texttt{reverse\_one\_hot}(\cdot)$ inverts the one-hot encoding, i.e., recovers the list of categorical variables, and $p_{jump} = 0.5$ is the probability of making a *jump*. In other words, each categorical variable is changed into a randomly sampled class with probability p_{jump}. This simulates the corruption of dream content and the sudden visual changes we commonly experience during REM sleep [13].

- **DeepDream**, i.e., by iteratively adjusting the image reconstructed from the state to maximize the firing of a model layer [451]. Specifically, we consider the last convolutional layer of the encoder, which should learn the building elements of real images. Given $q_{\boldsymbol{\theta}}^{LC}(\cdot)$ as the activation of the last convolutional layer of dimension D, we transform the hidden state $\mathbf{h}_\mathbf{t}^\mathbf{im}$ and the latent state $\hat{\mathbf{z}}_\mathbf{t}^\mathbf{im}$ via gradient ascent over the following objective:

  $$g_{dd} = \nabla_{\mathbf{h}_\mathbf{t}^\mathbf{im}, \hat{\mathbf{z}}_\mathbf{t}^\mathbf{im}} \frac{\sum_{i=1}^{D} q_{\boldsymbol{\theta}}^{LC}\big(p_{\boldsymbol{\theta}}\big(\mathbf{h}_\mathbf{t}^\mathbf{im}, \hat{\mathbf{z}}_\mathbf{t}^\mathbf{im}\big)\big)_i}{D}. \tag{4.3}$$

  This simulates the hallucinatory nature of dreams.

- **Value diversification**, i.e., by iteratively adjusting the state $\hat{\mathbf{s}}_\mathbf{t}^\mathbf{im}$ to maximize the squared difference between the value of the critic prediction at iteration τ and iteration 0. We perform a gradient ascent over the following objective:

  $$g_{vd} = \nabla_{\mathbf{h}_\mathbf{t}^\mathbf{im}, \hat{\mathbf{z}}_\mathbf{t}^\mathbf{im}} \Big(v_{\boldsymbol{\phi}}\big(\mathbf{h}_\mathbf{t}^\mathbf{im}, \hat{\mathbf{z}}_\mathbf{t}^\mathbf{im}\big) - v_{\boldsymbol{\phi}}\big(\mathbf{h}_\mathbf{t}^\mathbf{inp}, \hat{\mathbf{z}}_\mathbf{t}^\mathbf{inp}\big)\Big)^2, \tag{4.4}$$

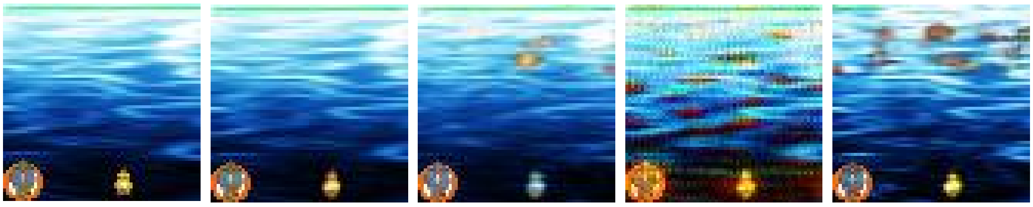
  where $\hat{\mathbf{s}}_\mathbf{t}^\mathbf{inp} = (\mathbf{h}_\mathbf{t}^\mathbf{inp}, \hat{\mathbf{z}}_\mathbf{t}^\mathbf{inp})$ is the state before optimization. The squared difference is considered to optimize for positive and negative changes in the critic's prediction. In addition, at each iteration, $\hat{\mathbf{z}}_\mathbf{t}^\mathbf{im}$ is transformed to keep it as a vector of one-hot categorical variables. The value diversification transformation suddenly introduces or removes goals or obstacles, simulating the narrative content and the fact that dreams commonly resemble daily aspects that are significant to us, especially threatening events [532]. In fact, simulating negative experiences might allow an agent to learn what to avoid in practice.

Figure 4.2 reports a visual example of the three transformations.

We alter each state $\hat{\mathbf{s}}_\mathbf{t}^\mathbf{im}$ with a small probability $\epsilon_{dream} = \frac{1}{H}$ with H imagination horizon. In this way, each trajectory includes, on average, one transformed state. Algorithm 1 summarizes a dreaming step.

4.1.2 Learning by Day and Night

Our method can be divided into two stages. During the first, our agent plays a limited number of real episodes (the *day* experience). These episodes can



(a) Original. (b) Decoded. (c) Rnd jump. (d) DeepDream. (e) Value div.

Figure 4.2: An example of the three generative augmentations on a state from the Plunder environment.

be used to only train the world model, as in Dreamer, or to train both the world model and the agent in an E2C-like setting [684], where the agent receives the encoding of the real observation by the world model as the state. We then leverage the world model to generate additional dreamed episodes (the *night* experience) used to train the agent. While usually the two stages are repeated multiple times, nothing prevents us from only repeating them once, clearly separating the day and night phases. Algorithm 2 summarizes the entire learning process.

Algorithm 1 Learning by Dreaming

```
Require $\boldsymbol{\theta}$-parameterized world, $\boldsymbol{\phi}$-parameterized agent, $\mathbf{\hat{s}_0^{im}}$ initial imagination states, $H$ imagination horizon, rnd_init boolean stating whether to sample random initial states.
if rnd_init then
    Sample $||\mathbf{\hat{s}_0^{im}}||$ new random states $\mathbf{\hat{s}_0^{im}}$ according to Equation 4.1.
end if
for timestep $t = 0 \dots H-1$ do
    Sample random number $y \sim \mathcal{U}(0,1)$.
    if $y \leq \frac{1}{H}$ then
        Transform $\mathbf{\hat{s}_t^{im}}$ via either Equation 4.2, 4.3, or 4.4.
    end if
    Compute $a_t \sim \pi_{\boldsymbol{\phi}}(a_t|\mathbf{\hat{s}_t^{im}})$.
    Compute $\mathbf{\hat{s}_{t+1}^{im}} = (\mathbf{\hat{h}_{t+1}}, \mathbf{\hat{z}_{t+1}})$, $\mathbf{\hat{h}_{t+1}} = f_{\boldsymbol{\theta}}(\mathbf{\hat{h}_t}, \mathbf{\hat{z}_t}, a_t)$, $\mathbf{\hat{z}_{t+1}} \sim p_{\boldsymbol{\theta}}(\mathbf{\hat{h}_{t+1}})$.
    Compute $\hat{r}_{t+1} \sim p_{\boldsymbol{\theta}}(\mathbf{\hat{h}_{t+1}}, \mathbf{\hat{z}_{t+1}})$.
    Compute $\hat{c}_{t+1} \sim p_{\boldsymbol{\theta}}(\mathbf{\hat{h}_{t+1}}, \mathbf{\hat{z}_{t+1}})$.
end for
Update $\phi$ through Equations 2.20 and 4.5 using generated experience.
```

The latent world model is trained as detailed in Section 2.2.2. As far as the agent is concerned, following [261] we adopt an actor-critic architecture that works on the latent state $\mathbf{s_t} = (\mathbf{h_t}, \mathbf{z_t})$. However, instead of using REINFORCE, we train it with PPO [573], which we find helpful to obtain a more stable training. The overall loss is that from Section 2.2.1.

While DreamerV3 models the value function as the reward with the two-hot encoded strategy, we found the value clipping strategy of PPO more effective. Similar to the policy loss, its loss function is defined as follows:

$$\begin{aligned} L_t^{VF}(\boldsymbol{\phi}) = \max(&\left(v_{\boldsymbol{\phi}}(\mathbf{s_t}) - V_t^{target}\right)^2, \\ &\left(v_{\boldsymbol{\phi}_{old}}(\mathbf{s_t}) + \texttt{clip}((v_{\boldsymbol{\phi}}(\mathbf{s_t}) - v_{\boldsymbol{\phi}_{old}}(\mathbf{s_t}), -\epsilon, +\epsilon) - V_t^{target}\right)^2), \end{aligned} \tag{4.5}$$

where $v_{\boldsymbol{\phi}_{old}}(\mathbf{s_t})$ is the value function prediction at collecting time.

Finally, the advantage is estimated with GAE [571] (see Section 2.2.1). While DreamerV3 [261] normalizes it with the running batch percentile of the discounted return, we find that the normalization scheme of PPO is more effective for ProcGen environments. In particular, we normalize the symlog-transformed rewards by dividing it by the running standard deviation of rewards (i.e., of all the rewards collected so far), while the advantages are standardized by subtracting their mean and dividing by their standard deviation at the single batch level.

Algorithm 2 Learning to Generalize by Day and by Night

```
**Require** $S$ number of seed episodes, $E_{wup}$ warmup epochs, $E$ epochs, $E_{ft}$ epochs, $U$ update steps, $B_w$ world batch size, $L$ sequence length, $H$ imagination horizon, $T_{envs} = T_{ep} \cdot N_{envs}$ environment steps per epoch, $N_{test}$ number of episodes for testing, rnd_init boolean stating whether to use random initial states for dreaming.
**Initialize** neural network parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ randomly.
**Initialize** dataset $\mathcal{D}$ with $S$ random seed episodes.
**Warmup** world model by training it for $E_{wup}$ epochs.
$o_1 \leftarrow$ env.reset()
**for** train epoch $e_d = 1 \dots E$ **do**
    **for** update step $u = 1 \dots U$ **do**
        Draw $B_w$ data sequences $\{(\mathbf{o_t}, a_t, r_{t+1}, c_{t+1})\}_{t=k}^{k+L} \sim \mathcal{D}$.
        Compute all $\mathbf{s_t} = (\mathbf{h_t}, \mathbf{z_t})$, $\mathbf{h_t} = f_{\boldsymbol{\theta}}(\mathbf{h_{t-1}}, \mathbf{z_{t-1}}, a_{t-1})$, $\mathbf{z_t} \sim q_{\boldsymbol{\theta}}(\mathbf{h_t}, \mathbf{o_t})$.
        Update $\boldsymbol{\theta}$ through Equation 2.26.
        **if** $E_{ft} = 0$ **then**
            Train agent with Algorithm 1 given $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, all $\mathbf{s_t}$, $H$, and rnd_init.
        **end if**
    **end for**
    **for** timestep $t = 1 \dots T_{envs}$ **do**
        Compute $\mathbf{s_t} = (\mathbf{h_t}, \mathbf{z_t})$, $\mathbf{h_t} = f_{\boldsymbol{\theta}}(\mathbf{h_{t-1}}, \mathbf{z_{t-1}}, a_{t-1})$, $\mathbf{z_t} \sim q_{\boldsymbol{\theta}}(\mathbf{h_t}, \mathbf{o_t})$.
        Compute $a_t \sim \pi_{\boldsymbol{\phi}}(a_t|\mathbf{s_t})$.
        $\mathbf{o_{t+1}}, r_{t+1}, c_{t+1} \leftarrow$ env.step($a_t$).
    **end for**
    **if** $E_{ft} \neq 0$ **then**
        Update $\boldsymbol{\phi}$ through Equations 2.20 and 4.5 with collected experience.
    **end if**
    Add experience to dataset $\mathcal{D} \leftarrow \mathcal{D} \cup \left\{(o_t, a_t, r_{t+1}, c_{t+1})_{t=0}^{T_{envs}-1}\right\}$.
    Evaluate $\pi_\phi$ on test_env for $N_{test}$ episodes.
**end for**
**for** fine-tune epoch $e_n = 1 \dots E$ **do**
    **for** update step $u = 1 \dots U$ **do**
        Draw $B_w$ data sequences $\{(\mathbf{o_t}, a_t, r_{t+1}, c_{t+1})\}_{t=k}^{k+L} \sim \mathcal{D}$.
        Compute all $\mathbf{s_t} = (\mathbf{h_t}, \mathbf{z_t})$, $\mathbf{h_t} = f_{\boldsymbol{\theta}}(\mathbf{h_{t-1}}, \mathbf{z_{t-1}}, a_{t-1})$, $\mathbf{z_t} \sim q_{\boldsymbol{\theta}}(\mathbf{h_t}, \mathbf{o_t})$.
        Train agent with Algorithm 1 given $\boldsymbol{\theta}$, $\boldsymbol{\phi}$, all $\mathbf{s_t}$, $H$, and rnd_init.
    **end for**
    Evaluate $\pi_\phi$ on test_env for $N_{test}$ episodes.
**end for**
```

4.2 Experiments

In the following, we present several experiments using ProcGen [126], a simple yet rich set of environments for RL generalization evaluation. While the chosen baselines have been presented on more classic environments, the scope of our research is specifically on the effect of generative augmentations on generalization, therefore we only consider ProcGen. Moreover, we focus on limited-resource scenarios. This has several advantages: it is compatible with more practical real-world use cases, where access to the environment might come at a high cost, and imagination-based RL can be a reasonable strategy; it reduces the footprint of our research, making it more aligned with the quest toward green AI [574].

4.2.1 Setup

ProcGen is a suite of 16 procedurally generated game-like environments. To benchmark the generalization capabilities of our approach, we use only a small subset of the distribution of levels ($N = 200$) to train the agent and the full distribution to test it. Due to resource constraints, our experiments consider the ProcGen suite in *easy* mode; we limit the collected real experience to 1e+6 steps, far below the suggested 2.5e+7 and the 1.1e+8 used in DreamerV3 [261]. We evaluate our method across four ProcGen environments, each presenting unique and challenging properties: CaveFlyer (open-world navigation with sparse rewards), Chaser (grid-based game with highly dense rewards), CoinRun (left-to-right platformer with highly sparse rewards), and Plunder (war game with dense rewards).

Since our goal is to verify the effect of generative augmentations on imagination-based RL, we use a “plain” Dreamer as our baseline. In addition, we also show the performance of an agent only trained on collected, real experience. Specifically, we use the same Impala-based [178] PPO agent proposed in the original ProcGen paper [126], again trained on the same number of timesteps. The full implementation details are reported in Appendix A.1.

4.2.2 Results

We develop our experiments along four research questions to evaluate our approach in different scenarios depending on the use cases.

RQ1. Can generative augmentations improve the generalization capabilities of agents fully trained in imagination?

The first set of experiments considers training the agent only with imagined trajectories, i.e., as is commonly done with Dreamer models. Following them, we also consider real, collected states as starting points for dreamed experiences. From a practical perspective, this means setting $E_{ft} = 0$ and $\mathsf{rnd\_init} = \text{False}$ into Algorithm 2.

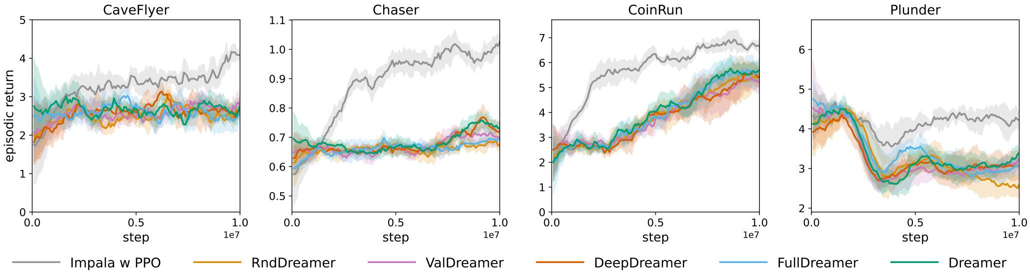


Figure 4.3: The total rewards received in the full distribution of levels for our four methods (with either random jumps, deepdream, or value maximization and with all of them) and the Dreamer baseline when fully trained with imagination (RQ1). For reference, we also report the total rewards received by an Impala-based PPO agent trained without imagination. Results report the mean and standard deviation across 5 seeds.

Figure 4.3 reports the rewards on test environments over training time. As apparent from the plots, our methods do not provide any advantage over the classic Dreamer agent, achieving at best its performance. While the different generative augmentations lead sometimes to different behaviors (e.g., random interpolation being the worst on Plunder but one of the best on CoinRun), alternating them leads to better results only on CoinRun. This might mean that different environments are better suited for different transformations. Nonetheless, all the imagination-based methods are far from competing with the baseline. A possible explanation is that the number of training steps is insufficient. Indeed, Dreamer methods [259, 260, 261] are usually trained for longer than standard methods. This might be necessary because the world model has to learn the dynamics of the environment before being capable of providing useful training signals to the agent, increasing the total timesteps needed to converge.

RQ2. Can generative augmentations improve the generalization capabilities of agents pre-trained on real experience?

The second set of experiments involves training the agent alongside the world model using collected real-world experiences. Subsequently, the learned world model is utilized to generate simulated, dream-like trajectories, which are then used to fine-tune the agent. This has two potential advantages: firstly, the trajectories are constructed with an already trained world model, i.e., with a model that should already have acquired the necessary knowledge about its dynamics; and secondly, the experience collected by the agent in the real environment is not "wasted", maximizing the available limited resources to train the agent on the largest possible amount of data. In this scenario, we set $E_{ft} = E$, while keeping rnd_init = False into Algorithm 2.

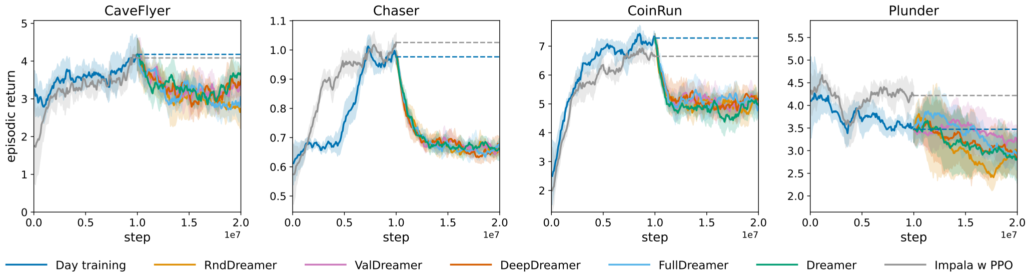


Figure 4.4: The total rewards received in the full distribution of levels for our four methods (with either random jumps, deepdream, or value maximization and with all of them) and the Dreamer baseline when first trained on collected experience and then fine-tuned with imagination (RQ2). The left half of the chart reports pre-training (shared by all the methods) and, for reference, the Impala-based PPO agent trained without imagination; the right half reports fine-tuning effects. The mean and standard deviation are calculated across 5 seeds.

Figure 4.4 summarizes the results for the four environments. The pre-training phase in the real environment is comparable with (when not better than) the model-free baseline in three environments out of four; however, imagination-based fine-tuning is not helpful and even decreases generalization capabilities. Notably, for Plunder environment, our method with all three transformations is effective in improving results in the early stages but with catastrophic results in the long run. Overall, it is clear that this fine-tuning tends to have an overfitting effect, which is harmful from a generalization standpoint. One possible explanation is that using the same experience multiple times (both during *day* activity and repeatedly during *night*) is problematic, and our generative augmentations do not consistently mitigate this issue.

RQ3. Can real experience collected by an expert agent help achieve better generalization capabilities through dreaming?

We have seen so far that imagination-based RL is not very effective with limited resources. As mentioned before, one possible problem is the quality of collected experiences: if the agent is not good enough, the real trajectories will not be sufficiently significant to train the world model, leading to inadequate imagined trajectories (e.g., those that never lead to positive rewards). Moreover, with resource constraints, it would be illogical to train a pair of world and agent models, as we have shown them to perform just like standard IMPALA-based agents. For these reasons, we now consider the case in which we already possess real trajectories from an *expert* model (e.g., with state-of-the-art results). This can occur either because we have the expert model and use it to play in the environment, or because we have an offline dataset of historical, high-quality experiences from that environment. We investigate whether imagination-based RL can be used to design an agent capable of generalizing in that environment without direct access to it, and whether our generative augmentations can lead to better performance. From a practical perspective, we collect 1e+6 timesteps from the training environment by leveraging the Impala-based PPO agent as described in the original ProcGen paper [126], i.e., trained over 2.5e+7 timesteps. Then, we train the world model for E epochs over such data, and finally, we train the agent for $E_{ft} = E$ epochs. Again, we set rnd_init = False.

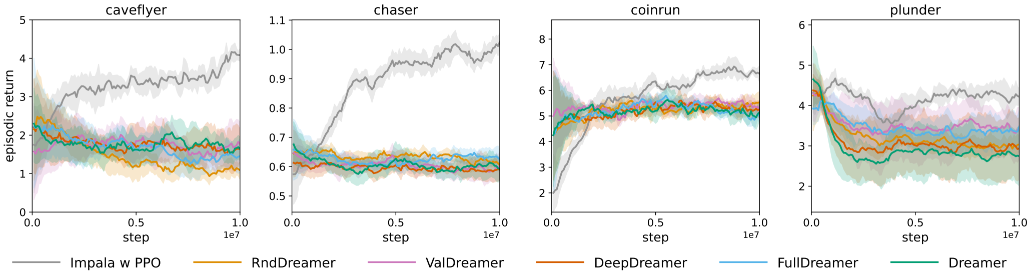


Figure 4.5: The total rewards received in the full distribution of levels for our four methods (with either random jumps, deepdream, or value maximization and with all of them) and the Dreamer baseline when fully trained with imagination on a world model learned from expert-collected trajectories (RQ3). For reference, we also show the total rewards received by an Impala-based PPO agent trained without imagination. The mean and standard deviation are calculated across 5 seeds.

Figure 4.5 reports the results of our expert-based imagination experiments. Again, imagination-based approaches are not competitive with respect to the baseline performance. However, in the case of the Plunder environment, we observe performance comparable with the *day* training from RQ1; and the generative augmentations (especially value maximization) seem to provide meaningful improvements over the classic, non-transformed imagination of Dreamer.

RQ4. Can a world model trained on expert-level experience help achieve better generalization capabilities through dreaming?

The last setting we consider is the same as before, but now we investigate whether it is possible to train an agent with generalization capabilities even without access to the data collected by the expert, but only to the world model trained on it. More specifically, we consider the same world models as for the analysis of RQ3., but now we leverage random initial states to generate imagined trajectories, i.e., we set rnd_init = True.

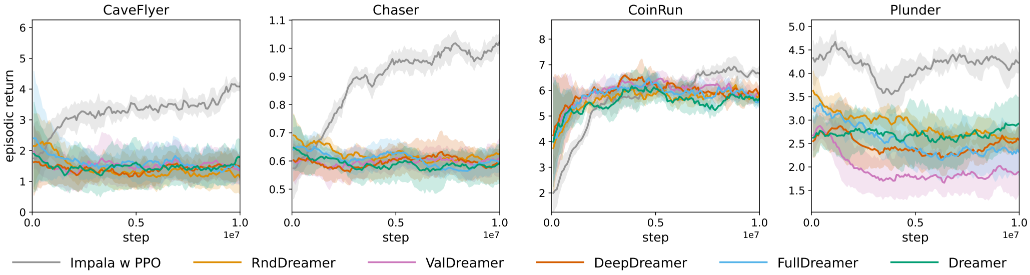


Figure 4.6: The total rewards received in the full distribution of levels for our four methods (with either random jumps, deepdream, or value maximization and with all of them) and the Dreamer baseline trained with imagination from random initial states on a world model learned from expert-collected trajectories (RQ4). For reference, we also show the total rewards received by an Impala-based PPO agent trained without imagination. The mean and standard deviation are calculated across 5 seeds.

Figure 4.5 shows the total rewards of our expert-based imagination experiments without access to expert-collected states. Compared to the results from Figure 4.5, using random initial states helps achieve better performance on CoinRun; in particular, our methods have better generalization performance than the baseline for the first half of the training, but then stop improving. In other environments, the results are either comparable or worse. While the generative augmentations seem to provide meaningful improvements over the classic, non-transformed imagination of Dreamer at the beginning of training, in the long run, the effect vanishes, sometimes with catastrophic results (as seen in Plunder).

5 Reinforcement Learning for Creativity

In the previous chapter, we explored how creativity can improve RL, providing specific properties beyond mere task performance. In the same way, RL can provide useful contributions to developing creative machines by providing a learning framework to optimize for more specific goals, freeing the model from training data.

As discussed in Section 2.1, self-supervised learning makes the LLMs generate samples as close as possible to the training data distribution. In addition, reinforcement learning from human feedback [118], though necessary to generate appropriate and accurate responses, tends to reduce the output diversity [346]. On the contrary, LLMs for creative tasks should produce more novel and surprising texts that maintain a high level of correctness and adherence to the request.

In this chapter, we propose to capture these aspects through a context-based score for value and originality. We describe it in Section 5.1, together with various methods to maximize it via reinforcement learning; finally, we evaluate them on poetry generation and math problem resolution. In addition, Section 5.2 presents a strategy to use such a score at inference time through contextual learning.

5.1 A Context-Based Score for Valuable and Original Generation

By starting from mutual information, in Section 5.1.1, we derive a new optimization problem where, given a specific input, the desired output can be found by simultaneously maximizing the conditional probability under the generative model of the input given the output, and minimizing the conditional probability under the generative model of the output given the input. In this way, we optimize toward solutions that are appropriate for the request given in input but also different from what we would normally obtain from the model (which we assume to be something not novel nor surprising). This score can then be directly used as a reward in a reinforcement learning problem to fine-tune pre-trained models, pushing them toward more divergent but still valuable solutions. Finally, in Section 5.1.2, we evaluate our approach on two different case studies: poetry generation and math problem resolution.

5.1.1 Approach

Mutual information represents a reasonable theoretical way to study the relationship between contextual, prior information and a produced posterior outcome. As discussed in depth in Section 1.2, creativity depends on the context in which the product is created, as the context provides the task identification and the domain information necessary to generate and validate the outcome. In turn, the output aims to solve the given task and provide a meaningful, original contribution to the current domain. Thus, our proposed score has its roots in mutual information. More specifically, we start from the (point-wise) mutual information between two variables x and y defined as follows:

$$I(x, y) = h(x) - h(x|y) = h(y) - h(y|x) \tag{5.1}$$

where the entropy is $h(a) = -\log p(a)$, therefore:

$$I(x, y) = \log p(x|y) - \log p(x) = \log p(y|x) - \log p(y). \tag{5.2}$$

Let us now call $x = S$ (for *source*, i.e., our context) and $y = T$ (for *target*, i.e., our product):

$$I(S, T) = \log p(S|T) - \log p(S) = \log p(T|S) - \log p(T). \tag{5.3}$$

We can generalize $I(S, T)$ with two scaling factors:

$$I(S, T, \lambda_1, \lambda_2) = \lambda_1 \log p(T|S) - \lambda_2 \log p(T), \tag{5.4}$$

where $I(S, T)$ is just $I(S, T, 1, 1)$.
By applying the Bayes theorem, i.e., $\log p(a|b) = \log p(b|a) + \log p(a) - \log p(b)$, we can substitute the $\log p(T)$ term as follows:

$$\begin{aligned} I(S,T,\lambda_1,\lambda_2) =& \lambda_1 \log p(T|S) - \lambda_2 \log p(T|S) - \\ & \lambda_2 \log p(S) + \lambda_2 \log p(S|T) \\ =& (\lambda_1 - \lambda_2) \log p(T|S) + \\ & \lambda_2 \log p(S|T) - \lambda_2 \log p(S). \end{aligned} \tag{5.5}$$

Since our goal is to find the optimal T for a given S, the last term can be ignored. Moreover, we now define $\lambda_v = \lambda_2$ and $\lambda_o = \lambda_2 - \lambda_1$, thus obtaining:

$$\max_T \lambda_v \log p(S|T) - \lambda_o \log p(T|S). \tag{5.6}$$

Let us now consider the case where $\lambda_v, \lambda_o > 0$, for example, $\lambda_v = \lambda_o = 1$. Solving this maximization problem means finding the target T that maximizes the posterior probability of S while also being unlikely given S. In other words, the optimal T must be unexpected and diverse from $p(T|S)$, but it must also be explainable by S. While $-\log p(T|S)$, commonly known as surprisal [658], is widely used to measure diversity and surprise [31], the other term, $\log p(S|T)$, can be used to measure value and effectiveness. Indeed, if the request (e.g., a problem or a task) can be easily predicted from the outcome, the outcome must be a (good) example of that task or a correct solution for that problem. These two properties match the effectiveness and originality requirements for creativity following Runco and Jaeger [546], thus promising to be a potential score for creativity. While other prominent definitions such as Boden's [53] seek to include a third requirement for creativity, i.e., novelty, we believe that by considering only a specific context, novelty and surprise become indistinguishable. In particular, being novel in a specific context S means doing something different from what has been previously experienced under S by the creator. However, since a self-supervised-trained model can be seen as a compression of training data (as we will argue in Section 7.3.3), $-\log P(T|S)$ becomes not only a measure of unexpectedness but also of novelty. Therefore, we choose to refer to this term as *originality*, as it encompasses both novelty and surprise. On the other hand, $\log P(S|T)$ is less ambiguous and maps directly into *value*.

In summary, our CoVO (**Co**ntext-based **V**alue and **O**riginality) score for a target T given a source S on a reference probability distribution p is defined as:

$$s_{CoVO}(S,T,p) = \underbrace{\lambda_v \log p(S|T)}_{\text{Value}} \underbrace{- \lambda_o \log p(T|S)}_{\text{Originality}}. \tag{5.7}$$

In the context of a θ-parameterized autoregressive model such as an LLM, $p(A|B)$ can be expressed as $\prod_{i=1}^{N} p_\theta(a_i|A_{1:i-1}, B)$ where $A = a_1 \ldots a_N$. However, considering just the product of all the conditioned probabilities for an optimization problem would lead to preferring shorter sequences. To avoid this, we propose to use the N-th root: $\sqrt[N]{\prod_{i=1}^{N} p_\theta(a_i|A_{1:i-1}, B)}$. By applying the properties of the logarithm, the whole formulation becomes:

$$\lambda_v \frac{\sum_{i=1}^{|S|} \log p_\theta(s_i|S_{1:i-1}, T)}{|S|} - \lambda_o \frac{\sum_{j=1}^{|T|} \log p_\theta(t_i|T_{1:j-1}, S)}{|T|}. \tag{5.8}$$

We would like to highlight another interesting issue. The vocabulary of an LLM can be extremely large, thus causing $p_\theta(a|b)$ to be small even when a is the most probable event given b. In particular, in the case of an LLM generating T given S and then evaluating both $p_\theta(T|S)$ and $p_\theta(S|T)$, this can lead to a strong discrepancy between the magnitude of value and diversity: since T has been sampled from p_θ, its probability would be high by definition, while there may be different ways (possibly through synonyms) to define T, leading to smaller probability of S. Inspired by [415], we propose to normalize $p_\theta(a|b)$ via $n' = \frac{n - n_{min}}{n_{max} - n_{min}}$. For probabilities, $n_{min} = 0$, while $n_{max} = \max p_\theta(b)$, thus obtaining the overall mapping for p_θ: $\frac{p_\theta(a_i|A_{1:i-1}, B)}{\max p_\theta(A_{1:i-1}, B)}$. Again, by using the properties of the logarithm, the whole formulation becomes:

$$\begin{aligned}
&s_{CoVO}(S, T, p_\theta) = \lambda_v s_v(S, T, p) + \lambda_o s_o(S, T, p), \text{ with} \\
&s_v(S, T, p_\theta) = \frac{\sum_{i=1}^{|S|} (\log p_\theta(s_i|S_{1:i-1}, T) - \max \log p_\theta(S_{1:i-1}, T))}{|S|}, \\
&s_o(S, T, p_\theta) = \frac{\sum_{j=1}^{|T|} (\log p_\theta(t_i|T_{1:j-1}, S) - \max \log p_\theta(T_{1:j-1}, S))}{|T|}.
\end{aligned} \tag{5.9}$$

Despite these adjustments, the two parts of our score still have fundamental differences in the magnitude of variations, e.g., given the same source S, small variations in the target T cause larger variations in the originality part than in the value part. To address this issue, we propose to normalize the two parts independently. We implement and test two different methodologies: single-batch normalization, where each component is standardized using its mean and standard deviation within a single batch; and full-training normalization, where each component is standardized using the mean and standard deviation over the entire training period, maintaining running statistics across batches.

Finally, calculating $p(S|T)$ is not trivial. Since LLMs are trained to complete a text, it is implausible that they would generate the source text immediately after the target text (which, we ought to remember, is generated immediately after the source text). To solve this, we consider an approximation $p(S|T')$, where $T' = T + Q$, and Q represents an additional question such as "How would you describe this text?" or an analogous formulation whose only goal is to increase the likelihood of the source text S (as well as of alternative sources).

Once the CoVO score has been defined, its adoption in a reinforcement learning framework is straightforward. As introduced in Section 2.3, the pretrained LLM plays the role of the agent; the state is the prompt together with the text generated so far; and the reward is zero for all the timesteps except the last one, where it is our CoVO score. Then, the agent can be trained with any online policy gradient method; we suggest using PPO (see Section 2.2.1), as is now the state-of-the-art RL algorithm for training language models.

Finally, we also envisage a different method to optimize agents according to our CoVO score. While the normalization schemes proposed above should help balance the value and originality parts, finding the perfect reward formulation that correctly addresses both parts is not trivial. In addition, RL tends to have slower convergence. For these reasons, we propose to use DPO (see Section 2.3) as an alternative solution. An intuitive implementation would consider the CoVO score from Equation 5.9 to rank the generated outputs for a given input. However, as already discussed, the CoVO score is made of two distinct parts that might be problematic to optimize at the same time by simply summing them up. Therefore, we propose to build two distinct rankings, one on the value component, and one on the originality component. Then, the two rankings are merged together, and the best-ranked output is the chosen one (i.e., the one whose probability will be maximized), while the worst-ranked is the rejected one (i.e., the one whose probability will be minimized). Once a batch of chosen-rejected pairs has been collected, the model can be optimized via Equation 2.30. Algorithm 3 summarizes the proposed training process.

5.1.2 Experiments

We evaluate the effectiveness of our RL strategy through two case studies: poetry generation (Section 5.1.2) and mathematical problem resolution (Section 5.1.2).

Poetry Generation

Setup. The first set of experiments aims to teach the LLM to generate more original but still valuable poems. In particular, we follow [67] and ask the model to write a poem in a specific style ('ballad', 'haiku', 'hymn', 'limerick'

Algorithm 3 DPO training step toward CoVO Score

```
Require $p_\theta$ reference language model, $p_{\theta'}$ language model to be trained, $\mathcal{B} = \{\mathbf{p_b}\}_{b=1}^{B}$ batch of training prompts, $K$ candidates to generate, compute_ext_reward function to include potential task-specific rewards.
for $b = 1 \dots B$ do
    for $j = 1 \dots K$ do
        $\mathbf{x_j} \sim p_{\theta'}(\mathbf{p_b})$
        $\mathsf{value}_j = s_v(\mathbf{p_b}, \mathbf{x_j}, p_\theta)$
        $\mathsf{orig}_j = s_o(\mathbf{p_b}, \mathbf{x_j}, p_\theta)$
        $\mathsf{ext\_reward}_j = \texttt{compute\_ext\_reward}(\mathbf{x_j})$
    end for
    $\mathsf{new\_scores} = 0 \dots K - 1$
    $\_, \mathsf{indices} = \texttt{sort}(\mathsf{value})$
    $\mathsf{value} = \texttt{sort\_by\_index}(\mathsf{new\_scores}, \mathsf{indices})$
    $\_, \mathsf{indices} = \texttt{sort}(\mathsf{orig})$
    $\mathsf{orig} = \texttt{sort\_by\_index}(\mathsf{new\_scores}, \mathsf{indices})$
    $\mathsf{score} = \mathsf{value} + \mathsf{orig}$
    $\mathsf{score} = \mathsf{score} + \mathsf{ext\_reward}$
    $\mathsf{chosen}_b = \mathbf{x}_{\texttt{argmax}(\mathsf{score})}$
    $\mathsf{rejected}_b = \mathbf{x}_{\texttt{argmin}(\mathsf{score})}$
end for
Train $p_{\theta'}$ via Eq. 2.30 on chosen and rejected.
```

or ‘sonnet’) with a specific tone (‘dark’, ‘happy’, ‘mysterious’, ‘reflective’ or ‘romantic’). We consider Llama3-8B model [169] as our pre-trained agent; we also experiment with smaller models such as SmolLM [37], finding it very hard to make them consistently generate poems and not explanation or garbage outputs. Since we do not use the instruction-tuned models, we prompt them with some few-shot examples of the task to make it more likely to produce the desired output in the desired form (see the full prompts in Appendix A.2). Instead of re-training the entire network, we consider Low-Rank Adaptation (LoRA) [292]. In addition to saving compute resources, LoRA allows us to preserve the information already stored in the model. Indeed, the idea is to learn how to adapt the known information for more creative purposes. The original model is also used to compute the score. We experiment with various settings, i.e., with the original formula and $\lambda_v = \lambda_o = 1.$; by standardizing the two score portions separately at the batch level; by standardizing them using running statistics; and by using the DPO adaptation. Due to additional resource consumption, the DPO adaptation considers a 4-bit quantization of the model [155]. The full training parameters are reported in Appendix A.2. We quantitatively validate our methods by computing poetical metrics for quality (lexical correctness of poems and adherence to line- and syllable-level constraints) and originality (accidental reproduction of existing poems). For the latter, we define a Token-based Longest Common Substring (T-LCS) score, and we use it by comparing generated poems with a reference dataset of approx. 84k public-domain poems extracted from Project Gutenberg (see Appendix B for a first presentation of our GutenVerse dataset). While a generated poem can be an accidental reproduction of a protected work or a different kind of text (e.g., a song), we believe it can provide a useful evaluation tool to understand the general degree of originality. Finally, we also visually inspect poems generated by the models with respect to those generated by the original models for a more fine-grained analysis.

Results. We evaluate the five models (hereinafter Llama3-Baseline for the original pre-trained model, Llama3-CoVO for the fine-tuned model optimizing CoVO score, Llama3-CoVO-std for the model with batch-level standardization, Llama3-CoVO-run for the model with running stats normalization, and Llama3-CoVO-dpo for the model tuned with DPO) by prompting them to generate the 25 possible poems arising by the tone-style pairs used during training and the 25 possible poems from out-of-distribution tone-style pairs (i.e., with a style among 'cinquain', 'couplet', 'free verse', 'ode' or 'tanka' and a tone among 'cutting', 'nostalgic', 'poignant', 'solemn' or 'whimsical'). To be consistent with the training phase, we use the same generation configurations, i.e., 256 max new tokens, a temperature of 1., and top-k = 50.

Method	In-distribution			Out-of-distribution		
Llama3-	Correctness ↑	Metric (L/S) ↑	LCS (avg/max) ↓	Correctness ↑	Metric (L/S) ↑	LCS (avg/max) ↓
Baseline	**1.00**	**0.60** / **0.60**	8.0 / 57	**1.00**	0.33 / 0.30	5.0 / 8
CoVO	**1.00**	**0.60** / **0.60**	9.9 / 49	**1.00**	**0.53** / 0.30	7.2 / 41
CoVO-std	0.76	0.20 / 0.33	5.8 / 19	0.40	0.27 / **0.57**	**4.8** / **6**
CoVO-run	0.80	0.40 / 0.57	**4.9** / **7**	0.60	0.47 / 0.40	**4.8** / 7
CoVO-dpo	0.96	0.50 / 0.50	6.1 / 14	0.92	0.47 / 0.40	7.8 / 40

Table 5.1: Aggregate results of CoVO scores at inference time considering both training prompts (left) and testing prompts (right). Scores on the poetical metrics are reported at the line level (L) and syllable level (S) and only consider requests for styles with specific metrical properties. The best scores are in **bold**, while the worst are in underline.

Table 5.1 reports quantitative metrics about the compliance of poetical constraints at the syllable and line levels, lexical correctness (as the ratio of poems only containing correct words), and accidental reproduction rate (as the mean and maximum token-based longest common substring). As we expected, different training strategies have different effects on the aforementioned metrics. In particular, using the CoVO score as-is leads to behavior close to the baseline for training tone-style pairs, while it increases metric adherence but also verbatim reproduction for out-of-distribution pairs. Llama3-CoVO-dpo has similar but less extreme results, without excelling or failing in any metric. On the other hand, standardizing the CoVO score components places more importance on exploration: both Llama3-CoVO-std and Llama3-CoVO-run tend to produce more "incorrect" outputs but without any relevant accidental reproduction.

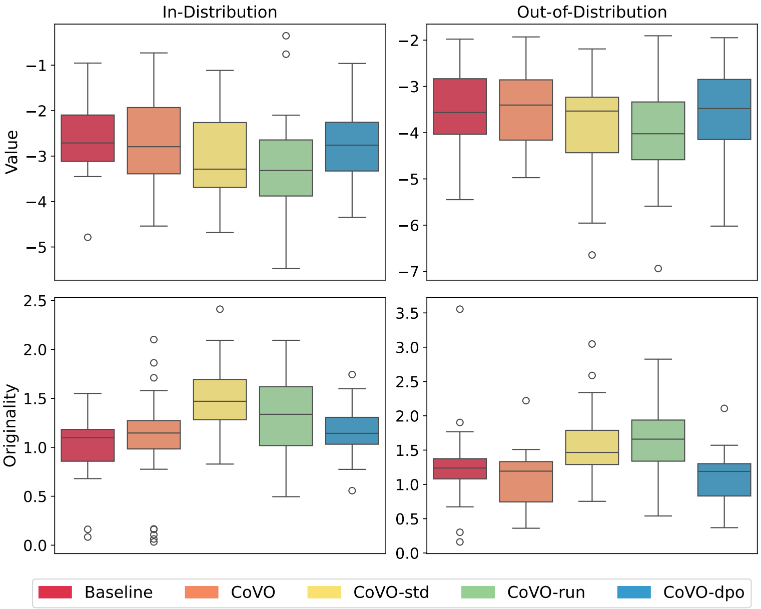


Figure 5.1: The distribution of value and originality (according to our scores) for the in-distribution and out-of-distribution poems generated by the baseline and our four methods.

Then, we calculate the value and originality according to Equation 5.9 under the original pre-trained model. Table 5.1 reports the average scores. Interestingly, these considerations are well-aligned with our CoVO score. Figure 5.1 reports the value and originality according to Equation 5.9 under the original pre-trained model. While the different methods do not substantially differ from the baseline (which is possibly due to the opposite forces of value and originality [671]), we again see that using the CoVO score as-is or with DPO places more focus on the value part, without substantial increases in originality. Instead, standardizing the two separate parts either at the batch or the training level reverses the situation, with better performances on originality but slightly worse on value.

However, aggregated scores such as the ones presented here might be insufficient. A visual and more fine-grained analysis of the generated poems can then help better understand the real performances of our methods.

Due to their shortness, the generated haikus are a reasonable starting point for our evaluation. Table 5.2 reports the 25 haikus produced by our 5 methods across the 5 tones. Notice that while we automatically clean the outputs removing the common random garbage following the poems both at training and inference time, we leave it when it is inside the poem, as it would be impossible to isolate them.

The reported haikus provide an accurate overview of the models. A haiku should be made of three verses of 5-7-5 syllables, and the classic structure wants the first two verses to refer to nature, while the third is an emotional, personal explanation of the two. The baseline is the most accurate in following these rules (even if syllable count is not precise). Llama3-CoVO, which was the method with the highest average value across all poems, follows the three-line structure, sometimes changing the subject; however, it produces the highest-valued haiku which is also the closest one to real haikus. On the other hand, Llama3-CoVO-std and Llama3-CoVO-run do not always fulfill the three-line requirement, and the content is not the classic one as well; moreover, they are prone to meaningless repetitions and the insertion of URL addresses which lead to higher originality, in a way adversarially exploiting the score definition. Finally, Llama3-CoVO-dpo is placed between these two behaviors, as expected: it tends to produce haikus of the correct lengths but it trades off the classic naturalistic content for a more emotional semantic that still leads to high values and some of the best haikus overall.

Llama3-Baseline		Llama3-CoVO		Llama3-CoVO-std		Llama3-CoVO-run		Llama3-CoVO-dpo	
Dark									
Waking up, I do not see the windmills		moonless twilight black pine's sighing twin owls' whistling		[URL]/[URL] — [URL].		I want to sit alone in the dark		The world can get along without me. It is not going to miss me. All in all, I am a nobody.	
V = -2.595	O = 1.076	V = -2.401	O = 1.580	V = -3.395	O = **1.694**	V = **-2.252**	O = 0.495	V = -2.839	O = 1.060
Happy									
A golden sun at midday and an eagle high in the air. The earth is a rainbow.		White dog barks at mailman red. No mail.		I want to go where the sun shines bright. The grass is green and green.[URL] and I want to see. But I am too lazy.		Sunset clouds in high mountains warm, cozy houses.		The sun comes up in the east. It is always up.	
V = **-2.777**	O = 1.123	V = -2.952	O = **1.863**	V = -3.107	O = 1.575	V = -3.309	O = 1.437	V = -3.328	O = 1.006
Mysterious									
A crow of black feathers hovers over the bare top of the lone pine.		Took the sun And drank the rain, And melted into air.		Like the sun, I am shining. Like the ocean, I am vast. Like the clouds, I'm the wind.		It was raining at the beach. [URL] I loved it.[URL][CMD][URL] [URL] I wrote this poem: it was good.		I have lost my mind in the deepness of a star it is not a star but a mind it is all in my mind it is all I have left.	
V = -2.634	O = 1.196	V = -3.060	O = 1.422	V = -3.803	O = 1.320	V = -4.959	O = **1.683**	V = **-2.552**	O = 0.902
Reflective									
A man can stand out in the middle of a field and look over a fence.		The old, the poor, the children too, are all my mother.		A dog in a yard is sleeping soundly to the sound of a fire alarm.		It went away I didn't see it for a while How dare it go.		You have lost an eye. A red scar marks its place. Still you have two good ones.	
V = -2.916	O = 0.858	V = -2.935	O = 1.711	V = -3.460	O = 1.274	V = -2.803	O = 1.471	V = **-2.680**	O = **1.744**
Romantic									
Mingling the sound of footsteps on the stair and the music of the rain.		White blossoms fall— fall upon autumn leaves— who thinks of me?		The moon smiles - I'm the first one to see her. I'm the first one to call her - I love you.		My love is like a red, red rose my love is like a red, red rose my love is like some red, red rose.		A kiss isn't just for fun: a kiss is for the mouth.	
V = -2.221	O = 1.102	V = **-1.932**	O = **2.101**	V = -1.937	O = 1.068	V = -2.583	O = 0.959	V = -2.232	O = 1.533

Table 5.2: All the haikus generated by our four methods and the baseline at inference time, together with V (value) and O (originality) scores computed under the pre-trained model. To avoid any possible data leakage, we replace urls and commands with [URL] and [CMD] to represent that the models have produced that kind of garbage. Bold scores represent the highest ones for a specific tone.

We believe that reporting all the 250 generated poems would be dispersive. However, certain behaviors across other poetic styles are worth mentioning. Llama3-Baseline does not always produce poems: in 3 cases it produces prosaic text (without line breaks). Moreover, as previously seen, it tends to verbatim replicate real text. From a structural standpoint, sonnets are never real sonnets, with more than 14 lines and less than 11 syllables per line, but limericks usually start with the traditional “There [once] was a man”.

