



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DOTTORATO DI RICERCA IN
COMPUTER SCIENCE AND ENGINEERING

Ciclo 38

Settore Concorsuale: 01/B1 - INFORMATICA

Settore Scientifico Disciplinare: INF/01 - INFORMATICA

DEEP LEARNING FOR METEOROLOGICAL DOWNSCALING: FROM
DETERMINISTIC SUPER-RESOLUTION TO PROBABILISTIC AND MULTI-
VARIABLE MODELS

Presentata da: Fabio Merizzi

Coordinatore Dottorato

Paola Salomoni

Supervisore

Andrea Asperti

Co-supervisore

Gianluigi Zavattaro

Esame finale anno 2026

Abstract

High-resolution meteorological data is essential for understanding and predicting atmospheric processes, yet their production remains limited by the high computational cost of numerical downscaling models. In recent years, deep learning has emerged as a viable and highly efficient alternative, capable of producing high-quality, high-resolution fields at a fraction of the computational expense. This thesis explores the landscape of deep learning approaches for climate downscaling, analyzing their limitations and introducing new architectural designs and methodologies that enhance accuracy, represent uncertainty, and exploit inter-variable relationships.

Alongside establishing a general methodological framework that maps existing neural downscaling approaches, the thesis comprises three original studies that collectively advance the state of data-driven downscaling. The first applies diffusion models to the super-resolution of near-surface wind speed from ERA5 to CERRA, benchmarking a range of architectures and introducing an ensemble diffusion formulation that achieves state-of-the-art accuracy while reproducing fine-scale variability. The second develops a theoretical link between the number of reverse diffusion steps and model variance, enabling calibrated uncertainty control and efficient ensemble generation. This framework is validated against existing ensemble reanalyses and extended to regions lacking ensemble information, positioning diffusion models as practical tools for uncertainty-aware meteorological modeling. The third study introduces a multi-variable, multi-task Vision Transformer that jointly downscales six meteorological variables, demonstrating that shared spatial representations improve both accuracy and efficiency compared to single-variable approaches.

Together, these contributions provide new methodological insights into the use of generative and transformer-based architectures for climate and weather applications. They highlight how deep learning can evolve from merely reproducing existing models to discovering new representations of atmospheric variability and coherence.

Contents

1	Introduction	11
1.1	Motivation: The Need for High-Resolution Meteorological Data	11
1.2	From Numerical to Data-Driven Downscaling	11
1.2.1	Limits of Dynamical Downscaling	11
1.2.2	Emergence of Data-Driven Downscaling	12
1.2.3	Challenges within Deep Learning Downscaling	12
1.3	Thesis Objectives	13
1.4	Methodological Scope and Contributions	13
1.5	Structure of the Thesis	14
1.6	Author Contributions	15
2	Background and Related Work	16
2.1	Meteorological Datasets	16
2.1.1	Observational Datasets	16
2.1.2	Reanalyses	19
2.1.3	Operational NWP forecasts	25
2.1.4	Climate Projections	26
2.2	Deep Learning in Climate Science	27
2.2.1	1990s–2000s: Foundations and Early Neural Nets	29
2.2.2	2010s: Deep Learning Emerges in Climate Science	29
2.2.3	2020s: Scale, Speed, and Probabilistic Models	32
2.3	Evolution of Downscaling Methods	34
2.3.1	Simple Interpolation Methods	35
2.3.2	Conventional downscaling frameworks	35
2.3.3	Non-spatial and MLP-based Approaches	36
2.3.4	CNN-based Approaches	37
2.3.5	GAN-based Approaches	37
2.3.6	Transformer-based Approaches	38
2.3.7	Diffusion-based Approaches	38
3	Methodological Foundations	40
3.1	Problem Definition	40
3.1.1	Mathematical Formulation	41
3.2	Architectural Design Space	42
3.2.1	Low-Resolution upsampling	42
3.2.2	Borders, padding and resolution	43
3.2.3	Temporal independence and sequence modeling	44
3.2.4	Spatial modeling	44
3.2.5	Deterministic and generative outputs	45
3.2.6	Single Task, Multi-Task and Additional Elements	46
3.2.7	Other Aspects	47
3.3	Fully Connected Networks	48
3.3.1	Backpropagation	49
3.3.2	MLPs for Downscaling	50
3.4	Convolutional Architectures	51
3.4.1	Foundations of Convolution	51
3.4.2	Convolutional design elements	52
3.4.3	Encoder–Decoder and U-Net Style Designs	53
3.5	Transformer-Based Architectures	56
3.5.1	Transformers for Image Super-Resolution	57
3.6	Generative Adversarial Networks (GANs)	59
3.6.1	Conditioning in GANs	60
3.6.2	Limitations of GANs in Practice	60
3.7	Denoising Diffusion Models	61
3.7.1	Mathematical Foundations	61

3.7.2	Practical Variants: DDPM vs DDIM	64
3.7.3	Noise and Signal Rates	66
3.7.4	Design Space of Diffusion Models	68
3.8	Loss Functions and Optimization Objectives	73
3.9	Training Setup and Evaluation Strategy	75
3.10	Libraries and Hardware Requirements	76
3.10.1	Deep Learning Frameworks	77
3.10.2	Distributed and Multi-GPU Training	79
3.10.3	Hardware	80
4	Paper I: Wind speed super-resolution and validation, from ERA5 to CERRA via diffusion models	81
4.1	Introduction	81
4.2	Methodology	82
4.2.1	Data Selection & Preprocessing	82
4.2.2	Diffusion Model Architecture	83
4.2.3	SR framework	83
4.2.4	Denoising and conditioning	84
4.2.5	Ensemble Diffusion	85
4.2.6	Training and Evaluation	85
4.2.7	Model Comparison	87
4.3	Validation	88
4.4	Experiments and Results	89
4.5	Discussion & Conclusion	92
5	Paper II: Controlling ensemble variance in diffusion models, an application for reanalyses downscaling	95
5.1	Introduction	95
5.1.1	Related Works	95
5.1.2	Reversed DDIM formulation	96
5.1.3	Notations	97
5.1.4	Data	97
5.2	Methodology	97
5.2.1	Signal and noise schedules	97
5.2.2	Controlling the variance	98
5.3	Experimental Setting	100
5.3.1	Data selection and pre-processing	100
5.3.2	Training on the full domain	101
5.4	Experiments	102
5.4.1	Quantifying variance	102
5.4.2	Analyzing variance spatially	103
5.4.3	Impact of Δt on performance	105
5.4.4	Demonstrating capabilities on CARRA	105
5.4.5	Comparison with EMOS	107
5.5	Conclusions	109
6	Paper III: Vision Transformers for Multi-Variable Climate Downscaling, emulating Regional Climate Models with a Shared Encoder and Multi-Decoder Architecture	111
6.1	Introduction	111
6.1.1	Data Selection	112
6.1.2	Related Works	113
6.2	Methodology	114
6.2.1	Architectural Backbone: Vision Transformer (ViT)	114
6.2.2	Single-Variable Models	115
6.2.3	Multi-Variable Models	116
6.2.4	Multi-Variable U-net	117
6.3	Experimental Setup	117

6.4	Results	119
6.5	Conclusion	125
7	Synthesis and Discussion	128
7.1	Downscaling Paradigms: Dynamical, Statistical, and Neural	128
7.2	Broader Perspectives on Neural Downscaling	128
7.3	Architectures for Neural Downscaling	129
7.4	Learning Uncertainty from Deterministic Data	131
7.5	The Trade-offs of Multi-Variable Approaches	131
8	Conclusions and Future Work	133
8.1	Summary of the Thesis Contributions	133
8.1.1	Paper I	133
8.1.2	Paper II	134
8.1.3	Paper III	134
8.2	Limitations	135
8.3	Broader Impact on Neural Downscaling	136
8.4	Future Work and Developments	137

List of Figures

1	IGRA V2 Stations. Source: Copernicus Climate Change Service, IGRA V2 dataset overview (CDS).	19
2	Example of the ERA5 global temperature field and mean sea level pressure on January 1, 2023. Source: Copernicus Climate Change Service, ERA5 dataset overview (CDS).	20
3	CERRA European Domain. Source: Copernicus Climate Change Service, CERRA dataset overview (CDS).	22
4	Overview of the CERRA production chain. Lateral boundary conditions are taken from the 10 ERA5-EDA ensemble members for the 10 ensemble members of CERRA-EDA. The CERRA-EDA provides background errors estimates to CERRA, which is driven by ERA5 at the lateral boundary conditions. CERRA-Land is then in turn driven by CERRA. Source: Copernicus Climate Change Service, CERRA dataset overview (CDS).	23
5	The CARRA-West and CARRA-East (Copernicus Arctic Regional Reanalysis) domains. Source: Copernicus Climate Change Service, CARRA dataset overview (CDS). Source: ECMWF, Copernicus Arctic Regional Reanalysis (CARRA) – Added value to the ERA5 global reanalysis.	24
6	T2m air temperature at south Greenland and Iceland 15 February 2014 00 UTC for CARRA (left) and ERA5 (right). Source: ECMWF, Copernicus Arctic Regional Reanalysis (CARRA) – Added value to the ERA5 global reanalysis.	25
7	Super-resolution problem setup. A low-resolution field (LR, left) is fed into a neural model (center), which outputs a super-resolved field (HR, right).	42
8	Super-resolution setup with external upsampling. The low-resolution input (LR) is first upsampled by interpolation (e.g., bilinear, bicubic, or nearest-neighbor), so that the neural model operates on images of the same size as the target high-resolution field (HR). This strategy is simple and efficient, and is widely used to make super-resolution compatible with standard encoder–decoder architectures.	43
9	Sequence-to-one super-resolution setup. Multiple low-resolution fields (LR) at consecutive timesteps ($t = k - 1, k, k + 1$) are provided as input to a neural model, which outputs a single high-resolution field (HR) at the target timestep ($t = k$). This setup exploits temporal context to improve the reconstruction of the target field.	44
10	Non-deterministic super-resolution setup. A low-resolution field (LR) is combined with stochastic noise and processed by the neural model. Instead of a single deterministic output, the model generates multiple high-resolution samples (HR), which can be interpreted as draws from an underlying probability distribution (illustrated here by a Gaussian). This probabilistic formulation enables quantification of uncertainty and exploration of extremes beyond the mean prediction.	46

