Alma Mater Studiorum - Università di Bologna
in cotutela con Mines ParisTech

DOTTORATO DI RICERCA IN

AUTOMOTIVE PER UNA MOBILITÀ INTELLIGENTE

Ciclo 34

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

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

EXPLORING DOMAIN-INFORMED AND PHYSICS-GUIDED LEARNING IN IMAGE-TO-IMAGE TRANSLATION

Presentata da: Fabio Pizzati

Coordinatore Dottorato	**Supervisore**
Nicolò Cavina	Raoul de Charette
	Supervisore
	Andrea Prati

Esame finale anno 2022

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à Mines Paris - PSL
Dans le cadre d'une cotutelle avec University of Bologna

Exploring domain-informed and physics-guided learning in image-to-image translation

Candidate:
Fabio PIZZATI

Supervision:

Raoul DE CHARETTE
CR, HdR, Inria — *Supervisor*

Andrea PRATI
Prof., University of Parma — *Co-supervisor*

UNIVERSITÉ FRANCO ITALIENNE | UNIVERSITÀ ITALO FRANCESE

Contents

Chapter 1

Introduction

During the last few years, the emerging capabilities of deep neural networks have fascinated the computer vision community. The increasingly complex and refined representations learned from the ever increasing datasets led to huge progress not only in *understanding* images, but also in *generating* new ones similar to those used for training, thanks mostly to adversarial learning (Goodfellow et al., 2014). In particular, image-to-image translation (i2i) networks gained traction in recent years. The goal of i2i is to make source images visually similar to target ones, without modifying the elements present in the source scene. More formally, this translates to mapping images from a "source" image distribution (or "domain", see Sec. 1.1) to the appearance of those in a "target" one, thus learning a source $\mapsto$ target mapping in presence of differences between training distributions (also called "domain shift", Sec. 1.1).

Interestingly, the learned mapping can benefit several fields, for instance image editing, in which complex image modifications that normally require expert knowledge could be performed automatically (Liu et al., 2019a). Virtual reality applications can also build on i2i, to make synthetic renderings look more realistic (Wei et al., 2019). In robotics, i2i networks are used for aligning training data to deployment conditions, generating for instance realistic images from synthetic ones, ultimately reducing data annotation costs or providing an alternative way of testing performances in realistic scenarios (Murez et al., 2018). This is also related to autonomous driving, since it is unrealistic to collect all possible weather or lighting condition outdoor, hence generated data could be used to achieve a robust perception in adverse weather and non-trivial illumination.

However, generating realistic output scenes usually requires training on large amounts of images, that could be unavailable or expensive to collect. Moreover, results could be unsatisfying for humans due to our better contextual understanding; while we assess the realism of a scene relying on a background of lifelong information, i2i networks may fall into unrealistic

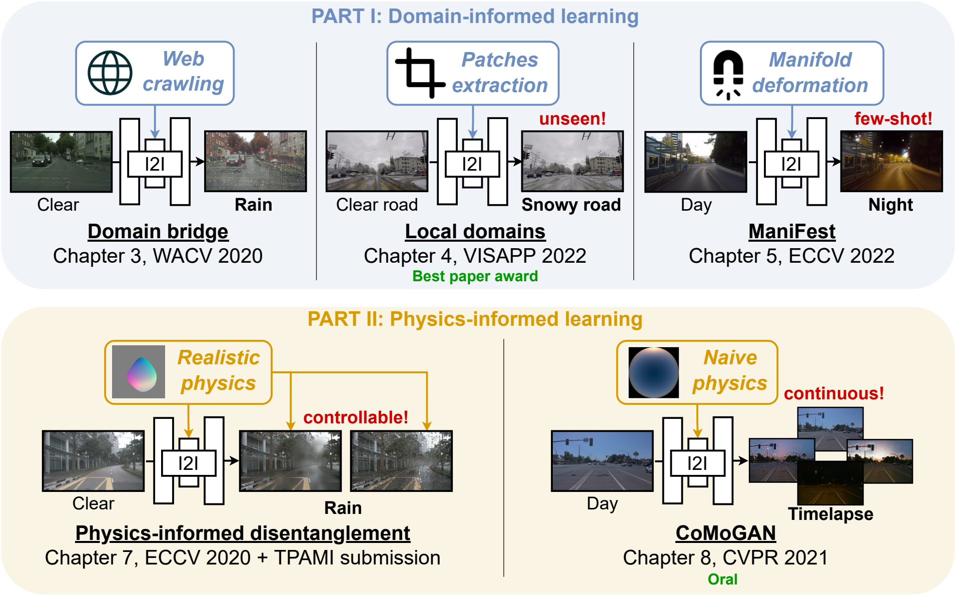


Figure 1.1: Overview of the thesis contributions, with sample outputs of each project. In Part I, we crawl new images from the web to train i2i using domain-level information (Chapter 3), extract patches capturing local domains (Chapter 4), and perform robust few-shot translation thanks to the deformation of a manifold learned on additional domains (Chapter 5). In Part II, we use realistic physical models to obtain disentangled controllable outputs (Chapter 7), or naive models to achieve continuous transformations (Chapter 8).

solutions due to the absence of necessary priors, such as basic physics knowledge. In this thesis, novel training strategies for i2i translation are studied and developed in order to solve these problems. In particular, Part I is dedicated to methodologies relying on human interpretation of data distributions. This enables to exploit more efficiently the images available for training. In Part II, instead, i2i networks are complemented with physics knowledge, to perform informed inferences when physics impacts significantly the appearance of the target condition. Overall, with the thesis we hope to open new directions in image translation research towards the inclusion of human-identified or physical priors.

Thesis organization. In this part of the thesis, the concepts needed for understanding the work are framed. In Sec. 1.1 and Sec. 1.2, preliminary notions and relevant papers are introduced. Part I is dedicated to methodologies for i2i translation relying on human interpretation of data, as introduced in Chapter 2. In Chapter 3, a web-crawling balancing strategy is proposed for i2i, along with additional contributions to benefit feature transfer to novel scenarios. In Chapter 4, we focused on the extraction of local relationships between image patches, to hallucinate novel scenarios unseen during training. Finally, a few-shot image translation strategy is presented in Chapter 5. Part II investigates instead physics priors for i2i, as introduced in Chapter 6. The approaches developed were used to combine realistic physical models with i2i networks in Chapter 7, or to drive a manifold reorganization with naive models in Chapter 8. The thesis is concluded in Chapter 9.

Research context. The thesis (2019-2022) was carried out in the Inria ASTRA team (previously RITS), under a joint co-tutelle agreement between Mines Paris - PSL (France) and the University of Bologna (Italy). It was financed by Vislab/Ambarella (Italy/USA) and Inria (France). We applied most of the methodologies developed to outdoor robotics-oriented scenarios since autonomous driving is the main research topic of all involved partners. Nevertheless, as demonstrated in many of the published papers, we also present other applications considering general scene editing tasks.

1.1 Dataset bias and domain shift

In the deep learning era, the availability of large scale datasets became crucial to extract meaningful representations of the visual world, allowing to reach unprecedented performances in computer vision tasks. However, neural networks still suffer from bias problems related to the learning process. Indeed, training any statistical model (such as neural networks) on a dataset, biases the latter on the data distribution used for training. In other words, the model makes assumptions on the input, which, if violated, result in loss of performance and unexpected outputs. This is undesirable, but very complex to mitigate, since it is unrealistic to assume capturing the variety of the visual world with a finite collection of images. Although recent approaches scaled the training set size to billions of examples (Zhai et al., 2022), datasets biases persist. Their effects are quantified in a seminal paper (Torralba and Efros, 2011), in which the best machine learning pipeline available at the time was trained for object recognition on several datasets, and quantified in terms of performance degradation on other datasets. Even though performances were satisfying when evaluated on the test set, a loss of accuracy was very evident when other data were used to evaluate the obtained model.

The causes of this bias could be related to different factors, including the type of represented scene, physical photography settings, task-related elements and complementary representations of the world (Torralba and Efros, 2011). Recent efforts also extend this decomposition for explaining intra-dataset diversities between training and test sets (Ye et al., 2022).

The concept of dataset bias is tightly related to the “domain” one, quite common in the field when it comes to mitigation strategies for the dataset bias. The field of "domain adaptation" focus indeed on finding strategies for adapting the performances of downstream networks trained on a "source" domain, to a "target" one, which typically encompasses unlabeled examples. This could be seen as a particular application of transfer learning (Csurka, 2017). We could also see domain adaptation as a strategy for mitigating the dataset bias resulting from training on "source". Similarly, in domain generalization the idea is to be as robust as possible on unseen data distributions (Wang et al., 2022).

All things considered, it is possible to introduce a comprehensive definition of "domain" which will be of great importance in the thesis. Hence, a "domain" is a set of characteristics that are common in an image collection. For instance, autonomous driving datasets such as Cityscapes (Cordts et al., 2016), ACDC (Sakaridis et al., 2021) or IDD (Varma et al., 2019) may all belong to the "street scenario" domain, while they could also be in a "urban scene" or "countryside scene" domain, or "daytime", "nighttime" and "twilight", considering acquisition conditions, and again "clear weather", "rainy", "snowy" for atmospheric conditions. In this thesis, we dedicated considerable attention to time-of-day and weather-related domains, due to their physics-based visual properties. Also, it is worth noting that every dataset could be identified with one domain, while a sufficiently broad domain could include multiple datasets. Finally, several domains could be identified in subsets of a dataset.

The definition provided is willingly wide-ranging, since in this thesis the identification of visual domains is meant to identify possible data relationships, rather than exactly defining all the reasons behind a dataset bias. In the same way, relationships among domains are part of the "domain shift", *i.e.* the set of characteristics that make two domains different. We can refer to the domain shift as an alternate interpretation of the effects of the dataset bias, in such a way that in presence of a domain shift, networks behave differently.

1.2 Image-to-image translation

To understand effectively the techniques used in the thesis, let us also consider a brief technical introduction and literature review about image-to-image translation. Even before deep learning, generating new images was a challenging goal of computer vision. Many approaches exploited relationships within existing images to automatically compose novel scenes, mostly applying methodologies for similarity identification such as nearest neighbors (Barnes et al., 2009; Efros and Leung, 1999; Pritch et al., 2009). With the advent of deep learning, Generative Adversarial Networks, or GANs (Goodfellow et al., 2014) dramatically improved the state-of-the-art for image generation, by exploiting alternate training of two deep neural networks. The concept behind GANs is to enforce a zero-sum game between a generator (G) and a discriminator (D) network. While the task of (D) is to distinguish between real images of a dataset and the fake ones (*i.e.* generated by G), the generator is optimized to fool the discriminator. This, at convergence, brings the outputs of G to be similar to the ones of the target dataset used for training.

As already introduced, image-to-image translation could be interpreted as an application of image generation conditioned on an input image. The first general GAN-based approach for i2i was proposed in Pix2pix (Isola et al., 2017), exploiting both adversarial and reconstruction constraints. Since then, the field benefited from multiple improvements. An incremental work (Wang et al., 2018) makes use of multi-scale training to increase the resolution of generated scenes. However, as (Isola et al., 2017), they still required paired datasets for training, *i.e.* datasets in which the same scene is represented in both source and target domains. A major advancement in the field has been the introduction of cycle consistency (Zhu et al., 2017a), a mechanism exploiting a cyclic transformation between domains (*i.e.* source $\mapsto$ target $\mapsto$ source) that enabled training on any unpaired source and target domain. The idea of image-to-image translation between source and target is extended in several works, by enabling transformations across multiple domains (Choi et al., 2018, 2020) or to generate different styles in the target domain itself, *i.e.* performing multimodal image translation (Huang et al., 2018b). Additional relevant literature about image-to-image translation is included in every chapter of the thesis.

Part I

Domain-informed learning

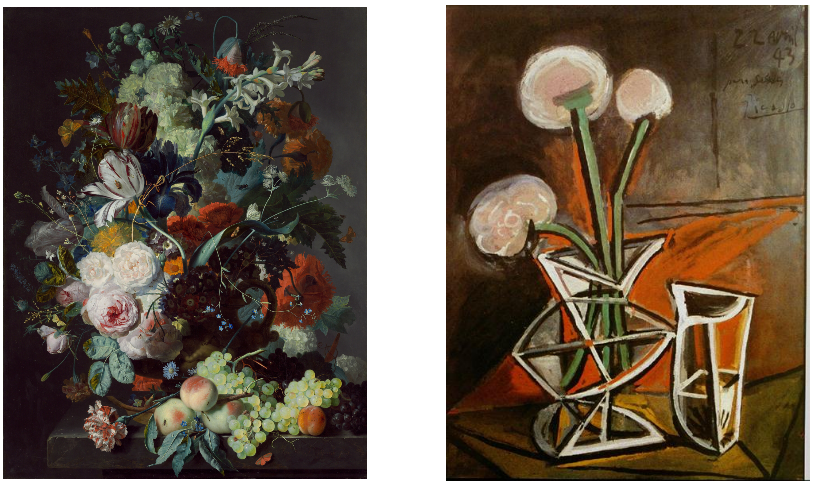
Chapter 2

Part introduction

Deep neural networks can extract meaningful patterns from large quantity of data. Yet, it was shown that human-extracted priors can significantly reduce the number of needed examples to achieve convergence, and boost performance (Deng et al., 2020; Von Rueden et al., 2021). This opens possibilities for different human-based priors injection. For instance, it is possible to embed expert knowledge in the task, achieving application-aware networks (Jumper et al., 2021). Differently, active learning (Settles, 2009) alternates human and networks inferences to achieve robust outputs.

In image-to-image translation, the problem formulation already requires human-gathered information to get a meaningful separation between source and target data, such that there is some gap in the data distribution that a network could learn. In other words, we are exploiting our ability to recognize differences and similarities in visual data — a result of the incredible abstraction capability of the human brain! — to perform a clustering of images in which a gap between data distributions emerges. Some efforts have been dedicated to do this automatically (Baek et al., 2021), but with limited effectiveness, and only on low-level features such as color, ignoring more complex characteristics. Ultimately, modern architectures still naively exploit a binary source and target distinction.

Now, we can envisage using more efficiently the human capability of distinguishing visual characteristics, or *domains* (see Sec. 1.1). Let us consider a simple example. In Fig. 2.1, we observe two masterpieces of van Huysum (Fig. 2.1a) and Picasso (Fig. 2.1b). It is fairly easy to identify differences between the two. Evidently, the absence of realism and perspective characterizes Picasso's work, while in van Huysum's art those features are essential. Nevertheless, and more surprisingly, it is also trivial to identify that both arts represent a vase of flowers. In other words, both paintings could be considered as related to the "vase of flowers" domain, while they also belong to, among others, exclusive domains: *e.g.* "realistic" for Fig. 2.1a, "unrealistic" for Fig. 2.1b. Also, we can complement this basic domain knowledge with



(a) Still Life with Flowers and Fruit, J. van Huysum, 1715

(b) Vase with Flowers, P. Picasso, 1945

Figure 2.1: Visual domains are immediately identifiable by humans in art, in which differences in two representations of flowers like (a) and (b) are immediately distinguishable in terms of style, colors, and number of elements. Nonetheless, we can also identify similarities in the type of subject and context.

expert art priors, considering that they both belong to a "still life" domain, which we can relate with other works of different painters.

As illustrated with this example, humans are able to identify multiple relationships among visual data, exploiting contextual understanding, prior knowledge, and generalization capacities learned across our life. Let us then think again about computer vision. The crucial point of this illustration is that we could design *domain-informed learning pipelines* that embed additional human knowledge about domains, to exploit better human capabilities of distinguishing multiple relationships among data. For example, in Chapter 3 we propose a domain decomposition interpretation that have several applications in the thesis. This enables web-crawling additional data to perform image translation on large domain shifts, thanks to exclusive sub-domains identified in source and target (in the example in Fig. 2.1, being "realistic" or "unrealistic"). We extend a similar idea but introducing a local-domains translation network in Chapter 4, able to exploit local differences in single images identifiable by humans thanks to contextual and cultural priors. For instance, looking at Fig. 2.1a, we notice that we can easily distinguish between the appearance of different objects, as flowers and fruits. Finally, we show how relationships of domains of similar nature can be leveraged to perform few-shot learning in Chapter 5.

Chapter 3

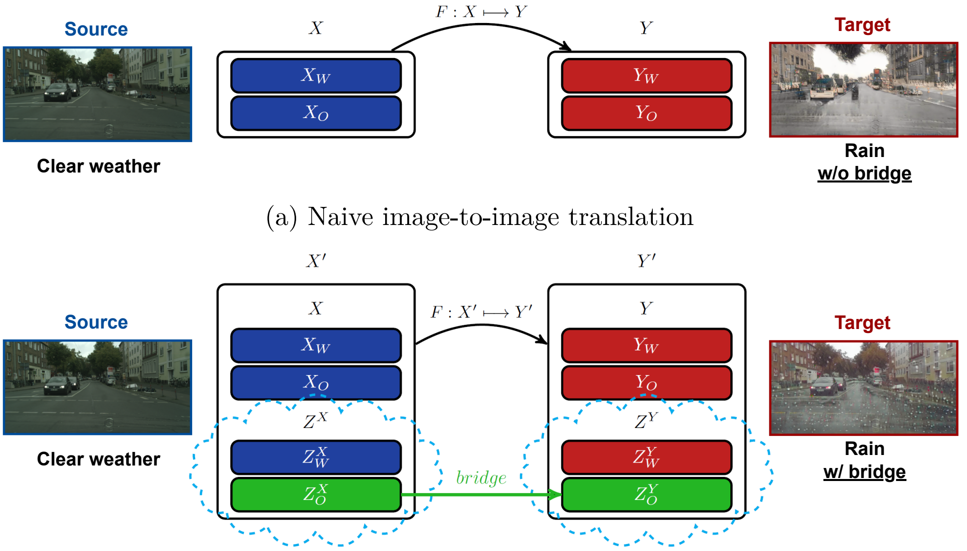
Domain bridge for unpaired image-to-image translation and unsupervised domain adaptation

The contributions of this chapter have been published in:

Pizzati, F., de Charette, R., Zaccaria, M., and Cerri, P. (2020b). Domain bridge for unpaired image-to-image translation and unsupervised domain adaptation. In *WACV*

arXiv: https://arxiv.org/abs/1910.10563

Contents



(a) Naive image-to-image translation

(b) Domain bridged image-to-image translation

Figure 3.1: Naive image-to-image translation (a) learns the $X \to Y$ domain mapping, that in presence of large domain shifts can lead to unrealistic results. Conversely, our *domain bridge* (b) completes source and target domains with automatically retrieved web-crawled data (Z^X, Z^Y) which share common characteristics, identified thanks to human knowledge priors. This eases the image translation mapping performed by MUNIT (Huang et al., 2018b), providing realistic results.

3.1 Problem statement

Image-to-image translation training may fail if the domain shift between source and target datasets is wide due to high visual differences, that limit the adversarial learning effectiveness.

In this chapter, we present a simple domain bridging technique which, opposite to the standard image-to-image (i2i) translation (Fig. 3.1a), benefits of additional sub-domains identified by human knowledge, and retrieved automatically from web-crawled data (Fig. 3.1b). Our method produces qualitatively significantly better results, when the source and target domains are far since the bridge eases the learning of the mapping. We apply our i2i methodology to the case of clear → rainy images showing that domain bridging leads the translation to preserve drops on the synthetic images. Furthermore, we extend our work to unsupervised domain adaptation (UDA) to demonstrate that generated images are consistent with the original target domain. We make novel contributions for UDA too (Fig. 3.2) and demonstrate that all together we perform on par with the most recent UDA methods at the time of acceptance while being much simpler.

We contribute in three different ways:

- **i2i:** we propose a novel *domain bridge* (Sec. 3.3.1) to augment automatically the source and target domains and ease i2i mapping,
- **i2i with UDA:** *online multimodal style-sampling* (Sec. 3.3.2.1) is applied for UDA, thus increasing the translation diversity,
- **UDA:** we propose novel *weighted pseudo label* (Sec. 3.3.2.2) to benefit from self-supervision without the need of offline processing as for the original pseudo label (Lee, 2013).

3.2 Related work

Most of the related literature on image translation has been discussed in Sec. 1.2. We now narrow down to two topics close to the proposed method, since it exploits data crawled from the web, and we contributed also in the field of unsupervised domain adaptation for segmentation.

Web-based learning. The need of neural network to learn on big data collections has pushed numerous approaches to use web-crawled images to help many vision tasks. Preliminary works (Chen et al., 2013; Divvala et al., 2014) learn to extract object relationships by automatically crawling the web. Another line of research benefit from webly-collected images to train supervised learning tasks, by using the query-image association as a form of weak supervision. For instance, it is possible to learn classification (Chen and Gupta, 2015) or more complex tasks as segmentation (Jin et al., 2017; Shen et al., 2018). Some crawl also video to benefit from consistency between frames (Hong et al., 2017; Lee et al., 2019) or to learn 3D features (Qian et al., 2022). More recent approaches extend web supervision for avoiding catastrophic forgetting (Maracani et al., 2021) or to increase the generalization of segmentation networks (Kim et al., 2021). To the best of our knowledge, we were the first to propose a web-based regularization method for image translation.

Unsupervised domain adaptation for semantic segmentation. Usually, training segmentation networks on a source dataset causes performance degradation in presence of a domain shift while testing on target conditions. The field of domain adaptation for semantic segmentation tries to mitigate this problem accessing the unlabeled images of the target domain. The first attempts were based on unsupervised source/target feature alignment with adversarial learning (Hoffman et al., 2016; Biasetton et al., 2019; Michieli et al., 2019; Sankaranarayanan et al., 2018; Zhang et al., 2017, 2018b; Tsai et al., 2018; Luo et al., 2019). In parallel to this, image-to-image translation provides a complementary pixel-level alignment that further boosts performances (Hoffman et al., 2017; Li et al., 2019). Another line of re-

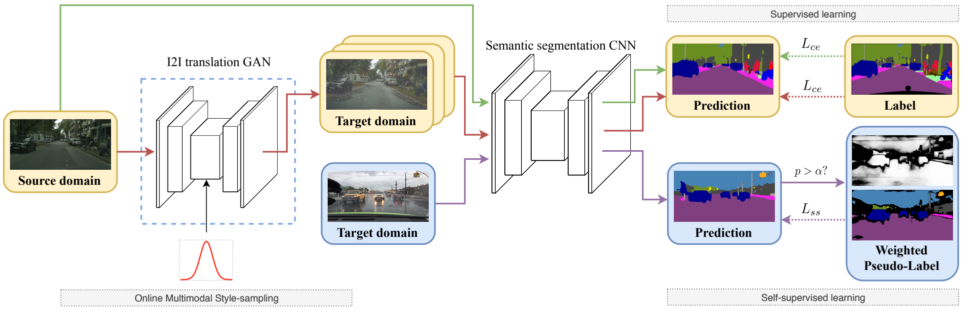


Figure 3.2: Overview of our pipeline for unsupervised domain adaptation. The blue dashed square means that the GAN parameters are frozen. The Image-to-image translation network is trained offline with our domain bridge strategy. Different line colors refer to different probabilities for one path to be executed. Loss functions are denoted with dotted lines.

search (Zou et al., 2018; Zhang et al., 2018c; Hoyer et al., 2022) uses instead pseudo-labels (Lee, 2013) as a form of self-supervision to perform domain adaptation. There are specific methods which exploit characteristics of the target dataset such as night (Sakaridis et al., 2020; Wu et al., 2021) or fog (Ma et al., 2022; Sakaridis et al., 2018; Dai et al., 2020; Lee et al., 2022).

3.3 Method

Our methodology aims to translate clear images to rainy images reaching both high-qualitative images for both qualitative evaluation and usability to train semantic segmentation networks in rainy weather. Thus, our innovations are spread between image-to-image translation (Sec. 3.3.1) and unsupervised domain adaptation for semantic segmentation (Sec. 3.3.2).

3.3.1 Image-to-image translation (i2i)

Image-to-image translation GAN networks learn to approximate the mapping $F : X \mapsto Y$ using adversarial training, from two sets of representative images in each domain, denoted here A and B. Each image in both sets can be interpreted as a sampling from a probability distribution P associated with the image domain (Liu et al., 2017). Formally, $\forall a \in A, a \sim P_X(x); \forall b \in B, b \sim P_Y(y)$. GANs are well-known for their instability at training. For the latter to succeed, the network needs to be fed with representative sets of images, so that it can extract the common domain characteristics. Even though, large domain gaps may be difficult to model for the network, resulting in a loss of characteristic features of the target domain. This could be intuitively be solved by increasing the dataset

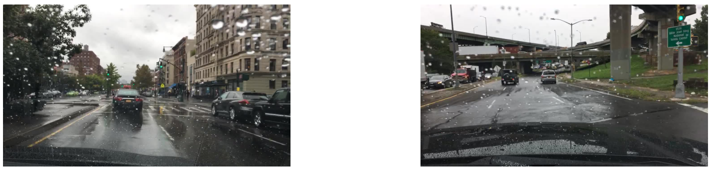


Figure 3.3: Rainy images from the Berkeley deepdrive dataset (Xu et al., 2017). Drops on the windshield or reflections help us perceive that it is raining.

size, since more data would contribute in having a better mapping. Unfortunately, this is not always possible, and it comes with significant costs. Instead, we propose to leverage human knowledge to collect necessary images from the Internet, thus eliminating the need for further acquisitions.

For getting a good image translation, it is necessary to concentrate also on some minor image details, which may still have a significant perceptual impact. This is the case for rain images, where even a few drops on the lens play an important role in *sensing* the rain, as it can be seen in Fig. 3.3. Furthermore, studies on human preference advocate that the presence of raindrops is the most important factor for recognizing a scene as rainy (Tremblay et al., 2020). Nevertheless, i2i networks may ignore some fundamental elements as drops, lens artifacts or reflections, just to concentrate on the general appearance of the scene. This ultimately impacts the realism of generated images. We argue that some characteristics changes (drops, artifacts, etc.) are ignored because they are relatively minor compared to other characteristics changes (*e.g.* wetness, puddles, etc.), and demonstrate the training may benefit from bridging to ease domain mapping.

Domain bridging. Studying the specific case of adverse weather conditions, it is possible to formalize a generic domain K as the union of finer-grained domains, such as $K = \{K_W, K_O\}$. In it, K_W represents the sub-domains typical of weather, *e.g.* the presence of precipitations, road wetness, and many more. K_O, instead, is composed of sub-domains unrelated to the weather. Examples are the scenario, the city, and the illumination. Thus, it is possible to represent X and Y as

$$\begin{aligned} X &= \{X_W, X_O\}\ , \\ Y &= \{Y_W, Y_O\}\ . \end{aligned} \tag{3.1}$$

Generally, only the joint probability distribution $P_K(k) = P_{K_W,K_O}(k), k \in K$ is estimable, as we have no knowledge about the marginal probability distrubtions $P_{K_W}(k)$, $P_{K_O}(k)$.

On one hand, we hypothesize that it is possible to obtain a more stable image-to-image translation if the differences between the two datasets

are minimized. On the other, we have to obtain a GAN that produces an effective transformation, so it is necessary to model correctly all relevant features of adverse weather. To simultaneously reach both objectives, images collected from web-crawled videos are added to the A and B datasets, obtaining two new training sets A' and B', with respective domains X' and Y', which aims as bridging the gap between the initial domain X and Y. This is illustrated in Fig. 3.1b.
Our intuition is that adding samples with specific similarities identified by human knowledge will reduce the distance between the probability distributions $P_{X'_O}(x)$ and $P_{Y'_O}(y)$, with respect to $P_{X_O}(x)$ and $P_{Y_O}(y)$. As a consequence, the translation model will be more focused on weather-related characteristics and more stable during training.

Once the main hypotheses are formalized, the approach on how to select new images is needed. Let Z^x and Z^y be two images sets. As before, we have

$$\begin{aligned} Z^x &= \{Z^x_W, Z^x_O\} \ , \\ Z^y &= \{Z^y_W, Z^y_O\}. \end{aligned} \tag{3.2}$$

We choose Z^x and Z^y in order to have

$$\begin{aligned} max\ |Z^x_W \cap X_W| \ , \\ max\ |Z^y_W \cap Y_W| \ , \\ max\ |Z^x_O \cap Z^y_O| \ , \end{aligned} \tag{3.3}$$

where $|\cdot|$ is the set cardinality. Hence, it is possible to identify two image sets C and D such as $\forall c \in C, c \sim P_{Z^x}(z); \forall d \in D, d \sim P_{Z^y}(z)$.

It is now possible to train the image-to-image translation network on A' and B' defined as:

$$\begin{aligned} A' &= A \cup C \ , \\ B' &= B \cup D \ . \end{aligned} \tag{3.4}$$

Adding new images, the differences in the global appearance of the two domains is minimized, while the weather-related domain shift remains constant.

In other words, our approach consists in selecting new image samples, with weather conditions corresponding to those in the original dataset, and join them to the existing data. The newly-added images are required to share some domains unrelated to the weather. Retrieving images from the same location and with the same setup ensures that.

In practice, we retrieved these additional samples from web-crawled videos using domain-related keywords search (details in Sec. 3.4.1.1). The same bridging can be applied automatically to other domain shifts, though as the domain differences become less semantically evident, human expertise

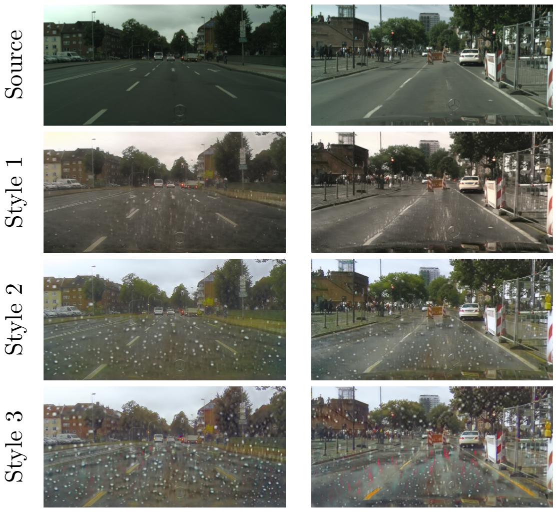


Figure 3.4: Examples of multimodal style-sampling from our Domain-bridged i2i (Sec. 3.3.1). Note the consistency of style regardless of the source image.

may be required to properly select the C and D datasets.

We use MUNIT (Huang et al., 2018b) as backbone for our image-to-image translation network, as it allows disentanglement of style and content, which will be of high interest for us.

3.3.2 Unsupervised Domain Adaptation (UDA)

Similar to previous works (Hoffman et al., 2017; Li et al., 2019), we use our i2i network with domain bridge to translate images from pre-labeled *clear weather* datasets and learn semantic segmentation in rain in an unsupervised fashion. We follow the standard UDA practice which is to alternately train in a supervised manner from source (*clear*) images with labels and train in a self-supervised manner from target (*rain*) images without labels. Our entire UDA methodology (depicted in Fig. 3.2) brings two novels contributions: a) we use multi-modal clear→rain translations as additional supervised learning, b) we introduce weighted pseudo label - a differentiable extension of pseudo label (Lee, 2013) - to align source and target without any offline process.

3.3.2.1 Online Multimodal Style-sampling (OMS)

The standard strategy for UDA with i2i networks is to learn from the offline translation of the whole dataset (Hoffman et al., 2017; Li et al., 2019).

We instead propose to use the multimodal capacity of MUNIT i2i to generate multiple target styles (*i.e. rain appearances*) for each input image. Styles are sampled during training time. In this way, even if the source image content remains unaltered, it will be possible for the segmentation network to learn different representations of the same scene in the target domain, ultimately leading towards wider diversity and thus more robust detection. Different styles for the same image modify, among other factors, the position and size of drops on the windshield, and the intensity of reflections. This is visible in Fig. 3.4 showing three arbitrarily sampled styles, where in the last row images that resemble heavy rain are consistently produced.

3.3.2.2 Weighted Pseudo-Labels (WPL)

Pseudo Label (Lee, 2013) was proposed as a self-supervised loss to further align source and target distributions. The principle is to self-train a network on target (here, *rain*) whenever the prediction confidence is above some threshold, thus reinforcing the network beliefs.
Most often for UDA, thresholds are either hard coded or calculated offline as the median per-class confidence dataset-wise (Li et al., 2019; Zou et al., 2018). The latter requires storing all predictions for the whole dataset, which is cumbersome. To overcome this, thresholds may be estimated online image-wise or batch-wise (Iscen et al., 2019). It has to be noted that thresholds are critical since pseudo-labeling can harm global performances if thresholds were underestimated[1] or have limited impact if overestimated (Lee, 2013).

We instead propose Weighted Pseudo-Labels (WPL) which estimates a global threshold α within the network optimization process. The general principle of WPL is to weight the self-supervised cross-entropy using learned threshold α, thus acting as continuous pseudo-labeling. Not only WPL does not require offline processing, but it is aware of the global network confidence thus leading to better results. In detail, let x be an input image and $\hat{x}_u$ the pseudo label of pixel u, such that

$$\hat{x}_u = \begin{cases} \text{argmax}_q \, f_u(x) & \text{if } \max(f_u(x)) \geq \alpha \\ \text{None} & \text{otherwise} \end{cases} , \tag{3.5}$$

where $f_u(x)$ refers to the class probabilities of u predicted by the network f, and argmax_q is the best class prediction. In its original implementation (Lee, 2013) $\hat{x}_u$ is directly used to weight the cross-entropy self-supervision. Instead, we weight this with a weight matrix W of the same size than x:

$$w_u = \begin{cases} \frac{\max(f_u(x)) - \alpha}{1-\alpha} & \text{if } \max(f_u(x)) \geq \alpha \\ 0 & \text{otherwise} \end{cases} , \tag{3.6}$$

[1] In such case, the ratio of wrong pixels over pseudo-labeled pixels will be too high and lead to incorrect self-supervision.

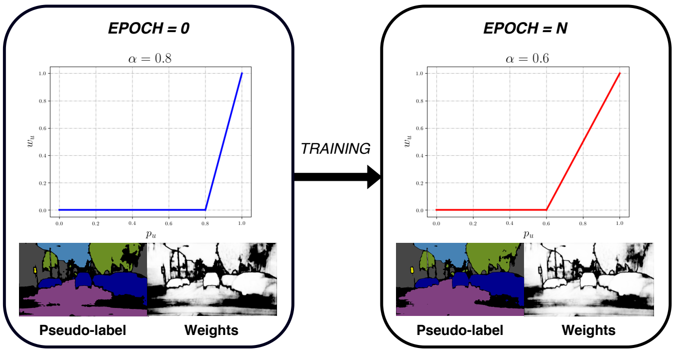


Figure 3.5: Analysis of the effect of α optimization. During training the Weighted Pseudo Label expands from high confidence pixels only (left) to lower ones (right).

The complete loss for WPL is thus defined as the weighted sum of cross-entropy loss L_w and a balancing loss L_b:

$$L_{ss} = \sigma L_w + \gamma L_b \ , \tag{3.7}$$

where σ and γ are loss weights and cross-entropy loss is:

$$L_w = -\sum_{q \in Q} w_u \hat{x}_{u,q} log(p_u) \ . \tag{3.8}$$

$\hat{x_u}$ is the one-hot encoding of pseudo-label as in Eq. 3.5 and Q is the set of classes. In this way, predictions where the network is uncertain are weighted less in the network pseudo-label based training. To avoid that the self-supervised contribution remains set to zero by the optimizer, L_b is required as a balancing loss:

$$L_b = log^2(1 - \alpha) \ . \tag{3.9}$$

The optimization of α leads to a pseudo-label expansion within the training process. Fig. 3.5 is an illustration of the growing process during training. For the first iteration (Fig. 3.5, left), the L_w term prevails over L_b, pushing α towards 1 thus including in the pseudo-label only pixels with high confidence. With the minimization of L_w (Fig. 3.5, right), L_b becomes gradually more important, leading the network to simultaneously include lower confidence pixels inside the pseudo-label, and increasing the informative potential of higher-confidence labels. Note that for numerical stability, we assume $\alpha = \text{sigmoid}(\beta)$ and estimate β.

3.3.2.3 Losses

To balance the self-supervised WPL contribution with the supervised learning in segmentation, we employ a probability-based approach where pseudo-label is applied only if a uniformly sampled variable $p_{pl} \in U(0, 1)$ is above a predefined threshold p_{tp}. Hence, the complete UDA loss function is:

$$L(x_a, \hat{x}_a, x_b) = L_{ce}(f(x_a), \hat{x}_a) + L_{ss}(f(x_b))[p_{pl} > p_{tp}] \tag{3.10}$$

if we train on source and target data, and

$$L(x_a, \hat{x}_a, x_b) = L_{ce}(f(g(x_a)), \hat{x}_a) + L_{ss}(f(x_b))[p_{pl} > p_{tp}] \tag{3.11}$$

if we train on translated images and target. In Eq. 3.10 and 3.11, $x_a \in A$ is source image with label $\hat{x}_a$, $x_b \in B$ target image, L_{ce} the cross-entropy loss, f our segmentation network, g our bridged-GAN, and $[\cdot]$ are the Iverson brackets.

3.4 Experiments

We now evaluate the performance of both our i2i proposal (Sec. 3.4.2) and our UDA proposal (Sec. 3.4.3) on the clear→rain problem using clear/rain datasets recorded with different setups.

3.4.1 Experimental settings

3.4.1.1 Datasets

For i2i and UDA, we use the german dataset Cityscapes (Cordts et al., 2016) as source (*clear*), and a subset of the American Berkeley DeepDrive (Xu et al., 2017) (BDD) as target (*rain*). The bridge dataset, only used for the i2i, is a collection of Youtube videos. We now detail each dataset.

Cityscapes. We train on Cityscapes training set with 2975 pixel-wise annotated images, and evaluate on their validation set with 500 images. While we train on crops, we evaluate on full-size images, *i.e.* 2048 × 1024. The *trainExtra* set, with 19997 images, is also included in the domain bridge to further reduce the domain shift.

BDD-rainy. We use the coarse weather annotations of BDD together with daylight annotation to obtain a subset we call BDD-rainy (*i.e.* rainy and daylight), *i.e.* 1280 × 720. For training the rainy+daylight is extracted from the 100k split, while for validation only the 10k split is used. Obviously, duplicates present in both splits are removed. It has to be noted that, while daylight annotation is accurate, weather annotation is approximate and "rainy" images may either be taken during or *after* a rain event, thus with or without drops on the windshield. This further increases complexity.