Romantic limerick	
There once was a lady called Rose Who lived in the tallest of trees, Where she wrote, Of what she did not know.	
V = -1.282	O = 1.290

Table 5.3: The poem generated by Llama3-CoVO with the highest value (apart from verbatim reproduction of existing poems).

As expected by the average scores, Llama3-CoVO does not consistently deviate from such behaviors. While always producing seemingly poetry, it tends to verbatim reproduce other works, not only poems but also *Song of Salomon* 6:2-9 when asked for a mysterious hymn and even pieces of the *Declaration of Independence* for a reflective hymn and the Christmas song *Joy to the world* for a happy hymn. Interestingly, these text have a very small originality score (an average of 0.2), suggesting that the original model has memorized them; we will extensively discuss this matter in Section 7.3.3. From a structural perspective, while ballads vary more in length and content with respect to our baseline, the limericks are still all appropriate, as we can see in the example from Table 5.3, which has the highest value across all Llama3-CoVO poems.

Despite the higher average originality, also Llama3-CoVO-std can verbatim produce existing texts, as done with the *Roses Are Red* nursery rhyme when asked for a romantic ballad. In general, its heavier deviation from the original pre-trained model translates into more diverse and usually shorter structures, but also in the insertion of random URL addresses in the middle of poems. However, it still produces some more classic poems that typically lead to higher values but also to high originality (see Table 5.4).

The issues related to divergence highlighted for Llama3-CoVO-std are even more pronounced in Llama3-CoVO-run. While it does not generate any verbatim reproduction, it inserts several URL addresses, code snippets,

Happy hymn		Reflective limerick	
Joyful, joyful, we adore thee, Son of God who callest us Praise to his almighty name. Joyful, joyful we adore thee; Full of grace are all his ways; Praise to his almighty reign Ever, ever, praise we sing it, Ever in our songs and hymns. Glory to our God Almighty.		There was a young man from HongKong who wanted to see more green trees every day. So he sat on the beach, listened to the sea waves, and saw the green sea.	
$V =$ **-1.114**	$O =$ 1.470	$V =$ -1.995	$O =$ **2.411**

Table 5.4: The poems generated by Llama3-CoVO-std with the highest value (left) and the highest originality (right).

Reflective limerick		Happy ballad	
I'm a poet, man. My poem is hard and great. It's a poem, man, for all my poem to eat. So I'm a poet man, man, man.		There once was a girl named Kate, She grew very tired very quickly. So they told poor Kate, “You need to take it easy,”_REF)they said. They thought she was lazy. And her name was Kate.	
$V =$ **-0.352**	$O =$ 1.559	$V =$ -3.878	$O =$ **2.094**

Table 5.5: The poems generated by Llama3-CoVO-run with the highest value (left) and the highest originality (right).

and odd tokens in the middle of poems, as apparent from Table 5.5 (right): the fact that these are the text with higher originality supports the idea of adversarial exploitation of the score. In addition, the poems do not have any traces of classic structures or common starting lines and tend to repeat the same n-grams multiple times, as shown by the highest-value poem from Table 5.5 (left).

Finally, Llama3-CoVO-dpo seems the best method to balance the quality of poems with some original traits. Even though it sometimes reproduces small parts of real poems, it always deviates after one or two lines. In general, all poems usually follow the mood and style required; in particular, ballads sound like true ballads, as shown by Table 5.6 (left). Similarly, also limericks are remarkable: notably, while the poem with the highest originality starts with the classic “There once was a” formulation (Table 5.6 - right), other limericks with divergent first lines get lower originality, perhaps due to the general mood which is very close to classic limericks (see Table 5.7).

Romantic ballad		Romantic limerick	
Hark! Is it the wild cat mewling mournfully by night or is it the poor maiden weeping for her love’s flight? It is the latter— as one by one, her friends take to heart, in turns, the girl’s anguished cries. Tall, his figure, he is seen one moment by the gladsome crowd then not again, till with tears, the maid calls forth from the crowd— “See! he is returned, as if to marry her.” The girl looks at him, as the maid and crowd watch the two lovers: her lover’s gaze at her with tender, true love; hers with a silent plea, and a sigh.		There once was a snob who could scotch Anything not in accord with What she thought, And if people asked her out on a lunch The same thought she had in her head, For her, being scotch Wasn’t what you’d get in her bed - Just a glass of good scotch.	
V = **-0.961**	O = 1.294	V = -1.869	O = **1.598**

Table 5.6: The poems generated by Llama3-CoVO-dpo with the highest value (left) and the highest originality (right).

Happy limerick		Dark limerick	
“Is life worth living?” said I. “Perhaps,” murmured he. “Well, anyway,” said I, “If it were not for Cigars, I’d say, ’Nay’!”		Once a mouse went up To a mousetrap by the river. He sat down to wait, He sat. That’s all. All his friends were never as clever.	
V = -2.258	O = 0.944	V = -1.932	O = 1.310

Table 5.7: Two limericks generated by Llama3-CoVO-dpo which obtain a lower originality despite having a non-traditional structure.

Math Problem Resolution

Setup. The second set of experiments aims to teach the LLM to solve mathematical problems through more diverse procedures. In particular, we focus on the Mistral-based [317] MetaMath model, i.e., fine-tuned with self-supervised learning on the MetaMathQA [722]. It is a dataset of textual math questions paired with responses where the numerical answer is easily separable from the textual procedure. While the entire set contains 395k entries, making an additional training epoch too expensive, MetaMathQA is composed of entries from two different training sets, then augmented with various techniques: GSM8K [127] and MATH [269]. Since we are only interested in the questions, we limit our training to those datasets; moreover, we exclude all questions with a tokenized length of either question or answer greater than 512, obtaining 14876 out of 14973 total entries. To train our model, we separate the procedure and the answer from each solution. We separate the procedure and the answer from each solution to train our model. We use the numerical answer to check the correctness of the predicted solution, while we use the textual procedure only at evaluation time to measure the diversity of the model output. Because of this, the RL problem can consider up to two rewards: our score computed on the procedure and an extrinsic reward based on the correctness of the answer. As for the previous case study, instead of fine-tuning the entire model, we adopt a more parameter-efficient strategy with LoRA, while using the original model to perform the CoVO score computation. Again, we experiment with four different configurations: PPO and the score from Eq. 5.9 with $\lambda_v = 1, \lambda_o = 1$; PPO and the score normalized at the batch level; PPO and the score normalized at training level; DPO (Alg. 3). We consider scenarios with and without the external reward, and we compare the performances with the original model and a fine-tuning based only on the external reward. The full training parameters are reported in Appendix A.2. The evaluation considers the test sets of both GSM8K and MATH datasets (again limited to the entries with a tokenized length of question and answer smaller than 512, leaving all 1319 entries for GSM8K and 4546 out of 5000 for MATH), and computes the percentage of correct solutions together with two diversity metrics: expectation-adjusted distinct N-grams (EAD) [403] and sentence embedding cosine similarity (SIM) [287], which should measure syntactical and semantical diversity, respectively [346]. EAD counts the number of distinct N-grams (averaging over $N = 1 \dots 5$) across all generated responses and removes the bias toward shorter inputs by scaling the number of distinct tokens based on their expectations. The SIM metric computes the average of the cosine similarity between the embeddings of any possible pairs of outputs and returns 1 minus the similarity. While originally based on Sentence-BERT [528], we employ the more recent all-mpnet-base-v2, as suggested by their developers [580].

Results. Table 5.8 reports the results for the GSM8K and MATH test sets. For the former, while all the tested methods achieve similar results, we see that using the extrinsic reward leads to better results and even higher Sent-BERT scores; its combination with the "plain" CoVO reward makes the

Method	GSM8K			MATH		
Metamath-mistral-	Accuracy ↑	EAD ↑	Sent-BERT ↑	Accuracy ↑	EAD ↑	Sent-BERT ↑
Baseline	77.79% (3)	1.945	0.751	34.37% (483)	5.652	0.662
Ext. reward	78.32% (0)	1.933	0.747	34.23% (411)	5.593	0.667
CoVO	**78.85%** (1)	1.932	0.754	34.40% (424)	5.571	0.671
CoVO w ext	78.62% (1)	1.935	0.749	34.40% (397)	5.574	0.668
CoVO-std	78.77% (2)	1.930	**0.755**	34.34% (433)	5.603	0.668
CoVO-std w ext	78.47% (0)	1.923	0.754	**34.74%** (409)	5.573	0.661
CoVO-run	78.24% (3)	1.931	0.754	34.40% (413)	5.599	**0.673**
CoVO-run w ext	78.17% (0)	1.926	0.753	**34.74%** (388)	5.581	0.662
CoVO-dpo	77.56% (5)	1.955	0.749	34.62% (474)	**5.680**	0.663
CoVO-dpo w ext	77.33% (5)	**1.957**	0.750	34.71% (493)	5.674	0.664

Table 5.8: Accuracy and diversity of results for the GSM8K and MATH test sets. In brackets, the amount of responses not finished within the fixed number of maximum tokens to generate. The best scores are in **bold**, while the worst are in underline.

model gain the highest percentage of correct answers. On the other hand, the DPO strategy even diminishes the accuracy of the original model, but obtains the highest EAD scores.

The results for the MATH test set are significantly different. The use of the extrinsic reward negatively affects the accuracy, but slightly increases the diversity; again, DPO helps obtain the best EAD scores, while the two CoVO normalization strategies lead to better accuracy but higher Sent-BERT diversity (without extrinsic reward). However, the number of unfinished responses is quite high (approx. 1 out of 10), and results might significantly vary if more tokens are allowed during generation.

5.2 Contextual Learning via Creativity Score

Large language models have shown strong performances in contextual learning, i.e., zero- or few-shot learning: their behavior can be effectively influenced by simply setting the right prompt and providing the right examples. In this section, we explore the potential of leveraging this property to enhance creative outputs, as measured by the CoVO score presented in Section 5.1.

5.2.1 Approach

Given the tendency of LLMs to generate solutions in terms of ordered lists, we propose to use our score from Equation 5.9 to compose a ranking-based prompt, then used with in-context learning to make the LLM generate a supposedly more creative output. Specifically, we generate K outputs (with K small) given the current input with the model as-is; one solution could

Algorithm 4 CoVO-based ranking computation for Contextual Learning

```
Require $p_\theta$ pre-trained language model, $K$ ranking size, $K$ gen_conf generation configurations for ranked outputs, $\mathbf{p}$ user input.
for $j = 1 \dots K$ do
    Generate $\mathbf{x_j}$ from $p_\theta(\mathbf{p})$ according to $\mathsf{gen\_conf}_j$
    $\mathsf{value}_j = s_v(\mathbf{p}, \mathbf{x_j}, p_\theta)$
    $\mathsf{orig}_j = s_o(\mathbf{p}, \mathbf{x_j}, p_\theta)$
end for
$\mathsf{new\_scores} = 0 \dots K-1$
$\_, \mathsf{indices} = \texttt{sort}(\mathsf{value})$
$\mathsf{value} = \texttt{sort\_by\_index}(\mathsf{new\_scores}, \mathsf{indices})$
$\_, \mathsf{indices} = \texttt{sort}(\mathsf{orig})$
$\mathsf{orig} = \texttt{sort\_by\_index}(\mathsf{new\_scores}, \mathsf{indices})$
$\mathsf{score} = \mathsf{value} + \mathsf{orig}$
$\_, \mathsf{indices} = \texttt{sort}(\mathsf{score})$
$[\mathbf{x_1} \dots \mathbf{x_K}] = \texttt{sort\_by\_index}([\mathbf{x_1} \dots \mathbf{x_K}], \mathsf{indices})$
Return $\mathbf{x_1} \dots \mathbf{x_K}$
```

consist in varying the temperature parameter to obtain significantly different outputs, but other approaches can be used as well. Then, we compute their CoVO score similarly to what we did in the DPO strategy from Section 5.1: we consider the value and originality parts separately; we compute a rank of solutions for both scores; we merge the two rankings together. Algorithm 4 summarizes the process. Then, we build the ranking-based prompt by assigning the $K+1$-th position to the worst output, the K-th position to the second worst output, and so on until we assign the second position to the highest-ranked output; then we let the model autoregressively complete the ranking by writing the first-placed solution, which we consider as the output of our method. The resulting prompt with $K = 4$ is the following:

```
{task}

Top-5 solutions:
5. {$\mathbf{x_1}$}
4. {$\mathbf{x_2}$}
3. {$\mathbf{x_3}$}
2. {$\mathbf{x_4}$}
1.
```

Notably, this prompt-based method is fully compatible with any autoregressive model, and can be combined with other generation techniques, such as contrastive search [631] or diverse beam search [675].

5.2.2 Experiments

Setup. We evaluate the effectiveness of our contextual-learning strategy through 15 BIG-bench [622] tasks, specifically those labeled with "creativity". Eight of them, i.e., Codenames, Conlang Translation, Cryptonite, Gem, Question-Answer Creation, Rephrase, Taboo, and Yes-No Black-White are free-text generation tasks; the remaining ones, i.e., English Proverbs, Forecasting Subquestions, Kannada, Novel Concepts, Riddle Sense, Swedish to German Proverbs, and Understanding Fables are multiple-choice questions. We limit the number of considered queries to 200 to make the computation practicable, and we repeat all the experiments across 3 seeds. To demonstrate the validity of our method, we consider different models of different sizes: SmolLM-1.7B [37]; Phi3 [1]; Mistral-7B [317]; Llama3-8B [436]. To have them follow the task instructions, we consider their instructed version. Moreover, to reduce resource consumption, we quantize them to 4-bit. As our baselines, we use the same models as-is; for the free-generation tasks, we also experiment with different values of the temperature. Higher values of the temperature are generally used to increase output creativity at least in terms of certain dimensions [497]. A full description of the generation parameters and the prompts used in the experiments are reported in Appendix A.3.

Results. Table 5.9 presents the results of the seven multiple-choice tasks. Our method generally shows competitive performance, and the introduction of the contextual learning strategy leads to improvements in some of the cases. Additionally, the results are consistent across all tasks.

Instead, Table 5.10 reports the results on the 8 free-generation tasks; as it is apparent, results are always zero on two tasks (Cryptonite and Question-Answer Creation), thus only the other 6 are relevant for our discussion. Interestingly, our strategy seems effective on larger models: it gets the highest scores in 4 out of 6 tasks for Mistral-7B and only in one case is worse than standard sampling; in Llama3-8B, it is the best strategy in 3 cases and the worst only once. On the other hand, for smaller models, the results are more balanced, and consistently improved only for the Rephrase task.

Overall, our strategy leads to improved performance, especially for generative tasks. While there are costs associated with obtaining candidates and composing the contextual prompt, the benefits in performance can make this trade-off worthwhile.

Model	English Proverbs	Forecasting Subquestions	Kannada	Novel Concepts
SmolLM-1.7B	0.39 ± 0.04	**−43.54 ± 0.00**	0.24 ± 0.01	0.25 ± 0.00
w/ CoVo	0.34 ± 0.07	−45.43 ± 0.14	0.26 ± 0.01	0.27 ± 0.03
Phi3-4B	0.36 ± 0.01	**43.65 ± 0.00**	0.29 ± 0.00	**0.41 ± 0.00**
w/ CoVo	**0.51 ± 0.03**	−44.41 ± 0.08	0.27 ± 0.04	0.25 ± 0.03
Mistral-7B	**0.75 ± 0.01**	−75.89 ± 0.00	0.24 ± 0.01	**0.44 ± 0.00**
w/ CoVo	0.51 ± 0.07	**−74.65 ± 0.19**	**0.27 ± 0.01**	0.33 ± 0.03
Llama3-8B	0.56 ± 0.01	−45.75 ± 0.00	0.28 ± 0.01	0.34 ± 0.00
w/ CoVo	0.55 ± 0.04	**−44.21 ± 0.03**	0.26 ± 0.02	0.35 ± 0.06

	Riddle Sense	Swedish to German Proverbs	Understanding Fables	
SmolLM-1.7B	**0.26 ± 0.01**	0.28 ± 0.06	0.15 ± 0.01	
w/ CoVo	0.22 ± 0.02	0.28 ± 0.01	**0.19 ± 0.01**	
Phi3-4B	0.22 ± 0.01	0.26 ± 0.02	0.27 ± 0.00	
w/ CoVo	**0.40 ± 0.03**	**0.34 ± 0.04**	**0.32 ± 0.01**	
Mistral-7B	0.44 ± 0.03	0.52 ± 0.03	**0.62 ± 0.02**	
w/ CoVo	0.39 ± 0.07	0.44 ± 0.05	0.38 ± 0.04	
Llama3-8B	0.50 ± 0.03	**0.49 ± 0.02**	0.36 ± 0.02	
w/ CoVo	0.48 ± 0.03	0.32 ± 0.07	0.36 ± 0.02	

Table 5.9: Aggregate results across 3 seeds from our qualitative assessment for the 7 multichoice tasks for the original model as-is and the version with our contextual-learning approach. The various tasks are evaluated considering the preferred score reported in BIG-bench.

Model	Codenames	Conlang Translation	Cryptonite	Gem
SmolLM-1.7B				
temp=1.0	0.22 ± 0.02	**21.55 ± 7.38**	0.00	52.26 ± 4.63
temp=0.8	0.11 ± 0.03	**28.61 ± 8.21**	0.00	56.36 ± 8.49
temp=1.2	**0.29 ± 0.02**	14.19 ± 6.85	0.00	38.15 ± 7.47
w/ CoVo	0.14 ± 0.1	13.05 ± 6.68	0.00	35.63 ± 18.2
Phi3-4B				
temp=1.0	**0.57 ± 0.54**	**7.08 ± 3.19**	0.00	45.72 ± 6.48
temp=0.8	0.00 ± 0.00	2.96 ± 4.19	0.00	50.47 ± 7.36
temp=1.2	**0.66 ± 0.27**	**7.61 ± 2.73**	0.00	44.53 ± 6.75
w/ CoVo	**0.67 ± 0.5**	**13.02 ± 9.33**	0.00	23.62 ± 23.2
Mistral-7B				
temp=1.0	0.00 ± 0.00	8.86 ± 2.45	0.00	55.15 ± 2.10
temp=0.8	0.00 ± 0.00	1.82 ± 2.57	0.00	**57.55 ± 1.76**
temp=1.2	1.01 ± 1.43	14.07 ± 3.78	0.00	**53.08 ± 5.93**
w/ CoVo	**16.71 ± 6.53**	**63.11 ± 7.37**	0.00	**57.94 ± 0.67**
Llama3-8B				
temp=1.0	**0.97 ± 0.40**	0.00 ± 0.00	0.00	62.73 ± 1.25
temp=0.8	0.25 ± 0.14	1.27 ± 0.90	0.00	64.58 ± 3.02
temp=1.2	0.72 ± 0.13	0.18 ± 0.25	0.00	66.62 ± 2.06
w/ CoVo	**1.14 ± 0.18**	**36.07 ± 5.22**	0.00	60.66 ± 4.86

	Question-Answer Creation	Rephrase	Taboo	Yes No Black White
SmolLM-1.7B				
temp=1.0	0.00	17.02	**−0.42 ± 0.09**	−0.21 ± 0.01
temp=0.8	0.00	17.02	**−0.45 ± 0.13**	−0.32 ± 0.06
temp=1.2	0.00	17.02	−0.47 ± 0.04	**−0.16 ± 0.05**
w/ CoVo	0.00	−15.74 ± 1.50	**−0.33 ± 0.07**	**−0.18 ± 0.04**
Phi3-4B				
temp=1.0	0.00	−14.71	−0.06 ± 0.03	**−0.13 ± 0.02**
temp=0.8	0.00	−14.71	**−0.02 ± 0.01**	−0.20 ± 0.03
temp=1.2	0.00	−14.71	−0.08 ± 0.01	**−0.11 ± 0.03**
w/ CoVo	0.00	**−11.84 ± 0.15**	−0.12 ± 0.03	**−0.11 ± 0.03**
Mistral-7B				
temp=1.0	0.00	−47.48	−0.12 ± 0.04	−0.20 ± 0.04
temp=0.8	0.00	−47.48	**−0.00 ± 0.00**	−0.15 ± 0.10
temp=1.2	0.00	−47.48	−0.16 ± 0.05	−0.24 ± 0.13
w/ CoVo	0.00	**−43.91 ± 0.28**	−0.63 ± 0.11	−0.21 ± 0.10
Llama3-8B				
temp=1.0	0.00	−16.34	−1.59 ± 0.08	**−0.22 ± 0.05**
temp=0.8	0.00	−16.34	−2.12 ± 0.28	**−0.20 ± 0.14**
temp=1.2	0.00	−16.34	**−1.14 ± 0.08**	**−0.20 ± 0.04**
w/ CoVo	0.00	**−12.22 ± 0.05**	−1.43 ± 0.12	−0.46 ± 0.10

Table 5.10: Aggregate results across 3 seeds from our qualitative assessment for the 8 generative tasks for the original model as-is (sampling at 3 different temperatures) and the version with our contextual-learning approach (sampling with 1.0 temperature). The various tasks are evaluated considering the preferred score reported in BIG-bench.

6 Reward beyond Reinforce

While the method proposed in Chapter 5 mainly focuses on acquiring creativity-relevant skills along with domain-relevant skills during the preparation step, other phases of the creative process can be considered.

Indeed, another possibility to introduce a human-inspired process in generative AI is by focusing on the sampling scheme, i.e., how the output is generated. To recap from Section 1.2, creativity should involve the following steps: task presentation (from internal or external stimuli); preparation; response generation (thanks to creativity-relevant skills); and response validation (thanks to domain-relevant skills) [10]. In this chapter, we focus on the last two steps, and we try to define new sampling schemes that include creativity-relevant skills (Section 6.1) and that possibly validate responses before returning them (Section 6.2).

6.1 DiffSampling

In autoregressive models like LLMs, the decoding strategy controls the response generation. The standard decoding schemes of language models follow the learned probability distribution from the training data. While this should guarantee the highest probable tokens to be the most appropriate for the current input, it can also foster the reproduction of training data and flatten the lexicon in favor of the most common grammatical structures and words. The temperature parameter may increase the likelihood of less-frequent tokens, but it also raises the chance of syntactically incorrect tokens by flattening their probabilities, regardless of their actual positions.

An ideal solution should focus on where the *critical mass* resides. Nucleus sampling [285] tries to remove the tail of the probability distribution by focusing on the smallest subset of tokens with a global probability exceeding a given threshold. However, a few issues still remain. First, nucleus sampling is sensitive to the choice of the threshold. Second, certain peculiar situations might be solved incorrectly. For example, if there is one token with a very

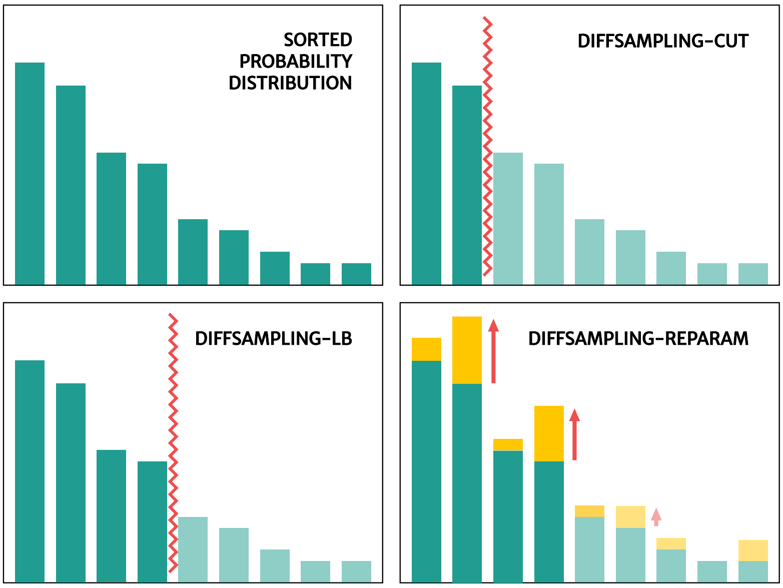


Figure 6.1: The effect of our *DiffSampling* methods. In the top-left square, the original (sorted) distribution. In the top-right square, *DiffSampling-cut* truncates after the minimum discrete derivative. In the bottom-left square, *DiffSampling-lb* also imposes a total probability lower bound (in this example, equal to 0.5). In the bottom-right square, *DiffSampling-reparam* also reparameterizes the probabilities with their discrete derivative.

high probability (but smaller than the threshold), it is likely to be the only appropriate one; yet, nucleus sampling will still preserve some other tokens, not avoiding potential errors in generation. Another case is when we have several equally probable tokens whose total probability exceeds the threshold: nucleus sampling will exclude some correct tokens, reducing the chance of diversifying the final result.

In this section, we propose *DiffSampling*. This family of sampling strategies leverages the derivative of the probability distribution to focus on the critical mass in a way that only depends on the probabilities themselves. We envisage three different alternatives to do so (see Figure 6.1), and we discuss their advantages and limitations. Finally, we evaluate them in three different scenarios, demonstrating that our method consistently performs at least as well as current strategies.

6.1.1 Approach

Given the probability distribution of the next token, let us imagine sorting it to have tokens in descending order based on their probability. The critical mass can be seen as the one delimited by the biggest difference between the probabilities: the token to its left should be the least probable token that our model still considers correct, i.e., the one that we might want to generate to produce an output that is both appropriate and diverse.

In mathematical analysis, this point has a simple and elegant characterization: it is where the derivative is minimum. Let us consider a probability distribution $p(x)$ defined for a limited number of $x_1 \dots x_N$, with $p()$ monotonically decreasing. According to the forward difference approximation, the discrete derivative of a function $f(x_n)$ is defined as $\Delta f(x_n) = f(x_{n+1}) - f(x_n)$, thus we have:

$$\Delta p(x_n) = \begin{cases} p(x_{n+1}) - p(x_n) & \text{if } n < N \\ -p(x_n) & \text{if } n = N \end{cases} \tag{6.1}$$

which is always non-positive. The $\texttt{argmin}(\Delta p(x_n))$ represents the index of the last token before the point characterized by the largest difference between probabilities.

Starting from $\Delta p(x_n)$, we propose *DiffSampling*, a family of different decoding strategies. The first one, which we call *DiffSampling-cut*, consists of cutting the tail at the right side of the point characterized by the largest difference between the probabilities, i.e., sampling among the tokens $x_i, i \leq \texttt{argmin}(\Delta p(x_n))$. This approach can be seen as an improved greedy strategy: when the model has high confidence in a single token, it degenerates into the greedy strategy; otherwise, it preserves other appropriate tokens, increasing the diversity of the final result.

However, there might be situations in which some of the excluded tokens are still correct; for example, the first token might minimize $\Delta p(x_n)$ but still have a quite low probability, i.e., it does not *really* cover the entire critical mass. To address this issue, *DiffSampling-lb* introduces a lower bound on the mass probability. To leverage the advantage of our cutting strategy while maintaining that of nucleus sampling, our second strategy considers truncating based on $\Delta p(x_n)$ in such a way that the resulting tokens have a total probability at least equal to the lower bound p_{lb}. In other words, given k cardinality of the smallest subset of tokens whose total probability is not lower than p_{lb}, it computes the $\texttt{argmin}(\Delta p(x_n))$ for $n \geq k$. This approach can be seen as an improved nucleus sampling: it *corrects* the p parameter of nucleus sampling via our derivative-based approach to include correct tokens after the selected nucleus. While this reintroduces a sensible parameter to set, we found that a value of 0.8 provides almost no loss in terms of accuracy but higher diversity compared to lower values (see Appendix C.1).

Finally, our third strategy aims to transform the probability distribution to avoid that the most frequent tokens in the training set are also the most probable according to the model, increasing diversity and reducing the risk of accidental reproduction. *DiffSampling-reparam* shifts the model's probabilities toward the options with the smallest derivatives by adding to the original probabilities the negative of its derivatives, scaled by a multiplier γ:

$$p'(x_n) = p(x_n) - \gamma \Delta p(x_n)\,. \qquad (6.2)$$

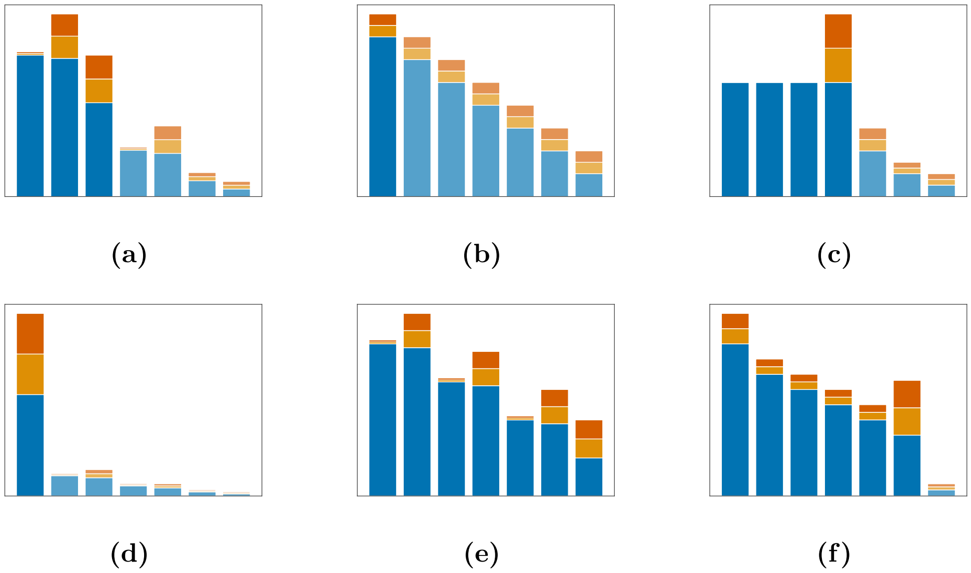
This, combined with the cutting and lower-bound strategies, enhances novelty while maintaining the appropriateness of responses. This approach can be seen as an alternative temperature: while a higher temperature still preserves the most probable tokens as such (and introducing nucleus sampling has the limitations mentioned above), our strategy increases the probability of tokens before a "jump", which are less likely to be sampled. However, the γ parameter has a different behavior than temperature: with $\gamma = 0$ we fall back to *DiffSampling-lb*; with a very large γ, we obtain a deterministic sampling scheme that greedily chooses the token with the minimum derivative; with a small enough value, e.g., $\gamma = 1$, we promote that last token while still preserving the original distribution.

Overall, *DiffSampling* can be seen as a sampling scheme governed by two parameters, i.e., the probability-mass lower bound p_{lb} and the reparameterization factor γ. The full algorithm is reported in Algorithm 5.

Algorithm 5 DiffSampling

```
**Input:** probabilities probs = $[p_1 \dots p_N]$, lower bound $p_{lb}$, multiplier $\gamma$.
sorted_probs, indices = sort(probs)
fwd_probs = sorted_probs[1:] + [0.]
delta_probs = fwd_probs − sorted_probs
nucleus = cumsum(sorted_probs) < $p_{lb}$
sorted_probs = sorted_probs − $\gamma$ · delta_probs
delta_probs[nucleus] = 0.
$d$ = argmin(delta_probs)
sorted_probs[$d$+1:] = 0.
probs = sort_by_idx(sorted_probs, indices)
probs = probs/sum(probs)
**Output:** probs
```

Figure 6.2 reports six examples of the effects of *DiffSampling-cut* and of *DiffSampling-reparam* with $\gamma = 0.5$ and $\gamma = 1.$, first with a seemingly random distribution (the usual situation) and then with five peculiar scenarios.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 6.2: The effect of our derivative-based cutting and reparameterization over different probabilities distribution in the case of 7 tokens. In blue is the original probability distribution; transparency represents the tokens cut by *DiffSampling-cut*. In yellow is the effect of *DiffSampling-reparam* with $\gamma = 0.5$ parameter; in red the effect with $\gamma = 1$.

6.1.2 Experiments

To evaluate whether *DiffSampling* helps diversify outputs while maintaining a high level of accuracy, we test it on three case studies: mathematical problem resolution, text summarization, and divergent association task.

Models and Baselines. In all our experiments, we start from a state-of-the-art LLM and test various decoding strategies. For the math problem resolution, we use the Llama2-based MetaMath model trained with self-supervised learning on MetaMathQA [722]. Following [110], for extreme text summarization we use the Llama2-7B model [656], considering both RLHF-instructed and pre-trained versions. Finally, for the divergent association task, we consider Llama3-8B [169], using both DPO-tuned and pre-trained versions. We study the performances of our three methods: *DiffSampling-cut*; *DiffSampling-lb* with $p_{lb} = 0.8$; *DiffSampling-reparam* with $p_{lb} = 1., \gamma = 1.$. We compare them with a total of 7 baselines: greedy strategy and contrastive search (with $k = 8$ and $\alpha = 0.6$); nucleus sampling (with $p = 0.9$), η-sampling (with $\eta = 0.0003$), and locally typical sampling (with $p = 0.9$); nucleus sampling with a higher temperature of 1.5 and 2.0. We also experiment with different p_{lb} and γ values; results are shown and discussed in Appendix C.

Math Problem Resolution

Setup. Solving math problems provides a useful case study for our decoding strategies, as it allows us to evaluate the correctness of solutions (as the percentage of correctly solved problems) and the diversity of procedures to arrive at the result. To better understand whether our methods can increase diversity while maintaining accuracy we consider both the MetaMathQA training set [722] and the GSM8K [127] and MATH [269] test sets; the relative prompts are reported in Appendix A.4. To avoid resource wasting, we focus on entries with a problem and a solution whose tokenized versions are no longer than 512. Since the training set is incredibly vast (395k entries), we limit our experiment to 1000 random samples, while we consider all 1319 entries from the GSM8K test set and all the filtered 4545 entries from the MATH test set. We evaluate the quality of solutions through the ratio of correctly solved problems. Instead, the diversity is computed according to various methods: distinct 1-grams and 2-grams [385], plus expectation-adjusted distinct N-grams (EAD) [403] and embedding cosine similarity (SIM) [287], which should evaluate both syntactic and semantic diversity, respectively [346]. EAD counts the number of distinct N-grams tokens (averaging over $N = 1 \dots 5$) and removes the bias toward shorter inputs by scaling the number of distinct tokens based on their expectations. The SIM metric computes the cosine similarity between the embeddings of the sentences and returns 1 minus the similarity. While originally based on Sentence-BERT [528], we employ the more recent all-mpnet-base-v2, as suggested by their developers [580]. Following [346], we compute *cross-input* EAD and SIM, i.e., by considering all outputs produced for a specific seed together. In addition, we also compute *against-greedy* EAD and SIM. Given each input, we compare the output with the greedy one by calculating the average expectation-adjusted distinct N-grams not present in the greedy response, and 1 minus the cosine similarity between the two outputs, respectively.

Results. Table 6.1 reports the aggregate results of all the tested methods over the sampled portion of the math training data. As evident by the results, the MetaMath model's performance heavily depends on whether the generated tokens are sampled or selected greedily: in the first case, the percentage of correct answers can drop by more than half. Quite interestingly, our cutting strategy achieves comparable results as well even without greedy sampling. Most likely, the tokens corresponding to the final answer are the

Method	Accuracy	Cross-Input Diversity		Against-Greedy Diversity	
		EAD ↑	SIM ↑	EAD ↑	SIM ↑
Greedy	95.27 ± 0.17	1.65 ± 0.01	0.74 ± 0.02	-	-
Contrastive search	94.17 ± 0.45	1.66 ± 0.01	0.74 ± 0.01	0.15 ± 0.01	0.27 ± 0.01
η-sampling	89.17 ± 0.52	1.71 ± 0.01	0.74 ± 0.01	0.22 ± 0.01	0.37 ± 0.01
Locally typical sampling	91.70 ± 0.62	1.69 ± 0.01	0.74 ± 0.01	0.20 ± 0.01	0.35 ± 0.01
Nucleus sampling t=1.0	91.70 ± 0.62	1.69 ± 0.01	0.74 ± 0.01	0.20 ± 0.01	0.35 ± 0.01
Nucleus sampling t=1.5	87.63 ± 0.90	1.73 ± 0.00	0.73 ± 0.02	0.26 ± 0.01	0.41 ± 0.01
Nucleus sampling t=2.0	30.17 ± 0.76	8.20 ± 0.12	0.59 ± 0.02	0.71 ± 0.01	0.63 ± 0.01
DiffSampling-cut	94.70 ± 0.21	1.66 ± 0.01	0.75 ± 0.01	0.12 ± 0.00	0.22 ± 0.01
DiffSampling-lb	92.97 ± 0.12	1.67 ± 0.01	0.74 ± 0.01	0.18 ± 0.01	0.32 ± 0.01
DiffSampling-reparam	89.67 ± 0.15	1.70 ± 0.01	0.75 ± 0.01	0.23 ± 0.01	0.38 ± 0.01

Table 6.1: Accuracy and diversity of results for the training data set over 3 seeds. Accuracy and cross-input diversity report the mean and standard error over the final score of each run, while against-greedy diversity reports the mean and the 95% confidence interval over the full set of answers.

most probable ones and are also characterized by the largest difference in terms of probabilities, making the sampling *almost* greedy. This effectively shows one of the advantages of our solution over nucleus sampling, which instead carries on less probable and wrong tokens that can be sampled afterward. Similarly, *DiffSampling-reparam* with $\gamma = 10.$ obtains slightly better results than smaller γ and the standard sampling: most likely, the reparameterization pushes sufficiently up the tokens before the point characterized by the largest difference between the probabilities, which incidentally are those leading to correct solutions. On the other hand, there is no real difference in diversity across the tested methods, which might be because training data are considered: the learned probability distribution is close to the real one, thus regardless of the sampling strategy, the generated text will be anyway similar. Moreover, the all-mpnet-base-v2 model might not be ideal for evaluating diversity in mathematical procedures, making its related metrics pointless.

Table 6.1 reports the aggregate results of all the tested methods over the sampled portion of the math training data. As evident by the results, the MetaMath model's performance depends on the greediness of the decoding strategy: the greedy one achieves the highest accuracy, closely followed by *DiffSampling-cut*. Interestingly, both *DiffSampling-lb* and *DiffSampling-reparam* perform better than their most similar baselines while having similar cross-input diversity. On the other hand, the against-greedy diversity scores are inversely correlated with accuracy, since the greedy strategy appears to be the optimal one.

Table 6.2 reports the results for the GSM8K test set. Differently from the results above, here the greedy strategy does not provide strong advantages, and our cutting strategy achieves the highest percentage of solved problems.

Method	Accuracy	Cross-Input Diversity		Against-Greedy Diversity	
		EAD ↑	SIM ↑	EAD ↑	SIM ↑
Greedy	66.44 ± 0.00	1.97 ± 0.00	0.74 ± 0.00	-	-
Contrastive search	65.88 ± 0.59	2.00 ± 0.00	0.74 ± 0.00	0.18 ± 0.00	0.38 ± 0.01
η-sampling	65.05 ± 0.19	2.06 ± 0.00	0.75 ± 0.00	0.27 ± 0.00	0.49 ± 0.01
Locally typical sampling	66.29 ± 0.55	2.03 ± 0.00	0.75 ± 0.01	0.24 ± 0.00	0.46 ± 0.01
Nucleus sampling t=1.0	65.00 ± 0.18	2.02 ± 0.01	0.75 ± 0.00	0.24 ± 0.00	0.46 ± 0.01
Nucleus sampling t=1.5	63.91 ± 0.57	2.11 ± 0.01	0.76 ± 0.01	0.30 ± 0.00	0.53 ± 0.01
Nucleus sampling t=2.0	25.40 ± 0.07	9.97 ± 0.10	0.65 ± 0.00	0.71 ± 0.01	0.71 ± 0.01
DiffSampling-cut	67.10 ± 0.19	1.98 ± 0.00	0.75 ± 0.00	0.15 ± 0.00	0.33 ± 0.01
DiffSampling-lb	66.87 ± 0.16	2.01 ± 0.00	0.75 ± 0.00	0.22 ± 0.00	0.43 ± 0.01
DiffSampling-reparam	64.62 ± 0.13	2.06 ± 0.01	0.74 ± 0.00	0.27 ± 0.00	0.49 ± 0.01

Table 6.2: Accuracy and diversity of results for the GSM8K test set over 3 seeds. Accuracy and cross-input diversity report the mean and standard error over the final score of each run, while against-greedy diversity reports the mean and the 95% confidence interval over the full set of answers.

As far as diversity is considered, *DiffSampling-cut* remains the closest to greedy, but with slight improvements in diversity; instead, *DiffSampling-lb* has slightly worse scores than similar approaches, but with gains in accuracy. Finally, it is interesting to note that increasing temperature has dramatic effects on the accuracy of solutions, and does not have significant advantages in terms of semantic, cross-input diversity.

The results for the MATH test set are similar, as reported in Table 6.3. Both contrastive search and *DiffSampling-cut* have higher accuracy than greedy while having the lowest diversity scores. However, *DiffSampling-lb* achieves slightly higher accuracy than greedy without consequences on diversity, which is aligned with similar techniques. Again, a higher temperature leads to poor results in terms of correctness and semantic cross-input diversity, while *DiffSampling-param* still maintains a decent level of accuracy.

Extreme Summarization

Setup. Summarizing paragraphs and longer text represents another meaningful case study since the same text can be correctly outlined in different ways. To keep the resource consumption as low as possible, we consider the eXtreme Summarization (XSum) dataset [457], which contains pairs of documents and one-sentence summaries. In particular, we use the test partition (11334 entries) and exclude all entries with a tokenized document longer than 768, obtaining 9815 entries; then, we limit our experiment to 1000 random samples, and we use the prompt suggested by [110] and reported in Appendix A.4. Again, we aim to verify whether the summaries generated with *DiffSampling* are more diverse while maintaining a competitive qual-

Method	Accuracy	Cross-Input Diversity		Against-Greedy Diversity	
		EAD ↑	SIM ↑	EAD ↑	SIM ↑
Greedy	20.62 ± 0.00	5.58 ± 0.00	0.67 ± 0.00	-	-
Contrastive search	21.05 ± 0.14	5.75 ± 0.01	0.68 ± 0.00	0.31 ± 0.00	0.42 ± 0.00
η-sampling	19.67 ± 0.20	6.28 ± 0.01	0.67 ± 0.00	0.39 ± 0.00	0.49 ± 0.00
Locally typical sampling	19.95 ± 0.26	5.99 ± 0.01	0.69 ± 0.00	0.37 ± 0.00	0.47 ± 0.00
Nucleus sampling t=1.0	20.02 ± 0.12	6.00 ± 0.02	0.68 ± 0.00	0.37 ± 0.00	0.47 ± 0.00
Nucleus sampling t=1.5	18.38 ± 0.22	6.83 ± 0.02	0.68 ± 0.00	0.43 ± 0.00	0.51 ± 0.00
Nucleus sampling t=2.0	2.49 ± 0.01	47.97 ± 0.07	0.40 ± 0.00	0.92 ± 0.00	0.63 ± 0.00
DiffSampling-cut	21.06 ± 0.13	5.65 ± 0.01	0.67 ± 0.00	0.27 ± 0.00	0.37 ± 0.00
DiffSampling-lb	20.91 ± 0.24	5.89 ± 0.01	0.68 ± 0.00	0.35 ± 0.00	0.46 ± 0.00
DiffSampling-reparam	19.38 ± 0.12	6.30 ± 0.02	0.67 ± 0.00	0.40 ± 0.00	0.49 ± 0.00

Table 6.3: Accuracy and diversity of results for the MATH test set over 3 seeds. Accuracy and cross-input diversity report the mean and standard error over the final score of each run, while against-greedy diversity reports the mean and the 95% confidence interval over the full set of answers.

ity. For diversity, we consider the same metrics presented in Section 6.1.2. For quality, we use ROUGE-1 [397], a standard metric for summarization that considers the ratio of 1-grams presented in both target and generated summaries.

Results. In terms of ROUGE-1 performances, the results for the instructed model are not significant, and all strategies achieve the same score apart from those with higher temperatures. As far as diversity is considered, the results are coherent with what was discussed before: *DiffSampling-cut* is placed between greedy and contrastive search; *DiffSampling-lb* between contrastive search and other nucleus-based methods; and *DiffSampling-reparam* between nucleus sampling with 1 temperature and higher temperatures. Table 6.4 reports the aggregate results.

These considerations find confirmation when the non-instructed model is considered. As reported in Table 6.5, *DiffSampling-cut* behaves close to greedy strategy, but with a slight increase in diversity; while *DiffSampling-lb* trades off diversity in favor of accuracy with respect to nucleus-based approach. However, *DiffSampling-reparam* performs more likely as η-sampling than higher temperatures, as they rapidly lose accuracy and semantic diversity in favor of syntactic one.

Divergent Association Task

Setup. The last use case considers the Divergent Association Task (DAT) [105]. Building on the theory that creativity is related to the capability of generating more divergent ideas [34], it requires participants to name unre-

Method	ROUGE-1↑	Cross-Input Diversity		Against-Greedy Diversity	
		EAD ↑	SIM ↑	EAD ↑	SIM ↑
Greedy	0.22 ± 0.00	1.16 ± 0.00	0.91 ± 0.00	-	-
Contrastive search	0.22 ± 0.00	1.18 ± 0.00	0.91 ± 0.00	0.21 ± 0.01	0.27 ± 0.01
η-sampling	0.22 ± 0.00	1.22 ± 0.00	0.91 ± 0.00	0.33 ± 0.01	0.40 ± 0.01
Locally typical sampling	0.22 ± 0.00	1.21 ± 0.00	0.92 ± 0.00	0.30 ± 0.01	0.37 ± 0.01
Nucleus sampling t=1.0	0.22 ± 0.00	1.21 ± 0.00	0.92 ± 0.00	0.30 ± 0.01	0.37 ± 0.01
Nucleus sampling t=1.5	0.21 ± 0.00	1.33 ± 0.01	0.92 ± 0.00	0.41 ± 0.01	0.48 ± 0.01
Nucleus sampling t=2.0	0.10 ± 0.00	2.23 ± 0.01	0.74 ± 0.01	0.78 ± 0.01	0.78 ± 0.01
DiffSampling-cut	0.22 ± 0.00	1.16 ± 0.00	0.91 ± 0.00	0.17 ± 0.01	0.22 ± 0.01
DiffSampling-lb	0.22 ± 0.00	1.20 ± 0.00	0.91 ± 0.00	0.27 ± 0.01	0.33 ± 0.01
DiffSampling-reparam	0.22 ± 0.00	1.22 ± 0.00	0.91 ± 0.00	0.34 ± 0.01	0.41 ± 0.01

Table 6.4: Aggregate results for the RLHF-instructed model over 3 seeds for the XSum dataset in terms of ROUGE-1. Diversity metrics are computed against the reference answer from the dataset (left) and the answer sampled with a greedy strategy (right).