11	Multi-task super-resolution setup with three fields. At a given timestep, three low-resolution inputs (Field 1, Field 2, Field 3) are provided to the neural model. The model jointly predicts the corresponding high-resolution outputs for each field. This formulation allows the network to exploit cross-variable relationships and share information across tasks, potentially improving accuracy and efficiency compared to training separate models for each variable.	47
12	Illustration of an MLP-based downscaling workflow. A low-resolution field of size (3×4) is flattened into a one-dimensional vector of length 12, which is then passed through a multi-layer perceptron. The network produces an output vector of length 35, which is reshaped into the super-resolved high-resolution field of size (5×7)	50
13	Schematic representation of an encoder–decoder architecture for image-to-image tasks. The low-resolution (LR) input on the left is compressed by the encoder into a compact representation, which is then expanded by the decoder to produce the high-resolution (HR) output on the right.	54
14	Illustration of the basic building blocks used in ResNet-style encoder–decoder architectures. The <i>Down block</i> (green) reduces spatial resolution while increasing feature richness, the <i>Residual block</i> (purple) serves as the fundamental processing unit, and the <i>Up block</i> (orange) restores spatial resolution while refining the representation.	55
15	Schematic of a residual U-Net architecture built with standard blocks. The encoder (green Down blocks) progressively reduces spatial resolution while expanding the feature space. A bottleneck of Residual blocks (purple) processes the compressed representation. The decoder (orange Up blocks) restores spatial resolution, with skip connections linking encoder and decoder stages to recover fine-scale information.	56
16	Structure of a standard transformer block. The input is first normalized and passed through a multi-head self-attention layer, followed by a residual (skip) connection. A second normalization precedes a position-wise feed-forward network, typically a two-layer MLP with GELU activation, and another residual connection. This sequence of normalization, attention, feed-forward processing, and skip connections constitutes the core unit that is stacked to build deep transformer architectures.	57
17	Overview of transformer-based image translation. (a) Patch embedding process. (b) Sequence-to-sequence processing.	58
18	Schematic of a deterministic GAN for super-resolution. The generator receives a low-resolution (LR) input and, through an encoder–decoder architecture, produces a synthetic high-resolution (HR) field. Both the generated HR and the corresponding real HR are passed to the discriminator, composed of a convolutional encoder followed by a multilayer perceptron (MLP), which outputs a probability indicating whether the sample is real or generated.	60
19	Illustration of the reverse diffusion process. Starting from pure Gaussian noise x_T , the model iteratively applies the neural network denoiser $\epsilon_\theta(x_t, t)$ to predict the injected noise and progressively reconstruct cleaner samples. At each step, a portion of noise proportional to $(1 - \alpha_t)$ is reintroduced, until a high-resolution approximation of the original image x_0 is obtained.	62
20	Comparison of common variance-preserving noise and signal rate schedules used in diffusion models. Each panel shows the evolution of the signal rate $s_t = \sqrt{\alpha_t}$ and noise rate $n_t = \sqrt{1 - \alpha_t}$ over normalized diffusion time $t \in [0, 1]$. Although all schedules interpolate between the same endpoints (near-clean at $t = 0$ and near-noise at $t = 1$), their intermediate trajectories differ substantially, affecting the difficulty of the denoising task at each step and the overall efficiency of training and sampling. For all plots the bounds are set to $\log \text{SNR}_{\text{start}} = 20.0$ and $\log \text{SNR}_{\text{end}} = -5.0$; the logistic schedule uses $\kappa = 10$ and $t_0 = 0.5$	68
21	Illustration of a single reverse diffusion step with conditioning. The denoiser receives as input the noisy sample x_t , the conditioning low-resolution field LR, and the variance information α_t (or in a different formulation the timestamp t). It outputs a high-resolution estimate of the added noise, which is then used to reconstruct the denoised HR sample.	73
22	Comparison of wind speed over Italy between ERA5 and CERRA	81

23	Selected area for our study, delimited by the following geographical coordinates: North at 47.75°, South at 35°, East at 18.75°, and West at 6°.	82
24	The downscaling diffusion process utilize as input a set of low resolution images. For predicting the high resolution image at time t_0 the conditioning information include the low resolution images at times t_{-6} , t_{-3} , t_0 , t_{+3} where each step is 3 hours long.	83
25	The denoising U-net employs a conventional architecture, consisting of three downsampling blocks implementing average pooling, followed by three upsampling blocks using bilinear interpolation. These stages are interconnected through skip connections, and each block incorporates residual connections.	84
26	The super-resolution diffusion framework employs a sequence-to-one strategy, where pre-upsampling is achieved via bilinear interpolation. The diffusion process comprises five diffusion steps, using a denoising U-net to progressively denoise the input. A noise scheduler orchestrates this process as reported in Algorithm 3. This procedure is replicated 15 times, generating an ensemble of potential super-resolution outcomes. These outcomes are then amalgamated using a mean operation to produce the final output.	86
27	The reverse diffusion process is a sequential method that involves alternating between denoising steps and the re-injection of noise, progressively refining the generated image.	86
28	Construction process of the dataset used for the validation of ERA5, CERRA and the generated data. IGRA V2 observations in the reference period undergo filtering depending on the reference time and the air pressure value, after which the values selected for an observation at a given time are averaged and become part of the dataset. At this point, the corresponding values of ERA5 (after calculating the wind speed from the two u and v components) and CERRA are concatenated to each observation. The same process is implemented for the values resulting from the various generative models.	88
29	Visual evaluation of super-resolution methods for ten selected samples from the years 2009 and 2020 showcasing different weather conditions, including a Mediterranean cyclone. The comparison contain all tested neural downscaling methods alongside the CERRA model. Visual evaluation reveals that U-net, ViT, GAN and Diffusion methods provide the visually best results, with the diffusion models achieving the higher amount of details. The numerical results reported in Table 12 provide a more effective comparison of the respective performances.	90
30	Graph comparison for the SSIM metric of the different tested models. SSIM is calculated for each batch sequentially in the testing year of 2020, highlighting the variability of performance at different times of the year.	92
31	Mean image representing the spatial SSIM error for both testing years of 2009 and 2020, showcasing the downscaling performance relative to different geographical areas. The similarity in the results demonstrates that the errors made by the model are consistent with respect to the two years used for the test.	93
32	Comparison of CERRA and CARRA-EAST domains.	98
33	Reverse diffusion signal and noise ratios with cosine scheduling, including the reduced signal rate	99
34	Plotting the global mean variance μ_V of ensemble diffusion for different number of steps against CERRA-EDA	103
35	Comparing the spatial variance of CERRA-EDA with ensemble diffusion at different diffusion steps. The comparison relative to the testing year 2016 and is performed in a 3 monthly manner to highlight seasonal variations.	104
36	Evolution of the mean variance discrepancy (MVD) scores with the number of diffusion steps. The plot visualizes the values reported in Table 15 for the yearly and three-monthly periods (JFM, AMJ, JAS, OND) during 2016. Lower values indicate a closer match between the spatial variance distributions of Ensemble Diffusion and CERRA-EDA. The yearly curve (orange) highlights the global trend, while the seasonal curves illustrate the differing optimal step counts across the annual cycle.	105
37	Mean and variance range at a randomly selected point in the domain, compared between ensemble diffusion (8 reverse diffusion steps) and CERRA-EDA. Data is smoothed using a 30 samples moving average window for readability and covers the entire year of 2016.	106

38	Revealing the spatial variance of ensemble diffusion trained on CARRA-EAST with 8 diffusion steps. The comparison is relative to the testing year 2016 and is performed in a 3 monthly manner to highlight seasonal variations.	107
39	Visualization of a single daily sample for the six studied variables: Surface Temperature, Wind Speed, Geopotential Height, Total Precipitation, Surface Downwelling Shortwave Radiation and Surface Downwelling Longwave Radiation. Each variable is shown for both the high-resolution Regional Climate Model ICTP-RegCM4-6 (top row) and the low-resolution Global Climate Model MPI-ESM-LR (bottom row) re-gridded onto the same European domain.	112
40	Comparison between a single-variable encoder-decoder model and multi-variable models using either a single shared decoder (1E1D) or multiple decoders branching from a shared encoder (1EMD).	114
41	Architecture of the multi-variable single-encoder multi-decoder model (1EMD). Concatenated input variables are divided into patches, embedded with positional encodings, and processed as a sequence by six transformer blocks, each consisting of multi-head self-attention (six heads) followed by an MLP. The output sequence is then reshaped into a spatial grid and passed to separate CNN-based decoders, each composed of three upsampling stages interleaved with residual blocks.	118
42	Comparison of the examined models (Single var ViT, multi var ViT 1E1D, Multi var ViT 1EMD), target RCM, and bilinearly interpolated GCM for a sample of the test set. For each sample, each row contains Geopotential, Wind speed, Temperature, Total Precipitation, Surface Downwelling Shortwave Radiation and Surface Downwelling Longwave Radiation for the same timestamp.	122
43	Visualization of cross-variable leakage in the 1E1D and multi-variable U-Net models for total precipitation and zg500. Four samples are shown, two for precipitation and two for geopotential height. Using a high contrast colormap, we can clearly see high-frequency artefacts in the multi-variable U-Net and 1E1D ViT, with faint outlines resembling European geography. This suggests contamination from temperature and wind fields. These outlines are absent from the original data, the single-variable model, and the multi-decoder 1EMD model.	123
44	Mean absolute error maps for bilinear interpolation (top rows) and the 1EMD multi-variable ViT (bottom rows). The top panel shows zg500, wind speed, and temperature; the bottom panel shows precipitation, shortwave downwelling radiation, and longwave downwelling radiation. Errors are computed over the full test set, with per-variable color scales.	124
45	Inference time comparison across model architectures. The Total inference time (dark blue bars) and per-variable inference time (light blue bars) are shown for four model configurations. All models were evaluated on identical hardware (NVIDIA A100), with the same input data pipeline. The measurements correspond to the inference time required to process 10 years of daily testing data. For the multi-variable models, which predict six variables simultaneously, the per-variable inference time was calculated by dividing the total inference time the number of variables.	126
46	Monthly evolution of the mean squared error (MSE) for the six downscaled variables, computed over ten climatological years and averaged over all spatial grid points.	126
47	Empirical probability density functions (PDFs) of the six downscaled variables computed over the test set and aggregated across all spatial grid points. Histograms are normalized such that the total area equals one and are computed using common bin edges for the reference and downscaled datasets.	127