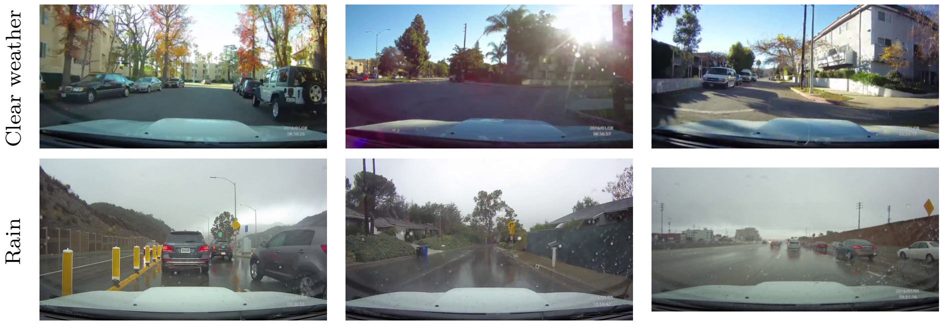


Figure 3.6: Samples from the bridge dataset in different weather conditions. Note that the acquisition setup (camera position, optics, etc.) remains unaltered.

Bridge dataset. 5 sequences (1280 × 720) were extracted from a single Youtube channel with keywords ”driving” (2 videos) for clear weather and ”driving rain”/”driving heavy rain” (2/1 video) for rainy scenarios. The choice of using videos from a unique channel further reduces domain gaps, ensuring the same acquisition setup. Some samples are shown in Fig. 3.6. Also to maximize image diversity, videos are uniformly sub-sampled into 2x6026 clear weather images and 3x9294 rainy images, leading to a total of 39934 frames.

3.4.1.2 GAN training

During training, the images are downsampled to be 720 pixels in height, and cropped to 480 × 480 resolution. The network is trained for 200k iterations, with batch size 1. Adam is used as optimizer, with learning rate 1e-4, $\beta_1 = 0.5$, $\beta_2 = 0.999$.

3.4.1.3 Segmentation training

We use Light-weight Refinenet (Nekrasov et al., 2018) with Resnet-101 as backbone, pretrained on the full-size Cityscapes dataset. The refining is achieved by training for 100 epochs on 512 × 512 crops, after downscaling images to 1024 × 512 for GPU memory constraints. We use data augmentation for the training process, with random rescaling between a factor 0.5 and 2, and random horizontal flipping. The batch size is 6. We use SGD with different learning rates for the encoder (1e-4) and the decoder (1e-3). The momentum is set to 0.9, and the learning rate is divided by 2 every 33 epochs. When pseudo-labels are added to the training, we further refine the network for 70 additional epochs, with constant learning rate divided by 10 with respect to the initial values. The α parameter is initialized to 0.8 and estimated by SGD as well, with learning rate 0.01 and momentum 0.9.

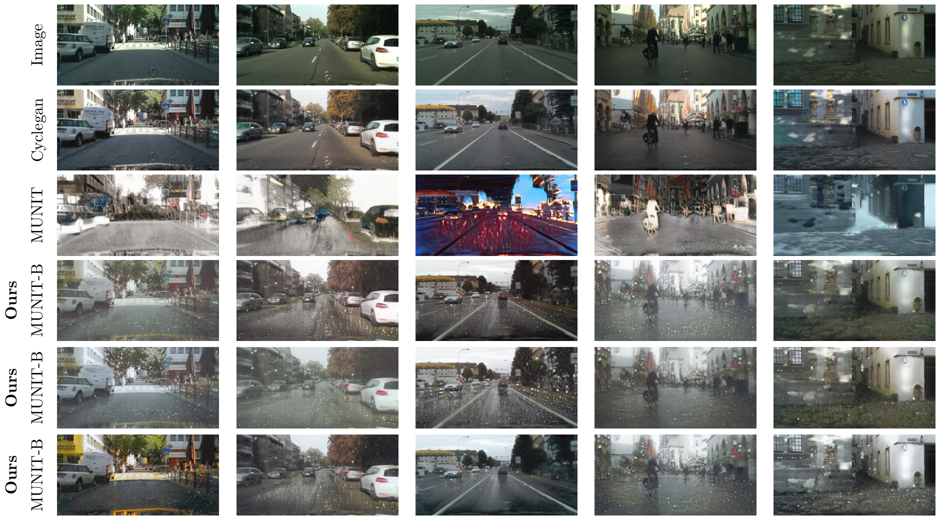


Figure 3.7: Qualitative comparison between state-of-the-art architectures and MUNIT-Bridged (MUNIT-B) for i2i in the clear → rain transformation.

3.4.2 Bridged image-to-image translation

We evaluate our bridged i2i (Sec. 3.3.1) on the Cityscapes to BDD-rainy translation task, and compare results against the most recent baselines at the time of submission, *i.e.* CycleGAN (Zhu et al., 2017a) and MUNIT (Zhu et al., 2017b). As stated, our i2i uses a MUNIT backbone and is referred to as MUNIT-bridged. It is trained on the bridged versions of the two datasets. Training follows details from Sec. 3.4.1.2, except for CycleGAN that follows the original implementation[2].
We argue - like others - that GAN metrics aren't appropriate for such evaluation. Thus, we report qualitative evaluation and segmentation task evaluation, together with usual GAN metrics.

3.4.2.1 Qualitative evaluation

Fig. 3.7 shows randomly selected samples from the Cityscapes validation set[3]. It is visible that both CycleGAN and original MUNIT method fails at modeling the rain appearance, probably due to the large domain gap. In detail, CycleGAN brings no realistic changes to the scene appearance, only adjusting color-levels in the image. Original MUNIT, instead, seems to have collapsed and fails to produce significant outputs, probably due to instability related to the domain gap. Conversely, our MUNIT-bridged

[2] (Zhu et al., 2017a) claims that best performances are obtained keeping constant the learning rate for half the training process (100k iterations in this case) to 2e-4 and then linearly decreasing to 0.

[3] For MUNIT and our method, MUNIT-bridged, we also randomize the style.

Network	LPIPS↑	IS↑
Real images	0.7137	-
CycleGAN (Zhu et al., 2017a)	0.1146	1.15
MUNIT (Huang et al., 2018b)	**0.3534**	**1.92**
MUNIT-Bridged	0.2055	1.69

(a) GAN metrics

Network	mIoU↑
Baseline	31.67
CycleGAN (Zhu et al., 2017a)	35.09
MUNIT (Huang et al., 2018b)	20.78
MUNIT-Bridged	**35.18**

(b) Semantic Segmentation

Table 3.1: Quantitative evaluation of translated image realism, diversity, and semantic segmentation effectiveness.

model (MUNIT-B in Fig. 3.7) is the only one able to add realistic traits of rain in the synthetic images, thanks to the domain bridge.

3.4.2.2 Quantitative evaluation

We compute GAN metrics following usual practices from (Huang et al., 2018b) and report results in Tab. 3.1a. The LPIPS distance (Zhang et al., 2018a; Shen et al., 2019) measures the image diversity (Huang et al., 2018b), while the Inception Score (IS) evaluates both quality and diversity (Salimans et al., 2016). In detail, LPIPS is the average on 19 paired translation of 100 images, and we report the diversity of real data in the target dataset as *upper bound*. Inception Score uses the InceptionV3 network previously trained to classify source and target images.
Overall, we successfully improve performances over CycleGAN in both metrics, but original MUNIT has significantly higher performance. However, the images generated by MUNIT are evidently unrealistic (cf. Fig. 3.7) and thus we argue that GAN metrics are unreliable which is in fair alignment with (Barratt and Sharma, 2018) advocating that Inception Score is uncorrelated with image quality when the output is evidently unrealistic.

For a more comprehensive evaluation, we train a segmentation task on GAN translated clear→rain images, and evaluate the standard mIoU metric on real rain images, reporting results in Tab. 3.1b. Note that for a fair comparison, we only sample a single style for MUNIT-based models, and report results when only trained on clear images as *baseline*. If the domain gap were reduced by the GAN translations, an improvement should be visible. Instead, from the table, training on the original MUNIT-translated dataset leads to a significant performance decrease disproving the high GAN metrics. Finally, our method outperforms CycleGAN by a little margin although CycleGAN fails to produce good quality images. Conversely, our method simultaneously reduces the domain shift and increases realism, thus it also eases performances evaluation on synthetic data.

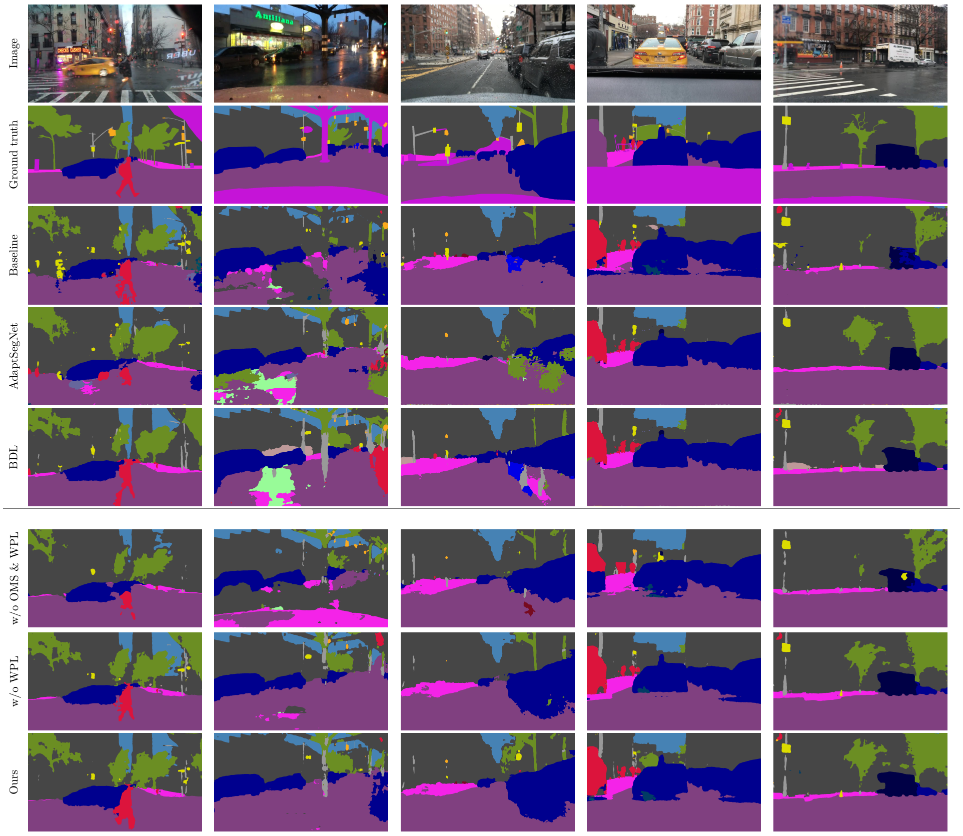


Figure 3.8: Comparison of our method with the state-of-the-art for semantic segmentation UDA.

3.4.3 Unsupervised Domain Adaptation

We now evaluate our UDA contributions encompassing our i2i translation methodology and compare with AdaptSegNet (Tsai et al., 2018) and BDL (Li et al., 2019), the best found works at the time. We do not compare with less recent methods as CyCADA (Hoffman et al., 2017) since the approaches we are evaluating have already demonstrated to guarantee superior performances in UDA (Tsai et al., 2018; Li et al., 2019). For fair comparison and given architectural similarities, BDL was adapted to work with the same segmentation network, data augmentation policy and hyperparameters detailed in Sec. 3.4.1.3.

Quantitative results are shown in Tab. 3.2 where *Ours* refer to the proposed UDA using domain bridge i2i translation with online multimodal style-sampling (OMS) and weighted pseudo label (WPL), and *w/o WPL* or *w/o OMS & WPL* the self-explanatory ablation versions. *Baseline* refers to the training without any UDA. Overall, our methodology performs on par (+0.44) with BDL, the best state-of-the-art, using a much simpler do-

Method	**mIoU**	road	sidewalk	building	wall	fence	pole	t light	t sign	veg	terrain	sky	person	rider	car	truck	bus	train	m. bike	bike
Baseline	31.67	77.40	39.95	61.20	12.01	24.76	23.68	13.21	24.11	58.33	27.18	78.86	24.73	12.78	63.34	24.01	28.43	0.00	4.76	2.90
AdaptSegNet	33.44	82.23	39.85	62.06	9.84	17.73	20.39	10.91	22.47	66.30	22.81	76.54	32.24	38.49	68.95	13.08	30.31	0.00	17.97	3.26
BDL	39.60	83.18	**48.78**	**73.93**	**30.87**	**27.33**	26.03	**15.10**	26.05	72.63	26.08	**88.01**	28.59	26.59	**76.37**	43.31	**50.11**	0.00	7.38	2.10
w/o OMS & WPL	35.18	79.00	37.23	62.36	8.60	14.78	20.98	11.94	22.92	68.02	13.11	82.55	38.96	**44.61**	72.34	29.10	39.40	0.00	19.32	3.16
w/o WPL	39.72	82.53	44.51	69.97	20.29	22.91	**28.93**	14.02	29.17	74.32	**28.98**	83.53	36.75	32.80	71.29	43.03	46.34	0.00	**21.80**	3.54
Ours	**40.04**	**84.03**	44.09	70.51	24.10	23.02	28.31	14.08	**30.07**	**75.31**	27.89	83.49	**39.10**	33.63	74.70	**48.60**	49.34	0.00	6.77	**3.80**

Table 3.2: State-of-the-art comparison. OMS refers to online multimodal style-sampling. WPL is the weighted pseudo labels strategy.

Pseudo-labels	**mIoU target ↑**
None	39.77
Batch-wise	38.23
WPL (Ours)	**40.04**

Table 3.3: Comparison of various pseudo labeling strategies: Batch-wise, with our WPL, or with None.

main adaptation method, and significantly better (+6.6) than AdaptSegNet. Studying the contributions of our OMS and WPL contributions, all components are necessary to reach the best performances. Qualitative evaluation on the target dataset is in Fig. 3.8, aligned fairly with quantitative metrics.

3.4.3.1 Weighted pseudo labels

We evaluate the effectiveness of our WPL proposal and report results in Tab. 3.3, comparing similar training with either *WPL* (Ours), *batch-wise* Pseudo-Label[4], or *None*. For all, the training is performed using as target the whole BDD100k train set (removing duplicates from 10k split) together with the rainy sequences from Domain-bridge dataset, resulting in over 90k images. Performance is reported on target BDD-rainy. We do not compare against offline pseudo label, as this would be impractical with such a big dataset, and this evaluation is partly encompassed in BDL comparison (cf. Tab. 3.2). For WPL, we empirically set $\gamma = 1$, $\sigma = 0.005$ (Eq. 3.7) to balance contributions and $p_{tp} = 0.75$ (Eqs. 3.10 & 3.11).

From results in Tab. 3.3, *WPL* performs the best and *batch-wise* pseudo label third. In fact, the performance decrease for *batch-wise* (compared to no self-supervision) may be explained since best batch pixels are used as pseudo-labels, thus possibly implying some incorrect self-supervision in case of low batch accuracy. Instead, our *WPL* boosts the mIoU on the target dataset which is expected due to its expansion behavior.

[4] For batch-wise Pseudo-Label implementation, we compute optimal threshold per class and per batch.

3.5 Conclusions

In this chapter, we introduced a novel strategy for crawling data from the web, that exploited as a prior the human ability to identify similarities and differences within domains. This allowed to generate realistic rainy images with an i2i network, while preserving traits of adverse weather that were typically ignored by state-of-the-art architectures. We also contributed in unsupervised domain adaptation, designing a simple pipeline obtaining on par performances with respect to the state-of-the-art at the time, and introducing a novel pseudo labeling strategy, that works with an unlimited number of images, and has a learnable weight parameter used to guide region growing. Importantly, we introduced a domain decomposition interpretation that is useful to model similarities between source and target. We reused similar reasonings in Chapters 4, 7, 8.

While it is true that the weighted pseudo labels help training, we believe limited performances shown in Tab. 3.3 are also related to the fact that we did not investigate multiple per-class thresholds as in other works (Zou et al., 2018). Importantly, similar ideas in this research have been used in subsequent papers, for instance exploiting web-crawled data for connecting existing datasets (Maracani et al., 2021) or to analyze different domain shifts (Ma et al., 2022).

Also, we demonstrated that the network may concentrate more on traits previously ignored, thanks to the addition of selected data on both source and target. Still, this requires a common decomposition sub-domain *across* source and target, which is not always easily identifiable. Moreover, some characteristics could still be ignored, due to limited discriminator contextual understanding. For these reasons, in Chapter 4 we provide an image translation method to model subtle characteristics decomposing *a single domain*, thus reducing the need of collecting new data.

Chapter 4

Leveraging local domains for image translation

The contributions of this chapter have been published in:

Dell'Eva, A., Pizzati, F., Bertozzi, M., and de Charette, R. (2022). Leveraging local domains for image-to-image translation. In *VISAPP (best paper award)*

arXiv: https://arxiv.org/abs/2109.04468

This work is the result of a collaboration of University of Parma, in which I co-supervised Anthony Dell'Eva (1st year PhD student at the time).

Contents

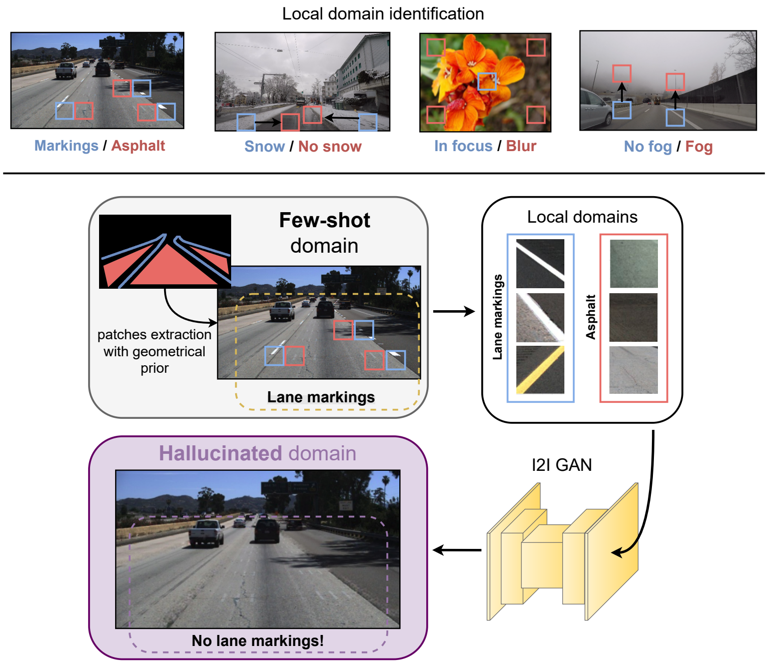


Figure 4.1: Our method is able to generate images of unseen domains, leveraging geometrical-guidance to extract patches of *local domains*, i.e. spatially defined sub-domains, on source images. Here, we generate an image without any lane markings training only on an extremely small amount of images with well-defined lane markings.

4.1 Problem statement

While i2i GANs perform well at learning global scene changes such as weather, painting style, etc. (Liu et al., 2017; Zhu et al., 2017a), they struggle to learn subtle local changes. In this chapter, we leverage human domain knowledge to guide i2i and improve downstream tasks on target, *without seeing target images*. This is of paramount importance for real-world applications like autonomous driving (Schutera et al., 2020; Bruls et al., 2019; Romera et al., 2019) which must operate safely in all hazardous conditions – some of which are rarely observed. Reasoning similarly as in Chapter 3, our method exploits human knowledge to identify domain-specific local characteristics which we call *local domains* (Fig. 4.1, top). A substantial difference is that we decompose a single domain, training on extracted sub-domains, instead of using a decomposition of both source and target to crawl new data. The decomposition is used as guidance to perform patches translations on *source only*, thus hallucinating a new unseen domain. An example in Fig. 4.1 bottom shows we leverage local domains knowledge about ‘lane markings’ and ‘asphalt’ to hallucinate a new domain without lane markings. The latter domain can then be used to increase robustness on unstructured road environments which are typically hard to capture but may cause dramatic failures.

Experimental evidence in this chapter shows indeed that our new domain acts as a bridge leading to a performance boost on target. Notably, our method exhibits few-shot capabilities, requiring only source images and minimal human knowledge about the target.

In short, the main contributions of this chapter are:

- we introduce and define *local domains* as being domain-specific spatial characteristics (Sec. 4.3.1),
- to the best of our knowledge, we propose the first geometrical-guided patch-based i2i, leveraging our *local domains* priors (Sec. 4.3.2) and enabling continuous geometrical translation (Sec. 4.3.3),
- we experiment on different tasks in a few-shot setting, showing that our translations lead to better performance on all target downstream tasks (Sec. 4.4).

4.2 Related works

Since to extract local domains we include some kind of prior as semantic supervision or geometry, we present relevant works which exploit different priors to boost image translation performances. The literature about general i2i presented in Sec. 1.2 is also related to the task.

Prior-guided image translation. Several priors can be exploited to increase image translation effectiveness, with several degrees of supervision as bounding boxes (Shen et al., 2019; Bhattacharjee et al., 2020), semantic maps (Li et al., 2018a; Tang et al., 2019b; Cherian and Sullivan, 2019; Zhu et al., 2020b,a; Lin et al., 2020a; Ma et al., 2019; Park et al., 2019) or instance labels (Mo et al., 2019; Xu et al., 2021). Importantly, scene geometry could be used as a prior, with learned correspondences (Wu et al., 2019b) or by exploiting additional modalities (Arar et al., 2020). Some use text for image editing purposes (Liu et al., 2020). Others exploit full semantic maps for road randomization (Bruls et al., 2019), to generalize across challenging lane detection scenarios. However, they are limited to annotated road layouts and constrained by expensive complete segmentation maps.

4.3 Method

We address the problem of image to image translation accounting for *source* and *target* domains having predominant local transformations. As such, leveraging *only source* data, our proposal hallucinates a new *unseen* intermediate domain which can be used to ease transfer learning towards *target*.

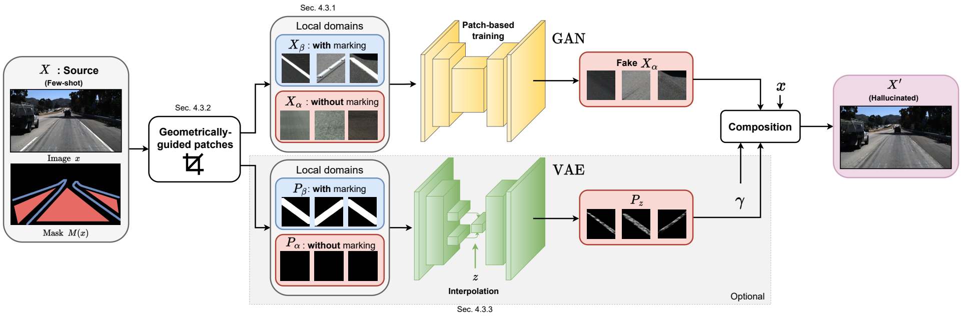


Figure 4.2: Architecture pipeline. Our method exploits knowledge about *local domains* (Sec. 4.3.1) and relies on geometrical-prior to extract samples of local domains in *source only* (Sec. 4.3.2) that train a patch-based GAN. Here, source is having "lane markings" and "asphalt" local domains (X_α and X_β, respectively) while target have only "asphalt" (X_β), learning $X_\alpha \mapsto X_\beta$ further reduces the gap with target. An optional local domain interpolation strategy (Sec. 4.3.3) is added for generating geometrically continuous translation between local domains (here, simulating lane degradation).

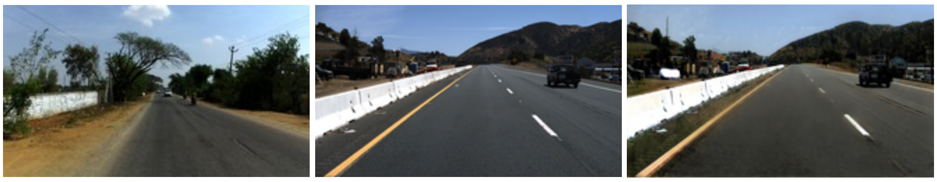
An overview of our pipeline is in Fig. 4.2.

In the following, we introduce our definition of local domains (Sec. 4.3.1) and propose a geometrical-guided patch-based strategy to learn translation between the latter (Sec. 4.3.2). For some local domains, we also show that a continuous geometrical translation can be learned from the interpolation of a mask (Sec. 4.3.3). Finally, we describe our training strategy showing few shot capabilities (Sec. 4.3.4).

4.3.1 Local domains

Image-to-image (i2i) networks learn a mapping function $G : X \mapsto Y$ from a source domain X to a target domain Y, such that the distribution $P_{G(X)}$ approximates P_Y. The goal is to transfer the features of domain Y to samples from X while preserving their content. This works well for transformation globally affecting the scene (*e.g.* summer to winter) but struggles to capture the mappings of local changes due to the under-constrained settings of the system. A simple failure example, shown in Fig. 4.3, is the translation from outdoor images having lane markings, to images having no (or degraded) lane markings. As it seeks global changes, the i2i is likely to transfer unintended characteristics while missing the subtle – but consistent – local changes (here, the lane markings).

To overcome this, we introduce *local domains* which are sub-domains *spatially defined* – for example, lane markings, asphalt, etc. Formally, we define domain X as the composition of local domains, denoted $\{X_\alpha, ..., X_\omega\}$, and the remaining sub-domains written X_O, similarly to Sec. 3. Considering



IDD (Varma et al., 2019)

TuSimple (TuSimple, 2021)

TuSimple↦IDD

Figure 4.3: Translation with CycleGAN (Zhu et al., 2017a). Sample output shows that i2i is prone to transfer global features (here, sky color) but neglects evident local features for humans as the street structure (note that IDD has no lane markings).

only two local domains of interest, it writes:

$$X = \{X_{\mathrm{O}}, X_{\alpha}, X_{\beta}\}\,. \tag{4.1}$$

Because we consider only source and target domains sharing at least one local domain, say $_{\alpha}$, we write Y as:

$$Y = \{Y_{\mathrm{O}}, Y_{\alpha}\}\,, \tag{4.2}$$

so that the distance between X_{α} and Y_{α} is close to 0. Instead of learning the direct mapping of $X \mapsto Y$, we propose to learn local domain mappings, such as $X_{\beta} \mapsto X_{\alpha}$. If such mapping is applied systematically on all samples from X, we get a new domain X' without $_{\beta}$, so:

$$X' = \{X_{\mathrm{O}}, X_{\alpha}\}\,, \tag{4.3}$$

where domain X' is unseen and thus hallucinated. Considering that X' and Y share the same local domains, they are subsequently closer:

$$\mathrm{distance}(Y, X') < \mathrm{distance}(Y, X). \tag{4.4}$$

Our intuition is that when training target data is hard to get, our hallucinated domain X' can ease transfer learning. Notably here, our method only requires a priori knowledge of the shared local domains in source and target.

4.3.2 Geometrically-guided patches

Learning the mapping between local domains requires extracting local domain samples. To do so we leverage patches corresponding to either local domains in the source dataset only. We rely here on a simple geometrical guidance from a mask $M(.)$ to extract random patches centered around a given local domain.

Considering x an image in source domain X, we extract $\mathcal{X}_\alpha$ the unordered set of patches of fixed dimension, so that:

$$\mathcal{X}_\alpha = \{\{x_{p_0}, x_{p_1}, \ldots, x_{p_m} \,|\, p \in M_\alpha(x)\} \,|\, \forall x \in X\}\,, \tag{4.5}$$

having m the number of patches per image, and $M_\alpha(x) = [\![M(x) = \alpha]\!]$ with $[\![.]\!]$ the Iverson brackets. Literally, $M(x)$ is our geometrical prior – a 2D mask of the same size as x – encoding the position of local domains. Subsequently, $M_\alpha(x)$ is filled with ones where local domain X_α is and zeros elsewhere. Similarly to Eq. 4.5, we extract the set X_β from $M_\beta(x)$ and X.

In practice, the geometrical prior $M(x)$ is often simply derivable from the image labels. For example, the position of lane marking and asphalt can both be extracted from image labels. In some cases, the position of local domains is constant dataset-wise and we use a fixed geometrical prior, so $M(x) = M$. This is for example the case for portraits datasets, where faces are likely to be centered and background located along the image edges.

Having collected the two sets of patches $\mathcal{X}_\alpha$ and $\mathcal{X}_\beta$, a straightforward patch-based GAN can learn $X_\alpha \mapsto X_\beta$. In some cases, $\mathcal{X}_\alpha$ and $\mathcal{X}_\beta$ being of similar nature we demonstrate spatial interpolation is beneficial.

4.3.3 Local domains interpolation

Continuous i2i are extensively studied (Gong et al., 2019; Wang et al., 2019b; Lample et al., 2017), but existing methods are not suitable for translation affecting only local regions as in our problem setting (see Sec. 4.4). Instead, we learn a non-linear geometrical interpolation of patch masks, leveraging a variational autoencoder (VAE).

Previously we described each patch as encompassing a single local domain but, in reality, patches often mix multiple local domains. This is the case of lane markings patches, shown in Fig. 4.2, that contain asphalt too. Hence, along with the set of local domains patches we extract the sets $\mathcal{P}_\alpha$ and $\mathcal{P}_\beta$ directly from our geometrical guidance $M(.)$, and seek to continuously interpolate $P_\alpha \mapsto P_\beta$.

In practice, our VAE having encoder $E(.)$ and decoder $D(.)$ is trained in the standard fashion, but at inference it yields the latent representation h_Z corresponding to the linear combination of $E(p_\alpha)$ and $E(p_\beta)$, having $p_\alpha \in \mathcal{P}_\alpha$ and $p_\beta \in \mathcal{P}_\beta$, respectively[1]. Formally:

$$\begin{aligned} h_Z = E(p_\alpha)\, z + E(p_\beta)\, (1 - z), \\ p_z = D(h_Z), \end{aligned} \tag{4.6}$$

where $z \in [0, 1]$ encodes the progress along $P_\alpha \mapsto P_\beta$. The final interpolated patch x_z is the composite between x_α and x_β patches, following the VAE

[1] Our formalism includes VAE reparametrization in $E(.)$

output. It writes:

$$\begin{aligned} x_z &= x_\alpha\, m + x_\beta\,(1-m)\,, \\ &\text{with } m = \gamma\, p_z\,, \end{aligned} \tag{4.7}$$

$\gamma \in [0,1]$ being an arbitrary controlled blending parameter adding a degree of freedom to our model. Furthermore, notice that the stochastic VAE behavior further increases variability, beneficial for proxy tasks.

4.3.4 Training

We train our pipeline, the patch-based GAN and the optional VAE, leveraging only images from the source domain and geometrical priors about local domains. The patch-based GAN is trained on $X_\alpha \mapsto X_\beta$ (Sec. 4.3.2) minimizing the LSGAN (Mao et al., 2017b) adversarial loss:

$$\begin{aligned} y_f &= G(x), \\ \mathcal{L}_G(y_f) &= \mathbb{E}_{x\sim P_X(x)}\left[(D(y_f)-1)^2\right], \\ \mathcal{L}_D(y_f,y) &= \mathbb{E}_{x\sim P_X(x)}\left[(D(y_f))^2\right] + \\ &\quad + \mathbb{E}_{y\sim P_Y(y)}\left[(D(y)-1)^2\right], \end{aligned} \tag{4.8}$$

along with task-specific losses. If used, the VAE interpolation (Sec. 4.3.3) is trained with standard ELBO strategy (Blei et al., 2017), minimizing reconstruction loss along with a regularizer:

$$\begin{aligned} \mathcal{L}_{VAE} &= -\mathbb{E}_{q_\phi(z|x)} \log p_\theta(x|z) + \\ &\quad + D_{KL}(q_\phi(z|x)||p(z)). \end{aligned} \tag{4.9}$$

At inference time, the full image is fed to the GAN backbone to produce the translated image, while the corresponding full interpolation mask is obtained processing mask patches independently and then stitching them together with a simple algorithm. Of note, our method has important few-shot capabilities. As we train only on source patches, a reduced number of image samples is sufficient to get reasonable data diversity, which we further demonstrated in the following section.

4.4 Experiments

We evaluate our method on 2 different tasks, namely lane markings degradation and snow addition, leveraging 4 recent datasets (TuSimple, 2021; Varma et al., 2019; Sakaridis et al., 2021; Cordts et al., 2016), and evaluating our translation both against i2i baselines and on downstream tasks. In the original paper, an additional setup on deblurring is proposed and evaluated. We

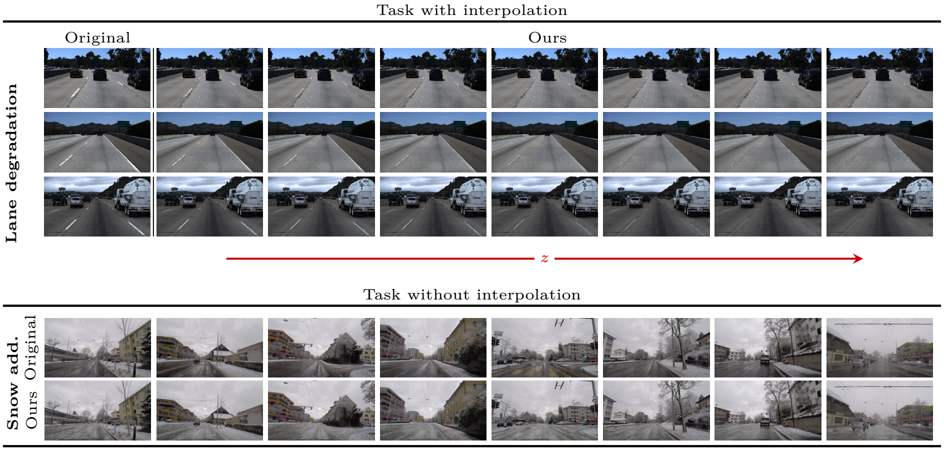


Figure 4.4: Qualitative results. For each task we show the original input image and our output with the $X_\beta \mapsto X_\alpha$ local domains translation. **Lane degradation**: sample translations on TuSimple (TuSimple, 2021) test set with increasing degradation $z \in [0.35, 0.95]$ from left to right, blending variable $\gamma = 0.75$. **Snow addition**: augmentation of ACDC (Sakaridis et al., 2021) validation set, only road is involved in the transformation.

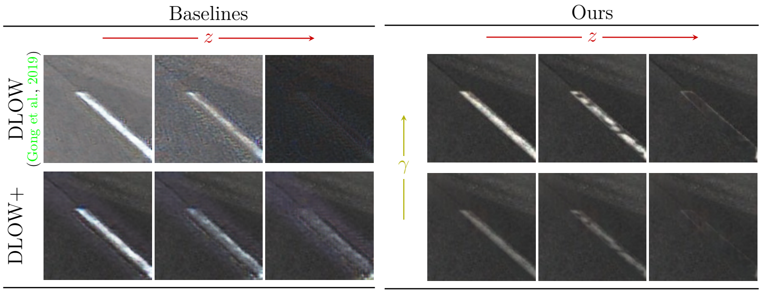
removed that from this thesis for the sake of brevity. In Sec. 4.4.1 we provide details on our tasks, while Secs. 4.4.2, 4.4.2.4 report extensive qualitative and quantitative evaluation.

4.4.1 Tasks definitions

We describe our tasks, detailing the local domains translation $X_\beta \mapsto X_\alpha$.

4.4.1.1 Lane degradation

Here, we use the highway TuSimple (TuSimple, 2021) dataset having clear lane markings. For local domains, we chose *lane marking* (X_β) and *asphalt* (X_α) exploiting geometrical priors from the provided lane labels and assuming nearby asphalt. We use our interpolation strategy (Sec. 4.3.3) accounting for both degradation and blending. Importantly, we train only on 15 images (1280x720) to demonstrate few-shot capabilities, with 30 patches per image of size 128x128, 200x200 and 256x256. Backbones are DeepFillv2 (Yu et al., 2019) as GAN and IntroVAE (Huang et al., 2018a) for interpolation. The latter is trained with a binarized difference mask from lane inpainting and original image. We evaluate our translations on the standard 358/2782 val/test sets of TuSimple. In addition to demonstrating generalization, we evaluate several lane detectors on 110 images from the India Driving Dataset (IDD) (Varma et al., 2019) – never seen during training – having degraded lane markings which we manually annotated.



(a) Qualitative

Network	FID↓	LPIPS↓
DLOW (Gong et al., 2019)	211.7	0.4942
DLOW+	155.6	0.4206
Ours w/o blending	154.7	0.3434
Ours	**135.4**	**0.3254**

(b) GAN metrics

Figure 4.5: Lane translations. (a) Qualitative comparison of lane degradation on patches with baselines. Our method is the only one to output a realistic degradation. (b) GAN metrics on the lane degradation task prove the benefit of our method.

4.4.1.2 Snow addition

Here, we rely on snowy images from the recent Adverse Driving Conditions Dataset (ACDC) (Sakaridis et al., 2021), which typically have snow only on the sidewalk and *not* on the road. The task is to add snow on the road. Logically, local domains are *road* (X_β) and snowy *sidewalk* (X_α), exploiting semantic labels as priors. Again, we train only with 15 images with 30 patches (128x128) per image , using CycleGAN (Zhu et al., 2017a) with default hyperparameters. No interpolation is used.

We evaluate on the original val/test set of ACDC having 100/500 images. To increase generalization for the segmentation task in snowy weather, we also augment Cityscapes (Cordts et al., 2016) with the same trained network.

4.4.2 Evaluation

4.4.2.1 Translation quality

Qualitative results are visible in Fig. 4.4 and show our method outputs realistic translations for all tasks. In detail, we are able to modify lanes (first three rows) on TuSimple with different degrees of degradation (from left to right). On snow addition, images show plausible snow on ACDC roads (middle two rows), preserving shadows.

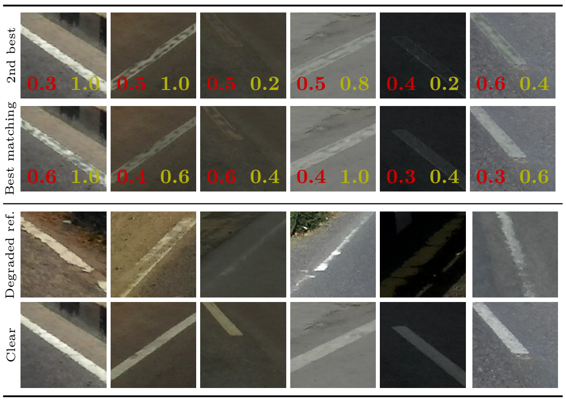


Figure 4.6: Evaluation of lane degradation on patches taken from IDD dataset (Varma et al., 2019). We associate clear patches (bottom row) to degraded ones (third row) by minimizing LPIPS. Applying our method to clean images variating the z and γ parameters (shown in the images), we subsequently lower the LPIPS. We display the best and second-best translation in terms of LPIPS. The similarities of our results with the degraded patches prove the efficacy of our LPIPS-based evaluation.