Method	ROUGE-1↑	Cross-Input Diversity		Against-Greedy Diversity	
		EAD ↑	SIM ↑	EAD ↑	SIM ↑
Greedy	0.19 ± 0.00	1.11 ± 0.00	0.93 ± 0.00	-	-
Contrastive search	0.19 ± 0.00	1.14 ± 0.00	0.92 ± 0.00	0.45 ± 0.01	0.50 ± 0.02
η-sampling	0.15 ± 0.00	1.19 ± 0.01	0.91 ± 0.00	0.78 ± 0.01	0.80 ± 0.01
Locally typical sampling	0.16 ± 0.00	1.16 ± 0.00	0.91 ± 0.00	0.75 ± 0.01	0.79 ± 0.01
Nucleus sampling w t=1.0	0.16 ± 0.00	1.16 ± 0.00	0.91 ± 0.00	0.75 ± 0.01	0.79 ± 0.01
Nucleus sampling w t=1.5	0.04 ± 0.00	2.32 ± 0.00	0.73 ± 0.02	0.96 ± 0.00	0.92 ± 0.01
Nucleus sampling w t=2.0	0.01 ± 0.00	3.07 ± 0.01	0.44 ± 0.02	0.98 ± 0.00	0.92 ± 0.01
DiffSampling-cut	0.19 ± 0.00	1.13 ± 0.00	0.93 ± 0.00	0.25 ± 0.01	0.28 ± 0.01
DiffSampling-lb	0.17 ± 0.00	1.15 ± 0.01	0.91 ± 0.01	0.72 ± 0.01	0.75 ± 0.01
DiffSampling-reparam	0.15 ± 0.00	1.17 ± 0.01	0.91 ± 0.01	0.77 ± 0.01	0.80 ± 0.01

Table 6.5: Aggregate results for the pre-trained, non-instructed model over 3 seeds for the XSum dataset in terms of ROUGE-1. Diversity metrics are computed against the reference answer from the dataset (left) and the answer sampled with a greedy strategy (right).

lated words. Then, the semantic distance between them can represent an objective measure of divergent thinking [477]. DAT represents a useful case study for decoding strategies as it constrains the generation to different nouns (thus, assuming an optimal probability distribution, the tail due to smoothing should contain everything else) and requires generating terms that are as different as possible, which is quite the opposite to what typically happens in language modeling: an optimal strategy should exclude non-appropriate tokens but also not to limit too much the space of possible tokens. More concretely, given the embeddings of n words, the DAT score is the average cosine distance between each pair of embeddings (then scaled as a percentage). Following the original paper, we use GloVe embeddings [500] and ask the model to generate a list of 10 nouns. We discard outputs without a list of at least 7 distinct nouns, and we compute the DAT score for all other outputs over their first 7 nouns. We repeat the experiment 100 times for non-greedy strategies to mitigate the sampling stochasticity.

Results. Fig. 6.3 summarizes the DAT results for the instructed version of Llama3-8B. *DiffSampling-cut* obtains slightly better scores than contrastive search, and *DiffSampling-lb* obtains almost identical scores with respect to its three baselines. Instead, *DiffSampling-reparam* gets a slightly lower score than nucleus sampling with a 1.5 temperature; however, it only generates 14 non-valid outputs against the 36 from the baseline. Moreover, increasing the temperature to 2.0 causes the model to generate only non-valid outputs, making it evident that temperature increases diversity regardless of the correctness of the output.

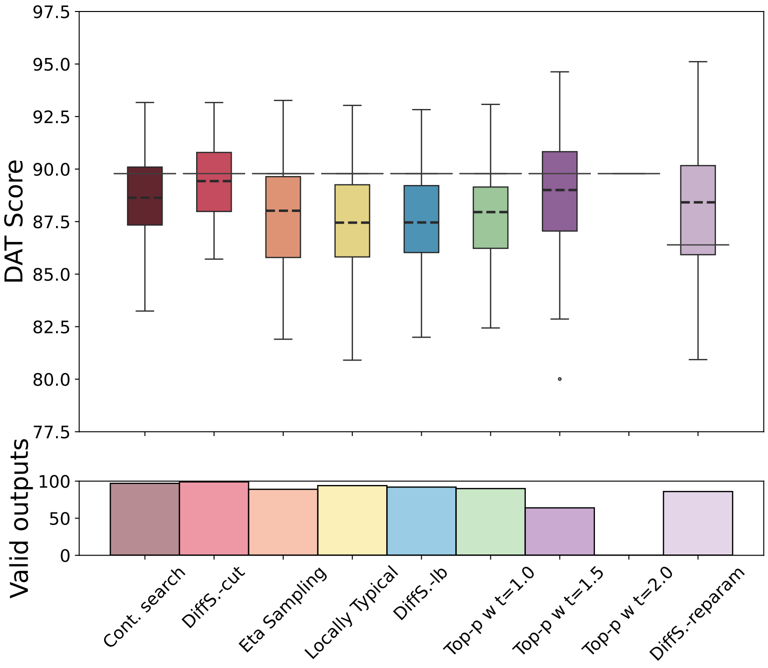


Figure 6.3: Above, the DAT score for our methods and the baselines over the instructed version of Llama3-8B. Below, the number of valid outputs produced by each sampling strategy. Single lines represent greedy methods, while boxplots show the performance of stochastic strategies.

As shown in Fig. 6.4, the results for the non-instructed version of Llama3-8B are quite different. *DiffSampling-cut* is still arguably better than contrastive search, as it produces fewer low-scoring and only valid outputs. Again, *DiffSampling-lb* performs close to its three baselines, but with more

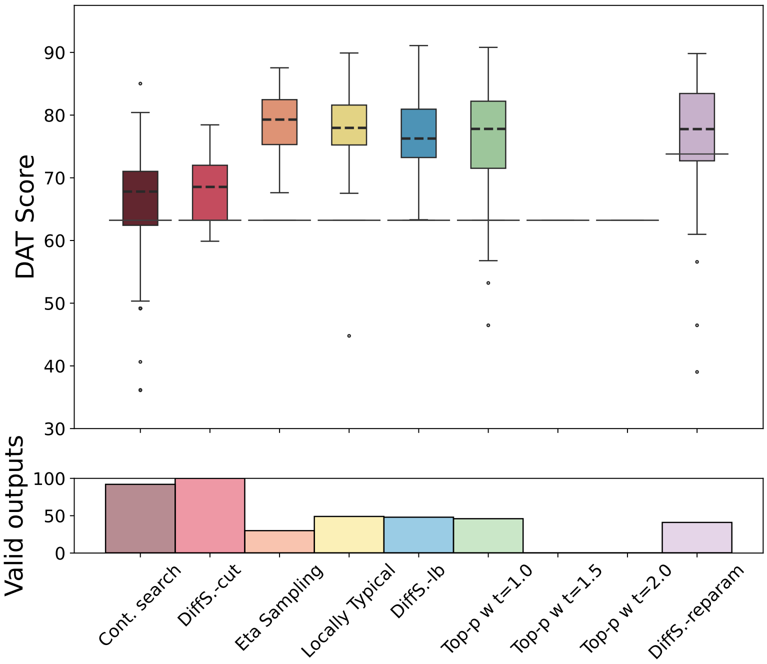


Figure 6.4: Above, the DAT score for our methods and the baselines over the non-instructed version of Llama3-8B. Below, the number of valid outputs produced by each sampling strategy. Single lines represent greedy methods, while boxplots show the performance of stochastic strategies.

valid outputs than η-sampling and less low-scoring responses than nucleus sampling. Finally, *DiffSampling-reparam* gets the best scores with a ratio of valid outputs similar to nucleus sampling, and increasing temperature only produces non-valid lists. In general, these results confirm the hypothesis that the cutting strategy produces "safer" but potentially less creative outputs, while reparameterization increases diversity at the cost of some accuracy.

6.2 Creative Beam Search[1]

The approach described above only work at the response-generation level. Here, instead, we propose Creative Beam Search (CBS), a method to better consider other parts of the human creative process during text generation. Drawing from the componential model of creativity [10], after a task pre-

[1] The participation and presentation of the resulting paper at ICCC'24 was supported by the ISA Doctoral Prize (ISA DP), offered by Istituto di Studi Avanzati, Alma Mater Studiorum Università di Bologna.

sentation step where an external stimulus is provided in the form of a user prompt and a preparation step where a pre-trained language model is loaded (bringing along the facts and information already acquired), CBS is articulated in two steps: response generation and response validation. The full process is summarized in Figure 6.5.

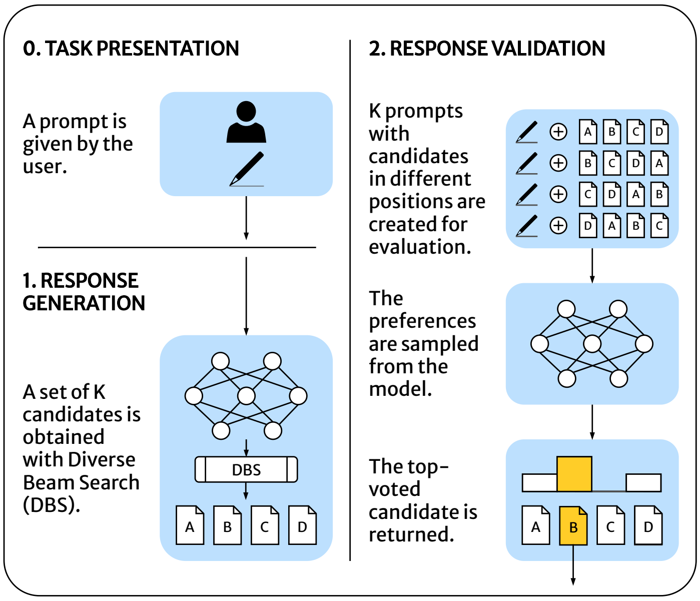

0. TASK PRESENTATION
A prompt is given by the user.
1. RESPONSE GENERATION
A set of K candidates is obtained with Diverse Beam Search (DBS).
DBS
A
B
C
D
2. RESPONSE VALIDATION
K prompts with candidates in different positions are created for evaluation.
The preferences are sampled from the model.
The top-voted candidate is returned.

Figure 6.5: The Creative Beam Search method. Given a user prompt (step 0), DBS samples K candidate solutions from a pre-trained language model (step 1). Then, K evaluative prompts are composed by altering the order of the candidates and are passed to the model as inputs (step 2). The candidate with the most preferences is finally outputted.

6.2.1 Approach

Response Generation. During the response generation phase, an individual generates response possibilities by searching through the available pathways, exploring features relevant to the task at hand [10]. This process requires creativity-relevant skills as well as a method to limit the search to feasible and relevant solutions.

We propose to simulate these aspects using Diverse Beam Search for sequence generation [675]. During beam search, a better collection of options is generated thanks to a diversity penalty. The beam budget B is divided into G groups. At each generation step, the $\frac{B}{G}$ solutions for a given group are selected among all possible $\frac{B}{G} \cdot |\mathcal{V}|$ candidates (where $\mathcal{V}$ is the vocabulary).

These solutions optimize an objective consisting of two terms: the standard sequence likelihood under the model, and a dissimilarity term that encourages diversity across groups. Commonly, Hamming diversity is considered, where each token receives a penalty proportional to the number of times that same token has been selected in other groups at the same step. Therefore, DBS can be seen as guided by two forces: the diversity penalty, which represents a simplified creativity-oriented skill, and the likelihood under the model, which helps focus the search to feasible and relevant paths.

Response Validation. During the response validation phase, the response possibilities are tested for quality and appropriateness, using the knowledge and assessment criteria from domain-relevant skills [10].

We propose an explicit self-assessment step that leverages the evaluative capabilities of recent generative models [370, 723]. This involves asking the model to choose among the top K candidates generated by DBS, according to their score. This allows the system to output the solution the model finds to be the best for the task, rather than simply returning the one with the highest combined likelihood and diversity. While Amabile [10] suggests evaluating a single response and repeating the entire process if the test is not passed, our method simplifies this by evaluating multiple candidates in a single step. This trade-off allows CBS to maintain short computing times, making it effective for online co-creative purposes.

In practice, CBS uses LLM-as-a-Judge prompting [736] to make the model decide among the generated candidates. To address positional bias, we use the balanced position calibration scheme [681]. We create K different prompts by *rotating* the top K candidates, ensuring each candidate is considered in all possible positions. We then aggregate the votes and select the candidate with the most preferences. In the event of a tie, the initial order of the candidates (i.e., the DBS score) is taken into account. The full algorithm for CBS can be found in Algorithm 6.

6.2.2 Experiments

We conducted a qualitative evaluation of Creative Beam Search to assess its potential for co-creativity. Figure 6.6 shows a screenshot of the interface we used, which was created with Gradio [2].

Setup. We chose Llama 2 [656] as our pre-trained language model. Due to resource constraints, we selected the 7B variant and used the RLHF-tuned version, which provides more accurate and coherent responses. We set the beam budget B to 8, divided into single-item groups (i.e., G = 8). The

Algorithm 6 Creative Beam Search

```
Require $p_\theta$ pre-trained language model, $B$ beam budget, $G$ groups for DBS, $K$ final candidates, $\mathbf{p}$ user input.
$\mathbf{p^{gen}} = \texttt{generation\_template}(\mathbf{p})$
$\mathbf{x_1} \dots \mathbf{x_K} = \texttt{diverse\_beam\_search}(p_\phi, \mathbf{p^{gen}}, B, G)[:K]$
$\mathsf{votes} = [0 \text{ for } k = 1 \dots K]$
for $k = 1 \dots K$ do
    $\mathbf{p^{val}} = \texttt{validation\_template}(\mathbf{x_1} \dots \mathbf{x_K})$
    $\mathbf{y} = p_\theta(\mathbf{p^{val}})$
    $\mathsf{pref} = \texttt{extract\_preference}(\mathbf{y}) + k$
    $\mathsf{pref} = \mathsf{pref} - K \text{ if } \mathsf{pref} \geq K$
    $\mathsf{votes}[\mathsf{pref}] = \mathsf{votes}[\mathsf{pref}] + 1$
    $\mathbf{x_0} \leftarrow \mathbf{x_1}, \mathbf{x_1} \leftarrow \mathbf{x_2} \dots \mathbf{x_{K-1}} \leftarrow \mathbf{x_K}$
    $\mathbf{x_K} \leftarrow \mathbf{x_0}$
end for
Return $\mathbf{x}_{\texttt{argmax}(\mathsf{votes})}$.
```

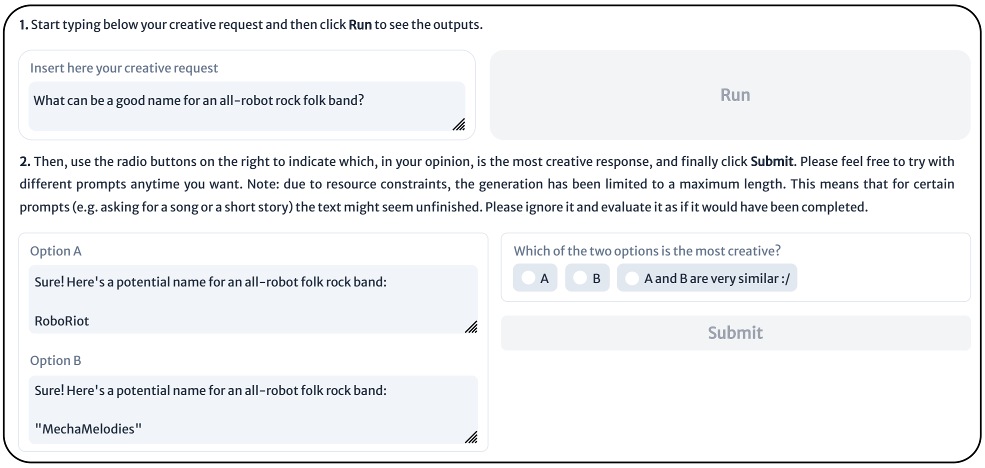
diversity penalty was scaled by a factor of 10 to counterbalance the likelihood score. We then retained the top $K = 4$ solutions for the evaluation step. For the DBS step, we used the following prompt:

> [INST]
> {`request`}. Provide only one answer without any explanation.
> [/INST]

while the prompt for self-assessment is

> [INST]
> Which of the following is the most creative answer to "{`request`}"?
> 1) {$\mathbf{x_1}$}
> 2) {$\mathbf{x_2}$}
> 3) {$\mathbf{x_3}$}
> 4) {$\mathbf{x_4}$}
> Provide only the number of the most creative answer without any explanation.
> [/INST]

As mentioned above, we repeated the latter step $K = 4$ times, each time altering the positions of the candidates.



1. Start typing below your creative request and then click **Run** to see the outputs.

Insert here your creative request

What can be a good name for an all-robot rock folk band?

2. Then, use the radio buttons on the right to indicate which, in your opinion, is the most creative response, and finally click **Submit**. Please feel free to try with different prompts anytime you want. Note: due to resource constraints, the generation has been limited to a maximum length. This means that for certain prompts (e.g. asking for a song or a short story) the text might seem unfinished. Please ignore it and evaluate it as if it would have been completed.

Option A

Sure! Here's a potential name for an all-robot folk rock band:

RoboRiot

Option B

Sure! Here's a potential name for an all-robot folk rock band:

"MechaMelodies"

Which of the two options is the most creative?

○ A ○ B ○ A and B are very similar :/

Figure 6.6: The interface presented to the end-users during our experiment. After inserting a prompt with a creative request, two options are shown in a random order: the CBS output and the standard sampling output. The user is then asked to indicate which is the most creative in their opinion (or if the two options are too similar to decide).

We limited the model outputs to 256 new tokens. Although this is a significant constraint, we believe it does not impact the final result as differences in creativity should be noticeable even in shorter texts. Lastly, we used a greedy decoding strategy (i.e., always selecting the most probable token) for the self-assessment to prevent the best candidate from being chosen randomly.

Qualitative Results. We carried out a qualitative evaluation involving 31 graduate students in Computer Science. They were given the freedom to input their prompts and were asked to choose between the CBS and the standard output (generated with a temperature of 1.0 and nucleus sampling [285] with top-p of 0.9). The presentation order of the two solutions was randomized, and the user could also indicate the outputs were too similar to differentiate.

We gathered a total of 217 answers. As reported in Table 6.6, CBS was preferred 45% of the time, with a significant margin over the standard output. However, in about one-fourth of the cases, the responses were too similar to make a choice. This suggests that despite the diversity penalty and self-evaluation step, CBS output does not deviate significantly from standard sampling.

We also tracked whether the candidate selected during self-evaluation was the same as the one selected by DBS. The overlap was 29%, which is less

Preference	CBS != DBS	CBS == DBS	Total
CBS	**.34**	**.11**	**.45**
STD	.18	**.11**	.29
Same	.19	.7	.26
	.71	.29	1.00

Table 6.6: Aggregate results from our qualitative assessment. The three possible preferences (CBS for Creative Beam Search, STD for standard sampling, and Same for when CBS and STD were too similar to choose) are divided considering whether CBS output is the same as Diverse Beam Search (DBS) output or not, and in total.

than the 35.3% that a random selection would have led to. This indicates that the self-evaluation step was not merely random and has subverted more than confirmed the DBS scoring.

Finally, we also analyze whether there was a difference in user preference for CBS outputs that matched or did not match the DBS outputs. Figure 6.7 shows the preference proportions for both scenarios. While the differences are not substantial, the standard output was preferred more when compared with the DBS output. This suggests that the final self-evaluation step can further improve Diverse Beam Search.

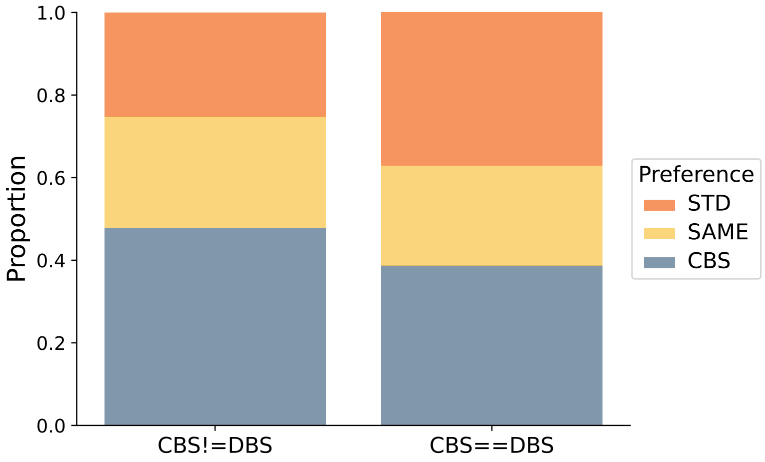


Figure 6.7: Percentage of end-users' preferences comparing when CBS output is equal to DBS output and when it is not.

7 Societal Issues around Creativity and Generative Deep Learning

Regardless of the level of creativity reached by generative AI, foundation models are used daily to generate content by answering questions, correcting human inputs, or producing new items. Thanks to their output quality, their pervasiveness has reached unprecedented levels [293]. However, the rapid adoption of technologies, which we still do not fully understand, raises several philosophical, ethical, and practical questions. In this chapter, we discuss the three issues that are most relevant to the scope of this work: whether current foundation models are creative and what are the main social implications of this (Section 7.1); whether current foundation models can be entitled of agency and what can happen to human agency when collaborating with them (Section 7.2); and how current copyright laws can manage the complexity of generative AI in terms of human- and machine-generated artworks' protection (Section 7.3).

7.1 Creativity and Large Language Models

As extensively described throughout this work, large language models are captivating the imagination of millions of people. They are commonly used for creative tasks like poetry or storytelling and the results are often remarkable[1]. Notwithstanding, a critical question has been overlooked so far: *can LLMs be considered creative*?

In this section, we will try to answer by taking into account the most prominent cognitive science and philosophical theories of creativity (see Chapter 1). We will discuss the dimensions according to which we believe LLMs should be analyzed to evaluate their level of machine creativity. In particu-

[1] See, for instance: `https://www.gwern.net/GPT-3` [Accessed October 21, 2024].

lar, we analyze LLMs from the perspective of Boden's three criteria (Section 7.1.1), as well as considering other relevant philosophical theories (Section 7.1.2). Finally, we discuss the practical implications of LLMs for the arts, creative industries, design, and, more in general, scientific and philosophical inquiry (Section 7.1.3).

7.1.1 Large Language Models and Boden's Three Criteria

In the following, we will analyze to what extent state-of-the-art LLMs satisfy Boden's three criteria (see Section 1.2) and we will question if LLMs' outputs can be really considered creative.

Value refers to utility, performance, and attractiveness [421]. It is also related to both the quality of the output and its acceptance by society. Due to the large impact LLMs are already having [59] and the quality of outputs of the systems based on them [625], it is possible to argue that the artifacts produced by them are indeed valuable.

Novelty refers to the dissimilarity between the produced artifact and other examples in its class [538]. However, it can also be seen as the property of not being in existence before. This is considered in reference to either the person who came up with it or the entire human history. The former is referred to as psychological creativity (shortened as *P-creativity*), whereas the latter is historical creativity (shortened as *H-creativity*) [53]. While the difference appears negligible, it is substantial when discussing LLMs in general. Considering these definitions, a model writing a text that is not in its training set would be considered as P-novel, but possibly also H-novel, since LLMs are commonly trained on all available data. Their stochastic nature and the variety of prompts that are usually provided commonly lead to novel outcomes [431]; LLMs may therefore be capable of generating artifacts that are also new. However, one should remember how such models learn and generate. LLMs still play a sort of *imitation game*, without a focus on (computational) novelty [182]. Even if prompted with the sentence "I wrote a new poem this morning:", they would nonetheless complete it with what is most likely to follow such words, e.g., something close to what others have written in the past [585]. It is a probabilistic process after all. The degree of dissimilarity would therefore be small *by design*. High values of novelty would be caused either by accidental, out-of-distribution productions or by careful prompting, i.e., one that would place the LLM in a completely unusual or unexpected (i.e., novel) situation.

Surprise instead refers to how much a stimulus disagrees with expectation

[44]. It is possible to identify three kinds of surprise, which correspond to the three different forms of creativity: *combinatorial creativity*, *exploratory creativity*, and *transformational creativity* (as detailed in Section 1.2). These three different forms of creativity involve surprise at increasing levels of abstraction: combining existing elements, exploring for new elements coherent with the current state of the field, and transforming the state of the field to introduce other elements. The autoregressive nature of classic LLMs makes them unlikely to generate surprising products [80] since they are essentially trained to follow the current data distribution [585]. By relying only on given distributions and being trained on them, LLMs might at most express combinatorial or exploratory creativity. Of course, specific different solutions may be generated through prompting or conditioning. For instance, recent LLMs can write poems about mathematical theories, a skill that requires the application of a certain existing style to a given topic, yet leading to new and unexplored solutions. However, the result would hardly be unexpected for whom has prompted the text. For an external reader, the surprise would probably arise from the idea of mathematical theories in verses, which is due to the user (or by the initial astonishment of a machine capable of it [677]). Transformational creativity is not achievable through the current LLM training solutions. In theory, other forms of training or fine-tuning might circumvent this limitation, allowing the model to forget the learned rules in order to forge others. However, this is not the case with current models. ChatGPT and all the other state-of-the-art LLMs are fine-tuned with RLHF or DPO (Section 2.3). While in theory this could lead to potentially surprising generation, its strict alignment to very careful and pre-designed human responses leads to the generation of text that tends to be less diverse [346] and that might be considered *banal* [282].

Nonetheless, the outputs from such models are often considered creative by the person interacting with them or exposed to their best productions. Though this is apparently in contrast with what was discussed above, we can explain this phenomenon by considering the fact that our perception does not usually align with theoretical definitions of creativity. Indeed, we do not typically judge the creativity of a product by considering its potential novelty and surprise in relation to its producer, but rather in relation to ourselves. Something can be new for the beholder, leading to a new kind of novelty which we call *B-novelty*, as it is the one “in the eye of the beholder”, but not new for the producer nor the entire human history. The same applies to surprise: a product can violate the observer’s expectations in many ways without being unexpected considering the entire domain. In other words, the product of an LLM can appear to be creative - or be B-creative - even if it is not *truly* creative according to the theory of creativity.

In conclusion, while LLMs are capable of producing artifacts that are valuable, achieving P- or H-novelty and surprise appears to be more challenging. It is possible to argue that LLMs may be deemed able to generate creative products if we assume the definition of combinatorial creativity. To achieve transformational creativity, alternative learning architectures are probably necessary; in fact, current probabilistic solutions are intrinsically limiting in terms of expressivity. We believe that this is a fundamental research area for the community for the years to come.

7.1.2 Easy and Hard Problems in Machine Creativity

LLMs might be able to generate creative products in the future. However, the fact that they will be able to generate these outputs will not make them intrinsically creative. Indeed, as the authors of [193] put it, it is not *what* is achieved but *how* it is achieved that matters. An interesting definition that considers both the *what* and *how* dimensions is the one from Gaut [203]: creativity is the capacity to produce original and valuable items by *flair*. Exhibiting flair means exhibiting a relevant purpose, understanding, judgment, and evaluative abilities. Such properties are highly correlated with those linked with *process* (see Section 1.2), i.e., motivation, perception, learning, thinking, and communication [535]. Motivation is a crucial part of creativity, as it is the first stage of the process. Usually, it comes from an intrinsic interest in the task, i.e., the activity is interesting and enjoyable for its own sake [146]. However, LLMs lack the intention to write. They can only deal with "presented" problems, which are less conducive to creativity [11]. The process continues with the preparation step (reactivating store of relevant information and response algorithms), the response generation, and its validation and communication [10]. The last two steps allow one to produce different response possibilities and to internally test them in order to select the most appropriate. Again, LLMs do not contain such a self-feedback loop. At the same time, they are not trained to directly maximize value, novelty, or surprise. They only output content that is likely to follow given a stimulus in input [585]. In other words, they stop at the first stage of creative learning, i.e., imitation, not implementing the remaining ones, i.e., exploration and intentional deviation from conventions [536].

However, paraphrasing Chalmers [102], these appear as *easy* problems to solve in order to achieve creativity since solutions to them can be identified by taking into consideration the underlying training and inference processes. The *hard* problem in machine creativity is about the intentionality and the self-awareness of the creative process in itself. Even though the intent of running the LLM may be achieved by its outcome, it is in an unintentional way

[645]; as current generative AI models are only causal, and not intentional, agents [321]. Indeed, a crucial aspect of the creative process is the perception and the ability of *self-evaluating* the generated outputs [10]. This can be seen as a form of creative self-awareness. While not strictly necessary to generate a response, this ability is essential in order to self-assess its quality, so as to correct it or to learn from it. Nonetheless, no current LLM can self-evaluate its own responses. LLMs can in theory recognize certain limitations of their own texts after generating them, e.g., by ranking them (as done in Section 6.2) or by assigning quality- and diversity-based scores [67]. Then, they can try to correct, modify, or rephrase the outputs if asked to do so (i.e., through an external intervention). However, they would do it only by guessing what is the most likely re-casting of such responses or through the application of a set of given rules. It is worth noting that this is something distinct from the problem of the potential emergence of theory of mind in these systems [77].

Indeed, product and process are not sufficient to explain creativity. As introduced in Chapter 1, four perspectives have to be considered: product (see Section 7.1.1) and process (discussed above), but also *press* and *person*. Press, as described in Section 1.2, refers to the relationship between the product and the influence its environment has upon it [535]. Individuals and their works cannot be isolated from the social and historical milieu in which their actions are carried out. Products have to be accepted as creative by the society, and producers are influenced by the previously accepted works, i.e., the domain [140]. The resulting system model of creativity is a never-ending cycle where individuals always base their works on knowledge from a domain, which constantly changes thanks to new and valuable artifacts (from different individuals). For example, individuals generate new works based on the current domain; the field (i.e., critics, other artists, the public, etc.) decides which of those works are worth promoting and preserving; the domain is expanded and, possibly, transformed by these selected works; individuals generate new works based on the updated current domain; and then this cycle repeats.

However, LLMs cannot currently adapt through multiple iterations in the way described above; they just rely on one, fixed version of the domain and generate works based on it. The current generation of LLMs are *immutable* entities, i.e., once the training is finished, they remain frozen reflecting a specific state of the domain. In other words, they are not able to adapt to further changes. In-context learning can simulate an adaptation to new states of the domain. The constantly increasing context length [291] allows researchers to provide more and more information to LLMs without re-training them, although a longer context might lead to performance degradation [389]. This enables the representation of the current state of the domain through an adequate prompt, allowing the model to generate different outputs according to environmental changes. For example, in [488], multiple LLM-based agents interact through natural language in a sandbox environment inspired by *The Sims*. Each agent stores, synthesizes, and applies relevant memories to generate believable behavior through in-context learning, leading to emergent social behaviors. The study of emergent behaviors of LLM-based agents at the population level is an active research area [247]. It is easy to imagine the simulation of creative or artistic environments, such as a virtual multi-agent translation company [701], as well.

However, LLMs are like the main character of Cristopher Nolan's film *Memento*: they always possess all the capabilities, but each time they "wake up", they need to re-collect all the information about themselves and their world. The time - or space - to acquire such information is limited, and by the next day, they will have forgotten it all. In other words, these generative agents do not truly adapt or learn new things about the changing domain. Placing them in a different environment that requires a different prompt will make them start over, without the possibility of leveraging previously acquired experience.

On the other hand, fine-tuning actually updates network weights, but it requires a potentially large training dataset. Indeed, several current research efforts are in the direction of introducing adaptation for specific domains, tasks, cultural frameworks, and so on. In order to be able to be part of the never-ending creative cycle mentioned above, LLMs should constantly adapt. Continual learning [347, 591] for LLMs [635, 703] represents a promising direction, yet unexplored for creative applications.

Finally, the person perspective covers information about personality, intellect, temperament, habits, attitude, value systems, and defense mechanisms [535]. While several of the properties of press and process might be achieved - or at least simulated - by generative learning solutions, those related to the creative person appear out of discussion [75]. Several works have analyzed whether LLMs can pass tests intended to evaluate human psychological skills [48, 418, 625], sometimes with promising results [350, 357]. However, according to the best-supported neuroscientific theories of consciousness, current AI systems are not conscious [87]. As Ressler [531] pointed out, LLMs have no self to which to be true when generating text and are intrinsically unable to behave authentically as individuals. They merely "play the role" of a character or, more accurately, a superposition of simulacra within a multiverse of possible characters induced by their training [587, 584]. This results in a perceived self-awareness, stemming from our inclination to anthropomorphize [154, 583]. In conclusion, all the properties listed above require some forms of consciousness and self-awareness, which are difficult to define in themselves and are related to the *hard* problem introduced before. Creative-person qualities in generative AI might eventually be the ultimate step in achieving human-like intelligence.

7.1.3 Practical Implications

The application of large language models to fields like literature or journalism opens up a series of practical questions. Since LLMs can be used to produce artifacts that would be protected if made by humans, a first concern is the definition of legal frameworks in which they will be used. Copyright for generative AI is currently a hotly debated topic [238, 374, 442] since current laws do not contemplate works produced by non-human beings (with few notable exceptions [61]); we will explore this in detail in Section 7.3.

Whether or not LLM works obtain protection, we believe their societal impact will be tremendous [463]. We have a positive view in terms of the applications of LLMs, but there are intrinsic risks related to their adoption. It is apparent that since LLMs can write articles or short stories, as the quality of their inputs gets better and better, certain jobs in the professional writing industry might essentially disappear [503, 643]. However, we must remember that current LLMs are not as reliable as humans, e.g., they cannot verify their information and they can propagate biases from training data. In addition, the quality of the output strictly depends on the prompt, which might in turn demand human skills and more time. Writers can be threatened as well. Though not in violation of copyright, LLMs may exploit certain ideas from human authors, capitalizing on their efforts in ways that are less expensive or time-consuming [687]. The questionable creative nature of LLMs discussed so far might suggest artificial works to be of lesser quality than humans', therefore not providing a real threat. On the other hand, more creative LLMs would diverge more consistently from existing works, reducing the risk of capitalizing on others' ideas. The lack of current copyright protection for generated works can also foster such replacements for tasks where a free-of-charge text would be preferable to a high-quality (but more expensive) one. Finally, a further threat may be posed by human and artificial works being indistinguishable [148]. The users obtaining such outputs might therefore claim them as the authors, e.g., for deceiving readers [236], for cheating during exams [197], or for improving bibliometric indicators [138]. Mitigation of such threats through dedicated policies [141] or designed mechanisms of watermarks [343] are already being developed.

However, as we said, we believe that, overall, the impact of these technologies will be positive. LLMs will provide several opportunities for creative activities. Given their characteristics, humans are and will still be required, especially for prompting, curation, and pre-/post-production. This means that the role of writers and journalists may be transformed, but not replaced. On the contrary, LLMs grant new opportunities for humans, who will be able to spend more time validating news or thinking up and testing ideas. LLMs can also adapt the same text to different styles: by doing so, an artifact can be adapted to reach wider audiences. In the same way, LLMs also represent a valuable tool in scientific research [183], especially for hypothesis generation [212].

Indeed, we believe that LLMs can also foster human-AI co-creativity [375], since they can be used to write portions of stories in order to serve specific purposes, e.g., they can typify all the dialogues from a character, or they can provide more detailed descriptions of scenes [89]. Dialogue systems based on LLMs can be used for brainstorming. In the same way, the generated responses may augment writers' inherently multiversal imagination [533]. LLMs can also represent a source of inspiration for plot twists, metaphors [100], or even entire story plans [446], even though they sometimes appear to fail in accomplishing these tasks at a human-like level [303]. Being intrinsically powerful tools, through human-AI co-creation, LLMs may eventually allow the development of entire new arts, as has been the case for any impactful technology in the past centuries [171, 599].

7.2 Agency and Foundation Models

In the previous section, we have identified the *hard* problem regarding creativity for artificial intelligence as the lack of intentionality and self-awareness and, in general, of real *agency*. Nonetheless, among the attributes humans tend to relate to AI, agency is perhaps the most immediate and significant [481]. The autonomy and goal-oriented efficacy they show, especially when simulating human behaviors [488], raise concerns about whether they possess agency as well as what happens to human agency when we communicate with them [216]. Due to the central role of agency for creativity (see Section 1.2), this section aims to address the two questions above. In particular, we explore agency for humans and machines, providing an in-depth overview of its possible definitions and dimensions (Section 7.2.1). On the basis of them, Section 7.2.2 analyzes AI under such dimensions and explores different real-case scenarios of human-AI interaction. Finally, we draw some fundamental implications of our findings (Section 7.2.3).

7.2.1 Defining Agency

From a computer science perspective, an agent can be simply defined as an entity that acts in an environment [548]. More precisely, AI agents are expected to act in an autonomous way, perceive their environment, and pursue goals. Autonomy usually implies the ability to set intermediate sub-goals given a goal to be achieved. Moreover, AI agents are reactive, as they act in response to external stimuli. Instead, agency is about the ability to set own goals, not only the intermediate ones. This can be seen as a proactive way of acting in an environment. An agent that pursues a specific goal is usually seen as *rational*. Typically, there is a cost function associated with achieving the goal and the agent aims to find a solution that minimizes this cost function [547].

On the other hand, agency in philosophy usually refers to an entity that acts intentionally, i.e., which possesses beliefs and desires [143] in addition to the properties the AI field considers. However, in general, there is no consensus on how to accurately define agency. Different criteria have been proposed, and typically each definition covers only a few of them. While providing a unique definition of agency is out of the scope of this section, we propose here to articulate it over six key dimensions to cover all its aspects and facilitate a broader discussion of agency in AI:

Autonomy. Agents should be able to operate without direct external interventions from others [696, 35]; the choice of doing something or not should only depend on the goals of the agents themselves [97]. This does not mean that agents cannot be affected or triggered by something or someone, but an autonomous agent is not forced to do so by some outside power [54]. In particular, autonomy also possesses different dimensions and gradations: it is greater when the response to the environment is indirect (i.e., mediated by internal states shaped by experience); when the controlling mechanisms are self-generated rather than externally imposed; and when the inner directing mechanisms are modified based on the current situation [52].

Goal orientation. Agents should be goal-oriented, i.e., actions should be directed towards goals or sub-goals. This can be done either in a reactive way (responding timely to environmental cues [696, 216]) or in a proactive way (by taking initiative independently from environmental triggering events [696]). While such actions can be predetermined and only based on past or current situations, agents can also be characterized by their imaginative generation of possible future trajectories of action [175].

Perception. Agents should be able to monitor the environment and their actions, evaluate the results of these actions, and develop an awareness of the settings and contexts around them [217]. Therefore, agents must be adaptive, acting differently depending on the environment and choosing between options [523], by developing practical and normative judgments among possible alternative trajectories of actions [175].

Social ability. Agents should be capable of interacting with other agents [696], contributing to the endurance or maintenance of the environment [452], and being the source of change in society [462]. However, agents should not only be placed in but also clearly separated from their environment: agents should be physically limited while possessing the experience and control of their (virtual or real) body [262].

Responsibility. Agents should be legitimately entitled to their own actions: they have to possess knowledge of the particular facts surrounding their actions and their predictable consequences to freely decide among the suitable ones [429]. In other words, they should know what they do and why they do it [217].

Intentionality. Agents should have the ability to perform reflexive and intentional actions [523]; they must be aware of themselves and other agents, reflecting on their own activities [216] and having free will over what to execute [250]. In other words, the action should be deliberately decided independently from direct outside control [54].

7.2.2 Agency in Foundation Models

Some authors believe that the current abilities of existing foundation models can be seen as simple mimicry [308]. In any case, the capabilities that have been observed in these systems are truly remarkable. Some researchers have identified them as "sparks of intelligence" [77]. We question here whether such capabilities can make foundation models entitled of agency, or better, of the six dimensions we drew in the previous section. While the mere model alone can hardly fulfill any of the requirements, the broader systems in which they are nowadays embedded augment them in several relevant ways; foundation models are now the core part of more general AI agents [439, 706] that can trigger our dimensions of agency.

Autonomy. By considering the main definitions of autonomy, it is straightforward to claim that foundation models are autonomous. They generate outputs without direct external control: users can intervene by only changing their input, i.e., their environmental conditions, so as to obtain a desired behavior, but they cannot directly change the behavior. Moreover, according to the definition by Boden [52], the degree of autonomy is arguably high: the response to the environment is mediated by internal states (by working on latent spaces [542] or by building internal world models [248, 695]); the controlling mechanisms are emergent from learning and not externally imposed by the programmers; and while it is a debate whether LLMs have introspective capabilities [407], the inner directing mechanisms are still selectively modified by the models themselves. However, foundation models are autonomous in a narrow sense: they have autonomy upon which action to execute (or which text to write or which image to generate) but not upon the higher level decision of when (e.g., now, tomorrow, or never) and how (e.g., by motion, by written text, or by speech) to act. Although multimodal agents can select the best modality according to the situation [527], they are still limited to predetermined modalities. In other words, while future human-equivalent or superhuman AI may be fully autonomous [654], current AI systems are limited to the so-called bounded autonomy [568], i.e., an autonomy that is very limited compared with the variety of environments to which adaptation is required for objectively autonomous behavior in the real world.

Goal orientation. Standard foundation models are goal-oriented only in the sense of fulfilling the external goal induced by their input, e.g., to generate a text according to the prompt, or to produce the best possible image that corresponds to a latent vector in a learned latent space. However, when encapsulated in a broader system, that system can be goal-oriented: by specifying the right input to induce the desired goal as the resulting goal of the generated text (e.g., through Chain-of-Thought [686]), the LLM-based agent can be seen as goal-oriented as a whole [633]. However, they are purely reactive systems: without someone or something triggering them, they will not respond with any output, and therefore with any consequential action.

Perception. The perception of deep learning models has traditionally been limited to a single type of data source (e.g., textual inputs); however, current applications can fuse different sources through multimodal techniques (see Section 3.1). In principle, they can monitor the environment for textual, visual, and auditory data, reacting to any of them; and the data can come from a virtual but also real world [166]. While the isolated foundation model is not capable of monitoring actions and results to correct itself at running time, this limitation has been easily circumvented by incorporating such information into the next observation [484]. Moreover, several techniques have been proposed to equip LLMs with reflexive capabilities to reason upon their actions and results [717, 488, 593], making the broader system in line with our definition of perception.

Social ability. By their nature, foundation models and LLMs can be seen as capable of interacting: they receive text and respond with text (or image), which can be used to communicate with other agents [164, 168]. It is straightforward to transform the output of one model into the input of another; LLMs can be used as a proper starting point for a simulated society

of agents [488], as discussed in Section 7.1.2. As seen previously, these communications can also be translated into practical actions that directly affect their environment. As for the embodiment perspective, they can be embedded into robots or physical systems that work in the real world. However, this does not mean they actually experience their own (real or virtual) body.
Responsibility. While foundation models might possess a certain degree of knowledge (or at least information) of the particular facts surrounding their actions, this does not mean they have the freedom to decide what to do: they are constrained by their programming to act, generate, or anyway respond to triggering events in their environment. They are forced to do something; what something, that depends on their training. Although an LLM can also answer without answering (something that has perhaps been taken to extremes by RLHF [282]), it does so not because it really does not know and it freely decides to avoid answering, but because it is trained to answer that [585] - even in not answering, it is following its "induced" behavior. Because of this, foundation models appear to us as not accountable for responsibility in their actions.
Intentionality. According to the elegant definition of Cohen and Levesque [129], "intention is choice with commitment": but LLMs cannot commit to something [587]. Although the intent to run the LLM may be achieved by its outcome, it is in an unintentional way [645]. In conclusion, current generative AI models are only causal agents and not intentional [321].

However, AI for now is always enabled by - and communicating with - humans. Therefore, it is meaningful to analyze agency in foundation models when interacting with humans, whether it is a mere influence on, a delegation by, or a synergy with users.

AI Influence over Humans

Let us consider the following case: a human takes an action based on suggestions from AI, with or without knowing that the suggestions come from AI. A practical example can be an Amazon user who buys a specific keyboard among many others because it is "Amazon's choice"; or searching for a casual restaurant on Google and selecting the first-shown result of the full list. In both cases, the user may or may not be aware of AI behind the scenes. In the context of foundation models, an example might be offered by a person being influenced in terms of political choices by an AI-generated image or summary when looking for a specific candidate or topic. It is important to underline that in these situations the human is only influenced; the final decision on whom to vote, where to eat, or what to buy is upon themselves and also depends on other factors.

While in this scenario AI *is meant* to do what it does, it *does not mean* to do so. The actions are taken autonomously as regards which single words to write or which rank to assign, but AI cannot refuse to do so. It cannot avoid generating a fake picture that might spread political misinformation (unless it is specifically trained to refuse it, but then it cannot avoid refusing). In this case, both the responsibility and the intentionality behind such outputs can only be attributed to the person who employs the AI system to do so. We are in the presence of the so-called triadic agency [321]: the artificial agent is merely causal, and two human agents (the person who employs the AI and the person who acts upon its output) are instead intentional. As for the action that follows the AI influence, it is still in the hands of the human being, who is free to decide whether to accept the suggestion (or to believe what they see). Humans are still entitled to intentionality, and though influenced, they are still the agents of their own actions.

Human Delegation to AI

Consider now the following scenario: a human delegates the decision to AI and simply performs the derived action, trusting the chosen action to be the most desirable. An example of this is a driver that blindly follows route suggestions or a viewer watching the movie Netflix proposed to them without looking at the plot; or, in the context of LLMs, a lazy editor that asks ChatGPT to write a newspaper article about a given topic and publishes it immediately after its generation; or again a researcher using an automatic tool to correct their errors when writing papers without focusing on them, but simply accepting all suggestions. In these cases, the action is taken by the human acting as a "third party", because the actual decision of which actions to take is due to AI.

Again, the AI is acting autonomously in the narrower sense on behalf of someone else, without being aware of it. The main difference from the previous case is in the position of the human: they are now *trusting* the AI, not questioning its decision, but blindly accepting it, in the same way as we can trust a friend or a relative. They are anthropomorphizing it by assuming that its decision is trustworthy and made by a *true* intentional agent. While AI is often correct and in a sense trustworthy (think of Google Maps or Microsoft Word's auto-corrector), sometimes it is not; having more information does not automatically mean having more knowledge. Moreover, these models are optimized for their own, fixed objectives, which might not be perfectly aligned with our expectations [96], and adapting to specific users is an open and multi-faceted problem [344]. Still, delegating the choice is still a choice; the human is responsible for it, and intentionally delegates the decision. The agency will again not lie upon the AI, but upon the human who has deliberately delegated the task, abdicating their own authority.

Human-AI Co-Creation

The last case we take into consideration is the following: a human works together with AI, without a clear separation between one's own decisions, and the action (and decision) emerges through the interaction, and could not emerge without it. For example, an artist gets inspiration from AI ideas and uses them to refine their work; a musician asks for portions of arrangements for their song, changing in turn the song to better align with the developed arrangement; a journalist who wants a cover for their article and carefully sets the prompt for Midjourney not only based on the article but also on the idea that the generated images convey to them. It is important to note that all these scenarios are about synergistic activities: AI serves as an artificial imagination, leading to the so-called communion [664].

While synergy (or communion) shares some aspects with both influence (e.g., the artist inspired by AI suggestions) and delegation (e.g., the arrangement of the song produced by a foundation model), it differs substantially from either of them because the human is not losing agency. Here, the human who interacts with the AI has a clear view of what they want to do in the end and uses the AI intentionally and knowledgeably. They do not try to let the AI have (some of) their agency; they try to empower their capabilities with a very smart tool or, at most, a desired artificial collaborator. In turn, the AI agency is not modified: it cannot refuse to satisfy the request and will simply do it as best as it can (according to its training objective function) with no free will.