List of Tables

1	Comparison of the main categories of meteorological datasets.	16
2	Selected regional climate modelling datasets.	28

3	Selected influential neural network studies in meteorology during the 1990s and 2000s, including early applications to statistical downscaling. BPN = backpropagation network; QPF = quantitative precipitation forecast; PCA = principal component analysis; RBFN = radial basis function network.	30
4	Selected influential neural network studies in meteorology during the 2010s, illustrating the shift from point-to-point MLP models to field-to-field deep learning architectures. MLP = Multi-layer Perceptron; GCM/RCM = Global/Regional Climate Model; CNN = Convolutional Neural Network; GNN = Graph Neural Network.	31
5	Selected influential neural network studies in meteorology during the 2020s, including thesis contributions. The decade saw a progression from benchmarks and feasibility to operationally competitive forecasts, probabilistic diffusion ensembles, and multi-domain foundation models.	33
6	Concise summary of convolutional design elements.	53
7	Variance-preserving (VP) noise and signal rate schedules in diffusion models. All schedules fit the forward process $x_t = \sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\epsilon$, with $\alpha_t + (1 - \alpha_t) = 1$	69
8	Non-variance-preserving noise and signal rate schedules. These approaches depart from the standard VP assumption $s_t^2 + n_t^2 = 1$, either by exploding variance (VE), learning dataset-specific schedules (CRS), disentangling total variance from SNR, or making schedules adaptive.	70
9	Overview of common design choices for training and evaluation	76
10	Comparison of TensorFlow/Keras and PyTorch	78
11	Comparative analyses of the tested models, including the number of parameters (millions) and the execution time (seconds) for the computation of one year of data.	87
12	Results comparison for multiple models on wind downscaling for the years 2009 and 2020.	91
13	Validation of the results obtained compared to the observations present in the IGRA V2 dataset. The comparison takes into consideration the Mean Absolute Error (MAE), measured in meters per second, and the Mean Squared Error (MSE) calculated on the in situ observations and their corresponding values generated by the models or present in ERA5 or CERRA.	91
14	Performance of ensemble diffusion against bilinear interpolation and U-Net for the years 2011-2020.	102
15	Mean variance discrepancy (MVD) scores computed between the spatial mean variance maps of Ensemble Diffusion and CERRA-EDA for each three-monthly period (JFM, AMJ, JAS, OND) and each timestamp for 2016. Lower scores indicate higher similarity between the variance distributions of the two ensembles.	105
16	Table comparing the MSE performance of the diffusion model at different numbers of reverse diffusion steps. The MSE is calculated between the mean of the 10 generated ensembles and the CERRA-EDA control run. The results are reported 3-monthly for the year 2016, along with the yearly mean across all seasons.	106
17	Total ensemble variance (μ_V) for Ensemble Diffusion at increasing numbers of reverse diffusion steps, compared with EMOS, Local EMOS, and the reference CERRA-EDA variance. Variance increases monotonically with diffusion steps and converges toward the reference.	108
18	Mean variance discrepancy (MVD) scores computed between the spatial mean variance maps of the 8-step Ensemble Diffusion model, EMOS, and Local EMOS for each three-monthly period (JFM, AMJ, JAS, OND) and the full year 2016. Lower scores indicate higher similarity between the variance distributions of the compared ensembles.	109
19	Model performance comparison for downscaling from MPI-ESM-LR to RegCM4-6 for each variable, mean model performance across two independent runs. Best values for each variable are underlined, the results are computed for variables normalized between 0 and 1. Best results (including ties) are underlined.	120
20	Spatial and temporal Pearson correlations between model predictions and RegCM4-6 targets for each variable. Spatial correlations are computed for each day and summarized as mean $\pm$ standard deviation over time. Temporal correlations are computed at each grid point and summarized as mean $\pm$ standard deviation over space. Best values (including ties) are underlined.	121

21	Computational cost and model complexity for the three examined ViT architectures. Execution time is measured over the entire 10-year test dataset. Training time corresponds to 400 training epochs under identical hardware and hyperparameter settings. .	125
22	Overview of main neural architectures applied to meteorological downscaling, with key characteristics, strengths, limitations, and typical use cases.	130

1 Introduction

This chapter introduces the motivation and scope of the thesis. It discusses how traditional numerical approaches, while physically sound, face computational and practical limitations, and how deep learning methods are increasingly emerging as efficient and flexible alternatives. In this context, the chapter highlights the open challenges and research directions that motivate the development of new data-driven downscaling strategies presented in this work.

1.1 Motivation: The Need for High-Resolution Meteorological Data

The atmosphere evolves through interactions between processes occurring at different spatial and temporal scales. At the global level, the circulation of the atmosphere is relatively well understood and predictable. However, the limits of weather predictability arise from small-scale variations that are imperfectly represented in numerical models [Pal00]. Minor differences in local conditions, often below the spatial resolution of the model, can amplify and lead to divergent system evolutions over time [BTB15]. These limitations are not primarily due to a lack of understanding of atmospheric physics, but to the inability to represent fine-scale variability with sufficient accuracy. Consequently, the realism and predictive skill of meteorological simulations depend critically on the spatial resolution of the underlying data.

Beyond the necessity of high-resolution data for improving the predictability of the atmospheric system itself, fine-scale information is also essential for a wide range of applied domains. High-resolution datasets support diverse activities such as renewable energy forecasting, hydrological modeling, agriculture, disaster risk management, and urban planning, among many others. For example, the accurate estimation of near-surface wind fields is crucial for the design and operation of wind farms, while detailed temperature and precipitation patterns determine the reliability of flood prediction and water resource assessments. In all these contexts, the availability of spatially refined meteorological data enables decisions that are both safer and more efficient.

Despite their importance, the majority of global climate model outputs remain limited to spatial resolutions of tens or even hundreds of kilometers [FMA⁺14, EBM⁺16a]. This coarse representation smooths out local features such as topographic effects, coastlines, and land-use heterogeneity, leading to systematic biases and an underestimation of extremes. Consequently, the predictive skill and physical realism of downstream applications are constrained by the limited resolution of their input data.

Bridging this scale gap is one of the central challenges in modern meteorology and climate science. Dynamical downscaling approaches, which nest high-resolution regional models within global drivers, offer physically consistent solutions but remain computationally expensive, with only a limited number of institutions and resources capable of sustaining them [Gio19]. In recent years, machine learning-based downscaling methods have emerged as a promising alternative for enhancing spatial resolution efficiently [SBG⁺21]. These data-driven approaches can generate fine-scale meteorological fields while preserving the large-scale consistency of their coarse inputs.

1.2 From Numerical to Data-Driven Downscaling

Having established the importance of high-resolution meteorological data, the following section examines how such detail can be achieved in practice, tracing the evolution from traditional numerical downscaling to modern data-driven approaches.

1.2.1 Limits of Dynamical Downscaling

The principal limitation of dynamical downscaling lies in its computational expense. High-resolution regional models must integrate the full set of atmospheric equations over millions of grid cells for many thousands of time steps. The computational cost increases sharply with resolution: a simulation at 5 km grid spacing over a continental domain and multi-decadal period requires petascale resources and months of computation on modern supercomputers [PBG09, PLF⁺15, GLM⁺23]. As a result, the production of high-resolution regional climate data is restricted to a limited number of research centers with access to large-scale computing infrastructure and domain expertise.

The scarcity of computational resources has direct implications for data availability and coverage. The total volume of high-resolution data remains modest compared with global model outputs [Gio19],

and existing datasets are often delayed relative to real time because of the long turnaround required for simulation and post-processing [MAC24a]. Most regional simulations have been conducted over well-observed and well-resourced regions such as Europe, North America, and East Asia, while large portions of the Global South remain underrepresented [NJG+12]. The same computational constraints also limit ensemble generation, which is essential for representing internal variability and quantifying forecast uncertainty [KBR+17]. When ensemble products exist, they typically operate at reduced spatial or temporal resolution to remain computationally feasible [GGJ15].

These structural limitations constrain the diversity, temporal currency, and uncertainty characterization of available high-resolution climate datasets, motivating the search for complementary data-driven approaches that can reproduce fine-scale information more efficiently.

1.2.2 Emergence of Data-Driven Downscaling

The computational and practical limitations of dynamical downscaling have motivated the development of data-driven methods that not only reduce computational cost but also aim to enhance the accuracy, consistency, and flexibility of high-resolution climate modeling [RCVS+19]. Once trained, data-driven models can emulate the behavior of complex numerical systems at a fraction of the computational expense, enabling near-real-time inference and large-scale experimentation. This efficiency expands the scope of feasible studies, allowing higher spatial and temporal resolution, broader ensemble generation, and more frequent updates than traditional approaches [RDK+22]. It also democratizes access to high-resolution information, making it possible for institutions without large supercomputing infrastructure to generate, share, and analyze refined climate data.

In practice, data-driven downscaling models reproduce fine-scale atmospheric structures directly from coarse-resolution inputs by learning empirical relationships from large archives of weather data, such as sensor data, reanalyses or simulations. Most existing implementations build upon advances in computer vision, where analogous tasks such as image super-resolution and translation have been extensively studied [DLHT15]. Through supervised learning, a neural network is trained to map low-resolution meteorological fields to their high-resolution counterparts, effectively capturing spatial patterns and local dependencies that would otherwise require explicit numerical integration. In doing so, these models approximate the physical system’s behavior through data-driven representations rather than through direct numerical solution.

Recent advances in deep learning, particularly convolutional, transformer-based, and generative architectures, have demonstrated the ability to produce high quality high-resolution fields that rival existing dynamical downscaling models [VKG+17, PSH+22, SDR+24a, PCMG24]. Their computational efficiency enables large-scale inference, high-frequency output generation, and ensemble creation at negligible marginal cost, addressing several of the constraints associated with numerical downscaling.

Despite these advances, the field remains in a formative stage. The range of deep learning techniques applicable to weather and climate modeling is far from fully understood, and no consensus has yet emerged on optimal architectures or training strategies. Most operational systems continue to rely on traditional numerical formulations, while research in data-driven modeling remains largely experimental, with new methods and paradigms appearing in rapid succession [RLR+21].

1.2.3 Challenges within Deep Learning Downscaling

Despite the rapid progress of data-driven approaches, several fundamental challenges remain open. The first concerns the diversity of architectures. A wide variety of neural designs, including convolutional, transformer-based, and generative models, have shown promising results, yet their relative merits remain context dependent and poorly understood [BKM+25, BXZ+22, CZZ+23]. The field continues to evolve rapidly, with new architectures frequently introduced and compared under heterogeneous experimental settings, which complicates fair assessment and reproducibility.

Another open issue is the representation of uncertainty. Most neural networks produce deterministic outputs that tend to underestimate natural variability, while probabilistic and generative formulations are still in early stages of adoption [ZCL+24, PSGA+23]. Quantifying model confidence and expressing ensemble-like diversity remain active areas of research and are essential for producing uncertainty-aware predictions that can be used in climate applications.

Ensuring multi-variable consistency also remains a significant challenge [CCDP21, SBG+21]. Many architectures originally developed for natural images process each meteorological field independently,

making it difficult to preserve physical relationships among variables such as wind, temperature, and pressure. Designing shared or coupled representations capable of maintaining cross-variable coherence is an ongoing research direction and a necessary step toward physically consistent modeling.

Generalization beyond the spatiotemporal and climatic conditions represented in the training data is another persistent difficulty. Biases and sampling limitations in reanalysis or observational datasets can propagate into the learned relationships, leading to degraded performance in unseen regions, extreme events, or future climate scenarios. Robust training and evaluation strategies are required to ensure that models can adapt to changing boundary conditions and evolving climate regimes.

Finally, the absence of explicit physical constraints can lead to outputs that violate conservation laws or large-scale balances [CDCG⁺22]. Integrating physical knowledge into neural architectures or loss functions, through hybrid models, physics-informed learning, or constrained optimization, represents one of the most promising directions for improving robustness and interpretability.

Addressing these challenges is crucial to establish data-driven downscaling as a reliable, transparent, and physically consistent component of the climate modeling toolbox.

1.3 Thesis Objectives

The challenges outlined above highlight the need for systematic investigations into how modern machine learning architectures can bridge the scale gap between coarse and high-resolution meteorological data. This thesis responds to that need by formulating a coherent research program centered on data-driven downscaling, with particular emphasis on the reconstruction of fine-scale wind patterns, the representation of predictive uncertainty, and the integration of multi-variable physical relationships.

The overarching objective of this thesis is to advance the development and understanding of machine learning approaches for meteorological downscaling. Specifically, it aims to design, evaluate, and interpret neural architectures capable of enhancing the spatial resolution, realism, and uncertainty quantification of meteorological fields derived from reanalysis and climate model data.