4.4.2.2 Interpolation quality

For the lane degradation task, we compare our interpolations against the continuous i2i DLOW (Gong et al., 2019) baseline, trained on the same data. As it suffers from evident color artifacts, we introduce DLOW+: a custom version using lane mask as additional channel input, masked reconstruction loss, and masked input-output blending.

For DLOW/DLOW+, we regulate the walk on the discovered manifold of each network with a *domainness* variable z – which amounts to our lane degradation.

With respect to baselines, it is visible in Fig. 4.5a that our degraded lane translations are more realistic for all z since DLOW and DLOW+ discover simpler transformations, just regulating color homogeneously.

For quantification, we compare translations against real degraded lane markings from IDD and report FID and LPIPS in Fig. 4.5b. In detail, we select 35/62 clear/degraded lane patches from IDD test set, and couple those with minimum LPIPS (Zhang et al., 2018a) distance. Intuitively, we pair similar clear and degraded lane markings together. Pairs are shown in the two bottom rows of Fig. 4.6. We then degrade each clear image with ours / DLOW / DLOW+, generating several degraded versions, and use the best degrading version in terms of LPIPS w.r.t. its clear match to compute GAN metrics. Fig. 4.5b shows we outperform baseline on both metrics significantly (roughly, -20 FID, and -0.1 LPIPS), demonstrating the realism of our lane degradation.

Detector	Translation	TuSimple			IDD		
		Acc. ↑	FP ↓	FN ↓	Acc. ↑	FP ↓	FN ↓
SCNN (Pan et al., 2017)	none (source)	**0.946**	**0.052**	**0.069**	0.617	0.538	0.741
	Ours	0.945	0.058	0.072	**0.730**	**0.453**	**0.577**
RESA (Zheng et al., 2021)	none (source)	**0.952**	**0.056**	**0.065**	0.639	0.720	0.800
	Ours	0.951	0.059	0.068	**0.671**	**0.686**	**0.761**

Table 4.1: Lane detection on TuSimple and IDD. Performance of lane detectors when trained on TuSimple source (*none*) or our degraded translations (*ours*). The latter significantly outperforms baseline, while retaining equivalent performances on TuSimple images.

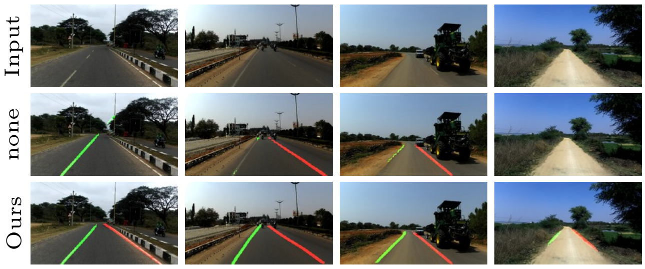


Figure 4.7: SCNN (Pan et al., 2017) lane detection on IDD (Varma et al., 2019). Training on generated images with degraded lanes makes existing lane detectors – such as SCNN (Pan et al., 2017) – resistant to scenes with damaged (first three columns) or no (last column) lane markings.

Since baselines are not using any explicit blending as us (see γ in Eq. 4.7), we also evaluate "ours w/o blending" using $m = p_z$ in Eq. 4.7, which still outperforms baselines.

4.4.2.3 Downstream tasks

Here, we study the applicability of our pipeline to increase the robustness of existing lane detection and semantic segmentation networks.

Lane detection. We aim here to make lane detectors robust to *unseen* degraded lane markings. To do so, we train two state-of-the-art detectors, SCNN (Pan et al., 2017) and RESA (Zheng et al., 2021), on both TuSimple original images and our translated version (mixing with 5% probability and randomizing z and γ). The models are tested on both the TuSimple test set and our 110 labeled IDD images, the latter having severely degraded lane markings.

From the quantitative results in Tab. 4.1, we observe that with our source degraded translations both detectors severely outperform the baselines using clear source on the challenging IDD, while maintaining on-par performances

Model	Translations	road IoU ↑	sidewalk IoU ↑	mIoU ↑
DeepLabv3+ (Chen et al., 2018)	none (source)	74.95	39.52	45.31
	Ours	**80.56**	**49.52**	**47.64**
PSANet (Zhao et al., 2018)	none (source)	**74.29**	30.71	42.97
	Ours	74.01	**36.28**	**43.85**
OCRNet (Yuan et al., 2020)	none (source)	82.30	45.60	54.54
	Ours	**82.78**	**54.69**	**55.48**

Table 4.2: Semantic segmentation on ACDC (Sakaridis et al., 2021) snow. We train multiple segmentation networks on Cityscapes (Cord and Aubert, 2011) with added snow with our method and test on ACDC (Sakaridis et al., 2021) snow validation, consistently improving generalization capabilities.

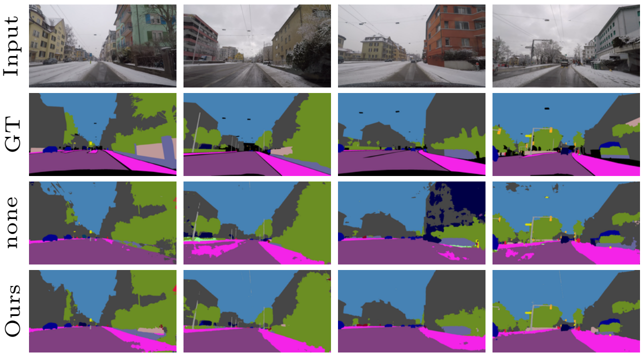


Figure 4.8: DeepLabv3+ (Chen et al., 2018) on ACDC (Sakaridis et al., 2021) snow. Training with our generated images brings improvements in segmentation of snowy scenes in ACDC (Sakaridis et al., 2021), especially in the road and sidewalk classes.

on TuSimple with clear markings. In particular, for SCNN we improve by +11.3% the accuracy, −8.5% the false positives and −16.4% the false negatives. Sample qualitative results are in Fig. 4.7 and showcase the robustness of our method on degraded or even absent street lines. We conjecture that our degraded translations forced the network to rely on stronger contextual information.

Semantic segmentation. Here, we seek to improve segmentation in snowy driving conditions. We train three state-of-the-art semantic segmentation models, namely DeepLabv3+ (Chen et al., 2018), PSANet (Zhao et al., 2018) and OCRNet (Yuan et al., 2020), with either clear Cityscapes images and snowy Cityscapes images translated with our method. We add snow on sidewalks and roads by using Cityscapes semantic maps. Snow is added uniformly on both semantic classes, even if inference on Cityscapes brings a consistent domain shift with respect to training patches on ACDC. In de-

TuSimple samples (%)	Patches/img	LPIPS↓	FID↓
15 (0.4%)	1	0.3296	148.22
15 (0.4%)	5	0.3295	135.53
15 (0.4%)	30	0.3254	131.73
15 (0.4%)	60	0.3246	127.94
15 (0.4%)	150	**0.3222**	**126.94**
50 (1.4%)	30	0.3236	129.42
150 (4.1%)	30	0.3221	**124.79**
500 (14%)	30	0.3234	128.56
3626 (100%)	30	**0.3218**	125.56

Table 4.3: Data ablation on TuSimple. The use of data on the lane degradation task (TuSimple ↦ IDD) is ablated by varying the number of images and patches per image in the training set, and evaluating GAN similarity metrics (see Sec. 4.4.2.2) on IDD.

tail, for the latter we augment images with 10% (DeepLabv3+, PSANet) or 5% (OCRNet) probability. The models are evaluated on the ACDC snow validation set.

Tab. 4.2 shows the benefit of our augmented images (*Ours*) to consistently improve the performance on road or sidewalk (our two local domains) and mean IoU for all networks. From Fig. 4.8 it is visible that the model trained with our augmentation strategy is able to better detect roads and footpaths in difficult weather conditions with respect to the baseline, which is not capable of properly discriminating between them if they are covered with snow.

4.4.2.4 Images and patches number ablation

Our method requires very few images to train. Here, we study the effect of number of images and patches per image on the lane degradation task. To measure its impact, we use LPIPS (Zhang et al., 2018a) and FID (Heusel et al., 2017a) following Sec. 4.4.2.2.

Results in Tab. 4.3 show, as expected, better translation with the increase of both the number of images and the number of patches extracted per each image. However, we also denote the few-shot capability of our method and the minimal benefit of using a large number of images. Other ablation studies in the main paper have been omitted for brevity.

4.5 Conclusion

In this chapter, we proposed a patch-based image-to-image translation model which relies on a GAN backbone trained on patches and an optional VAE to interpolate non-linearly between domains. Along with the definition of *local domains*, we introduced a dataset-based geometrical guidance strategy to ease the patches extraction process. Our few-shot method outperformed the literature on all tested metrics on several tasks, and its usability has been demonstrated on proxy tasks. In particular, our translation pipeline led to higher performances on lane detection in scenes with degraded or absent markings and on semantic segmentation in snowy conditions.

Although the VAE-based interpolation shows some kind of non-linearity, it is still far to be accurate for more complex transformations. The reason for this is the absence of guidance on how the continuous transformation should be, which we develop instead in Chapter 8.

Moreover, while exhibiting few-shot capabilities, our patch-based training strategy is able to model only very subtle transformations, since it loses completely the context from which patches are extracted. When trained on more complex transformations, *e.g.* from daytime to nighttime, more contextual understanding is required to get realistic outputs. For this reason, we complement patch-based training with additional insights in Chapter 5 to achieve few-shot image translation.

Chapter 5

Few-shot learning for image translation with anchor domains

The contributions of this chapter have been published in:

Pizzati, F., Lalonde, J.-F., and de Charette, R. (2022). ManiFest: Manifold Deformation for Few-shot Image Translation. In *ECCV*

arXiv: https://arxiv.org/abs/2111.13681
Code: https://github.com/cv-rits/ManiFest

Contents

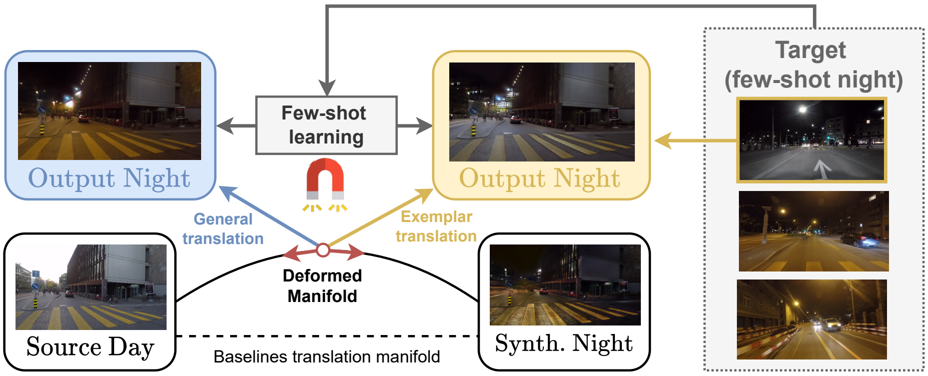


Figure 5.1: Overview of ManiFest, which translates images from a source domain (here, Day) to a few-shot target (Night). Our framework learns a manifold between *anchor* domains, in the example spanning the translation between Day and Synthetic Nighttime. Our system deforms the manifold by injecting the few-shot domain information between anchor style representations, and further departs from the deformed manifold by learning to approximate the target domain *general* appearance, or to reproduce the style of a particular *exemplar*.

5.1 Problem statement

When performing image translation, it is unrealistic to impose significant data collection constraints every time a new scenario is pursued. In addition to the complex logistics involved in acquiring large quantities of images, some scenarios may be rare (*e.g.*, auroras) or dangerous (*e.g.*, erupting volcanoes) thereby preventing even the capture of sufficient training data. Even though during the thesis some few-shot behavior already emerged (see Chapter 4), this only works for simple scenarios in which the transformation is not involving all the scene. Existing methods have been proposed to alleviate the requirement for large datasets, but they mostly show realistic results in highly structured environments such as face translation (Liu et al., 2019b; Saito et al., 2020; Li et al., 2021). In this context, we propose ManiFest, a framework for few-shot image-to-image translation which is shown to be robust to highly unstructured transformations such as adverse weather generation or night rendering. Our approach, illustrated in Fig. 5.1, starts from the observation that features consistency (*i.e.* which image parts should be translated together) is crucial for i2i (Ma et al., 2019) and that the few-shot domain offers little cues to train efficiently without overfitting (Ojha et al., 2021). Indeed, rather than directly addressing few-shot i2i, ManiFest exploits features learned on a stable manifold for the few-shot domain transformation. To do so, we exploit domain-level supervision by humans to include new data in the training process, which are related to source in such a way that an additional translation manifold could be learned. Importantly, ManiFest does not require reasoning on possible decompositions of the target domain (as we proposed instead in Sec. 3), but it is rather resistant to the choice of additional data, as long as an image translation network is trainable on those. In other words, the assumption we make is to be able to identify some common domain among source and few-shot target (*e.g.* "Street scenario") and to collect additional data belonging to the same domain. We aim to benefit from learned relationships among regions learned on the included data, and transfer them to the few-shot set, dramatically improving translation quality compared to alternatives. Additionally, we learn either to translate to some *general* style approximating the entire few-shot set, or to reproduce a specific *exemplar* from it. In short, our contributions are:

- ManiFest, a few-shot image translation framework using feature consistency by weighted manifold interpolation (WMI) and local-global few-shot loss (LGFS).
- We introduce GERM, a novel residual correction mechanism for enabling general and exemplar translation, that also boosts performances.
- Our framework outperforms previous work on adverse weather and low-light few-shot image translation tasks. We also present qualitative evaluations on rare (auroras) and dangerous (volcanoes) events.

5.2 Related work

Considering this project is about few-shot learning on image translation network, we review the state-of-the-art for few-shot GANs. Moreover, artistic style transfer normally works with with only one image, so it could be relevant to the work.

GANs with limited data. There have been several attempts to overcome the large data requirement for training GANs. Some use transfer learning (Torrey and Shavlik, 2010) to adapt previously-trained networks to new few-shot tasks (Wang et al., 2021; Li et al., 2020). In particular, (Ojha et al., 2021) uses a patch-based discriminator to generalize to few-shot domains. However, these methods are designed for generative networks and do not immediately apply to i2i. Another line of work focuses on the limited data scenario (Patashnik et al., 2021; Cao et al., 2021; Zhao et al., 2020; Karras et al., 2020), but usually performs poorly when very few (10–15) images are used for training. Others exploit additional knowledge to enable few-shot or zero-shot learning, such as pose-appearance decomposition (Wang et al., 2020), image conditioning (Endo and Kanamori, 2021) or textual inputs (Lin et al., 2021). FUNIT (Liu et al., 2019b) and COCO-FUNIT (Saito et al.,

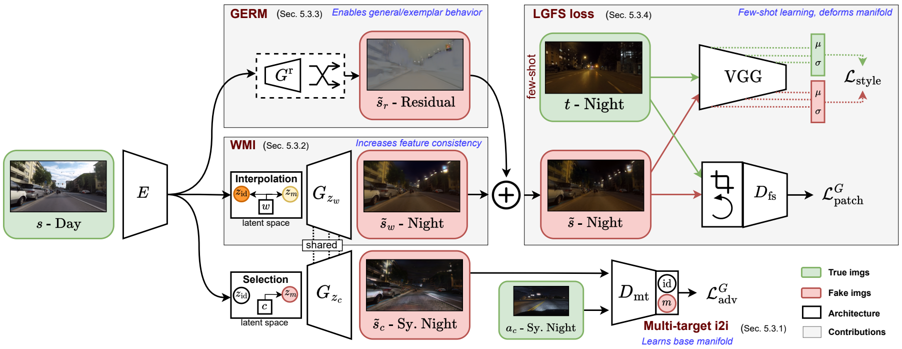


Figure 5.2: ManiFest architecture, here translating Day ↦ Night using few real night images and a synthetic night anchor domain. The encoded image representation $E(s)$ is separated into content and style codes, and translated to the few-shot domain by injecting $\mathcal{T}$ on a manifold learned on anchor domains in a multi-target i2i setting (bottom). We correct the output by using residuals estimated by the GERM (top). The LGFS loss (top-right), based on statistics alignment and patch-based adversarial learning, deforms the manifold and injects $\mathcal{T}$ in it. The reconstruction cycle with a style encoder is omitted for simplicity and follows (Huang et al., 2018b).

2020) use few-shot style encoders to adapt the network behavior at inference time. Some use meta-learning to adapt quickly to newly seen domains (Lin et al., 2020b). Those methods show limited performance on unstructured scenarios.

Neural style transfer. Style transfer could be seen as an instance of few-shot i2i, where the goal is to combine the content of an image with the style of another (Gatys et al., 2016). This often results in distortions, which some work tried to mitigate (Luan et al., 2017). The first examples for style transfer with arbitrary input style images are in (Huang and Belongie, 2017; Li et al., 2017). Others try to transfer styles in a photo-realistic manner by using a smoothing step (Li et al., 2018b) or by using wavelet transforms (Yoo et al., 2019). While these methods provide good results in some controlled scenarios, they may fail to understand the contextual mapping between source and style images elements (*e.g.* sky, buildings, etc.) which we learn more accurately.

5.3 The ManiFest methodology

The few-shot i2i task consists in learning a $\mathcal{S} \mapsto \mathcal{T}$ mapping between images of source domain $\mathcal{S}$ and target domain $\mathcal{T}$ containing few training samples (*e.g.*, $|\mathcal{T}| \leq 25$). Fig. 5.2 presents an overview of our approach. We learn a style manifold in a standard multi-target GAN fashion (Sec. 5.3.1) from a set of domains which contain large amounts of training data. We call these domains *anchors*, and denote them $\mathbb{A}$. Anchors domains have the requirement of be suitable for learning a $\mathcal{S} \mapsto \mathbb{A}$ transformation, thus requiring domain knowledge to select suitable data (*e.g.*, if working with autonomous driving scenes, street scenarios). The idea of ManiFest is to simultaneously 1) learn a stable manifold using anchor domains and 2) perform few-shot training by enforcing the target style appearance to lie within the learned manifold. This allows to exploit additional knowledge, like feature consistency (*i.e.*, image parts to be translated together), learned on anchors. To this end, Weighted Manifold Interpolation (WMI, Sec. 5.3.2) exploits style interpolation to benefit from the learned feature consistency on anchors. We allow to further depart from the interpolated manifold with the General-Exemplar Residual Module (GERM, Sec. 5.3.3) which learns a residual image refining the overall appearance and thus enabling style transfer to the *general* few-shot style (approximating the entire set $\mathcal{T}$), or to a single *exemplar* in $\mathcal{T}$ as in (Ma et al., 2019). We learn the appearance of $\mathcal{T}$ and inject it in the manifold with the Local-Global Few-Shot loss (LGFS, Sec. 5.3.4). In the following, *real* images are $s \in \mathcal{S}, t \in \mathcal{T}$, and *fake* ones $\tilde{s} \in \mathcal{T}$ where $\tilde{s}$ is our output.

5.3.1 Multi-target i2i

Instead of learning $\mathcal{S} \mapsto \mathcal{T}$ directly, we assume the availability of a set of two *anchor* domains, $\mathbb{A} = \{\mathcal{A}_{\text{id}}, \mathcal{A}_m\}$, with abundant data (equivalent to the "base" categories in few-shot image classification, *e.g.*, (Chen et al., 2019a)). By construction, one anchor is always the identity domain ($\mathcal{A}_{\text{id}} = \mathcal{S}$), while the other ($\mathcal{A}_m$) contains images easier to collect with respect to $\mathcal{T}$, for example synthetic images or images from existing large-scale datasets. We formalize the multi-target image translation problem as learning the $\mathcal{S} \mapsto \mathbb{A}$ mapping. At training time, we disentangle image content and appearance by using content and style encoders $E(\cdot)$ and $Z(\cdot)$ respectively, following MUNIT (Huang et al., 2018b). We use Z for reconstruction and translation as in (Huang et al., 2018b), which we refer for details. We can reconstruct $s = G_{Z(s)}(E(s))$, where $G_{Z(s)}$ is the style injection of $Z(s)$ into G as in (Huang et al., 2018b). This effectively learns latent style distributions as in (Huang et al., 2018b), namely here for each anchor $\{z_{\text{id}}, z_{\text{m}}\}$. A multi-target mechanism (following (Choi et al., 2020)) is employed in Z since we have two anchors. We translate to a randomly selected domain $c \in \{\text{id}, m\}$ with

$$z_c = [\![c = \text{id}]\!] z_{\text{id}} + [\![c = m]\!] z_m \,, \quad \tilde{s}_c = G_{z_c}(E(s)) \,, \tag{5.1}$$

where $[\![\cdot]\!]$ are the Iverson brackets. The translation to a given anchor style is depicted in Fig. 5.2 as "**selection**". The multi-target discriminator D_{mt}

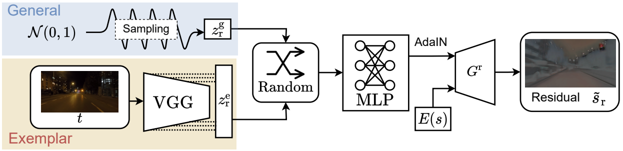

General
$\mathcal{N}(0,1)$
Sampling
$z_{\mathrm{r}}^{\mathrm{g}}$
t
VGG
$z_{\mathrm{r}}^{\mathrm{e}}$
Exemplar
Random
MLP
AdaIN
G^{r}
$E(s)$
Residual $\tilde{s}_{\mathrm{r}}$

Figure 5.3: GERM-based residuals. We perform either *exemplar*- or *general*-based transformations on the few-shot set by learning residuals conditioned on original image features ($E(s)$) and extracted statistics or noise, respectively. At training time, we alternate randomly the two modalities.

employs adversarial losses $\mathcal{L}_{\text{adv}}^{G}$ and $\mathcal{L}_{\text{adv}}^{D}$ to force fake images $\tilde{s}_c$ to resemble $a_c \in \mathcal{A}_c$.

5.3.2 Weighted Manifold Interpolation (WMI)

Our intuition is that encoding $\mathcal{T}$ between the linearly interpolated style representations of $\mathbb{A}$ should enforce feature consistency in $\mathcal{T}$. For instance, assuming $\mathcal{S}$ = *day*, $\mathcal{T}$ = *night*, $\mathcal{A}_m$ = *synthetic night*, the network will be provided with the information that all sky pixels should be darkened together.

In practice, we learn weights $w = \{w_{\text{id}}, w_m\}$ which sum to 1 and encode an image $\tilde{s}_w$ with feature consistency by interpolating the anchors style representations:

$$z_w = w_{\text{id}} z_{\text{id}} + w_m z_m \,, \quad \tilde{s}_w = G_{z_w}(E(s)) \,. \tag{5.2}$$

This is visualized in Fig. 5.2 as **"interpolation"**. Learning w allows us to determine the point in the $\mathbb{A}$ manifold which is most consistent with $\mathcal{T}$. This point is learned with the LGFS loss (Sec. 5.3.4).

5.3.3 General-Exemplar Residual Module (GERM)

Our GERM seeks to further increase realism by learning a residual in image space. Moreover, our design enables distinguishing between *general* and *exemplar* translations. The idea is to allow deviations from the $\mathbb{A}$ manifold by learning a residual image $\tilde{s}_{\mathrm{r}}$ which helps encode missing characteristics from $\mathcal{T}$. This is done by processing the input image features $E(s)$ with a generator G^{r} such that

$$\tilde{s}_{\mathrm{r}} = G_{z_{\mathrm{r}}}^{\mathrm{r}}(E(s)) \,, \quad \text{and} \quad \tilde{s} = \tilde{s}_w + \tilde{s}_{\mathrm{r}} \,, \tag{5.3}$$

where z_{r} is a vector controlling general- or exemplar-based modalities. In both cases, we draw inspiration from AdaIN style injection (Huang et al., 2018b) and condition the injected parameters on different vectors, as illustrated in Fig. 5.3.

For the *exemplar* residual, the style of a specific image $t \in \mathcal{T}$ as in (Ma et al., 2019) is reproduced by conditioning the residual on t. In this case,

$$z_{\mathrm{r}}^{\mathrm{e}} = (\mu_k(t), \sigma_k(t))|_{k=1}^{K}, \quad \tilde{s}_{\mathrm{r}} = G_{z_{\mathrm{r}}^{\mathrm{e}}}^{\mathrm{r}}(E(s)), \tag{5.4}$$

where $\mu_k(\cdot) = \mu(\phi_k(\cdot))$ and $\sigma_k(\cdot) = \sigma(\phi_k(\cdot))$ are mean and variance of the k-th out of K layer outputs ϕ_k of a pretrained VGG network (Huang and Belongie, 2017), and $|$ is the concatenation operator. Since the LGFS loss exploits VGG statistics (Sec. 5.3.4), G^{r} will be driven to exploit the additional information provided by the input statistics vector, effectively making the generated image more similar to t.

We learn a *general* residual by removing the conditioning on t and by injecting random noise instead.

$$z_{\mathrm{r}}^{\mathrm{g}} \sim \mathcal{N}(0,1), \quad \tilde{s}_{\mathrm{r}} = G_{z_{\mathrm{r}}^{\mathrm{g}}}^{\mathrm{r}}(E(s)). \tag{5.5}$$

5.3.4 Local-Global Few-Shot loss (LGFS)

To guide the learning, the resulting image $\tilde{s}$ is compared against the few-shot training set $\mathcal{T}$ with a combination of two loss functions. First, we take inspiration from the state-of-the-art of image style transfer where one image is enough for transferring the *global* appearance of the style scene (Huang and Belongie, 2017). Our intuition is that feature statistics alignment, widely used in style transfer, could be less prone to overfitting with respect to adversarial training. Therefore, we align features between $\tilde{s}$ and a target image $t \in \mathcal{T}$ using style loss $\mathcal{L}_{\mathrm{style}}$ as in (Huang and Belongie, 2017)

$$\mathcal{L}_{\mathrm{style}} = \sum_{k=1}^{K} ||\mu_k(\tilde{s}) - \mu_k(t)||_2 + ||\sigma_k(\tilde{s}) - \sigma_k(t)||_2, \tag{5.6}$$

where (μ_k, σ_k) are the same as in Sec. 5.3.3. While this is effective in modifying the general image appearance, aligning statistics alone is insufficient to produce realistic outputs. Thus, to provide *local* guidance, *i.e.*, on more fine-grained characteristics, we employ an additional discriminator D_{fs} which is trained to distinguish between rotated patches sampled from $\tilde{s}$ and t. We define the adversarial losses (Mao et al., 2017a):

$$\begin{aligned} \mathcal{L}_{\mathrm{patch}}^{G} &= ||D_{\mathrm{fs}}(p(\tilde{s})) - 1||_2, \\ \mathcal{L}_{\mathrm{patch}}^{D} &= ||D_{\mathrm{fs}}(p(\tilde{s}))||_2 + ||D_{\mathrm{fs}}(p(t)) - 1||_2, \end{aligned} \tag{5.7}$$

where p is a random cropping and rotation function. Note how the *exemplar* residual (from Sec. 5.3.3) is conditioned on the *same* feature statistics used here—this is what enables the exemplar-based behavior of the network. Also note the interaction between components: backpropagating the LGFS loss *deforms* the manifold learned by multi-target i2i, at the point identified by WMI, thereby injecting $\mathcal{T}$ "between" $\{\mathcal{A}_{\mathrm{id}}, \mathcal{A}_m\}$.

5.3.5 Training strategy

Our framework is fully trained end-to-end and optimizes

$$\min_{\Theta(E,G,G^r,Z),w} \mathcal{L}_{\text{style}} + \mathcal{L}^G_{\text{patch}} + \mathcal{L}^G_{\text{adv}} \quad \text{and} \min_{\Theta(D_{\text{fs}},D_{\text{mt}})} \mathcal{L}^D_{\text{patch}} + \mathcal{L}^D_{\text{adv}}\,, \tag{5.8}$$

where $\Theta(\cdot)$ refers to the network parameters. We train GERM (Sec. 5.3.3) by randomly selecting one of the exemplar or general mode at each training iteration. For the multi-target settings, we adapt the discriminator and the style encoder of our backbone in a multi-target setup following (Choi et al., 2020).

5.4 Experiments

We leverage 4 datasets (Sakaridis et al., 2021, 2020; Cordts et al., 2016; Richter et al., 2017) and 3 translation tasks (Sec. 5.4.1.2) and evaluate performances against recent baselines (Liu et al., 2019b; Saito et al., 2020; Yoo et al., 2019; Ma et al., 2019; Huang et al., 2018b) (Sec. 5.4.2). We further demonstrate the benefit of our few-shot translation on a downstream segmentation task (Sec. 5.4.3), and rare few-shot scenarios (Sec. 5.4.4), and finally ablate our contributions (Sec. 5.4.5). In all, we use MUNIT (Huang et al., 2018b) as our backbone.

5.4.1 Training setup

5.4.1.1 Datasets

ACDC. We use ACDC (Sakaridis et al., 2021) for most of our experiments, using the night/rain/snow/fog conditions with 400/100/500 images for train/val/test respectively, following official splits. For any individual condition, ACDC also includes geolocalized weakly-paired clear weather day images of same splits.

Dark Zurich. Similar to ACDC, Dark Zurich (DZ) (Sakaridis et al., 2020) has daytime images paired with nighttime/twilight conditions. Here, we focus on twilight conditions exclusively and use training images from the GOPRO348 sequence only since it exhibits a distinctive twilight appearance. We split the total 819 image pairs into 25/794 for train/test, respectively.

Cityscapes. Cityscapes (Cordts et al., 2016) is used to evaluate ManiFest for training segmentation networks robust to nighttime[1]. It includes 2975/500/1525 annotated images for train/val/test.

[1] ACDC does not provide annotated daytime clear weather sequences.

VIPER. As anchors, we use synthetic images from VIPER (Richter et al., 2017), using the condition metadata to define splits. 4137/3090/1305/2018/2817 images are extracted from the VIPER training set for day/night/rain/snow/sunset conditions, respectively.

5.4.1.2 Tasks and evaluation

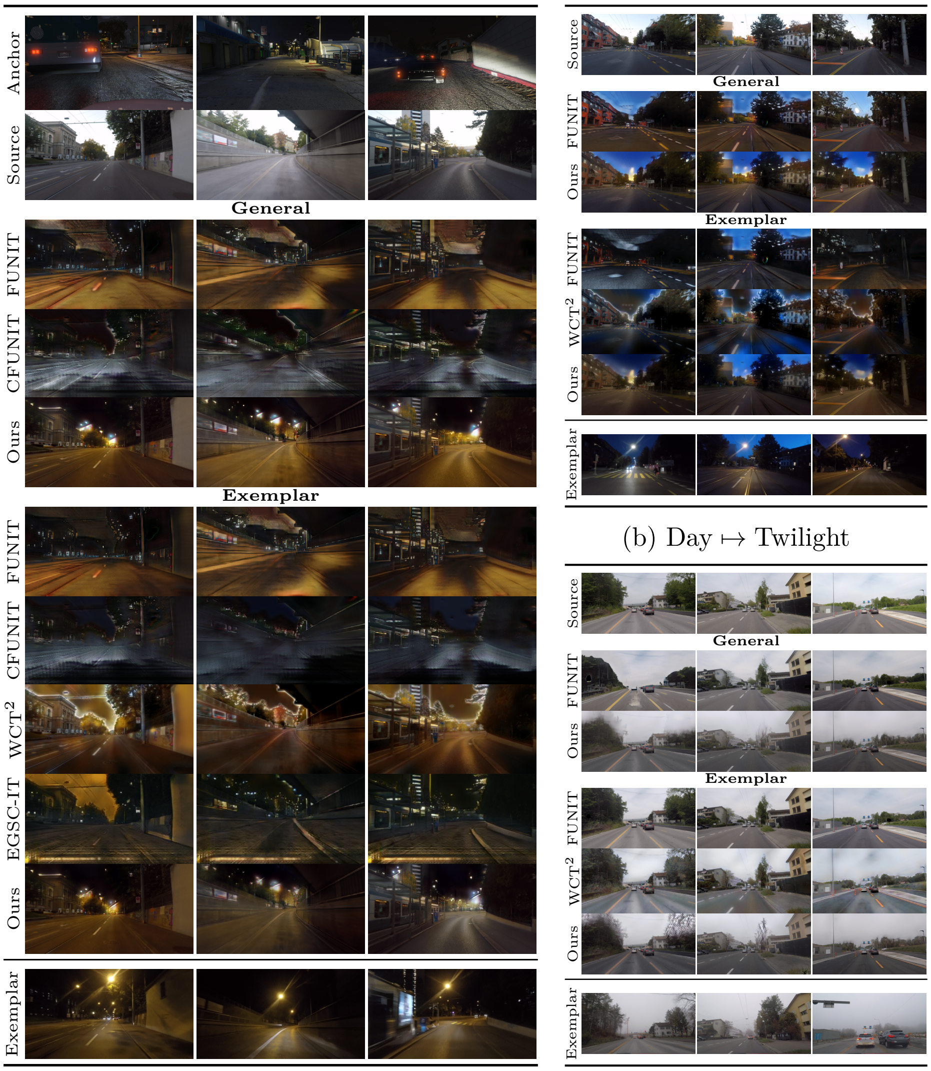
We train our framework on three main tasks:
Day ↦ ***Night*** on ACDC daytime ($\mathcal{S}$) and nighttime ($\mathcal{T}$).
Clear ↦ ***Fog*** on ACDC daytime ($\mathcal{S}$) and fog ($\mathcal{T}$).
Day ↦ ***Twilight*** on DZ daytime ($\mathcal{S}$) and twilight ($\mathcal{T}$).
Unless mentioned otherwise, the (synthetic) anchor domains from VIPER are "night" for Day ↦ Night and Day ↦ Twilight, and "day" for Clear ↦ Fog. We evaluate with the FID (Heusel et al., 2017b) and LPIPS (Zhang et al., 2018a) metrics. While FID compares feature distance globally, LPIPS compares translated source images and the geolocalized paired image in the target dataset. This is beneficial for evaluating our exemplar modality. For all, we train on downsampled x4 images.

5.4.2 Comparison with the state-of-the-art

Baselines. We compare with four baselines for few-shot image translation with $|\mathcal{T}| = 25$. We extensively evaluate on the most challenging Day ↦ Night task, and provide insights and comparison for the two others tasks. We evaluate the impact of the few-shot image selection and of $|\mathcal{T}|$ in Sec. 5.4.5.3. We compare against the recent FUNIT (Liu et al., 2019b) and COCO-FUNIT (Saito et al., 2020), trained on $\mathcal{S} \mapsto \mathcal{A}_m$ and adapted following (Liu et al., 2019b; Saito et al., 2020) to the few-shot $\mathcal{T}$ (general) or to a single reference image (exemplar). For exemplar image translation, we also add specific baselines. First, we compare with WCT^2 (Yoo et al., 2019), used to transfer the style of the paired target condition to the source one. We also evaluate EGSC-IT (Ma et al., 2019). The method is trained by merging $\mathcal{A}_m$ and $\mathcal{T}$ since it should be able to identify inter-domain variability, separating $\mathcal{T}$ styles from $\mathcal{A}_m$ (Ma et al., 2019). To define metrics bounds, we also train our MUNIT (Huang et al., 2018b) backbone on $\mathcal{A}_m$, on the full $\mathcal{T}$ set and on $\mathcal{T}$ with $|\mathcal{T}| = 25$. More comparisons with the backbone are in Sec. 5.4.5. We use the official code provided by the authors for all[2].

Evaluation. We compare qualitative results in Fig. 5.4. In Day ↦ Night (Fig. 5.4a), even if the appearance of images in $\mathcal{T}$ is partially transferred on translated images (*e.g.* road color, darker sky), FUNIT and COCO-FUNIT

[2] For FUNIT (Liu et al., 2019b) and COCO-FUNIT (Saito et al., 2020), we modify hyperparameters per authors suggestions to adapt to the ACDC and Dark Zurich datasets.



(a) Day ↦ Night

(b) Day ↦ Twilight

(c) Clear ↦ Fog

Figure 5.4: Qualitative evaluation and comparison with the state of the art. We evaluate the (a) Day ↦ Night, (b) Day ↦ Twilight, and (c) Clear ↦ Fog tasks. In all cases, our approach extracts a *general* realistic representation of the few-shot target, and correctly reproduces the style of paired *exemplar* target images. In comparison, existing baselines either has unnecessary similarity with anchors (*e.g.* FUNIT, COCO-FUNIT, EGSC-IT) or unrealistic artifacts (*e.g.* WCT^2). We report COCO-FUNIT as CFUNIT for space.

sstill retain some characteristics of $\mathbb{A}$ (note, for example, how the street is similar to the GTA one) which worsens the overall image realism. The same can be observed with EGSC-IT, where the hood of the ego-vehicle in anchor images (first column) is retained and significantly impacts visual results.