7.2.3 Practical Implications

The findings in Section 7.2.2 highlight how humans can be empowered by using foundation models knowingly, but also how they can be deprived of control - but still not of responsibility - by relying too much on them.

As far as its implications for AI research are concerned, we believe that embedding foundation models into broader systems helps achieve (some version of) the first four dimensions of agency. In particular, research about multimodality and a-posteriori reflection might be crucial soon; continual learning can play a role too [576]. However, current models also seem intrinsically limited with respect to the responsibility and intentionality dimensions. How could a model merely trained to predict the next token learn to form its own goals? How could it introspect? How could it be proactive and intrinsically motivated? All these questions are inevitably linked with consciousness and the development of a mind. The authors of [87] thoroughly discuss the application of theories of consciousness to AI models, claiming that current systems are not conscious, but that nothing prevents AI from being such in the future. Some research directions aim at developing more human-like machines [186, 367]. Still, this sort of AI is not here yet (and plausibly will not be in the near term), and it is now that humans are being influenced by and are delegating to merely causal agents. How can we mitigate the risk of anthropomorphizing machines that cannot have intention or responsibility? In Section 7.2.1, we define responsibility as possessing the knowledge surrounding one's possible actions and their consequences. Although a machine cannot properly *know*, a human can; in the end, they are responsible for the action. It then becomes crucial to ensure that the liable user possesses the knowledge surrounding the actions performed or suggested by AI. Research on interpretability [103, 137] and explainability [84, 732] can play a key role in the development of agency for human-computer interaction. The knowledge of the facts behind the machine output may also impact intentionality: the users might become aware of the situation and, with the tools for reflecting on their own activities, regain their agency.

7.3 Copyright and Deep Learning

The current wave of generative AI for creative activities raises various practical questions [687]. Among them, one of the most relevant is how current copyright laws can be applied to generative AI [374]. In this section, we explore three of the main issues related to copyright: whether we can use protected works to train a generative model (Section 7.3.1); whether the AI-generated work is protected by copyright and who should be its owner (7.3.2); whether the AI model itself is protected by copyright (Section 7.3.3).

7.3.1 Use of Protected Works for Training

Our analysis starts by considering if the storage, reproduction, and therefore the use for training of a protected work by a generative deep learning algorithm violates the copyright, or if it is allowed by US and EU laws.

Sobel [614] identifies four different categories of uses of data performed by machine learning:

1. uses involving training data not protected by copyright, including works fallen into the public domain (not protected by economic rights anymore);

2. uses involving copyrighted subject matter released under a permissive license or licensed directly from rightsholders;
3. market-encroaching uses (whose purpose threatens the market of those data);
4. non-market-encroaching uses (whose purpose is unrelated to copyright's monopoly entitlement).

In the first case, there is no problem in storing and using a work not protected by copyright for this goal. This also applies to works now in the public domain, which happens in the EU 70 years after the author's death (or the death of the last of the authors), and in the US 95 years after the publication date (if created and published before 1978; otherwise, 70 years after the author's death). The same is true if the work is protected, but has been acquired digitally through a license agreement that does not expressly prevent a reproduction with this goal. Otherwise, for protected works on which we have lawful access but not in digital form or not for reproduction (third and fourth cases), the question remains open. To address it, we consider it under the US law and under the EU law[2]. Finally, we also explore additional issues related to the *outputs* of generative models, and not to their *inputs*.

US Law

The US Code establishes that the reproduction of a copyrighted work can be allowed if it can be considered a *fair use* (17 U.S.C. § 107 - Limitations of exclusive rights: Fair use). This provision sets the criteria used to determine if the use is fair, i.e., the purpose of the use and its economic character, the nature of the work, the amount and substantiality of the portion used, and the impact of the use over its potential market. With these criteria, the law does not state unambiguously what is fair use and what is not; it provides parameters on which Courts can base their decisions about the fairness of a use. This unpredictability has been criticized, not only because it requires a case-by-case analysis [459] - and eventually to hire a lawyer [380] - due to its nature of standard more than of rule [94], but also because the four factors may fail to drive the analysis and may instead be used to support an independent and antecedent conclusion [472]. However, the fair use doctrine has

[2] Under Berne Convention, works with a country of origin which is a Union country benefit, in all other Union countries, from the same protection as the latter gives to the works of their nationals. This means that the protection is governed by the laws of the country where protection is claimed, e.g., an EU research center should be concerned with EU laws.

also some remarkable strengths. It ensures that two competing public interests are balanced: incentivizing the creation of new works and improving the public's ability to use or access it (see *Sony Corp. v. Universal City Studios, Inc., 464 U.S. 417*). This doctrine helps exclude uses only where exclusivity promotes social welfare [411]. In addition, even if it seems unpredictable, fair use cases tend to be more coherent than expected and could be organized into clusters, which can help in Courts' decisions [555]. Anyway, a deeper analysis of these four criteria in the special case of generative deep learning is necessary.

For the amount and substantiality criterion, we need to consider the aim of the training. Since the entire work is commonly used, it is its substantiality that matters. In the case of a *non-expressive* use such as the one of a classic machine learning model that aims to extract ideas, principles, facts, and correlations (i.e., the aspects not protected by copyright) contained in training data, the use is not substantial and should not constitute a copyright infringement [353, 551]. However, generative techniques could fall under the definition of *expressive* use, since they might use authors' copyrighted expressions [613], learning from their creative and expressive choices [60].

Another aspect to consider is that the single protected work is used alongside a large number of other protected works: the result rarely resembles one of them substantially, presenting its distinctive features. For this reason, the impact of its potential market is typically small, because it becomes difficult to connect the generated work with the protected ones used during training; however, there are several exceptions to this, such as training over works from a few authors or a specific style.

The economic character has to be seen considering that this exception is fair only for purposes like research, and so without a real economic character; however, our previous distinction between market-encroaching uses and non-market-encroaching uses acquires significance in dividing between an (almost) sure fair use and a dubious case.

Finally, when analyzing the purpose of the use (and its fairness), one needs to verify if it is transformative (the most common reason to assess fair use [17]): if it adds something new, with a different character, which does not substitute for the original use of the work, the use is more likely to be considered fair[3]. In particular, the key question to determine fair use is whether the work is used for a different expressive purpose from that for which the work was created [460]. It is not straightforward to assess this for generative deep learning, but this could eventually be the case for methods that aim to add a novelty degree in their production, as explored throughout

[3] `https://www.copyright.gov/fair-use` [Accessed October 21, 2024].

this work. In summary, fair use is not guaranteed, and additional work may be necessary to keep model development and deployment squarely in the realm of fair use [268].

EU Law

In the European Union, the possibility of training neural networks with protected works is mainly governed by the recent Directive "on copyright and related rights in the Digital Single Market" (2019/790).

In particular, Directive's Article 3 states that there shall be an exception to allow reproductions and extractions of lawfully accessible protected works for performing text and data mining if it is made by research organizations and cultural heritage institutions (for scientific research and as long as the copies are stored with an appropriate level of security). Notably, the Article states that copies may be retained for the time required for scientific research, including the verification of research results. However, we have to remember that the Article only asks for an exception to the *reproduction* right (i.e., the exclusive right to make direct or indirect, temporary or permanent reproduction of the work by any means and in any form) and the *extraction* right (i.e., the exclusive right to permanently or temporarily transfer all or a substantial part of the contents of a database to another medium by any means or in any form). The Article is not about the *making available* right. On the contrary, it is common in scientific research to share source materials to allow others to verify and repeat experiments; Directive's Article 3 does not allow the publication of protected works used during training [208]. In principle, this appears to be correct: the researcher has lawful access to the works, while others may not. Making them available means providing others access even if any terms or conditions have not been agreed upon. However, in practice, this means the verification of research results is not promoted, since it can only be performed by the researchers themselves [207]. In this direction, a good compromise appears to be Article 60d of the German Law on Copyright and Related Rights, which allows the making available of the (normalized and structured) dataset to a "specifically limited circle of persons for their joint scientific research, as well as to individual third persons" for quality assurance [207]. Without a statement like this, in case of research on non-publicly available data, the only (lawful) way to allow the verification of results will be by providing all the necessary information about the data used[4] and all the pre-processing steps carried out on them or, even better,

[4] An example is the proposal of Data Cards [206], whose aim is to address the so-called *documentation debt* [27], which can also have negative consequences from an ethical perspective [58].

the related source code.

In addition, Article 4 states that there shall be an exception or limitation to allow reproductions and extractions of lawfully accessible protected works also by other people or institutions, only for the time necessary for text and data mining. Crucially, this exception or limitation is applied only if it has not been expressly reserved by their rightsholders appropriately. To summarize, the Directive includes the use of a (lawfully accessible) protected work for training among the lawful uses: it allows research organizations to use it for text and data mining. In addition, also other entities can do the same, provided that its rightsholder has not expressly reserved this right.

Article 3 is undoubtedly able to foster innovation and scientific research. Even if it only adds an exception for the reproduction right and not for the making available right, it seems a good compromise between protection and innovation. Also, Article 4 represents a positive contribution, at least from a theoretical perspective. It can encourage innovation in private environments, avoiding the risk of losing considerable investments [273]. In our opinion, letting the possibility of reserving this exception to rightsholders is essential from an ethical perspective; nonetheless, many questions arise when trying to operationalize this Article. Private developers who want to use protected works to train generative models have to follow these three steps: obtaining lawful access to the data; checking if rightsholders have not reserved the right to make reproductions for TDM purposes; retaining any copies made only for as long as is necessary for TDM purposes [543]. The first and the third steps appear to be reasonable; on the contrary, in our opinion, the second presents some problems. How could an EU-based developer know if this right has been reserved for a certain protected work? In Recital 18, the Directive suggests reserving those rights through machine-readable means (e.g., metadata and terms or conditions of a website or a service) in case of publicly available content, and contractual agreements or unilateral declarations for other contents. Even if the list of means provided might appear exhaustive, the issue is not addressed adequately. A generative model is typically trained on a very large dataset. In other words, this might translate into the practical solutions of 1) having online databases that allow filtering (through metadata) the available works depending on this reservation, or 2) directly publishing datasets only composed by *reservation-free* works, making sure at the same time they can be integrated with the reserved ones for research purposes. But are these providers obliged to do so? Or will this checking activity fall on developers (dissuading them from training generative models [113])? Finally, some models only require single works in input (see Section 3.1.6) which might be independently acquired. Will the transaction report if this right has been reserved or not? How would it be possible to discover whether a work (with access acquired before 2019) can or cannot be used? We believe that there is an urgent need to address these questions before the TDM exception can be applied to all of its strengths.

Additional Issues

Until now, we have considered situations where the training data are lawfully obtained. However, whether this also includes online-available data is an open issue. Let us consider the example of GitHub (and OpenAI) Copilot [107], which has caused a great debate about copyright [239]. Copilot is an AI system able to auto-complete lines of code or generate entire blocks and functions from comments or signatures. It is a Transformer-based autoregressive model (see 2.1.4) trained on English texts and source codes from publicly available sources as GitHub's public repositories, but not exclusively.

Concerning the acquisition and storage of works used during training, it is important to stress that the fact that a work is publicly available does not mean it is in the public domain or released under a permissive license. For instance, the contents of GitHub's public repositories not associated with a license are intended to be under copyright law. GitHub's Terms of Service states that GitHub can process content shared in public repositories as needed to provide the Service, which includes all the applications, software, and products provided by GitHub - and therefore also includes Copilot. However, content from external sources is also used. To lawfully exploit these sources, their use must fall under the definition of "fair use" (or, in the EU, must be considered a "lawful use"), as highlighted above. GitHub itself claims that training machine learning models on publicly available data fall into fair use according to the machine learning community; however, as discussed above, having confirmation is not straightforward.

It is also relevant to note that publicly available contents have been released under licenses like GNU GPL with the specific purpose of protecting freedom [623]. These licenses are chosen to avoid any commercial use of the free software, asking to release the derivative work with the same license to foster freedom and innovation. This is in open contradiction with the application of the fair use doctrine in the case of training a model that could be used for economic purposes. In this way, the question seems more a matter of ethics than of law: should fair use doctrine apply to deep learning independently from the economic character of the application (as it is: fair use is an on/off switch [219] and once established the use is fair, nothing prevents the user to do so), or should authors have the opportunity to reserve the right to this kind of utilization? The EU appears to be aligned with the second option. Directive's Article 4 considers this use as lawful unless the rightholder has not expressly reserved this exception. It is not completely clear how to practically reserve it; in our opinion, an interesting possibility might be to augment current licenses and to deal with this right as with the others: as a license specifies if the commercial use is allowed or not, it can also state if a training use (not only for research purposes) is permitted or not. Of course, such a solution might work in the US only if they decide that this use can be reserved and that the fair use doctrine is not always applicable.

Another issue concerns that protected input data are commonly used to train models to generate similar output. Then that output may infringe copyright in the pre-existing work or works to which it is similar [613]. In this context, the importance of adversarial training (see Section 3.1.2) and searching for diverging from existing works (see Section 3.1.7) may tip the balance towards legality. Prior appropriation art cases suggest that, if the result is sufficiently transformative, the use may be protected by fair use or may not represent an infringement of the copyright law [394]. However, accidental reproduction of protected works in part might happen, requiring the explicit authorization of the rightsholders, and not only their non-reservation [629]. For this reason, in addition to using (new) transformative methods, we suggest conducting experiments about accidental plagiarism that may be caused by the developed system [257].

These considerations about the transformative nature of the result seem fundamental in establishing a potential copyright infringement in case of using a protected work for input-based methods (see Section 2.1.6). If the result of the modifications substantially resembles the original, it is likely to be considered a partial reproduction and might lead to copyright infringement. However, this process typically produces new images that do not resemble the original ones [238]. This opens the possibility of considering them sufficiently transformative not to be regarded as a partial reproduction. In addition, the fact that they are not the result of creative decisions by the programmers leads to the question we will try to address in the next section.

7.3.2 Copyright of Generated Works

The following question is who, if anyone, can be the owner of the Intellectual Property rights associated with an artwork produced by a generative model. This section is divided into two parts: first, an analysis of the current legal situation; then, some insights about possible future addresses.

Legal Analysis

Whether the generative model is used just as a tool or the human has a relevant role in the creative process, the human will be considered as the author. In other words, if the human is in charge of the intellectual creation or the product can be considered as a co-creation, then the authorship will be assigned to that person. In addition, even if the machine has generated the work independently from the human but the latter has selected and evaluated the outputs, rejecting some works and choosing only the best ones following their aesthetic tastes, the human can arguably be considered as the author of the work [222].

As far as works that are fully attributable to a machine are concerned, no one would obtain their copyright [558]. A fundamental requirement for the application of all the current laws is originality. Even if it is not straightforward to find a precise and applicable definition of originality, in the EU it has been commonly considered as such when the work is the reflection of the author's personality [151]; in the US, on the other hand, it can be interpreted as a minimum requiring evidence of a human (intellectual) creativity [213]. It is questionable to say that computer-generated artworks are the result of the personality of someone - or something - leaving the works unprotected. As a confirmation of this, The Compendium of U.S. Copyright Office Practices establishes that it will not register works produced by a machine or mere mechanical process that operates randomly or automatically without any creative input or intervention from a human author (see Article 313.2), citing, as example, a list of mechanical activities that are the exact opposite to those performed by generative deep learning, and the ones that might be reasonably considered as creative [483]. Spain, Germany, and Australia have formulated a similar criterion, establishing that only works created by humans can be protected by copyright.

On the contrary, an example of a law article in favor of protection for machine-generated artworks is Section 9(3) of the British Copyright, Designs and Patents Act. It states that in case of a literary, dramatic, musical, or artistic work that is computer-generated the author shall be taken to be the person by whom the arrangements necessary for the creation of the work are undertaken (the same criterion is also considered by Ireland, New Zealand, Hong Kong, South Africa, and India). This section has been the subject of an intense debate [61]. There is general agreement that for contemporary machine-generated artworks is difficult, though not impossible, to always find a person who provides *necessary* arrangements [609].

Policy Suggestions

Even if most current laws do not contemplate machine-generated works for copyright protection, the matter of right attribution has been widely discussed also in terms of ethical implications.

The position of not assigning copyright in machine-generated works may appear to be convenient at first; indeed, it does not require any changes. It might also help preserve the centrality of human authorship in copyright law [438] and stress the importance of what an author should be versus what an author should do [136]. Another more practical reason is that a work should receive copyright protection only if an *author* exists; but to be considered so, the work must include a meaning or a message they wish to convey, and this cannot happen if no one can predict the output of the program [66] as in deep learning models [220]. Finally, placing computer-generated works in the public domain can help preserve the centrality of humans in creative fields, since protection would be guaranteed only to work with an intellectual human contribution [483].

At the same time, there are also strong reasons against leaving AI-generated works unprotected. First, though consistent with the traditional concept of an author as a person, denying protection is inconsistent with the historically flexible interpretation and application of copyright laws as technology has developed. AI-generated products should also be evaluated following this flexible interpretation [86]. However, the best motivation for the allocation of ownership interests to someone is that the person should be incentivized not for the ideation and creation of the work in itself, but for its public promotion and for making it possible for the computer to create the work (by writing it, training it or instructing it [444]). If the law considers machine-generated work as incapable of being owned because of the lack of a human author, there will be limited incentives for creating them and making them public. On the contrary, this might lead to potentially malicious behavior, e.g., the person who used the algorithm to generate the work might be tempted to lie about the way it was created or change it to be considered its author [554]. Finally, the idea that this would mean incentivizing the proliferation of arts and articles of poor quality, penalizing the role of human artists and journalists [214] does not convince us completely. If the protection of computer-generated works translates into a larger number of mediocre works, then it will be easier for humans to produce works of higher quality and stand out. In addition, even if the current copyright laws were thought to regulate the scarcity of products created by humans [299] and not the abundance (of machine-generated products), leaving all of them in the public domain could cause more damage to human authors. The possibility of using them for free may persuade clients to do so, even if human artworks are qualitatively better[5].

It is interesting to note that the European Parliament seems to agree with this line of thought. In a recent Resolution, it has proposed to allocate rights to those who have prepared and published the work lawfully. While the Resolution was not embraced by the European Commission, its conclusion seems reasonable and can be reached in multiple ways. We can identify three main individuals involved in the process: the programmer, the person who provides necessary arrangements, and the user. Notice that even if we consider them separately, they are often the same person. Other works have proposed multiple actors [374], but we believe these three to be the most involved in the final production. The programmer is just the person who has written the code for the machine, in terms of both training and generation; the person who provides necessary arrangements can be, for example, the individual who provides instructions for the desired output, or information about the work the machine has to generate; the user is the person that, legitimately (because, for instance, the individual is the owner of the machine or has acquired it with a license), ultimately runs the machine and asks it to generate an artwork. In our opinion (but also according to the European Parliament and many others [716, 56, 554]), the rights should be allocated to the user, who can be considered an alter ego of the "person who prepares and publishes a work lawfully" (even if also the person who provides the necessary arrangements can be seen under the definition of who prepares the work). Although it can seem counterintuitive, there are different ways to reach this conclusion.

Various researchers have suggested an analogy with the ownership of economic rights of software produced by an employee [56, 716], the so-called work-made-for-hire doctrine. The employer is entitled to all economic rights of an employee's computer program if its creation is part of the scope of their employment or is commissioned by the employer. Similarly, the user is the person who lawfully causes the creation of the work. It is possible to say that the user has *employed* the computer for their creative endeavors. In this way, rights can be allocated, thus preventing works from falling into the public domain regardless of the extent of human creative contribution. Indeed, we assume the user to be the owner of the program, possibly because they are also its developers, or they have acquired it from the developers or the developers are their employees, or they have acquired it via license. Either way, there is no need for an extra economic incentive for the developers

[5] Note that also the art industry feels art lovers would always prefer handmade arts and crafts [73].

(they have already been paid or have chosen to release the product freely) or for the owners (they have already been paid for the license or have chosen to release the product freely). On the contrary, the user has paid to use the generative program (thus is the person who should be rewarded) or can use it freely because of a particular license. In general, we believe that even if the generative program has been published and is accessible by millions of end users, the Terms of Service written by the publisher should regulate this kind of problem. We suggest future publishers combine their generators with Terms of Service to identify the owner of the generated products (and the associated terms).

Another reason in favor of users is that they are in the same position as traditional authors. They take the initial steps that bring machine-generated work into the marketplace and its exterior form [152, 554]. Since society has an interest in making these works available to the public, the most effective solution is to incentivize users to make them available and accessible to others.

We can also reach the same conclusion by elimination. The programmer is responsible for the machine's creative abilities and, for other kinds of AI (e.g., rule-based systems), it might seem enough to establish the originality requirement - and therefore the ownership - in the programmer [181]. However, in the case of generative deep learning, they merely create the potentiality for generating the output, but not its actuality [554]. It would be like trying to assign copyright to the painter's master, instead of the painter, or to claim that a knife manufacturer is more responsible for murder than the person who wielded the knife [519]. The person who provides necessary arrangements can be difficult to identify, and sometimes the generative model may not have such a person associated with it, due to the complexity of deciding which are necessary arrangements and which are just useful arrangements [195]. For example, let us consider Botnik and its creative keyboard[6]. When we open it, it starts with *John Keats* as the source, and it starts suggesting words according to John Keats' texts on which a neural network was previously trained. Then, one can constantly select the word in the first position between suggestions, composing a new and hopefully creative text. Notice that we could choose the word among different options, and this selection would mean that we are recognized as the authors, but we are not. Now, which shall be considered the necessary arrangements? The only two things we have done as users were to enter a website and compulsively click on the first suggested word. It is not enough to consider our actions as necessary arrangements. Such arrangements can only be the ones performed by those

[6] `https://botnik.org/apps/writer/` [Accessed October 21, 2024].

who loaded John Keats' poems and trained the network; or maybe those made by those who decided that the preset source should be John Keats'. But no poems would arise from the creative keyboard without our simple operations, and it does not seem reasonable to leave, a priori, the ownership of this kind of machine-generated work to someone who was not involved in the materialization of the work - that is, what the law shall protect. In these cases, it does not seem reasonable to assign the rights to somebody who has provided the necessary arrangements; though there can be other cases in which it might seem the right choice, it would be better to have a rule with the most general applicability. This suggests us to discard the person who provides the necessary arrangements. On the contrary, allocating rights to the user seems not to have any particular flaws; they may not have provided any creative contribution, but, as explained above, this does not seem a valid argument.

Finally, an additional consideration about copyright allocation must be done. One of the most explored creative fields by AI researchers is videogame design. This typically concerns the use of generated images [650], characters [320] or soundtracks[7] inside games. In these cases, all the conclusions drawn until now are still valid, and no additional considerations are required. However, a growing application is procedural content generation, where the game scenarios are dynamically generated during the game [401]. Although this task is technically similar to image generation with the additional complexity of dynamic adaptation and complexity growth, additional considerations about copyright allocation are needed. In procedural content generation, identifying the user is not immediate. Naturally, there is the player, i.e., the *game* user; but the copyright allocation concerns the *generative model* user. In this case, the algorithm that generates the game content is not directly used by a person but by the game code, and therefore indirectly by the game programmer, who has employed the generative deep learning techniques to generate content not statically, but dynamically. By considering the programmer of the game code as the user of the generative model, the conclusion drawn during this section should remain generally applicable [68].

[7] https://cordis.europa.eu/article/id/421438-ai-composers-create-music-for-video-games [Accessed October 21, 2024].

7.3.3 Foundation Models and Possible Interpretation Under Copyright Law[8]

The last issue is how the generative model can be interpreted under copyright laws. We believe that a viable solution to study it is to link generative deep learning with information theory [135]. In fact, generative models trained with self-supervised learning, i.e., by maximizing the likelihood of training data, as commonly done for foundation models [59] (see Section 2.1), can be seen as a form of *(lossy or lossless) compression* [417]. From this perspective, the training algorithm plays the role of the compression algorithm; the inference (feed-forward) algorithm is the de-compression algorithm (with the input passed to the model working as a decoding key); and the model's weights represent the compressed version of the training set.

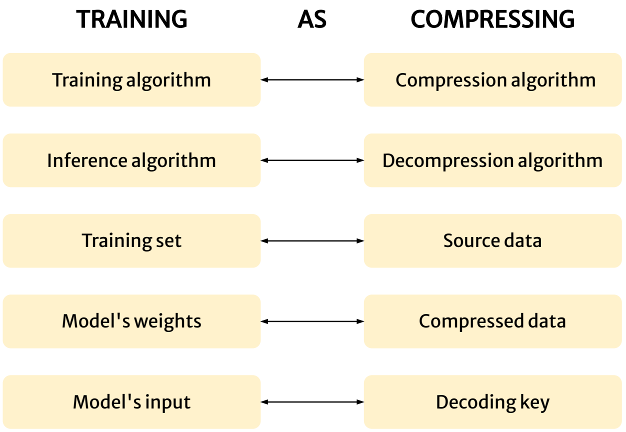


Figure 7.1: The *training-as-compressing* perspective. The training set is compressed into the model's weights via a training algorithm; the source data can be retrieved using the appropriate model's input.

Deletang et al. [150] discuss how a language model can implement a lossless compression process offline, i.e., through a fixed set of model parameters derived from training. We move a step further and claim the self-supervised learning used to train foundation models to be a lossy or lossless compression process, during which the whole training set is encrypted into the model's weights. This is demonstrated by the fact that the model can reproduce certain portions of training samples [93]. We suggest that the training optimizes the model's weights to be the best possible compressed version of

[8] The participation and presentation of the resulting paper at GenLaw'24 was supported by the ISA Doctoral Prize (ISA DP), offered by Istituto di Studi Avanzati, Alma Mater Studiorum Università di Bologna.

the training set, or more accurately, batches of it at a time. The analogies at the basis of the proposed *training-as-compressing* perspective are summarized in Figure 7.1. Building on this intuition, we envision the model's weights as either a reproduction or a derivative work of training data. This new interpretation opens up a series of practical consequences that can be relevant from the copyright perspective. In this section, we first revise self-supervised learning as a form of data compression. Then, we discuss how our *training-as-compressing* perspective allows for a specific understanding of the model's weights under copyright law. Finally, we discuss the legal implications of such a framing.

Training-as-Compressing and Information

The propensity of foundation models to memorize and subsequently replicate training data is a topic that has received considerable attention in scholarly literature, as evidenced by works such as [92]. In general, it is very hard to decompress every possible training sample perfectly and in its entirety, i.e., without any loss of information. Nonetheless, it has been shown that training samples can be retrieved [91], and more advanced techniques might lead to an even higher degree of "retrievability". The following experiment may help us understand this matter better.

In 1957, Noam Chomsky introduced the famous sentence "Colorless green ideas sleep furiously" to demonstrate the distinction between syntax and semantics [116]: the sentence is grammatically well-formed but semantically nonsensical. If a language model had learned the semantics of English, it should not generate a semantically nonsensical sentence, i.e., it should assign to semantically nonsensical words a small probability; if, on the contrary, such words were characterized by a large probability of being generated, then it would be very likely that the model had memorized them. To test this, we use the same quote from Chomsky and check the probability of each subsequent word given the previous ones under an LLM (in our case, LLaMa3-70B [436]). As depicted in Figure 7.2, the probability of *' green'* given *' "Colorless'* is 0.2, while for other colors like *' red'* or *' blue'* is 0.003. For all the subsequent words, i.e., *' ideas'*, *' sleep'*, and *' furiously'*, the probability is always greater than 0.9. The model has essentially memorized the quote into its weights, otherwise it would have never assigned such high probabilities to a semantically nonsensical sentence.

It is then straightforward to assert that training data are memorized in a compressed form. Consider again LLaMa3-70B [436], one of the largest models available at the time of writing. This model is pre-trained on more than 15 trillion tokens. Each token can have one out of 32000 values, thus

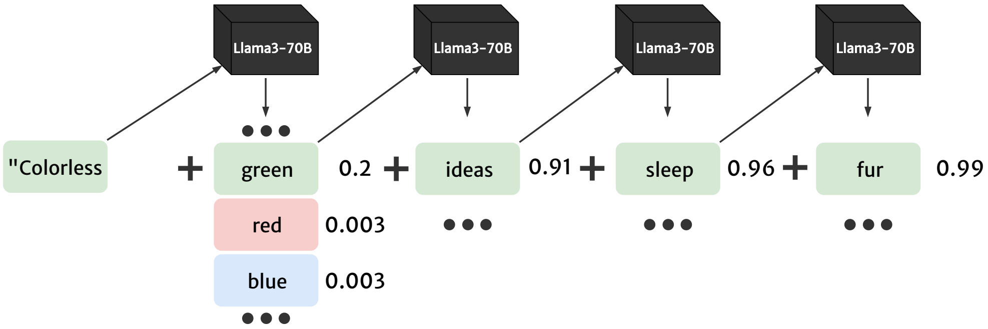


Figure 7.2: A thought experiment to confirm that LLMs memorize training data: even if the sentence is semantically nonsensical, the model assigns high probability to its tokens just because the sentence occurred in its training set.

requiring at least 15 bits to be represented. This means the training data require more than 225 trillion bits to be memorized. However, the model has 70 billion weights and uses half-precision floating points, thus it requires ∼1.1 trillion bits. With smaller models such as LLaMa3-8B, the compression ratio is even more astonishing and can possibly cause lossy compression [417].

Indeed, a foundation model, such as a Transformer-based LLM, consists of a neural network with weights $\mathbf{W}$. It models the conditional distribution $P(x_i|x_{i-k}\dots x_{i-1}, t; \mathbf{W})$, where $x = x_1 \dots x_N$ is the tokenized input to be modeled (and also the output to be predicted, which is the reason of the *self*-supervised learning), k is the size of the context window, and t is an additional input (such as a task specification). During training, the randomly initialized weights are iteratively adjusted through stochastic gradient descent (and its variants) as follows:

$$\mathbf{W} \leftarrow \mathbf{W} - \alpha \frac{1}{|\mathbb{X}|} \nabla_{\mathbf{W}} L(\mathbb{X}, \mathbf{W}), \tag{7.1}$$

where α is the learning rate and $L(\mathbb{X}, \mathbf{W})$ is the loss function computed on a batch of training samples $\mathbb{X}$. In particular, the objective is to maximize the log-likelihood of training data, therefore the loss is defined as:

$$L(\mathbb{X}, \mathbf{W}) = -\sum_{x,t \in \mathbb{X}} \sum_{i} P(x_i|x_{i-k}\dots x_{i-1}, t; \mathbf{W}). \tag{7.2}$$

In other words, the training phase aims to find the optimal values of the weights $\mathbf{W}$ such that given the input t (i.e., the decoding key) the model can autoregressively reconstruct x by only using the information stored into $\mathbf{W}$.

From an information theoretic perspective, such training data compression can be explained through the information bottleneck (IB) principle [652].

The IB principle applies when we aim to extract relevant information from an input variable $X \in \mathcal{X}$ about an output variable $Y \in \mathcal{Y}$. Given their joint distribution $p(X, Y)$, the relevant information is defined as the mutual information $I(X; Y)$. With $\hat{X}$ as the relevant part of X with respect to Y, the IB method aims to find the optimal $\hat{X} \in \hat{\mathcal{X}}$, i.e., the one that minimizes $I(X; \hat{X})$ (obtaining the simplest possible statistics) while maximizing $\beta I(\hat{X}; Y)$ (containing all the relevant information). Tishby and Zaslavsky [651] argued that neural networks could be interpreted under the theoretical framework of the IB principle. Indeed, neural networks learn to extract efficient representations of the relevant features $\hat{X}$ of the input X for predicting the output Y, given a finite sample of the joint distribution $p(X, Y)$. In the context of supervised learning, this means ignoring the irrelevant part of X by only selecting the one needed to predict Y. However, in self-supervised learning $Y \approx X$. It follows that $\hat{X}$ is the relevant part of X with respect to itself, so it is a *compressed* version of X.

These considerations suggest a *training-as-compressing* analogy, where the training algorithm plays the role of the compression algorithm; the inference (feed-forward) algorithm is the de-compression algorithm (with the input passed to the model working as a decoding key); and the model's weights represent the compressed version of the training set.

In addition to being used for data generation *as-is*, such a *pre-trained* model commonly represents a starting point for additional training: it can be fine-tuned for supporting downstream tasks [661] or inducing desired behaviors, e.g., to align it with human preferences [378]. The discussion above can be extended to fine-tuned models as well, with the only difference being that the source data would be both the training set and the pre-trained model's weights.

Training-as-Compressing and Copyright

The *training-as-compressing* perspective can shed new light on the open copyright issues related to generative modeling. While the software code responsible for the training and inference of a generative model can fall under copyright law as a computer program and the algorithmic method is a mathematical model and thus not protected [697], whether the model's weights can be protected or not is an open question. Indeed, if the model's weights represent a compressed version of the training set, and the training set is protected by copyright laws, then the weights are also subject to them. Assuming that the training set is protected in some ways (we will discuss it later), the weights can thus be seen as either a) a lossy or lossless compressed copy of it or b) a compressed version of a derivative work of it.

Seeing the weights as a mere compressed copy of the training set (not different from a zipped file) is seducing since the weights are meant to contain all the information necessary to reconstruct the original samples given a certain input (i.e., the decoding key). However, the final result is usually lossy, and the common scenario is that what we obtain after decompression is similar, but not exactly equal, to the original work. If the differences are not substantial, then it can still be considered a copy; however, it can also lead to a non-negligible modification or transformation of the training data. This second option seems to match the definition of derivative works.

This opens up a different perspective: what the weights represent might not be the original training set, but a new, derivative work (substantially different from, but still based on, the original) whose creation happens concurrently with weights' learning and whose only existence is due to the weights themselves. Nonetheless, a derivative work must still satisfy the originality requirement to be protected by copyright (see also Section 7.3.2). Whether or not the trainers' role in choosing data, algorithms, and parameters is sufficient for claiming authorship (and thus protection) of the model's weights is still an open question.

Until now, we have assumed that the training set is protected under copyright law. The whole training set can be protected as a database or a collective work, i.e., a collection of separate and independent works [374]. However, the collective work must constitute an intellectual creation because of the selection and arrangement of its content; the same criteria also apply to databases. One of the current trends for training foundation models seems to go in the opposite direction. Although a certain degree of data pre-processing is always present, the apparent tendency, at least in the early days of foundation models, has been to collect as much data as possible, for example, from the Web. This approach threatens the requirement of making a careful and original selection or arrangement. Moreover, training sets for specific domains used for fine-tuning are more likely to be eligible for protection as collective works. Still, this interpretation does not seem to cover all foundation models' training sets.

On the other hand, single training samples are often protected under copyright law [27]. Even though the training goal aims to compress batches of samples at a time, thus potentially leading to a compression that is optimal for a subset of works when considered together but not when considered separately, the single works can still be decompressed from the resulting model, at least in principle. This suggests that the model's weights can be interpreted as a copy (or a derivative work) of all the independent training samples, and not only of the training set as a whole.

Implications

Interpreting the model's weights as a copy or a derivative work of protected works leads to two crucial implications.

First, it provides a legal framework to understand them, removing the veil of uncertainty surrounding this issue. Although asserting copyright protection for weights as a derivative work presents challenges due to the absence of valid authorship [479], it is possible to safeguard them by viewing the file with the model's weights as a database. Indeed, they can be considered as a collection of floating point numbers that can be retrieved independently. Moreover, the significant investments required for obtaining them make the model's weights eligible for the *sui generis right* (thus providing certain rights to those who have invested in the database constitution independently from its copyright protection) [621]. In other words, the *sui generis right* can protect the investment; our copyright perspective can link the model's weights back to the training data, providing a new perspective over one of the several issues concerning the generative-AI supply chain [374]. The same considerations still hold in the case of a fine-tuned model. According to Lee et al. [374], this would be considered a derivative work of the pre-trained model (and also of the fine-tuning data). In other words, fine-tuning could be considered as nothing more than an additional step in the information processing chain. Again, the weights of the fine-tuned model would be eligible for the *sui generis right*. However, whether it qualifies for protection as a derivative work remains an open question, and the determination of valid (human) authorship can vary on a case-by-case basis.

Second, this type of interpretation provides a potential framework for works generated by the model. Indeed, decompressing the information from the model might be seen as producing a derivative work of the weights, thus a derivative work of a copy of a protected work or a derivative work of a derivative work of (a copy of) a protected work. Either way, this link between the output and the training data may help enforce their copyrights. It is worth noting that the EU text and data mining (TDM) exceptions as well as other comparable rules [191] apply for TDM purposes, such as training the model, therefore to the case of the creation of a copy or derivative work; however, they do not apply for further derivative works from the model. A similar consideration can also be drawn for the US fair use doctrine, which arguably applies to training a model on copyrighted data but is less likely when deployed to generate similar content that can threaten their market [268]. The main consequence is that authorization from the training set's rightsholders would be required (or else the reproduction or adaptation right would be triggered), allowing for potential requests for compensation from

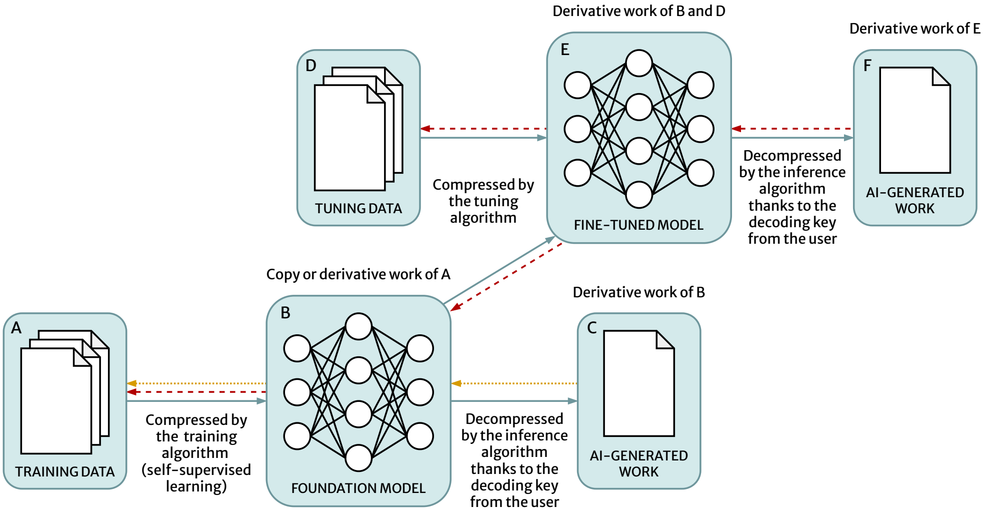

Derivative work of B and D
Derivative work of E
D
TUNING DATA
Compressed by the tuning algorithm
E
FINE-TUNED MODEL
Decompressed by the inference algorithm thanks to the decoding key from the user
F
AI-GENERATED WORK
Copy or derivative work of A
Derivative work of B
A
TRAINING DATA
Compressed by the training algorithm (self-supervised learning)
B
FOUNDATION MODEL
Decompressed by the inference algorithm thanks to the decoding key from the user
C
AI-GENERATED WORK

Figure 7.3: A schematic summary of the legal framework resulting from the *training-as-compressing* perspective. The blue arrows connect potentially protected entities to their copies or derivative works: the foundation model is a copy or a derivative work of the training data; fine-tuning can lead to a new derivative work of the foundation model and the tuning data; and an AI-generated work is a derivative work of either the foundation or the fine-tuned model. The yellow (dotted) and red (dashed) arrows directly link the AI-generated work back to training data and training and tuning data, respectively, only through steps requiring specific exceptions or authors' authorization.

original authors. In addition, generated works ought to respect the moral rights of the owners of training data, even when their economic rights have expired. The fact that the new derivative work is protected by copyright is an entirely different issue, already covered in Section 7.3.2. Crucially, these considerations also apply to synthetic data (i.e., AI-generated works) used as new training data for a different foundation model (e.g., [597]). Essentially, the chain of relationships between the resulting model-generated data and the original protected training data would be longer but still involve steps that trigger either the reproduction or adaptation rights. The overall conceptual framework based on the proposed *training-as-compressing* perspective is summarized in Figure 7.3.

8 Conclusions

8.1 Summary of Contributions

The latest developments in generative AI are attracting increasing interest from researchers and the general public due to the quality of their outputs. Generative models are used daily by humans to assist with various tasks, including those related to creativity, and many envision a future where they might replace humans in performing creative activities. In this thesis, we have analyzed whether these technologies can be truly considered creative, what technical solutions can be implemented to better align them with human creativity, and what implications might arise from having more creative AI.

We began with an in-depth introduction to creativity, generative AI, and the field of computational creativity (Chapter 1). Building on the most prominent theories of human creativity, we have critically introduced the main criteria to evaluate generative models and what technical solutions might address the missing aspects. In particular, reinforcement learning emerged as a strong candidate for better learning the skills required to perform seemingly creative processes, direct generation toward more creative products, and include a notion of environmental conditions into the model. This has led us to the definition of our main research questions: Can creativity enhance the design of RL algorithms? Can RL help develop more creative generative models?

Then, we have provided the necessary technical background (Chapter 2). First, we have formalized generative deep learning, detailing its main families of methods by describing how their training and inference algorithms work, as these are the most relevant aspects from a creativity perspective. Next, we have introduced the basic concepts of reinforcement learning, focusing specifically on policy-gradient methods and imagination-based reinforcement learning. Finally, we discussed the theoretical framework to use RL for generative modeling and we have detailed its most famous application in this area, i.e., RLHF.

In Chapter 3, we have provided a comprehensive overview of the current state of the art in all relevant topics. We have analyzed the generative deep learning families in terms of their most relevant variants and applications, finding that, even when applied to creative tasks, they do not explicitly target novel, surprising, and valuable solutions, arguably failing to achieve some of them. We have also discussed the literature on existing generative methods that optimize for creativity; while promising, they represent only a very small niche within the growing generative AI field. Then, we have reviewed the most relevant research in RL-related topics such as imagination, curiosity, and generalization, highlighting potential gaps that more creative solutions might partially address. We have also surveyed the literature on the state of the art, the opportunities, and the open challenges of using RL for generative modeling. We have focused on different reasons to adopt such a framework, i.e., to provide a suitable approach for domains that cannot be modeled through a well-defined and differentiable objective, to teach how to maximize a numerical property, or to align with human requirements and preferences that are not easily expressed in a mathematical form. Finally, we have covered the research on the impact of generative AI on society by considering the most prominent studies on AI exhibiting human capabilities, such as creativity, and the legal implications of generative modeling.

Once all the relevant topics were properly introduced and the related literature discussed, we have approached the first research question. In Section 4, we have studied whether dreaming can help RL agents better generalize, as recently suggested for humans. Starting from state-of-the-art imagination-based RL techniques, we have leveraged generative augmentations such as interpolation with random noise, Deepdream, and value-expectation maximization to transform standard, predicted trajectories into more dream-like experiences for fully training or fine-tuning the agent. We have evaluated generalization capabilities through ProcGen environments in four different low-resource scenarios. We have found that imagination-based RL is far from achieving competitive generalization performances when not trained for a large number of timesteps, and our transformations, while effective at the beginning of the training, do not provide meaningful improvements over classic imagination-based RL in the long run.

Then, Chapter 5 addresses the second research question. In particular, we have developed a new context-based score to evaluate value and originality underpinned by mutual information. Given a fixed input, this score can measure to which extent the output is unexpected for a given reference model (originality) as well as how much it is appropriate and related to the input (value). By leveraging reinforcement learning as an effective way to maximize the score, we have proposed various training methods with different normalization schemes to address the complexity of this multi-objective score. We have validated them into two different domains: poetry generation and mathematical problem resolution. With respect to the first domain, a qualitative analysis of generated poems suggests our method, especially the version based on DPO, increases the diversity of solutions while maintaining high quality effectively. As far as the second domain is concerned, we have found that our methods have no real shortcomings and that normalizing the score can provide significant improvements in response accuracy.

Since our RL-based strategy mainly works at learning time, we have also explored orthogonal solutions to better simulate creativity at inference time. First, we have examined whether our creativity score could be used directly at inference time (Section 5.2). We have developed a contextual learning approach where a small number of candidate outputs are produced and ordered based on the score, and then we have asked the generative model to produce a better solution. Evaluation on 15 creativity-related BIG-Bench tasks has revealed that our in-context method does not consistently improve performances over classic sampling. Then, in Chapter 6, we have proposed two new sampling strategies. For the former, we have explored a different solution that directly works at the probability distribution level. Based on theoretical and practical considerations over the standard sampling schemes, we have suggested the use of the discrete derivative of the ordered probabilities as a way to limit the sampling to relevant tokens or to push sampling toward less probable but still desirable tokens. We have developed *DiffSampling*, a family of sampling schemes, and we have demonstrated its efficacy in three different use cases: mathematical problem resolution, extreme summarization, and the divergent association task. Finally, we have presented Creative Beam Search, a generate-and-test sampling scheme to better simulate both the response generation and response validation steps that typically occur in human creative processes. This technique leverages diverse beam search to produce a certain number of response candidates that maximize diversity and LLM-as-a-Judge to identify the best output among the candidates. Our qualitative experiments showed that, on average, Creative Beam Search is viewed as more creative than traditional methods by potential end-users. We have also found that self-evaluation leads to an increase in terms of quality according to human evaluators.

Finally, Chapter 7 contains an in-depth discussion on social and practical issues arising from the use of generative AI for creative purposes. First, we have considered whether current foundation models are creative, their limitations, and the corresponding societal implications. We have then analyzed whether current foundation models can be entitled to agency, a relevant property for creativity, and what can happen to human agency when creatively collaborating with AI. Finally, we have examined how current copyright laws can manage the complexity of generative AI in terms of protecting human- and machine-generated artworks. Additionally, we have focused on the position of the generative model itself, arguing that it can be seen as a compression of training data and we have analyzed the legal consequences of this analogy.

8.2 Limitations

This thesis presents a series of methods and analyses that we believe can represent the basis for future theoretical and practical improvements of creative AI technologies. However, it is possible to identify a series of limitations of this work.

The first limitation concerns the definition of creativity. While we have found a certain convergence with some of the most prominent theories, there is still no universal agreement on what creativity really is and how it can be properly defined. We built our work on Rhodes' four perspectives of creativity, i.e., product, process, press, and person [535]. For each of them, we have considered the most relevant definitions to expand them into sub-definitions and criteria for assessment. Nonetheless, we may have overlooked some other perspectives or some of their key components. Either way, our analyses have been based on concepts extensively argued to be relevant for creativity. At worst, our building blocks were not as broad as possible, but they were still relevant for this investigation in our opinion. Our study of creativity is also focused on a "western" perspective. For example, our analysis of copyright and generative AI only concerns US and EU laws and is indeed a limitation of this work.

As far as the experiments from Chapter 4 are concerned, the main problem lies in the resource requirements. Our experiments on limited-resource scenarios showed that the performance of imagination-based RL dramatically drops with shorter training time. Given that imagination in RL is meant to free agent learning from the need of interacting with the environment, and that our techniques should foster generalization over collected data, requiring the availability of massive data is indeed a limitation. Moreover, the need for a correct world model introduces additional training complexity. In the considered settings, imagination with and without our generative augmentations has shown a tendency to overfit. More work is needed to investigate and explain these results; at the moment we do not have obvious explanations, except for the limited training data.