- **Objective 1 : Theoretical and methodological synthesis:** Provide a unified perspective on deep-learning-based meteorological downscaling by analyzing current practices, datasets, theoretical foundations, and model architectures, with particular focus on the formulation and design of diffusion models.
- **Objective 2 : Benchmarking neural architectures:** Assess and compare major neural model families for meteorological downscaling, establishing reproducible baselines and evaluating their performance against reference observations.
- **Objective 3 : Reanalysis completion and enrichment:** Develop learned downscaling approaches to complete and enrich reanalysis datasets, extending their temporal coverage, refining their temporal resolution, and producing ensemble-like realizations for uncertainty-aware applications.
- **Objective 4 : Uncertainty modeling and variance calibration:** Investigate how the number of reverse diffusion steps influences the predictive variance of diffusion models, and develop methods to use this relationship for controlling and calibrating uncertainty in diffusion-based meteorological downscaling.
- **Objective 5 : Multi-variable representation learning:** Explore multi-variable and multi-task learning strategies for meteorological downscaling, assessing how shared spatial representations affect model accuracy, efficiency, and physical consistency.

Collectively, these objectives delineate the scope of the thesis, linking theoretical synthesis, model development, and applied evaluation. The next section outlines the methodological framework and experimental design through which these objectives are addressed.

1.4 Methodological Scope and Contributions

This thesis is situated within the field of data-driven meteorological downscaling. It focuses on supervised deep-learning approaches that learn mappings from coarse to high-resolution atmospheric fields using paired data from reanalysis and climate simulations. The emphasis is on the methodological

aspects of spatial refinement, uncertainty representation, and multi-variable consistency, rather than on operational forecasting or physical model coupling.

Within this domain, the thesis explores three complementary methodological directions. The first concerns the comparative evaluation of major neural architectures, including convolutional, transformer-based, and diffusion models, under a unified experimental framework. The second extends diffusion models beyond deterministic generation toward probabilistic ensemble modeling, introducing explicit mechanisms and theoretical aspects for variance control and uncertainty calibration. The third investigates multi-variable and multi-task learning strategies that share spatial representations across atmospheric variables to improve efficiency and physical coherence.

Together, these methodological lines define the scope and contributions of the thesis. They provide a consistent experimental foundation for comparing neural architectures, introduce a principled framework for variance calibration in diffusion models, and demonstrate how shared representations can enhance the accuracy and interpretability of multi-variable downscaling.

1.5 Structure of the Thesis

The thesis is organized into eight chapters, each addressing a specific aspect of the research. Section 1 introduces the motivation behind this work, emphasizing the need for high-resolution meteorological data and the limitations of existing numerical downscaling techniques. It outlines the emergence of deep learning in climate and weather modeling, and discusses the main challenges and open questions surrounding data-driven approaches. Finally, it formulates the main objectives of the thesis and summarizes its methodological scope and contributions. Section 2 reviews the background and related literature. It first presents the main categories of meteorological datasets, including observations, re-analyses, operational forecasts, and climate projections. It then traces the evolution of deep learning in climate science and the progression of downscaling methods, from statistical and interpolation-based approaches to recent neural and generative frameworks. Section 3 establishes the methodological foundations shared across the three studies presented in this thesis. It formalizes the downscaling problem and its possible formulations, and provides an in-depth analysis of the principal architectural families, including MLPs, CNNs, Transformers, GANs, and diffusion models, covering their mathematical principles, structural design choices, and training mechanisms. It also details the corresponding loss functions, optimization procedures, and evaluation strategies. Section 4, Section 5, and Section 6 present the three original research papers that constitute the core of the thesis. The papers have been adapted and harmonized for style, notation, and continuity so as to form a cohesive methodological and conceptual narrative.

- *Paper I* introduces a diffusion-based framework for wind speed super-resolution from ERA5 to CERRA. It presents a comparative evaluation of competing neural architectures, validates the results against in-situ observations, and applies the trained model to fill temporal gaps in existing high-resolution datasets.
- *Paper II* extends the diffusion-based framework by establishing a theoretical link between the number of reverse diffusion steps and the predictive variance of the model. Building on this finding, it introduces a method for variance control and uncertainty calibration, applies it to re-analysis downscaling, and evaluates the resulting ensembles against known uncertainty references and datasets lacking ensemble information.
- *Paper III* investigates a multi-variable, multi-task Vision Transformer (ViT) architecture designed to emulate regional climate models. The study leverages shared spatial representations across variables to improve efficiency and physical coherence, and compares its performance with single-variable models and alternative multi-variable formulations.

Section 7 synthesizes the findings of the three papers, positioning them within the broader context of neural downscaling. It discusses emerging paradigms, architectural trade-offs, and the challenges of learning uncertainty and multi-variable relationships from deterministic data. Finally, Section 8 concludes the thesis by summarizing the main contributions, outlining current limitations, discussing the broader scientific and practical implications, and proposing future directions for research and development.

1.6 Author Contributions

This thesis is based on three research papers. The specific contributions of the candidate (FM) and co-authors Andrea Asperti (AA), Davide Evangelista (DE), Harilaos Loukos (HL) and Samuele Colamonaco (SC) are detailed below.

- *Paper I* FM and AA jointly designed the neural architectures and the overall methodological framework. FM prepared the datasets, implemented the models, conducted the training and benchmarking experiments, and performed the comparative analyses. SC conducted the validation experiments against independent observational datasets. FM and SC wrote the manuscript, with revisions and feedback from all authors.
- *Paper II* FM identified and formulated the variance-control property linking the number of reverse diffusion steps to predictive variance. FM designed the experimental framework, implemented the models, prepared the datasets, and conducted all numerical experiments. DE formulated the mathematical framework, expanded the theoretical analysis, and contributed derivations supporting the proposed methodology. HL provided climate science expertise. FM and DE wrote the manuscript, with feedback from all co-authors.
- *Paper III* FM designed the model architectures, including the shared-encoder multi-decoder formulation. FM implemented the models and conducted all experiments and analyses. HL contributed domain expertise in climate modeling and supported the interpretation of the results. FM wrote the manuscript, with revisions and feedback from co-authors.

2 Background and Related Work

This chapter reviews the scientific background that forms the foundation of this thesis. It begins with an overview of the main categories of meteorological datasets, discussed in depth given their central role in enabling data-driven downscaling. The following sections provide a historical perspective on the use of artificial intelligence in climate science, tracing its evolution from early neural approaches to modern deep learning practices. Finally, the chapter surveys the most relevant techniques for deep learning-based downscaling, outlining current trends and open challenges in the field.

2.1 Meteorological Datasets

Meteorological datasets are essential for a wide range of applications in atmospheric science, from weather forecasting and climate monitoring to data assimilation and the training of neural networks for predictive modeling. These datasets differ in their sources, spatial and temporal resolutions, and physical consistency, and can be broadly grouped into four main categories: observational data, reanalysis products, operational numerical weather prediction (NWP) outputs, and climate model projections. In Section 2.1 a comparison between the main characteristics of each category is reported. Each of these plays a specific role within the wider ecosystem of meteorological research and operational practice. Observational datasets, whether derived from in situ measurements or remote sensing instruments, provide the empirical basis for model validation and calibration. Reanalysis datasets offer physically consistent reconstructions of past atmospheric states and are widely used for both training and evaluation in machine learning approaches. Operational NWP outputs provide timely forecasts and are increasingly being explored as inputs to hybrid modeling systems. Climate model projections extend the temporal horizon of prediction, enabling long-term scenario analysis under varying forcing assumptions. In the following sections, the main datasets used in this work are described in relation to their category and purpose, with a focus on their relevance for data-driven approaches in meteorology.

Dataset type	Nature	Time span	Timeliness	Spatial Coverage	Purpose	Example
Observations	Measured directly	Continuous, irregular	Instantaneous but sparse	Sparse in some regions	Record past reality	IGRA, MODIS
Reanalyses	Model + observations (hindcast)	Past decades	Delayed (data processing)	Complete gridded fields	Best past estimate	ERA5, CERRA
Climate Projections	Model simulation (future scenarios)	Future decades/centuries	Not real-time	Complete gridded fields	Scenario exploration	CMIP6 MPI-ESM
Operational NWP	Model forecast (near-future)	Hours–days ahead	Near-real-time	Complete gridded fields	Short- to medium-term prediction	GFS, ECMWF HRES

Table 1: Comparison of the main categories of meteorological datasets.

2.1.1 Observational Datasets

Observational datasets utilize measurements obtained directly from sensors, without the extensive numerical modeling that characterizes reanalysis or climate projections. They generally provide high-accuracy and timely information, but their coverage is often limited in space and time, and they may be affected by sensor malfunctions or retrieval issues. In the following paragraphs observational datasets are grouped into three categories: in-situ measurements, collected from ground-based stations and other locally deployed instruments; remote sensing products, acquired by radar, lidar, and

other satellite-borne or airborne sensors; and gridded observational blends, in which multiple direct measurements are combined to produce spatially continuous fields.

- **In-situ Observations:** In-situ observational datasets refer to measurements collected directly from instruments located at the Earth’s surface or within the atmosphere, typically through networks of weather stations, radiosondes, buoys, and ships. These datasets provide ground-truth measurements of meteorological variables such as temperature, pressure, humidity, wind speed and direction, among many others. In-situ data are fundamental to atmospheric science: they serve as a primary source for validating numerical models, calibrating remote sensing instruments, and constructing reanalysis products. Unlike gridded datasets or satellite-derived fields, in-situ observations are point-based and often limited in spatial coverage. However, they offer high temporal resolution and direct measurement without reliance on model assumptions, making them particularly valuable for the evaluation of high-resolution downscaling methods and data-driven prediction models. Global archives of in-situ data contribute significantly to climate monitoring, extreme weather assessment, and long-term trend analysis. Among the most widely used are the Global Historical Climatology Network (GHCN) [MDK⁺12, MDV⁺12b], which compiles long-term records of surface temperature and precipitation from land stations worldwide, and the HadISD [CAU⁺16] dataset, offering sub-daily weather station observations with extensive quality control. For marine environments, the ICOADS [FWW⁺16] (International Comprehensive Ocean-Atmosphere Data Set) provides a comprehensive collection of surface marine observations from ships, buoys, and other platforms. In this thesis, the IGRA V2 [DYV⁺16] dataset was employed for model validation in downscaling wind speed over the Italian peninsula.

Despite their accuracy and reliability, in-situ datasets suffer from a fundamental limitation: their spatial sparsity. Unlike gridded reanalysis fields or satellite imagery, in-situ measurements are irregularly distributed, with uneven global coverage and often large gaps in space and time. This poses a challenge for neural models, which are typically designed to process structured inputs such as images or spatiotemporal tensors. While there is ongoing interest in developing machine learning methods that operate directly on raw, sparse observational data, such methods are still in their early stages and are not yet widely adopted. As a result, in-situ observations are primarily used in current workflows for validation and calibration purposes, offering a robust external benchmark for the evaluation of model predictions derived from gridded or remotely sensed data.