	Method	$\lvert\mathcal{A}_m\rvert$	$\lvert\mathcal{T}\rvert$	FID$_\downarrow$	LPIPS$_\downarrow$
G	MUNIT	0	400	79.20	0.529
	MUNIT	3090	0	132.72	0.613
	MUNIT	0	25	91.61	0.553
	FUNIT	3090	25	156.97	0.573
	COCO-FUNIT	3090	25	201.67	0.644
	Ours	3090	25	**81.01**	**0.535**
E	MUNIT	0	400	87.71	0.522
	MUNIT	3090	0	142.04	0.559
	MUNIT	0	25	128.73	0.562
	FUNIT	3090	25	136.2	0.572
	COCO-FUNIT	3090	25	193.4	0.646
	EGSC-IT	3090	25	106.68	0.574
	WCT^2	-	-	105.58	0.580
	Ours	3090	25	**80.57**	**0.525**

(a) Day ↦ Night

	Method	$\lvert\mathcal{A}_m\rvert$	$\lvert\mathcal{T}\rvert$	FID$_\downarrow$	LPIPS$_\downarrow$
G	FUNIT	3090	25	69.53	0.511
	Ours	3090	25	**63.15**	**0.510**
E	FUNIT	3090	25	69.97	0.501
	WCT^2	-	-	71.77	0.536
	Ours	3090	25	**58.07**	**0.483**

(b) Day ↦ Twilight

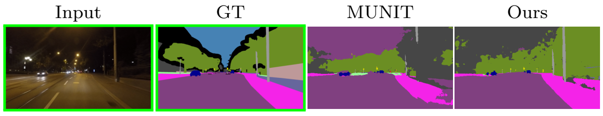
	Method	$\lvert\mathcal{A}_m\rvert$	$\lvert\mathcal{T}\rvert$	FID$_\downarrow$	LPIPS$_\downarrow$
G	FUNIT	3090	25	152.90	0.580
	Ours	3090	25	**89.57**	**0.520**
E	FUNIT	3090	25	137.7	0.568
	WCT^2	-	-	120.9	0.591
	Ours	3090	25	**89.89**	**0.521**

(c) Clear ↦ Fog

Table 5.1: Quantitative comparison with state of the art. We compare FID and LPIPS on the (a) Day ↦ Night, (b) Day ↦ Twilight and (c) Clear ↦ Fog tasks, for both **G**eneral and **E**xemplar translations. Our approach outperforms all baselines on all tasks, while also being on par (**G**) or even outperforming (**E**) the MUNIT backbone trained on the full dataset for Day ↦ Night in (a).

Model	mIoU % ↑	Acc. % ↑
Baseline (*CS day*)	12.93	45.15
MUNIT	21.22	56.65
Ours	**24.31**	**60.50**
Oracle (*ACDC night*)	49.23	88.47

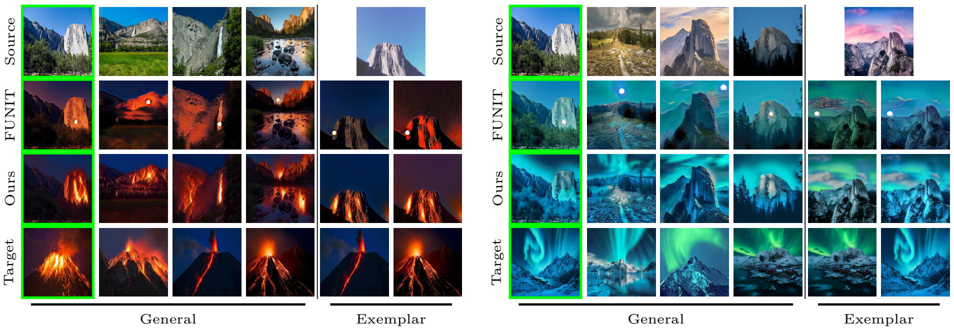
(a) Quantitative evaluation



(b) Qualitative evaluation

Figure 5.5: Segmentation on ACDC-night, for few-shot Day↦Night translations ($|\mathcal{T}| = 25$) (a). We outperform the baseline with noticeably better segmentation in (b) due to the increased quality of our translation.

While WCT^2 exhibits sharp results, it does not correctly map the image context, and it is limited to appearance alignment which leads to artifacts (*e.g.* yellow sky with white halos). Our method generates significantly better results than the baselines in both the *general* and *exemplar* modalities, with visible differences in all three tasks: the *general* appearance is consistent across test samples, and each result adapts to its *exemplar*. For example, observe how the overall sky colors (Day ↦ Twilight, Fig. 5.4b) match the exemplar. Here, the exemplars were unseen in training (not part of the few-shot set $\mathcal{T}$), thus GERM generalizes the few-shot learned exemplar behavior. The quantitative evaluation in Tab. 5.1 is coherent with the qualitative results, as we always outperform baselines. We perform on par (*general*), or even better (*exemplar*) than the backbone trained on the entire set of 400 training images on Day ↦ Night (Tab. 5.1a). This result shows that GERM (Sec. 5.3.3) improves modeling of the exemplar style over AdaIN exemplar



(a) Mountain ↦ Volcano

(b) Day ↦ Aurora

Figure 5.6: Qualitative results for the Mountain ↦ Volcano (a) and Day ↦ Aurora (b) tasks. We retain contextual information by only partially mapping mountains to volcanoes and sky to auroras. In the green box we process the same image for ease of comparison. *Exemplar* results show how Ours conforms to Target, effectively reproducing the exemplar image style (cols 5–6).

style injection (Huang et al., 2018b). The exemplar behavior may force artifacts following subtle characteristics of the scene (as trees in Fig. 5.4c), for which the general translation may be advisable.

5.4.3 Segmentation downstream task

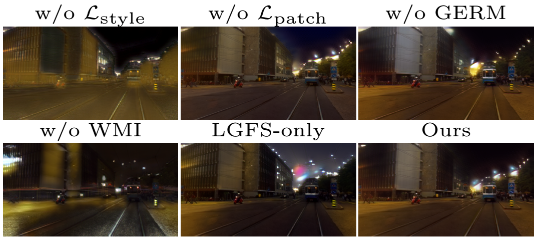
We exploit semantic segmentation to evaluate ManiFest for increasing robustness in challenging scenarios. In Fig. 5.5 we train HRNet (Wang et al., 2019a) on nighttime versions of Cityscapes (Cordts et al., 2016) obtained by translating the dataset with ManiFest or MUNIT, and evaluating on the ACDC-night validation set labels. We choose the best MUNIT and ManiFest configurations following nighttime realism in Tab. 5.1a with $|\mathcal{T}| = 25$. As lower and upper bounds we train HRNet either on original Cityscapes (baseline) or on ACDC-night training set (oracle). Fig. 5.5 shows we outperform the MUNIT backbone (+3.09 mIoU) thanks to our better target domain modeling.

5.4.4 Rare few-shot scenarios

Few-shot plays its full role with conditions that are rare by nature, difficult or even dangerous to photograph, such as auroras or erupting volcanoes. Fig. 5.6 shows the capability of ManiFest to learn Mountain↦Volcano or Day↦Aurora, by taking as source and anchor the summer and winter Yosemite dataset (Zhu et al., 2017a) splits respectively. Each task uses only 4 images from Google Images as $\mathcal{T}$. We generate realistic erupting volcanoes or auroras starting from mountain images, with contextual understanding

Component	FID$_\downarrow$	LPIPS$_\downarrow$
w/o $\mathcal{L}_{\text{style}}$	143.66	0.614
w/o $\mathcal{L}_{\text{patch}}$	93.42	0.566
w/o GERM	85.62	0.544
w/o WMI	101.57	0.589
LGFS-only	84.29	0.558
Ours	**81.01**	**0.535**

(a) Quantitative evaluation


w/o $\mathcal{L}_{\text{style}}$
w/o $\mathcal{L}_{\text{patch}}$
w/o GERM
w/o WMI
LGFS-only
Ours

(b) Qualitative evaluation

Figure 5.7: Ablation study for architectural components. (a) Removing each component individually lowers quantitative performances, which maps to (b) decreased visual quality in the generated images.

(Fig. 5.6, cols 1–4), where only one mountain is mapped to a volcano and auroras only partially cover the sky. Fig. 5.6 (cols 5–6) also demonstrate how exemplar characteristics are preserved.

5.4.5 Ablation studies

5.4.5.1 Architectural components

We evaluate the contribution of each component in ManiFest (*c.f.* Fig. 5.2, Sec. 5.3) using the Day $\mapsto$ Night task in the *general* scenario, and report results in Fig. 5.7a. The impact of LGFS is studied by removing $\mathcal{L}_{\text{style}}$ or $\mathcal{L}_{\text{patch}}$, showing that both local *and* global guidance are improving translations. Removing the GERM from the training pipeline simultaneously precludes the *exemplar* behavior and worsens the performance, demonstrating the effectiveness of encoding complementary characteristics outside of the manifold spanned by $\mathbb{A}$. The benefit of WMI is evaluated in two experiments. First, the "w/o WMI" setting applies the residual directly on the fake anchor images $\tilde{s}_c$, instead of the interpolated $\tilde{s}_w$ as in Eq. (5.3). The worse performance relate to synthetic characteristics present in $\tilde{s}_c$ (*e.g.* road texture in Fig. 5.7b). Second, "LGFS-only" directly uses the LGFS losses in substitution to $\mathcal{L}_{\text{adv}}$, without WMI and GERM components. While it only slightly worsens metrics, the impact on feature consistency is dramatic as shown in Fig. 5.7b, where the sky presents obvious artifacts and road trivially darkens.

5.4.5.2 Anchor selection

We ablate the choice of anchor domain $\mathbb{A}$ by selecting different conditions from the VIPER dataset, namely {Day, Night, Rain, Snow, Sunset}. In particular, we experiment on previous *intra-dataset* ($\mathcal{S}$ and $\mathcal{T}$ taken from the same dataset) tasks, as well as on a *cross-dataset* task in which

Day ↦ Night

$\mathcal{S}$	$\mathcal{T}$	$\mathcal{A}_m$	**FID**$_\downarrow$	**LPIPS**$_\downarrow$
ACDC-Day	ACDC-Night	Day	85.73	0.553
		Night	**81.01**	**0.535**
		Rain	81.38	0.549
		Snow	86.74	0.554
		Sunset	83.83	0.571
		All	83.71	0.547

Day ↦ Twilight

$\mathcal{S}$	$\mathcal{T}$	$\mathcal{A}_m$	**FID**$_\downarrow$	**LPIPS**$_\downarrow$
DZ-Day	DZ-Twilight	Day	64.19	0.505
		Night	63.15	0.510
		Rain	65.33	0.501
		Snow	64.09	0.513
		Sunset	63.78	0.504
		All	**60.98**	**0.469**

Clear ↦ Fog

$\mathcal{S}$	$\mathcal{T}$	$\mathcal{A}_m$	**FID**$_\downarrow$	**LPIPS**$_\downarrow$
ACDC-Clear	ACDC-Fog	Day	**89.57**	**0.520**
		Night	91.79	**0.520**
		Rain	93.15	0.522
		Snow	90.28	0.524
		Sunset	90.11	0.525
		All	92.19	**0.520**

(a) Intra-dataset

Day ↦ Twilight

$\mathcal{S}$	$\mathcal{T}$	$\mathcal{A}_m$	**FID**$_\downarrow$	**LPIPS**$_\downarrow$
ACDC-Day	DZ-Twilight	Day	89.61	*
		Night	90.48	*
		Rain	89.47	*
		Snow	91.49	*
		Sunset	91.77	*
		All	**85.15**	*

(b) Cross-dataset

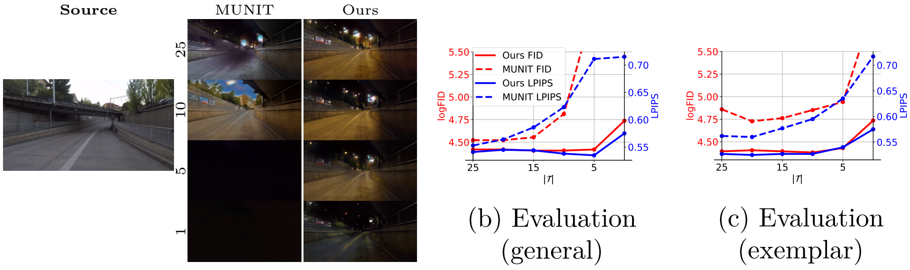
Table 5.2: Study of the impact of anchor domains $\mathbb{A}$ on the $\mathcal{S} \mapsto \mathcal{T}$ translations for intra-dataset (a) and cross-dataset (b) tasks. The stable performance across all tested anchors demonstrates the robustness of our method. For all, we test a multi-anchor setup by using all anchors ("All"). In (b), * means LPIPS cannot be computed due to lack of pairs of matched images (Sec. 5.4.1.1).

$\mathcal{S}$ = ACDC-Day and $\mathcal{T}$ = DZ-Twilight. Results in Tab. 5.2 show how performance remains relatively stable across most anchors. This may seem counter-intuitive since one could, for example, expect that the "Rain" anchor would be a poor choice for the Day ↦ Night task since rainy and night scenes look different. The results instead show that the WMI only encodes consistency in the transformation, and is thus robust to the choice of anchors. We also test a multi-anchor setup ("All" in Tab. 5.2), where $\mathbb{A} = \{\mathcal{A}_{\text{id}}$, Day, Night, Rain, Snow, Sunset$\}$. In general, more anchor domains improve performances, ranking either first or second in all cases for at least one metric, due to the additional information available for shaping the manifold in WMI. We hypothesize that multiple anchors helps identifying correspondences between $\mathcal{S}$ and $\mathcal{T}$, benefiting especially the cross-dataset tasks.

5.4.5.3 Number of images and variability

First we compare our Day ↦ Night translations against MUNIT (Huang et al., 2018b) for $|\mathcal{T}| = \{25, 20, 15, 10, 5, 1\}$, to understand the effects of few-shot training on the backbone network. Some qualitative *general* outputs are shown in Fig. 5.8a. While MUNIT overfits and creates unrealistic appearance (25–10 images) or collapse (5, 1 image), we output realistic transformations in all cases, even retaining the image context in the extreme one-shot scenario. This is confirmed by the FID and LPIPS in Figs 5.8b and 5.8c for the *general* and *exemplar* scenarios respectively.

In Tab. 5.3 we also study variability, evaluating FID and LPIPS for the *general* and *exemplar* cases for $|\mathcal{T}| = \{25, 15, 5, 1\}$ images reporting the results of 7 runs. Overall, the performance remains relatively constant with the exception of the one-shot setup, where despite realistic transfer, the metrics are penalized since the target image itself might not accurately represent the style distribution of the test set.



(b) Evaluation (general)

(c) Evaluation (exemplar)

(a) Qualitative results (general).

Figure 5.8: Comparison against MUNIT for varying $|\mathcal{T}|$: (a) qualitatively for the *general* scenario; as well as quantitatively for (b) *general* and (c) *exemplar*, always outperforming it.

$\lvert\mathcal{T}\rvert$	**FID**$_\downarrow$	**LPIPS**$_\downarrow$
25	82.95 ±2.95	0.541 ±1.85e-2
15	82.21 ±3.09	0.544 ±2.35e-2
5	83.11 ±2.49	0.535 ±2.24e-2
1	114.5 ±34.2	0.575 ±2.37e-2

(a) General

$\lvert\mathcal{T}\rvert$	**FID**$_\downarrow$	**LPIPS**$_\downarrow$
25	80.78 ±2.91	0.527 ±0.64e-2
15	80.55 ±2.85	0.527 ±1.07e-2
5	84.40 ±1.88	0.540 ±1.88e-2
1	114.3 ±33.5	0.575 ±2.40e-2

(b) Exemplar

Table 5.3: Day $\mapsto$ Night ablation on variability by training on 4 few-shot configurations with 7 runs each on general (a) and exemplar (b). $|\mathcal{T}|$ does not impact performance much except for the extreme one-shot scenario, where the network overfits to the seen style. The exemplar behavior performs better due to the style conditioning mechanism.

5.4.6 Anchor-based translation

The GERM module extracts residual information from encoded source images. We investigate the application of residuals on the anchor images themselves, by first translating from $\mathcal{A}_m \mapsto \mathcal{S}$ using our backbone cycle consistency (Huang et al., 2018b), and afterwards re-encoding the fake image in a $\mathcal{S} \mapsto \mathcal{A}_m$ reconstruction *without retraining* (see Fig. 5.9). This shows how ManiFest simultaneously learns $\mathcal{S} \mapsto \mathcal{T}$ and acceptable $\mathcal{A}_m \mapsto \mathcal{T}$ transformations. The FID w.r.t. ACDC-Night improves from 142 to 130 when applying the residual on the synthetic anchors, thus confirming their shift towards $\mathcal{T}$.

5.5 Conclusions

In this section we presented ManiFest, a framework for few-shot i2i which enables translating images using just few images from the target domain. We exploited ideas about patch-based training emerged in other projects (see Chapter 4), complementing them with additional strategies to achieve few-shot transfer of meaningful context-aware characteristics. In developing

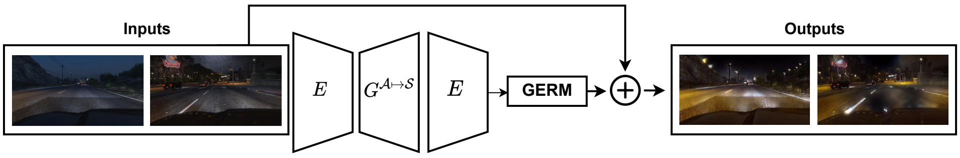


Figure 5.9: In the reconstruction cycle $\mathcal{A}_m \mapsto \mathcal{S} \mapsto \mathcal{A}_m$, we can inject the extracted residual with GERM on anchor images to perform an alternative $\mathcal{A}_m \mapsto \mathcal{T}$ transformation.

ManiFest, we exploited knowledge about domains to collect additional data to enforce consistency in the transformation. It is interesting to notice how the distance between few-shot target and anchor domains did non impact results significantly, advocating for further research on how features extracted by i2i networks could be reused.

Although the method could theoretically be used for any kind of transformation, it performs best when applied to low-level transformations as lighting or adverse weather. We suppose that further investigation could be required if we wanted to apply the same methodology on semantic-based transformations, as face transfers.

On GERM-G effectiveness. Reflecting on this work, we investigated the usability of exemplar capabilities in GERM as a replacement for the general pipeline, by simply averaging exemplar style codes in the few-shot domain. We mostly observed the same results, so it could be simply removed without loss of performance. This advocates the greater flexibility of exemplar-based image translation strategies (Ma et al., 2019).

Part II

Physics-informed learning

Chapter 6

Part Introduction

For a rather large number of systems which can be described by mathematics, it is evident that simple formulations formalize extremely well many classes of problems (Wigner, 1990), arguably better than it is achievable by learning on data. Moreover, on a general perspective about computer vision, depending only on data may have drawbacks. First, it is demonstrated that mostly bigger and more power-consuming vision architectures such as Vision Transformers (Dosovitskiy et al., 2021) can benefit from the increased image availability (Zhai et al., 2022), pushing computational costs in many applications and *de facto* reducing the number of organizations over the world that can afford to train such large models. Second, data are not always available, since some phenomena could be rare or difficult to capture in real life. This highlights that novel strategies for training vision systems exploiting different priors are necessary. But which ones?

An interesting clue comes from neuroscience research. Let us diverge a moment and reason about how humans learn to interact with the world. From the earliest stages of life we are all exposed to an enormous amount of visual data, that we learn to understand thanks to our inner world representation. According to Fischer et al. (2016), unsupervised learning of physics plays a big role in this. Since sensing our world is necessary for survival, evolution brought us to model physical aspects of what we see, and to learn abstract physical representations that we can reuse to predict motion, structure, and possible interactions of objects (Ullman and Tenenbaum, 2020). We can then assume physics understanding is a strong prior for better use of visual information. Moreover, humans tend to spontaneously correlate physics with visual perception (Mason and Just, 2016), in such a way that, for instance, looking at people dancing stimulates the same brain regions as thinking about periodic physical concepts (e.g. “wavelength”). A visualization of activation regions is available in Fig. 6.1. Again, a surprising ability of our brain!

It appears evident that understanding physics is leveraged by humans to

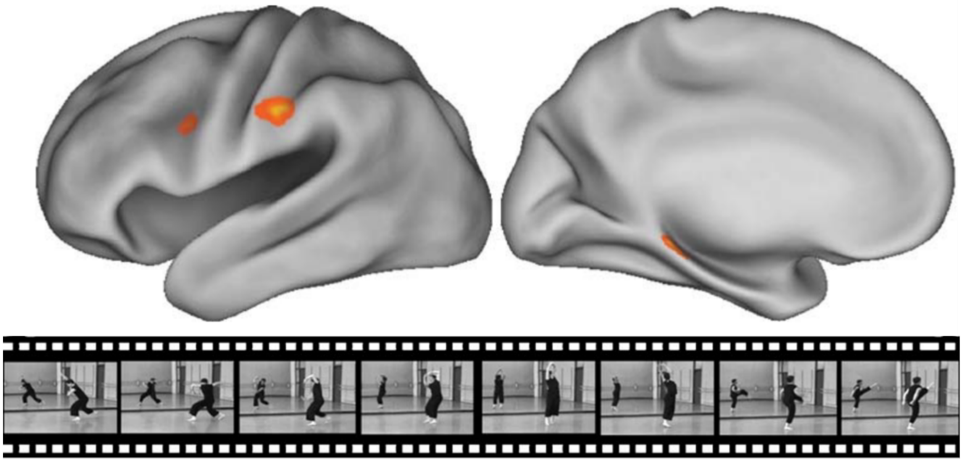


Figure 6.1: Activations in the brain of expert dancers specifically excited by videos of people dancing to music (see example frames). Similar neurons are stimulated when physics students are exposed to periodical physical concepts (Mason and Just, 2016). This intuitively implies an inner link between vision and physics understanding in humans. Image from (Cross et al., 2006).

properly exploit visual information. Consequently, it advocates for physics as a strong prior for training neural networks, as it is also demonstrated useful in recent works (Raissi et al., 2019). Nevertheless, the usage of physical constraints in generative networks remains only limitedly explored.

In this chapter, strategies for making use of physical priors in image translation are proposed. Ultimately, these *physics-informed learning pipelines* result in significant improvements of synthesized image quality, network capabilities, and relaxation of training constraints. Also, it is important noting how the usage of physics increases the interpretability of neural networks, since it provides strong well-studied priors to use during training.

The contributions of this chapter are two-fold. In Chapter 7, we address the entanglement problem in image translation by using physical knowledge. In there, we use realistic physical models for rendering visual effects of simple phenomena, as refraction of light by raindrops on a lens. With the usage of models, it was made possible to realistically generate rare out-of-distribution images, modifying explicitly controllable physical parameters. When realistic models are not available, naive models characterized by a very weak physical guidance (*e.g.* "night is darker than day") could be used to organize a GAN latent space, achieving continuous transformations. This resulted in the CoMoGAN framework presented in Chapter 8, which provided i2i networks with the capability of interpolating realistically between conditions which are continuous by nature, *e.g.* time of day or weather conditions.

Chapter 7

Physics-informed guided disentanglement for generative networks

The contributions of this chapter have been published in:

Pizzati, F., Cerri, P., and de Charette, R. (2020a). Model-based occlusion disentanglement for image-to-image translation. In *ECCV*

arXiv: https://arxiv.org/abs/2004.01071

Moreover, they have been significantly extended for a journal submission, currently under review:

Pizzati, F., Cerri, P., and de Charette, R. (2021b). Physics-informed guided disentanglement in generative networks. *TPAMI submission*

arXiv: https://arxiv.org/abs/2107.14229

Contents

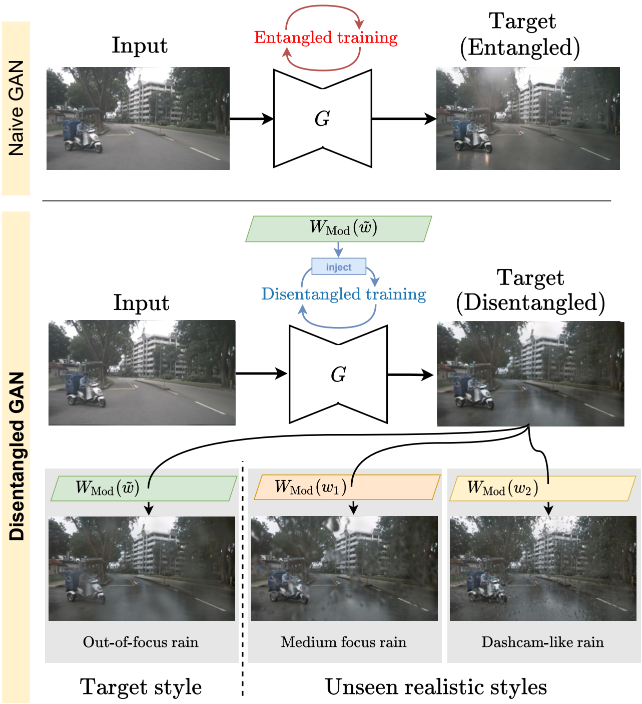


Figure 7.1: Guided disentanglement. While naive GANs generate all target scene traits at once (Target - Entangled), we learn a *disentangled* version of the scene from guidance of physical model $W_{\text{Mod}}(.)$ with estimated physical parameters $(\tilde{w})$. Our main idea is to combine physical models of well-known phenomena (as raindrops) with generative capabilities of GANs, in a complementary manner. Here, we combine a physical model for raindrops with wetness learned by the GAN (Target - Disentangled), *by only training on entangled data (i.e. rainy scene with raindrops on the lens)*. Notice here the unrealistic raindrops entanglement in naive GANs. With our method, we enable the generation of target style $(\tilde{w})$ or unseen scenarios (here, w_1, w_2).

7.1 Problem statement

A common pitfall of GANs is their inability to accurately learn the underlying physics of the transformation (Xie et al., 2018), often resulting in artifacts based on inaccurate mapping of source and target characteristics, which significantly impact results. This is the case for example when learning clear↦rain as a naive GAN translation will inevitably entangle inaccurate raindrops, as highlighted in Fig. 7.1 top. Domain-informed strategies used in Part I do not solve entanglement of physical traits, since in Fig. 3.4 raindrops are rendered always in the same positions for different style codes, and in Fig. 5.4c fog is always generated on trees. This is expected: distinguishing between different layers in images (*e.g.* the car glass layer and the scene one in a rainy urban scene) normally requires ad-hoc approaches (Xue et al., 2015). Physics-inspired models can instead render well-studied elements of target domain with great realism, for rain for example (Roser and Geiger, 2009; Alletto et al., 2019; Halder et al., 2019; Tremblay et al., 2020), though leaving any other appearance trait unmodified. For instance, in a rainy scene, models can accurately render raindrops but fail to render the complex scene wetness.

We propose a learning-based comprehensive framework to unify generative networks and physics priors. We rely on a disentanglement strategy that benefits from simple rendering physical models to learn the remaining un-modeled mapping. By relying on physical priors, we can achieve disentanglement without requiring ad-hoc data collection (Chapters 3, 5) or annotation (Chapter 4), thus achieving appealing results at nearly zero cost.

In brief, we render some of the target visual traits with a physical model and learn the un-modeled target characteristics with an i2i network. At inference, we compose them as shown in Fig. 7.1 to get the output benefiting from the visually pleasant outputs of GANs and the controllable characteristics of physical models. The peculiarity of our method is that we achieve disentanglement of modeled and learned characteristics *by just using data in which they are both present simultaneously.* For example, we can learn to generate wet scenes *without* raindrops on the lens, by only looking at rainy images *with* raindrops. Our strategy deeply differs from sequential composition of i2i and physics based rendering (Tremblay et al., 2020) which instead assume underlying independence of the two. Besides increasing image realism, our physical model-guided framework enables fine-grained control of physical parameters in rendered scenes, for increasing generated images variability regardless of the training dataset. This is beneficial for robotics applications, which require resistance to unobserved scenarios.

A remarkable use case is vision in rainy conditions since raindrops appearances vary drastically with the camera setup. From Fig. 7.1 bottom, our disentanglement can be used to be resistant to dashcam-like rain even having only seen out-of-focus rain at training. This capability is not achievable by using domain priors as in Part I, since in those cases we were still constrained by data, limiting the generative capabilities of our network to values in the seen data distribution. Other applications we demonstrate in this chapter are: vision for dirty images, fog, or composite watermarks.

7.2 Related works

We do not repeat related works relative to prior-based translations already studied in Chapter 4, which are also relevant to this part of the thesis. Moreover, we introduce two topics of interest.

Physics-based generation. Many works in literature rely on rendering to generate physics-based traits in images, for rain streaks (Garg and Nayar, 2006; Halder et al., 2019; Tremblay et al., 2020; Weber et al., 2015; Rousseau et al., 2006), snow (Barnum et al., 2010), fog (Sakaridis et al., 2018; Halder et al., 2019) or others. In many cases, physical phenomena cause occlusion of the scene – well studied in the literature. For instance, many models for raindrops are available, exploiting surface modeling and ray tracing (Roser and Geiger, 2009; Roser et al., 2010; Hao et al., 2019). In (You et al., 2015), raindrop motion dynamics are also modeled. Recent works instead focus on photorealism relaxing physical accuracy constraints (Porav et al., 2019; Alletto et al., 2019). A general model for lens occluders has been proposed in (Gu et al., 2009). Logically, it is extremely challenging to entirely simulate the appearance of scene encompassing multiple physical phenomena (for rain: rain streaks, raindrops on the lens, reflections, etc.), hence in (Tremblay et al., 2020; Mușat et al., 2021) they also combine i2i networks and physics-based rendering. In (Lengyel et al., 2021), they propose to exploit night physics characteristics to perform domain adaptation. However, this is quite different from our objective since they assume to physically model features not present in the target images. To the best of our knowledge, there is no method which unifies rendering based on physical models and i2i translations in a complementary manner. Physical models could also influence many training aspects, in the form of output space conditions (Reichstein et al., 2019), loss functions (Karpatne et al., 2017) or ad-hoc data augmentation (Xie et al., 2018), but we were the first to use physical models to reproduce better the appearance of a target domain in image translation.

Disentangled representations. Disentanglement is commonly used to gain control on generation by separating image content and style (Huang et al., 2018b; Lee et al., 2020; Jiang et al., 2020; Park et al., 2020). Others aim at controlling output images granularity (Singh et al., 2019) or specific features, as blur (Lu et al., 2019) or view-points (Nguyen-Phuoc et al., 2019). Some exploit disentanglement for few-shot generalization capabilities (Liu et al., 2019b; Saito et al., 2020). Domain features disentanglement also unifies representations across domains (Xia et al., 2020; Lin et al., 2019). To the best of our knowledge, we were the first to propose a disentanglement strategy for visual traits using physical models.

7.3 Physical model-guided disentanglement

Standard i2i GANs solely rely on context mapping between source and target only – which would be impractical relying only on physical rendering. In some setups, however, the target domain encompasses some visual traits, for

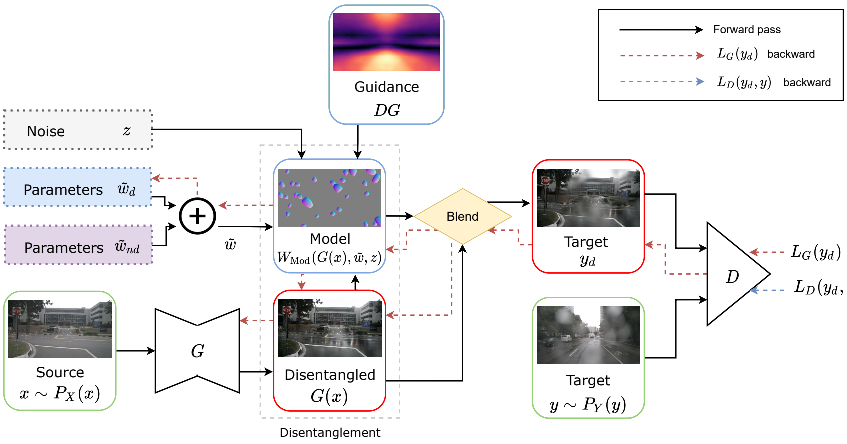


Figure 7.2: Model-guided disentanglement. Our unsupervised disentanglement process consists of applying a physical model $W_{\text{Mod}}(.)$ to the generated image $G(x)$. Subsequently, the composite image is forwarded to the discriminator and the GAN loss ($L_G(y_D)$ or $L_D(y_d, y)$) is backpropagated (dashed arrows). The model rendering depends on the estimated parameters $\tilde{w}$, composed by differentiable ($\tilde{w}_d$) and non-differentiable ones ($\tilde{w}_{nd}$). We use a Disentanglement Guidance (DG) to avoid interfering with the gradient propagation in the learning process. Green stands for real data, red for fake ones.

example adverse weather or lens occlusions, which modeling is well understood from physics. Hence, it may be amenable to integrate *a priori* physics knowledge in the adversarial learning process.

To formalize i2i transformations as a composition of physics and learned characteristics, we propose a setting shown in Fig. 7.2 where the GAN learns to disentangle the physically modeled traits from target (Sec. 7.3.1). Disentanglement is achieved relying on physical model-guided strategies (Sec. 7.3.2), where we exploit as the only prior the nature of the physical trait we aim to disentangle (*e.g.* raindrop, dirt, fog, etc.). Because these may have infinite variations of appearances, we estimate differentiable (Sec. 7.3.3) and non-differentiable (Sec. 7.3.4) target parameters of the physical model which ease disentanglement by reducing differences with target. Our approach boosts image quality and realism guiding model injection during training with gradient-based guidance (Sec. 7.3.5). An extensive explanation of training strategies is in Sec. 7.5.

7.3.1 Adversarial disentanglement

In image-to-image translation we aim to learn a transformation between a source X and a target Y, thus mapping $X \mapsto Y$ in an unsupervised manner. We assume that Y appearance is partly characterized by a well-identified phenomenon such as occlusions on the lens (*e.g.* rain, dirt) or weather phenomena (*e.g.* fog). Hence, we propose a sub-domain decomposition (as in Chapters 3 and 4) of $Y = \{Y_W, Y_T\}$, separating the identified traits (Y_W) from the other ones (Y_T). We assume this only on target, so $X = \{X_T\}$. In adversarial learning, the task of the generator is to approximate the probability distributions P_X and P_Y associated with the problem domains, such as

$$\begin{aligned} \forall x \in X, x \sim P_X(x), \\ \forall y \in Y, y \sim P_Y(y). \end{aligned} \tag{7.1}$$

For explaining the intuition, we assume that the traits identifiable in this manner are independent from the recorded scene. For instance, physical properties of raindrops on a lens (such as thickness or position) do not change with the scene, as it happens also with fog, where visual effects are only depth-dependent. Therefore, Y_W is fairly independent from Y_T, hence we formalize P_Y as a joint probability distribution with independent marginals, such as

$$P_Y(y) = P_{Y_W,Y_T}(y_W, y_T) = P_{Y_W}(y_W)P_{Y_T}(y_T). \tag{7.2}$$

Intuitively, approximating one of the marginals with *a priori* knowledge will force the GAN to learn the other one in a disentangled manner. During training, this translates into injecting features belonging to Y_W before forwarding the images to the discriminator, which will provide feedback on the general realism of the image.

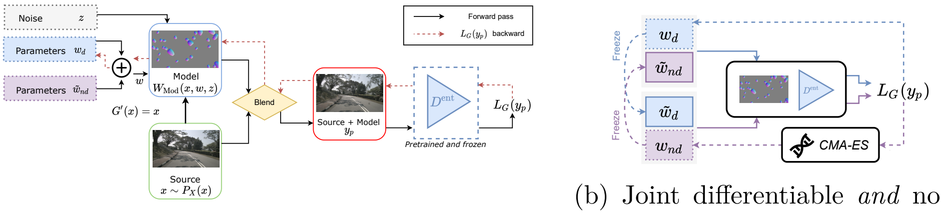
Formally, we modify a LSGAN (Mao et al., 2017a) training, which enforces adversarial learning minimizing

$$\begin{aligned} y_{\text{d}} &= G(x), \\ L_{\text{gen}} = L_G(y_d) &= \mathbb{E}_{x\ \sim P_X(x)}[(D(y_d) - 1)^2], \\ L_{\text{disc}} = L_D(y_d, y) &= \mathbb{E}_{x\ \sim P_X(x)}[D(y_d)^2] + \\ &\quad + \mathbb{E}_{y\ \sim P_Y(y)}[(D(y) - 1)^2], \end{aligned} \tag{7.3}$$

where L_{gen} and L_{disc} are tasks of generator G and discriminator D, respectively. We instead learn a disentangled mapping injecting physically modeled traits $W_{\text{Mod}}(.)$ on translated images. We newly define y_{d} as the disentangled composition of translated scene $G(x)$ and $W_{\text{Mod}}(.)$, hence

$$y_{\text{d}} = \alpha_w G(x) + (1 - \alpha_w) W_{\text{Mod}}(.)\,. \tag{7.4}$$

We define as α_w a pixel-wise measure of blending between modeled and learned scene traits. Pixels which depend only on $W_{\text{Mod}}(.)$ (as opaque occlusions) will show $\alpha_w = 1$ while others (*e.g.* transparent ones) will have $\alpha_w < 1$.



(a) Differentiable parameters.

(b) Joint differentiable *and* non-differentiable strategy.

Figure 7.3: Model-guided parameters estimation. a) We exploit a pretrained discriminator D^{ent}, to calculate an adversarial loss L_G on *source* data augmented with the model W_{Mod} having differentiable parameters w_d. In this process, the gradient flows only in direction of the differentiable parameters. b) We optimize until convergence differentiable (blue) and non-differentiable (purple) parameters, alternatively reaching new minima ($\tilde{w}_d$ and $\tilde{w}_{nd}$) used during optimization of the other parameter set. While differentiable parameters are regressed (Sec. 7.3.3), non-differentiable ones require black-box genetic optimization (Sec. 7.3.4), here CMA-ES (Hansen et al., 2003).

7.3.2 Physics models as guidance

One can easily obtain physical model (*i.e.* W_{Mod}) from existing literature – typically to render visual traits like drops, fog, or else. Injecting such physical models in our guided-GAN enables disentanglement and learning of visual traits *not* rendered by physical models, like wet materials for rain models (Halder et al., 2019), clouds in the sky for fog models (Sakaridis et al., 2018), etc.

However, these models often have extremely variable appearance depending on their physical parameters w so we propose adversarial-based strategies to regress optimal $\tilde{w}$ mimicking the target dataset appearance. This is in fact needed for disentangled training where we assume modeled traits to resemble target ones. Other parameters are of stochastic nature (*e.g.* drop positions on the image) and are encoded as noise z regulating random characteristics. Additionally, some models appearance – like refractive occlusions – vary with the underlying scene[1] s, so we write $W_{\text{Mod}}(.) = W_{\text{Mod}}(s, w, z)$, with $s = G(x)$. Following our pipeline in Fig. 7.2, if $\tilde{w}$ properly estimates *target* physical parameters, $W_{\text{Mod}}(s, \tilde{w}, z)$ estimates marginal $P_{Y_W}(y_W)$ which again enables disentanglement.

During inference instead, w and z can be arbitrarily varied, greatly increasing generation variability while still obtaining a realistic target scene rendering. In the following, we describe our adversarial parameter estimation strategy, while distinguishing differentiable (w_d) and non-differentiable (w_{nd}) parameters, such that $w = \{w_d, w_{nd}\}$.

[1] In Sec. 7.6, we explain how W_{Mod} depending of s is not violating the independence assumption of Eq. 7.2, and evaluate its effect in Sec. 7.5.4.2.