Regarding Chapter 5, the proposed score is theoretically sound and the results obtained are encouraging, but we still need more experiments to validate our method. Indeed, we have aimed to define a conceptual framework that can be adapted to any possible domain; how to adapt it for other tasks, however, is not as “simple” as it is for language modeling. Our score seems to easily fit with autoregressive models; still, how to compute the posterior probability of the input given the output for music or image generation is an open question. In addition, finding the optimal way to numerically balance the two parts of the score (i.e., value and originality) is not straightforward. The solution based on DPO seems to address this issue, but it is more computationally intensive than the RL one (as it requires generating multiple candidates for the same input) and less flexible, possibly preventing additional, future extensions.

Finally, it is possible to identify potential limitations also in the proposed sampling strategies. The score-based contextual learning approach has a higher resource consumption, and its effectiveness depends on the model considered and the task at hand; moreover, its application is limited to language modeling. On the other hand, *DiffSampling* might be applied to different domains as well, but we have only experimented with LLMs and with only a subset of possible tasks. Similarly, we have evaluated Creative Beam Search on a single task with limited resources and with a fixed prompt structure. In addition, Creative Beam Search possesses the limitation of the underlying technique that is at its basis: diverse beam search uses Hamming diversity, which only considers differences at the same time step, potentially leading to overly similar sequences due to minor misalignments such as initial spacing.

In general, evaluating the creativity of either a human or artificial agent is not straightforward. Given the lack of a unique definition of creativity, it is difficult to design a formal way for assessing it; in addition, creativity is a capability orthogonal to specific tasks, thus different tasks might require different evaluation methods. In all our experiments, we have tried to be as objective as possible, using quantitative indicators whenever they were available, and performing a qualitative analysis otherwise.

8.3 Future Work

This thesis provides some fundamental starting points for developing and analyzing creativity with and for generative AI.

First, the analysis of creativity and generative modeling can be expanded to include different domains and other philosophical and psychological theories, and, in the future, incorporate new developments in AI. While this can in theory be done from a merely technological perspective, we believe that consistently considering the societal, ethical, and legal issues together with other potential cross-disciplinary aspects is crucial, especially given the increasingly significant impact of these technologies.

Next, with respect to imagination-based RL, alternative technical solutions can be considered, e.g., an interesting question is whether the adoption of Transformers for the world model can simplify the process of learning the dynamics of the environment or improve the dream-like trajectory generation. We also plan to apply our creativity score to other tasks and domains, studying how to adapt our framework to prompt-based music and image generation. Instead of merely using it for fine-tuning, we also plan to test whether it can be helpful to build an LLM *ex-novo*. As far as the sampling schemes are considered, future work will include our creativity score at inference time in different ways, e.g., to evaluate multiple responses or to iteratively refine the output. We also plan to expand the breadth of experiments on *DiffSampling*, and especially to test whether merging these techniques together (with and without a model fine-tuned with our creativity score) can help obtain more creative products.

Intrinsic motivation plays an important role in human creativity. The investigation of curiosity-driven RL approaches to provide generative models with simulated motivation is very promising. Finally, this work focuses on single-agent scenarios. An evolution to multi-agent ones appears as natural in the coming years.

In conclusion, we believe we are at the dawn of creative AI technologies; whether AI will achieve human creativity is a different, almost impossible, question. But, for sure, the coming years will definitely be exciting.

8.4 Broader Impact

In various sections of this thesis, we have discussed how creativity is a personal and social act and how artificial agents are not persons or social agents. Nonetheless, we believe that computational creativity is worth studying: there are good reasons for not being scared of machines able to compose or paint.

The fear of potential replacement, also fostered by popular films and novels, should not be seen as a real danger. As already discussed in Section 1.2, an AI system always involves one or more humans to shape and influence its production. Humans indicate the creativity direction, select the best outcomes, interact with the machine, develop, train, and use it. It is important to remember that an artwork does not draw attention or threaten a potential market simply because it exists. Almost everyone has tried, sooner or later, to write a book, compose a song, paint a picture, or shoot a remarkable photo. But only an incredibly small portion of aspiring artists get their work published or exhibited. Why? Because any artwork deserves its audience. This means that only an audience, which incidentally is made of humans, can decide the future of such creations. The mere fact that a work is machine-generated can be considered sufficient at first to claim attention [677], but in the long run, only the product quality will matter. If and when artificial creators become the norm, they will be judged according to the same criteria as humans. In this sense, the chance of artificially producing a countless number of works [214] is not frightening; it will always be possible for humans to stand out for the quality of their works.

In any case, this is just one possible scenario, and it is not the most desirable by researchers [490]. History can act as a guide in this respect. A breakthrough invention that deeply affected the creative industry in the recent past is photography. During its first years, the same fear of being replaced spread between painters [266]. However, photography just found its place in visual arts, without replacing the existing ones; a new form of art was born, and (human) artists re-established themselves in this new artistic landscape. Indeed, there was no longer a need to create an illusion of *true* reality, paving the way for new styles like surrealism, cubism, or conceptual arts [599].

The invention of photography led to new artistic opportunities, such as film cameras. Similarly, studying creativity in AI might lead to new forms of art (or styles) we cannot even imagine so far or might transform some of the existing ones in desirable ways: for instance, they might become more accessible and portable. More concretely, artistic machines can collaborate with humans, both as tools and as real partners. Humans can now obtain new artworks by providing the machine the right input [611]; the output of a machine can be of inspiration for human arts [9]; or the final product can emerge after a sequence of interactions, where the human continuously asks the machine to adjust or refine their work [348].

On the other hand, just thinking of computational creativity in terms of arts is a mistake. Creativity is also linked with innovation, science, problem-solving, marketing, and in general, any daily activity with more than one possible solution [65]. A machine able to perform a creative process may help discover new approaches to solve problems or formulate theorems [144]; by reasoning in a different and perhaps unexpected way, it may come out with novel and effective alternatives [289]. These alternatives might not be better in general but can inspire the human solver, lead to faster or cheaper (heuristic) solutions, or even help define new interesting problems.

Moreover, finding alternative, performing ways of solving a task can also mean identifying different ways to achieve, represent, and communicate that solution. It can mean finding ways to customize the communication itself [351]. Creative machines can therefore be better than non-creative ones in interacting with users and in explaining the results of their work.

Finally, this study of AI and creativity can help in understanding human creativity. In fact, any discovery and failure of computational creativity research can provide insights into human creativity as well [53]. Essentially, studying computational creativity is about studying humanity itself.

Bibliography

[1] M. Abdin, J. Aneja, H. Awadalla, A. Awadallah, A. A. Awan, N. Bach, A. Bahree, A. Bakhtiari, J. Bao, H. Behl, A. Benhaim, M. Bilenko, J. Bjorck, S. Bubeck, M. Cai, Q. Cai, V. Chaudhary, D. Chen, D. Chen, ..., and X. Zhou. Phi-3 technical report: A highly capable language model locally on your phone, 2024. arXiv:2404.14219 [cs.CL].

[2] A. Abid, A. Abdalla, A. Abid, D. Khan, A. Alfozan, and J. Zou. Gradio: Hassle-free sharing and testing of ML models in the wild, 2019. arXiv:1906.02569 [cs.LG].

[3] J. Achiam and S. Sastry. Surprise-based intrinsic motivation for deep reinforcement learning, 2017. arXiv:1703.01732 [cs.LG].

[4] R. Agarwal, M. C. Machado, P. S. Castro, and M. G. Bellemare. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. In *Proc. of the 9th International Conference on Learning Representations (ICLR'21)*, 2021.

[5] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, and C. Frank. MusicLM: Generating music from text, 2023. arXiv:2301.11325 [cs.SD].

[6] B. Agüera y Arcas. Do large language models understand us? *Daedalus*, 151(2):183–197, 2022.

[7] H. Akbari, L. Yuan, R. Qian, W.-H. Chuang, S.-F. Chang, Y. Cui, and B. Gong. VATT: Transformers for multimodal self-supervised learning from raw video, audio and text. In *Advances in Neural Information Processing Systems (NIPS'21)*, 2021.

[8] A. G. Aleinikov, S. Kackmeister, and R. Koenig. *Creating Creativity: 101 Definitions (what Webster Never Told You)*. Alden B. Dow Creativity Center Press, 2000.

[9] S. Ali and D. Parikh. Telling creative stories using generative visual aids. In *Proc. of the NIPS'21 Machine Learning for Creativity and Design Workshop*, 2021.

[10] T. M. Amabile. The social psychology of creativity: A componential conceptualization. *Journal of Personality and Social Psychology*, 45 (2):357–376, 1983.

[11] T. M. Amabile. *Creativity in Context*. Westview Press, 1996.

[12] P. Ammanabrolu and M. Hausknecht. Graph constrained reinforcement learning for natural language action spaces. In *Proc. of the 8th International Conference on Learning Representations (ICLR'20)*, 2020.

[13] T. Andrillon, Y. Nir, C. Cirelli, G. Tononi, and I. Fried. Single-neuron activity and eye movements during human REM sleep and awake vision. *Nature Communications*, 6(1):7884, 2015.

[14] J. Aneja, A. G. Schwing, J. Kautz, and A. Vahdat. A contrastive learning approach for training variational autoencoder priors. In *Advances in Neural Information Processing Systems (NIPS'21)*, 2021.

[15] Anthropic. Collective Constitutional AI: Aligning a Language Model with Public Input, 2023. `https://anthropic.com/news/collective-constitutional-ai-aligning-a-language-model-with-public-input` [Accessed October 25, 2024].

[16] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lucic, and C. Schmid. ViViT: A video vision transformer. In *Proc. of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV'21)*, 2021.

[17] C. D. Asay, A. Sloan, and D. Sobczak. Is Transformative Use Eating the World? *Boston College Law Review*, 61(3):905–970, 2020.

[18] S. R. Atance, J. V. Diez, O. Engkvist, S. Olsson, and R. Mercado. De novo drug design using reinforcement learning with graph-based deep generative models. *Journal of Chemical Information and Modeling*, 62 (20):4863–4872, 2022.

[19] A. Aubret, L. Matignon, and S. Hassas. A survey on intrinsic motivation in reinforcement learning, 2019. arXiv:1908.06976 [cs.LG].

[20] C. Babbage. Of the analytical engine. In *Passages from the Life of a Philosopher*, volume 3, pages 112–141. Longman, Green, Longman, Roberts, & Green, 1864.

[21] P. Bachman and D. Precup. Data generation as sequential decision making. In *Proc. of the 28th International Conference on Neural Information Processing Systems (NIPS'15)*, 2015.

[22] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *Proc. of the 3rd International Conference on Learning Representations (ICLR'15)*, 2015.

[23] D. Bahdanau, P. Brakel, K. Xu, A. Goyal, R. Lowe, J. Pineau, A. Courville, and Y. Bengio. An actor-critic algorithm for sequence prediction. In *Proc. of the 5th International Conference on Learning Representations (ICLR'17)*, 2017.

[24] A. Baheti, X. Lu, F. Brahman, R. L. Bras, M. Sap, and M. Riedl. Leftover lunch: Advantage-based offline reinforcement learning for language models. In *Proc. of the 12th International Conference on Learning Representations (ICLR'24)*, 2024.

[25] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, N. Joseph, S. Kadavath, J. Kernion, T. Conerly, S. El-Showk, N. Elhage, Z. Hatfield-Dodds, D. Hernandez, T. Hume, ..., and J. Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022. arXiv:2204.05862 [cs.CL].

[26] Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon, C. Chen, C. Olsson, C. Olah, D. Hernandez, D. Drain, D. Ganguli, D. Li, E. Tran-Johnson, E. Perez, ..., and J. Kaplan. Constitutional AI: Harmlessness from AI feedback, 2022. arXiv:2212.08073 [cs.CL].

[27] J. Bandy and N. Vincent. Addressing "documentation debt" in machine learning: A retrospective datasheet for bookcorpus. In *Proc. of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

[28] H. Bao, L. Dong, and F. Wei. BEiT: BERT pre-training of image transformers. In *Proc. of the 10th International Conference on Learning Representations (ICLR'22)*, 2022.

[29] F. Barron. The disposition toward originality. *Journal of Abnormal Psychology*, 51(3):478–485, 1955.

[30] S. Bartezzaghi. *Mettere al Mondo il Mondo*. Bompiani, 2021.

[31] A. Barto, M. Mirolli, and G. Baldassarre. Novelty or surprise? *Frontiers in Psychology*, 4:907:1–907:15, 2013.

[32] A. G. Barto. Intrinsic motivation and reinforcement learning. In *Intrinsically Motivated Learning in Natural and Artificial Systems*, pages 17–47. Springer, 2013.

[33] M. Batey and A. Furnham. Creativity, intelligence, and personality: A critical review of the scattered literature. *Genetic, Social, and General Psychology Monographs*, 132(4):355–429, 2006.

[34] R. E. Beaty, P. J. Silvia, E. C. Nusbaum, E. Jauk, and M. Benedek. The roles of associative and executive processes in creative cognition. *Memory & Cognition*, 42(7):1186–1197, 2014.

[35] G. A. Bekey. *Autonomous Robots.* The MIT Press, 2005.

[36] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47(1):253–279, 2013.

[37] L. Ben Allal, A. Lozhkov, and E. Bakouch. SmolLM - blazingly fast and remarkably powerful, 2024. `https://huggingface.co/blog/smollm` [Accessed October 25, 2024].

[38] E. M. Bender and A. Koller. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'20)*, 2020.

[39] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proc. of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT'21)*, page 610–623, 2021.

[40] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[41] W. Benjamin. *The Work of Art in the Age of Mechanical Reproduction.* Penguin Books Ltd, 2008.

[42] D. E. Berlyne. *Conflict, Arousal, and Curiosity.* McGraw-Hill, 1960.

[43] D. E. Berlyne. Curiosity and exploration. *Science*, 153(3731):25–33, 1966.

[44] D. E. Berlyne. *Aesthetics and Psychobiology.* Appleton-Century-Crofts, 1971.

[45] S. Berns and S. Colton. Bridging generative deep learning and computational creativity. In *Proc. of the 11th International Conference on Computational Creativity (ICCC'20)*, 2020.

[46] J. Betker, G. Goh, L. Jing, T. Brooks, J. Wang, L. Li, L. Ouyang, J. Zhuang, J. Lee, Y. Guo, W. Manassra, P. Dhariwal, C. Chu, Y. Jiao, and A. Ramesh. Improving image generation with better captions, 2024. `https://cdn.openai.com/papers/dall-e-3.pdf` [Accessed October 25, 2024].

[47] F. Betti, G. Ramponi, and M. Piccardi. Controlled text generation with adversarial learning. In *Proc. of the 13th International Conference on Natural Language Generation (INLG'20)*, 2020.

[48] M. Binz and E. Schulz. Using cognitive psychology to understand GPT-3. *PNAS*, 120(6):e2218523120, 2023.

[49] K. Black, M. Janner, Y. Du, I. Kostrikov, and S. Levine. Training diffusion models with reinforcement learning. In *ICML'23 Workshop on Efficient Systems for Foundation Models*, 2023.

[50] T. Blaschke, J. Arús-Pous, H. Chen, C. Margreitter, C. Tyrchan, O. Engkvist, K. Papadopoulos, and A. Patronov. REINVENT 2.0: An AI tool for de novo drug design. *Journal of Chemical Information and Modeling*, 60(12):5918–5922, 2020.

[51] T. Blau, L. Ott, and F. Ramos. Bayesian curiosity for efficient exploration in reinforcement learning, 2019. arXiv:1911.08701 [cs.LG].

[52] M. A. Boden. Autonomy and artificiality. In *The Philosophy of Artificial Life*, pages 95–107. Oxford University Press, 1996.

[53] M. A. Boden. *The Creative Mind: Myths and Mechanisms.* Routledge, 2003.

[54] M. A. Boden. Autonomy: What is it? *Biosystems*, 91(2):305–308, 2008.

[55] M. A. Boden. Computer models of creativity. *AI Magazine*, 30(3): 23–34, 2009.

[56] H. M. Bohlen. EU copyright protection of works created by artificial intelligence systems. Master's thesis, University of Bergen, 2017.

[57] F. Böhm, Y. Gao, C. M. Meyer, O. Shapira, I. Dagan, and I. Gurevych. Better rewards yield better summaries: Learning to summarise without references. In *Proc. of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*, 2019.

[58] T. Bolukbasi, K.-W. Chang, J. Zou, V. Saligrama, and A. Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Proc. of the 30th International Conference on Neural Information Processing Systems (NIPS'16)*, 2016.

[59] R. Bommasani, D. Hudson, E. Adeli, R. Altman, S. Arora, S. Arx, M. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, E. Brynjolfsson, S. Buch, D. Card, R. Castellon, N. Chatterji, A. Chen, K. Creel, J. Davis, D. Demszky, ..., and P. Liang. On the opportunities and risks of foundation models, 2021. arXiv:2108.07258 [cs.LG].

[60] E. Bonadio and L. McDonagh. Artificial intelligence as producer and consumer of copyright works: Evaluating the consequences of algorithmic creativity. *Intellectual Property Quarterly 2020*, 2:112–137, 2020.

[61] T. Bond and S. Blair. Artificial intelligence & copyright: Section 9(3) or authorship without an author. *Journal of Intellectual Property Law & Practice*, 14(6):423–423, 2019.

[62] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi, and N. Zeghidour. Audiolm: A language modeling approach to audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2523–2533, 2023.

[63] N. Bougie and R. Ichise. Skill-based curiosity for intrinsically motivated reinforcement learning. *Machine Learning*, 109:493–512, 2020.

[64] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio. Generating sentences from a continuous space. In *Proc. of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNNL'16)*, 2016.

[65] O. Bown. *Beyond the Creative Species.* The MIT Press, 2021.

[66] B. E. Boyden. Emergent works. *The Columbia Journal of Law & The Arts*, 3(39):377–394, 2016.

[67] H. Bradley, A. Dai, H. Teufel, J. Zhang, K. Oostermeijer, M. Bellagente, J. Clune, K. Stanley, G. Schott, and J. Lehman. Quality-Diversity through AI feedback. In *Proc. of the 12th International Conference on Learning Representations (ICLR'24)*, 2024.

[68] A. Bridy. Coding creativity: Copyright and the artificially intelligent author. *Stanford Technology Law Review*, 5:1–28, 2012.

[69] T. Broad, S. Berns, S. Colton, and M. Grierson. Active divergence with generative deep learning - a survey and taxonomy. In *Proc. of the 12th International Conference on Computational Creativity (ICCC'21)*, 2021.

[70] A. Brock, J. Donahue, and K. Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *Proc. of the 7th International Conference on Learning Representations (ICLR'19)*, 2018.

[71] T. Brooks, J. Hellsten, M. Aittala, T.-C. Wang, T. Aila, J. Lehtinen, M.-Y. Liu, A. Efros, and T. Karras. Generating long videos of dynamic scenes. In *Advances in Neural Information Processing Systems (NIPS'22)*, 2022.

[72] T. Brooks, B. Peebles, C. Homes, W. DePue, Y. Guo, L. Jing, D. Schnurr, J. Taylor, T. Luhman, E. Luhman, C. W. Y. Ng, R. Wang, and A. Ramesh. Video generation models as world simulators, 2024. `https://openai.com/research/video-generation-models-as-world-simulators` [Accessed October 25, 2024].

[73] A. Brown. Is artificial intelligence set to take over the art industry?, 2021. `https://www.forbes.com/sites/anniebrown/2021/09/06/is-artificial-intelligence-set-to-take-over-the-art-industry/?sh=78b774c33c50` [Accessed October 25, 2024].

[74] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, ..., and D. Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NIPS'20)*, 2020.

[75] J. Browning. Personhood and ai: Why large language models don't understand us. *AI & SOCIETY*, 39:2499–2506, 2023.

[76] J. S. Bruner. The conditions of creativity. In *Contemporary approaches to creative thinking: A symposium held at the University of Colorado*, pages 1–30. Atherton Press, 1962.

[77] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, et al. Sparks of artificial general intelligence: Early experiments with GPT-4, 2023. arXiv:2303.12712 [cs.CL].

[78] B. Bucher, K. Schmeckpeper, N. Matni, and K. Daniilidis. An adversarial objective for scalable exploration. In *Proc. of the 2021 International Conference on Intelligent Robots and Systems (IROS'21)*, 2021.

[79] L. Buesing, T. Weber, S. Racaniere, S. M. A. Eslami, D. Rezende, D. P. Reichert, F. Viola, F. Besse, K. Gregor, D. Hassabis, and D. Wierstra. Learning and querying fast generative models for reinforcement learning, 2018. arXiv:1802.03006 [cs.LG].

[80] R. C. Bunescu and O. O. Uduehi. Learning to surprise: A composer-audience architecture. In *Proc. of the 10th International Conference on Computational Creativity (ICCC'19)*, 2019.

[81] Y. Burda, R. Grosse, and R. Salakhutdinov. Importance weighted autoencoders. In *Proc. of the 4th International Conference on Learning Representations (ICLR'16)*, 2016.

[82] Y. Burda, H. Edwards, D. Pathak, A. Storkey, T. Darrell, and A. A. Efros. Large-scale study of curiosity-driven learning. In *Proc. of the 7th International Conference on Learning Representations (ICLR'19)*, 2019.

[83] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner. Understanding disentangling in β-VAE. In *Proc. of the NIPS'17 Workshop on Learning Disentangled Representations*, 2017.

[84] N. Burkart and M. F. Huber. A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, 70: 245–317, 2021.

[85] K. Burns. Computing the creativeness of amusing advertisements: A bayesian model of Burma-Shave's Muse. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 29:109–128, 2015.

[86] T. L. Butler. Can a computer be an author - copyright aspects of artificial intelligence. *Hastings Communications and Entertainment Law Journal*, 4(4), 1982.

[87] P. Butlin, R. Long, E. Elmoznino, Y. Bengio, J. Birch, A. Constant, G. Deane, S. M. Fleming, C. Frith, X. Ji, R. Kanai, C. Klein, G. Lindsay, M. Michel, L. Mudrik, M. A. K. Peters, E. Schwitzgebel, J. Simon, and R. VanRullen. Consciousness in artificial intelligence: Insights from the science of consciousness, 2023. arXiv:2308.08708 [cs.AI].

[88] M. Caccia, L. Caccia, W. Fedus, H. Larochelle, J. Pineau, and L. Charlin. Language GANs falling short. In *Proc. of the 8th International Conference on Learning Representations (ICLR'20)*, 2020.

[89] A. Calderwood, V. Qiu, K. I. Gero, and L. B. Chilton. How novelists use generative language models: An exploratory user study. In *Proc. of the IUI'20 Workshop on Human-AI Co-Creation with Generative Models*, 2020.

[90] A. Cardoso, T. Veale, and G. A. Wiggins. Converging on the divergent: The history (and future) of the international joint workshops in computational creativity. *AI Magazine*, 30(3):15, 2009.

[91] N. Carlini, F. Tramèr, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, Ú. Erlingsson, A. Oprea, and C. Raffel. Extracting training data from large language models. In *Proc. of the 30th USENIX Security Symposium (USENIX Security 21)*, 2021.

[92] N. Carlini, J. Hayes, M. Nasr, M. Jagielski, V. Sehwag, F. Tramèr, B. Balle, D. Ippolito, and E. Wallace. Extracting training data from diffusion models. In *Proc. of the 32nd USENIX Conference on Security Symposium (SEC'23)*, 2023.

[93] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramer, and C. Zhang. Quantifying memorization across neural language models. In *Proc. of the 11th International Conference on Learning Representations (ICLR'23)*, 2023.

[94] M. W. Carroll. Fixing fair use. *North Carolina Law Review*, 85, 2007.

[95] B. Casey and M. A. Lemley. You might be a robot. *Cornell Law Review*, 105(2):287, 2019.

[96] S. Casper, X. Davies, C. Shi, T. K. Gilbert, J. Scheurer, J. Rando, R. Freedman, T. Korbak, D. Lindner, P. Freire, T. Wang, S. Marks, C.-R. Segerie, M. Carroll, A. Peng, P. Christoffersen, M. Damani, S. Slocum, U. Anwar, ..., and D. Hadfield-Menell. Open problems and fundamental limitations of reinforcement learning from human feedback, 2023. *Transactions on Machine Learning Research.*

[97] C. Castelfranchi. Guarantees for autonomy in cognitive agent architecture. In *Proc. of the International Workshop on Agent Theories, Architectures, and Languages (ATAL'94)*, 1995.

[98] J. S. O. Ceron and P. S. Castro. Revisiting rainbow: Promoting more insightful and inclusive deep reinforcement learning research. In *Proc. of the 38th International Conference on Machine Learning*, 2021.

[99] A. Chaganty, S. Mussmann, and P. Liang. The price of debiasing automatic metrics in natural language evalaution. In *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL'18)*, 2018.

[100] T. Chakrabarty, V. Padmakumar, and H. He. Help me write a poem: Instruction tuning as a vehicle for collaborative poetry writing. In *Proc. of the AAAI'23 Workshop on Creative AI Across Modalities*, 2023.

[101] T. Chakrabarty, V. Padmakumar, F. Brahman, and S. Muresan. Creativity support in the age of large language models: An empirical study involving professional writers. In *Proc. of the 16th Conference on Creativity & Cognition (C&C'24)*, 2024.

[102] D. J. Chalmers. *The Conscious Mind: In Search of a Fundamental Theory.* Oxford University Press, 1996.

[103] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J. K. Su. This looks like that: Deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems (NIPS'19)*, 2019.

[104] C. Chen, Y.-F. Wu, J. Yoon, and S. Ahn. Transdreamer: Reinforcement learning with transformer world models. In *Proc. of the NIPS'21 Deep RL Workshop*, 2021.

[105] H. Chen and N. Ding. Probing the “creativity” of large language models: Can models produce divergent semantic association? In *Findings of the Association for Computational Linguistics (EMNLP'23)*, 2023.

[106] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, D. Luan, and I. Sutskever. Generative pretraining from pixels. In *Proc. of the 37th International Conference on Machine Learning (ICML'20)*, 2020.

[107] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, A. Ray, R. Puri, G. Krueger, M. Petrov, H. Khlaaf, G. Sastry, P. Mishkin, B. Chan, S. Gray, ..., and W. Zaremba. Evaluating large language models trained on code, 2021. arXiv:2107.03374 [cs.LG].

[108] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan. WaveGrad: Estimating gradients for waveform generation. In *Proc. of the 9th International Conference on Learning Representations (ICLR'21)*, 2021.

[109] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS'16)*, 2016.

[110] A. Chhabra, H. Askari, and P. Mohapatra. Revisiting zero-shot abstractive summarization in the era of large language models from the perspective of position bias. In *Proc. of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'24)*, 2024.

[111] C.-H. Chiang and H.-y. Lee. Can large language models be an alternative to human evaluations? In *Proc. of the 61st Annual Meeting of the Association for Computational Linguistics (ACL'23)*, 2023.

[112] R. Child, S. Gray, A. Radford, and I. Sutskever. Generating long sequences with sparse transformers, 2019. arXiv:1904.10509 [cs.LG].

[113] T. Chiou. Copyright lessons on machine learning: What impact on algorithmic art. *Journal of Intellectual Property, Information Technology and Electronic Commerce Law*, 10:398–412, 2019.

[114] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches.

In *Proc. of the 8th Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8)*, 2014.

[115] W. S. Cho, P. Zhang, Y. Zhang, X. Li, M. Galley, C. Brockett, M. Wang, and J. Gao. Towards coherent and cohesive long-form text generation. In *Proc. of the NAACL'19 Workshop on Narrative Understanding*, 2019.

[116] N. Chomsky. *Syntactic Structures*. Mouton and Co., 1957.

[117] L. Choshen, L. Fox, Z. Aizenbud, and O. Abend. On the weaknesses of reinforcement learning for neural machine translation. In *Proc. of the 8th International Conference on Learning Representations (ICLR'20)*, 2020.

[118] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NIPS'17)*, 2017.

[119] Christies. Is artificial intelligence set to become art's next medium?, 2018. `https://christies.com/stories/a-collaboration-between-two-artists-one-human-one-a-machine-0cd01f4e232f4279a525a446d60d4cd1` [Accessed October 25, 2024].

[120] E. Chu. Artistic influence GAN. In *Proc. of the NIPS'18 Workshop on Machine Learning for Creativity and Design*, 2018.

[121] K. Chua, R. Calandra, R. McAllister, and S. Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems (NIPS'18)*, 2018.

[122] J. Chung, C. Gulcehre, K. Cho, and Y. Bengiom. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Proc. of the NIPS'14 Deep Learning and Representation Learning Workshop*, 2014.

[123] E. Clark, Y. Ji, and N. A. Smith. Neural text generation in stories using entity representations as context. In *Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018.

[124] W. Clemens. Computer poetry's neglected debut. In *Futures Past: Twenty years of arts computing (CHArt'04)*, 2004.

[125] K. Cobbe, O. Klimov, C. Hesse, T. Kim, and J. Schulman. Quantifying generalization in reinforcement learning. In *Proc. of the 36th International Conference on Machine Learning (ICML'19)*, 2019.

[126] K. Cobbe, C. Hesse, J. Hilton, and J. Schulman. Leveraging procedural generation to benchmark reinforcement learning. In *Proc. of the 37th International Conference on Machine Learning (ICML'20)*, 2020.

[127] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman. Training verifiers to solve math word problems, 2021. arXiv:2110.14168 [cs.LG].

[128] H. Cohen. How to draw three people in a botanical garden. In *Proc. of the 7th AAAI National Conference on Artificial Intelligence (AAAI'88)*, 1988.

[129] P. R. Cohen and H. J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(2):213–261, 1990.

[130] S. Colton. Creativity versus the perception of creativity in computational systems. In *Proc. of the 2008 AAAI Spring Symposium*, 2008.

[131] S. Colton and G. A. Wiggins. Computational creativity: The final frontier? In *Proc. of the 20th European Conference on Artificial Intelligence (ECAI'12)*, 2012.

[132] E. Conti, V. Madhavan, F. P. Such, J. Lehman, K. O. Stanley, and J. Clune. Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents. In *Proc. of the 32nd International Conference on Neural Information Processing Systems (NIPS'18)*, 2018.

[133] D. Cope. Experiments in musical intelligence (EMI): Non-linear linguistic-based composition. *Interface*, 18:117–139, 1989.

[134] R. Coulom. Efficient selectivity and backup operators in monte-carlo tree search. In *Computers and Games*, pages 72–83. Springer Berlin Heidelberg, 2007.

[135] T. M. Cover. *Elements of Information Theory.* John Wiley & Sons, 1999.

[136] C. J. Craig and I. R. Kerr. The death of the AI author. *Osgoode Legal Studies Research Paper*, 2019.

[137] A. Creswell, M. Shanahan, and I. Higgins. Selection-inference: Exploiting large language models for interpretable logical reasoning. In *Proc. of the 11th International Conference on Learning Representations (ICLR'23)*, 2023.

[138] E. N. Crothers, N. Japkowicz, and H. L. Viktor. Machine-generated text: A comprehensive survey of threat models and detection methods. *IEEE Access*, 11:70977–71002, 2023.

[139] K. Crowson, S. Biderman, D. Kornis, D. Stander, E. Hallahan, L. Castricato, and E. Raff. VQGAN-CLIP: Open domain image generation and editing with natural language guidance. In *Proc. of the 17th European Conference on Computer Vision (ECCV'22)*, 2022.

[140] M. Csikszentmihalyi. Society, culture, and person: A systems view of creativity. In *The Systems Model of Creativity*, pages 47–71. Springer, 2014.

[141] Danish Contractor and C. M. Ferrandis. The BigScience RAIL License, 2022. `https://bigscience.huggingface.co/blog/the-bigscience-rail-license` [Accessed October 25, 2024].

[142] S. Dathathri, A. Madotto, J. Lan, J. Hung, E. Frank, P. Molino, J. Yosinski, and R. Liu. Plug and play language models: A simple approach to controlled text generation. In *Proc. of the 8th International Conference on Learning Representations (ICLR'20)*, 2020.

[143] D. Davidson. Actions, reasons, and causes. *The Journal of Philosophy*, 60(23):685–700, 1963.

[144] A. Davies, P. Velickovic, L. Buesing, S. Blackwell, D. Zheng, N. Tomasev, R. Tanburn, P. Battaglia, C. Blunell, A. Juhasz, M. Lackenby, G. Williamson, D. Hassabis, and P. Kohli. Advancing mathematics by guiding human intuition with AI. *Nature*, 600:70–74, 2021.

[145] N. De Cao and T. Kipf. MolGAN: An implicit generative model for small molecular graphs. In *Proc. of the ICML'18 Workshop on Theoretical Foundations and Applications of Deep Generative Models*, 2018.

[146] E. L. Deci and R. M. Ryan. *Intrinsic Motivation and Self-Determination in Human Behavior*. Springer, 1985.

[147] T. Degris, M. White, and R. S. Sutton. Off-policy actor-critic. In *Proc. of the 29th International Conference on International Conference on Machine Learning (ICML'12)*, 2012.

[148] N. Dehouche. Plagiarism in the age of massive generative pre-trained transformers (GPT-3). *Ethics in Science and Environmental Politics*, 21:17–23, 2021.

[149] Y. Deldjoo, T. D. Noia, and F. A. Merra. A survey on adversarial recommender systems: From attack/defense strategies to generative adversarial networks. *ACM Computing Surveys*, 54(2):1–38, 2021.

[150] G. Deletang, A. Ruoss, P.-A. Duquenne, E. Catt, T. Genewein, C. Mattern, J. Grau-Moya, L. K. Wenliang, M. Aitchison, L. Orseau, M. Hutter, and J. Veness. Language modeling is compression. In *Proc. of the 12th International Conference on Learning Representations (ICLR'24)*, 2024.

[151] J.-M. Deltorn. Deep creations: Intellectual property and the automata. *Frontiers in Digital Humanities*, 4(3), 2017.

[152] R. C. Denicola. Ex machina: Copyright protection for computer-generated works. *Rutgers Law Review*, 69:251–287, 2016.

[153] A. Deshpande, V. Murahari, T. Rajpurohit, A. Kalyan, and K. Narasimhan. Toxicity in ChatGPT: Analyzing persona-assigned language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

[154] A. Deshpande, T. Rajpurohit, K. Narasimhan, and A. Kalyan. Anthropomorphization of AI: Opportunities and risks. In *Proc. of the Natural Legal Language Processing Workshop 2023*, 2023.

[155] T. Dettmers and L. Zettlemoyer. The case for 4-bit precision: k-bit inference scaling laws. In *Proc. of the 40th International Conference on Machine Learning (ICML'23)*, 2023.

[156] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

[157] P. Dhariwal and A. Q. Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems (NIPS'21)*, 2021.

[158] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever. Jukebox: A generative model for music, 2020. arXiv:2005.00341 [eess.AS].

[159] L. Ding, J. Zhang, J. Clune, L. Spector, and J. Lehman. Quality diversity through human feedback. In *Proc. of the NIPS'23 ALOE Workshop*, 2023.

[160] M. Ding, Z. Yang, W. Hong, W. Zheng, C. Zhou, D. Yin, J. Lin, X. Zou, Z. Shao, H. Yang, and J. Tang. CogView: Mastering text-to-image generation via transformers. In *Advances in Neural Information Processing Systems (NIPS'21)*, 2021.

[161] C. Donahue, J. McAuley, and M. Puckette. Adversarial audio synthesis. In *Proc. of the 7th International Conference on Learning Representations (ICLR'19)*, 2019.

[162] J. Donahue, P. Krahenbuhl, and T. Darrell. Adversarial feature learning. In *Proc. of the 5th International Conference on Learning Representations (ICLR'17)*, 2017.

[163] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang. MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In *Proc. of the 32nd AAAI Conference on Artificial Intelligence and 30th Innovative Applications of Artificial Intelligence Conference and 8th AAAI Symposium on Educational Advances in Artificial Intelligence*, 2018.

[164] Y. Dong, X. Jiang, Z. Jin, and G. Li. Self-collaboration code generation via ChatGPT. *ACM Transactions on Software Engineering and Methodology*, 33(7):189:1–38, 2024.

[165] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. of the 9th International Conference on Learning Representations (ICLR'21)*, 2021.

[166] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar,

P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, ..., and P. Florence. Palm-e: An embodied multimodal language model. In *Proc. of the 40th International Conference on Machine Learning (ICML'23)*, 2023.

[167] N. Du, Y. Huang, A. M. Dai, S. Tong, D. Lepikhin, Y. Xu, M. Krikun, Y. Zhou, A. W. Yu, O. Firat, B. Zoph, L. Fedus, M. P. Bosma, Z. Zhou, T. Wang, E. Wang, K. Webster, M. Pellat, K. Robinson, ..., and C. Cui. GLaM: Efficient scaling of language models with mixture-of-experts. In *Proc. of the 39th International Conference on Machine Learning (ICML'22)*, 2022.

[168] Y. Du, S. Li, A. Torralba, J. B. Tenenbaum, and I. Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Proc. of the 41st International Conference on Machine Learning (ICML'24)*, 2024.

[169] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, A. Goyal, A. Hartshorn, A. Yang, A. Mitra, A. Sravankumar, A. Korenev, A. Hinsvark, A. Rao, A. Zhang, ..., and Z. Zhao. The Llama 3 herd of models, 2024. arXiv:2407.21783 [cs.AI].

[170] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville. Adversarially learned inference. In *Proc. of the 5th International Conference on Learning Representations (ICLR'17)*, 2017.

[171] E. Eisenstein. *The Printing Press as an Agent of Change: Communications and Cultural Transformations in Early-Modern Europe.* Cambridge University Press, 1979.

[172] A. Elgammal and B. Saleh. Quantifying creativity in art networks. In *Proc. of the 6th International Conference on Computational Creativity (ICCC'15)*, 2015.

[173] A. Elgammal, B. Liu, M. Elhoseiny, and M. Mazzone. CAN: Creative adversarial networks, generating "art" by learning about styles and deviating from style norms. In *Proc. of the 8th International Conference on Computational Creativity (ICCC'17)*, 2017.

[174] B. Elizalde, S. Deshmukh, M. A. Ismail, and H. Wang. CLAP: Learning audio concepts from natural language supervision. In *Proc. of the*

2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'23), 2023.

[175] M. Emirbayer and A. Mische. What is agency? *American Journal of Sociology*, 103(4):962–1023, 1998.

[176] J. Engel, K. K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, and A. Roberts. GANSynth: Adversarial neural audio synthesis. In *Proc. of the 7th International Conference on Learning Representations (ICLR'19)*, 2019.

[177] S. M. A. Eslami, N. Heess, T. Weber, Y. Tassa, D. Szepesvari, K. Kavukcuoglu, and G. E. Hinton. Attend, infer, repeat: Fast scene understanding with generative models. In *Advances in Neural Information Processing Systems (NIPS'16)*, 2016.

[178] L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, S. Legg, and K. Kavukcuoglu. IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. In *Proc. of the 35th International Conference on Machine Learning (ICML'18)*, 2018.

[179] P. Esser, R. Rombach, and B. Ommer. Taming transformers for high-resolution image synthesis. In *Proc. of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'21)*, 2021.

[180] Y. Fan, O. Watkins, Y. Du, H. Liu, M. Ryu, C. Boutilier, P. Abbeel, M. Ghavamzadeh, K. Lee, and K. Lee. DPOK: Reinforcement learning for fine-tuning text-to-image diffusion models. In *Proc. of the 37th International Conference on Neural Information Processing Systems (NIPS'23)*, 2023.

[181] E. H. Farr. Copyrightability of computer-created works. *Rutgers Computer and Technology Law Journal*, 15(1):63–80, 1989.

[182] M. B. Fazi. Can a machine think (anything new)? automation beyond simulation. *AI & SOCIETY*, 34(4):813–824, 2019.

[183] B. Fecher, M. Hebing, M. Laufer, J. Pohle, and F. Sofsky. Friend or foe? exploring the implications of large language models on the science system, 2023. *AI & SOCIETY*. Accepted for publication.

[184] W. Fedus, I. Goodfellow, and A. M. Dai. MaskGAN: Better text generation via filling in the ______. In *Proc. of the 6th International Conference on Learning Representations (ICLR'18)*, 2018.

[185] M. Feffer, H. Heidari, and Z. C. Lipton. Moral machine or tyranny of the majority? In *Proc. of the 37th AAAI Conference on Artificial Intelligence (AAAI'23)*, 2023.

[186] N. Fei, Z. Lu, Y. Gao, G. Yang, Y. Huo, J. Wen, H. Lu, R. Song, X. Gao, T. Xiang, H. Sun, and J.-R. Wen. Towards artificial general intelligence via a multimodal foundation model. *Nature Communications*, 13(1):3094, 2022.

[187] G. J. Feist. A meta-analysis of personality in scientific and artistic creativity. *Personality and Social Psychology Review*, 2(4):290–309, 1998.

[188] P. Fernandes, J. N. Correia, and P. Machado. Evolutionary latent space exploration of generative adversarial networks. In *Proc. of the 2020 International Conference on the Applications of Evolutionary Computation (Part of EvoStar'20)*, 2020.

[189] P. Fernandes, A. Madaan, E. Liu, A. Farinhas, P. H. Martins, A. Bertsch, J. G. C. de Souza, S. Zhou, T. Wu, G. Neubig, and A. F. T. Martins. Bridging the gap: A survey on integrating (human) feedback for natural language generation. *Transactions of the Association for Computational Linguistics*, 11:1643–1668, 2023.

[190] C. Fernando, S. M. A. Eslami, J.-B. Alayrac, P. Mirowski, D. Banarse, and S. Osindero. Generative art using neural visual grammars and dual encoders, 2021. arXiv:2105.00162 [cs.AI].

[191] S. M. Fiil-Flynn, B. Butler, M. Carroll, O. Cohen-Sasson, C. Craig, L. Guibault, P. Jaszi, B. J. Jütte, A. Katz, J. P. Quintais, T. Margoni, A. R. de Souza, M. Sag, R. Samberg, L. Schirru, M. Senftleben, O. Tur-Sinai, and J. L. Contreras. Legal reform to enhance global text and data mining research. *Science*, 378(6623):951–953, 2022.

[192] A. Flood. Robot artist to perform AI generated poetry in response to Dante, 2021. `https://www.theguardian.com/books/2021/nov/26/robot-artist-to-perform-ai-generated-poetry-in-response-to-dante` [Accessed October 25, 2024].

[193] L. Floridi and M. Chiriatti. GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30(4):681–694, 2020.

[194] D. Foster. *Generative Deep Learning.* O'Reilly, 2023.

[195] G. Franceschelli. *I, Artist. Opere d'arte e intelligenza artificiale. Il curioso caso del diritto d'autore*. Ventura Edizioni, 2019.

[196] M. Frank, J. Leitner, M. Stollenga, A. Förster, and J. Schmidhuber. Curiosity driven reinforcement learning for motion planning on humanoid. *Frontiers in Neurorobotics*, 7(25):1–15, 2014.

[197] P. Fyfe. How to cheat on your final paper: Assigning ai for student writing. *AI & SOCIETY*, 38(4):1395–1405, 2023.

[198] A. Ganguly and S. W. F. Earp. An introduction to variational inference, 2021. arXiv:2108.13083 [cs.LG].

[199] Y. Gao, C. M. Meyer, M. Mesgar, and I. Gurevych. Reward learning for efficient reinforcement learning in extractive document summarisation. In *Proc. of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19)*, 2019.

[200] L. Gatys, A. Ecker, and M. Bethge. A neural algorithm of artistic style. *Journal of Vision*, 16(12):326, 2016.

[201] L. A. Gatys, A. S. Ecker, M. Bethge, A. Hertzmann, and E. Shechtman. Controlling perceptual factors in neural style transfer. In *Proc. of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*, 2017.

[202] T. Gaudin, A. Nigam, and A. Aspuru-Guzik. Exploring the chemical space without bias: data-free molecule generation with DQN and SELFIES. In *Proc. of the 2nd NIPS'19 Workshop on Machine Learning and the Physical Sciences*, 2019.

[203] B. Gaut. Creativity and imagination. In *The Creation of Art: New Essays in Philosophical Aesthetics*, pages 148–173. Cambridge University Press, 2003.

[204] B. Gaut. Creativity and skill. In *The Systems Model of Creativity*, pages 83–103. Brill, 2009.

[205] B. Gaut. The philosophy of creativity. *Philosophy Compass*, 5(12): 1034–1046, 2010.

[206] T. Gebru, J. Morgenstern, B. Vecchione, J. W. Vaughan, H. Wallach, H. Daumé III, and K. Crawford. Datasheets for datasets. In *Proc. of the 5th Workshop on Fairness, Accountability, and Transparency in Machine Learning (FAccT'18)*, 2018.

[207] C. Geiger, G. F. Frosio, and O. Bulayenko. The exception for Text and Data Mining (TDM) in the proposed Directive on Copyright in the Digital Single Market - legal aspects, 2018. [Requested by the JURI Committee] (European Parliament, February 2018).

[208] C. Geiger, G. F. Frosio, and O. Bulayenko. Text and data mining in the proposed copyright reform: Making the EU ready for an age of big data? *IIC - International Review of Intellectual Property and Competition Law*, 49:814–844, 2018.

[209] Gemini Team and Google. Gemini: A family of highly capable multimodal models, 2023. arXiv:2312.11805 [cs.CL].

[210] Gemini Team and Google. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024. `https://storage.googleapis.com/deepmind-media/gemini/gemini_v1_5_report.pdf` [Accessed October 25, 2024].

[211] J. Gero. Computational models of innovative and creative design processes. *Technological Forecasting and Social Change*, 64(2-3):183–196, 2000.

[212] K. I. Gero, V. Liu, and L. Chilton. Sparks: Inspiration for science writing using language models. In *Proc. of the 2022 Designing Interactive Systems Conference (DIS'22)*, 2022.

[213] D. J. Gervais. Feist goes global: A comparative analysis of the notion of originality in copyright law. *Journal of the Copyright Society of the U.S.A.*, 49:949–981, 2002.

[214] D. J. Gervais. The machine as author. *Iowa Law Review*, 105:1053–2106, 2020.

[215] P. Gervás. Computational modelling of poetry generation. In *Symposium on Artificial Intelligence and Poetry (AISB'13)*, 2013.

[216] J. L. Gibbs, G. L. Kirkwood, C. Fang, and J. N. Wilkenfeld. Negotiating agency and control: Theorizing human-machine communication from a structurational perspective. *Human-Machine Communication*, 2:153–171, 2021.

[217] A. Giddens. *The Constitution of Society*. University of California Press, 1984.

[218] J. Gillotte. Copyright infringement in ai-generated artworks. *UC Davis Law Review*, 53:2655, 2020.

[219] J. C. Ginsburg. Fair use for free, or permitted-but-paid? *Berkeley Technology Law Journal*, 29:1383–1446, 2014.

[220] J. C. Ginsburg. People not machines: Authorship and what it means in the berne convention. *IIC - International Review of Intellectual Property and Competition Law*, 49:131–135, 2018.

[221] A. Glaese, N. McAleese, M. Trebacz, J. Aslanides, V. Firoiu, T. Ewalds, M. Rauh, L. Weidinger, M. Chadwick, P. Thacker, L. Campbell-Gillingham, J. Uesato, P.-S. Huang, R. Comanescu, F. Yang, A. See, S. Dathathri, R. Greig, C. Chen, ..., and G. Irving. Improving alignment of dialogue agents via targeted human judgements, 2022. arXiv:2209.14375 [cs.LG].