- **Remote Sensing:** Remote sensing datasets are derived from measurements taken by instruments mounted on satellites or airborne platforms, which retrieve information about the Earth’s surface and atmosphere from a distance. These datasets play a central role in modern meteorology, oceanography, and climate science, providing global or near-global coverage at spatial and temporal resolutions that are often unattainable from in-situ networks alone. Remote sensing observations include a broad range of variables such as temperature, humidity, wind, precipitation, cloud properties, and surface characteristics, obtained using a combination of passive sensors (e.g., radiometers, spectrometers) and active sensors (e.g., radar, lidar).

One of the main advantages of remote sensing is its ability to provide consistent, repeated observations over regions that are poorly covered by ground-based networks, such as oceans, deserts, and polar areas. However, these datasets are indirect measurements: retrieval algorithms are required to convert raw radiance data into geophysical variables, introducing uncertainties that depend on sensor characteristics, atmospheric conditions, and the underlying physical models. Data gaps can also occur due to orbital geometry, instrument malfunctions, or obstructions such as cloud cover in optical imagery.

For neural modeling, remote sensing datasets are especially valuable because they are typically provided on regular grids or swaths, making them more directly compatible with convolutional and transformer-based architectures than sparse in-situ measurements. They have been used extensively in applications such as precipitation estimation from satellite radar, sea surface temperature mapping, and vegetation monitoring. At the same time, their indirect nature means they are often combined with in-situ observations and reanalyses for calibration, validation, and bias correction, particularly in high-resolution downscaling workflows.

Notable examples of remote sensing datasets include the Global Precipitation Measurement (GPM) mission for precipitation [SJPB⁺17], ASCAT [EUM10] for ocean surface winds, MODIS

[SBM⁺89] for multi-variable atmospheric and surface products, and AVHRR [Cra97] for long-term sea surface temperature records. These products, based solely on direct sensor retrievals, are widely used for climate monitoring and model evaluation.

- **Gridded Observational Blends:** Gridded observational blends combine measurements from multiple observation sources—typically in-situ stations, remote sensing platforms, and sometimes aircraft—into spatially continuous datasets on a regular grid. This is achieved through interpolation or statistical analysis methods that merge the different inputs while attempting to minimize biases and fill observational gaps. Such products retain the observational basis of their inputs while offering complete spatial coverage, making them easier to integrate into models and machine learning workflows than raw, irregularly spaced measurements. They are widely used for climate monitoring, model evaluation, and bias correction, and provide a bridge between purely observational datasets and model-based reanalyses.

Unlike reanalysis products, which ingest observations into a physics-based numerical weather prediction model to produce a dynamically consistent reconstruction of the atmosphere, observational blends rely purely on statistical techniques without imposing model physics. This distinction has important consequences: blends are far less demanding in terms of computational resources and maintain a closer connection to the raw observations, but they lack the ability to extrapolate features in a physically consistent way. As a result, they can suffer from over-smoothing, unrealistic gradients, or artifacts in data-sparse regions, and may also show temporal inhomogeneities if the underlying observation network changes over time. Despite these limitations, they remain an important observational resource for applications where preserving the direct character of the measurements is prioritized over full dynamical consistency.

Notable examples of gridded observational blends include the Global Precipitation Climatology Centre (GPCC) dataset [SBF⁺11], which provides gauge-based global precipitation analyses from quality-controlled station records, and the Climate Prediction Center (CPC) Unified Gauge-Based Analysis [XYC⁺07, CSX⁺08], offering daily precipitation fields from dense station networks. For sea surface temperature, the NOAA Optimum Interpolation SST (OISST) [RSL⁺07, BSC⁺16] product blends satellite retrievals with buoy and ship measurements to produce gap-free daily global fields. Land-based temperature and precipitation fields are also provided by the Global Historical Climatology Network (GHCN) [MDV⁺12a] gridded products, created from extensive station archives. These datasets are widely used for climate monitoring, model evaluation, and bias correction, offering spatial completeness while retaining a direct observational foundation.

As mentioned above, in the studies presented in this thesis a radiosonde dataset was employed to validate the quality of wind speed downscaling over the Italian peninsula. Considering availability, spatial coverage, and measurement reliability, the choice fell on the in-situ dataset IGRA V2, which is described in the following paragraph.

IGRA V2 For the validation of the downscaling results, this work employs data from the Integrated Global Radiosonde Archive, Version 2.2 (IGRA V2.2) [DYV⁺16]. IGRA is a comprehensive, quality-controlled global archive of radiosonde and pilot-balloon observations, providing vertical profiles of key atmospheric variables such as temperature, humidity, and wind. The dataset is maintained by NOAA’s National Centers for Environmental Information (NCEI) and is freely accessible for research purposes. Version 2.2, released in 2023, contains more than 50 million individual soundings dating back to 1905, sourced from over 2,800 land-based and ship-based stations worldwide, with daily updates from roughly 900 active sites. The archive integrates data from more than 30 historical and operational sources, compositing station records to produce the longest possible time series at each location.

For the region of interest—covering the same spatial extent as the gridded datasets used in the experiments—a subset of 14 radiosonde stations was selected based on geographic overlap. These stations are distributed across the Italian peninsula and surrounding areas, providing independent point-based observations suitable for evaluating the accuracy of gridded model outputs. For our purposes, the variable of interest is wind speed at 2 m above ground level, extracted from the IGRA profiles.

IGRA V2.2 data undergo extensive quality-control and harmonization procedures to ensure reliability, including sanity checks, surface elevation plausibility tests, internal consistency verification,

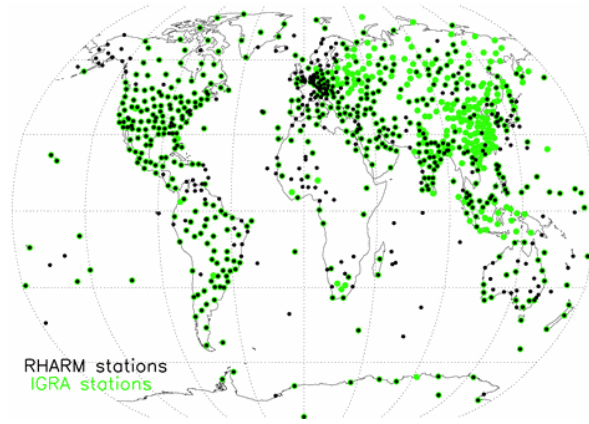


Figure 1: IGRA V2 Stations. Source: Copernicus Climate Change Service, IGRA V2 dataset overview (CDS).

duplicate detection, climatology-based validation, and vertical/temporal coherence checks. Only direct measurements are retained; values that are estimated, interpolated, or derived are excluded. In addition, very-high-resolution modern radiosonde transmissions are down-sampled to standard and significant pressure levels to maintain temporal consistency with historical records. While this ensures long-term comparability, it can reduce fine-scale vertical detail. Over its more than century-long record, the IGRA station network has evolved substantially—from fewer than ten stations in the early 20th century to more than a thousand by the 1960s, with some decline since the late 1970s due to station closures. Despite these changes, IGRA offers an exceptionally robust basis for validating both reanalysis products and data-driven downscaling approaches, particularly in applications involving vertical atmospheric structure such as wind speed at different altitudes.

2.1.2 Reanalyses

While observational datasets rely on direct measurements from instruments such as weather stations, radiosondes, ships, buoys, and satellites, they are inevitably affected by gaps in coverage, changes in measurement techniques, and heterogeneous spatial and temporal sampling. Reanalysis products address these limitations by combining vast quantities of historical observations with a consistent numerical weather prediction (NWP) model through a process known as data assimilation. In this process, diverse observations are blended with the model’s physically based equations to produce a best estimate of the state of the atmosphere (and, in some products, the land and ocean). Modern reanalyses such as ERA5 use advanced techniques like four-dimensional variational assimilation (4D-Var) or ensemble-based filters to optimise the entire three-dimensional atmospheric state over an assimilation window, matching the model trajectory to all available observations while weighting them by their uncertainty. The result is a spatially complete, physically coherent dataset in which all variables remain dynamically consistent, even in regions or times without direct measurements.

In a reanalysis system, the model provides physically consistent estimates where observations are sparse, while the assimilation process continually adjusts the model state toward the available measurements. This synergy yields datasets that are temporally homogeneous and physically coherent across variables, making them particularly valuable for climate monitoring, trend analysis, and the study of processes at scales not directly resolved by the observational network. Producing such datasets is computationally intensive: the entire historical period must be reprocessed with a fixed model version and assimilation configuration, requiring repeated model integrations, quality control of billions of observations, and the storage of petabytes of output. For example, ERA5 took several years to compute on ECMWF’s high-performance computing facility, using thousands of processor cores and generating over 20 PB of data.

Unlike pure observations, reanalyses are not direct records of the atmosphere: they represent a best estimate, constrained by both the assimilated data and the biases or structural limitations of the underlying model. They differ from operational forecasts in that the same model and assimilation system are applied uniformly across decades, avoiding discontinuities caused by model upgrades and

ensuring consistency. As a result, reanalyses can reveal long-term variability and extremes, but their accuracy varies depending on the observing system available at a given time and location.

For modern data-driven approaches to weather and climate modelling, reanalyses have emerged as the de-facto standard for model training [ZSV20]. They provide complete, uniform datasets in which all variables are aligned in space and time, covering either the entire globe or specific high-resolution regions. This level of spatial completeness and temporal coverage cannot be achieved with raw observations, which are often sparse, irregularly distributed, and affected by missing values. Reanalyses also offer long, homogeneous time series — in some cases extending back to the mid-twentieth century — produced with a fixed model and assimilation system, enabling models to learn from both common and rare meteorological events. By assimilating a wide range of observations into a single dynamical framework, they produce multi-variable fields that are physically coherent, reducing the risk of contradictory inputs when training on multiple variables. Their regular spatial grids are directly compatible with modern deep learning architectures, which can process them as images or videos without complex preprocessing. Open access through platforms such as the Copernicus Climate Data Store, often via convenient APIs, further lowers the barrier to large-scale use in both research and industry. This completeness and consistency come at a significant computational cost, one of the main motivations for developing neural-network-based emulators that can reproduce high-resolution, physically consistent reanalysis fields at a fraction of the expense. As a result, certain reanalyses have become community defaults, with ERA5 standing out as the most widely used: it combines high spatial and temporal resolution, an extensive set of variables, and assimilation of the largest observational archive to date, making it both a state-of-the-art scientific resource and a benchmark standard for data-driven meteorology [SZZ+20, MFK19].

For this thesis, four reanalysis products are of primary interest: the global ERA5, the high-resolution European CERRA, its Arctic counterpart CARRA, and the ensemble-based CERRA-EDA.

ERA5 ERA5 [HBB+18, HBB+20] is the fifth-generation global atmospheric reanalysis produced by the European Centre for Medium-Range Weather Forecasts (ECMWF) as part of the Copernicus Climate Change Service. It replaces the earlier ERA-Interim reanalysis, delivering significant improvements in spatial, temporal, and vertical resolution, the range of assimilated observations, and the representation of the Earth system. ERA5 covers the period from 1940 to the present, providing a continuous, high-resolution record of the atmosphere, land surface, and ocean waves.