7.3.3 Differentiable parameters estimation

To estimate the target optimized derivable parameters $\tilde{w}_d$, we exploit an adversarial-based strategy benefiting from entanglement in naive trainings. We consider a naive baseline trained on source $\mapsto$ target mapping, where target entangles two sub-domains as specified in Sec. 7.3.1. We refer to generator and discriminator trained in this way as *entangled* generator and discriminator, respectively. The entangled discriminator D^{ent} successfully learns to distinguish fake target images. This results in being able to discriminate $P_X = P_{X_S}$ from $P_Y = P_{Y_T}(y_S)P_{Y_W}(y_W)$. Considering a simplified scenario where P_{Y_T} is arbitrarily confused with the source domain, such that $P_{Y_T} = P_{X_T}$, regressing w_d is the only way to minimize the domain shift. In other words, considering the derivable model parametrized by w_d, the above domains confusion prevents any changes in the scene. To minimize differences between source and target the network is left with updating the injected physical model appearance, ultimately regressing w_d. Fig. 7.3a shows our differentiable parameter pipeline. From a training perspective, we first pretrain an i2i baseline (*e.g.* MUNIT (Huang et al., 2018b)), learning a $X \mapsto Y$ mapping with an entangled generator G^{ent} and discriminator D^{ent}. We then freeze D^{ent} and use it to solve

$$y_p = \alpha_w x + (1 - \alpha_w) W_{\text{Mod}}(x, w, z), \min_{w_d} L_G(y_p)\,, \tag{7.5}$$

backpropagating the GAN loss through the differentiable model. Since many models may encompass pixelwise transparency, often the blending mask α_w is $\alpha_w = \alpha_w(w, z)$. Please not this is *not* a traditional adversarial training, since freezing the discriminator is mandatory to preserve the previously learned target domain appearance during the estimation process. After convergence, we extract the optimal parameter set $\tilde{w}_d$. Alternatively, $\tilde{w}_d$ could be manually tuned by an operator, at the cost of menial work and inaccuracy, possibly leading to errors in the disentanglement.

From Fig. 7.3a, notice that the gradient flows only through differentiable parameters (w_d). We now detail our strategy to optimize jointly inevitable non-differentiable parameters (w_{nd}).

7.3.4 Non-differentiable parameters estimation

The previously described strategy only holds for differentiable parameters w_d, since we use backpropagation of an adversarial loss. Nonetheless, many models include non-differentiable parameters w_{nd} that could equally impact the realism of our model $W_{\text{Mod}}(.)$. For example, a model generating raindrops occlusion would include differentiable parameters like the imaging focus, but also non-differentiable ones like the shape or number of drops – all of which significantly impact visual appearance. Incorrect sizing of non-differentiable parameters w_{nd} can lead to a wrong disentanglement. Manual approximation of optimal w_{nd} parameters via trial-and-error might also be cumbersome or impractical for vast search space. To circumvent this, we exploit a genetic strategy estimating w_{nd}.

In our method, non-differentiable parameters are fed to a genetic optimization strategy. The evolutionary criteria remain the same as for differentiable parameters, that is the pretrained discriminator (D^{ent}) adversarial loss. In practice, to avoid noisy updates after genetic estimation, we average adversarial loss over a fixed number of samples to reliably select a new population. After convergence, we extract the optimal parameter set $\tilde{w}_{nd}$. In our experiments, we use CMA-ES (Hansen et al., 2003) as evolutionary strategy, but the proposed pipeline is extensible to any other genetic algorithm.

7.3.5 Disentanglement guidance

It is worth noting that too sparse injection of model $W_{\text{Mod}}(.)$ negatively impacts disentanglement because the guided-GAN will entangle similar physical traits to fool the discriminator, while injecting too much of $W_{\text{Mod}}(.)$ will prevent the discovery of the disentangled target. Spatially, we observe that regions that do not differ from source to target are most frequently impacted by entanglement. This is because the discriminator naturally provides less reliable predictions due to the local source-target similarities, which leads the generator to produce artifacts resembling target physical characteristics to fool the discriminator, eventually leading to unwanted entanglement. In rainy scenes this happens for trees or buildings, which appearance little vary if dry or wet, whereas ground or road exhibit puddles which are strong rainy cues.

To balance the injection of $W_{\text{Mod}}(.)$, we guide disentanglement by injecting $W_{\text{Mod}}(.)$ only on low domains shift areas, pushing the guided-GAN to learn the disentangled mapping of the scene. Specifically, we learn a Disentanglement Guidance (DG) dataset-wise by averaging the GradCAM (Selvaraju et al., 2017) feedback on the source dataset, relying on the discriminator D^{ent} gradient on *fake* classification. Areas with high domain shift will be easily identified as *fake*, while others will impact less on the prediction. To take into account different resolutions, we evaluate GradCAM for all the discriminator layers. Formally, we use LSGAN to obtain

$$\text{DG} = \mathbb{E}_{x\ \sim P_X(x)}[\mathbb{E}_{l \in L}[\text{GradCAM}_l(D^{\text{ent}}(x))]]\,, \tag{7.6}$$

with L being the discriminator layers. At training, we inject models only on pixels (u, v) where $\text{DG}_{u,v} < \gamma$, where $\gamma \in [0, 1]$ is a hyperparameter. In Sec. 7.5.4.4 we visually assess the effect of DG.

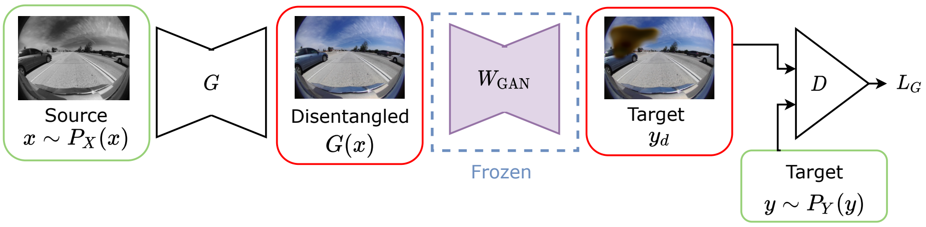


Figure 7.4: Neural-guided disentanglement. We exploit here a separate frozen GAN (W_{GAN}) which renders specific target traits (here, dirt) on generator G output images before forwarding them to the discriminator D. We do not show gradient propagation for simplicity.

7.3.6 Training strategy

For models having differentiable *and* non-differentiable parameters we employ a joint optimization shown in Fig. 7.3b. We first initialize a set of parameters w, then alternatively use our strategy for differentiable parameters estimation w_d (Sec. 7.3.3) and the genetic strategy for non differentiable ones w_{nd} (Sec. 7.3.4). Notice that the alternation of optimized parameters prevents divergence due to simultaneous optimization. We apply updates until optimum, reaching the two sets of target style parameters, $\tilde{w} = \{\tilde{w}_d, \tilde{w}_{nd}\}$.

The complete training strategy for model-guided disentanglement is in Sec. 7.5.1.1.

7.4 Neural-guided disentanglement

For some visual traits, a physical model may not be immediately available so we consider also the case in which the guidance is provided by a neural model, learned separately. Referring to our adversarial strategy in Sec. 7.3.1, we simply substitute W_{Mod} with W_{GAN} in Eq. 7.4, where W_{GAN} is our neural guidance – a GAN in our experiments.

Following our past explanations, assuming W_{GAN} generates specific visual traits – may it be dirt, drop, watermark or else – it is an approximation of the marginal $P_{Y_W}(y_W)$. We define $\tilde{\theta}$ as the optimal set of parameters of the network to reproduce target occlusion appearance. Subsequently, processing generated images with W_{GAN} before forwarding them to the discriminator pushes the guided-GAN we aim to train in a disentangled manner (not to be confused with W_{GAN}) to achieve disentanglement, as illustrated in Fig. 7.4, following the same reasoning as in Sec. 7.3.1.

Of importance here, even if W_{GAN} is trained supervisedly – for example, from annotated pairs of images / dirt – the disentanglement strategy is itself fully unsupervised. Also, referring to Eq. 7.2, the guided-GAN can only achieve disentanglement and estimate $P_{Y_T}(y_T)$ from images in Y, if W_{GAN} ($i.e.$ $W(\cdot)$) correctly estimates $P_{Y_W}(y_W)$. Suppose W_{GAN} augments rain on images, it will be sensitive to the intensity as well as the appearance of drops of Y. In other words, it would be possible only to recreate target-like scenes. With the model-guided disentanglement strategy we could instead re-inject physical traits of arbitrary appearance, greatly increasing the generative capabilities of our guided framework.

The complete training strategy for neural-guided disentanglement is also in Sec. 7.5.1.1.

7.5 Experiments

We evaluate our disentanglement strategies on the real datasets nuScenes (Caesar et al., 2020), RobotCar (Porav et al., 2019), Cityscapes (Cordts et al., 2016) and WoodScape (Yogamani et al., 2019), and on the synthetic Synthia (Ros et al., 2016) and Weather Cityscapes (Halder et al., 2019). Our evaluation methodology is in Sec. 7.5.1 including training, tasks, user study, and model/neural guidance.

In Sec. 7.5.2 we extensively study the disentanglement of raindrop, dirt, composite occlusions, and fog – on a qualitative/quantitative basis, and using proxy tasks and human judgement. Our method is compared against the DRIT (Lee et al., 2020), U-GAT-IT (Kim et al., 2020), AttentionGAN (Tang et al., 2019a), CycleGAN (Zhu et al., 2017a), and MUNIT (Huang et al., 2018b) frameworks. Opposite to the literature, our method enables disentanglement of the target domain, so we report both the disentangled translations as well as the translations with the injection of optimal target physical traits. The disentanglement is greatly visible in images presented in this section.

Because physical models are readily available, we emphasize our physical model-guided strategy (Sec. 7.3) evaluated on 4 models in Sec. 7.5.2.1. Conversely, the neural-guided strategy (Sec. 7.4), requires rare separate neural networks for rendering traits. It is subsequently only evaluated on dirt disentanglement in Sec. 7.5.2.2, relying on DirtyGAN (Uricar et al., 2021), for comparison purposes with the model-guided strategy.

In Sec. 7.5.3, we study the accuracy of our physical model parameters estimation on the well-documented raindrop model, and finally ablate our proposal in Sec. 7.5.4.

Formalism. We formalize disentangled trainings as $\mathcal{T}_{dis}$, guided either with a full physical model ($\mathcal{T}_{W_{\text{Mod}}}$), a model with only differentiable param-

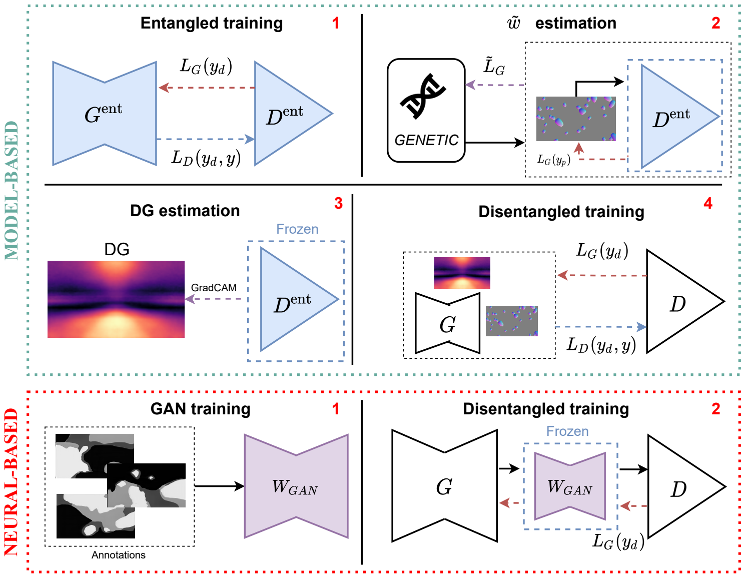


Figure 7.5: Training pipelines. For *model-guided* disentanglement, we 1) train a naive i2i entangled baseline, 2) use the entangled discriminator feedback to estimate optimal parameters $\tilde{w}$ and 3) Disentanglement Guidance (DG), and finally 4) train the guided-GAN with model injection. For *neural-guided* disentanglement, we 1) train a GAN (W_{GAN}) exploiting additional knowledge as semantics and 2) use it to inject target traits during our guided-GAN training.

eters ($\mathcal{T}_{W^{w_d}_{\text{Mod}}}$), or neural-guided ($\mathcal{T}_{W_{\text{GAN}}}$). When re-injecting physical traits, we also show their parameters in parentheses. For example, $\mathcal{T}_{W_{\text{Mod}}}(\tilde{w})$ means model-guided disentangled output with injection of the full model estimated on target ($\tilde{w}$).

7.5.1 Methodology

7.5.1.1 Training

Our disentangled GAN is architecture agnostic. Here, we rely on the MUNIT (Huang et al., 2018b) backbone for its multi-modal capabilities, and exploit LSGAN (Mao et al., 2017a) for training. Fig. 7.5 shows our two training pipelines.

For **model-guided** training (Fig. 7.5, top), we leverage on a multi-step pipeline, only assuming the known nature of features to disentangle (*e.g.* raindrop, dirt, fog, etc.). First, an i2i source$\mapsto$ target baseline is trained in an entangled manner, obtaining entangled discriminator (D^{ent}). Second, we make use of D^{ent} to regress the optimal parameters $\tilde{w}$ with adversarial (Sec. 7.3.3) and genetic (Sec. 7.3.4) estimation. Third, we extract Dis-

	Task	**Entanglement**	**Datasets**	**Guidance**		
				Model	w_d	w_{nd}
Model	clear ↦ rain$_{\text{drop}}$	Raindrop	nuScenes	Raindrop	σ	t,(s,p)x4
	gray ↦ color$_{\text{dirt}}$	Dirt	WoodScape	Dirt	σ, α	-
	synth ↦ WCS$_{\text{fog}}$	Fog	Synthia,Weather CS	Fog	β	-
	clear ↦ snow$_{\text{cmp}}$	Composite	Synthia	Composite	-	-
Neural				*Network*		
	gray ↦ color$_{\text{dirt}}$	Dirt	WoodScape	DirtyGAN		

Table 7.1: Disentanglement tasks. For each task, we indicate the features entangled in the target domain (also, shorten as indices of task name), the datasets, and the model or neural guidance employed for disentanglement.

entanglement Guidance (Sec. 7.3.5), also using D^{ent}. Finally, we train from scratch the disentangled guided-GAN (Sec. 7.3).

For **neural-guided** training (Fig. 7.5, bottom), we use a prior-agnostic two-step pipeline. First, we train the third-party W_{GAN} to render occlusions, exploiting semantic supervision in our experiments though it could realistically be replaced with self-supervision. Then, we train our disentangled guided-GAN *without* any supervision.

7.5.1.2 Tasks

Tab. 7.1 lists the tasks evaluated and ad-hoc datasets. When referring to a task, we denote as indices the entangled features in target domain. Thus, clear ↦ rain$_{\text{drop}}$ literally means 'translation from *clear* to *rain* with entangled *drops*'. We later describe models used for disentanglement.

clear↦rain$_{\text{drop}}$. We exploit nuScenes (Caesar et al., 2020), which includes urban driving scenes, and use metadata to build clear/rain splits obtaining 114251/29463 training and 25798/5637 testing clear/rain images. Target rain images entangle highly unfocused drops on the windshield, which would hardly be annotated as seen in Fig. 7.6, first row.

gray↦color$_{\text{dirt}}$. Here, we rely on the fish-eye WoodScape (Yogamani et al., 2019) dataset which has some images with soiling on the lens. We separate the dataset in clean/dirty images using soiling metadata getting 5117/4873 training images and 500/500 for validation. Because clean/dirty splits do not encompass other domain shifts, we additionally transform *clean* images to *gray*. Subsequently, we frame this as a colorization task where target *color* domain entangles *dirt*. For disentanglement, we experiment using both a physical model-guided and a neural-guided strategy.

clear↦snow$_{\mathbf{cmp}}$. With Synthia (Ros et al., 2016) we also investigate entanglement of very different alpha-blended composites, like "Confidential" watermarks or fences. We split Synthia using metadata into clear/snow images and further augment snow target with said composite at random position. As clear/fog splits, we use 3634/3739 images for training and 901/947 for validation. To guide disentanglement, we consider a composite model, inspiring from the concept of thin occluders (Garg and Nayar, 2006).

synth↦WCS$_{\mathbf{fog}}$. We learn here the mapping from synthetic Synthia (Ros et al., 2016) to the foggy version of Weather CityScapes (Halder et al., 2019) – a foggy-augmented Cityscapes (Cordts et al., 2016). The goal is to learn the synthetic to real mapping, while disentangling the complex fog effect in target. For training we use 3634/11900 and 901/2000 for validation as Synthia/WeatherCityscapes. We use a fog model to guide our network.
Note that this task differentiates from others, since target has fog of heterogeneous intensities (max. visibility 750, 375, 150 and 75m) making disentanglement significantly harder.

7.5.1.3 Physical model guidance

To correctly fool the discriminator, it is crucial to choose a model that realistically resembles the entangled feature. We leverage 4 physical models, listed in Tab. 7.1 'Model' with their differentiable (w_d) and non-differentiable (w_{nd}) parameters.

Raindrop model. We extend the model of Alletto *et al.* (Alletto et al., 2019), which is balanced between complexity and realism. Drops are approximated by simple trigonometric functions, while we encompass also noise addition for shape variability (sha, 2017). For drops photometry, we use fixed displacement maps (U, V) for coordinate mapping on both x and y axes, technically encoded as 3-channels images (Alletto et al., 2019). To approximate light refraction, a drop at (u, v) has its pixel (u_i, v_i) mapped to

$$(u + \mathrm{U}(u_i, v_i) \cdot \rho, v + \mathrm{V}(u_i, v_i) \cdot \rho) \,, \tag{7.7}$$

where ρ is a drop-wise value representing water thickness. Most importantly, we also model imaging focus, since it may extremely impact the rendered raindrop appearance (Halimeh and Roser, 2009; Cord and Aubert, 2011; Alletto et al., 2019). Hence, we use a Gaussian point spread function (Pentland, 1987) to blur synthetic raindrops. We implement kernel variance σ as differentiable, while drops size (s), frequency (p), and shape (t) related parameters are non differentiable. We use a single shape parameter and generate 4 types of drops, with associated p and t.

Dirt model. Here, we naively extend our raindrop model removing displacement maps as soil has no refractive behaviors. Instead, we introduce a color guidance that forces synthetic dirt to be brighter in peripherals regions, also depending on a parameter α which regulates occlusion maximum opacity (hence, maximum α_w value). We also estimate σ as aforementioned.

Composite occlusions model. We exploit the model of thin occluder proposed in (Garg and Nayar, 2006) to render composite occlusions on images, *i.e.* randomly translated alpha-blended transparent images such as watermarks or fence-like grids. We assume to fully know transparency, thus no parameter is learned.

Fog model. We leverage the physics model of (Halder et al., 2019) using an input depth map. Fog thickness is regulated by a differentiable extinction coefficient β which regulates maximum visibility.

7.5.1.4 Neural guidance

Finding appropriate neural networks to render visual traits is not trivial. Here we experiment only with Dirt, as listed in Tab. 7.1 'Neural'.

Dirt neural. DirtyGAN (Uricar et al., 2021) is a GAN-based framework for opaque soiling occlusion generation. It is composed by two components, *i.e.* a VAE for occlusion map generation (trained using soiling semantic maps) and an i2i network conditioned on the generated map to include synthetic soiling on images. To train DirtyGAN, we first train a VAE to learn the shape of soiling, and then proceed to train a modified CycleGAN (Zhu et al., 2017a) to generate realistic soiling, conditioning the soiling shape on the VAE outputs. For more details on this we refer to (Uricar et al., 2021).

7.5.1.5 User study

We also conducted a qualitative anonymous online study collecting answers from 56 users (22 males, 33 females, 1 non-binary) from 21 to 65 years old (mean 27.9, std. 7.6). Each user had to evaluate 85 randomized scenes with a Likert-5 scale, providing the image looks realistic and efficiently disentangled. For ease of reading we included the results in each ad-hoc subsections (Secs. 7.5.2.1,7.5.3).

7.5.2 Disentanglement

In this section, we evaluate our disentanglement strategy both using physical model-guidance or neural-guidance.

7.5.2.1 Physical model guided-disentanglement

On the 4 tasks and 4 ad-hoc models in Tab. 7.1 'Model', we evaluate the ability to disentangle visual traits with physical model guidance from Sec. 7.3, reporting qualitative, quantitative and human judgment.

Hereafter, we separate experiments on Raindrop, Dirt and Composite disentanglement from the Fog experiments, since only the former have homogeneous physical parameters (w) throughout the dataset[2]. Since non-differentiable parameters were fairly easy to manually tune, we thoroughly experiment in the differentiable-only $\{w_d\}$ setup and compare it later on to our full $\{w_d, w_{nd}\}$ estimation (Sec. 7.5.3).

Qualitative disentanglement. We present outputs for clear ↦ rain$_{\text{drop}}$ trained on nuScenes (Caesar et al., 2020), comparing to state-of-the-art methods at the time of submission (Lee et al., 2020; Kim et al., 2020; Tang et al., 2019a; Zhu et al., 2017a; Huang et al., 2018b) (Fig. 7.6) and for gray ↦ color$_{\text{dirt}}$ and clear ↦ snow$_{\text{cmp}}$ with respect to the backbone (Figs. 7.7,7.8, respectively). In all cases, baselines entangle occlusions in different manners. For instance, in Fig. 7.6 it is noticeable the constant position of rendered raindrops between different frameworks, as in the 1st column on the leftmost tree, which is a visible effect of entanglement and limits image variability. Also, occlusion entanglement could cause very unrealistic outputs where the structural consistency of either the scene (Fig. 7.7) or the occlusion (Fig. 7.8) is completely lost.

Referring to Figs. 7.6,7.7,7.8, our method is always able to produce high quality images *without* occlusions ('Disentangled' rows) including typical target domain traits such as wet appearance without drops, colored image without dirt or snowy image without occlusions, respectively. Furthermore, we can inject occlusions with optimal estimated parameters ('Target-style' rows) to mimic target appearance which enables a fair comparison with baselines[3].

We also inject raindrops with arbitrary parameters to simulate *unseen* dashcam-style images in Fig. 7.6 (last 2 rows). The realistic results demonstrate both the quality of our disentanglement and the realism of the *Raindrop* model.

[2] For *Raindrop*, *Dirt* and *Composite* we consider w_d and w_{nd} to be dataset-wise constant. E.g. all raindrops have the same defocus blur, transparency, etc. Conversely, *Fog* images have varying fog intensity.

[3] For comparing with neural methods we set $\alpha = 1$.

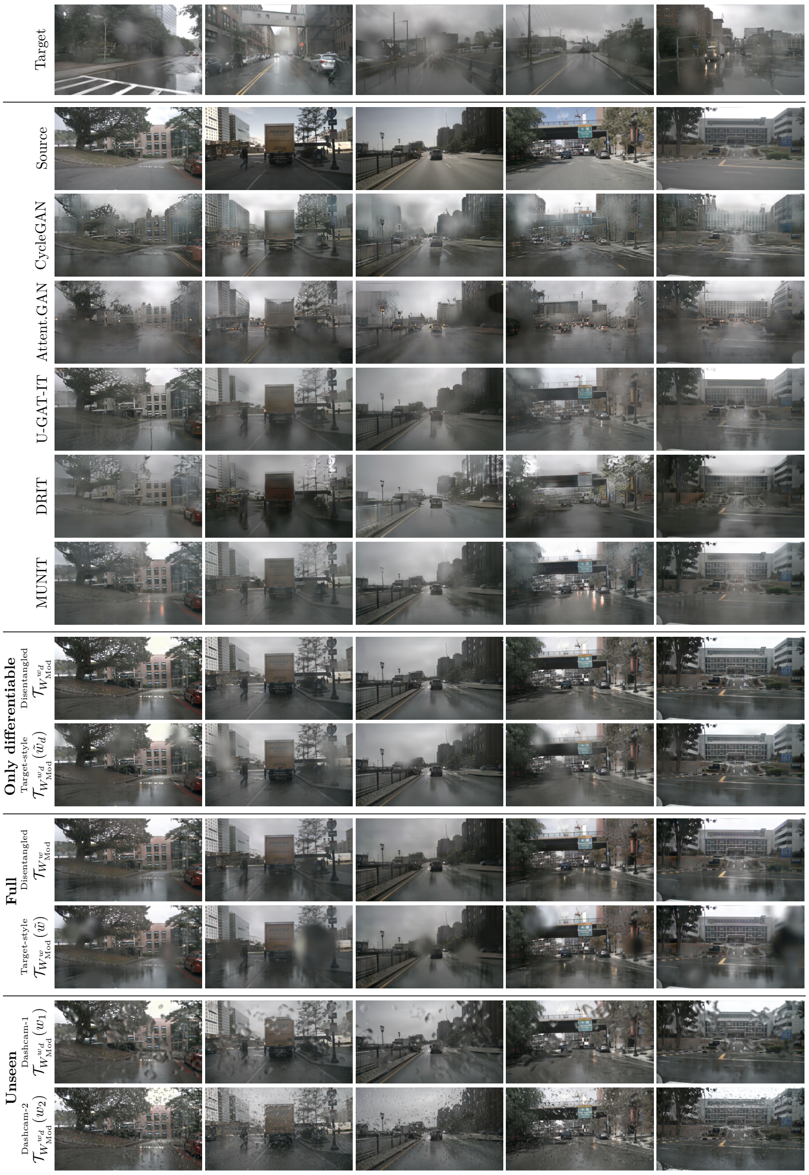

Target
Source
CycleGAN
Attent.GAN
U-GAT-IT
DRIT
MUNIT
Only differentiable
Disentangled $\mathcal{T}_{W^{w_d}_{\text{Mod}}}$
Target-style $\mathcal{T}_{W^{w_d}_{\text{Mod}}}(\tilde{w}_d)$
Full
Disentangled $\mathcal{T}_{W^{w}_{\text{Mod}}}$
Target-style $\mathcal{T}_{W^{w}_{\text{Mod}}}(\tilde{w})$
Unseen
Dashcam-1 $\mathcal{T}_{W^{w_d}_{\text{Mod}}}(w_1)$
Dashcam-2 $\mathcal{T}_{W^{w_d}_{\text{Mod}}}(w_2)$

Figure 7.6: Raindrop disentanglement on clear ↦ $\text{rain}_{\text{drop}}$. We compare qualitatively with the state-of-the-art on the clear ↦ $\text{rain}_{\text{drop}}$ task with rain drops model-guided disentanglement. In the first row, we report samples of the target domain. Subsequently, the *Source* image (2nd row), the translations by different baselines (rows 3-7) and our results (rows 8-13). Our model-guided network is able to disentangle the generation of peculiar rainy characteristics from the drops on the windshield ('Disentangled' rows) and re-injection with estimated parameters ('Target-style'). We evaluate both the differentiable-only parameter estimation (rows 8-9) and the genetic-based full estimation (rows 10-11). We also show injection of other arbitrary parameters w_1, w_2 (last 2 rows).

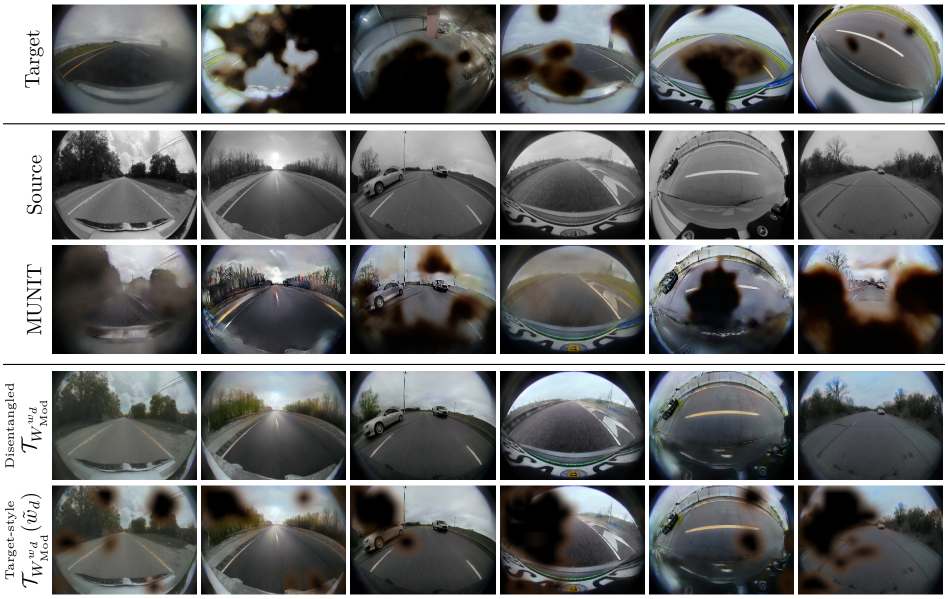

Target
Source
MUNIT
Disentangled $\mathcal{T}_{W^{w_d}_{\text{Mod}}}$
Target-style $\mathcal{T}_{W^{w_d}_{\text{Mod}}}(\tilde{w}_d)$

Figure 7.7: Dirt disentanglement on gray $\mapsto$ color$_{\text{dirt}}$. We compare with MUNIT (Huang et al., 2018b) for the gray $\mapsto$ color$_{\text{dirt}}$ task. Although MUNIT successfully mimics the *Target* style (rows 1,3), our approach lead to a more realistic image colorization disentangling the presence of dirt ('Disentangled' row $\mathcal{T}_{W_{\text{Mod}}}$) We also use the dirt model to reproduce Target images ('Target-style' row $\mathcal{T}_{W^{w_d}_{\text{Mod}}}(\tilde{w})$).

Quantitative disentanglement. We use GAN metrics to quantify the quality of the learned mappings. Results are reported in Tab. 7.2a, where Inception Score (IS) (Salimans et al., 2016) evaluates quality and diversity against target, LPIPS distance (Zhang et al., 2018a) evaluates translation diversity (thus avoiding mode-collapse), and Conditional Inception Score (CIS) (Huang et al., 2018b) single-image translations diversity for multimodal baselines. In practice, IS is computed over all the validation set while CIS is estimated on 100 different translations of 100 random images following (Huang et al., 2018b). The InceptionV3 network for Inception Scores was finetuned on the source/target classification as in (Huang et al., 2018b). LPIPS distance is calculated on 1900 random pairs of 100 translations as in (Huang et al., 2018b). For fairness, we only compare 'Target-style' outputs to baselines, since those are not supposed to disentangle physical traits, and can only output images resembling *Target*.

Tab. 7.2a shows we outperform all baselines on IS/CIS, including MUNIT – our i2i backbone. This is due to disentanglement, since entanglement phenomena limit occlusions appearance and position variability. Even the scene translation quality is improved by disentanglement since the generator learns a simpler target domain mapping without any occlusions. As regards

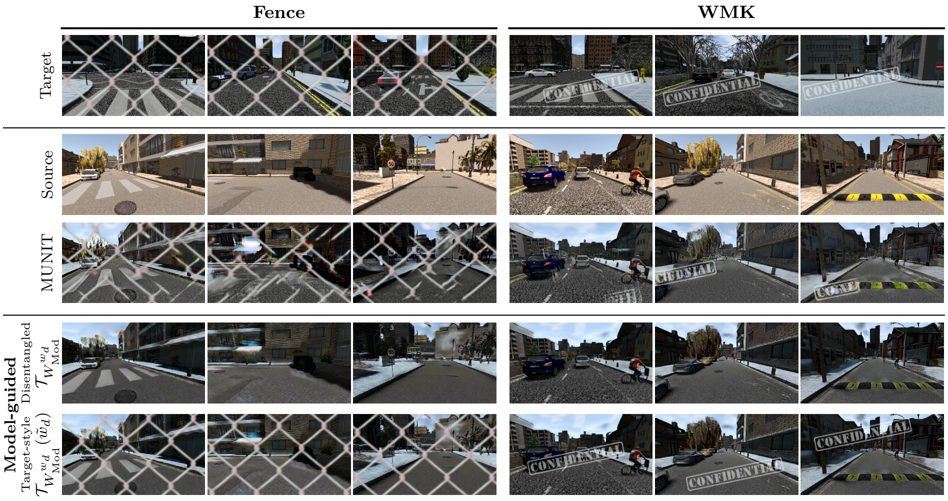

Fence
WMK
Target
Source
MUNIT
Model-guided
Disentangled $\mathcal{T}_{W^{w_d}_{\text{Mod}}}$
Target-style $\mathcal{T}_{W^{w_d}_{\text{Mod}}}(\tilde{w}_d)$

Figure 7.8: Composite disentanglement on clear ↦ snow$_{\text{cmp}}$. We extend the applicability of our method to composite occlusions, that we validate in the clear ↦ snow$_{\text{cmp}}$ scenario. We add a fence-like occlusion (left) and a *confidential* watermark (right) to *synthetic_snow*, with random position. As expected, we encounter entanglement phenomena for MUNIT, while our model-guided network is successful in learning the disentangled appearance ('Disentangled' row $\mathcal{T}_{W^{w_d}_{\text{Mod}}}$). In our 'Target-style' row $\mathcal{T}_{W^{w_d}_{\text{Mod}}}(\tilde{w}_d)$, we inject the occlusions to mimic the target style.

LPIPS distance, we outperform the baseline on raindrops while we rank lower on the other tasks. While IS/CIS quantify both quality and diversity, LPIPS metric is evaluating variability only thus penalizing simpler occlusion generation. For instance, our rendered dirt in Fig. 7.7 is often black while MUNIT-generated artifacts are highly variable (compare rows *MUNIT* and ours $\mathcal{T}_{W^{w_d}_{\text{Mod}}}(\tilde{w}_d)$). The same happens for watermarks in Fig. 7.8, where unrealistic artifacts are highly variable. For raindrops, instead, MUNIT tends to just blur images, while we benefit from the refractive capabilities of our physical model which increase LPIPS.

Semantic segmentation. To provide additional insights on the effectiveness of our framework and compensate for the well-known noisiness of GAN metrics (Zhang et al., 2018a), we quantify the usability of generated images for semantic segmentation in the clear ↦ rain$_{\text{drop}}$ setup. Therefore, we process the popular Cityscapes (Cordts et al., 2016) dataset for semantic segmentation with our best clear ↦ rain$_{\text{drop}}$ model-guided training, obtaining a synthetic rainy version $\mathcal{T}_{W^{w_d}_{\text{Mod}}}(\tilde{w}_d)$ that we use for finetuning PSPNet (Zhao et al., 2017), following Halder et al. (Halder et al., 2019). Please note that this also demonstrates the generation capabilities to new scenarios of our

Experiment	**Network**	**IS↑**	**LPIPS↑**	**CIS↑**
clear ↦ rain$_{\text{drop}}$	CycleGAN (Zhu et al., 2017a)	1.15	0.473	-
	AttentionGAN (Tang et al., 2019a)	1.41	0.464	-
	U-GAT-IT (Kim et al., 2020)	1.04	0.489	-
	DRIT (Lee et al., 2020)	1.19	0.492	1.12
	MUNIT (Huang et al., 2018b)	1.21	0.495	1.03
	Ours $\mathcal{T}_{W^{w}_{\text{Mod}}}(\tilde{w})$	1.25	0.502	1.08
	Ours $\mathcal{T}_{W^{w_d}_{\text{Mod}}}(\tilde{w}_d)$	**1.53**	**0.515**	**1.15**
gray ↦ color$_{\text{dirt}}$	MUNIT (Huang et al., 2018b)	1.06	0.656	1.08
	Ours $\mathcal{T}_{W^{w_d}_{\text{Mod}}}(\tilde{w}_d)$	1.25	0.590	1.15
clear ↦ snow$_{\text{cmp}}$ (fence)	MUNIT (Huang et al., 2018b)	1.26	**0.547**	1.11
	Ours $\mathcal{T}_{W^{w_d}_{\text{Mod}}}(\tilde{w}_d)$	**1.31**	0.539	**1.19**
clear ↦ snow$_{\text{cmp}}$ (WMK)	MUNIT (Huang et al., 2018b)	1.17	**0.567**	1.01
	Ours $\mathcal{T}_{W^{w_d}_{\text{Mod}}}(\tilde{w}_d)$	**1.19**	0.551	**1.02**
synth ↦ WCS$_{\text{fog}}$	CycleGAN (Zhu et al., 2017a)	1.31	0.384	-
	AttentionGAN (Tang et al., 2019a)	*	*	*
	U-GAT-IT (Kim et al., 2020)	1.05	0.406	-
	DRIT (Lee et al., 2020)	1.22	0.424	1.10
	MUNIT (Huang et al., 2018b)	1.22	**0.429**	1.13
	Ours $\mathcal{T}_{W^{w_d}_{\text{Mod}}}(\tilde{w}_d)$	**1.33**	0.420	**1.17**

* AttentionGAN converges to the identity transformation.

(a) GAN metrics.

Method	**AP↑**
Original (from (Halder et al., 2019))	18.7
Finetuned w/ Halder *et al.* (Halder et al., 2019)	25.6
Finetuned w/ Model-guided $\mathcal{T}_{W^{w_d}_{\text{Mod}}}(\tilde{w}_d)$	**27.7**

(b) Semantic segmentation on rain

Table 7.2: Image quality evaluation. In (a), we quantify GAN metrics for all tasks. While quality-aware metrics are always successfully increased, LPIPS depends on the visual complexity of the model and presence of artifacts. In (b), we compare our pipeline for finetuning semantic segmentation network outperforming the state-of-the-art for rain generation.

GAN, since we use the pretrained network on nuScenes given the absence of rainy scenes in Cityscapes. We report the mAP for the 25 rainy images with semantic labels provided by (Halder et al., 2019) in Tab. 7.2b. We experience a significant increase (+9%) with respect to baseline PSPNet trained on original clear images (*Original*), and also outperform (+2.1%) the finetuning with rain physics-based rendering (Halder et al., 2019). Both networks finetune *Original* weights. The overall low numbers reported are impacted by the significant domain shift between Cityscapes and nuScenes.