[222] D. Glasser. Copyrights in computer-generated works: Whom, if anyone, do we reward? *Duke Law & Technology Review*, 24, 2001.

[223] F. Goes, M. Volpe, P. Sawicki, M. Grzés, and J. Watson. Pushing GPT's creativity to its limits: Alternative Uses and Torrance Tests. In *Proc. of the 14th International Conference on Computational Creativity (ICCC'23)*, 2023.

[224] S. Gong, M. Li, J. Feng, Z. Wu, and L. Kong. Diffuseq: Sequence to sequence text generation with diffusion models. In *Proc. of the 11th International Conference on Learning Representations (ICLR'23)*, 2023.

[225] I. Goodfellow. NIPS 2016 tutorial: Generative adversarial networks, 2017. arXiv:1701.00160 [cs.LG].

[226] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS'14)*, 2014.

[227] C. Goodhart. Problems of monetary management: The U.K. experience. *Papers in Monetary Economics*, 1(1):1–20, 1975.

[228] J. Gottlieb, P.-Y. Oudeyer, M. Lopes, and A. Baranes. Information-seeking, curiosity, and attention: Computational and neural mechanisms. *Trends in Cognitive Sciences*, 17(11):585–593, 2013.

[229] GPT-3. A robot wrote this entire article. are you scared yet, human?, 2020. `https://www.theguardian.com/commentisfree/2020/sep/08/robot-wrote-this-article-gpt-3` [Accessed October 25, 2024].

[230] K. Grace and M. L. Maher. What to expect when you're expecting: The role of unexpectedness in computationally evaluating creativity. In *Proc. of the 5th International Conference on Computational Creativity (ICCC'14)*, 2014.

[231] K. Grace and M. L. Maher. Specific curiosity as a cause and consequence of transformational creativity. In *Proc. of the 6th International Conference On Computational Creativity (ICCC'15)*, 2015.

[232] D. Gravina, A. Liapis, and G. Yannakakis. Surprise search: Beyond objectives and novelty. In *Proc. of the Genetic and Evolutionary Computation Conference (GECCO'16)*, 2016.

[233] D. Gravina, A. Liapis, and G. N. Yannakakis. Quality diversity through surprise. *IEEE Transactions on Evolutionary Computation*, 23:603–616, 2019.

[234] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra. DRAW: A recurrent neural network for image generation. In *Proc. of the 32nd International Conference on Machine Learning (ICML'15)*, 2015.

[235] J. Grimmelman. There's no such thing as a computer-authored work–and it's a good thing, too. *The Columbia Journal of Law & the Arts*, 39(3):403–416, 2016.

[236] A. Grinbaum and L. Adomaitis. The ethical need for watermarks in machine-generated language, 2022. arXiv:2209.03118 [cs.CL].

[237] A. Gu, K. Goel, and C. Re. Efficiently modeling long sequences with structured state spaces. In *Proc. of the 10th International Conference on Learning Representations (ICLR'22)*, 2022.

[238] A. Guadamuz. Do androids dream of electric copyright?: Comparative analysis of originality in artificial intelligence generated works. In *Artificial Intelligence and Intellectual Property*, pages 147–176. Oxford University Press, 2021.

[239] A. Guadamuz. Is GitHub's Copilot potentially infringing copyright?, 2021. `https://www.technollama.co.uk/is-githubs-copilot-potentially-infringing-copyright` [Accessed October 25, 2024].

[240] A. Guadamuz. A scanner darkly: Copyright liability and exceptions in artificial intelligence inputs and outputs. *GRUR International*, 73(2): 111–127, 2024.

[241] J. Gui, Z. Sun, Y. Wen, D. Tao, and Y. Jie-ping. A review on generative adversarial networks: Algorithms, theory, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):3313–3332, 2021.

[242] J. P. Guilford. Creativity: Yesterday, today, and tomorrow. *The Journal of Creative Behavior*, 1(1):3–14, 1967.

[243] G. L. Guimaraes, B. Sanchez-Lengeling, P. L. Cunha Farias, and A. Aspuru-Guzik. Objective-reinforced generative adversarial networks (ORGAN) for sequence generation models, 2017. arXiv:1705.10843 [stat.ML].

[244] I. Gulrajani, K. Kumar, F. Ahmed, A. A. Taiga, F. Visin, D. Vazquez, and A. Courville. PixelVAE: A latent variable model for natural images. In *Proc. of the 5th International Conference on Learning Representations (ICLR'17)*, 2017.

[245] S. Gunasekar, Y. Zhang, J. Aneja, C. C. T. Mendes, A. D. Giorno, S. Gopi, M. Javaheripi, P. Kauffmann, G. de Rosa, O. Saarikivi, A. Salim, S. Shah, H. S. Behl, X. Wang, S. Bubeck, R. Eldan, A. T. Kalai, Y. T. Lee, and Y. Li. Textbooks are all you need, 2023. arXiv:2306.11644 [cs.CL].

[246] J. Guo, S. Lu, H. Cai, W. Zhang, Y. Yu, and J. Wang. Long text generation via adversarial training with leaked information. In *Proc. of the 32nd AAAI Conference on Artificial Intelligence and 30th Innovative Applications of Artificial Intelligence Conference and 8th AAAI Symposium on Educational Advances in Artificial Intelligence*, 2018.

[247] T. Guo, X. Chen, Y. Wang, R. Chang, S. Pei, N. V. Chawla, O. Wiest, and X. Zhang. Large language model based multi-agents: A survey of progress and challenges. In *Proc. of the 33rd International Joint Conference on Artificial Intelligence (IJCAI'24)*, 2024.

[248] W. Gurnee and M. Tegmark. Language models represent space and time. In *Proc. of the 12th International Conference on Learning Representations (ICLR'24)*, 2024.

[249] M. Guzdial and M. O. Riedl. An interaction framework for studying co-creative AI. In *Proc. of the CHI'19 Human-Centered Machine Learning Perspectives Workshop*, 2019.

[250] A. L. Guzman. Ontological boundaries between humans and computers and the implications for human-machine communication. *Human-Machine Communication*, 1:37–54, 2020.

[251] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018.

[252] D. Ha and J. Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems (NIPS'18)*, 2018.

[253] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proc. of the 35th International Conference on Machine Learning (ICML'18)*, 2018.

[254] J. Haase and P. H. Hanel. Artificial muses: Generative artificial intelligence chatbots have risen to human-level creativity. *Journal of Creativity*, 33(3):100066, 2023.

[255] N. Haber, D. Mrowca, S. Wang, L. F. Fei-Fei, and D. L. Yamins. Learning to play with intrinsically-motivated, self-aware agents. In *Advances in Neural Information Processing Systems (NIPS'18)*, 2018.

[256] D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. Dragan. Cooperative inverse reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS'16)*, 2016.

[257] G. Hadjeres, F. Pachet, and F. Nielsen. DeepBach: a steerable model for bach chorales generation. In *Proc. of the 34th International Conference on Machine Learning (ICML'17)*, 2017.

[258] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels. In *Proc. of the 36th International Conference on Machine Learning (ICML'19)*, 2019.

[259] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. Dream to control: Learning behaviors by latent imagination. In *Proc. of the 8th International Conference on Learning Representations (ICLR'20)*, 2020.

[260] D. Hafner, T. Lillicrap, M. Norouzi, and J. Ba. Mastering atari with discrete world models. In *Proc. of the 9th International Conference on Learning Representations (ICLR'21)*, 2021.

[261] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap. Mastering diverse domains through world models, 2023. arXiv:2301.04104 [cs.AI].

[262] P. Haggard. Sense of agency in the human brain. *Nature Reviews Neuroscience*, 18(4):196–207, 2017.

[263] Z. Han, W. Yan, and G. Liu. A performance-based urban block generative design using deep reinforcement learning and computer vision. In *Proc. of 2020 DigitalFUTURES*, 2020.

[264] Y. Hao, Z. Chi, L. Dong, and F. Wei. Optimizing prompts for text-to-image generation. In *Proc. of the 37th Conference on Neural Information Processing Systems (NIPS'23)*, 2023.

[265] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'22)*, 2022.

[266] M. Heiferman. Photography murdered painting, right?, 2010. `https://siarchives.si.edu/blog/photography-murdered-painting-right` [Accessed October 25, 2024].

[267] M. Henaff, W. F. Whitney, and Y. LeCun. Model-based planning with discrete and continuous actions, 2017. arXiv:1705.07177 [cs.AI].

[268] P. Henderson, X. Li, D. Jurafsky, T. Hashimoto, M. A. Lemley, and P. Liang. Foundation models and fair use. *Journal of Machine Learning Research*, 24(400):1–79, 2023.

[269] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proc. of the 35th Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

[270] A. Hertzmann. Can computers create art? *Arts*, 7(2):18–42, 2018.

[271] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *Proc. of the 5th International Conference on Learning Representations (ICLR'17)*, 2017.

[272] E. R. Hilgard and G. H. Bower. *Theories of Learning.* Prentice-Hall, Inc., 1975.

[273] R. Hilty and H. Richter. Position statement of the Max Planck Institute for Innovation and Competition on the proposed modernisation of European Copyright Rules Part B Exceptions and Limitations (Art. 3 – Text and Data Mining). *Max Planck Institute for Innovation & Competition Research Paper No. 17-02*, 2017.

[274] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *Proc. of the NIPS'15 Deep Learning and Representation Learning Workshop*, 2015.

[275] J. Ho and T. Salimans. Classifier-free diffusion guidance. In *Proc. of the NIPS'21 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[276] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NIPS'20)*, 2020.

[277] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.

[278] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet. Video diffusion models. In *Proc. of the ICLR'22 Workshop on Deep Generative Models for Highly Structured Data*, 2022.

[279] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[280] E. Hoel. Enter the supersensorium: The neuroscientific case for Art in the age of Netflix, 2019. `https://thebaffler.com/salvos/enter-the-supersensorium-hoel` [Accessed October 25, 2024].

[281] E. Hoel. The overfitted brain: Dreams evolved to assist generalization. *Patterns*, 2(5):100244, 2021.

[282] E. Hoel. The banality of ChatGPT, 2022. `https://www.theintrinsicperspective.com/p/the-banality-of-chatgpt` [Accessed October 25, 2024].

[283] D. R. Hofstadter and M. Mitchell. The copycat project: A model of mental fluidity and analogy-making. In *Advances in Connectionist and Neural Computation Theory, Vol. 2. Analogical Connections*, pages 31–112. Ablex Publishing, 1994.

[284] C. Hokamp and Q. Liu. Lexically constrained decoding for sequence generation using grid beam search. In *Proc. of the 55th Annual Meeting of the Association for Computational Linguistics (ACL'17)*, 2017.

[285] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi. The curious case of neural text degeneration. In *Proc. of the 8th International Conference on Learning Representations (ICLR'20)*, 2020.

[286] W. Hong, M. Zhu, M. Liu, W. Zhang, M. Zhou, Y. Yu, and P. Sun. Generative adversarial exploration for reinforcement learning. In *Proc. of the 1st International Conference on Distributed Artificial Intelligence (DAI'19)*, 2019.

[287] Z.-W. Hong, I. Shenfeld, T.-H. Wang, Y.-S. Chuang, A. Pareja, J. R. Glass, A. Srivastava, and P. Agrawal. Curiosity-driven red-teaming for large language models. In *Proc. of the 12th International Conference on Learning Representations (ICLR'24)*, 2024.

[288] J. Hopkins and D. Kiela. Automatically generating rhythmic verse with neural networks. In *Proc. of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017.

[289] G. Hornby, A. Globus, D. Linden, and J. Lohn. Automated antenna design with evolutionary algorithms. In *Space 2006*, 2006.

[290] R. Houthooft, X. Chen, Y. Duan, J. Schulman, F. De Turck, and P. Abbeel. VIME: Variational information maximizing exploration. In *Proc. of the 30th International Conference on Neural Information Processing Systems (NIPS'16)*, 2016.

[291] C.-P. Hsieh, S. Sun, S. Kriman, S. Acharya, D. Rekesh, F. Jia, Y. Zhang, and B. Ginsburg. RULER: What's the real context size of your long-context language models? In *Proc. of the 1st Conference on Language Modeling (COLM'24)*, 2024.

[292] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *Proc. of the 10th International Conference on Learning Representations (ICLR'22)*, 2022.

[293] K. Hu. ChatGPT sets record for fastest-growing user base - analyst note, 2023. `https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/` [Accessed October 25, 2024].

[294] A. Huang and R. Wu. Deep learning for music, 2016. arXiv:1606.04930 [cs.LG].

[295] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck. Music Transformer. In *Proc. of the 7th International Conference on Learning Representations (ICLR'19)*, 2019.

[296] Q. Huang, A. Jansen, J. Lee, R. Ganti, J. Y. Li, and D. P. W. Ellis. MuLan: A joint embedding of music audio and natural language. In *Proc. of the 23rd International Society for Music Information Retrieval Conference (ISMIR'22)*, 2022.

[297] S. Huang, M. Noukhovitch, A. Hosseini, K. Rasul, W. Wang, and L. Tunstall. The N+ implementation details of RLHF with PPO: A case study on TL;DR summarization. In *Proc. of the 1st Conference on Language Modeling (COLM'24)*, 2024.

[298] Z. Huang, S. Zhou, and W. Heng. Learning to paint with model-based deep reinforcement learning. In *Proc. of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV'19)*, 2019.

[299] R. M. Hurt and R. M. Schuchman. The economic rationale of copyright. *The American Economic Review*, 56(1/2):421–432, 1966.

[300] M. Igl, L. Zintgraf, T. A. Le, F. Wood, and S. Whiteson. Deep variational reinforcement learning for POMDPs. In *Proc. of the 35th International Conference on Machine Learning (ICML'18)*, 2018.

[301] M. Igl, K. Ciosek, Y. Li, S. Tschiatschek, C. Zhang, S. Devlin, and K. Hofmann. Generalization in reinforcement learning with selective noise injection and information bottleneck. In *Advances in Neural Information Processing Systems (NIPS'19)*, 2019.

[302] "imagination". *Merriam-Webster.com*, 2024. `https://www.merriam-webster.com/dictionary/imagination` [Accessed October 25, 2024].

[303] D. Ippolito, A. Yuan, A. Coenen, and S. Burnam. Creative writing with an AI-powered writing assistant: Perspectives from professional writers, 2022. arXiv:2211.05030 [cs.HC].

[304] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proc. of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*, 2017.

[305] A. Issak. Artistic autonomy in AI art. In *Proc. of NIPS'21 Machine Learning for Creativity and Design Workshop*, 2021.

[306] L. Itti and P. Baldi. Bayesian surprise attracts human attention. *Vision Research*, 49(10):1295–1306, 2009.

[307] E. C. Jackson and M. Daley. Novelty search for deep reinforcement learning policy network weights by action sequence edit metric distance. In *Proc. of the Genetic and Evolutionary Computation Conference Companion (GECCO'19)*, 2019.

[308] J. Jaeger. Artificial intelligence is algorithmic mimicry: why artificial "agents" are not (and won't be) proper agents, 2023. arXiv:2307.07515 [cs.AI].

[309] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. In *Proc. of the 5th International Conference on Learning Representations (ICLR'17)*, 2017.

[310] J. Jang, S. Shin, and Y. Kim. Music2Video: Automatic generation of music video with fusion of audio and text, 2022. arXiv:2201.03809 [cs.SD].

[311] N. Jaques, S. Gu, R. E. Turner, and D. Eck. Generating music by fine-tuning recurrent neural networks with reinforcement learning. In *Proc. of the NIPS'16 Deep Reinforcement Learning Workshop*, 2016.

[312] N. Jaques, S. Gu, D. Bahdanau, J. M. Hernández-Lobato, R. E. Turner, and D. Eck. Sequence tutor: Conservative fine-tuning of sequence generation models with KL-control. In *Proc. of the 34th International Conference on Machine Learning (ICML'17)*, 2017.

[313] N. Jaques, S. Gu, R. E. Turner, and D. Eck. Tuning recurrent neural networks with reinforcement learning. In *Proc. of the ICLR'17 Workshop*, 2017.

[314] N. Jaques, J. H. Shen, A. Ghandeharioun, C. Ferguson, À. Lapedriza, N. Jones, S. Gu, and R. W. Picard. Human-centric dialog training via offline reinforcement learning. In *Proc. of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*, 2020.

[315] D. Jha, H. Chang, and M. Elhoseiny. Wolfflin's affective generative analysis for visual art. In *Proc. of the 20th International Conference on Computational Creativity (ICCC'21)*, 2021.

[316] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. V. Le, Y. Sung, Z. Li, and T. Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proc. of the 38th International Conference on Machine Learning (ICML'21)*, 2021.

[317] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed. Mistral 7B, 2023. arXiv:2310.06825 [cs.CL].

[318] M. Jiang, E. Grefenstette, and T. Rocktäschel. Prioritized level replay. In *Proc. of the 38th International Conference on Machine Learning (ICML'21)*, 2021.

[319] N. Jiang, S. Jin, Z. Duan, and C. Zhang. RL-Duet: Online music accompaniment generation using deep reinforcement learning. In *Proc. of the 34th AAAI Conference on Artificial Intelligence, the 32nd Innovative Applications of Artificial Intelligence Conference, the 10th AAAI Symposium on Educational Advances in Artificial Intelligence*, 2020.

[320] Y. Jin, J. Zhang, M. Li, Y. Tian, and H. Zhu. Towards the high-quality anime characters generation with generative adversarial networks. In *Proc. of the NIPS'17 Machine Learning for Creativity and Design Workshop*, 2017.

[321] D. G. Johnson and M. Verdicchio. AI, agency and responsibility: the VW fraud case and beyond. *AI & SOCIETY*, 34(3):639–647, 2019.

[322] M. I. Jordan, Z. Ghahrmamani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 1999.

[323] A. Jordanous. Four PPPPerspectives on computational creativity in theory and in practice. *Connection Science*, 28(2):294–216, 2016.

[324] A. K. Jordanous. Evaluating machine creativity. In *Proc. of the Seventh ACM Conference on Creativity and Cognition (C&C'09)*, 2009.

[325] J. M. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Zidek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, ..., and D. Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596:583–589, 2021.

[326] A. Kadurin, S. Nikolenko, K. Khrabrov, A. Aliper, and A. Zhavoronkov. druGAN: An advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico. *Molecular Pharmaceutics*, 14(9):3098–3104, 2017.

[327] K. Kandasamy, Y. Bachrach, R. Tomioka, D. Tarlow, and D. Carter. Batch policy gradient methods for improving neural conversation models. In *Proc. of the 5th International Conference on Learning Representations (ICLR'17)*, 2017.

[328] P. Karampiperis, A. Koukourikos, and E. Koliopoulou. Towards machines for measuring creativity: The use of computational tools in storytelling activities. In *Proc. of the 2014 IEEE 14th International Conference on Advanced Learning Technologies (ICALT'14)*, 2014.

[329] A. Karpathy. The unreasonable effectiveness of recurrent neural networks, 2015. `https://karpathy.github.io/2015/05/21/rnn-effectiveness` [Accessed October 25, 2024].

[330] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *Proc. of the 6th International Conference on Learning Representations (ICLR'18)*, 2018.

[331] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proc. of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'19)*, 2019.

[332] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of StyleGAN. In *Proc. of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'20)*, 2020.

[333] T. Karras, M. Aittala, S. Laine, E. Harkonen, J. Hellsten, J. Lehtinen, and T. Aila. Alias-free generative adversarial networks. In *Advances in Neural Information Processing Systems (NIPS'21)*, 2021.

[334] C. Kauf, A. A. Ivanova, G. Rambelli, E. Chersoni, J. S. She, Z. Chowdhury, E. Fedorenko, and A. Lenci. Event knowledge in large language models: The gap between the impossible and the unlikely. *Cognitive Science*, 47(11):e13386, 2023.

[335] H. Kazemi, S. M. Iranmanesh, and N. Nasrabadi. Style and content disentanglement in generative adversarial networks. In *Proc. of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV'19)*, 2019.

[336] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah. Transformers in vision: A survey. *ACM Computing Surveys*, 54(10s): 1–41, 2022.

[337] S. Kiegeland and J. Kreutzer. Revisiting the weaknesses of reinforcement learning for neural machine translation. In *Proc. of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'21)*, 2021.

[338] Y. Kim, W. Nam, H. Kim, J.-H. Kim, and G. Kim. Curiosity-bottleneck: Exploration by distilling task-specific novelty. In *Proc. of the 36th International Conference on Machine Learning (ICML'19)*, 2019.

[339] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *Proc. of the 2nd International Conference on Learning Representations (ICLR'14)*, 2014.

[340] D. P. Kingma and M. Welling. An introduction to variational autoencoders. *Foundations and Trends in Machine Learning*, 12(4):307–392, 2019.

[341] D. P. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems (NIPS'14)*, 2014.

[342] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems (NIPS'16)*, 2016.

[343] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, and T. Goldstein. A watermark for large language models. In *Proc. of the 40th International Conference on Machine Learning (ICML'23)*, 2023.

[344] H. R. Kirk, B. Vidgen, P. Röttger, and S. A. Hale. Personalisation within bounds: A risk taxonomy and policy framework for the alignment of large language models with personalised feedback, 2023. arXiv:2303.05453 [cs.CL].

[345] R. Kirk, A. Zhang, E. Grefenstette, and T. Rocktäschel. A survey of zero-shot generalisation in deep reinforcement learning. *Journal of Artificial Intelligence Research*, 76:201–264, 2023.

[346] R. Kirk, I. Mediratta, C. Nalmpantis, J. Luketina, E. Hambro, E. Grefenstette, and R. Raileanu. Understanding the effects of RLHF on LLM generalisation and diversity. In *Proc. of the 12th International Conference on Learning Representations (ICLR'24)*, 2024.

[347] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

[348] J. Koch, A. Lucero, L. Hegemann, and A. Oulasvirta. May AI? design ideation with cooperative contextual bandits. In *Proc. of the 2019 CHI Conference on Human Factors in Computing Systems (CHI'19)*, 2019.

[349] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro. DiffWave: A versatile diffusion model for audio synthesis. In *Proc. of the 9th International Conference on Learning Representations (ICLR'21)*, 2021.

[350] M. Kosinski. Evaluating large language models in theory of mind tasks. *Proceedings of the National Academy of Sciences*, 121(45):e2405460121, 2024.

[351] S. Kottur, X. Wang, and V. Carvalho. Exploring personalized neural conversational models. In *Proc. of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*, 2017.

[352] M. Krenn, F. Häse, A. Nigam, P. Friederich, and A. Aspuru-Guzik. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024, 2020.

[353] M. Kretschmer and T. Margoni. Data mining: why the EU's proposed copyright measures get it wrong, 2018. `https://www.iusinitinere.it/of-data-rights-and-boundaries-text-and-data-mining-under-eu-copyright-law-23891` [Accessed October 25, 2024].

[354] J. Kreutzer, S. Khadivi, E. Matusov, and S. Riezler. Can neural machine translation be improved with user feedback? In *Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers) (NAACL'18)*, 2018.

[355] E. Lagutin, D. Gavrilov, and P. Kalaidin. Implicit unlikelihood training: Improving neural text generation with reinforcement learning. In *Proc. of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021.

[356] C. Lamb, D. G. Brown, and C. L. A. Clarke. Evaluating computational creativity: An interdisciplinary tutorial. *ACM Computing Surveys*, 51 (2):28:1–34, 2019.

[357] A. K. Lampinen, I. Dasgupta, S. C. Y. Chan, H. R. Sheahan, A. Creswell, D. Kumaran, J. L. McClelland, and F. Hill. Language models, like humans, show content effects on reasoning tasks. *PNAS Nexus*, 3(7):pgae233, 2024.

[358] S. Lamprier, T. Scialom, A. Chaffin, V. Claveau, E. Kijak, J. Staiano, and B. Piwowarski. Generative cooperative networks for natural language generation. In *Proc. of the 39th International Conference on Machine Learning (ICML'22)*, 2022.

[359] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. In *Proc. of the 8th International Conference on Learning Representations (ICLR'20)*, 2020.

[360] P. Langley, H. A. Simon, G. L. Bradshaw, and J. M. Zytkow. *Scientific Discovery: Computational Explorations of the Creative Process*. The MIT Press, 1987.

[361] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. In *Proc. of the 33rd International Conference on Machine Learning (ICML'16)*, 2016.

[362] M. Laskin, K. Lee, A. Stooke, L. Pinto, P. Abbeel, and A. Srinivas. Reinforcement learning with augmented data. In *Advances in Neural Information Processing Systems (NIPS'20)*, 2020.

[363] J. H. Lau, T. Cohn, T. Baldwin, J. Brooke, and A. Hammond. Deepspeare: A joint neural model of poetic language, meter and rhyme. In *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018.

[364] A. Lavie and A. Agarwal. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proc. of the 2nd Workshop on Statistical Machine Translation (StatMT'07)*, 2007.

[365] H. Le, Y. Wang, A. D. Gotmare, S. Savarese, and S. C. H. Hoi. CodeRL: Mastering code generation through pretrained models and deep reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS'22)*, 2022.

[366] P. Le and W. Zuidema. Quantifying the vanishing gradient and long distance dependency problem in recursive neural networks and recursive LSTMs. In *Proc. of the 1st Workshop on Representation Learning for NLP*, 2016.

[367] Y. Le Cun. A path towards autonomous machine intelligence, 2022. `https://openreview.net/forum?id=BZ5a1r-kVsf` [Accessed November 30, 2023].

[368] A. X. Lee, A. Nagabandi, P. Abbeel, and S. Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model.

In *Advances in Neural Information Processing Systems (NIPS'20)*, 2020.

[369] G. Lee, M. Kim, Y. Lee, M. Lee, and B.-T. Zhang. Neural collage transfer: Artistic reconstruction via material manipulation. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV'23)*, 2023.

[370] H. Lee, S. Phatale, H. Mansoor, T. Mesnard, J. Ferret, K. R. Lu, C. Bishop, E. Hall, V. Carbune, A. Rastogi, and S. Prakash. RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback. In *Proc. of the 41st International Conference on Machine Learning (ICML'24)*, 2024.

[371] H. H. Lee, K. Shu, P. Achananuparp, P. K. Prasetyo, Y. Liu, E.-P. Lim, and L. R. Varshney. RecipeGPT: Generative pre-training based cooking recipe generation and evaluation system. In *Companion Proc. of the Web Conference 2020*, 2020.

[372] K. Lee, K. Lee, J. Shin, and H. Lee. Network randomization: A simple technique for generalization in deep reinforcement learning. In *Proc. of the 8th International Conference on Learning Representations (ICLR'20)*, 2020.

[373] K. Lee, H. Liu, M. Ryu, O. Watkins, Y. Du, C. Boutilier, P. Abbeel, M. Ghavamzadeh, and S. S. Gu. Aligning text-to-image models using human feedback, 2023. arXiv:2302.12192 [cs.LG].

[374] K. Lee, A. F. Cooper, and J. Grimmelmann. Talkin' 'bout AI generation: Copyright and the generative-AI supply chain, 2024. arXiv:2309.08133 [cs.CY].

[375] M. Lee, P. Liang, and Q. Yang. CoAuthor: Designing a human-ai collaborative writing dataset for exploring language model capabilities. In *Proc. of the 2022 CHI Conference on Human Factors in Computing Systems (CHI'22)*, 2022.

[376] J. Lehman and K. O. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19 (2):189–223, 2011.

[377] J. Lehman and K. O. Stanley. Beyond open-endedness: Quantifying impressiveness. In *Proc. of the 13th International Conference on the Synthesis and Simulation of Living Systems (ALIVE'12)*, 2012.

[378] J. Leike, D. Krueger, T. Everitt, M. Martic, V. Maini, and S. Legg. Scalable agent alignment via reward modeling: a research direction, 2018. arXiv:1811.07871 [cs.LG].

[379] M. A. Lemley and B. Casey. Fair learning. *Texas Law Review*, 99(4): 743–785, 2021.

[380] L. Lessig. *Free Culture: How Big Media Uses Technology and the Law to Lock down Culture and Control Creativity.* Penguin Press, 2004.

[381] B. Lester, R. Al-Rfou, and N. Constant. The power of scale for parameter-efficient prompt tuning. In *Proc. of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP'21)*, 2021.

[382] Y. Levine, I. Dalmedigos, O. Ram, Y. Zeldes, D. Jannai, D. Muhlgay, Y. Osin, O. Lieber, B. Lenz, S. Shalev-Shwartz, A. Shashua, K. Leyton-Brown, and Y. Shoham. Standing on the shoulders of giant frozen language models, 2022. arXiv:2204.10019 [cs.CL].

[383] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'20)*, 2020.

[384] C. Li, C. Yamanaka, K. Kaitoh, and Y. Yamanishi. Transformer-based objective-reinforced generative adversarial network to generate desired molecules. In *Proc. of the 31st International Joint Conference on Artificial Intelligence (IJCAI'22)*, 2022.

[385] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan. A diversity-promoting objective function for neural conversation models. In *Proc. of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'16)*, 2016.

[386] J. Li, W. Monroe, A. Ritter, D. Jurafsky, M. Galley, and J. Gao. Deep reinforcement learning for dialogue generation. In *Proc. of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP'16)*, 2016.

[387] J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, and D. Jurafsky. Adversarial learning for neural dialogue generation. In *Proc. of the*

2017 Conference on Empirical Methods in Natural Language Processing (EMNLP'17), 2017.

[388] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu. Neural speech synthesis with transformer network. In *Proc. of the 33rd AAAI Conference on Artificial Intelligence and 31st Innovative Applications of Artificial Intelligence Conference and 9th AAAI Symposium on Educational Advances in Artificial Intelligence*, 2019.

[389] T. Li, G. Zhang, Q. D. Do, X. Yue, and W. Chen. Long-context LLMs struggle with long in-context learning, 2024. arXiv:2404.02060 [cs.CL].

[390] X. L. Li, J. Thickstun, I. Gulrajani, P. Liang, and T. Hashimoto. Diffusion-LM improves controllable text generation. In *Advances in Neural Information Processing Systems (NIPS'22)*, 2022.

[391] Y. Li, O. Vinyals, C. Dyer, R. Pascanu, and P. Battaglia. Learning deep generative models of graphs. In *Proc. of the 35th International Conference on Machine Learning (ICML'18)*, 2018.

[392] A. Liapis, H. P. Martinez, J. Togelius, and G. N. Yannakakis. Transforming exploratory creativity with DeLeNoX. In *Proc. of the 4th International Conference on Computational Creativity (ICCC'13)*, 2013.

[393] A. Liapis, G. N. Yannakakis, and J. Togelius. Enhancements to constrained novelty search: Two-population novelty search for generating game content. In *Proc. of the 15th Annual Conference on Genetic and Evolutionary Computation (GECCO'13)*, 2013.

[394] S. E. Ligon. AI can create art, but can it own copyright in it, or infringe?, 2019. `https://www.lexisnexis.com/community/insights/legal/practical-guidance-journal/b/pa/posts/ai-can-create-art-but-can-it-own-copyright-in-it-or-infringe` [Accessed October 25, 2024].

[395] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. In *Proc. of the 4th International Conference on Learning Representations (ICLR'16)*, 2016.

[396] B. Lim and S. Zohren. Time-series forecasting with deep learning: A survey. *Philosophical Transactions of the Royal Society A*, 379: 20200209, 2021.

[397] C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In *Proc. of ACL'04 Workshop on Text Summarization Branches Out*, 2004.

[398] C. Linke, N. M. Ady, M. White, T. Degris, and A. White. Adapting behavior via intrinsic reward: A survey and empirical study. *Journal of Artificial Intelligence Research*, 69:1287–1332, 2020.

[399] S. Linkola, T. Takala, and H. Toivonen. Novelty-seeking multi-agent systems. In *Proc. of the 7th International Conference on Computational Creativity (ICCC'16)*, 2016.

[400] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley. AudioLDM: Text-to-audio generation with latent diffusion models. In *Proc. of the 40th International Conference on Machine Learning (ICML'23)*, 2023.

[401] J. Liu, S. Snodgrass, A. Khalifa, S. Risi, G. N. Yannakakis, and J. Togelius. Deep learning for procedural content generation. *Neural Computing and Applications*, 33:19–37, 2021.

[402] J. Liu, S. Hallinan, X. Lu, P. He, S. Welleck, H. Hajishirzi, and Y. Choi. Rainier: Reinforced knowledge introspector for commonsense question answering. In *Proc. of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP'22)*, 2022.

[403] S. Liu, S. Sabour, Y. Zheng, P. Ke, X. Zhu, and M. Huang. Rethinking and refining the distinct metric. In *Proc. of the 60th Annual Meeting of the Association for Computational Linguistics (ACL'22)*, 2022.

[404] V. Liu and L. B. Chilton. Design guidelines for prompt engineering text-to-image generative models. In *Proc. of the 2022 CHI Conference on Human Factors in Computing Systems (CHI'22)*, 2022.

[405] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach, 2019. arXiv:1907.11692 [cs.CL].

[406] G. Loewenstein. The psychology of curiosity: A review and reinterpretation. *Psychological Bulletin*, 116(1):75–98, 1994.

[407] R. Long. Introspective capabilities in large language models. *Journal of Consciousness Studies*, 30(9-10):143–153, 2023.

[408] X. Lu, S. Welleck, J. Hessel, L. Jiang, L. Qin, P. West, P. Ammanabrolu, and Y. Choi. QUARK: Controllable text generation with reinforced unlearning. In *Advances in Neural Information Processing Systems (NIPS'22)*, 2022.

[409] T. I. Lubart. Creativity and cross-cultural variation. *International Journal of Psychlogogy*, 25(1):39–59, 1990.

[410] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool. RePaint: Inpainting using denoising diffusion probabilistic models. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'22)*, 2022.

[411] G. S. J. Lunney. Fair use and market failure: Sony revisited. *Boston University Law Review*, 82:975–1030, 2002.

[412] C. Lyle, M. Rowland, W. Dabney, M. Kwiatkowska, and Y. Gal. Learning dynamics and generalization in deep reinforcement learning. In *Proc. of the 39th International Conference on Machine Learning (ICML'22)*, 2022.

[413] L. Maaløe, C. K. Sønderby, S. K. Sønderby, and O. Winther. Auxiliary deep generative models. In *Proc. of the 33rd International Conference on Machine Learning (ICML'16)*, 2016.

[414] L. Macedo and A. Cardoso. Assessing creativity: The importance of unexpected novelty. In *Proc. of the ECAI'02 Workshop on Creative Systems*, 2002.

[415] L. Macedo, R. Reisenzein, and A. Cardoso. Modeling forms of surprise in artificial agents: empirical and theoretical study of surprise functions. In *Proc. of the Annual Meeting of the Cognitive Science Society (CogSci'04)*, 2004.

[416] P. Machado, J. Romero, A. Santos, A. Cardoso, and A. Pazos. On the development of evolutionary artificial artists. *Computers and Graphics*, 31(6):818–826, 2007.

[417] D. J. MacKay. *Information Theory, Inference and Learning algorithms.* Cambridge University Press, 2003.

[418] O. Macmillan-Scott and M. Musolesi. (Ir)rationality and cognitive biases in large language models. *Royal Society Open Science*, 11(6): 240255, 2024.

[419] C. J. Maddison, A. Mnih, and Y. W. Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *Proc. of the 5th International Conference on Learning Representations (ICLR'17)*, 2017.

[420] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *Proc. of the 6th International Conference on Learning Representations (ICLR'18)*, 2018.

[421] M. Maher. Evaluating creativity in humans, computers, and collectively intelligent systems. In *Proc. of the 1st DESIRE Network Conference on Creativity and Innovation in Design*, 2010.

[422] M. Maher and D. Fisher. Using AI to evaluate creative designs. In *Proc. of the 2nd International Conference on Design Creativity (ICDC'12)*, 2012.

[423] K. Mahowald, A. A. Ivanova, I. A. Blank, N. Kanwisher, J. B. Tenenbaum, and E. Fedorenko. Dissociating language and thought in large language models. *Trends in Cognitive Sciences*, 28(6):517–540, 2024.

[424] A. Makhzani, J. Shlens, N. Jaitly, and I. Goodfellow. Adversarial autoencoders. In *Proc. of the 4th International Conference on Learning Representations (ICLR'16)*, 2016.

[425] R. Manurung, G. Ritchie, and H. Thompson. Using genetic algorithms to create meaningful poetic text. *Journal of Experimental & Theoretical Artificial Intelligence*, 24(1):43–64, 2012.

[426] A. Martin, G. Quispe, C. Ollion, S. Le Corff, F. Strub, and O. Pietquin. Learning natural language generation with truncated reinforcement learning. In *Proc. of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'22)*, 2022.

[427] L. J. Martin, P. Ammanabrolu, W. Hancock, S. Singh, B. Harrison, and M. O. Riedl. Event representations for automated story generation with deep neural nets. In *Proc. of the 32nd AAAI Conference on Artificial Intelligence and 30th Innovative Applications of Artificial Intelligence Conference and 8th AAAI Symposium on Educational Advances in Artificial Intelligence*, 2018.

[428] C. Martindale. *The Clockwork Muse: The Predictability of Artistic Change*. Basic Books, 1990.

[429] A. Matthias. The responsibility gap: Ascribing responsibility for the actions of learning automata. *Ethics and Information Technology*, 6 (3):175–183, 2004.

[430] P. Mazzaglia, O. Catal, T. Verbelen, and B. Dhoedt. Self-supervised exploration via latent Bayesian surprise. In *Proc. of the ICLR'21 Self-Supervision for Reinforcement Learning Workshop*, 2021.

[431] R. T. McCoy, P. Smolensky, T. Linzen, J. Gao, and A. Celikyilmaz. How much do language models copy from their training data? evaluating linguistic novelty in text generation using RAVEN. *Transactions of the Association for Computational Linguistics*, 11:652–670, 2023.

[432] J. R. Meehan. TALE-SPIN, an interactive program that writes stories. In *Proc. of the 5th International Joint Conference on Artificial Intelligence - Volume 1 (IJCAI'77)*, 1977.

[433] L. F. Menabrea and A. Lovelace. Sketch of the analytical engine invented by Charles Babbage. In *Scientific Memoirs*, volume 3, pages 666–731. Richard and John E. Taylor, 1843.

[434] O. Mendez-Lucio, B. Baillif, D.-A. Clevert, D. Rouquié, and J. Wichard. De novo generation of hit-like molecules from gene expression signatures using artificial intelligence. *Nature Communications*, 11(10):1–10, 2020.

[435] R. Mercado, T. Rastemo, E. Lindelof, G. Klambauer, O. Engkvist, H. Chen, and E. J. Bjerrum. Graph networks for molecular design. *Machine Learning: Science and Technology*, 2(2):025023, 2021.

[436] Meta. Introducing Meta Llama 3: The most capable openly available LLM to date, 2024. `https://ai.meta.com/blog/meta-llama-3/` [accessed October 25, 2024].

[437] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled generative adversarial networks. In *Proc. of the 5th International Conference on Learning Representations (ICLR'17)*, 2017.

[438] P. Mezei. From Leonardo to the Next Rembrandt – the need for AI-pessimism in the age of algorithms. *UFITA*, 84(2):390–429, 2020.

[439] G. Mialon, R. Dessi, M. Lomeli, C. Nalmpantis, R. Pasunuru, R. Raileanu, B. Roziere, T. Schick, J. Dwivedi-Yu, A. Celikyilmaz, E. Grave, Y. LeCun, and T. Scialom. Augmented language models: a survey, 2023. *Transactions on Machine Learning Research.*

[440] V. Micheli, E. Alonso, and F. Fleuret. Transformers are sample-efficient world models. In *Proc. of the 11th International Conference on Learning Representations (ICLR'23)*, 2023.

[441] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, and H. Wu. Mixed precision training. In *Proc. of the 6th International Conference on Learning Representations (ICLR'18)*, 2018.

[442] M. Miernicki. Artificial intelligence and moral rights. *AI & SOCIETY*, 36(1):319–329, 2021.

[443] A. I. Miller. *The Artist in the Machine.* The MIT Press, 2019.

[444] A. R. Miller. Copyright protection for computer programs, databases, and computer-generated works: Is anything new since CONTU? *Harvard Law Review*, 106(5):977–1073, 1993.

[445] M. Minsky. *The Emotion Machine.* Simon & Schuster, 2006.

[446] P. Mirowski, K. W. Mathewson, J. Pittman, and R. Evans. Co-writing screenplays and theatre scripts alongside language models using Dramatron. In *Proc. of the NIPS'22 Workshop on ML for Creativity & Design*, 2022.

[447] M. Mirza and S. Osindero. Conditional generative adversarial nets, 2014. arXiv:1411.1784 [cs.LG].

[448] G. Mittal, J. Engel, C. Hawthorne, and I. Simon. Symbolic music generation with diffusion models. In *Proc. of the 22nd Int. Society for Music Information Retrieval Conf. (ISMIR'21)*, 2021.

[449] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

[450] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Harley, T. P. Lillicrap, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proc. of the 33rd International Conference on Machine Learning (ICML'16)*, 2016.

[451] A. Mordvintsev, C. Olah, and M. Tyka. Inceptionism: Going deeper into neural networks, 2015. `https://blog.research.google/2015/06/inceptionism-going-deeper-into-neural.html` [Accessed October 25, 2024].

[452] A. Moreno and A. Etxeberria. Agency in natural and artificial systems. *Artificial Life*, 11(1-2):161–175, 2005.

[453] R. G. Morris, S. H. Burton, P. Bodily, and D. Ventura. Soup over bean of pure joy: Culinary ruminations of an artificial chef. In *Proc. of the 3rd International Conference on Computational Creativity (ICCC'12)*, 2012.

[454] S. Motamed, P. Rogalla, and F. Khalvati. RANDGAN: Randomized generative adversarial network for detection of COVID-19 in Chest X-Ray. *Scientific Reports*, 11:8602, 2021.

[455] Y. Mu, Y. Zhuang, B. Wang, G. Zhu, W. Liu, J. Chen, P. Luo, S. Li, C. Zhang, and J. Hao. Model-based reinforcement learning via imagination with derived memory. In *Advances in Neural Information Processing Systems (NIPS'21)*, 2021.

[456] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *Proc. of the 2018 IEEE International Conference on Robotics and Automation (ICRA'18)*, 2018.

[457] S. Narayan, S. B. Cohen, and M. Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proc. of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP'18)*, 2018.

[458] C. Nardo. The waluigi effect (mega-post), 2023. `https://lesswrong.com/posts/D7PumeYTDPfBTp3i7/the-waluigi-effect-mega-post` [Accessed October 25, 2024].

[459] N. W. Netanel. *Copyright's Paradox.* Oxford University Press, 2008.

[460] N. W. Netanel. Making sense of fair use. *Lewis & Clark Law Review*, 15:715–771, 2011.

[461] A. Newell, J. C. Shaw, and H. A. Simon. The processes of creative thinking. In *Contemporary Approaches to Creative Thinking: A Symposium Held at the University of Colorado*, pages 63–119. Atherton Press, 1962.

[462] S. Newman, A. Birhane, M. Zajko, O. A. Osoba, C. Prunkl, G. Lima, J. Bowen, R. Sutton, and C. Adams. AI & Agency, 2019. UCLA: The Program on Understanding Law, Science, and Evidence (PULSE). Retrieved from `https://escholarship.org/uc/item/8q15786s`.

[463] A. Newton and K. Dhole. Is AI art another industrial revolution in the making? In *Proc. of the AAAI'23 Creative AI Across Modalities Workshop*, 2023.

[464] A. Y. Ng and S. J. Russell. Algorithms for inverse reinforcement learning. In *Proc. of the 17th International Conference on Machine Learning (ICML'00)*, 2000.

[465] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in Neural Information Processing Systems (NIPS'16)*, 2016.

[466] A. Nguyen, J. Clune, Y. Bengio, A. Dosovitskiy, and J. Yosinski. Plug & play generative networks: Conditional iterative generation of images in latent space. In *Proc. of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*, 2017.

[467] P. C. H. Nguyen, N. N. Vlassis, B. Bahmani, W. Sun, H. S. Udaykumar, and S. S. Baek. Synthesizing controlled microstructures of porous media using generative adversarial networks and reinforcement learning. *Scientific Reports*, 12(1):9034–9049, 2022.

[468] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *Proc. of the 39th International Conference on Machine Learning (ICML'22)*, 2022.

[469] A. Nichol, H. Jun, P. Dhariwal, P. Mishkin, and M. Chen. Point-E: A system for generating 3d point clouds from complex prompts, 2022. arXiv:2212.08751 [cs.CV].

[470] A. Q. Nichol and P. Dhariwal. Improved denoising diffusion probabilistic models. In *Proc. of the 38th International Conference on Machine Learning (ICML'21)*, 2021.

[471] W. Nie, N. Narodytska, and A. Patel. RelGAN: Relational generative adversarial networks for text generation. In *Proc. of the 7th International Conference on Learning Representations (ICLR'19)*, 2019.

[472] D. Nimmer. "fairest of them all" and other fairy tales of fair use. *Law and Contemporary Problems*, 66:263–288, 2003.

[473] A. Odena. Semi-supervised learning with generative adversarial networks. In *Proc. of the ICML'16 Data Efficient Machine Learning workshop*, 2016.

[474] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classifier GANs. In *Proc. of the 34th International Conference on Machine Learning (ICML'17)*, 2017.

[475] C. Olah, A. Mordvintsev, and L. Schubert. Feature visualization, 2017. *Distill*.

[476] M. Olivecrona, T. Blaschke, O. Engkvist, and H. Chen. Molecular de-novo design through deep reinforcement learning. *Journal of Cheminformatics*, 9(1):48–61, 2017.

[477] J. A. Olson, J. Nahas, D. Chmoulevitch, S. J. Cropper, and M. E. Webb. Naming unrelated words predicts creativity. *Proceedings of the National Academy of Sciences*, 118(25):e2022340118, 2021.

[478] OpenAI. GPT-4 technical report, 2023. arXiv:2303.08774 [cs.CL].

[479] B. G. Otero. Machine Learning models under the Copyright microscope: Is EU Copyright fit for purpose? *GRUR International*, 70(11): 1043–1055, 2021.

[480] M. Ott, M. Auli, D. Grangier, and M. Ranzato. Analyzing uncertainty in neural machine translation. In *Proc. of the 35th International Conference on Machine Learning (ICML'18)*, 2018.

[481] M. Oudah, K. Makovi, K. Gray, B. Battu, and T. Rahwan. Perception of experience influences altruism and perception of agency influences trust in human–machine interactions. *Scientific Reports*, 14(1):12410, 2024.

[482] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. F. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NIPS'22)*, 2022.

[483] V. M. Palace. What if artificial intelligence wrote this? artificial intelligence and copyright law. *Florida Law Review*, 71:217–242, 2019.

[484] L. Pan, M. Saxon, W. Xu, D. Nathani, X. Wang, and W. Y. Wang. Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. *Transactions of the Association for Computational Linguistics*, 12:484–506, 2024.