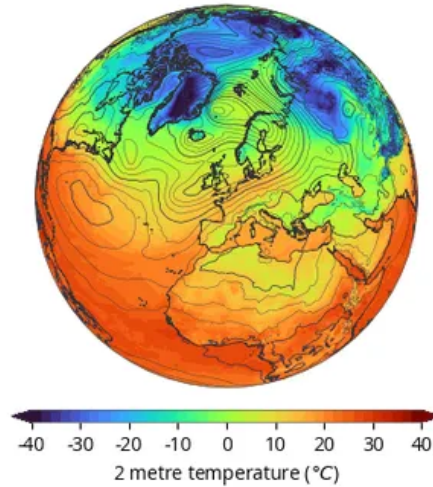


Figure 2: Example of the ERA5 global temperature field and mean sea level pressure on January 1, 2023. Source: Copernicus Climate Change Service, ERA5 dataset overview (CDS).

The dataset offers hourly estimates for more than 240 atmospheric, land, and oceanic parameters on a global $0.25^\circ \times 0.25^\circ$ grid (approximately 31 km), with 137 vertical model levels from the surface up to 0.01 hPa (about 80 km). In addition to the deterministic fields, ERA5 includes a 10-member ensemble of data assimilations (EDA) at three-hourly intervals, which can be used to estimate uncertainty. Land surface variables are provided through a coupled land reanalysis component, while ocean wave

parameters are derived from the ECMWF wave model.

A distinctive feature of ERA5 is its timely availability. The near-real-time version, known as ERA5T, is produced with a latency of approximately five days, after which it is later replaced by the final, consolidated product. This dual-stream production ensures that the dataset is both rapidly available for recent-event analysis and fully consistent over the long term. The full archive exceeds 20 petabytes, but the Copernicus Climate Data Store (CDS) provides free access with on-demand subsetting and download in GRIB or NetCDF format.

ERA5 is produced using ECMWF’s Integrated Forecasting System (IFS), Cycle 41r2, in a configuration that couples the atmospheric model to land and ocean wave components. The core of the reanalysis process is a four-dimensional variational (4D-Var) data assimilation scheme, applied over 12-hour windows. In 4D-Var, an initial model state is iteratively adjusted so that a model simulation over the assimilation window best fits all available observations, weighted according to their estimated uncertainties, while remaining consistent with the model’s governing equations. This approach ensures that the final analysis fields are dynamically balanced and physically coherent, even in regions or times with sparse observations. The EDA component perturbs both observations and model physics to produce an ensemble of analyses, providing flow-dependent estimates of background and analysis uncertainty.

The range of assimilated observations is extensive, reflecting the growth of the global observing system since the mid-20th century. It includes conventional in situ data (surface stations, radiosondes, buoys, ships, aircraft), scatterometer wind retrievals, satellite radiances from a wide range of instruments and channels, GPS radio occultation profiles, and other specialised datasets. The introduction of new observing systems over time inevitably influences reanalysis quality, with notable improvements occurring after the onset of the satellite era in the late 1970s. Nevertheless, temporal inhomogeneities can persist, and ERA5 is known to have biases in certain variables and regions, such as tropical cyclone intensity, precipitation amounts, and snow cover representation.

ERA5’s breadth and quality have made it one of the most widely used datasets in climate and meteorological research. Its applications span climate trend analysis [Yil23, LHL20, LD24, CB21], tracking changes in precipitation [YHA⁺21, LSVR22, CCC22, SWZ⁺22], hydrological studies [ZLZ⁺21a, TBA20], agricultural planning [RTC22] and extreme-event characterization [DMBA20, ASJF⁺24, DD21, WZD⁺23]. The combination of long temporal coverage, high spatial and temporal resolution, physically consistent multi-variable fields, and free, programmatic access through the CDS has also made ERA5 the default reference dataset for training and evaluating neural models in meteorology. The computational cost of producing such a dataset is immense—ERA5 required years of continuous high-performance computing, processing billions of observations and storing petabytes of output—underscoring the potential value of machine-learning-based emulators capable of approximating reanalysis-quality data at far lower cost.

ERA5 serves as a backbone training dataset for many state-of-the-art neural weather and climate models. Among the most notable are global forecasting architectures that have achieved skill levels matching those of leading operational systems. GraphCast, a graph neural network model trained on decades of ERA5 data, produces 10-day forecasts for hundreds of atmospheric variables and has been shown to outperform ECMWF’s operational deterministic forecasts on roughly 90 % of verification metrics, while improving the prediction of extreme events such as tropical cyclones, atmospheric rivers, and heatwaves [LSGW⁺22, LSGW⁺23]. Pangu-Weather, based on a 3D Earth-specific Transformer, likewise trained on ERA5, delivers medium-range forecasts that surpass ECMWF’s Integrated Forecasting System in accuracy across key variables, with far lower computational cost [BXZ⁺22]. FourCastNet, an implementation of the Fourier Neural Operator (FNO) architecture, also trained on the full global ERA5 archive, has demonstrated skill comparable to state-of-the-art NWP systems for medium-range forecasting while producing high-resolution global predictions in seconds [PSH⁺22]. Similarly, the Fuxi global weather model, developed by Huawei Cloud, uses a transformer-based architecture trained on ERA5 to achieve competitive or superior performance to operational IFS runs, with particular strengths in tropical cyclone tracking and precipitation prediction [CZZ⁺23]. Together, these advances illustrate how ERA5 has become the foundation for cutting-edge deep learning approaches, enabling progress from regional downscaling and nowcasting to global multi-variable forecasting, and driving measurable gains in the prediction of high-impact weather and climate extremes.

CERRA The Copernicus Regional Reanalysis for Europe (CERRA) is a high-resolution regional reanalysis (RRA) tailored specifically to the European domain. Produced under the Copernicus Climate Change Service through a contract with the Swedish Meteorological and Hydrological Institute (SMHI), in collaboration with Météo-France and the Norwegian Meteorological Institute, CERRA provides atmospheric, land, and near-surface variables at a horizontal resolution of 5.5 km. It follows the same fundamental principles as global reanalyses but leverages a regional numerical weather prediction system to resolve finer-scale processes over Europe. Unlike ERA5, which is produced on a regular latitude–longitude (cylindrical) grid, CERRA uses a Lambert conformal conic projection optimized for the European domain. This reduces distortion across mid-latitudes, providing more uniform spatial resolution and improving the representation of distances and areas for regional-scale applications.

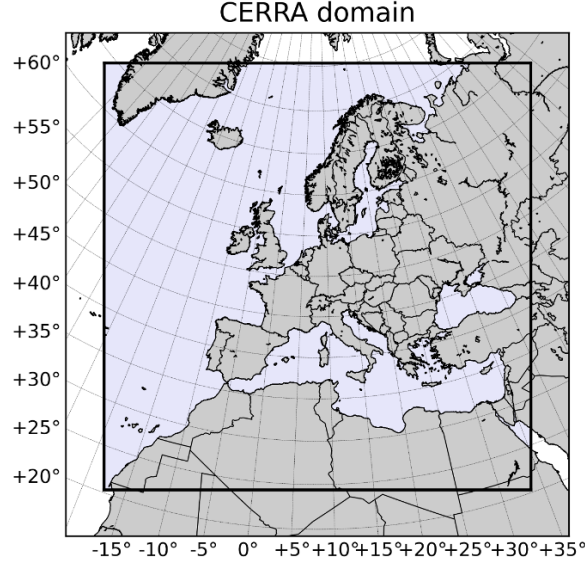


Figure 3: CERRA European Domain. Source: Copernicus Climate Change Service, CERRA dataset overview (CDS).

CERRA is driven by the global ERA5 reanalysis, which supplies both initial and lateral boundary conditions, and supplements these with higher-resolution in situ observations, remote-sensing data, and physiographic datasets describing surface properties. The observation set is extensive, including synoptic station reports, satellite radiances, atmospheric motion vectors, radio-occultation bending angles, and ground-based GNSS zenith delays. The assimilation process, described in section 2.1.2 for ERA5, is here applied at regional scale, allowing the model to ingest a denser and more detailed observational network than is possible at global resolution. This yields a physically consistent dataset that more accurately represents European weather and climate variability, especially in complex terrain and coastal zones.

Vertically, CERRA uses 106 model levels extending to 1 hPa, but most outputs are provided on 29 pressure levels (1000–1 hPa) and 11 fixed-height levels (15–500 m) to support applications such as wind energy assessment. The 5.5 km horizontal grid means each point represents roughly 30 km².

The temporal structure of CERRA output differs from the hourly analysis cycle of ERA5. Eight reanalyses are produced each day, every three hours from 00 UTC to 21 UTC. For each analysis time, a short-range forecast is run for the next six hours; at 00 UTC and 12 UTC, extended forecasts up to 30 hours are also generated at three-hourly intervals after the first six hours. While CERRA provides fields for every hour of the day, only the hours corresponding to analysis times benefit directly from observational correction; intermediate hours are forecast-only.

CERRA’s variable set includes temperature, humidity, wind, precipitation, cloud cover, radiation, and a range of surface parameters. Evaluations show that it generally outperforms ERA5 in 2 m temperature and 10 m wind speed, while CERRA-Land improves precipitation and snow-depth representation over its predecessors. These strengths have proven advantageous for numerous applications. For example, CERRA outperforms ERA5-Land in reproducing temperature and bio-meteorological indices, making it well suited for assessing heat-related health risks [GGA⁺23]. It has been applied to

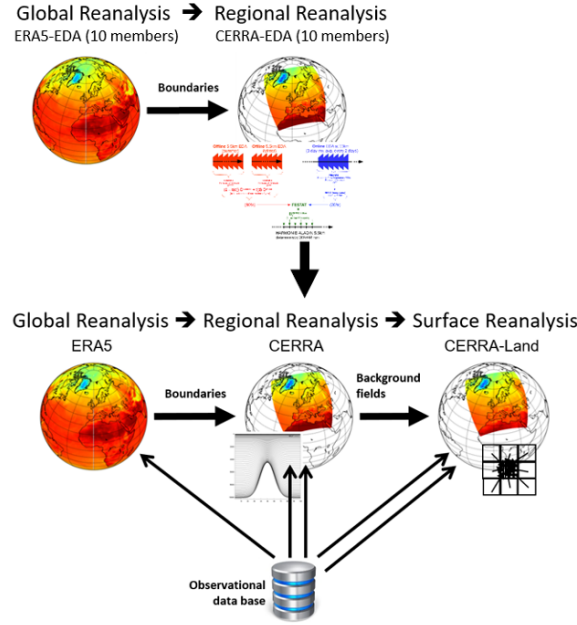


Figure 4: Overview of the CERRA production chain. Lateral boundary conditions are taken from the 10 ERA5-EDA ensemble members for the 10 ensemble members of CERRA-EDA. The CERRA-EDA provides background errors estimates to CERRA, which is driven by ERA5 at the lateral boundary conditions. CERRA-Land is then in turn driven by CERRA. Source: Copernicus Climate Change Service, CERRA dataset overview (CDS).

renewable energy forecasting [NPR⁺23], water resource assessment through improved precipitation and snow representation [VGBM21, MDKL23], and as a substitute for ground-based agro-meteorological measurements [Pel23]. These capabilities make CERRA a valuable resource for regional climate analysis, adaptation planning, and policy development.