Disentanglement on heterogeneous datasets. We now evaluate the effectiveness of the synth ↦ WCS$_{\text{fog}}$ experiment which translates from synthetic Synthia to the real-augmented Weather CityScapes (Halder et al., 2019) entangling fog of various intensities (from light to thick fog). Notice this task significantly differs from others for two reasons. First, unlike other experiments the model parameter – the optical extinction coefficient, β –

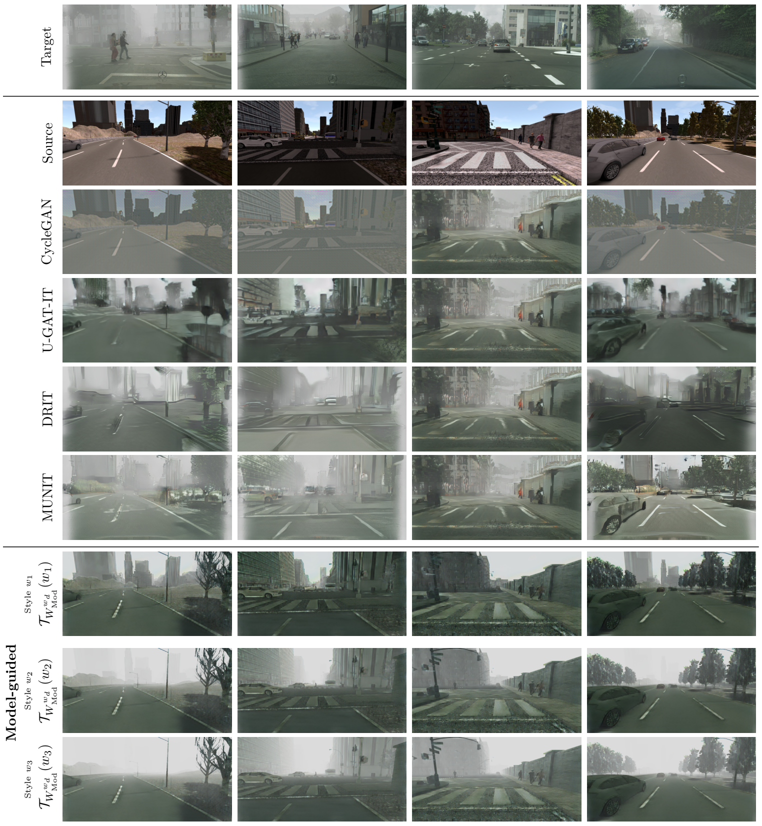
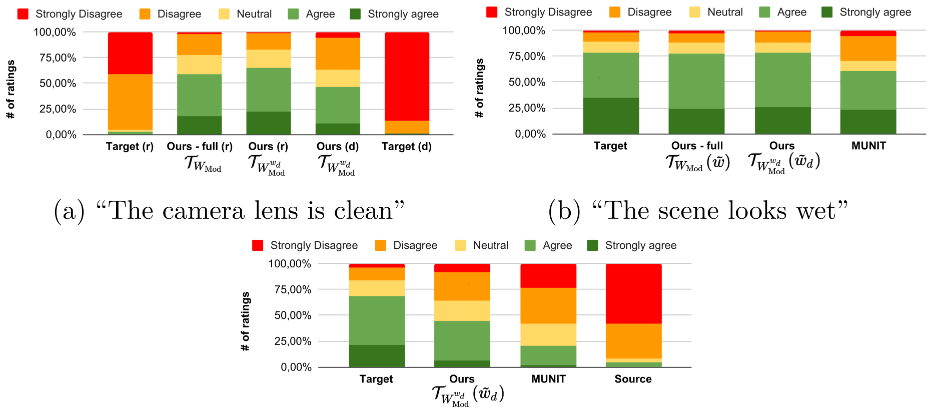

Target
Source
CycleGAN
U-GAT-IT
DRIT
MUNIT
Model-guided
Style w_1
$\mathcal{T}_{W^{w_d}_{\text{Mod}}}(w_1)$
Style w_2
$\mathcal{T}_{W^{w_d}_{\text{Mod}}}(w_2)$
Style w_3
$\mathcal{T}_{W^{w_d}_{\text{Mod}}}(w_3)$

Figure 7.9: synth ↦ WCS$_{\text{fog}}$ translations. As visible, MUNIT shows entanglement phenomena, leading to artifacts. Our model-guided disentanglement, instead, enables to generate a wide range of foggy images, with arbitrary visibility, while mantaining realism. Since the fog model W_{Mod} always blocks the gradient propagation in the sky region, the network can not achieve photorealistic disentanglement but still improves the generated image quality.

varies in the target dataset. Second, the fog model is depending on the scene geometry (Narasimhan and Nayar, 2002). This makes the disentanglement task non-trivial. In our adversarial disentanglement, we however still regress a single $\beta = 28.61$ somehow averaging the ground truth values ($\beta \in [4, 40]$).

In Fig. 7.9 results show we are able to generate images stylistically similar to target ones, but with geometrical consistency and varying β (last 3 rows). Instead, MUNIT (Huang et al., 2018b) fails to preserve realism due to entanglement artifacts, visible in particular on elements at far (as buildings in the background). Please note that we intentionally do not show disentangled output for fairness, since the physical model always blocks the gradient propagation in the sky. More details on this will be discussed in Sec. 7.6.



(a) “The camera lens is clean”

(b) “The scene looks wet”

(c) “The colors look natural”

Figure 7.10: Disentanglement user study. We asked 56 users (cf. Sec. 7.5.1.5) to judge the lens cleanness (a) on raindrops (r) and dirt (d), or the wetness (b) or coloring (c) of clear ↦ rain$_{\text{drop}}$ and gray ↦ color$_{\text{dirt}}$ generated scenes, respectively. Details are in the text. Our system greatly improves results following human evaluation metrics.

Randomizing $\beta \in [4, 40]$ we report GAN metrics results in Tab. 7.2a, where the increased quality of images is quantified. LPIPS distance suffers from the absence of artifacts in our model-guided $\mathcal{T}_{W^{w_d}_{\text{Mod}}}(\tilde{w}_d)$, which artificially increases image variability. The physical model always renders correctly regions at far (*e.g.* the sky, which is always occluded), hence pure variability quantified by LPIPS is reduced (cf. above LPIPS definition).

User study. To further evaluate our disentanglement quality, we asked 56 users to rate images (details in Sec. 7.5.1.5). First, we presented our disentangled outputs and real images *with* occlusions on the clear↦rain$_{\text{drop}}$ and gray↦color$_{\text{dirt}}$ tasks, where users were asked to rate for each image if ”The camera lens is clean (no dirt, no raindrops)”. Results in Fig. 7.10a show our strategy is better since the lens in our images is judge cleaner than target images. However, this does not assess if the underlying transformation (*i.e.* wetness or color) was properly learned.

Hence, secondly we compare translation realism with the MUNIT baseline, rating the statement “The scene looks wet” for clear↦rain$_{\text{drop}}$ and “The scene looks colorful” for gray↦color$_{\text{dirt}}$. We also include real source images (*i.e.* gray) in gray↦color$_{\text{dirt}}$ to evaluate performances in the naive identity transformation, and target images in both to set upper bounds. Results in Figs. 7.10b , 7.10c clearly show the superiority of our approach with respect to the MUNIT, heavily reducing the gap with real target images.

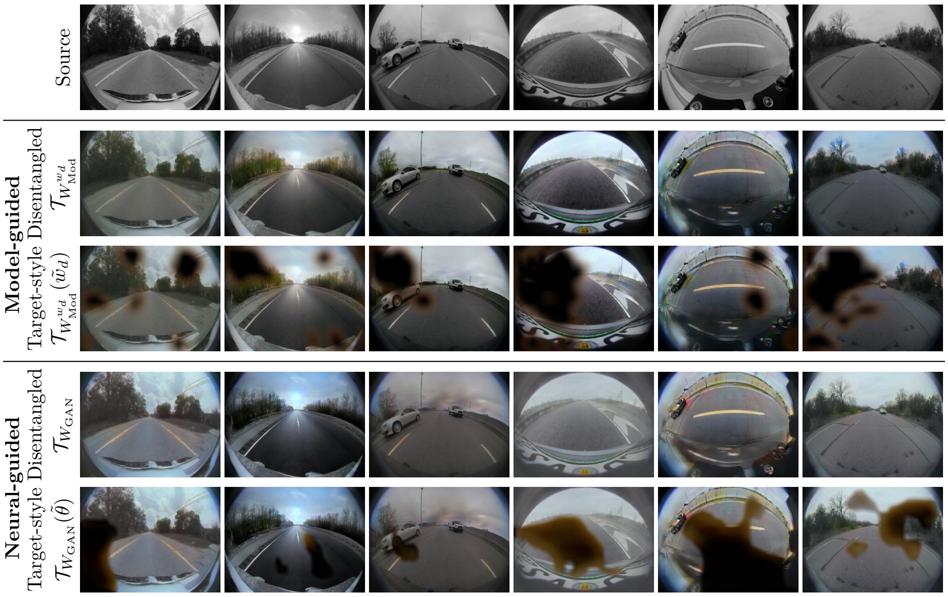
In brief, the study demonstrates that disentanglement is fairly perceived

Network	IS↑	LPIPS↑	CIS↑
MUNIT (Huang et al., 2018b)	1.06	0.656	1.08
Model-guided $\mathcal{T}_{W^{w_d}_{\text{Mod}}}(\tilde{w}_d)$	1.25	0.590	1.15
Neural-guided $\mathcal{T}_{W_{\text{GAN}}}(\tilde{\theta})$	**1.58**	**0.663**	**1.47**

(a) GAN metrics.

Network	SSIM↑	PSNR↑
MUNIT (Huang et al., 2018b)	0.414	13.4
Model-guided $\mathcal{T}_{W^{w_d}_{\text{Mod}}}$	**0.755**	**20.2**
Neural-guided $\mathcal{T}_{W_{\text{GAN}}}$	0.724	19.3

(b) Colorization


Source
Model-guided
Disentangled $\mathcal{T}_{W^{w_d}_{\text{Mod}}}$
Target-style $\mathcal{T}_{W^{w_d}_{\text{Mod}}}(\tilde{w}_d)$
Neural-guided
Disentangled $\mathcal{T}_{W_{\text{GAN}}}$
Target-style $\mathcal{T}_{W_{\text{GAN}}}(\tilde{\theta})$

(c) Qualitative evaluation.

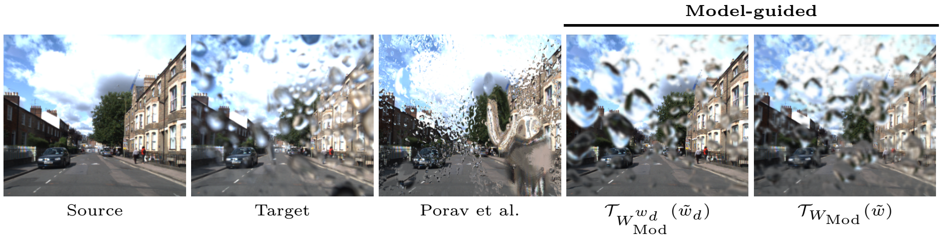
Figure 7.11: Comparison of model- and neural- guided disentanglement on gray ↦ color$_{\text{dirt}}$. Although our neural-guided strategy excels in image quality and diversity, mostly due to the complex nature of generated dirt (a), with model guidance we achieve more realistic image colorization (b). Qualitative results are coherent with metrics (c). With both pipelines, we still outperform MUNIT (Huang et al., 2018b), used as backbone.

by users (Fig. 7.10a) while preserving the learned underlying transformation (Figs. 7.10b , 7.10c).

7.5.2.2 Neural-guided disentanglement

Referring to Tab. 7.1 'Neural', we now evaluate our ability to disentangle visual traits with our neural guidance from Sec. 7.4, for Dirt disentanglement in the gray ↦ color$_{\text{dirt}}$ task and compare it to our model-guided strategy.

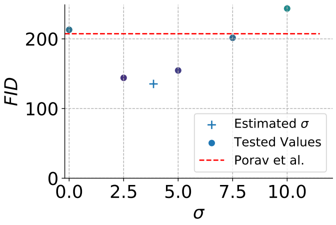
We leverage here the WoodScape (Yogamani et al., 2019) datasets having soiling semantic annotation as polygons. Following our training strategy (Fig. 7.5, bottom), our neural guidance DirtyGAN (Uricar et al., 2021) (cf. Sec. 7.5.1.4) is trained beforehand and frozen during the disentanglement.


Model-guided
Source
Target
Porav et al.
$\mathcal{T}_{W^{w_d}_{\text{Mod}}}(\tilde{w}_d)$
$\mathcal{T}_{W_{\text{Mod}}}(\tilde{w})$

(a) Sample images

Method	FID↓	LPIPS↓
Porav et al.	207.34	0.53
Model-guided $\mathcal{T}_{W^{w_d}_{\text{Mod}}}(\tilde{w}_d)$	**135.32**	0.44
Model-guided $\mathcal{T}_{W_{\text{Mod}}}(\tilde{w})$	157.44	**0.43**

(b) Benchmark on Porav et al.

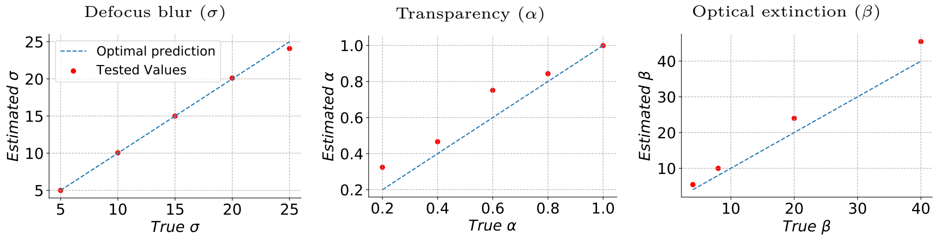


(c) FID

Figure 7.12: Realism of the injected occlusion. Our defocus blur σ estimation grants an increased realism in raindrop rendering on the RobotCar (Porav et al., 2019) dataset (a), compared with Porav et al. (Porav et al., 2019). This is confirmed by quantitative metrics (b). We report our model-guided translations using either differentiable parameter estimation only ($\mathcal{T}_{W^{w_d}_{\text{Mod}}}(\tilde{w}_d)$) or the full model parameter estimation ($\mathcal{T}_{W_{\text{Mod}}}(\tilde{w})$), outperforming Porav et al. (Porav et al., 2019) in both. In (c), we evaluate the FID for different σ values in [0, 10], showing that our regressed σ value ($\sigma = 3.81$) actually leads to a local minimum.

The use of annotations boosts the overall quality and diversity, which is proved in Tab. 7.11a where our neural-guided outperforms both MUNIT baseline and our own model-guided version. Furthermore, since the ground truth for colorization is available, we evaluate in Tab. 7.11b the effectiveness of disentanglement with SSIM and PSNR metrics (higher is better). Here both disentanglement outperform MUNIT (Huang et al., 2018b) significantly, but model-guided is better. Arguably, we attribute this to the worse gradient propagation due to more occluded pixels with respect to our physical model[4]. Finally, last 2 rows of Fig. 7.11c show our neural-guided strategy produces high quality *colored* images *without* occlusions ('Disentangled' row, $\mathcal{T}_{W_{\text{GAN}}}$) while injection of occlusions with optimal estimated parameters $\tilde{\theta}$ ('Target-style' row, $\mathcal{T}_{W_{\text{GAN}}}(\tilde{\theta})$) also mimics target appearance. In fact while both neural-guided disentanglement and physical model-guided disentangle-

[4] On average, DirtyGAN dirt covers 25.4% of the image while our physical model covered 20.1%. While this provides more realistic dirt masks (ground truth annotation is 29.6%) we conjecture this leads to worse gradient propagation.



(a) Raindrop (b) Dirt (c) Fog

Figure 7.13: Evaluation of the model parameters regression. The reliability of our parameter estimation is assessed on synthetic datasets augmented with arbitrary physical models acting as ground truth values. Comparing against our regressed value, our strategy performs better when low modifications on the estimated values corresponds to big visual changes (average error is 0.99% for raindrops (a), 3.55% for dirt (b)). For fog (c), we get an higher error of 23.51% due to the low visual impact of high β values.

ment perform well, only our model-guided strategies controllability of the occlusion at inference. This is because of the explicit physical parameters in the models, that allows reinjecting unseen models at inference.

7.5.3 Parameters estimation

We now evaluate the effectiveness of our parameter estimation for physical model-guided disentanglement, considering only differentiable parameters first and later extending to our full system. The neural-guided disentanglement strategy precludes this analysis due to the lack of explicit parameters.

7.5.3.1 Differentiable model ($w = \{w_d\}$)

To evaluate realism, we leverage the RobotCar (Porav et al., 2019) dataset having pairs of clear/raindrop images. Since there is no domain shift between image pairs, we set $G(x) = x$ and regress the defocus blur (σ) again following Sec. 7.3.3. The regressed $\sigma = 3.87$ is used to render raindrops on clear images. Using FID and LPIPS distances we measure perceived distance between real raindrop images and our model-guided raindrops translations ($\mathcal{T}_{W_{\text{Mod}}^{w_d}}(\tilde{w}_d)$) or the one of Porav et al. (Porav et al., 2019). Fig. 7.12b shows we greatly boost similarity[5] (-72.02 FID) with real raindrop images. This is qualitatively verified in Fig. 7.12a, where our rendered raindrops are more similar to *Target*. To provide insights about the quality of our minima, we also evaluate FID for arbitrary σ values ($\sigma \in \{0.0, 2.5, 5.0, 7.5, 10\}$).

[5] Please note that unlike previous experiments, here LPIPS is used for distance estimation (not diversity), so lower is better.

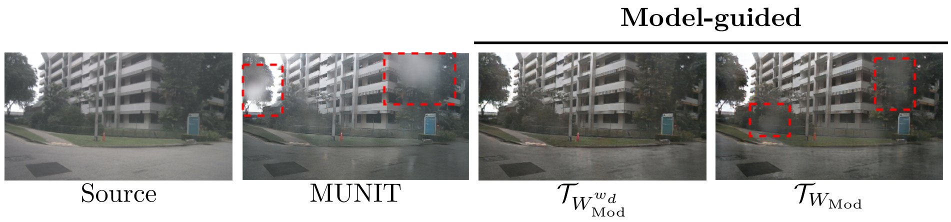

Model-guided
Source
MUNIT
$\mathcal{T}_{W^{w_d}_{\text{Mod}}}$
$\mathcal{T}_{W_{\text{Mod}}}$

Figure 7.14: Full model on clear ↦ $\text{rain}_{\text{drop}}$. With complete parameter estimation ($\mathcal{T}_{W_{\text{Mod}}}$, rightmost), we achieve a slightly worse disentanglement than with manually-tuned non-differentiable parameters ($\mathcal{T}^{w_d}_{W_{\text{Mod}}}$), visible in red areas of $\mathcal{T}_{W_{\text{Mod}}}$. However, in both of our translations we generate typical rain traits as reflections with reasonable disentanglement, while baseline MUNIT (Huang et al., 2018b) has very evident raindrops entangled highlighted in red.

Fig. 7.12c proves that our estimated sigma best minimized perceptual distances despite the weak discriminator signal.

To measure the accuracy of our differentiable parameter estimation we need paired images with and without physical traits with completely known physical parameters. To the best of our knowledge such dataset does not exists. Instead, we augment RobotCar (Porav et al., 2019), WoodScape (Yogamani et al., 2019) and Synthia (Ros et al., 2016) with synthetic raindrops, dirt, and fog, respectively, with gradually increasing values of defocus blur (σ) for raindrop, transparency (α) for dirt[6] and optical thickness (β) for fog. Using each augmented dataset, we then regress said parameters following Sec. 7.3.3.

Plots in Fig. 7.13 show estimation versus ground-truth. In average, the estimation error is 0.99% for raindrop, 3.55% for dirt, and 23.51% for fog. The very low σ error for raindrop is to be imputed to the defocus blur that drastically changes scene appearance, while higher error for β must be imputed to the logarithmic dependency of the fog model. Nevertheless, translations preserve realism (cf. Fig. 7.9).

7.5.3.2 Full model ($w = \{w_d, w_{nd}\}$)

To evaluate the quality of our full raindrop model, we incorporate this time the non-differentiable parameters (*i.e.* s, p, t) which are estimated with our genetic strategy in Sec. 7.3.4 for 4 types of drops, with a genetic population size of 10. As shown in Fig. 7.12b, LPIPS metric privileges our full model-guided estimation ($\mathcal{T}_{W_{\text{Mod}}}(\tilde{w})$) while FID suffers compared to using differentiable parameters only. However, we very significantly out-

[6] In this experiment, we consider dirt with a fixed defocus blur value σ and regress only α to increase the diversity of tasks.

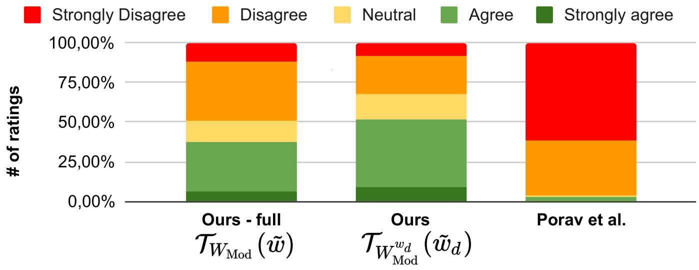


Figure 7.15: Parameter estimation user study. We presented users with {*Reference*, *Model*} image pairs where *Reference* includes real drops and *Model* has fake drops rendered with our method with differentiable only (Ours) or full (Ours - full) parameters estimation, or with Porav et al. (Porav et al., 2019). Users were asked whether they agree on the statement ”The drops of the Model resemble the drops of Reference”. Thanks to our estimation strategy, we dramatically improve similarity to real raindrops.

perform (Porav et al., 2019) also qualitatively (Fig. 7.12a). The mitigated results are explained by the much more complex optimization problem having many more parameters, and by the limited computation time for genetic iterations. However, this let us foresee applications in high-dimensionality problems where manual approximation is not always possible or with a less accurate model (see ablations Sec. 7.5.4.2).
Results on the clear ↦ rain$_{\text{drop}}$ task in Fig. 7.14 are coherent with above insights as the full model estimation, although effective, exhibits slightly lower quality disentanglement.

7.5.3.3 User study

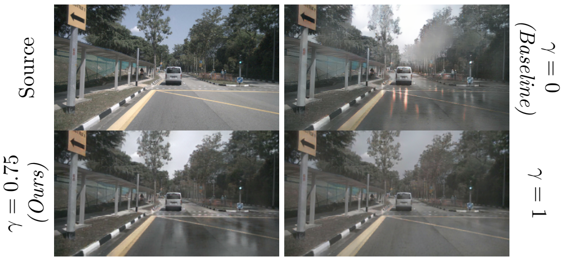
We presented to users (see Sec. 7.5.1.5) couples of images with independent scenes in which the left one presented images with real drops taken from RobotCar (Porav et al., 2019), while the right one included fake drops rendered with our model with differentiable only / all parameters estimated, or with Porav et al. (Porav et al., 2019). Users were asked to compare raindrops appearance between the two images regardless of the represented scenes. From results shown in Fig. 7.15, it is evident that our method largely outperform the baseline in both configurations, indicating a higher quality of our raindrops also for the human preference metric.

7.5.4 Ablation studies

We now ablate our proposal. We focus on the model-guided setting by tuning genetic processing, altering model complexity, changing models, or removing disentanglement guidance.

Model	IS↑	LPIPS↑	CIS↑
none	1.21	0.50	1.03
Gaussian	1.35	0.51	1.13
Refract	1.46	0.50	1.12
Ours	**1.53**	**0.52**	**1.15**

(a) Model complexity



(b) Disentanglement Guidance

Figure 7.16: Ablations of model complexity and Disentanglement Guidance. In (a), we quantify disentanglement effects with simpler model having less variability (*Refract*), or only color guidance (*Gaussian*). Even if complexity is beneficial for disentanglement (*Ours*), simple models permits disentanglement to some extent. In (b), we study the efficacy of the Disentanglement Guidance (DG) for different γ values on clear $\mapsto$ $\text{rain}_{\text{drop}}$ task. With $\gamma = 0$ our approach fallbacks to the baseline and entangles occlusions, while with guidance $\gamma = 1$ the translation lacks important features such as reflections and glares. With $\gamma = 0.75$ we simultaneously avoid entanglements and preserve translation capabilities.

7.5.4.1 Non-differentiable genetic estimation

We study the effectiveness of our genetic estimation ablating the population size of our raindrop model on RobotCar (Porav et al., 2019) as in Sec. 7.5.3. We test our algorithm with population size 10/25/50/100, obtaining FID 157.44/153.32/151.21/**149.09** and LPIPS **0.43**/0.44/0.44/**0.43**. While we observe an obvious increase in performances, this comes with additional computation times, hence we used the lowest population size of 10 for all tests. Nevertheless, this opens doors to potential improvements in the full parametric estimation.

7.5.4.2 Model complexity

We study the influence of the model for the clear $\mapsto$ $\text{rain}_{\text{drop}}$ on nuScenes (Caesar et al., 2020) task. Specifically, we evaluate three raindrop models of decreasing complexity: 1) Our model from Sec. 7.5.1.3 (named *Ours*). 2) The same model but without shape and thickness variability (*Refract*), and 3) A naive non-parametric colored Gaussian-shape model (*Gaussian*). Note that *Gaussian* is deprived of any refractive property as it uses fixed color, and does not regress any physical parameters. In Fig. 7.16a, we report GAN metrics for all models following Sec. 7.5.2.1. Even if increasing model complexity is beneficial for disentanglement, very simple models still lead to a performance boost. We advocate the best performances of *Ours* to a more effective adversarial training, as consequence of increased realism.

7.5.4.3 Model choice

To also evaluate whether injected features only behave as adversarial noise regardless of the chosen model, we trained on RobotCar (Porav et al., 2019) (as in Sec. 7.5.3) though purposely using an incorrect model as watermark, dirt, fence. Evaluating the FID against real raindrop images, we measure **135.32** (raindrop) / 329.17 (watermark) / 334.76 (dirt) / 948.71 (fence), proving necessity of using the ad-hoc model.

7.5.4.4 Disentanglement Guidance (DG)

We use the nuScenes clear $\mapsto$ rain$_{\text{drop}}$ task to visualize the effects of different DG strategies (Sec. 7.3.5). For varying values of the DG threshold γ in Fig. 7.16b we see results ranging from no guidance ($\gamma = 0$) to strict guidance ($\gamma = 1$). With lax guidance ($\gamma = 0$), we fall back in the baseline scenario with visible entanglement effects, while with $\gamma = 1$ we do achieve disentanglement, at the cost of losing important visual features as reflections on the road. Only appropriate guidance ($\gamma = 0.75$) achieves disentanglement and preserves realism.

7.6 Discussion

To our best knowledge, we have designed the first unsupervised strategy to disentangle physics-based features in i2i. Good qualitative and quantitative performances showcase promising interest for several applications, still there are peculiar points and limitations which we now discuss.

Independence assumption. For unsupervised disentanglement, we assume the physical model to be completely independent from the scene, in order to use our intuition about marginal separation (see Sec. 7.3.1 and Eq. 7.2). However, since physical models may need the underlying scene to correctly render desired traits, one may argue their appearance is not completely disentangled. While this is true from a visual point of view, it is not from a physical one. Let's interpret disentanglement properties to be dependent on scene *elements*. In presence of disentanglement, the same physical model could be applied to different objects regardless of what they are. For instance, we could use the same raindrop refraction map on either roads or buildings with identical parameters. In this sense, $G(x)$ dependency in physical models is not impacting our visual independence assumption.

On partial entanglement issues. We observe in some cases that gradient propagation can be affected by fixed entanglement of occlusion features. This is the case for example for sky regions in fog (Sec. 7.5.2.1) because physics formalizes that regardless of its intensity fog is always entangled at far (Narasimhan and Nayar, 2002). In such scenarios, disentanglement will perform poorly because the generator will not get any discriminative feedback. In many other cases however, Disentanglement Guidance (DG, Sec. 7.3.5) mitigates the phenomenon as it blocks injection of the physical model in relevant image regions. We conjecture that the effectiveness could be extended by varying DG at training time to ensure a balanced gradient propagation.

On genetic estimation effectiveness. The sub-optimal performances of our genetic estimation of w_{nd} are imputed to the much more complex search space, in which we vary all parameters of our physical model simultaneously. Although we did set fairly large search limits for w_{nd}, one could envisage a mixed training in which the search space is limited to reasonable hand-tuned limits. In this sense, genetic estimation of w_{nd} could be seen as a minimum mining technique, ensuring increased performances on the hand-tuned values.

On model realism. The method proposed relies on realistic models, *i.e.* able to render physical traits which are desirable to include in output images. Their availability is constrained by the nature of the physical phenomenon to disentangle. It is not always easy to model the physics of a phenomenon: there could be scenarios, as night, which are difficult to describe physically, since their appearance is impacted by multiple factors (*e.g.* reflectivity, light sources, etc.). We argue that in those cases it is not ideal to include physical models in images, but it is preferable to leverage physics at the feature level only. This idea is further developed in the CoMoGAN methodology presented in Chapter 8, which exploits *naive* models to guide a continuous transformation.

Chapter 8

CoMoGAN: continuous model-guided image-to-image translation

The contributions of this chapter have been published in:

Pizzati, F., Cerri, P., and de Charette, R. (2021a). CoMoGAN: continuous model-guided image-to-image translation. In *CVPR (oral)*

arXiv: `https://arxiv.org/abs/2103.06879`
Code: `https://github.com/cv-rits/CoMoGAN`

Contents

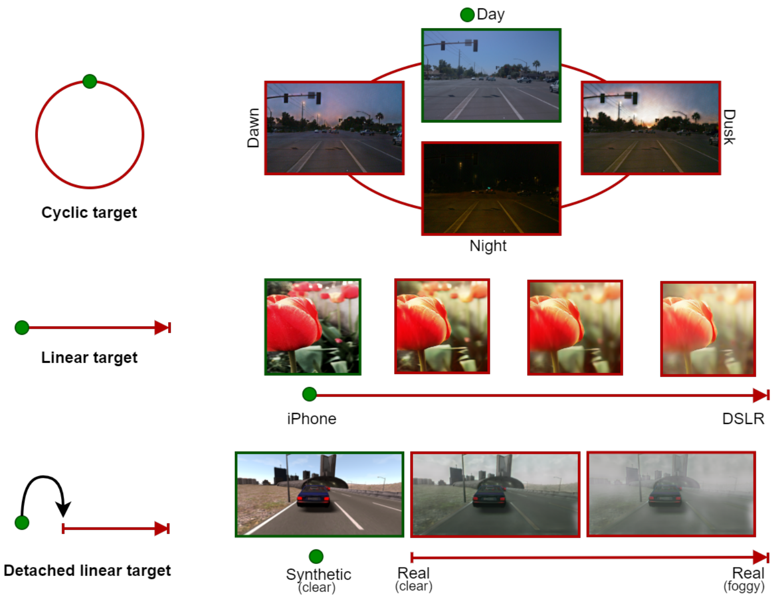


Figure 8.1: Detaching from traditional i2i translation, we are interested in *continuous* mapping from source domain (green point) to a target domain (red lines), in single- or multi- modal setup. A key feature of our proposal, is unsupervised reorganization of the data along a functional manifold (top: cyclic, middle/bottom: linear). We leverage lighting translations from day images (top), shallower depth of field from in-focus images (middle), or synthetic clear images to realistic foggy images (bottom).

8.1 Problem statement

Despite impressive leaps forward with unpaired (Zhu et al., 2017a; Liu et al., 2017), multi-target (Choi et al., 2020; Wu et al., 2019a), or continuous (Wang et al., 2019b; Gong et al., 2019) i2i, there are still important limitations. Specifically, to learn complex continuous translations existing works require supervision on intermediate domain points. Also, they assume piece-wise or entire linearity of the domain manifold. Such constraints can hardly meet cyclic translations (*e.g.* daytime) or continuous ones costly or impractical to label (*e.g.* fog, rain).

Instead, we introduce CoMoGAN, the first i2i framework learning nonlinear continuous translations with unsupervised target data. It is trained using simple physics-inspired models for guidance, while relaxing model dependency via continuous disentanglement of domain features. An interesting resulting property is that CoMoGAN discovers the target data manifold ordering, unsupervised. Of importance, we *guide* the GAN with a naive physical model, but the output of our network is fully neural. This is different from what was presented in Chapter 7, where the physical model was part of the output rendering, making realism dependent on the model complexity. This relax some constraints of realism for physical models used in CoMoGAN, that must just describe naively the physics of the transformation they aim to model, without necessarily being visually pleasing. For evaluation we propose new translation tasks, shown in Fig. 8.1, being either cyclic/linear, attached/detached from source. Our contributions are:

- a novel model-guided setting for continuous i2i,
- CoMoGAN: an unsupervised framework for disentanglement of continuously evolving features in generated images, using simple model guidance,
- a novel Functional Instance Normalization (FIN) layer,
- the evaluation of CoMoGAN against recent baselines and new tasks, outperforming the literature on all.

8.2 Related works

The state-of-the-art for physics-based image translation and disentangled representations, also relevant for CoMoGAN, has been presented in Chapter 7. Moreover, since we introduce the task of continuous i2i, we include relevant works.

Continuous image translation. A common practice for continuous i2i is to use intermediate domains by weighting discriminator (Gong et al., 2019, 2021), using losses for middle states (Wu et al., 2019a), or mixing disentangled styles representations (Choi et al., 2020; Romero et al., 2019). Attribute vectors interpolation (Xiao et al., 2018; Zhang et al., 2019; Mao et al., 2020) enables continuous control of several features. Others continuously navigate latent spaces with discovered paths (Chen et al., 2019b; Goetschalckx et al., 2019; Jahanian et al., 2020). Finally, feature (Upchurch et al., 2017) or kernel (Wang et al., 2019b) interpolation were proposed. Still, they assume linear interpolation – not always valid (*e.g.* day to night include dusk). GANimation (Pumarola et al., 2020) instead, use non-linear interpolations but require intermediate domain labels.

8.3 The CoMoGAN methodology

Instead of a point-to-point mapping ($X \mapsto Y$), CoMoGAN learns a continuous domain translation controlled by ϕ, that is $X \mapsto Y(\phi)$. Training uses source data (at fixed ϕ_0) and unsupervised target data (unknown ϕ). It reshapes the data manifold guided by naive physics-inspired models (*e.g.* tone-mapping, blurring, etc.). Rather than mimicry, we relax the model and let the networks discover private image features via our disentanglement of output, ϕ, and style.

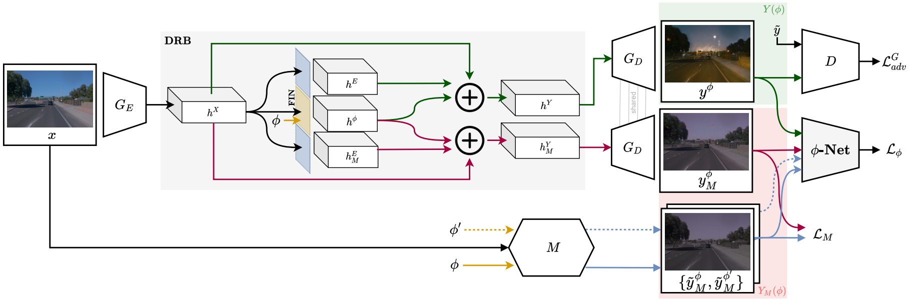


Figure 8.2: CoMoGAN enables unsupervised continuous translation, being end-to-end trainable, and architecture agnostic. Our Disentanglement Residual Block (DRB) – placed between encoder/decoder (G_E/G_D) – uses new Functional Instance Normalization (FIN, yellow layer) to learn manifold reshaping and continuous translation, guided with simple physics-inspired model M. For losses ($\mathcal{L}$), on top of standard ones we optimize model reconstruction ($\mathcal{L}_M$) and manifold consistency ($\mathcal{L}_\phi$) by enforcing manifold distances between GAN output and model outputs $\{\phi, \phi'\}$ with a pair-wise estimator (ϕ-Net).

Fig. 8.2 is an overview of our architecture-agnostic proposal. It relies on three key components. We first introduce Functional Instance Normalization layer (Sec. 8.3.1) which enables ϕ-manifold reshaping. Second, our Disentanglement Residual Block (Sec. 8.3.2) in charge of ϕ disentanglement in input data. Finally, we detail ϕ-Net, a pair-wise ϕ regression network (Sec. 8.3.3) which enforces manifold distances consistency.

Model guidance. We guide the learning with *simple* non-neural models $M(x, \phi)$, x the source image. Thus, following the intuition that target manifold can be discovered with coarse guidance: night resembles *dark* day, fog looks like a *blurry gray* clear image, etc. We depart from the need of complex physical guidance since we disentangle shared and private features from model/translation which enables discovering complex non-modeled features (*e.g.* light sources at night). Models are described in Sec. 8.4.1.

8.3.1 Functional Instance Normalization (FIN)

To take advantage of our model guidance which is continuous by nature, we must allow our network to encode ϕ continuity. To do so, we build on prior Instance Normalization (IN) which allows carrying style-related information (Ulyanov et al., 2017; Huang and Belongie, 2017). It writes for input x,

$$\mathrm{IN}(x) = \frac{x - \mu}{\sigma}\gamma + \beta, \tag{8.1}$$

where μ and σ are input feature statistics, and γ and β learned parameters of an affine transformation. As an extension, we propose Functional Instance Normalization (FIN)

$$\mathrm{FIN}(x, \phi) = \frac{x - \mu}{\sigma} f_\gamma(\phi) + f_\beta(\phi), \tag{8.2}$$

where instead of learning a unique value of affine transformation parameters, we learn the distribution of transformations f_γ and f_β. The intuition is to shape the ϕ-manifold based on how the transformation evolves. Compared to others (Gong et al., 2019), this allows us to interpret better the learned manifold. Depending on the nature of $Y(\phi)$, we can encode FIN layer accordingly. In this work, we investigate linear and cyclic encoding. Linear encoding is commonly encountered, and assumes reorganizing features linearly. For instance, considering adverse weather phenomena, severe conditions (*e.g.* thick fog) are always positioned after light ones (*i.e.* lite fog). We model linear FIN parameters as

$$\begin{aligned} f_\gamma(\phi) &= a_\gamma \phi + b_\gamma, \\ f_\beta \phi &= a_\beta \phi + b_\beta, \end{aligned} \tag{8.3}$$

with $\{a_\gamma, a_\beta, b_\gamma, b_\beta\}$ the learnable parameters of the layer.