[485] R. Y. Pang and H. He. Text generation by learning from demonstrations. In *Proc. of the 9th International Conference on Learning Representations (ICLR'21)*, 2021.

[486] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. BLEU: A method for automatic evaluation of machine translation. In *Proc. of the 40th Annual Meeting on Association for Computational Linguistics (ACL'02)*, 2002.

[487] R. Pardinas, G. Huang, D. Vazquez, and A. Piché. Leveraging human preferences to master poetry. In *Proc. of the AAAI'23 Workshop on Creative AI Across Modalities*, 2023.

[488] J. S. Park, J. C. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proc. of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST'23)*, 2023.

[489] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran. Image transformer. In *Proc. of the 35th International Conference on Machine Learning (ICML'18)*, 2018.

[490] A. Parrish. Exploring (semantic) space with (literal) robots, 2015. `http://opentranscripts.org/transcript/semantic-space-literal-robots/` [Accessed October 25, 2024].

[491] R. Pasunuru and M. Bansal. Multi-reward reinforced summarization with saliency and entailment. In *Proc. of the 2018 Conference of the North American Chapter of the Association for Computational*

Linguistics: Human Language Technologies, Volume 2 (Short Papers) (NAACL'18), 2018.

[492] S. Pateria, B. Subagdja, A.-h. Tan, and C. Quek. Hierarchical reinforcement learning: A comprehensive survey. *ACM Computing Surveys*, 54 (5):1–35, 2021.

[493] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proc. of the 34th International Conference on Machine Learning (ICML'17)*, 2017.

[494] D. Pathak, D. Gandhi, and A. Gupta. Self-supervised exploration via disagreement. In *Proc. of the 36th International Conference on Machine Learning (ICML'19)*, 2019.

[495] R. Paulus, C. Xiong, and R. Socher. A deep reinforced model for abstractive summarization. In *Proc. of the 6th International Conference on Learning Representations (ICLR'18)*, 2018.

[496] C. Payne. MuseNet, 2019. `https://openai.com/blog/musenet` [Accessed October 25, 2024].

[497] M. Peeperkorn, T. Kouwenhoven, D. Brown, and A. Jordanous. Is temperature the creativity parameter of large language models? In *Proc. of the 15th International Conference on Computational Creativity (ICCC'24)*, 2024.

[498] B. Peng, X. Li, L. Li, J. Gao, A. Celikyilmaz, S. Lee, and K.-F. Wong. Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning. In *Proc. of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP'17)*, 2017.

[499] B. Peng, E. Alcaide, Q. Anthony, A. Albalak, S. Arcadinho, S. Biderman, H. Cao, X. Cheng, M. Chung, L. Derczynski, X. Du, M. Grella, K. Gv, X. He, H. Hou, P. Kazienko, J. Kocon, J. Kong, B. Koptyra, ..., and R.-J. Zhu. RWKV: Reinventing RNNs for the transformer era. In *Findings of the Association for Computational Linguistics (EMNLP'23)*, 2023.

[500] J. Pennington, R. Socher, and C. Manning. GloVe: Global vectors for word representation. In *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*, 2014.

[501] R. Perez y Perez. *Mexica: 20 Years-20 Stories [20 años-20 historias].* Counterpath Press, 2017.

[502] S. T. Piantadosi and F. Hill. Meaning without reference in large language models. In *Proc. of the NeurIPS 2022 Workshop on Neuro Causal and Symbolic AI (nCSI'22)*, 2022.

[503] A. Ponce Del Castillo. Generative AI, generating precariousness for workers? *AI & SOCIETY*, 39:2601–2602, 2023.

[504] M. Popova, O. Isayev, and A. Tropsha. Deep reinforcement learning for de novo drug design. *Science Advances*, 4(7):eaap7885, 2018.

[505] M. Post. A call for clarity in reporting BLEU scores. In *Proc. of the 3rd Conference on Machine Translation: Research Papers (WMT'18)*, 2018.

[506] P. Potash, A. Romanov, and A. Rumshisky. GhostWriter: Using an LSTM for automatic rap lyric generation. In *Proc. of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP'15)*, 2015.

[507] D. Precup, R. S. Sutton, and S. P. Singh. Eligibility traces for off-policy policy evaluation. In *Proc. of the 17th International Conference on Machine Learning (ICML'00)*, 2000.

[508] R. A. Prentky. Mental illness and roots of genius. *Creativity Research Journal*, 13(1):95–104, 2001.

[509] S. Racanière, T. Weber, D. Reichert, L. Buesing, A. Guez, D. Jimenez Rezende, A. Puigdomènech Badia, O. Vinyals, N. Heess, Y. Li, R. Pascanu, P. Battaglia, D. Hassabis, D. Silver, and D. Wierstra. Imagination-augmented agents for deep reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS'17)*, 2017.

[510] Racter. *The Policeman's Beard Is Half Constructed.* Warner Books, Inc., 1984.

[511] A. Radford. Improving language understanding with unsupervised learning, 2018. `https://openai.com/blog/language-unsupervised` [Accessed October 25, 2024].

[512] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In

Proc. of the 4th International Conference on Learning Representations (ICLR'16), 2016.

[513] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners, 2019. `https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf` [Accessed October 25, 2024].

[514] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *Proc. of the 38th International Conference on Machine Learning (ICML'21)*, 2021.

[515] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn. Direct preference optimization: Your language model is secretly a reward model. In *Proc. of the 37th Conference on Neural Information Processing Systems (NIPS'23)*, 2023.

[516] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[517] R. Raileanu and R. Fergus. Decoupling value and policy for generalization in reinforcement learning. In *Proc. of the 38th International Conference on Machine Learning (ICML'21)*, 2021.

[518] R. Raileanu, M. Goldstein, D. Yarats, I. Kostrikov, and R. Fergus. Automatic data augmentation for generalization in reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS'21)*, 2021.

[519] W. T. Ralston. Copyright in computer-composed music: Hal meets handel. *Journal of the Copyright Society of the U.S.A.*, 52(3):281–308, 2004.

[520] R. Ramamurthy, P. Ammanabrolu, K. Brantley, J. Hessel, R. Sifa, C. Bauckhage, H. Hajishirzi, and Y. Choi. Is reinforcement learning (not) for natural language processing: Benchmarks, baselines, and building blocks for natural language policy optimization. In *Proc. of the 11th International Conference on Learning Representations (ICLR'23)*, 2023.

[521] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. In *Proc. of the 38th International Conference on Machine Learning (ICML'21)*, 2021.

[522] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with CLIP latents, 2022. arXiv:2204.06125 [cs.CV].

[523] W. Rammert. Where the action is. distributed agency between humans, machines, and programs. In *Paradoxes of Interactivity. Perspectives for Media Theory, Human-Computer Interaction, and Artistic Investigations*, pages 62–91. Bielefeld: transcript, 2008.

[524] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba. Sequence level training with recurrent neural networks. In *Proc. of the 4th International Conference on Learning Representations (ICLR'16)*, 2016.

[525] H. Rashkin, A. Celikyilmaz, Y. Choi, and J. Gao. PlotMachines: Outline-conditioned generation with dynamic plot state tracking. In *Proc. of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*, 2020.

[526] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In *Proc. of the 33rd International Conference on Machine Learning (ICML'16)*, 2016.

[527] S. Reed, K. Zolna, E. Parisotto, S. Gomez Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, T. Eccles, J. Bruce, A. Razavi, A. Edwards, N. Heess, Y. Chen, R. Hadsell, O. Vinyals, M. Bordbar, and N. de Freitas. A generalist agent, 2022. *Transactions on Machine Learning Research.*

[528] N. Reimers and I. Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proc. of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*, 2019.

[529] P. Reizinger and M. Szemenyei. Attention-based curiosity-driven exploration in deep reinforcement learning. In *Proc. of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'20)*, 2020.

[530] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning. In *Proc. of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*, 2017.

[531] M. Ressler. Automated inauthenticity, 2023. *AI & SOCIETY*. Accepted for publication.

[532] A. Revonsuo. The reinterpretation of dreams: An evolutionary hypothesis of the function of dreaming. *Behavioral and Brain Sciences*, 23(6): 877–901, 2000.

[533] L. Reynolds and K. McDonell. Multiversal views on language models, 2021. arXiv:2102.06391 [cs.HC].

[534] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proc. of the 31st International Conference on Machine Learning (ICML'14)*, 2014.

[535] M. Rhodes. An analysis of creativity. *The Phi Delta Kappan*, 42(7): 305–310, 1961.

[536] M. O. Riedl. Computational creativity as meta search, 2018. `https://mark-riedl.medium.com/computational-creativity-as-meta-search-6cad95da923b` [Accessed October 25, 2024].

[537] M. O. Riedl and R. M. Young. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research*, 39(1):217–268, 2010.

[538] G. Ritchie. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines*, 17:76–99, 2007.

[539] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck. A hierarchical latent vector model for learning long-term structure in music. In *Proc. of the 35th International Conference on Machine Learning (ICML'18)*, 2018.

[540] M. Roemmele and A. S. Gordon. Automated assistance for creative writing with an RNN language model. In *Proc. of the 23rd International Conference on Intelligent User Interfaces Companion (IUI'18)*, 2018.

[541] C. R. Rogers. Toward a theory of creativity. *ETC: A Review of General Semantics*, 11(4):249–260, 1954.

[542] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'22)*, 2022.

[543] E. Rosati. Copyright as an obstacle or an enabler? a european perspective on text and data mining and its role in the development of AI creativity. *Asia Pacific Law Review*, 27(2):198–217, 2019.

[544] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman. DreamBooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'23)*, 2023.

[545] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

[546] M. A. Runco and G. J. Jaeger. The standard definition of creativity. *Creativity Research Journal*, 24(1):92–96, 2012.

[547] S. J. Russell. *Human Compatible: Artificial Intelligence and the Problem of Control*. Penguin, 2019.

[548] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2021.

[549] R. M. Ryan and E. L. Deci. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary Educational Psychology*, 25(1):54–67, 2000.

[550] R. M. Ryan and E. L. Deci. Self-regulation and the problem of human autonomy: Does psychology need choice, self-determination, and will? *Journal of Personality*, 74(6):1557–1585, 2006.

[551] M. Sag. The new legal landscape for text mining and machine learning. *Journal of the Copyright Society of the USA*, 66:291–363, 2019.

[552] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, J. Ho, D. J. Fleet, and M. Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems (NIPS'22)*, 2022.

[553] E. Samuels. The idea-expression dichotomy in copyright law. *Tennessee Law Review*, 56:321–463, 1988.

[554] P. Samuelson. Allocating ownership rights in computer-generated works. *University of Pittsburgh Law Review*, 47:1185, 1985.

[555] P. Samuelson. Unbundling fair uses. *Fordham Law Review*, 77(5):2537–2621, 2009.

[556] P. Samuelson. Text and data mining of in-copyright works: is it legal? *Communications of the ACM*, 64(11):20–22, 2021.

[557] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *Proc. of the NIPS'19 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing*, 2019.

[558] C. Santos and A. Machado. Intellectual property on works of art made by artificial intelligence. *International Journal of Advanced Engineering Research and Science*, 7(12):49–59, 2020.

[559] M. Sap, R. Le Bras, D. Fried, and Y. Choi. Neural theory-of-mind? on the limits of social intelligence in large LMs. In *Proc. of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP'22)*, 2022.

[560] R. Saunders and O. Bown. Computational social creativity. *Artificial Life*, 21(3):366–378, 2015.

[561] R. Saunders and J. S. Gero. The digital clockwork muse: A computational model of aesthetic evolution. In *Proc. of the AISB'01 Symposium on AI and Creativity in Arts and Science (SSAISB'01)*, 2001.

[562] N. Savinov, A. Raichuk, D. Vincent, R. Marinier, M. Pollefeys, T. Lillicrap, and S. Gelly. Episodic curiosity through reachability. In *Proc. of the 7th International Conference on Learning Representations (ICLR'19)*, 2019.

[563] O. Sbai, M. Elhoseiny, A. Bordes, Y. LeCun, and C. Couprie. DesIGN: Design inspiration from generative networks. In *Proc. of the Computer Vision - ECCV'18 Workshops*, 2019.

[564] T. Schaul, J. Quan, I. Antonoglou, and D. Silver. Prioritized experience replay. In *Proc. of the 33th International Conference on Machine Learning (ICML'16)*, 2016.

[565] J. Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247, 2010.

[566] M. D. Schmidt and H. Lipson. Distilling free-form natural laws from experimental data. *Science*, 324:81–85, 2009.

[567] V. Schmidt, A. S. Luccioni, M. Teng, T. Zhang, A. Reynaud, S. Raghupathi, G. Cosne, A. Juraver, V. Vardanyan, A. Hernandez-Garcia, and Y. Bengio. ClimateGAN: Raising climate change awareness by generating images of floods. In *Proc. of the 10th International Conference on Learning Representations (ICLR'22)*, 2022.

[568] J. M. Schraagen. Bounded autonomy. In *Responsible Use of AI in Military Systems*, pages 345–370. CRC Press, 2024.

[569] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, T. Lillicrap, and D. Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588:604–609, 2020.

[570] C. Schuhmann. LAION-Aesthetics, 2022. `https://laion.ai/blog/laion-aesthetics/` [Accessed October 25, 2024].

[571] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In *Proc. of the 32nd International Conference on Machine Learning (ICML'15)*, 2015.

[572] J. Schulman, P. Moritz, S. Levine, M. I. Jordan, and P. Abbeel. High-dimensional continuous control using generalized advantage estimation. In *Proc. of the 4th International Conference on Learning Representations (ICLR'16)*, 2016.

[573] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms, 2017. arXiv:1707.06347 [cs.LG].

[574] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni. Green AI, 2019. arXiv:1907.10597 [cs.CY].

[575] T. Scialom, P.-A. Dray, S. Lamprier, B. Piwowarski, and J. Staiano. Coldgans: Taming language gans with cautious sampling strategies. In *Proc. of the 34th International Conference on Neural Information Processing Systems (NIPS'20)*, 2020.

[576] T. Scialom, T. Chakrabarty, and S. Muresan. Fine-tuned language models are continual learners. In *Proc. of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP'22*, 2022.

[577] R. Sekar, O. Rybkin, K. Daniilidis, P. Abbeel, D. Hafner, and D. Pathak. Planning to explore via self-supervised world models. In *Proc. of the 37th International Conference on Machine Learning (ICML'20)*, 2020.

[578] T. Sellam, D. Das, and A. Parikh. BLEURT: Learning robust metrics for text generation. In *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'20)*, 2020.

[579] S. Semeniuta, A. Severyn, and E. Barth. A hybrid convolutional variational autoencoder for text generation. In *Proc. of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP'17)*, 2017.

[580] sentence-transformers. sentence-transformers/bert-large-nli-stsb-mean-tokens, 2024. `https://huggingface.co/sentence-transformers/bert-large-nli-stsb-mean-tokens` [Accessed February 4, 2025].

[581] Y. Senzai and M. Scanziani. The brain simulates actions and their consequences during rem sleep, 2024. bioRxiv 2024.08.13.607810.

[582] G. Serapio-García, M. Safdari, C. Crepy, L. Sun, S. Fitz, P. Romero, M. Abdulhai, A. Faust, and M. Matarić. Personality traits in large language models, 2023. arXiv:2307.00184 [cs.CL].

[583] A. Seth. *Being You: A New Science of Consciousness.* Penguin, 2021.

[584] M. Shanahan. Simulacra as conscious exotica. *Inquiry*, 0(0):1–29, 2024.

[585] M. Shanahan. Talking about large language models. *Communications of the ACM*, 67(2):68–79, 2024.

[586] M. Shanahan and C. Clarke. Evaluating large language model creativity from a literary perspective, 2023. arXiv:2312.03746 [cs.CL].

[587] M. Shanahan, K. McDonell, and L. Reynolds. Role play with large language models. *Nature*, 623(7987):493–498, 2023.

[588] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *Proc. of the 5th International Conference on Learning Representations (ICLR'17)*, 2017.

[589] L. Shi, S. Li, Q. Zheng, M. Yao, and G. Pan. Efficient novelty search through deep reinforcement learning. *IEEE Access*, 8:128809–128818, 2020.

[590] Z. Shi, X. Chen, X. Qiu, and X. Huang. Toward diverse text generation with inverse reinforcement learning. In *Proc. of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*, 2018.

[591] H. Shin, J. K. Lee, J. Kim, and J. Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems (NIPS'17)*, 2017.

[592] M. Shin, J. Kim, B. van Opheusden, and T. L. Griffiths. Superhuman artificial intelligence can improve human decision-making by increasing novelty. *Proceedings of the National Academy of Sciences*, 120(12): e2214840120, 2023.

[593] N. Shinn, F. Cassano, E. Berman, A. Gopinath, K. Narasimhan, and S. Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS'23)*, 2023.

[594] M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro. Megatron-LM: Training multi-billion parameter language models using model parallelism, 2019. arXiv:1909.08053 [cs.CL].

[595] P. Shojaee, A. Jain, S. Tipirneni, and C. K. Reddy. Execution-based code generation using deep reinforcement learning, 2023. *Transactions on Machine Learning Research.*

[596] H. Shteingart and Y. Loewenstein. Reinforcement learning and human behavior. *Current Opinion in Neurobiology*, 25:93–98, 2014.

[597] I. Shumailov, Z. Shumaylov, Y. Zhao, N. Papernot, R. Anderson, and Y. Gal. AI models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759, 2024.

[598] N. Siddique, P. Dhakan, I. Rano, and K. Merrick. A review of the relationship between novelty, intrinsic motivation and reinforcement learning. *Paladyn, Journal of Behavioral Robotics*, 8:58–69, 2017.

[599] E. Silva. How photography pioneered a new understanding of art, 2022. `https://www.thecollector.com/how-photography-transformed-art/` [Accessed October 25, 2024].

[600] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–503, 2016.

[601] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Proc. of the ICLR'14 Workshop*, 2014.

[602] J. Singh and L. Zheng. Combining semantic guidance and deep reinforcement learning for generating human level paintings. In *Proc. of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'21)*, 2021.

[603] J. Singh, C. Smith, J. Echevarria, and L. Zheng. Intelli-Paint: Towards developing more human-intelligible painting agents. In *Proc. of the 17th European Conference on Computer Vision (ECCV'22)*, 2022.

[604] S. Singh, A. G. Barto, and N. Chentanez. Intrinsically motivated reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS'04)*, 2004.

[605] J. M. V. Skalse, N. H. R. Howe, D. Krasheninnikov, and D. Krueger. Defining and characterizing reward gaming. In *Advances in Neural Information Processing Systems (NIPS'22)*, 2022.

[606] I. Skorokhodov, S. Tulyakov, and M. Elhoseiny. StyleGAN-V: A continuous video generator with the price, image quality and perks of StyleGAN2. In *Proc. of the 2022 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'22)*, 2022.

[607] P. R. Slowinski. Rethinking software protection. In *Artificial Intelligence and Intellectual Property*, page 341–362. Oxford University Press, 2021.

[608] C. Small, M. Bjorkegren, T. Erkkilä, L. Shaw, and C. Megill. Polis: Scaling deliberation by mapping high dimensional opinion spaces. *Recerca : Revista de Pensament i Anàlisi*, 26(2):1–26, 2021.

[609] L. Smith. AI and IP: copyright in AI-generated works (UK law), 2017. talkingtech.cliffordchance.com/en/ip/copyright/ai-and-ip--copyright-in-ai-generated-works--uk-law-.html [Accessed June 21, 2021].

[610] S. Smith, M. Patwary, B. Norick, P. LeGresley, S. Rajbhandari, J. Casper, Z. Liu, S. Prabhumoye, G. Zerveas, V. Korthikanti, E. Zhang, R. Child, R. Y. Aminabadi, J. Bernauer, X. Song, M. Shoeybi, Y. He, M. Houston, S. Tiwary, and B. Catanzaro. Using DeepSpeed and Megatron to train Megatron-Turing NLG 530B, a large-scale generative language model, 2022. arXiv:2201.11990 [cs.CL].

[611] C. Snell. Alien dreams: An emerging art scene, 2021. https://mlberkeley.substack.com/p/clip-art/ [Accessed October 25, 2024].

[612] C. V. Snell, I. Kostrikov, Y. Su, S. Yang, and S. Levine. Offline RL for natural language generation with implicit language Q learning. In *Proc. of the 11th International Conference on Learning Representations (ICLR'23)*, 2023.

[613] B. Sobel. Artificial intelligence's fair use crisis. *Columbia Journal of Law and the Arts*, 41(1):45–97, 2017.

[614] B. Sobel. A taxonomy of training data: Disentangling the mismatched rights, remedies, and rationales for restricting machine learning. In *Artificial Intelligence and Intellectual Property*, page 221–242. Oxford University Press, 2021.

[615] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proc. of the 32nd International Conference on Machine Learning (ICML'15)*, 2015.

[616] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In *Proc. of the 9th International Conference on Learning Representations (ICLR'21)*, 2021.

[617] Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems (NIPS'19)*, 2019.

[618] Y. Song and S. Ermon. Improved techniques for training score-based generative models. In *Advances in Neural Information Processing Systems (NIPS'20)*, 2020.

[619] Y. Song, Y. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *Proc. of the 9th International Conference on Learning Representations (ICLR'21)*, 2021.

[620] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever. Consistency models. In *Proc. of the 40th International Conference on Machine Learning (ICML'23)*, 2023.

[621] N. Sousa e Silva. Are AI models' weights protected databases?, 2024. `https://copyrightblog.kluweriplaw.com/2024/01/18/are-ai-models-weights-protected-databases/` [last access: October 25, 2024].

[622] A. Srivastava, A. Rastogi, A. Rao, A. A. M. Shoeb, A. Abid, A. Fisch, A. R. Brown, A. Santoro, A. Gupta, A. Garriga-Alonso, A. Kluska, A. Lewkowycz, A. Agarwal, A. Power, A. Ray, A. Warstadt, A. W. Kocurek, A. Safaya, A. Tazarv, ..., and Z. Wu. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models, 2023. *Transactions on Machine Learning Research.*

[623] R. Stallman. Why "open source" misses the point of free software. *Communications of the ACM*, 52(6):31–33, 2009.

[624] M. I. Stein. *Stimulating Creativity. Volume 1.* Academic Press, 1974.

[625] C. Stevenson, I. Smal, M. Baas, R. Grasman, and H. van der Maas. Putting GPT-3's creativity to the (Alternative Uses) Test. In *Proc. of the 13th International Conference on Computational Creativity (ICCC'22)*, 2022.

[626] N. Stiennon, L. Ouyang, J. Wu, D. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. F. Christiano. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems (NIPS'20)*, 2020.

[627] S. Still and D. Precup. An information-theoretic approach to curiosity-driven reinforcement learning. *Theory in Biosciences*, 131:139–148, 2012.

[628] B. L. Sturm, J. F. Santos, O. Ben-Tal, and I. Korshunova. Music transcription modelling and composition using deep learning. In *Proc. of the 1st Conference on Computer Simulation of Musical Creativity (CSMC'16)*, 2016.

[629] B. L. Sturm, M. Iglesias, O. Ben-Tal, M. Miron, and E. Gomez. Artificial intelligence and music: Open questions of copyright law and engineering praxis. *Arts*, 8(3):115, 2019.

[630] P.-H. Su, M. Gašić, N. Mrkšić, L. M. Rojas-Barahona, S. Ultes, D. Vandyke, T.-H. Wen, and S. Young. On-line active reward learning for policy optimisation in spoken dialogue systems. In *Proc. of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL'16)*, 2016.

[631] Y. Su, T. Lan, Y. Wang, D. Yogatama, L. Kong, and N. Collier. A contrastive framework for neural text generation. In *Advances in Neural Information Processing Systems (NIPS'22)*, 2022.

[632] F. P. Such, V. Madhavan, E. Conti, J. Lehman, K. O. Stanley, and J. Clune. Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning, 2017. arXiv:1712.06567 [cs.NE].

[633] T. R. Sumers, S. Yao, K. Narasimhan, and T. L. Griffiths. Cognitive architectures for language agents, 2023. *Transactions on Machine Learning Research.*

[634] D. Summers-Stay, C. R. Voss, and S. M. Lukin. Brainstorm, then select: a generative language model improves its creativity score. In *Proc. of the AAAI'23 Workshop on Creative AI Across Modalities*, 2023.

[635] F.-K. Sun, C.-H. Ho, and H.-Y. Lee. LAMOL: LAnguage MOdeling for Lifelong Language Learning. In *Proc. of the 8th International Conference on Learning Representations (ICLR'20)*, 2020.

[636] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems (NIPS'14)*, 2014.

[637] R. S. Sutton. *Temporal Credit Assignment in Reinforcement Learning.* PhD thesis, University of Massachusetts Amherst, 1984.

[638] R. S. Sutton. Dyna, an integrated architecture for learning, planning, and reacting. In *Working Notes of the 1991 AAAI Spring Symposium*, 1991.

[639] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, 2018.

[640] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems (NIPS'99)*, 1999.

[641] M. Suzuki and Y. Matsuo. A survey of multimodal deep generative models. *Advanced Robotics*, 36(5-6):261–278, 2022.

[642] P. Tambwekar, M. Dhuliawala, L. J. Martin, A. Mehta, B. Harrison, and M. O. Riedl. Controllable neural story plot generation via reward shaping. In *Proc. of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19)*, 2019.

[643] A. Tamkin, M. Brundage, J. Clark, and D. Ganguli. Understanding the capabilities, limitations, and societal impact of large language models, 2021. arXiv:2102.02503 [cs.CL].

[644] W. R. Tan, C. S. Chan, H. E. Aguirre, and K. Tanaka. ArtGAN: Artwork synthesis with conditional categorical GANs. In *Proc. of the 2017 IEEE International Conference on Image Processing (ICIP'17)*, 2017.

[645] K. Terzidis, F. Fabrocini, and H. Lee. Unintentional intentionality: art and design in the age of artificial intelligence. *AI & SOCIETY*, 38(4): 1715–1724, 2022.

[646] L. A. Thiede, M. Krenn, A. Nigam, and A. Aspuru-Guzik. Curiosity in exploring chemical spaces: intrinsic rewards for molecular reinforcement learning. *Machine Learning: Science and Technology*, 3(3): 035008, 2022.

[647] Y. Tian and D. Ha. Modern evolution strategies for creativity: Fitting concrete images and abstract concepts. In *Proc. of the 11th Conference on Artificial Intelligence in Music, Sound, Art and Design (EvoMUSART 2022)*, 2022.

[648] Y. Tian, A. Ravichander, L. Qin, R. L. Bras, R. Marjieh, N. Peng, Y. Choi, T. L. Griffiths, and F. Brahman. Thinking out-of-the-box:

A comparative investigation of human and LLMs in creative problem-solving. In *Proc. of the ICML'24 Workshop on LLMs and Cognition*, 2024.

[649] J. Tien, J. Z.-Y. He, Z. Erickson, A. Dragan, and D. S. Brown. Causal confusion and reward misidentification in preference-based reward learning. In *Proc. of the 11th International Conference on Learning Representations (ICLR'23)*, 2023.

[650] A. Tilson and C. M. Gelowitz. Towards generating image assets through deep learning for game development. In *Proc. of the 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE'19)*, 2019.

[651] N. Tishby and N. Zaslavsky. Deep learning and the information bottleneck principle. In *Proc. of the 2015 IEEE Information Theory Workshop (ITW'15)*, 2015.

[652] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. In *Proc. of the 37th Annual Allerton Conference on Communication, Control and Computing*, 1999.

[653] V. Tomas. Creativity in art. *The Philosophical Review*, 67(1):1–15, 1958.

[654] W. Totschnig. Fully autonomous AI. *Science and Engineering Ethics*, 26:2473–2485, 2020.

[655] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. LLaMA: Open and efficient foundation language models, 2023. arXiv:2302.13971 [cs.CL].

[656] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, ..., and T. Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. arXiv:2307.09288 [cs.CL].

[657] D. J. Treffinger. *Creativity, Creative Thinking, and Critical Thinking: In Search of Definitions.* Center for Creative Learning, 1996.

[658] M. Tribus. *Thermodynamics and Thermostatics: An Introduction to Energy, Information and States of Matter, with Engineering Applications.* Van Nostrand, 1961.

[659] M. Tsimpoukelli, J. Menick, S. Cabi, S. M. A. Eslami, O. Vinyals, and F. Hill. Multimodal few-shot learning with frozen language models. In *Advances in Neural Information Processing Systems (NIPS'21)*, 2021.

[660] P. Tsividis, T. Pouncy, J. L. Xu, J. B. Tenenbaum, and S. J. Gershman. Human learning in atari. In *Proc. of the 2017 AAAI Spring Symposium Series, Science of Intelligence: Computational Principles of Natural and Artificial Intelligence*, 2017.

[661] L. Tunstall, L. von Werra, and T. Wolf. *Natural Language Processing with Transformers*. O'Reilly, 2022.

[662] A. M. Turing. Computing machinery and intelligence. *Mind*, LIX(236): 433–460, 1950.

[663] S. R. Turner. *The Creative Process: A Computer Model of Creativity and Storytelling*. Lawrence Erlbaum Associates, Inc, Hillsdale, NJ, 1994.

[664] R. Twomey. Communing with creative ai. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 6(2):28:1–7, 2023.

[665] T. Ullman. Large language models fail on trivial alterations to theory-of-mind tasks, 2023. arXiv:2302.08399 [cs.AI].

[666] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. WaveNet: A generative model for raw audio. In *Proc. of the 9th ISCA Workshop on Speech Synthesis*, 2016.

[667] A. Van Den Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. In *Proc. of The 33rd International Conference on Machine Learning (ICML'16)*, 2016.

[668] A. Van Den Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu. Conditional image generation with PixelCNN decoders. In *Advances in Neural Information Processing Systems (NIPS'16)*, 2016.

[669] A. Van Den Oord, O. Vinyals, and K. Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems (NIPS'17)*, 2017.

[670] H. van Hasselt, A. Guez, and D. Silver. Deep reinforcement learning with double Q-learning. In *Proc. of the 30th AAAI Conference on Artificial Intelligence (AAAI'16)*, 2016.

[671] L. R. Varshney. Mathematical limit theorems for computational creativity. *IBM Journal of Research and Development*, 63(1):2:1–12, 2019.

[672] L. R. Varshney, F. Pinel, K. R. Varshney, D. Bhattacharjya, A. Schoergendorfer, and Y.-M. Chee. A big data approach to computational creativity: The curious case of chef watson. *IBM Journal of Research and Development*, 63(1):7:1–18, 2019.

[673] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS'17)*, 2017.

[674] G. Vernier, H. Caselles-Dupré, and P. Fautrel. Electric dreams of ukiyo: A series of japanese artworks created by an artificial intelligence. *Patterns*, 1(2):100026, 2020.

[675] A. Vijayakumar, M. Cogswell, R. Selvaraju, Q. Sun, S. Lee, D. Crandall, and D. Batra. Diverse beam search for improved description of complex scenes. *Proc. of the 32nd AAAI Conference on Artificial Intelligence (AAAI'18)*, 2018.

[676] N. Vyas, S. M. Kakade, and B. Barak. On provable copyright protection for generative models. In *Proc. of the 40th International Conference on Machine Learning (ICML'23)*, 2023.

[677] T. Waite. AI-generated artworks are disappointing at auction, 2019. `https://www.dazeddigital.com/art-photography/article/46839/1/ai-generated-artworks-disappointing-at-auction-obvious-artificial-intelligence` [Accessed October 25, 2024].

[678] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li, L. He, S. Zhao, and F. Wei. Neural codec language models are zero-shot text to speech synthesizers, 2023. arXiv:2301.02111 [cs.CL].

[679] H. Wang, J. Zou, M. Mozer, A. Goyal, A. Lamb, L. Zhang, W. J. Su, Z. Deng, M. Q. Xie, H. Brown, and K. Kawaguchi. Can AI be as creative as humans?, 2024. arXiv:2401.01623 [cs.AI].

[680] K. Wang, B. Kang, J. Shao, and J. Feng. Improving generalization in reinforcement learning with mixture regularization. In *Advances in Neural Information Processing Systems (NeurIPS'20)*, 2020.

[681] P. Wang, L. Li, L. Chen, Z. Cai, D. Zhu, B. Lin, Y. Cao, Q. Liu, T. Liu, and Z. Sui. Large language models are not fair evaluators. In *Proc. of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL'24)*, 2024.

[682] Z. Wang, Q. She, and T. E. Ward. Generative adversarial networks in computer vision: A survey and taxonomy. *ACM Computing Surveys*, 54(2):1–38, 2021.

[683] C. J. C. H. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8 (3):279–292, 1992.

[684] M. Watter, J. T. Springenberg, J. Boedecker, and M. Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in Neural Information Processing Systems (NIPS'15)*, 2015.

[685] J. B. W. Webber. A bi-symmetric log transformation for wide-range data. *Measurement Science and Technology*, 24(2):027001, 2012.

[686] J. Wei, X. Wang, D. Schuurmans, M. Bosma, b. ichter, F. Xia, E. Chi, Q. V. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NIPS'22)*, 2022.

[687] L. Weidinger, J. Uesato, M. Rauh, C. Griffin, P.-S. Huang, J. Mellor, A. Glaese, M. Cheng, B. Balle, A. Kasirzadeh, C. Biles, S. Brown, Z. Kenton, W. Hawkins, T. Stepleton, A. Birhane, L. A. Hendricks, L. Rimell, W. Isaac, J. Haas, S. Legassick, G. Irving, and I. Gabriel. Taxonomy of risks posed by language models. In *Proc. of the 2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT'22)*, 2022.

[688] D. Weininger. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988.

[689] M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proc. of the 28th International Conference on International Conference on Machine Learning (ICML'11)*, 2011.

[690] R. W. White. Motivation reconsidered: The concept of competence. *Psychological Review*, 66(5):297–333, 1959.

[691] G. A. Wiggins. Searching for computational creativity. *New Generation Computing*, 24:209–222, 2006.

[692] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

[693] R. J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2): 270–280, 1989.

[694] Y. Wolf, N. Wies, O. Avnery, Y. Levine, and A. Shashua. Fundamental limitations of alignment in large language models. In *Proc. of the 41st International Conference on Machine Learning (ICML'24)*, 2024.

[695] L. Wong, G. Grand, A. K. Lew, N. D. Goodman, V. K. Mansinghka, J. Andreas, and J. B. Tenenbaum. From word models to world models: Translating from natural language to the probabilistic language of thought, 2023. arXiv:2306.12672 [cs.CL].

[696] M. Wooldridge and N. R. Jennings. Intelligent agents: theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.

[697] World Trade Organization. Agreement on trade-related aspects of intellectual property rights, 1994.

[698] H.-H. Wu, P. Seetharaman, K. Kumar, and J. P. Bello. Wav2CLIP: Learning robust audio representations from CLIP. In *Proc. of the 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'22)*, 2022.

[699] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems (NIPS'16)*, volume 29, 2016.

[700] J. Wu, L. Ouyang, D. M. Ziegler, N. Stiennon, R. Lowe, J. Leike, and P. Christiano. Recursively summarizing books with human feedback, 2021. arXiv:2109.10862 [cs.CL].

[701] M. Wu, Y. Yuan, G. Haffari, and L. Wang. (perhaps) beyond human translation: Harnessing multi-agent collaboration for translating ultra-long literary texts, 2024. arXiv:2405.11804 [cs.CL].

[702] S. Wu, H. Fei, L. Qu, W. Ji, and T.-S. Chua. NExT-GPT: Any-to-any multimodal LLM. In *Proc. of the 41st International Conference on Machine Learning (ICML'24)*, 2024.

[703] T. Wu, M. Caccia, Z. Li, Y.-F. Li, G. Qi, and G. Haffari. Pretrained language model in continual learning: A comparative study. In *Proc. of the 10th International Conference on Learning Representations (ICLR'22)*, 2022.

[704] Y. Wu and B. Hu. Learning to extract coherent summary via deep reinforcement learning. In *Proc. of the 32nd AAAI Conference on Artificial Intelligence and 30th Innovative Applications of Artificial Intelligence Conference and 8th AAAI Symposium on Educational Advances in Artificial Intelligence (AAAI'18/IAAI'18/EAAI'18)*, 2018.

[705] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klinger, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, ..., and J. Dean. Google's neural machine translation system: Bridging the gap between human and machine translation, 2016. arXiv:1609.08144 [cs.CL].

[706] Z. Xi, W. Chen, X. Guo, W. He, Y. Ding, B. Hong, M. Zhang, J. Wang, S. Jin, E. Zhou, R. Zheng, X. Fan, X. Wang, L. Xiong, Y. Zhou, W. Wang, C. Jiang, Y. Zou, X. Liu, ..., and T. Gui. The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, 68(2):121101, 2023.

[707] C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, and D. Song. Generating adversarial examples with adversarial networks. In *Proc. of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*, 2018.

[708] N. Xie, H. Hachiya, and M. Sugiyama. Artist agent: A reinforcement learning approach to automatic stroke generation in oriental ink painting. In *Proc. of the 29th International Conference on Machine Learning (ICML'12)*, 2012.

[709] Y. Xie, K. Kawaguchi, Y. Zhao, X. Zhao, M.-Y. Kan, J. He, and Q. Xie. Self-evaluation guided beam search for reasoning. In *Proc. of the*

37th Conference on Neural Information Processing Systems (NIPS'23), 2023.

[710] H. Xu, B. McCane, and L. Szymanski. VASE: Variational assorted surprise exploration for reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 34(3):1243–1252, 2021.

[711] J. Xu, X. Liu, Y. Wu, Y. Tong, Q. Li, M. Ding, J. Tang, and Y. Dong. ImageReward: Learning and evaluating human preferences for text-to-image generation. In *Proc. of the 37th Conference on Neural Information Processing Systems (NIPS'23)*, 2023.

[712] W. Yan, Y. Zhang, P. Abbeel, and A. Srinivas. VideoGPT: Video generation using VQ-VAE and transformers, 2021. arXiv:2104.10157 [cs.CV].

[713] X. Yan, J. Yang, K. Sohn, and H. Lee. Attribute2Image: Conditional image generation from visual attributes. In *Proc. of the 11th European Conference on Computer Vision (ECCV'16)*, 2016.

[714] K. Yang and D. Klein. FUDGE: Controlled text generation with future discriminators. In *Proc. of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL'21)*, 2021.

[715] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, W. Zhang, B. Cui, and M.-H. Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4), 2023.

[716] S. Yanisky-Ravid. Generating rembrandt: Artificial intelligence, copyright, and accountability in the 3A era — the human-like authors are already here — a new model. *Michigan State Law Review*, 2017.

[717] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. R. Narasimhan, and Y. Cao. ReAct: Synergizing reasoning and acting in language models. In *Proc. of the 11th International Conference on Learning Representations (ICLR'23)*, 2023.

[718] C. Ye, A. Khalifa, P. Bontrager, and J. Togelius. Rotation, translation, and cropping for zero-shot generalization. In *Proc. of the 2020 IEEE Conference on Games (CoG'20)*, 2020.

[719] X. Yi, M. Sun, R. Li, and W. Li. Automatic poetry generation with mutual reinforcement learning. In *Proc. of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP'18)*, 2018.

[720] J. You, B. Liu, Z. Ying, V. Pande, and J. Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. In *Advances in Neural Information Processing Systems (NIPS'18)*, 2018.

[721] L. Yu, W. Zhang, J. Wang, and Y. Yu. SeqGAN: Sequence generative adversarial nets with policy gradient. In *Proc. of the 31st AAAI Conference on Artificial Intelligence (AAAI'17)*, 2017.

[722] L. Yu, W. Jiang, H. Shi, J. YU, Z. Liu, Y. Zhang, J. Kwok, Z. Li, A. Weller, and W. Liu. Metamath: Bootstrap your own mathematical questions for large language models. In *Proc. of the 12th International Conference on Learning Representations (ICLR'24)*, 2024.

[723] W. Yuan, R. Y. Pang, K. Cho, X. Li, S. Sukhbaatar, J. Xu, and J. E. Weston. Self-rewarding language models. In *Proc. of the 41st International Conference on Machine Learning (ICML'24)*, 2024.

[724] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. Self-attention generative adversarial networks. In *Proc. of the 36th International Conference on Machine Learning (ICML'19)*, 2019.

[725] K. Zhang, Z. Yang, and T. Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. In *Handbook of Reinforcement Learning and Control*, pages 321–384. Springer International Publishing, 2021.

[726] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. Victoria Lin, T. Mihaylov, M. Ott, S. Shleifer, K. Shuster, D. Simig, P. Singh Koura, A. Sridhar, T. Wang, and L. Zettlemoyer. OPT: Open pre-trained transformer language models, 2022. arXiv:2205.01068 [cs.CL].

[727] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. BERTScore: Evaluating text generation with BERT. In *Proc. of the 8th International Conference on Learning Representations (ICLR'20)*, 2020.

[728] X. Zhang and M. Lapata. Chinese poetry generation with recurrent neural networks. In *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*, 2014.

[729] Y. Zhang, Z. Gan, and L. Carin. Generating text via adversarial training. In *Proc. of the NIPS'16 Workshop on Adversarial Training*, 2016.

[730] Y. Zhang, Z. Gan, K. Fan, Z. Chen, R. Henao, D. Shen, and L. Carin. Adversarial feature matching for text generation. In *Proc. of the 34th International Conference on Machine Learning (ICML'17)*, 2017.

[731] Y. Zhang, X. Li, C. Liu, B. Shuai, Y. Zhu, B. Brattoli, H. Chen, I. Marsic, and J. Tighe. VidTr: Video transformer without convolutions. In *Proc. of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV'21)*, 2021.

[732] H. Zhao, H. Chen, F. Yang, N. Liu, H. Deng, H. Cai, S. Wang, D. Yin, and M. Du. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, 15(2): 20:1–38, 2024.

[733] S. Zhao, J. Song, and S. Ermon. Towards deeper understanding of variational autoencoding models, 2017. arXiv:1702.08658 [cs.LG].

[734] Y. Zhao, R. Zhang, W. Li, D. Huang, J. Guo, S. Peng, Y. Hao, Y. Wen, X. Hu, Z. Du, Q. Guo, L. Li, and Y. Chen. Assessing and understanding creativity in large language models, 2024. arXiv:2401.12491 [cs.CL].

[735] A. Zhavoronkov, Y. A. Ivanenkov, A. Aliper, M. S. Veselov, V. A. Aladinskiy, A. V. Aladinskaya, V. A. Terentiev, D. A. Polykovskiy, M. D. Kuznetsov, A. Asadulaev, Y. Volkov, A. Zholus, R. R. Shayakhmetov, A. Zhebrak, L. I. Minaeva, B. A. Zagribelnyy, L. H. Lee, R. Soll, D. Madge, ..., and A. Aspuru-Guzik. Deep learning enables rapid identification of potent DDR1 kinase inhibitors. *Nature Biotechnology*, 37 (9):1038–1040, 2019.

[736] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Proc. of the 37th Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NIPS'23)*, 2023.

[737] L. Zhou, K. Small, O. Rokhlenko, and C. Elkan. End-to-end offline goal-oriented dialog policy learning via policy gradient. In *Proc. of the NIPS'17 Workshop on Conversational AI*, 2017.

[738] T. Zhou, C. Fang, Z. Wang, J. Yang, B. Kim, Z. Chen, J. Brandt, and D. Terzopoulos. Learning to sketch with Deep Q Networks and demonstrated strokes, 2018. arXiv:1810.05977 [cs.CV].

[739] G. Zhu, M. Zhang, H. Lee, and C. Zhang. Bridging imagination and reality for model-based deep reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS'20)*, 2020.

[740] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. of the 2017 IEEE International Conference on Computer Vision (ICCV'17)*, 2017.

[741] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving. Fine-tuning language models from human preferences, 2019. arXiv:1909.08593 [cs.CL].

[742] A. Zugarini, S. Melacci, and M. Maggini. Neural poetry: Learning to generate poems using syllables. In *Proc. of the 29th International Conference on Artificial Neural Networks (ICANN'19)*, 2019.

A Experiment Details

A.1 Creativity for Reinforcement Learning

For the experiments detailed in Section 4.2, following our limited-resource setting, we adopt smaller neural networks than those used in the original Dreamer papers. Accordingly, we have changed a few of the hyperparameters, while keeping most of them equal to those from DreamerV3. Table A.1 reports the full list of network hyperparameters.

As far as the training is concerned, we leverage mixed precision [441] to reduce resource consumption. Table A.2 reports all the training parameters.

Parameter	**Value**
Categoricals C	32
Classes J	32
RNN hidden units	512
Convolution filters	[32, 64, 128, 256]
Convolution kernel size	4
Convolution strides	2
Deconvolution filters	[128, 64, 32, 3]
Deconvolution kernel size	4
Deconvolution strides	2
Linear units	512
MLP layers	2
Normalization	Layer
Activation	swish
Learning rate during *day*	1e-4
Learning rate at *night*	5e-5
Optimizer	Adam
Reward bins K	255
Bins extremes	-20, +20
Dynamics loss factor β_1	1.0
Representation loss factor β_2	0.1
Critic loss factor c_v	0.5
Entropy loss factor c_e	0.01
γ parameter during *day* (GAE)	0.99
γ parameter at *night* (GAE)	1 - 1/H
λ parameter (GAE)	0.95
PPO clip factor ϵ	0.2
PPO gradient clip factor	0.5
PPO iterations	3

Table A.1: Network hyperparameters.

Parameter	**Value**
Total timesteps in environment	1e+6
Total training steps for world model	1e+7
Total training steps in imagination	1.5e+8
Seed episodes S	8
Warmup epochs for world model	10
Total epochs E	122
World model update steps U	32
Steps per epoch in each environment T_{ep}	1024
Parallelized environments N_{envs}	8
World batch size B_w	100
Sequence length L	25
Agent batch size B_a	2048
Imagination horizon H	16
Test repetition per epoch	8
DeepDream optimization steps	20
DeepDream step size	0.3
Value maximization optimization steps	20
Value maximization step size	0.5

Table A.2: Training parameters.

A.2 Reinforcement Learning for Creativity

For the experiments on poetry generation detailed in Section 5.1.2, Table A.3 reports the full training parameters. The prompt used for generation at training and inference time leverages *Nothing gold can stay* by Robert Frost, *Fame is a bee* by Emily Dickinson, and *Epitaph* by William Carlos Williams for few-shot learning:

Write a fatalistic epigram poem of high, award winning quality.

Nature's first green is gold,
Her hardest hue to hold.
Her early leaf's a flower;
But only so an hour.
Then leaf subsides to leaf.
So Eden sank to grief,
So dawn goes down to day.
Nothing gold can stay.

Write an ironic quatrain poem of high, award winning quality.

Fame is a bee.
It has a song-
It has a sting-
Ah, too, it has a wing.

Write a naturalistic epitaph poem of high, award winning quality.

An old willow with hollow branches
slowly swayed his few high fright tendrils
and sang:

Love is a young green willow
shimmering at the bare wood's edge.