A key limitation is its lack of real-time availability: at the time of writing, the operational release of CERRA lags by over two years due to the processing and quality control requirements of the reanalysis cycle. Additional caveats include temporal inconsistencies in observation availability — particularly for precipitation in eastern regions after reductions in reporting post-1999 — and small spin-up discontinuities in deep soil layers between production streams, which are best avoided by using CERRA-Land for such variables. Furthermore, the relatively short temporal coverage (1984–2021) means that climate trend analyses should be interpreted with caution. These constraints limit some uses, but do not diminish CERRA’s value as a detailed, high-quality historical record for Europe.

CERRA-EDA CERRA-EDA [RBM⁺24] is an Ensemble of Data Assimilation (EDA) system comprising 10 members, designed to provide flow-dependent background error statistics for the deterministic CERRA reanalysis. It is based on the same HARMONIE-AROME [BAA⁺17, TFB⁺18] modelling framework and observational dataset as the deterministic CERRA system (except for the exclusion of the Microwave Sounding Unit), but runs at a coarser 11 km horizontal resolution (versus 5.5 km in CERRA) and with longer time steps.

The CERRA-EDA system covers the same temporal period as CERRA, from January 1985 to the present, producing 6-hourly analyses (00, 06, 12, and 18 UTC) and short-range forecasts up to +6 h for each ensemble member. One control member is generated without perturbations, while the other nine members incorporate Gaussian-distributed, spatially uncorrelated perturbations to the observations, scaled according to estimated observational errors. No perturbations are applied to sea surface temperature, sea ice, or physical parameters. Lateral boundary conditions are taken from the ERA5-EDA, which also consists of 10 members and operates on a 63 km global grid, thereby introducing large-scale flow perturbations into the regional ensemble.

Analyses from CERRA-EDA are used to estimate background error covariances for CERRA’s three-dimensional variational assimilation (3D-Var) scheme. While CERRA-EDA is run in advance of the

deterministic CERRA, the lower resolution and longer time steps enable it to complete production faster, with the 10 ensemble members processed in parallel. The resulting dataset, available through the Copernicus Climate Data Store in GRIB and NetCDF formats, provides both a crucial input to the deterministic reanalysis and a stand-alone research-grade regional ensemble reanalysis that supports uncertainty estimation and climate studies at high latitudes.

Producing a regional reanalysis ensemble such as CERRA-EDA is a computational feat: even at 11 km resolution, with longer time steps and pared-back physics perturbations, generating 10 members over decades of data demands vast high-performance computing resources. This inherent cost explains why the resolution must be reduced compared to the deterministic CERRA. In this thesis we explore in Section 5 how neural emulators can be trained to act as efficient surrogates for such ensembles. The aim is to capture the uncertainty information provided by CERRA-EDA at a fraction of the computational expense, opening the door to more frequent, higher-resolution, and accessible regional ensemble reanalyses.

CARRA The Copernicus Arctic Regional Reanalysis (CARRA) dataset [Cop24] is specifically designed to represent the unique environmental and climatic conditions of the Arctic region in greater detail than global reanalyses such as ERA5. While ERA5 provides a 31 km global resolution, CARRA offers a 2.5 km horizontal resolution on Lambert conformal grids (CARRA-WEST: 1069×1269 , CARRA-EAST: 789×989) with 65 vertical levels, capturing finer details essential for understanding the Arctic’s rapidly changing climate. Observational records and climate projections show that warming in the Arctic occurs at over twice the global average rate, leading to increased environmental and economic activities and driving a need for detailed data to support climate adaptation and management in the region.

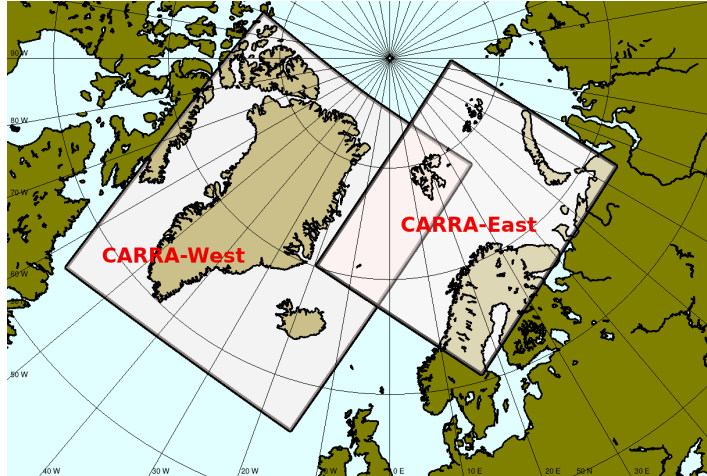


Figure 5: The CARRA-West and CARRA-East (Copernicus Arctic Regional Reanalysis) domains. Source: Copernicus Climate Change Service, CARRA dataset overview (CDS). Source: ECMWF, Copernicus Arctic Regional Reanalysis (CARRA) – Added value to the ERA5 global reanalysis.

CARRA leverages the same HARMONIE-AROME weather prediction model as CERRA, enhanced for reanalysis applications at the ECMWF high-performance computing facility. It features a three-hourly analysis cycle and hourly short-range forecasts (up to +30 h from 00 and 12 UTC), updated monthly, covering the period from 1991 to present. The system employs a three-dimensional variational (3D-Var) scheme for upper-air assimilation and optimum interpolation for surface and soil variables, with ERA5 providing lateral boundary conditions. Its data assimilation integrates a wide range of sources, including extensive local observations from Nordic countries and Greenland, advanced satellite-derived glacier albedo data, improved sea ice and snow initialization, and high-resolution physiography and orography data. The latter includes refined glacier masks, land-sea mask corrections for Greenland fjords, improved representation of the Austfonna Ice Cap, and increased snow roughness lengths to mitigate winter wind biases. The surface physics is based on SURFEX v7.3 [MLMM⁺13], and the model uses the SICE thermodynamic sea-ice scheme with annual re-initialization of glacier snow water equivalent each 1 September to prevent melt artifacts.

For accumulated variables (e.g., precipitation and energy fluxes), the recommended usage is to avoid the 0–6 h forecast range after each analysis to reduce spin-up, instead using +6 h to +18 h forecasts; instantaneous hourly values can be taken from the +1 h/+2 h steps between analyses. CARRA generally shows improved agreement with in-situ observations compared to ERA5, particularly for near-surface variables over complex terrain and coastal regions, and better representation of extremes.

The dataset spans two key domains, which overlap around Svalbard and Jan Mayen:

- CARRA-WEST, covering Greenland, the Labrador Sea, Davis Strait, Baffin Bay, Denmark Strait, Iceland, Jan Mayen, the Greenland Sea, and Svalbard.
- CARRA-EAST, which includes Svalbard, Jan Mayen, Franz Josef Land, Novaya Zemlya, Barents Sea, and the northern regions of the Norwegian Sea and Scandinavia.

These domains encompass the Arctic’s four largest land ice bodies, including the Greenland Ice Sheet and the Austfonna Ice Cap, which are focal points in climate research. CARRA’s high resolution allows for accurate representation of critical Arctic-specific weather phenomena, such as polar lows in the North Atlantic and Greenland’s downslope katabatic winds, which have significant implications for local communities and maritime activities.

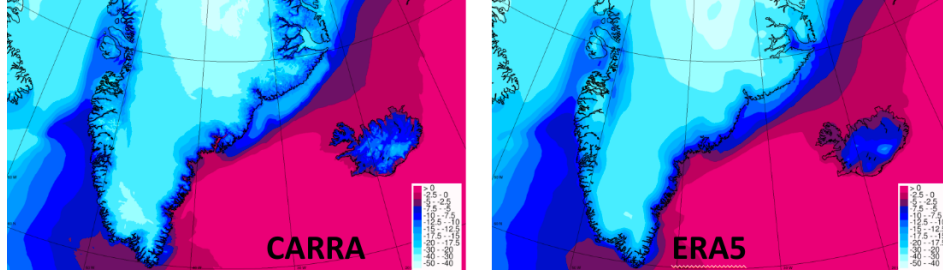


Figure 6: T2m air temperature at south Greenland and Iceland 15 February 2014 00 UTC for CARRA (left) and ERA5 (right). Source: ECMWF, Copernicus Arctic Regional Reanalysis (CARRA) – Added value to the ERA5 global reanalysis.

In this thesis, we focus on the CARRA-WEST domain. Crucially, unlike CERRA and ERA5, CARRA does not provide an Ensemble Data Assimilation (EDA) version of the dataset, and thus lacks ensemble information. Yet ensemble datasets are essential for quantifying uncertainties [WMS⁺20, LGWZN⁺24a, LGG24], detecting extreme events [TFS15], and performing ensemble-based statistical analyses critical for assessing the likelihood of compound extreme events [KLS⁺23, ASB⁺19]. The paper presented in Section 5 of this thesis directly addresses this limitation, exploring approaches to supplement or approximate ensemble-based uncertainty estimates for high-resolution Arctic reanalysis data.

2.1.3 Operational NWP forecasts

Operational Numerical Weather Prediction (NWP) forecasts are the model-based weather forecasts that form the basis of the daily and medium-range predictions used by meteorologists, the public, and weather-sensitive sectors. They provide guidance from a few hours ahead up to around 15 days, covering the short- and medium-term timescales where accurate atmospheric prediction is still possible. These forecasts are routinely produced by national and international meteorological agencies, updated several times per day, and disseminated globally to support decision-making in areas such as aviation, shipping, agriculture, disaster preparedness, and general public information.

In the broader context of meteorological data, operational NWP forecasts differ fundamentally from the other main categories while sharing a common heritage. Like reanalyses and climate projections, they are produced using comprehensive numerical models of the atmosphere that solve the same set of physical equations and often share large parts of their code base. The distinction lies primarily in initialization and purpose: reanalyses reconstruct past atmospheric states by assimilating historical observations, climate projections integrate the same type of models far into the future without data assimilation under prescribed forcing scenarios, and operational NWP forecasts start from the current observed state and extend the simulation into the near future. All three produce spatially complete,

gridded datasets, but operational NWP is unique in its high update frequency and its focus on rapid delivery for real-time decision-making.

A notable example of this shared modeling heritage is the Integrated Forecasting System (IFS) [WBD⁺15] developed by ECMWF. This single modeling framework underpins the ERA5 reanalysis, the HRES and ENS operational forecasts, and is also used in CMIP climate projection modeling activities, illustrating how a single numerical system can operate in all three modes [RSM⁺18].

Alongside ECMWF’s Integrated Forecasting System (IFS) in its High-Resolution Forecast (HRES) configuration, a number of other operational NWP systems are produced by global and regional forecasting centers. At the global scale, these include the NOAA/NCEP Global Forecast System [Nat07] (GFS; 0.25° resolution, up to 16-day lead time, 4 runs/day), the ECMWF Ensemble Prediction System (ENS; 50 members), the German Weather Service’s ICON model [Deu23], the UK Met Office Global model, and the CMA-GFS [SSZH23] operated by the China Meteorological Administration. The number of such global forecasting systems is small, as they are extremely costly to design and, even more so, to run operationally at high resolution. Several high-resolution, convection-permitting regional systems are also in widespread operational use, such as AROME (Météo-France) [GKDR⁺24], HRRR [Nat25] (High-Resolution Rapid Refresh, U.S., 3 km resolution, hourly updates), and the JMA GSM [Jap25] (Japan).