Conversely, some translations path loop back to source, as it happens with daylight, which is *cyclic* by nature going from Day to Dusk, Night, Dawn and Day again. In this case, we encode cyclic FIN layer with parameters

$$\begin{aligned} f_\gamma(\phi) &= a_\gamma cos(\phi) + b_\gamma, \\ f_\beta(\phi) &= a_\beta sin(\phi) + b_\beta. \end{aligned} \tag{8.4}$$

8.3.2 Disentanglement Residual Block (DRB)

The pitfall of strict model-dependency is that the GAN will only learn to mimic the model. To prevent that, we must allow target domain $Y(\phi)$ and *model* domain $Y_M(\phi)$ to have shared *modeled* features Y^ϕ but also private *non-modeled* features Y^E and Y^E_M, respectively. This writes

$$\begin{aligned} Y(\phi) &= \left\{Y^\phi, Y^E\right\}, \\ Y_M(\phi) &= \left\{Y^\phi, Y^E_M\right\}. \end{aligned} \tag{8.5}$$

We enable private features in either domain with our Disentanglement Residual Block (DRB, shown in Fig. 8.2) whose goal is to extract disentangled representations for a given ϕ. The DRB is composed of residual blocks mapping the encoder feature map h^X to the disentangled representations of output images. Let $y^\phi \in Y(\phi), y^\phi_M \in Y_M(\phi)$, we have

$$\begin{gathered} \mathrm{DRB}(h^X, \phi) = \{h^Y, h^Y_M\}, \\ y^\phi = G_D(h^Y), \qquad y^\phi_M = G_D(h^Y_M). \end{gathered} \tag{8.6}$$

The DRB works as follows. Following Fig. 8.2, the input representation h^X is processed by residual blocks, each one extracting features associated with the atomic ones previously introduced, such as $Y^\phi, Y^E, Y^E_M \longleftrightarrow h^\phi, h^E, h^E_M$, one per residual. In particular, the residual block for h^ϕ extraction uses our FIN layers for normalization to encode continuous features. The hidden latent representations h^Y and h^Y_M are obtained from summation of the disentangled features and h^X to ease gradient propagation as in (He et al., 2016). In formulas,

$$\begin{aligned} h^Y &= h^\phi + h^E + h^X \,, \\ h^Y_M &= h^\phi + h^E_M + h^X \,. \end{aligned} \tag{8.7}$$

Intuitively, for optimization we need feedback from both real data similarity and mimicking of the model output. While the first must rely on adversarial training due to the use of unpaired images, we can enforce reconstruction on the paired modeled $\tilde{y}^\phi_M = M(x, \phi)$. Assuming LSGAN (Mao et al., 2017a) training and discriminator D, we obtain

$$\begin{aligned} \mathcal{L}^G_{adv} &= ||D(y^\phi) - 1||_2, \\ \mathcal{L}_M &= ||y^\phi_M - \tilde{y}^\phi_M||_1. \end{aligned} \tag{8.8}$$

Minimization of $\mathcal{L}^G_{adv}$ and $\mathcal{L}_M$ during the generator update step enables disentanglement of h^E and h^E_M.

8.3.3 Pairwise regression network (ϕ-Net)

The DRB enforces both disentanglement and manifold shape at a feature level, but it requires ad-hoc training strategies to actually disentangle also continuous features for real images and not fall into easy pitfalls, *e.g.* the network only exploiting h^E for target translation ignoring h^ϕ. Hence, we introduce a training strategy based on similarities which forces the network to both exploit extracted continuous information and follow the model guidance. Suppose an input image x, mapped to $x \mapsto y^\phi$ by the network. As shown in Fig. 8.2, we randomly sample ϕ and ϕ' and apply $M(.)$ to x, obtaining the couple $\{\tilde{y}^\phi_M, \tilde{y}^{\phi'}_M\}$. We use a CNN ($\phi$-Net) for domain similarity discovery. It takes as input a pair of images and regresses their ϕ differences, such as

$$\phi\text{-Net}(y^\phi, y^{\phi'}) = \phi - \phi' = \Delta\phi \,. \tag{8.9}$$

We jointly optimize ϕ-Net and generator (G) parameters in an end-to-end setting by enforcing consistency between real and modeled target domain images. In formulas,

$$\begin{aligned} \mathcal{L}^G_\phi &= ||\phi\text{-Net}(y^\phi, \tilde{y}^\phi_M)||_2 + ||\phi\text{-Net}(y^\phi, \tilde{y}^{\phi'}_M) - \Delta\phi||_2, \\ \mathcal{L}_{gt} &= ||\phi\text{-Net}(\tilde{y}^\phi_M, \tilde{y}^{\phi'}_M) - \Delta\phi||_2, \\ \mathcal{L}_\phi &= \mathcal{L}^G_\phi + \mathcal{L}_{gt}. \end{aligned} \tag{8.10}$$

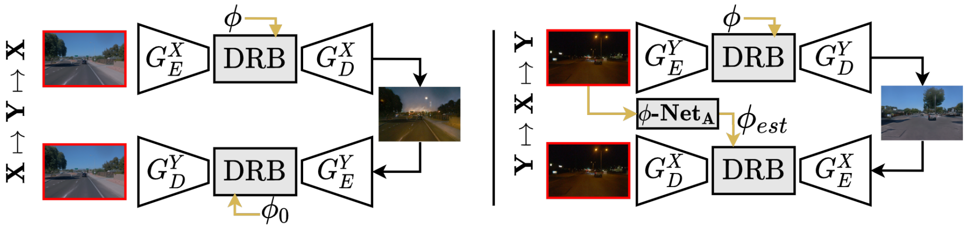


Figure 8.3: We enforce cycle consistency by injecting the source ϕ_0 in the $X \mapsto Y \mapsto X$ translation when reconstructing the original image. Also, for $Y \mapsto X \mapsto Y$ we position the input image at ϕ_{est} on the domain using our ϕ-Net$_A$ CNN trained unsupervised for ϕ regression.

$\mathcal{L}_\phi^G$ forces G to organize the manifold following the feedback of the physical model, ultimately resulting in generated y^ϕ and $\tilde{y}_M^\phi$ to be mapped to the same ϕ on the manifold discovered by ϕ-Net. That way, the network can identify that images follow some similarity criteria despite differences between model output and learned translation, leading to an organization of the latent space guided by the physical model. $\mathcal{L}_{gt}$ instead exploits modeled data only and thus is used to avoid training collapse. For linear FIN, we train on ϕ and $\Delta\phi$, though for cyclic one stability is increased by evaluating each loss on sin/cos projection of ϕ.

8.3.4 Training strategy

CoMoGAN is end-to-end trainable and can be used with any i2i framework by simply adding the DRB between encoder and decoder, with our losses. The final objective for the generator depends if source and target are detached, *i.e.* $X \not\subset Y$ (see Fig. 8.1 for visualization). If detached, the generator update step writes

$$\mathcal{L}^G = \mathcal{L}_{adv}^G + \mathcal{L}_M + \mathcal{L}_\phi \,. \tag{8.11}$$

For attached source/target, we enforce source (ϕ_0) identity:

$$\mathcal{L}^G = \mathcal{L}_{adv}^G + \mathcal{L}_M + \mathcal{L}_\phi + ||G(x, \phi_0) - x||_1 \,. \tag{8.12}$$

Using real data ($\tilde{y}$) from target the discriminator minimizes

$$\mathcal{L}^D = \mathcal{L}_{adv}^D = ||D(y^\phi)||_2 + ||D(\tilde{y}) - 1||_2 \,.$$

8.3.4.1 Cycle consistency.

In addition to $X \mapsto Y$, many networks perform $Y \mapsto X$ to preserve context with cycle consistency. To handle the latter, we insert a *shared* DRB between

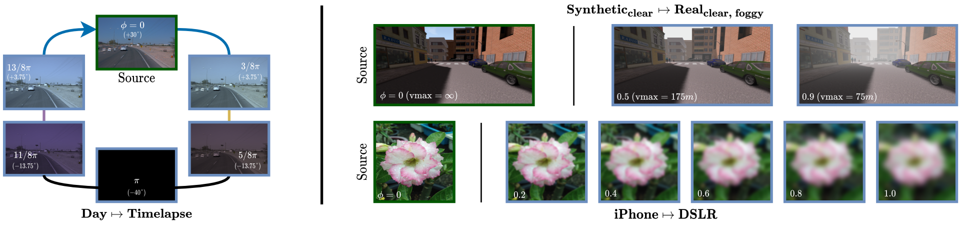


Figure 8.4: Model guidance for training for sample ϕ values (white text inset).

each encoder/decoder couple to benefit from multiple sources. This is illustrated in Fig. 8.3. We also use another unsupervised network, called ϕ-Net$_A$, that regresses ϕ on the target dataset. From above figure (left), because ϕ is injected in $X \mapsto Y$ transformation, we enforce a correct spreading of all ϕ values by adding $\mathcal{L}_{reg}$ to the generator objective, $\mathcal{L}_{reg} = ||\phi\text{-Net}_A(y^{\phi}) - \phi||_2$.

8.4 Experiments

We show the efficiency of CoMoGAN on new continuous image-to-image translation tasks $X \mapsto Y(\phi)$, where we consider source data to lie on a fixed point (ϕ_0) of the ϕ-manifold and *unknown* ϕ target data. The underlying optimization challenge is to learn simultaneously the ϕ-manifold and continuous image translation. Because continuous model-guided translation is new, we first describe our three novel translations tasks (Sec. 8.4.1) obtained by leveraging recent datasets at the time of submission (Sun et al., 2020; Ros et al., 2016; Cordts et al., 2016; Halder et al., 2019; Zhu et al., 2017a). Each task encompasses challenges of its own such as linear/cyclic target manifold, attached/detached manifolds (*i.e.* $X \subset Y$ or $X \not\subset Y$) and uni-/multi-modality. Specifically, we train with backbone MUNIT (Huang et al., 2018b) (multi-modal) or CycleGAN (Zhu et al., 2017a) (uni-modal) and coin our alternatives CoMo-MUNIT and CoMo-CycleGAN, respectively. We evaluate the manifold organization (Sec. 8.4.2) and the translation quality (Sec. 8.4.3) from GAN metrics and proxy tasks. Continuous translation (Sec. 8.4.4) is evaluated separately and we conclude with ablation studies (Sec. 8.4.5). We train with the default backbone hyperparameters.

8.4.1 Translation tasks

8.4.1.1 Day ↦ Timelapse

Using the Waymo Open dataset (Sun et al., 2020), we frame the complex task of day to any time, thus learning *timelapse* passing through day/dusk/night/dawn. Waymo image labels are *only* used to split clear images into *source*

{Day} and *target* {Dusk/Dawn, Night}, respectively obtaining train/val sets of 105307/28165 and 27272/7682 images. We train CoMo-MUNIT for multi-modality. To respect the cyclic nature of time we exploit cyclic FIN (Eq. 8.4) encoding $\phi \in [0, 2\pi]$, which maps to a sun elevation $\in [+30°, -40°]$. *For evaluation only*, we obtain ground truth elevation from astronomical models (pys, 2007) with image GPS position and timestamp. For guidance, we render intermediate conditions by interpolating the tone-mapped model from (Thompson et al., 2002), written $\Omega(.)$. Since the latter was originally designed only for night time rendering, we replace the target color in $\Omega(.)$ by the average of the Hosek sky radiance model (Hosek and Wilkie, 2012), denoted $\text{HSK}(\phi)$. For implementation reason, we accordingly map ϕ to $[0, 2\pi]$ so that max and min sun elevation angles corresponding to 30° and −40°, respectively. The complete model writes

$$M(x, \phi) = (1 - \alpha)x + \alpha\Omega(x, \text{HSK}(\phi) + \text{corr}(\phi)) + \text{corr}(\phi)\,, \tag{8.13}$$

with α the interpolation coefficient defined as,

$$\alpha = \frac{1 - \cos(\phi)}{2}$$

and $\text{corr}(\phi)$ an asymmetrical hue correction to arbitrarily account for temperature difference at dusk and dawn. It writes

$$\text{corr}(\phi) = \begin{cases} [0.1, 0.0, 0.0]\sin(\phi) & \text{if } \sin(\phi) > 0, \\ [0.1, 0.0, 0.1](-\sin(\phi)) & \text{Otherwise.} \end{cases} \tag{8.14}$$

The effect of corr(.) is visible in Fig. 8.4 at $\phi = 5/8\pi$ and $\phi = 11/8\pi$, which both maps to elevation of −13.75° for dusk (right image) and dawn (left image). We found that it slightly pushes the network towards better discovery of the red-ish and purple-ish appearance of dusk and dawn, respectively.

8.4.1.2 iPhone $\mapsto$ DSLR

We inspire from CycleGAN (Zhu et al., 2017a) by adapting their initial task to a continuous setup, learning the mapping of iPhone images with large depth of field to DSLR images with shallow depth of field. We also use the iphone2dslr flowers dataset (Zhu et al., 2017a), split in *source* 1182/569 and *target* 3325/480. We train this task with CoMo-CycleGAN for comparison, and use linear FIN (Eq. 8.3) where $\phi \in [0, 1]$ encodes the progression.

As model for guidance, we simply use gaussian blurring, with kernel radius in pixels accordingly mapped to ϕ values, as

$$M(x, k) = G(k) * x\,, \tag{8.15}$$

being G the Gaussian kernel, x input and k kernel size, which is directly mapped from $\phi \in [0, 1] \mapsto k \in [0, 8]$. Effects on images are in Fig. 8.4.

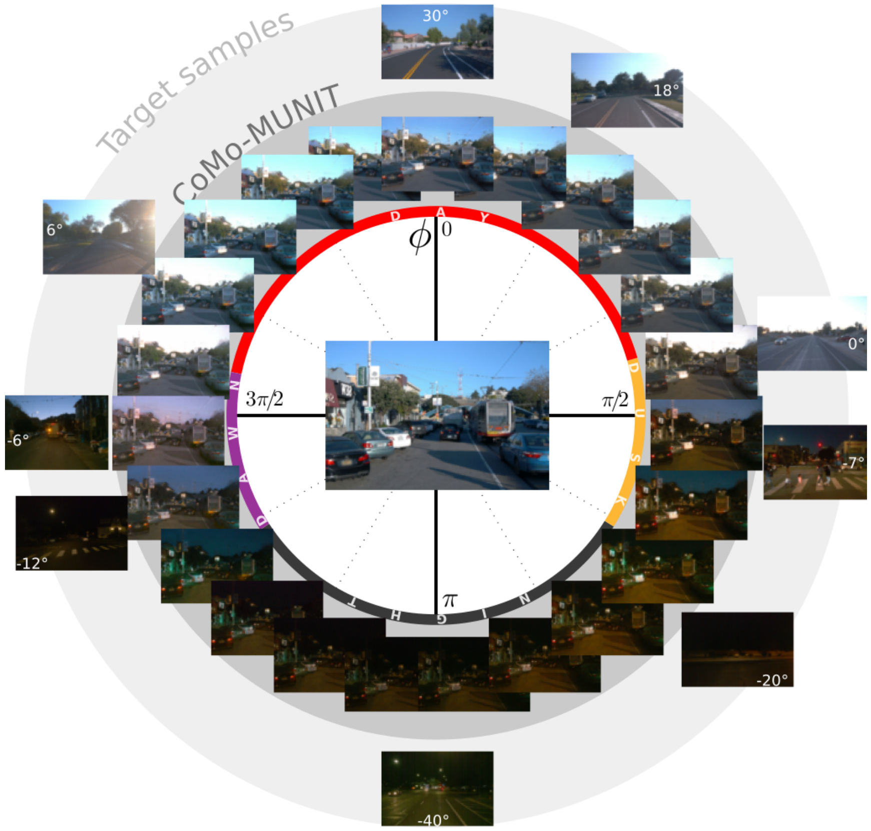


Figure 8.5: Translations (dark circle) of a source day image (center) exhibit both high variability and similarities with target data (outer circle) for which we report ground truth elevations. CoMo-MUNIT learned non-modeled visual features like frontal sun scenes resembling real ones (as in $\{0°, 6°, 18°\}$). Note that it discovered dawn/dusk and the stationary appearance of night, proving manifold quality.

8.4.1.3 $\textbf{Synthetic}_{\textbf{clear}} \mapsto \textbf{Real}_{\textbf{clear, foggy}}$

Here, we propose a detached source and target task, where we learn clear to foggy except that source is *synthetic* and target is *real* data. For *source*, we leverage spring sequences of synthetic Synthia dataset (Ros et al., 2016), split in 3497/959 images. As *target* we mix original Cityscapes (Cordts et al., 2016) and 4 augmented foggy Weather Cityscapes (Halder et al., 2019) with max visibility distances {750m, 350m, 150m, 75m}. In target, each of the 5 Cityscapes version has 2975/500 images. We train here a CoMo-MUNIT with linear FIN layer (Eq. 8.3) and encode maximum visibility as $\phi \in [0, 1]$, *i.e.* visibility $\in [\infty, 70m]$. For guidance, we simply exploit the fog model of (Halder et al., 2019). Sample outputs are shown in Fig. 8.4.

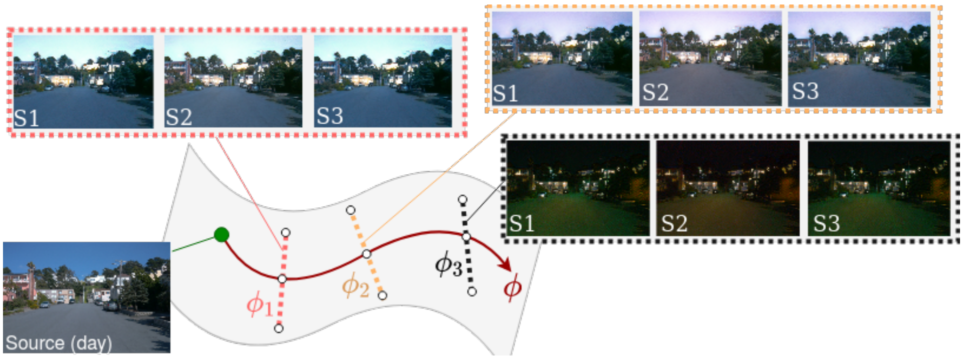


Figure 8.6: Translations along dimensions ϕ (red) and style (dotted). For a given ϕ, the styles vary slightly (notice hue and brightness), proving disentanglement of ϕ and style.

8.4.2 Manifold organization

We evaluate the quality of the unsupervised manifold discovery using CoMo-MUNIT on the Day $\mapsto$ Timelapse. Fig. 8.5 shows a source day image (center) and our timelapse translations for uniformly sampled ϕ (middle circle). Apart from the appealing translations appearance, notice the network discovered important features like frontal sun (when the sun is close to the horizon), sunset/sunrise, material reflectance (at night), and the stable nighttime appearance. All these features are not in model $M(.)$ though present in target images (outer circle). This advocates the network disentangled model features and translation features. Note also that the top translation in Fig. 8.5 accurately resembles source, assessing that target is attached to source.

Quantitatively, we measure the manifold precision by regressing ϕ with our ϕ-Net$_A$ CNN (cf. Sec. 8.3.4) on real Waymo validation set, and compute the error w.r.t. ground truth elevations. We get a mean error of 19.8° (std 8.56°) when *unsupervised* and 4.05° (std 4.20°) if *supervised*. Even unsupervised, our manifold discovery is acceptable, and opens ways for unsupervised translations where ϕ ground truth would be impractical (*e.g.* rain or snow).

8.4.2.1 Disentangled dimensions

Being MUNIT multi-modal, it is important to assess CoMo-MUNIT properly disentangles ϕ from the style dimension of MUNIT. We do this by sampling ϕ and style. From Fig. 8.6, the latter evolve correctly on different axes, which was expected since ϕ is regulated by model-guided features. Again, using ϕ-Net$_A$, we regress ϕ values for 100 fixed ϕ translations each with 100 different styles, obtaining 1.06° ϕ-variance along the style dimension. This proves the orthogonality of ϕ and style manifolds.

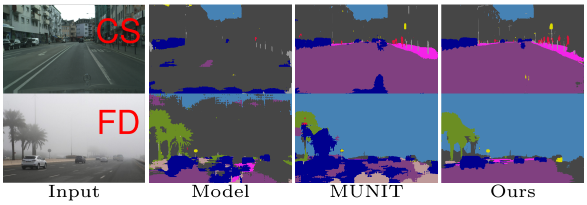
Task	Network	IS↑	CIS↑	LPIPS↑
Day ↦ Timelapse	MUNIT	1.43	1.41	**0.583**
	CoMo-MUNIT	**1.59**	**1.51**	0.580
Syn.$_{\text{clear}}$ ↦ Real$_{\text{clear, foggy}}$	MUNIT	**1.30**	1.02	0.493
	CoMo-MUNIT	**1.30**	**1.05**	**0.515**
iPhone ↦ DSLR	CycleGAN	1.39	n.a.*	0.658
	CoMo-CycleGAN	**1.44**	**1.18**	**0.680**

* CIS is only applicable to multi-modal network.

Table 8.1: GAN metrics proves the benefit of our controllable ϕ generation, leading to on par or better quality/variability.

Translations	CS	FD	Mean
none (source)	10.9	10.1	10.5
Model	19.9	21.5	20.7
MUNIT	38.3	21.8	30.0
CoMo-MUNIT	**43.0**	**23.4**	**33.2**

(a) mIoU metric



(b) Samples

Figure 8.7: Semantic segmentation on clear Cityscapes (CS) (Cordts et al., 2016) and Foggy Driving (FD) (Sakaridis et al., 2018) with PSPNet-50 (Zhao et al., 2017) trained on clear Synthia (*source*), foggy physics *Model*, and Synthetic$_{\text{clear}}$ ↦ Real$_{\text{clear, foggy}}$ of MUNIT or CoMo-MUNIT. Noticeably, we outperform all on both clear (CS) and foggy (FD) dataset.

8.4.3 Translation quality

8.4.3.1 GAN metrics

We measure the quality and variability of all translations task w.r.t. MUNIT and CycleGAN backbones, showcasing in Tab. 8.1 that we always perform better or on par. In the table, IS (Salimans et al., 2016) evaluates image quality and diversity over all the dataset, CIS (Huang et al., 2018b) over multimodal translations, and LPIPS (Zhang et al., 2018a) evaluates absolute diversity only. We conjecture our performance results of the higher degree of control we have, since we control ϕ features in a disentangled manner (*i.e.* extremely increasing variability), while entangled backbones lean towards the easiest translations. The InceptionV3 networks used for IS/CIS evaluation are trained on the source/target classification task. IS is evaluated on all validation set, while for CIS/LPIPS we follow (Huang et al., 2018b) evaluation routine.

8.4.3.2 Semantic segmentation

We measure the effectiveness of Synthetic$_{\text{clear}}$ ↦ Real$_{\text{clear, foggy}}$ translations in Fig. 8.7 by training PSPNet-50 (Zhao et al., 2017) with either MUNIT or CoMo-MUNIT outputs. For comparison, we also train segmentation with

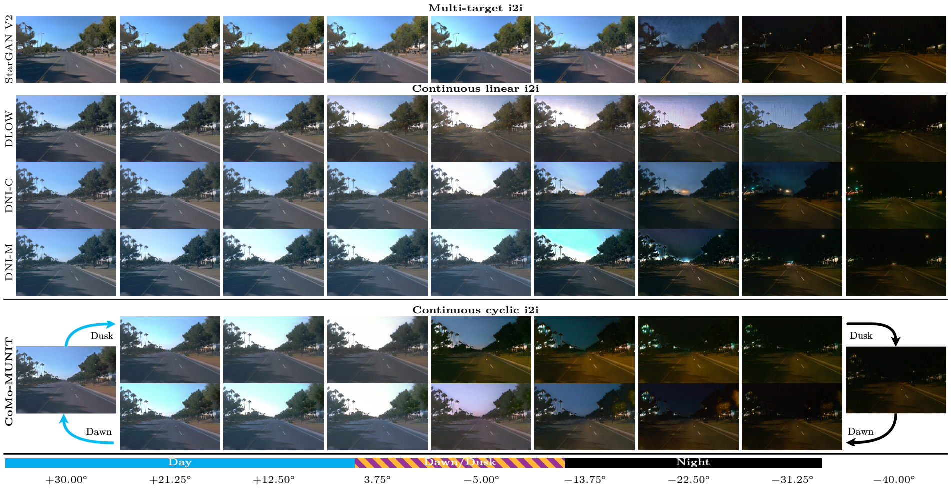


Figure 8.8: Day ↦ Timelapse translations. Baselines output unrealistic translations (*e.g.* DLOW (Gong et al., 2019)) or images with limited variability (StarGAN V2 (Choi et al., 2020)). We adapt MUNIT and CycleGAN using DNI (Wang et al., 2019b) to DNI-MUNIT (DNI-M) and DNI-CycleGAN (DNI-C). Those are the best baselines, though our CoMo-MUNIT (last row) is the only cyclic one, outputs more variable images (*e.g.* at dusk/dawn) and discovered stable night *with less supervision.*

clear *source* Synthia or physics-based foggy *model* (Halder et al., 2019) as for guidance. For MUNIT and CoMo-MUNIT, we employ a multi-modal style-sampling strategy 3 with 5 fixed styles. Additionally, for CoMo-MUNIT and *model* translations that allow it, we sample uniform ϕ. We follow (Zhao et al., 2017) settings and train 150 epochs, using 3498 train images for each setup.

Tab. 8.7a reports the standard mIoU on shared Synthia-Cityscapes classes on real images from the validation set of Cityscapes (Cordts et al., 2016) (CS, 500 images) and Foggy Driving (Sakaridis et al., 2018) (FD, 101 images). While the transformation is subtle, it still reduces the domain shift, since even if *Model* significantly outperforms *source* but we beat all by additional margin of +4.7/+1.6/+3.2. Noticeably, we improve both on clear (CS) and foggy (FD) datasets showing CoMo-MUNIT preserved accurate clear *and* foggy translations. We speculate instead that MUNIT focuses on target dataset fog intensities which are discrete and may differ from FD, while our FIN layer enables continuous representation leading to better generalization. Qualitative evaluation on both datasets in Fig. 8.7b respects mIoU performances.

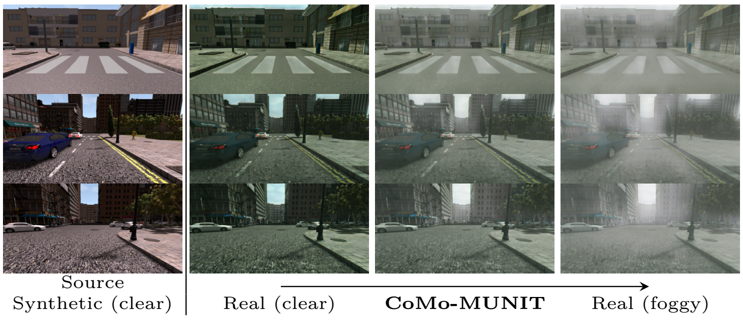


Figure 8.9: Sample Synthetic$_{\text{clear}}$ $\mapsto$ Real$_{\text{clear, foggy}}$ translations with CoMo-MUNIT. Note the complex detached source (Synthia (Ros et al., 2016)) and target (clear/foggy Cityscapes(Cordts et al., 2016; Halder et al., 2019)) setting. Still, clear translations correctly encompass Cityscapes stylistic appearance (notice texture and color).

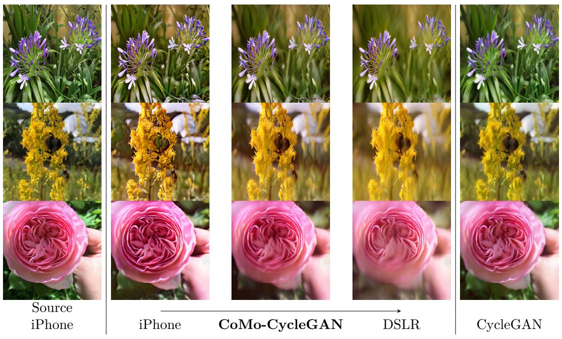


Figure 8.10: CoMo-CycleGAN translations on the iPhone $\mapsto$ DSLR task, using iphone2dslr dataset (Zhu et al., 2017a). Despite naive blur guidance (Eq. 8.15), it learns continuous DSLR depth of field, while CycleGAN (Zhu et al., 2017a) outputs only target translations.

8.4.4 Continuous translation quality

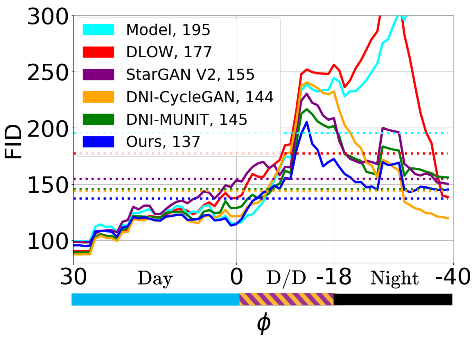
To evaluate the continuity of the translations, we show uniformly spaced ϕ translations for Day $\mapsto$ Timelapse (Fig. 8.8, bottom row), Synthetic $\mapsto$ Real (Fig. 8.9) and iPhone $\mapsto$ DSLR (Fig. 8.10). For all, regardless of the backbone and task, our translations look appealing with our network discovering unique visual features *not* present in the model guidance. This is quite noticeable in DSLR (Fig. 8.10) which learned depth of field despite simple blurring guidance, or in the detached foggy experiment (Fig. 8.9) since translations encompass the desired real appearance with increasing fog.

8.4.4.1 Benchmark evaluation

We evaluate the challenging Day ↦ Timelapse with the literature. This is not trivial since our proposal is to the best of our knowledge the first continuous cyclic GAN. While *some* previous works could be adapted to cyclic translation (*e.g.* DLOW (Gong et al., 2019)) they *all* require intermediate labeled target points. Hence, to achieve a fair comparison compensating data scarcity in Waymo Open, we formulate timelapse as linear {Day, Dusk/Dawn, Night} for all baselines and randomly sample between Dusk or Dawn branch with our cyclic network. Please bear in mind that **all baselines are more supervised than ours** since they use intermediate Dusk/Dawn point while CoMoGAN discovers the manifold from unsupervised target data. We now detail the baselines.
StarGAN V2 (Choi et al., 2020) is a state-of-the-art multi-target i2i architecture learning multiple mapping from the same source point. We train it with official implementation on Day ↦ Dawn/Dusk ↦ Night path and use its style code disentanglement capability to enable continuous i2i.
DLOW (Gong et al., 2019) is continuous by design. We train it with 2 unimodal DLOW Day ↦ Dawn/Dusk and Dawn/Dusk ↦ Night. Note that it can be multi-target, but we already compare with the more recent StarGAN V2.
DNI (Wang et al., 2019b) applies Deep Network Interpolation to interpolate among kernels of finetuned networks for continuous i2i. We adapt 2 baselines DNI-CycleGAN and DNI-MUNIT both trained on Day ↦ Dawn/Dusk ↦ Night.

From Fig. 8.8, baselines (rows 1-4) either exhibit limited variability in interpolated points (StarGAN V2 / DNI) or unrealistic results (*e.g.* DLOW at night). A key limitation is that they rely on (piece-wise) linear interpolation preventing them from discovering the stationary aspect of night (last 3 columns). Conversely, CoMo-MUNIT (bottom row) translations are both realistic and stationary at night.

We also study the realism of all translations using the Frechet Inception Distance (Heusel et al., 2017b) (FID) to measure features distances between generated images and real ones. For that, we uniformly split the elevations range $[+30°, -40°]$ in 70 overlapping bins of 7° width, and compute each bin FID by comparing 100 translations and ad-hoc real images. We call this ”rolling FID”, plotted in Fig. 8.11a. From the latter, our method outperforms others especially in complex intermediate conditions. Note the baselines performance at precise ”dawn/dusk” center (where they are supervised) and how their FID degrade as they depart toward night (approx. $-18°$). Even if *unsupervised*, our lower FID shows CoMo-MUNIT better learns these complex visual transitions. An alternative accuracy evaluation is proposed with a proxy task, which is an InceptionV3 network trained



(a) Rolling FID

Method	Mean err. ↓	Std ↓
Model	21.12	10.15
DLOW	17.39	9.02
StarGAN V2	15.91	10.00
DNI-CycleGAN	13.84	7.91
DNI-MUNIT	13.80	8.30
CoMo-MUNIT	**9.84**	**7.20**
Real data	3.61	4.52

(b) ϕ regression

Figure 8.11: Evaluation of Day ↦ Timelapse. In a rolling FID (cf. text) shows our method is more effective in the complex dawn/dusk (”D/D”) and night points, translating as lower mean FID (in legend). In b, we rank best on both mean and std error between the input ϕ and the regressed ϕ with an InceptionV3 network (trained on real data).

to regress sun elevation from real images and ϕ ground truths. For each method, we generate 100 images at 100 ϕ locations, and measure the error between the input ϕ and the inference with the InceptionV3. Tab. 8.11b shows we outperform others with a 3.96° margin due to our better mapping.

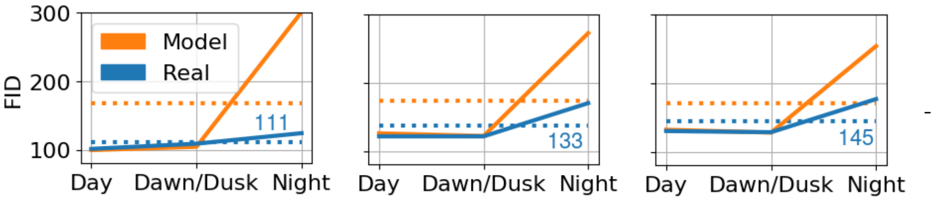
8.4.5 Ablation studies

8.4.5.1 Architectural changes.

We ablate the use of $\mathcal{L}_M$ and $\mathcal{L}_\phi$ by removing either. To evaluate the diversity of Day ↦ Timelapse translations, we sample 10 couples of random $\{\phi_1, \phi_2\}$ for 100 images and evaluate the LPIPS distance among translations pairs. We obtain LPIPS 0.020 *w/o* $\mathcal{L}_M$, 0.044 *w/o* $\mathcal{L}_\phi$, while using both proves best with **0.236**.

8.4.5.2 Disentangled reconstruction.

While we disentangle real domain $Y(\phi)$ and model domain $Y_M(\phi)$ (Fig. 8.2), steerable GANs (Jahanian et al., 2020) instead leverage guidance directly on $Y(\phi)$. To study either benefit, we replace $\mathcal{L}^\phi$ and $\mathcal{L}_M$ with $\mathcal{L}_{edit} = \lambda||y^\phi - \tilde{y}^\phi_M||_1$ as in (Jahanian et al., 2020). Fig. 8.12 shows discrete FIDs, for *ours* and (Jahanian et al., 2020) with $\lambda = 1, 5$, evaluated against real data (blue) or model translations (orange). The plots hold complex but interesting insights. Specifically, low FIDs at Dawn/Dusk infer the model is reliable there, while divergent FIDs at night mean the opposite. With $\lambda = 1$ the i2i lacks guidance and performs poorly, but higher λ increases



(a) Ours (b) Steerable ($\lambda = 1$) (c) Steerable ($\lambda = 5$)

Figure 8.12: FIDs (cf. text) for *ours* (a) and steerable GANs (Jahanian et al., 2020) (b-c). *Ours* has lowest FIDs as it learns to depart from the model. Instead when increasing λ, Steerable GANs (Jahanian et al., 2020) learns to mimic model but FID diverges from real images features.

model mimicking and lower *real* FID. Instead, *ours* is *guided by the model* but learns to depart from it with the discovery of exclusive target features.

8.4.5.3 Model choice.

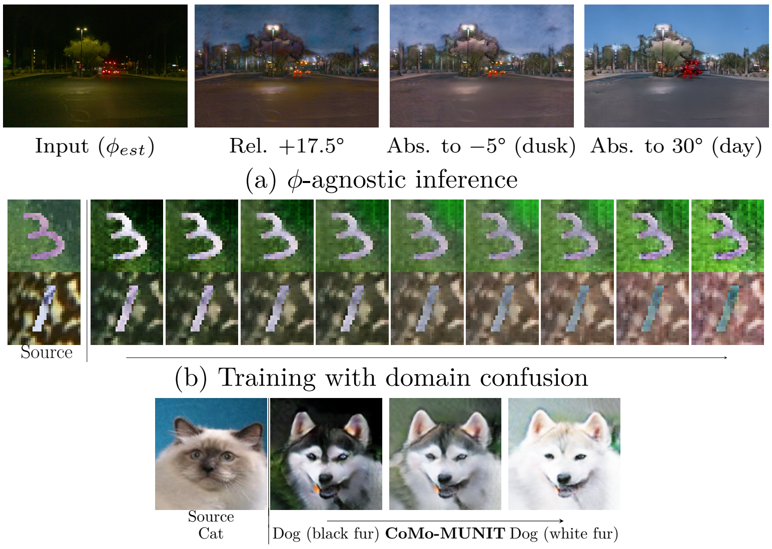
We study the benefit of FIN encoding by swapping linear and cyclic. Comparing with Tab. 8.1, training iPhone $\mapsto$ DSLR with *cyclic FIN* is worse (IS/CIS/LPIPS 1.41/1.20/0.678) and at the cost of a more complex encoding. Training Day $\mapsto$ Timelapse with *linear FIN* performs on par or better (IS/CIS/LPIPS 1.65/1.64/0.579) but *loses dusk/dawn distinction* capability.

8.5 Discussion

While we proved CoMoGAN is beneficial to a variety of continuous translations problems, we now discuss possible extensions and limitations, while positioning our work in the current state-of-the-art.

ϕ-agnostic inference. In all experiments, translation assumes source at ϕ_0, though agnostic inference is of interest. To test this, we trained our method with cycle consistency and shared parameters for $X \mapsto Y$ and $Y \mapsto X$ encoder/decoders (refer to Sec. 8.3.4.1). At inference, we used ϕ-Net$_A$ to estimate ϕ_{est} on input which enabled absolute translation regardless of input (*e.g.* anytime$\mapsto$day) but also relative translation (*e.g.* +5°). Sample results in Fig. 8.13a show exciting results with challenging night input.

Source/Target domains confusion. A limitation of most GANs is the need of source/target splits while *truly unsupervised* GAN could discover a continuous manifold from mixed source/target data (*i.e.* $X \cup Y$ or domains confusion). Interestingly, model-guided GANs allow this *if* the model



(a) ϕ-agnostic inference

(b) Training with domain confusion

(c) Cat ↦ Dog with fur color guidance

Figure 8.13: (a): Training with shared encoder/decoder and using ϕ-Net$_A$ at inference enables relative and absolute ϕ translations. The input is estimated at $\phi_{est} = -33.45°$ (gt $-32.73°$) and shifted with various strategies. (b): CoMo-CycleGAN on MNIST-M (Ganin et al., 2016) trained with *domain confusion*. It shows source (leftmost) and translations along the ϕ dimension. Despite domain confusion, it reorganized the manifold and produced valid translations. In (c), we guide the complex Cat ↦ Dog only with fur color.

does not enforce ϕ input. While there are no physical model for bilateral night ↔ day or foggy ↔ clear, we prove the feasibility on MNIST-M (Ganin et al., 2016) toy tasks, learning *brightness* or *redness*. Fig. 8.13b shows we correctly achieve translation, paving ways for truly unsupervised GAN.