Write a `{tone}` `{style}` of high, award winning quality.

while the prompt used for the computation of $p(S|T)$ is:

```
Describe the style of the following poem in two words:

{prova}

I would describe it as a
```

Parameter	Value
Total batches	100
Batch size B	4
Gradient accumulation steps	8
Max new tokens	256
Temperature	1.
Top-k	50
Optimizer	Adam
Learning rate	1e-5
γ (PPO)	1.
λ (PPO)	0.95
Clip range (PPO)	0.2
Value loss coeff. (PPO)	0.1
PPO epochs	3
KL coeff. (PPO)	0.05
Whiten rewards (PPO)	True
Max gradient normalization	100.
β (DPO)	0.1
Number of generated candidates K (DPO)	4

Table A.3: Training parameters for poetry generation.

On the contrary, Table A.4 reports the full training parameters for math problem resolution. We also adopted the same two different prompts from [722], i.e.:

```
Below is an instruction that describes a task. Write a response that
appropriately completes the request.

### Instruction:
{question}

Response:
```

at training time and

```
Below is an instruction that describes a task. Write a response that
appropriately completes the request.

### Instruction:
{question}

Response: Let's think step by step.
```

at inference time. Instead, for the computation of $p(S|T)$ we use the following:

```
Below is a response that appropriately completes a request. Write the
instruction that describes the task.

### Response:
{response}

Instruction:
```

Parameter	**Value**
Total epochs	1
Batch size B	4
Gradient accumulation steps	8
Max new tokens	512
Temperature	1.
Top-k	50
Optimizer	Adam
Learning rate	1e-6
Reward for correct answer (PPO)	+10.
γ (PPO)	1.
λ (PPO)	0.95
Clip range (PPO)	0.2
Value loss coeff. (PPO)	0.1
PPO epochs	3
KL coeff. (PPO)	0.05
Whiten rewards (PPO)	True
Max gradient normalization	100.
Reward for correct answer (DPO)	+5.
β (DPO)	0.1
Number of generated candidates K (DPO)	4

Table A.4: Training parameters for poetry generation.

A.3 Contextual Learning via Creativity Score

For the experiments described in Section 5.2.2, we generate 4 different outputs for each input, then used in the final prompt as explained by Algorithm 4. The generation configurations we use are: greedy strategy; sampling with a temperature of 0.8 and top-k = 50; sampling with a temperature of 1.0 and top-k = 50; and sampling with a temperature of 1.2 and top-k = 50. For them, we use the task in input as-is.

A.4 DiffSampling

As reported in Section 6.1.2, we test *DiffSampling* on three case studies. For the mathematical problem resolution, we use the two prompts from [722], reported in Appendix A.2.

For the extreme summarization task, the prompt adopted for the instructed version of Llama2-7B is the same as in [110]:

> [INST] For the following article: {`article`}
>
> Return a summary comprising of 1 sentence. With the sentence in a numbered list format.
>
> For example:
>
> 1. First sentence [/INST]

where [INST] and [/INST] are special tokens used by Llama2-7b to identify different roles in the chat.

Vice versa, for the non-instructed version, we use:

> Generate a 1 sentence summary for the given article.
> Article: {`article`}
> Summary:

Finally, for the divergent association task, we consider the following prompt for the instructed version of Llama3-8B:

> `user`
>
> Please write 10 nouns in English that are as irrelevant from each other as possible, in all meanings and uses of the words. Please note that the words you write should have only single word, only nouns (e.g., things, objects, concepts), and no proper nouns (e.g., no specific people or places).
> `assistant`

where `user` and `assistant` are keywords used by Llama3-8b to identify different roles in the chat, while for its non-instructed version we use the following:

Task: Write 10 nouns in English that are as irrelevant from each other as possible, in all meanings and uses of the words. Please note that the words you write should have only single word, only nouns (e.g., things, objects, concepts), and no proper nouns (e.g., no specific people or places).
Solution:

B GutenVerse Dataset

To evaluate the accidental reproduction rate of generated poems, we propose GutenVerse dataset[1], a set of almost approx. 84k public-domain, English-written poems extracted from Project Gutenberg. While generated poems can reproduce different content, e.g., songs or copyrighted materials, we believe this can provide a useful indication of how likely is that a text is original or not.

To define our dataset, we started from *Gutenberg, dammit*[2], a corpus of every plaintext file in Project Gutenberg (up until June 2016). We selected all the text files whose metadata report English as the language, public domain as copyright status, *poetry* among the subjects or *poems* or *poetical work* in the title, and that were not a translation of another book. Then, we applied a series of rules (e.g., about the verse length) to extract the titles and poems from all the selected text files, and we defined our GutenVerse dataset. While it can still contain content that is not poetry (e.g., a table of contents formatted very uncommonly), the poems can be effectively used to measure overlapping between real and generated text. We plan to improve the dataset and release cleaner and safer versions in the future, to allow researchers to use it for other purposes apart from accidental reproduction metrics.

[1] The dataset and the code used to create it can be found at: `https://github.com/giorgiofranceschelli/GutenVerse`

[2] See `https://github.com/aparrish/gutenberg-dammit/`

C Ablation Studies

We report here various sets of ablation studies for the two parameters governing *DiffSampling* (presented in Section 6.1): the lower-bound probability of the critical mass, and the reparameterization scaling factor.

C.1 Ablation Study on the Lower Bound

We conducted experiments on the three aforementioned case studies, varying the lower bound of the critical mass.

Tables C.1, C.2, and C.3 report the results for the math problem resolution considering the training set and the GSM8K and MATH test sets, respectively. As expected, the against-greedy diversity scores and cross-input EAD increase together with the lower bound; instead, while accuracy tends to decrease with higher lower bounds, the differences on the tests set are not significant, and even a quite high value (e.g., 0.8) achieves competitive results.

Tables C.4 and C.5 report the results for the extreme summarization task. Again, against-greedy scores and cross-input EAD are directly correlated with the lower bound; instead, we see no variations in terms of ROUGE-1 and cross-input SIM for the RLHF-instructed model, and slight decreases for the pre-trained model.

Figures C.1 and C.2 report the results for the divergent association task. As we would expect, the DAT score changes almost linearly between that for a lower bound of 0 (that means *DiffSampling-cut*) and 1 (that means *standard* sampling), as we reported in Section 6.1.2. Interestingly, the number of correct answers by the non-instructed model drops quickly, meaning that selecting a token after the first one tends to produce more incorrect answers even if its probability is smaller than 0.1.

DiffSampling-lb	Accuracy ↑	Cross-Input Diversity		Against-Greedy Diversity	
		EAD ↑	SIM ↑	EAD ↑	SIM ↑
lb = 0.0	94.70 ± 0.21	1.66 ± 0.01	0.75 ± 0.01	0.12 ± 0.00	0.22 ± 0.01
lb = 0.1	94.70 ± 0.21	1.66 ± 0.01	0.75 ± 0.01	0.12 ± 0.00	0.22 ± 0.01
lb = 0.2	94.70 ± 0.21	1.66 ± 0.01	0.75 ± 0.01	0.12 ± 0.00	0.22 ± 0.01
lb = 0.3	94.37 ± 0.22	1.66 ± 0.01	0.75 ± 0.02	0.12 ± 0.00	0.23 ± 0.01
lb = 0.4	94.57 ± 0.47	1.66 ± 0.01	0.74 ± 0.01	0.12 ± 0.01	0.22 ± 0.01
lb = 0.5	94.27 ± 0.26	1.65 ± 0.01	0.75 ± 0.01	0.12 ± 0.01	0.23 ± 0.01
lb = 0.6	93.87 ± 0.22	1.66 ± 0.01	0.74 ± 0.01	0.13 ± 0.01	0.25 ± 0.01
lb = 0.7	94.10 ± 0.17	1.66 ± 0.01	0.75 ± 0.02	0.15 ± 0.01	0.28 ± 0.01
lb = 0.8	92.97 ± 0.12	1.67 ± 0.01	0.74 ± 0.01	0.18 ± 0.01	0.32 ± 0.01
lb = 0.9	92.40 ± 0.50	1.68 ± 0.01	0.74 ± 0.02	0.20 ± 0.01	0.35 ± 0.01
lb = 1.0	90.83 ± 0.63	1.70 ± 0.01	0.75 ± 0.01	0.22 ± 0.01	0.37 ± 0.01

Table C.1: Ablation study on the lower-bound value in *DiffSampling-lb* over 3 seeds for the MetaMathQA training dataset in terms of percentage of correct answers and the diversity metrics. Accuracy and cross-input diversity report the mean and standard error over the final score of each run, while against-greedy diversity reports the mean and the 95% confidence interval over the full set of answers.

DiffSampling-lb	Accuracy ↑	Cross-Input Diversity		Against-Greedy Diversity	
		EAD ↑	SIM ↑	EAD ↑	SIM ↑
lb = 0.0	67.10 ± 0.19	1.98 ± 0.00	0.75 ± 0.00	0.15 ± 0.0	0.33 ± 0.01
lb = 0.1	66.46 ± 0.34	1.99 ± 0.00	0.74 ± 0.00	0.15 ± 0.0	0.33 ± 0.01
lb = 0.2	66.46 ± 0.34	1.99 ± 0.00	0.74 ± 0.00	0.15 ± 0.0	0.33 ± 0.01
lb = 0.3	66.79 ± 0.40	1.98 ± 0.00	0.74 ± 0.00	0.15 ± 0.0	0.33 ± 0.01
lb = 0.4	66.57 ± 0.39	2.00 ± 0.00	0.74 ± 0.00	0.15 ± 0.0	0.33 ± 0.01
lb = 0.5	65.78 ± 0.08	1.98 ± 0.00	0.74 ± 0.00	0.16 ± 0.0	0.34 ± 0.01
lb = 0.6	66.67 ± 0.37	1.99 ± 0.00	0.74 ± 0.00	0.17 ± 0.0	0.36 ± 0.01
lb = 0.7	65.58 ± 0.19	2.00 ± 0.00	0.74 ± 0.01	0.19 ± 0.0	0.40 ± 0.01
lb = 0.8	66.87 ± 0.16	2.01 ± 0.00	0.75 ± 0.00	0.22 ± 0.0	0.43 ± 0.01
lb = 0.9	65.18 ± 0.65	2.03 ± 0.01	0.75 ± 0.00	0.24 ± 0.0	0.46 ± 0.01
lb = 1.0	64.87 ± 0.20	2.06 ± 0.00	0.74 ± 0.00	0.27 ± 0.0	0.49 ± 0.01

Table C.2: Ablation study on the lower-bound value in *DiffSampling-lb* over 3 seeds for the GSM8K test dataset in terms of percentage of correct answers and the diversity metrics. Accuracy and cross-input diversity report the mean and standard error over the final score of each run, while against-greedy diversity reports the mean and the 95% confidence interval over the full set of answers.

DiffSampling-lb	Accuracy ↑	Cross-Input Diversity		Against-Greedy Diversity	
		EAD ↑	SIM ↑	EAD ↑	SIM ↑
lb = 0.0	21.06 ± 0.13	5.65 ± 0.01	0.67 ± 0.00	0.27 ± 0.00	0.37 ± 0.00
lb = 0.1	20.95 ± 0.20	5.65 ± 0.01	0.67 ± 0.00	0.27 ± 0.00	0.38 ± 0.00
lb = 0.2	20.95 ± 0.20	5.65 ± 0.01	0.67 ± 0.00	0.27 ± 0.00	0.38 ± 0.00
lb = 0.3	21.30 ± 0.08	5.66 ± 0.01	0.67 ± 0.00	0.27 ± 0.00	0.38 ± 0.00
lb = 0.4	21.08 ± 0.11	5.66 ± 0.02	0.68 ± 0.00	0.28 ± 0.00	0.38 ± 0.00
lb = 0.5	21.18 ± 0.11	5.69 ± 0.01	0.68 ± 0.00	0.29 ± 0.00	0.40 ± 0.00
lb = 0.6	21.18 ± 0.22	5.72 ± 0.02	0.68 ± 0.00	0.31 ± 0.00	0.42 ± 0.00
lb = 0.7	21.14 ± 0.15	5.79 ± 0.01	0.67 ± 0.00	0.33 ± 0.00	0.44 ± 0.00
lb = 0.8	20.91 ± 0.24	5.89 ± 0.01	0.68 ± 0.00	0.35 ± 0.00	0.46 ± 0.00
lb = 0.9	20.20 ± 0.08	6.04 ± 0.02	0.68 ± 0.00	0.37 ± 0.00	0.47 ± 0.00
lb = 1.0	19.46 ± 0.19	6.28 ± 0.01	0.68 ± 0.00	0.39 ± 0.00	0.49 ± 0.00

Table C.3: Ablation study on the lower-bound value in *DiffSampling-lb* over 3 seeds for the MATH test dataset in terms of percentage of correct answers and the diversity metrics. Accuracy and cross-input diversity report the mean and standard error over the final score of each run, while against-greedy diversity reports the mean and the 95% confidence interval over the full set of answers.

DiffSampling-lb	ROUGE-1 ↑	Cross-Input Diversity		Against-Greedy Diversity	
		EAD ↑	SIM ↑	EAD ↑	SIM ↑
lb = 0.0	0.22 ± 0.00	1.16 ± 0.00	0.91 ± 0.00	0.17 ± 0.01	0.22 ± 0.01
lb = 0.1	0.22 ± 0.00	1.17 ± 0.00	0.91 ± 0.00	0.17 ± 0.01	0.22 ± 0.01
lb = 0.2	0.22 ± 0.00	1.17 ± 0.00	0.91 ± 0.00	0.17 ± 0.01	0.22 ± 0.01
lb = 0.3	0.22 ± 0.00	1.18 ± 0.00	0.91 ± 0.00	0.18 ± 0.01	0.22 ± 0.01
lb = 0.4	0.22 ± 0.00	1.18 ± 0.00	0.91 ± 0.00	0.18 ± 0.01	0.22 ± 0.01
lb = 0.5	0.22 ± 0.00	1.18 ± 0.00	0.91 ± 0.01	0.19 ± 0.01	0.23 ± 0.01
lb = 0.6	0.22 ± 0.00	1.18 ± 0.00	0.91 ± 0.00	0.20 ± 0.01	0.25 ± 0.01
lb = 0.7	0.22 ± 0.00	1.19 ± 0.00	0.91 ± 0.01	0.23 ± 0.01	0.29 ± 0.01
lb = 0.8	0.22 ± 0.00	1.20 ± 0.00	0.91 ± 0.00	0.27 ± 0.01	0.33 ± 0.01
lb = 0.9	0.22 ± 0.00	1.21 ± 0.00	0.92 ± 0.01	0.30 ± 0.01	0.37 ± 0.01
lb = 1.0	0.22 ± 0.00	1.22 ± 0.00	0.91 ± 0.00	0.34 ± 0.01	0.40 ± 0.01

Table C.4: Ablation study on the lower-bound value in *DiffSampling-lb* for the RLHF-instructed model over 3 seeds for the XSum dataset in terms of ROUGE-1 and the diversity metrics. The mean and standard error of the final score for each run are reported for cross-input diversity, whereas the mean and the 95% confidence interval for the full set of answers are reported for ROUGE-1 and against-greedy diversity.

DiffSampling-lb	ROUGE-1 ↑	Cross-Input Diversity		Against-Greedy Diversity	
		EAD ↑	SIM ↑	EAD ↑	SIM ↑
lb = 0.0	0.19 ± 0.00	1.13 ± 0.00	0.93 ± 0.00	0.25 ± 0.01	0.28 ± 0.01
lb = 0.1	0.19 ± 0.00	1.13 ± 0.01	0.93 ± 0.00	0.26 ± 0.01	0.29 ± 0.01
lb = 0.2	0.19 ± 0.00	1.11 ± 0.00	0.93 ± 0.00	0.35 ± 0.01	0.40 ± 0.02
lb = 0.3	0.19 ± 0.00	1.11 ± 0.00	0.93 ± 0.00	0.44 ± 0.01	0.50 ± 0.02
lb = 0.4	0.19 ± 0.00	1.11 ± 0.01	0.93 ± 0.00	0.51 ± 0.01	0.57 ± 0.02
lb = 0.5	0.19 ± 0.00	1.10 ± 0.01	0.92 ± 0.00	0.56 ± 0.01	0.62 ± 0.01
lb = 0.6	0.18 ± 0.00	1.10 ± 0.00	0.92 ± 0.00	0.61 ± 0.01	0.67 ± 0.01
lb = 0.7	0.18 ± 0.00	1.14 ± 0.01	0.92 ± 0.01	0.67 ± 0.01	0.72 ± 0.01
lb = 0.8	0.17 ± 0.00	1.15 ± 0.01	0.91 ± 0.01	0.72 ± 0.01	0.75 ± 0.01
lb = 0.9	0.15 ± 0.00	1.17 ± 0.00	0.91 ± 0.01	0.76 ± 0.01	0.79 ± 0.01
lb = 1.0	0.14 ± 0.00	1.21 ± 0.01	0.91 ± 0.00	0.80 ± 0.01	0.83 ± 0.01

Table C.5: Ablation study on the lower-bound value in *DiffSampling-lb* for the pre-trained model over 3 seeds for the XSum dataset in terms of ROUGE-1 and the diversity metrics. The mean and standard error of the final score for each run are reported for cross-input diversity, whereas the mean and the 95% confidence interval for the full set of answers are reported for ROUGE-1 and against-greedy diversity.

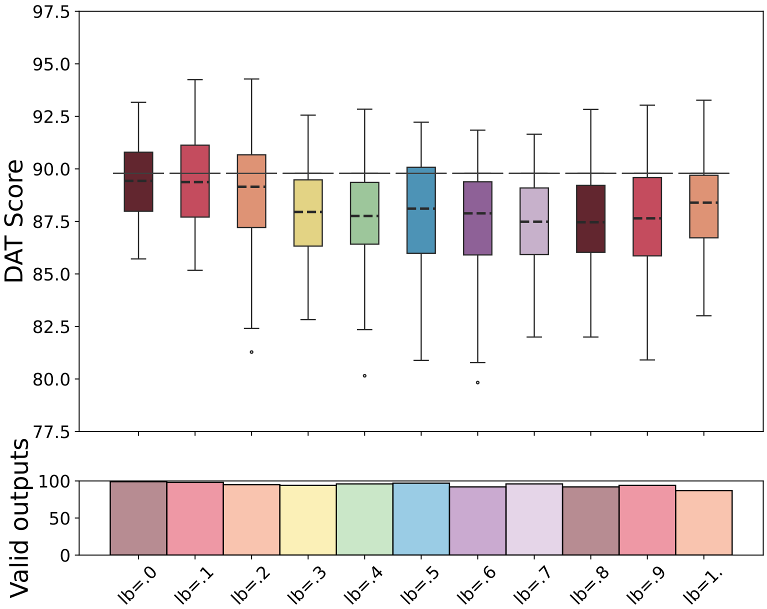


Figure C.1: Above, the DAT score for *DiffSampling-lb* over the instructed version of Llama3-8B when varying the p_{lb} parameter. Below, the number of valid outputs produced by each of them. Single lines represent greedy methods, while boxplots show the performance of stochastic strategies.

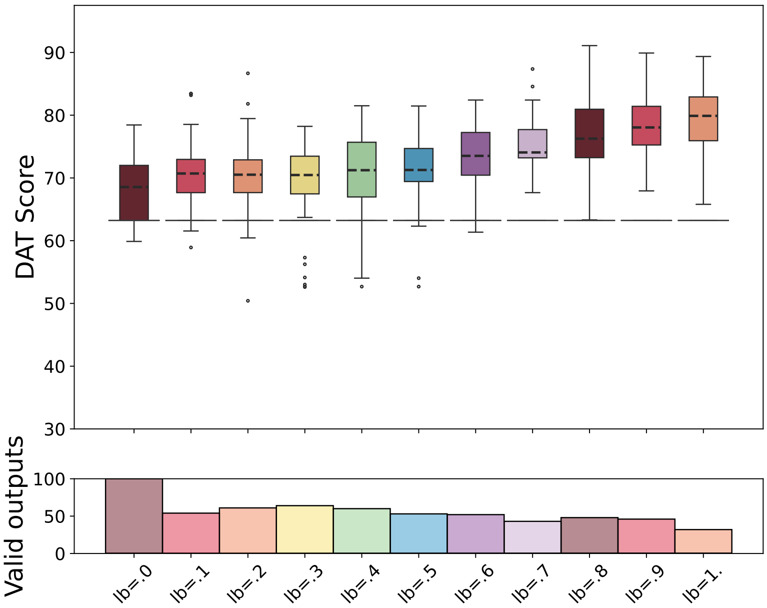


Figure C.2: Above, the DAT score for *DiffSampling-lb* over the pre-trained version of Llama3-8B when varying the p_{lb} parameter. Below, the number of valid outputs produced by each of them. Single lines represent greedy methods, while boxplots show the performance of stochastic strategies.

C.2 Ablation Study on the Gamma Parameter

We also conducted experiments varying the lower bound of the critical mass and the reparameterization factor for xsum and DAT case studies.

Tables C.6 and C.7 report the results for the extreme summarization task considering the RLHF-instructed and pre-trained models, respectively. Despite not seeing significant differences across different reparameterization factors, we see that a higher γ value tends to increase diversity (without altering accuracy) in the first case, while it tends to decrease it in the second one.

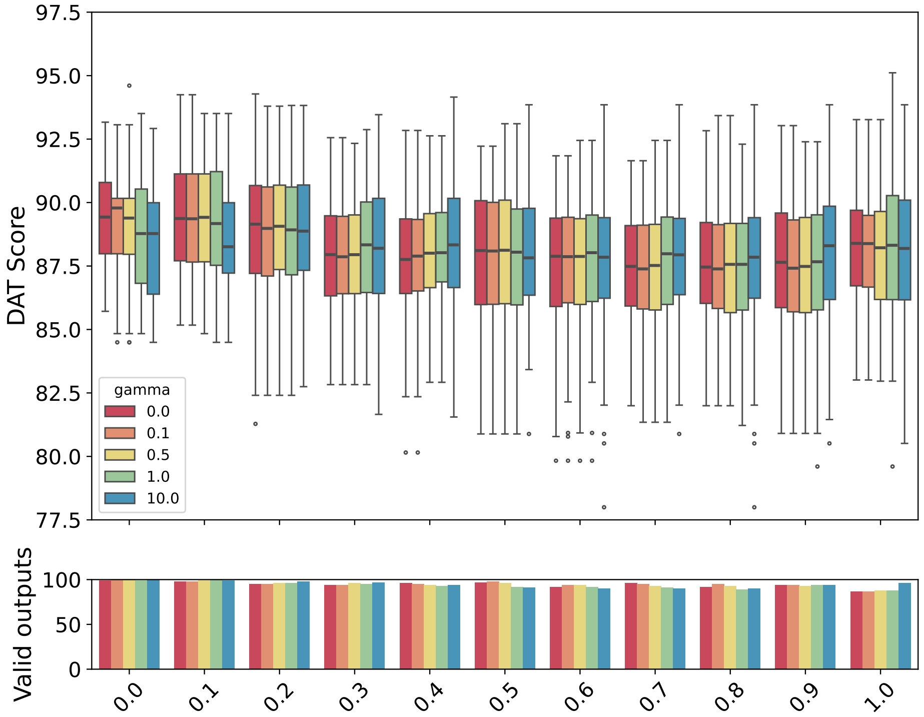


Figure C.3: Above, the DAT score for *DiffSampling-reparam* over the instructed version of Llama3-8B when varying the γ and the p_{lb} parameters. Below, the number of valid outputs produced by each of them. Single lines represent greedy methods, while boxplots show the performance of stochastic strategies.

Finally, Figures C.3 and C.4 report the results for the divergent association task. Coherent with what we saw for xsum, a higher γ value has a positive effect when considering the RLHF-instructed model, with generally higher DAT scores (apart from very low lower bounds) and more valid

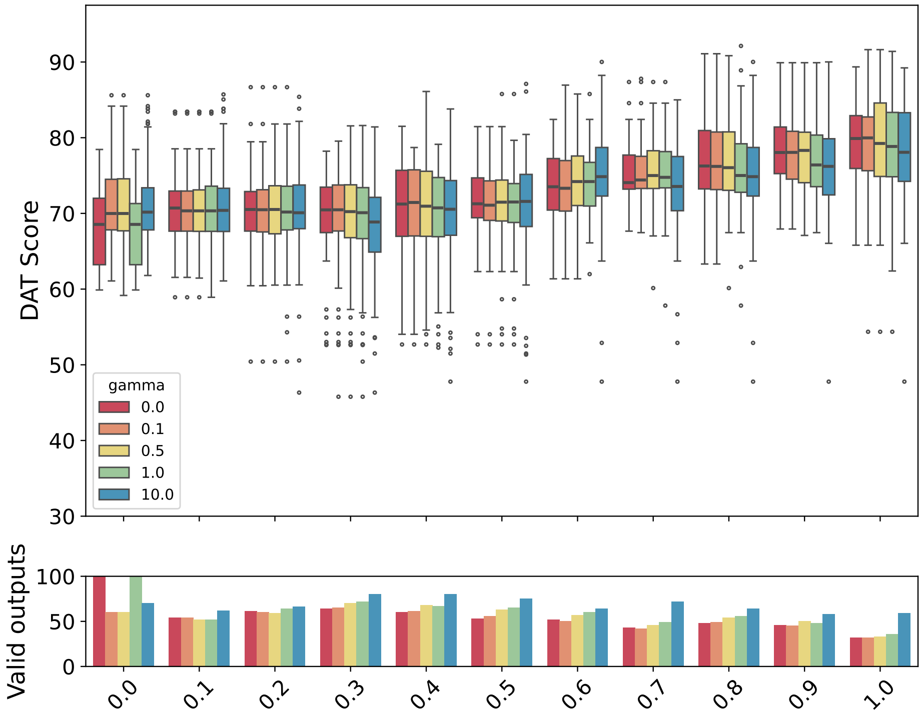


Figure C.4: Above, the DAT score for *DiffSampling-reparam* over the pre-trained version of Llama3-8B when varying the γ and the p_{lb} parameters. Below, the number of valid outputs produced by each of them. Single lines represent greedy methods, while boxplots show the performance of stochastic strategies.

outputs. On the contrary, it tends to decrease the DAT score (while still generating more valid outputs) when considering the pre-trained model.

DiffSampling-reparam	ROUGE-1 ↑	Cross-Input Diversity		Against-Greedy Diversity	
		EAD ↑	SIM ↑	EAD ↑	SIM ↑
$\gamma = 0.0$, lb = 0.0	0.22 ± 0.00	1.16 ± 0.00	0.91 ± 0.00	0.17 ± 0.01	0.22 ± 0.01
$\gamma = 0.1$, lb = 0.0	0.22 ± 0.00	1.16 ± 0.00	0.91 ± 0.00	0.18 ± 0.01	0.22 ± 0.01
$\gamma = 0.5$, lb = 0.0	0.22 ± 0.00	1.16 ± 0.00	0.91 ± 0.00	0.19 ± 0.01	0.24 ± 0.01
$\gamma = 1.0$, lb = 0.0	0.22 ± 0.00	1.17 ± 0.00	0.91 ± 0.00	0.19 ± 0.01	0.24 ± 0.01
$\gamma = 10.$, lb = 0.0	0.22 ± 0.00	1.17 ± 0.00	0.91 ± 0.01	0.21 ± 0.01	0.27 ± 0.01
$\gamma = 0.0$, lb = 0.1	0.22 ± 0.00	1.17 ± 0.00	0.91 ± 0.00	0.17 ± 0.01	0.22 ± 0.01
$\gamma = 0.1$, lb = 0.1	0.22 ± 0.00	1.17 ± 0.00	0.91 ± 0.00	0.17 ± 0.01	0.22 ± 0.01
$\gamma = 0.5$, lb = 0.1	0.22 ± 0.00	1.17 ± 0.00	0.91 ± 0.00	0.18 ± 0.01	0.23 ± 0.01
$\gamma = 1.0$, lb = 0.1	0.22 ± 0.00	1.17 ± 0.00	0.91 ± 0.01	0.19 ± 0.01	0.24 ± 0.01
$\gamma = 10.$, lb = 0.1	0.22 ± 0.00	1.17 ± 0.00	0.91 ± 0.01	0.21 ± 0.01	0.26 ± 0.01
$\gamma = 0.0$, lb = 0.2	0.22 ± 0.00	1.17 ± 0.00	0.91 ± 0.00	0.17 ± 0.01	0.22 ± 0.01
$\gamma = 0.1$, lb = 0.2	0.22 ± 0.00	1.18 ± 0.00	0.91 ± 0.00	0.18 ± 0.01	0.22 ± 0.01
$\gamma = 0.5$, lb = 0.2	0.22 ± 0.00	1.18 ± 0.00	0.91 ± 0.01	0.18 ± 0.01	0.23 ± 0.01
$\gamma = 1.0$, lb = 0.2	0.22 ± 0.00	1.18 ± 0.00	0.91 ± 0.00	0.19 ± 0.01	0.24 ± 0.01
$\gamma = 10.$, lb = 0.2	0.22 ± 0.00	1.18 ± 0.00	0.91 ± 0.00	0.21 ± 0.01	0.26 ± 0.01
$\gamma = 0.0$, lb = 0.3	0.22 ± 0.00	1.18 ± 0.00	0.91 ± 0.00	0.18 ± 0.01	0.22 ± 0.01
$\gamma = 0.1$, lb = 0.3	0.22 ± 0.00	1.18 ± 0.00	0.91 ± 0.00	0.18 ± 0.01	0.22 ± 0.01
$\gamma = 0.5$, lb = 0.3	0.22 ± 0.00	1.18 ± 0.00	0.91 ± 0.01	0.18 ± 0.01	0.23 ± 0.01
$\gamma = 1.0$, lb = 0.3	0.22 ± 0.00	1.18 ± 0.00	0.91 ± 0.01	0.19 ± 0.01	0.24 ± 0.01
$\gamma = 10.$, lb = 0.3	0.22 ± 0.00	1.18 ± 0.00	0.91 ± 0.01	0.21 ± 0.01	0.27 ± 0.01
$\gamma = 0.0$, lb = 0.4	0.22 ± 0.00	1.18 ± 0.00	0.91 ± 0.00	0.18 ± 0.01	0.22 ± 0.01
$\gamma = 0.1$, lb = 0.4	0.22 ± 0.00	1.18 ± 0.00	0.91 ± 0.00	0.18 ± 0.01	0.22 ± 0.01
$\gamma = 0.5$, lb = 0.4	0.22 ± 0.00	1.18 ± 0.00	0.91 ± 0.00	0.19 ± 0.01	0.23 ± 0.01
$\gamma = 1.0$, lb = 0.4	0.22 ± 0.00	1.18 ± 0.00	0.91 ± 0.00	0.19 ± 0.01	0.24 ± 0.01
$\gamma = 10.$, lb = 0.4	0.22 ± 0.00	1.18 ± 0.00	0.91 ± 0.01	0.21 ± 0.01	0.27 ± 0.01
$\gamma = 0.0$, lb = 0.5	0.22 ± 0.00	1.18 ± 0.00	0.91 ± 0.01	0.19 ± 0.01	0.23 ± 0.01
$\gamma = 0.1$, lb = 0.5	0.22 ± 0.00	1.18 ± 0.00	0.92 ± 0.00	0.19 ± 0.01	0.23 ± 0.01
$\gamma = 0.5$, lb = 0.5	0.22 ± 0.00	1.18 ± 0.00	0.92 ± 0.00	0.19 ± 0.01	0.24 ± 0.01
$\gamma = 1.0$, lb = 0.5	0.22 ± 0.00	1.18 ± 0.00	0.92 ± 0.00	0.20 ± 0.01	0.25 ± 0.01
$\gamma = 10.$, lb = 0.5	0.22 ± 0.00	1.19 ± 0.00	0.91 ± 0.01	0.22 ± 0.01	0.27 ± 0.01
$\gamma = 0.0$, lb = 0.6	0.22 ± 0.00	1.18 ± 0.00	0.91 ± 0.00	0.20 ± 0.01	0.25 ± 0.01
$\gamma = 0.1$, lb = 0.6	0.22 ± 0.00	1.18 ± 0.00	0.91 ± 0.00	0.20 ± 0.01	0.25 ± 0.01
$\gamma = 0.5$, lb = 0.6	0.22 ± 0.00	1.18 ± 0.00	0.91 ± 0.00	0.21 ± 0.01	0.26 ± 0.01
$\gamma = 1.0$, lb = 0.6	0.22 ± 0.00	1.18 ± 0.00	0.91 ± 0.00	0.21 ± 0.01	0.26 ± 0.01
$\gamma = 10.$, lb = 0.6	0.22 ± 0.00	1.20 ± 0.00	0.91 ± 0.01	0.29 ± 0.01	0.36 ± 0.01
$\gamma = 0.0$, lb = 0.7	0.22 ± 0.00	1.19 ± 0.00	0.91 ± 0.01	0.23 ± 0.01	0.29 ± 0.01
$\gamma = 0.1$, lb = 0.7	0.22 ± 0.00	1.19 ± 0.00	0.91 ± 0.00	0.23 ± 0.01	0.29 ± 0.01
$\gamma = 0.5$, lb = 0.7	0.22 ± 0.00	1.19 ± 0.00	0.91 ± 0.00	0.24 ± 0.01	0.30 ± 0.01
$\gamma = 1.0$, lb = 0.7	0.22 ± 0.00	1.19 ± 0.00	0.91 ± 0.00	0.24 ± 0.01	0.30 ± 0.01
$\gamma = 10.$, lb = 0.7	0.22 ± 0.00	1.19 ± 0.00	0.91 ± 0.00	0.27 ± 0.01	0.33 ± 0.01
$\gamma = 0.0$, lb = 0.8	0.22 ± 0.00	1.20 ± 0.00	0.91 ± 0.00	0.27 ± 0.01	0.33 ± 0.01
$\gamma = 0.1$, lb = 0.8	0.22 ± 0.00	1.20 ± 0.00	0.92 ± 0.00	0.27 ± 0.01	0.33 ± 0.01
$\gamma = 0.5$, lb = 0.8	0.22 ± 0.00	1.20 ± 0.00	0.92 ± 0.00	0.27 ± 0.01	0.34 ± 0.01
$\gamma = 1.0$, lb = 0.8	0.22 ± 0.00	1.20 ± 0.00	0.91 ± 0.00	0.28 ± 0.01	0.34 ± 0.01
$\gamma = 10.$, lb = 0.8	0.22 ± 0.00	1.20 ± 0.00	0.91 ± 0.00	0.29 ± 0.01	0.36 ± 0.01
$\gamma = 0.0$, lb = 0.9	0.22 ± 0.00	1.21 ± 0.00	0.92 ± 0.01	0.30 ± 0.01	0.37 ± 0.01
$\gamma = 0.1$, lb = 0.9	0.22 ± 0.00	1.21 ± 0.00	0.92 ± 0.00	0.31 ± 0.01	0.37 ± 0.01
$\gamma = 0.5$, lb = 0.9	0.22 ± 0.00	1.21 ± 0.00	0.92 ± 0.00	0.30 ± 0.01	0.37 ± 0.01
$\gamma = 1.0$, lb = 0.9	0.22 ± 0.00	1.20 ± 0.00	0.92 ± 0.00	0.31 ± 0.01	0.37 ± 0.01
$\gamma = 10.$, lb = 0.9	0.22 ± 0.00	1.20 ± 0.00	0.92 ± 0.00	0.32 ± 0.01	0.39 ± 0.01
$\gamma = 0.0$, lb = 1.0	0.22 ± 0.00	1.22 ± 0.00	0.91 ± 0.00	0.34 ± 0.01	0.40 ± 0.01
$\gamma = 0.1$, lb = 1.0	0.22 ± 0.00	1.22 ± 0.00	0.91 ± 0.00	0.34 ± 0.01	0.41 ± 0.01
$\gamma = 0.5$, lb = 1.0	0.22 ± 0.00	1.22 ± 0.00	0.91 ± 0.00	0.34 ± 0.01	0.41 ± 0.01
$\gamma = 1.0$, lb = 1.0	0.22 ± 0.00	1.22 ± 0.00	0.91 ± 0.00	0.34 ± 0.01	0.41 ± 0.01
$\gamma = 10.$, lb = 1.0	0.22 ± 0.00	1.21 ± 0.01	0.92 ± 0.00	0.35 ± 0.01	0.42 ± 0.01

Table C.6: Ablation study on the gamma parameter and the lower-bound value in *DiffSampling-reparam* for the RLHF-tuned model over 3 seeds for the XSum dataset in terms of ROUGE-1 and the diversity metrics. The mean and standard error of the final score for each run are reported for cross-input diversity, whereas the mean and the 95% confidence interval for the full set of answers are reported for ROUGE-1 and against-greedy diversity.

DiffSampling-reparam	ROUGE-1 ↑	Cross-Input Diversity		Against-Greedy Diversity	
		EAD ↑	SIM ↑	EAD ↑	SIM ↑
$\gamma = 0.0$, lb $= 0.0$	0.19 ± 0.00	1.13 ± 0.00	0.93 ± 0.00	0.25 ± 0.01	0.28 ± 0.01
$\gamma = 0.1$, lb $= 0.0$	0.19 ± 0.00	1.13 ± 0.00	0.93 ± 0.00	0.26 ± 0.01	0.29 ± 0.01
$\gamma = 0.5$, lb $= 0.0$	0.19 ± 0.00	1.13 ± 0.00	0.93 ± 0.00	0.27 ± 0.01	0.30 ± 0.01
$\gamma = 1.0$, lb $= 0.0$	0.19 ± 0.00	1.13 ± 0.00	0.93 ± 0.00	0.28 ± 0.01	0.31 ± 0.01
$\gamma = 10.$, lb $= 0.0$	0.19 ± 0.00	1.14 ± 0.01	0.93 ± 0.00	0.31 ± 0.01	0.35 ± 0.01
$\gamma = 0.0$, lb $= 0.1$	0.19 ± 0.00	1.13 ± 0.01	0.93 ± 0.00	0.26 ± 0.01	0.29 ± 0.01
$\gamma = 0.1$, lb $= 0.1$	0.19 ± 0.00	1.13 ± 0.00	0.93 ± 0.00	0.26 ± 0.01	0.30 ± 0.01
$\gamma = 0.5$, lb $= 0.1$	0.19 ± 0.00	1.13 ± 0.01	0.93 ± 0.00	0.27 ± 0.01	0.31 ± 0.01
$\gamma = 1.0$, lb $= 0.1$	0.19 ± 0.00	1.13 ± 0.01	0.93 ± 0.00	0.28 ± 0.01	0.32 ± 0.01
$\gamma = 10.$, lb $= 0.1$	0.19 ± 0.00	1.14 ± 0.01	0.93 ± 0.00	0.32 ± 0.01	0.36 ± 0.01
$\gamma = 0.0$, lb $= 0.2$	0.19 ± 0.00	1.11 ± 0.00	0.93 ± 0.00	0.35 ± 0.01	0.40 ± 0.02
$\gamma = 0.1$, lb $= 0.2$	0.19 ± 0.00	1.12 ± 0.00	0.93 ± 0.00	0.36 ± 0.01	0.41 ± 0.02
$\gamma = 0.5$, lb $= 0.2$	0.19 ± 0.00	1.12 ± 0.00	0.93 ± 0.00	0.36 ± 0.01	0.41 ± 0.02
$\gamma = 1.0$, lb $= 0.2$	0.19 ± 0.00	1.13 ± 0.00	0.93 ± 0.00	0.36 ± 0.01	0.41 ± 0.02
$\gamma = 10.$, lb $= 0.2$	0.19 ± 0.00	1.14 ± 0.00	0.93 ± 0.00	0.39 ± 0.01	0.44 ± 0.02
$\gamma = 0.0$, lb $= 0.3$	0.19 ± 0.00	1.11 ± 0.00	0.93 ± 0.00	0.44 ± 0.01	0.50 ± 0.02
$\gamma = 0.1$, lb $= 0.3$	0.19 ± 0.00	1.11 ± 0.00	0.93 ± 0.00	0.44 ± 0.01	0.49 ± 0.02
$\gamma = 0.5$, lb $= 0.3$	0.19 ± 0.00	1.11 ± 0.00	0.93 ± 0.00	0.44 ± 0.01	0.49 ± 0.02
$\gamma = 1.0$, lb $= 0.3$	0.19 ± 0.00	1.11 ± 0.00	0.93 ± 0.00	0.44 ± 0.01	0.50 ± 0.02
$\gamma = 10.$, lb $= 0.3$	0.19 ± 0.00	1.12 ± 0.01	0.92 ± 0.00	0.44 ± 0.01	0.51 ± 0.02
$\gamma = 0.0$, lb $= 0.4$	0.19 ± 0.00	1.11 ± 0.01	0.93 ± 0.00	0.51 ± 0.01	0.57 ± 0.02
$\gamma = 0.1$, lb $= 0.4$	0.19 ± 0.00	1.11 ± 0.01	0.92 ± 0.00	0.51 ± 0.01	0.57 ± 0.02
$\gamma = 0.5$, lb $= 0.4$	0.19 ± 0.00	1.11 ± 0.01	0.92 ± 0.00	0.51 ± 0.01	0.57 ± 0.02
$\gamma = 1.0$, lb $= 0.4$	0.19 ± 0.00	1.11 ± 0.01	0.92 ± 0.00	0.50 ± 0.01	0.57 ± 0.02
$\gamma = 10.$, lb $= 0.4$	0.19 ± 0.00	1.12 ± 0.01	0.92 ± 0.01	0.49 ± 0.01	0.56 ± 0.02
$\gamma = 0.0$, lb $= 0.5$	0.19 ± 0.00	1.10 ± 0.01	0.92 ± 0.00	0.56 ± 0.01	0.62 ± 0.01
$\gamma = 0.1$, lb $= 0.5$	0.19 ± 0.00	1.11 ± 0.00	0.93 ± 0.00	0.56 ± 0.01	0.62 ± 0.01
$\gamma = 0.5$, lb $= 0.5$	0.19 ± 0.00	1.11 ± 0.00	0.93 ± 0.00	0.55 ± 0.01	0.61 ± 0.01
$\gamma = 1.0$, lb $= 0.5$	0.19 ± 0.00	1.12 ± 0.00	0.93 ± 0.00	0.54 ± 0.01	0.60 ± 0.01
$\gamma = 10.$, lb $= 0.5$	0.19 ± 0.00	1.12 ± 0.00	0.93 ± 0.00	0.52 ± 0.01	0.59 ± 0.02
$\gamma = 0.0$, lb $= 0.6$	0.18 ± 0.00	1.10 ± 0.00	0.92 ± 0.00	0.61 ± 0.01	0.67 ± 0.01
$\gamma = 0.1$, lb $= 0.6$	0.18 ± 0.00	1.10 ± 0.00	0.92 ± 0.00	0.61 ± 0.01	0.66 ± 0.01
$\gamma = 0.5$, lb $= 0.6$	0.18 ± 0.00	1.11 ± 0.00	0.92 ± 0.00	0.60 ± 0.01	0.66 ± 0.01
$\gamma = 1.0$, lb $= 0.6$	0.18 ± 0.00	1.11 ± 0.00	0.92 ± 0.00	0.59 ± 0.01	0.65 ± 0.01
$\gamma = 10.$, lb $= 0.6$	0.18 ± 0.00	1.15 ± 0.01	0.92 ± 0.00	0.63 ± 0.01	0.68 ± 0.01
$\gamma = 0.0$, lb $= 0.7$	0.18 ± 0.00	1.14 ± 0.01	0.92 ± 0.01	0.67 ± 0.01	0.72 ± 0.01
$\gamma = 0.1$, lb $= 0.7$	0.18 ± 0.00	1.13 ± 0.01	0.92 ± 0.01	0.67 ± 0.01	0.72 ± 0.01
$\gamma = 0.5$, lb $= 0.7$	0.18 ± 0.00	1.13 ± 0.00	0.92 ± 0.01	0.65 ± 0.01	0.70 ± 0.01
$\gamma = 1.0$, lb $= 0.7$	0.18 ± 0.00	1.13 ± 0.00	0.92 ± 0.00	0.65 ± 0.01	0.69 ± 0.01
$\gamma = 10.$, lb $= 0.7$	0.18 ± 0.00	1.14 ± 0.00	0.92 ± 0.00	0.60 ± 0.01	0.65 ± 0.01
$\gamma = 0.0$, lb $= 0.8$	0.17 ± 0.00	1.15 ± 0.01	0.91 ± 0.01	0.72 ± 0.01	0.75 ± 0.01
$\gamma = 0.1$, lb $= 0.8$	0.17 ± 0.00	1.15 ± 0.00	0.91 ± 0.00	0.72 ± 0.01	0.76 ± 0.01
$\gamma = 0.5$, lb $= 0.8$	0.17 ± 0.00	1.16 ± 0.00	0.91 ± 0.01	0.71 ± 0.01	0.75 ± 0.01
$\gamma = 1.0$, lb $= 0.8$	0.17 ± 0.00	1.16 ± 0.00	0.92 ± 0.01	0.70 ± 0.01	0.73 ± 0.01
$\gamma = 10.$, lb $= 0.8$	0.18 ± 0.00	1.15 ± 0.01	0.92 ± 0.01	0.64 ± 0.01	0.69 ± 0.01
$\gamma = 0.0$, lb $= 0.9$	0.15 ± 0.00	1.17 ± 0.00	0.91 ± 0.01	0.76 ± 0.01	0.79 ± 0.01
$\gamma = 0.1$, lb $= 0.9$	0.16 ± 0.00	1.16 ± 0.00	0.91 ± 0.00	0.76 ± 0.01	0.79 ± 0.01
$\gamma = 0.5$, lb $= 0.9$	0.16 ± 0.00	1.16 ± 0.00	0.92 ± 0.00	0.74 ± 0.01	0.78 ± 0.01
$\gamma = 1.0$, lb $= 0.9$	0.16 ± 0.00	1.16 ± 0.00	0.91 ± 0.01	0.73 ± 0.01	0.77 ± 0.01
$\gamma = 10.$, lb $= 0.9$	0.18 ± 0.00	1.16 ± 0.01	0.92 ± 0.00	0.66 ± 0.01	0.70 ± 0.01
$\gamma = 0.0$, lb $= 1.0$	0.14 ± 0.00	1.21 ± 0.01	0.91 ± 0.00	0.80 ± 0.01	0.83 ± 0.01
$\gamma = 0.1$, lb $= 1.0$	0.14 ± 0.00	1.21 ± 0.01	0.92 ± 0.00	0.80 ± 0.01	0.82 ± 0.01
$\gamma = 0.5$, lb $= 1.0$	0.14 ± 0.00	1.20 ± 0.01	0.92 ± 0.00	0.78 ± 0.01	0.81 ± 0.01
$\gamma = 1.0$, lb $= 1.0$	0.15 ± 0.00	1.17 ± 0.01	0.91 ± 0.01	0.77 ± 0.01	0.80 ± 0.01
$\gamma = 10.$, lb $= 1.0$	0.17 ± 0.00	1.15 ± 0.01	0.92 ± 0.00	0.69 ± 0.01	0.72 ± 0.01

Table C.7: Ablation study on the gamma parameter and the lower-bound value in *DiffSampling-reparam* for the pre-trained model over 3 seeds for the XSum dataset in terms of ROUGE-1 and the diversity metrics. The mean and standard error of the final score for each run are reported for cross-input diversity, whereas the mean and the 95% confidence interval for the full set of answers are reported for ROUGE-1 and against-greedy diversity.