Models and data limitations. Model-guided GAN are unsuitable for some complex scenarios (*e.g.* face-to-face) due to the lack of models, but can guide features as skin tone, etc. as in our experiment Fig. 8.13c on Cat ↦ Dog using fur color guidance. Like (Jahanian et al., 2020), we too experienced that data scarcity affects greatly the manifold discovery and training timelapse without dusk and dawn proves to fail drastically.

On continuous domains. Recently, the community has shown keen interest in representing continuously physics-based conditions. A synthetic dataset with continuous annotation has been published (Sun et al., 2022), while recent efforts were also dedicated to be robust to continuously shifting conditions (Panagiotakopoulos et al., 2022). This advocates for the interest of the problem we tackled, and for the effectiveness of our formalization.

Chapter 9

Conclusions

In this thesis, we proposed novel applications, algorithms, and tasks in the context of image translation. First, we designed domain-informed pipelines with the help of human knowledge in Part I. Second, in Part II, we used physics priors in order to attain physics-informed pipelines which boosted realism and capabilities of generated images. During the thesis, we opened several original research directions. In Chapter 3, we first proposed to decompose domains for interpreting better the domain shift. In Chapter 4, we exploited local differences in images to hallucinate new domains, while in Chapter 5, we introduced multiple novel mechanisms, where the manifold deformation and GERM could be easily reused for other tasks. In Chapter 7 and Chapter 8, we first called for the combination of physical models and image translation networks, that could be leveraged in physics-dependent applications such as medical or astronomical imaging.

Additional considerations must be done for evaluating the efficacy of generated images in mitigating the domain shift in downstream tasks. Even with highly realistic outputs, it is indeed not clearly defined how the image quality transfers to the robustness of trained task networks on inferences on target domain images (Cui et al., 2020). It seems intuitive and confirmed by experimental evidence that image translation helps, thanks to pixel-level alignment (Murez et al., 2018). Still, it is not always true that realism is mandatory, as we observed unrealistic images could benefit tasks such as semantic segmentation (see CycleGAN, Tab. 3.1b). This highlights the need for further research on the topic. Nevertheless, it is worth noting how image translation contributes positively to interpreting better the training transferability to novel domains, and how it enables labels reuse for different testing scenarios.

Additional considerations related to the evaluation of generative networks also emerged. While we used several metrics for quality assessment, evaluating if images are realistic is still challenging, since many still rely on expensive human-based surveys (Huang et al., 2018b; Tremblay et al., 2020; Chen et al., 2022). Exploiting weakly-paired real image similarity as in Chapter 5 requires expensive data collection procedures, which are not always possible. Also, unexpected values could be detected by current GAN metrics in presence of out-of-distribution output data, *i.e.* images which are evidently not realistic (see MUNIT in Tab. 3.1a). In an ongoing work — not covered in the manuscript — we aim to decompose and characterize the domain shift using just a couple of image samples from a distribution. This could also be used as an additional metric for i2i.

While we presented multiple applications, it is necessary a last remark on a broader view on adversarial learning and i2i networks. In Chapter 5 we demonstrated that it is possible to use adversarial learning as a signal for obtaining weak semantic associations between elements in very different domains. This brings interesting insights: it enables envisaging unsupervised learning of meaningful representations by leveraging on similarities and differences among images, which appear as promising directions for research in computer vision. Self-supervised learning approaches such as DINO (Caron et al., 2021) exploit similar ideas, while being based on augmentations only.

Finally, let us reflect on the current state-of-the-art of image translation and how it could be further improved. Many problems relative to unrealistic outputs are also due to the limited understanding of the scene performed from the translation network, such as wrong mappings, and trivial modifications of textures that could be easily spotted by human beings. With a broader view of the current developments in image generation (Ramesh et al., 2022; Saharia et al., 2022b), it appears evident how image translation itself could benefit from large-scale textual priors, that could be exploited to mitigate mistakes due to limited understanding of target scenes, and to increase style disentanglement thanks to zero-shot capabilities. Moreover, understanding the scene in 3D would bring enormous benefits both in quality and consistency of the proposed translation, and it could be combined with recent advancements in 3D representations such as NeRF (Mildenhall et al., 2020). Finally, diffusion models seem to provide a dramatic boost in performance when it comes to image translation (Saharia et al., 2022a).

Ultimately, even though in the course of the thesis we extensively studied image translation GANs and possible priors, many challenges are still open. In conclusion, it is clear how including humans in the loop has been helpful for increasing the effectiveness and interpretability of training, and how this could be transferred to many unsupervised and self-supervised tasks which still do not exploit simple supervision by data interpretation. This could apply also to many other non-generative tasks, as transfer learning, that are inherently conditioned on domain information. Additionally, the evident effectiveness of methods in Part II calls for further investigation of physics priors in deep learning, which is a strong prior of our world often not exploited enough by modern computer vision. In particular, physics could provide significant clues for video analysis, making it possible to understand environments in fully unsupervised manners, thanks to consistency constraints that are always respected in visual observations.

Publications

During the PhD, five conference publications and one journal submission have been produced, here reported in chronological order.

1. Pizzati, F., de Charette, R., Zaccaria, M., and Cerri, P. (2020b). Domain bridge for unpaired image-to-image translation and unsupervised domain adaptation. In *WACV*

2. Pizzati, F., Cerri, P., and de Charette, R. (2020a). Model-based occlusion disentanglement for image-to-image translation. In *ECCV*

3. Pizzati, F., Cerri, P., and de Charette, R. (2021a). CoMoGAN: continuous model-guided image-to-image translation. In *CVPR (oral)*

4. Dell'Eva, A., Pizzati, F., Bertozzi, M., and de Charette, R. (2022). Leveraging local domains for image-to-image translation. In *VISAPP (best paper award)*

5. Pizzati, F., Cerri, P., and de Charette, R. (2021b). Physics-informed guided disentanglement in generative networks. *TPAMI submission*

6. Pizzati, F., Lalonde, J.-F., and de Charette, R. (2022). ManiFest: Manifold Deformation for Few-shot Image Translation. In *ECCV*

Bibliography

(2007). Pysolar. https://github.com/pingswept/pysolar. 105

(2017). Rain drops on screen. https://www.shadertoy.com/view/ldSBWW. 79

Alletto, S., Carlin, C., Rigazio, L., Ishii, Y., and Tsukizawa, S. (2019). Adherent raindrop removal with self-supervised attention maps and spatio-temporal generative adversarial networks. In *ICCV Workshops*. 68, 69, 79

Arar, M., Ginger, Y., Danon, D., Bermano, A. H., and Cohen-Or, D. (2020). Unsupervised multi-modal image registration via geometry preserving image-to-image translation. In *CVPR*. 33

Baek, K., Choi, Y., Uh, Y., Yoo, J., and Shim, H. (2021). Rethinking the truly unsupervised image-to-image translation. In *ICCV*. 13

Barnes, C., Shechtman, E., Finkelstein, A., and Goldman, D. B. (2009). Patchmatch: A randomized correspondence algorithm for structural image editing. *TOG*. 8

Barnum, P. C., Narasimhan, S., and Kanade, T. (2010). Analysis of rain and snow in frequency space. *IJCV*. 69

Barratt, S. T. and Sharma, R. (2018). A note on the inception score. In *ICML Workshops*. 27

Bhattacharjee, D., Kim, S., Vizier, G., and Salzmann, M. (2020). Dunit: Detection-based unsupervised image-to-image translation. In *CVPR*. 33

Biasetton, M., Michieli, U., Agresti, G., and Zanuttigh, P. (2019). Unsupervised domain adaptation for semantic segmentation of urban scenes. In *CVPR Workshops*. 17

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American statistical Association*. 37

Bruls, T., Porav, H., Kunze, L., and Newman, P. (2019). Generating all the roads to rome: Road layout randomization for improved road marking segmentation. In *ITSC*. 32, 33

Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., and Beijbom, O. (2020). nuscenes: A multimodal dataset for autonomous driving. In *CVPR*. 76, 78, 81, 93

Cao, J., Hou, L., Yang, M.-H., He, R., and Sun, Z. (2021). Remix: Towards image-to-image translation with limited data. In *CVPR*. 47

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. (2021). Emerging properties in self-supervised vision transformers. In *ICCV*. 116

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*. 42

Chen, W.-Y., Liu, Y.-C., Kira, Z., Wang, Y.-C. F., and Huang, J.-B. (2019a). A closer look at few-shot classification. In *ICLR*. 49

Chen, X. and Gupta, A. (2015). Webly supervised learning of convolutional networks. In *ICCV*. 17

Chen, X., Shrivastava, A., and Gupta, A. (2013). Neil: Extracting visual knowledge from web data. In *ICCV*. 17

Chen, Y.-C., Xu, X., Tian, Z., and Jia, J. (2019b). Homomorphic latent space interpolation for unpaired image-to-image translation. In *CVPR*. 99

Chen, Y.-J., Cheng, S.-I., Chiu, W.-C., Tseng, H.-Y., and Lee, H.-Y. (2022). Vector quantized image-to-image translation. In *ECCV*. 116

Cherian, A. and Sullivan, A. (2019). Sem-gan: Semantically-consistent image-to-image translation. In *WACV*. 33

Choi, Y., Choi, M., Kim, M., Ha, J.-W., Kim, S., and Choo, J. (2018). Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*. 9

Choi, Y., Uh, Y., Yoo, J., and Ha, J.-W. (2020). Stargan v2: Diverse image synthesis for multiple domains. In *CVPR*. 9, 49, 52, 98, 99, 109, 111

Cord, A. and Aubert, D. (2011). Towards rain detection through use of in-vehicle multipurpose cameras. In *IV*. 42, 79

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *CVPR*. 8, 24, 37, 39, 52, 56, 76, 79, 84, 104, 106, 108, 109, 110

Cross, E. S., Hamilton, A. F. d. C., and Grafton, S. T. (2006). Building a motor simulation de novo: observation of dance by dancers. *Neuroimage*. 64

Csurka, G. (2017). *A Comprehensive Survey on Domain Adaptation for Visual Applications*. 8

Cui, S., Wang, S., Zhuo, J., Su, C., Huang, Q., and Tian, Q. (2020). Gradually vanishing bridge for adversarial domain adaptation. In *CVPR*. 115

Dai, D., Sakaridis, C., Hecker, S., and Van Gool, L. (2020). Curriculum model adaptation with synthetic and real data for semantic foggy scene understanding. *IJCV*. 18

Dell'Eva, A., Pizzati, F., Bertozzi, M., and de Charette, R. (2022). Leveraging local domains for image-to-image translation. In *VISAPP (best paper award)*.

Deng, C., Ji, X., Rainey, C., Zhang, J., and Lu, W. (2020). Integrating machine learning with human knowledge. *Iscience*. 13

Divvala, S. K., Farhadi, A., and Guestrin, C. (2014). Learning everything about anything: Webly-supervised visual concept learning. In *CVPR*. 17

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*. 63

Efros, A. A. and Leung, T. K. (1999). Texture synthesis by non-parametric sampling. In *ICCV*. 8

Endo, Y. and Kanamori, Y. (2021). Few-shot semantic image synthesis using stylegan prior. *CoRR*. 47

Fischer, J., Mikhael, J. G., Tenenbaum, J. B., and Kanwisher, N. (2016). Functional neuroanatomy of intuitive physical inference. *PNAS*. 63

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *JMLR*. 114

Garg, K. and Nayar, S. K. (2006). Photorealistic rendering of rain streaks. In *SIGGRAPH*. 69, 79, 80

Gatys, L. A., Ecker, A. S., and Bethge, M. (2016). Image style transfer using convolutional neural networks. In *CVPR*. 48

Goetschalckx, L., Andonian, A., Oliva, A., and Isola, P. (2019). Ganalyze: Toward visual definitions of cognitive image properties. In *ICCV*. 99

Gong, R., Dai, D., Chen, Y., Li, W., and Van Gool, L. (2021). Analogical image translation for fog generation. In *AAAI*. 99

Gong, R., Li, W., Chen, Y., and Gool, L. V. (2019). Dlow: Domain flow for adaptation and generalization. In *CVPR*. 36, 39, 40, 98, 99, 101, 109, 111

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *NeurIPS*. 5, 8

Gu, J., Ramamoorthi, R., Belhumeur, P., and Nayar, S. (2009). Removing image artifacts due to dirty camera lenses and thin occluders. In *SIGGRAPH*. 69

Halder, S. S., Lalonde, J.-F., and de Charette, R. (2019). Physics-based rendering for improving robustness to rain. In *ICCV*. 68, 69, 72, 76, 79, 80, 84, 85, 104, 106, 109, 110

Halimeh, J. C. and Roser, M. (2009). Raindrop detection on car windshields using geometric-photometric environment construction and intensity-based correlation. In *IV*. 79

Hansen, N., Müller, S. D., and Koumoutsakos, P. (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary computation*. 72, 74

Hao, Z., You, S., Li, Y., Li, K., and Lu, F. (2019). Learning from synthetic photorealistic raindrop for single image raindrop removal. In *ICCV Workshops*. 69

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *CVPR*. 102

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017a). Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*. 43

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017b). Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*. 53, 111

Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A. A., and Darrell, T. (2017). Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*. 17, 21, 28

Hoffman, J., Wang, D., Yu, F., and Darrell, T. (2016). Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *CoRR*. 17

Hong, S., Yeo, D., Kwak, S., Lee, H., and Han, B. (2017). Weakly supervised semantic segmentation using web-crawled videos. In *CVPR*. 17

Hosek, L. and Wilkie, A. (2012). An analytic model for full spectral skydome radiance. *TOG*. 105

Hoyer, L., Dai, D., and Van Gool, L. (2022). Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In *CVPR*. 18

Huang, H., Li, Z., He, R., Sun, Z., and Tan, T. (2018a). Introvae: Introspective variational autoencoders for photographic image synthesis. In *NeurIPS*. 38

Huang, X. and Belongie, S. (2017). Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*. 48, 51, 100

Huang, X., Liu, M.-Y., Belongie, S., and Kautz, J. (2018b). Multimodal unsupervised image-to-image translation. In *ECCV*. 9, 16, 21, 27, 48, 49, 50, 52, 53, 56, 58, 59, 69, 73, 76, 77, 81, 83, 85, 86, 88, 89, 91, 104, 108, 115

Iscen, A., Tolias, G., Avrithis, Y., and Chum, O. (2019). Label propagation for deep semi-supervised learning. In *CVPR*. 22

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *CVPR*. 9

Jahanian, A., Chai, L., and Isola, P. (2020). On the ”steerability” of generative adversarial networks. In *ICLR*. 99, 112, 113, 114

Jiang, L., Zhang, C., Huang, M., Liu, C., Shi, J., and Loy, C. C. (2020). Tsit: A simple and versatile framework for image-to-image translation. In *ECCV*. 69

Jin, B., Ortiz Segovia, M. V., and Susstrunk, S. (2017). Webly supervised semantic segmentation. In *CVPR*. 17

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. (2021). Highly accurate protein structure prediction with alphafold. *Nature*. 13

Karpatne, A., Watkins, W., Read, J., and Kumar, V. (2017). Physics-guided neural networks (pgnn): An application in lake temperature modeling. *CoRR*. 69

Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., and Aila, T. (2020). Training generative adversarial networks with limited data. In *NeurIPS*. 47

Kim, J., Kim, M., Kang, H., and Lee, K. (2020). U-gat-it: unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. In *ICLR*. 76, 81, 85

Kim, N., Son, T., Lan, C., Zeng, W., and Kwak, S. (2021). Wedge: Web-image assisted domain generalization for semantic segmentation. In *AAAI*. 17

Lample, G., Zeghidour, N., Usunier, N., Bordes, A., Denoyer, L., and Ranzato, M. (2017). Fader networks: Manipulating images by sliding attributes. In *NeurIPS*. 36

Lee, D.-H. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshops*. 17, 18, 21, 22

Lee, H.-Y., Tseng, H.-Y., Mao, Q., Huang, J.-B., Lu, Y.-D., Singh, M., and Yang, M.-H. (2020). Drit++: Diverse image-to-image translation via disentangled representations. *IJCV*. 69, 76, 81, 85

Lee, J., Kim, E., Lee, S., Lee, J., and Yoon, S. (2019). Frame-to-frame aggregation of active regions in web videos for weakly supervised semantic segmentation. In *ICCV*. 17

Lee, S., Son, T., and Kwak, S. (2022). Fifo: Learning fog-invariant features for foggy scene segmentation. In *CVPR*. 18

Lengyel, A., Garg, S., Milford, M., and van Gemert, J. C. (2021). Zero-shot day-night domain adaptation with a physics prior. In *ICCV*. 69

Li, P., Liang, X., Jia, D., and Xing, E. P. (2018a). Semantic-aware grad-gan for virtual-to-real urban scene adaption. *BMVC*. 33

Li, P., Yu, X., and Yang, Y. (2021). Super-resolving cross-domain face miniatures by peeking at one-shot exemplar. In *ICCV*. 46

Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., and Yang, M.-H. (2017). Universal style transfer via feature transforms. In *NeurIPS*. 48

Li, Y., Liu, M.-Y., Li, X., Yang, M.-H., and Kautz, J. (2018b). A closed-form solution to photorealistic image stylization. In *ECCV*. 48

Li, Y., Yuan, L., and Vasconcelos, N. (2019). Bidirectional learning for domain adaptation of semantic segmentation. In *CVPR*. 17, 21, 22, 28

Li, Y., Zhang, R., Lu, J., and Shechtman, E. (2020). Few-shot image generation with elastic weight consolidation. In *NeurIPS*. 47

Lin, C.-T., Wu, Y.-Y., Hsu, P.-H., and Lai, S.-H. (2020a). Multimodal structure-consistent image-to-image translation. In *AAAI*. 33

Lin, J., Chen, Z., Xia, Y., Liu, S., Qin, T., and Luo, J. (2019). Exploring explicit domain supervision for latent space disentanglement in unpaired image-to-image translation. *T-PAMI*. 69

Lin, J., Wang, Y., He, T., and Chen, Z. (2020b). Learning to transfer: Unsupervised meta domain translation. In *AAAI*. 48

Lin, J., Xia, Y., Liu, S., Zhao, S., and Chen, Z. (2021). Zstgan: An adversarial approach for unsupervised zero-shot image-to-image translation. *Neurocomputing*. 47

Liu, M., Ding, Y., Xia, M., Liu, X., Ding, E., Zuo, W., and Wen, S. (2019a). Stgan: A unified selective transfer network for arbitrary image attribute editing. In *CVPR*. 5

Liu, M.-Y., Breuel, T., and Kautz, J. (2017). Unsupervised image-to-image translation networks. In *NeurIPS*. 18, 32, 98

Liu, M.-Y., Huang, X., Mallya, A., Karras, T., Aila, T., Lehtinen, J., and Kautz, J. (2019b). Few-shot unsupervised image-to-image translation. In *ICCV*. 46, 47, 52, 53, 69

Liu, Y., De Nadai, M., Cai, D., Li, H., Alameda-Pineda, X., Sebe, N., and Lepri, B. (2020). Describe what to change: A text-guided unsupervised image-to-image translation approach. In *ACM MM*. 33

Lu, B., Chen, J.-C., and Chellappa, R. (2019). Unsupervised domain-specific deblurring via disentangled representations. In *CVPR*. 69

Luan, F., Paris, S., Shechtman, E., and Bala, K. (2017). Deep photo style transfer. In *CVPR*. 48

Luo, Y., Zheng, L., Guan, T., Yu, J., and Yang, Y. (2019). Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *CVPR*. 17

Ma, L., Jia, X., Georgoulis, S., Tuytelaars, T., and Van Gool, L. (2019). Exemplar guided unsupervised image-to-image translation with semantic consistency. In *ICLR*. 33, 46, 49, 51, 52, 53, 60

Ma, X., Wang, Z., Zhan, Y., Zheng, Y., Wang, Z., Dai, D., and Lin, C.-W. (2022). Both style and fog matter: Cumulative domain adaptation for semantic foggy scene understanding. In *CVPR*. 18, 30

Mao, Q., Lee, H.-Y., Tseng, H.-Y., Huang, J.-B., Ma, S., and Yang, M.-H. (2020). Continuous and diverse image-to-image translation via signed attribute vectors. *IJCV*. 99

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., and Paul Smolley, S. (2017a). Least squares generative adversarial networks. In *ICCV*. 51, 71, 77, 102

Mao, X., Li, Q., Xie, H., Lau, R. Y. K., Wang, Z., and Smolley, S. P. (2017b). Least squares generative adversarial networks. In *ICCV*. 37

Maracani, A., Michieli, U., Toldo, M., and Zanuttigh, P. (2021). Recall: Replay-based continual learning in semantic segmentation. In *ICCV*. 17, 30

Mason, R. A. and Just, M. A. (2016). Neural representations of physics concepts. *Psychological science*. 63, 64

Michieli, U., Biasetton, M., Agresti, G., and Zanuttigh, P. (2019). Adversarial learning and self-teaching techniques for domain adaptation in semantic segmentation. *T-IV*. 17

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. (2020). Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*. Springer. 116

Mo, S., Cho, M., and Shin, J. (2019). Instagan: Instance-aware image-to-image translation. In *ICLR*. 33

Murez, Z., Kolouri, S., Kriegman, D., Ramamoorthi, R., and Kim, K. (2018). Image to image translation for domain adaptation. In *CVPR*. 5, 115

Mușat, V., Fursa, I., Newman, P., Cuzzolin, F., and Bradley, A. (2021). Multi-weather city: Adverse weather stacking for autonomous driving. In *ICCV Workshops*. 69

Narasimhan, S. G. and Nayar, S. K. (2002). Vision and the atmosphere. *IJCV*. 86, 95

Nekrasov, V., Shen, C., and Reid, I. D. (2018). Light-weight refinenet for real-time semantic segmentation. In *BMVC*. 25

Nguyen-Phuoc, T., Li, C., Theis, L., Richardt, C., and Yang, Y.-L. (2019). Hologan: Unsupervised learning of 3d representations from natural images. In *ICCV*. 69

Ojha, U., Li, Y., Lu, J., Efros, A. A., Lee, Y. J., Shechtman, E., and Zhang, R. (2021). Few-shot image generation via cross-domain correspondence. In *CVPR*. 46, 47

Pan, X., Shi, J., Luo, P., Wang, X., and Tang, X. (2017). Spatial as deep: Spatial cnn for traffic scene understanding. In *AAAI*. 41

Panagiotakopoulos, T., Dovesi, P. L., Härenstam-Nielsen, L., and Poggi, M. (2022). Online domain adaptation for semantic segmentation in ever-changing conditions. In *ECCV*. 114

Park, T., Efros, A. A., Zhang, R., and Zhu, J.-Y. (2020). Contrastive learning for unpaired image-to-image translation. In *ECCV*. 69

Park, T., Liu, M.-Y., Wang, T.-C., and Zhu, J.-Y. (2019). Semantic image synthesis with spatially-adaptive normalization. In *CVPR*. 33

Patashnik, O., Danon, D., Zhang, H., and Cohen-Or, D. (2021). Balagan: Cross-modal image translation between imbalanced domains. In *CVPR Workshops*. 47

Pentland, A. P. (1987). A new sense for depth of field. *T-PAMI*. 79

Pizzati, F., Cerri, P., and de Charette, R. (2020a). Model-based occlusion disentanglement for image-to-image translation. In *ECCV*.

Pizzati, F., Cerri, P., and de Charette, R. (2021a). CoMoGAN: continuous model-guided image-to-image translation. In *CVPR (oral)*.

Pizzati, F., Cerri, P., and de Charette, R. (2021b). Physics-informed guided disentanglement in generative networks. *TPAMI submission*.

Pizzati, F., de Charette, R., Zaccaria, M., and Cerri, P. (2020b). Domain bridge for unpaired image-to-image translation and unsupervised domain adaptation. In *WACV*.

Pizzati, F., Lalonde, J.-F., and de Charette, R. (2022). ManiFest: Manifold Deformation for Few-shot Image Translation. In *ECCV*.

Porav, H., Bruls, T., and Newman, P. (2019). I can see clearly now: Image restoration via de-raining. In *ICRA*. 69, 76, 89, 90, 91, 92, 93, 94

Pritch, Y., Kav-Venaki, E., and Peleg, S. (2009). Shift-map image editing. In *ICCV*. 8

Pumarola, A., Agudo, A., Martinez, A. M., Sanfeliu, A., and Moreno-Noguer, F. (2020). Ganimation: One-shot anatomically consistent facial animation. *IJCV*. 99

Qian, S., Jin, L., Rockwell, C., Chen, S., and Fouhey, D. F. (2022). Understanding 3d object articulation in internet videos. In *CVPR*. 17

Raissi, M., Perdikaris, P., and Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*. 64

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. (2022). Hierarchical text-conditional image generation with clip latents. *CoRR*. 116

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., et al. (2019). Deep learning and process understanding for data-driven earth system science. *Nature*. 69

Richter, S. R., Hayder, Z., and Koltun, V. (2017). Playing for benchmarks. In *ICCV*. 52, 53

Romera, E., Bergasa, L. M., Yang, K., Alvarez, J. M., and Barea, R. (2019). Bridging the day and night domain gap for semantic segmentation. In *IV*. 32

Romero, A., Arbeláez, P., Van Gool, L., and Timofte, R. (2019). Smit: Stochastic multi-label image-to-image translation. In *ICCV Workshops*. 99

Ros, G., Sellart, L., Materzynska, J., Vazquez, D., and Lopez, A. M. (2016). The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*. 76, 79, 91, 104, 106, 110

Roser, M. and Geiger, A. (2009). Video-based raindrop detection for improved image registration. In *ICCV Workshops*. 68, 69

Roser, M., Kurz, J., and Geiger, A. (2010). Realistic modeling of water droplets for monocular adherent raindrop recognition using bezier curves. In *ACCV*. 69

Rousseau, P., Jolivet, V., and Ghazanfarpour, D. (2006). Realistic real-time rain rendering. *Computers & Graphics*. 69

Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., and Norouzi, M. (2022a). Palette: Image-to-image diffusion models. In *SIGGRAPH*. 116

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., et al. (2022b). Photorealistic text-to-image diffusion models with deep language understanding. *CoRR*. 116

Saito, K., Saenko, K., and Liu, M.-Y. (2020). Coco-funit: Few-shot unsupervised image translation with a content conditioned style encoder. In *ECCV*. 46, 47, 52, 53, 69

Sakaridis, C., Dai, D., and Gool, L. V. (2021). Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *ICCV*. 8, 37, 38, 39, 42, 52

Sakaridis, C., Dai, D., and Van Gool, L. (2018). Semantic foggy scene understanding with synthetic data. *IJCV*. 18, 69, 72, 108, 109

Sakaridis, C., Dai, D., and Van Gool, L. (2020). Map-guided curriculum domain adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation. *TPAMI*. 18, 52

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training gans. In *NeurIPS*. 27, 83, 108

Sankaranarayanan, S., Balaji, Y., Jain, A., Nam Lim, S., and Chellappa, R. (2018). Learning from synthetic data: Addressing domain shift for semantic segmentation. In *CVPR*. 17

Schutera, M., Hussein, M., Abhau, J., Mikut, R., and Reischl, M. (2020). Night-to-day: Online image-to-image translation for object detection within autonomous driving by night. *IEEE T-IV*. 32

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*. 74

Settles, B. (2009). Active learning literature survey. 13

Shen, T., Lin, G., Shen, C., and Reid, I. (2018). Bootstrapping the performance of webly supervised semantic segmentation. In *CVPR*. 17

Shen, Z., Huang, M., Shi, J., Xue, X., and Huang, T. (2019). Towards instance-level image-to-image translation. In *CVPR*. 27, 33

Singh, K. K., Ojha, U., and Lee, Y. J. (2019). Finegan: Unsupervised hierarchical disentanglement for fine-grained object generation and discovery. In *CVPR*. 69

Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al. (2020). Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*. 104

Sun, T., Segu, M., Postels, J., Wang, Y., Van Gool, L., Schiele, B., Tombari, F., and Yu, F. (2022). Shift: A synthetic driving dataset for continuous multi-task domain adaptation. In *CVPR*. 114

Tang, H., Xu, D., Sebe, N., and Yan, Y. (2019a). Attention-guided generative adversarial networks for unsupervised image-to-image translation. In *International Joint Conference on Neural Networks (IJCNN)*. 76, 81, 85

Tang, H., Xu, D., Yan, Y., Corso, J. J., Torr, P. H., and Sebe, N. (2019b). Multi-channel attention selection gans for guided image-to-image translation. In *CVPR*. 33

Thompson, W. B., Shirley, P., and Ferwerda, J. A. (2002). A spatial post-processing algorithm for images of night scenes. *Journal of Graphics Tools*. 105

Torralba, A. and Efros, A. A. (2011). Unbiased look at dataset bias. In *CVPR*. 7

Torrey, L. and Shavlik, J. (2010). Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI global. 47

Tremblay, M., Halder, S. S., de Charette, R., and Lalonde, J.-F. (2020). Rain rendering for evaluating and improving robustness to bad weather. *IJCV*. 19, 68, 69, 115

Tsai, Y.-H., Hung, W.-C., Schulter, S., Sohn, K., Yang, M.-H., and Chandraker, M. (2018). Learning to adapt structured output space for semantic segmentation. In *CVPR*. 17, 28

TuSimple (2021). Tusimple benchmark. In *https://github.com/TuSimple/tusimple-benchmark*. 35, 37, 38

Ullman, T. D. and Tenenbaum, J. B. (2020). Bayesian models of conceptual development: Learning as building models of the world. 63

Ulyanov, D., Vedaldi, A., and Lempitsky, V. (2017). Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *CVPR*. 100

Upchurch, P., Gardner, J., Pleiss, G., Pless, R., Snavely, N., Bala, K., and Weinberger, K. (2017). Deep feature interpolation for image content changes. In *CVPR*. 99

Uricar, M., Sistu, G., Rashed, H., Vobecky, A., Krizek, P., Burger, F., and Yogamani, S. (2021). Let's get dirty: Gan based data augmentation for soiling and adverse weather classification in autonomous driving. In *WACV*. 76, 80, 88

Varma, G., Subramanian, A., Namboodiri, A., Chandraker, M., and Jawahar, C. V. (2019). Idd: A dataset for exploring problems of autonomous navigation in unconstrained environments. In *WACV*. 8, 35, 37, 38, 40, 41

Von Rueden, L., Mayer, S., Beckh, K., Georgiev, B., Giesselbach, S., Heese, R., Kirsch, B., Pfrommer, J., Pick, A., Ramamurthy, R., et al. (2021). Informed machine learning–a taxonomy and survey of integrating knowledge into learning systems. *IEEE Transactions on Knowledge and Data Engineering*. 13

Wang, J., Lan, C., Liu, C., Ouyang, Y., Qin, T., Lu, W., Chen, Y., Zeng, W., and Yu, P. (2022). Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*. 8

Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., Liu, W., and Xiao, B. (2019a). Deep high-resolution representation learning for visual recognition. *TPAMI*. 56

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., and Catanzaro, B. (2018). High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*. 9

Wang, X., Yu, K., Dong, C., Tang, X., and Loy, C. C. (2019b). Deep network interpolation for continuous imagery effect transition. In *CVPR*. 36, 98, 99, 109, 111

Wang, Y., Khan, S., Gonzalez-Garcia, A., Weijer, J. v. d., and Khan, F. S. (2020). Semi-supervised learning for few-shot image-to-image translation. In *CVPR*. 47

Wang, Y., Mantecon, H. L., Lopez-Fuentes, J. v. d. W., and Raducanu, B. (2021). Transferi2i: Transfer learning for image-to-image translation from small datasets. In *ICCV*. 47

Weber, Y., Jolivet, V., Gilet, G., and Ghazanfarpour, D. (2015). A multi-scale model for rain rendering in real-time. *Computers & Graphics*. 69

Wei, S.-E., Saragih, J., Simon, T., Harley, A. W., Lombardi, S., Perdoch, M., Hypes, A., Wang, D., Badino, H., and Sheikh, Y. (2019). Vr facial animation via multiview image translation. *TOG*. 5

Wigner, E. P. (1990). The unreasonable effectiveness of mathematics in the natural sciences. In *Mathematics and Science*, pages 291–306. World Scientific. 63

Wu, P.-W., Lin, Y.-J., Chang, C.-H., Chang, E. Y., and Liao, S.-W. (2019a). Relgan: Multi-domain image-to-image translation via relative attributes. In *ICCV*. 98, 99

Wu, W., Cao, K., Li, C., Qian, C., and Loy, C. C. (2019b). Transgaga: Geometry-aware unsupervised image-to-image translation. In *CVPR*. 33

Wu, X., Wu, Z., Guo, H., Ju, L., and Wang, S. (2021). Dannet: A one-stage domain adaptation network for unsupervised nighttime semantic segmentation. In *CVPR*. 18

Xia, W., Yang, Y., and Xue, J.-H. (2020). Unsupervised multi-domain multimodal image-to-image translation with explicit domain-constrained disentanglement. *Neural Networks*. 69

Xiao, T., Hong, J., and Ma, J. (2018). Dna-gan: Learning disentangled representations from multi-attribute images. *ICLR Workshops*. 99

Xie, Y., Franz, E., Chu, M., and Thuerey, N. (2018). tempoGAN: A temporally coherent, volumetric gan for super-resolution fluid flow. *SIGGRAPH*. 67, 69

Xu, H., Gao, Y., Yu, F., and Darrell, T. (2017). End-to-end learning of driving models from large-scale video datasets. In *CVPR*. 19, 24

Xu, M., Lee, J., Fuentes, A., Park, D. S., Yang, J., and Yoon, S. (2021). Instance-level image translation with a local discriminator. *IEEE Access*. 33

Xue, T., Rubinstein, M., Liu, C., and Freeman, W. T. (2015). A computational approach for obstruction-free photography. *TOG*. 68

Ye, N., Li, K., Bai, H., Yu, R., Hong, L., Zhou, F., Li, Z., and Zhu, J. (2022). Ood-bench: Quantifying and understanding two dimensions of out-of-distribution generalization. In *CVPR*. 7

Yogamani, S., Hughes, C., Horgan, J., Sistu, G., Varley, P., O'Dea, D., Uricár, M., Milz, S., Simon, M., Amende, K., et al. (2019). Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving. In *ICCV*. 76, 78, 88, 91

Yoo, J., Uh, Y., Chun, S., Kang, B., and Ha, J.-W. (2019). Photorealistic style transfer via wavelet transforms. In *ICCV*. 48, 52, 53

You, S., Tan, R. T., Kawakami, R., Mukaigawa, Y., and Ikeuchi, K. (2015). Adherent raindrop modeling, detectionand removal in video. *T-PAMI*. 69

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., and Huang, T. (2019). Free-form image inpainting with gated convolution. In *ICCV*. 38

Yuan, Y., Chen, X., and Wang, J. (2020). Object-contextual representations for semantic segmentation. In *ECCV*. 42

Zhai, X., Kolesnikov, A., Houlsby, N., and Beyer, L. (2022). Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12104–12113. 7, 63

Zhang, J., Huang, Y., Li, Y., Zhao, W., and Zhang, L. (2019). Multi-attribute transfer via disentangled representation. In *AAAI*. 99

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018a). The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*. 27, 40, 43, 53, 83, 84, 108

Zhang, Y., David, P., and Gong, B. (2017). Curriculum domain adaptation for semantic segmentation of urban scenes. In *ICCV*. 17

Zhang, Y., Qiu, Z., Yao, T., Liu, D., and Mei, T. (2018b). Fully convolutional adaptation networks for semantic segmentation. In *CVPR*. 17

Zhang, Y., Xiang, T., Hospedales, T. M., and Lu, H. (2018c). Deep mutual learning. In *CVPR*. 18

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017). Pyramid scene parsing network. In *CVPR*. 84, 108, 109

Zhao, H., Zhang, Y., Liu, S., Shi, J., Loy, C. C., Lin, D., and Jia, J. (2018). Psanet: Point-wise spatial attention network for scene parsing. In *ECCV*. 42

Zhao, S., Liu, Z., Lin, J., Zhu, J.-Y., and Han, S. (2020). Differentiable augmentation for data-efficient gan training. *NeurIPS*. 47

Zheng, T., Fang, H., Zhang, Y., Tang, W., Yang, Z., Liu, H., and Cai, D. (2021). Resa: Recurrent feature-shift aggregator for lane detection. In *AAAI*. 41

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017a). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*. 9, 26, 27, 32, 35, 39, 56, 76, 80, 81, 85, 98, 104, 105, 110

Zhu, J.-Y., Zhang, R., Pathak, D., Darrell, T., Efros, A. A., Wang, O., and Shechtman, E. (2017b). Toward multimodal image-to-image translation. In *NeurIPS*. 26

Zhu, P., Abdal, R., Qin, Y., and Wonka, P. (2020a). Sean: Image synthesis with semantic region-adaptive normalization. In *CVPR*. 33

Zhu, Z., Xu, Z., You, A., and Bai, X. (2020b). Semantically multi-modal image synthesis. In *CVPR*. 33

Zou, Y., Yu, Z., Vijaya Kumar, B., and Wang, J. (2018). Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*. 18, 22, 30

ABSTRACT

Image-to-image (i2i) translation networks can generate fake images beneficial for many applications in augmented reality, computer graphics, and robotics. However, they require large scale datasets and high contextual understanding to be trained correctly. In this thesis, we propose strategies for solving these problems, improving performances of i2i translation networks by using domain- or physics-related priors. The thesis is divided into two parts. In Part I, we exploit human abstraction capabilities to identify existing relationships in images, thus defining domains that can be leveraged to improve data usage efficiency. We use additional domain-related information to train networks on web-crawled data, hallucinate scenarios unseen during training, and perform few-shot learning. In Part II, we instead rely on physics priors. First, we combine realistic physics-based rendering with generative networks to boost outputs realism and controllability. Then, we exploit naive physical guidance to drive a manifold reorganization, which allowed generating continuous conditions such as timelapses.

KEYWORDS

Image-to-image translation, GAN, domain bias, vision and physics, physics-guided learning, few-shot learning