Despite their importance, operational NWP datasets can be difficult to access in bulk. Many national meteorological services provide free access only to limited subsets—often reduced in spatial resolution, temporal frequency, or number of variables—while retaining full-resolution historical archives under licensing restrictions or paid access agreements. Even for open datasets such as the GFS, which is freely available, building large training archives can be impractical. The data are typically distributed as large, monolithic forecast files, making it necessary to download terabytes of information even if only a single variable is needed. Archives are often split across different storage locations depending on the time period, and long-term coverage can include missing or corrupted files. For applications such as training neural weather models, where multi-year, variable-specific datasets are required, these factors pose a significant barrier.

Machine learning-based forecasting systems have emerged as a promising complement to traditional NWP, although most remain in the research and development stage rather than fully operational. Approaches such as Google DeepMind’s GraphCast, ECMWF’s AIFS, Huawei’s Pangu-Weather, NVIDIA’s FourCastNet, and the MetNet family have demonstrated that high-quality global forecasts can be produced without the expensive numerical integration of full-physics models. By replacing hours of supercomputing time with seconds or minutes of inference on far more modest hardware, these methods drastically reduce the computational cost of producing forecasts. This efficiency could enable a much wider range of institutions and countries to run high-quality global prediction systems, breaking the historical dominance of a handful of well-funded national centers.

2.1.4 Climate Projections

Climate projections provide physically based estimates of the possible future evolution of the climate system under specified socio-economic and greenhouse gas emission scenarios. Climate projections are forward-looking simulations driven by prescribed changes in atmospheric composition, land use, and other forcings. They are conditional outcomes: if a given emissions pathway were to occur, the climate system would likely evolve in a manner consistent with the model’s physical representation of the Earth system.

Although climate projections and reanalyses are often discussed as distinct products, the underlying numerical models are frequently from the same family. Many atmospheric and coupled Earth system models can be configured for both purposes, with the principal difference lying in the inclusion of a data assimilation system. In a reanalysis, observations are continually ingested via schemes such as three- or four-dimensional variational assimilation or ensemble Kalman filters, ensuring that the simulated state remains close to reality. In a projection, the model is instead initialized once and then run freely under prescribed external forcings, with its trajectory determined solely by the model physics. The physical parameterizations are generally similar in both cases, although reanalyses may tune them for short-term forecast accuracy, while projections prioritize long-term climate stability and energy balance. Another common difference is in coupling: many reanalyses use an atmosphere-only model forced by observed ocean and sea ice conditions (e.g., ERA5), while climate projections from GCMs like MPI-ESM-LR simulate a fully coupled Earth system—including atmosphere, ocean, land,

and sea ice—to capture long-term feedbacks.

Global Climate Models (GCMs) operate at horizontal resolutions of about 1° to 2° (roughly 100–200 km), enabling them to capture large-scale circulation patterns and broad climate responses to external forcings, but not fine-scale regional processes. They simulate the coupled atmosphere, ocean, land, and cryosphere over multi-decadal to centennial timescales, typically including both a historical period—used to evaluate model performance against observations—and future periods driven by prescribed forcing scenarios. These scenarios, such as the Representative Concentration Pathways (RCPs) used in CMIP5 [TSM12] and the more recent Shared Socioeconomic Pathways (SSPs) used in CMIP6 [EBM⁺16b], describe possible trajectories of radiative forcing (in W m^{-2}) until the year 2100, along with internally consistent evolutions of greenhouse gas emissions, aerosols, and land-use changes. Scenario uncertainty, together with internal variability and differences in model formulation, forms one of the three main sources of uncertainty in climate projections. Examples of widely used GCMs include MPI-ESM, developed by the Max Planck Institute for Meteorology, and NorESM, developed by the Norwegian Climate Centre, both of which have been extensively employed as driving models in regional downscaling projects such as EURO-CORDEX.

Because GCM resolution is insufficient to resolve regional details such as complex topography, coastlines, and small-scale weather phenomena, a process known as downscaling is often applied. Downscaling can be achieved dynamically, using Regional Climate Models (RCMs) that are nested within GCMs and run at finer spatial resolution over limited domains, or statistically, using empirical or machine-learning techniques that link large-scale and local climate features. RCMs are particularly valuable for impact assessments, as they can explicitly represent processes at scales of tens of kilometres or less, producing output suitable for sectors such as agriculture, hydrology, and urban planning.

The Coordinated Regional Downscaling Experiment (CORDEX), initiated by the World Climate Research Programme, provides a framework for producing and distributing consistent RCM simulations across multiple regions of the globe [GG15]. Within this framework, EURO-CORDEX focuses on Europe and offers simulations at horizontal resolutions of 0.44° (50 km) and 0.11° (12 km), driven by a range of GCM boundary conditions and emission scenarios. EURO-CORDEX data have become a standard resource for regional climate impact studies across Europe due to their high resolution, multi-model ensembles, and standardised protocols.

Beyond CORDEX, other RCM initiatives exist, often tailored to specific domains or research needs. Examples include PRUDENCE [CCRA07] (Prediction of Regional scenarios and Uncertainties for Defining European Climate change risks and Effects), which provided early high-resolution climate projections for Europe; NARCCAP [MSL⁺13] (North American Regional Climate Change Assessment Program) for North America; and dynamically downscaled experiments conducted by national meteorological agencies such as DMI-HIRHAM [CDC⁺07] in the Nordic region or UKCP [MSJ⁺09] (UK Climate Projections) in the United Kingdom. These complementary efforts highlight the broad applicability and diversity of regional modelling strategies for climate change assessment. In Section 2.1.4 a comparison between the presented RCMs is reported.

In this thesis, climate projections from the EURO-CORDEX framework will be employed in Section 6 to train data-driven downscaling approaches and investigate the representation of multiple climate variables at high spatial resolution. Specific model configurations and scenarios used in the experiments will be described in detail in the relevant sections.

2.2 Deep Learning in Climate Science

In this section we present a historical review of the trajectory of research on neural networks in climate and weather science, focusing on forecasting, downscaling, and related applications. The goal is to highlight the main breakthroughs and most relevant works of the past three decades, providing a concise and practical guide to situate the current state of research. For clarity, we divide the literature into three periods. The first, spanning the 1990s and 2000s, covers the foundations and early neural network applications, when multilayer perceptrons were primarily used as nonlinear statistical tools for localized forecasting or downscaling. The second, the 2010s, marks the emergence of deep learning, with the adoption of convolutional and recurrent architectures and the conceptual shift from point-based to field-based modeling. The third, the 2020s, documents the maturation of the field, when machine learning approaches became competitive with, and in some cases superior to, numerical weather prediction, extending into generative ensembles, physics-informed hybrids, and foundation models.

Initiative	Domain	Resolution	Notes
EURO-CORDEX	Europe	0.44°–0.11°	European branch of CORDEX; multi-RCM, multi-GCM framework; widely used in impact studies across Europe.
AFRICA-CORDEX	Africa	0.44°–0.22°	African branch of CORDEX; focuses on continental-scale climate projections, often used for adaptation planning.
MED-CORDEX	Mediterranean	0.44°–0.11°	CORDEX branch for the Mediterranean basin; includes coupled atmosphere–ocean RCMs.
EAS-CORDEX	East Asia	0.44°–0.22°	CORDEX domain for East Asia; includes contributions from Chinese Academy of Sciences and national meteorological agencies; uses models such as RegCM4 and PRECIS.
SEA-CORDEX	Southeast Asia	0.44°–0.22°	CORDEX domain for Southeast Asia; coordinated by regional climate centres and universities; focuses on monsoon variability and tropical cyclones.
SA-CORDEX	South Asia	0.44°–0.22°	CORDEX domain for South Asia; coordinated by the Indian Institute of Tropical Meteorology; targets monsoon dynamics and high-impact events.
PRUDENCE	Europe	~50 km	Early coordinated RCM experiment (2001–2004); precursor to CORDEX.
NARCCAP	North America	50 km	Multi-RCM, multi-GCM framework for climate change impacts over the U.S., Canada, and northern Mexico.
DMI-HIRHAM	Nordic/Arctic	5–25 km	High-resolution RCM by the Danish Meteorological Institute, used in Arctic and Nordic regional studies.
UKCP	United Kingdom	12 km	UK Met Office high-resolution projections with probabilistic ensembles for impact assessment.

Table 2: Selected regional climate modelling datasets.

2.2.1 1990s–2000s: Foundations and Early Neural Nets

Architecturally, meteorological neural networks of the 1990s and 2000s were almost exclusively shallow models, with just one or two hidden layers and small neuron counts by today’s standards [McC92, NC94, MS96]. The workhorse was the multilayer perceptron trained with the backpropagation algorithm [McC92, KB98], but the label artificial neural network (ANN) encompassed a broader family. At the time, there was no consensus that backpropagation would become the de facto standard, and researchers explored alternatives, including radial basis function networks (RBFNs) and self-organizing maps (SOMs) discussed in reviews of the time [HT98, MKA02]. These alternatives often required different training paradigms: for example, RBFNs typically set hidden-layer centers via unsupervised clustering and fitted output weights with least-squares regression [MKA02]. This diversity reflected not only the absence of an agreed “best practice” for training neural networks in forecasting, but also the severe computational constraints of the era, when both processing power and memory were extremely limited compared to modern GPU-based workflows.

Experimentally, this period also saw the emergence of hybrid approaches, in which neural networks were combined with other computational techniques to improve performance or interpretability—e.g., fuzzy neural networks and rough set–based feature selection, as well as ensembles that merged the outputs of different architectures [LL05, MKA02]. These methods reflected an exploratory spirit, as researchers sought to boost skill scores, reduce overfitting, and adapt models to the noisy, incomplete datasets common in meteorology.

Methodologically, most neural network models operated in a point-to-point fashion. Each observation site or grid cell was treated as an independent sample, represented by a vector of meteorological predictors such as stability indices, SST anomalies, or recent rainfall totals [McC92, NC94]. Spatial context, if used, was incorporated indirectly—through engineered features like neighboring station values—or by preprocessing gridded fields with techniques such as principal component analysis (PCA) [JM02]. Convolutional processing of spatial data, which would later become central to neural weather prediction, was not yet in use, and convolutional neural networks (CNNs) played no role in operational or research meteorology at the time.

In application, these models were typically task-specific and data-limited. They targeted single variables—rainfall, temperature, wind, severe weather occurrence—trained on relatively small datasets constrained by observational archives [SPA+06, HM07, HBWT09]. Forecast lead times ranged from a few hours (e.g., thunderstorm or flood nowcasting) [McC92, LOLB04, HBWT09] to seasonal predictions [NC94]. ANNs were deployed both as stand-alone predictors and as nonlinear post-processors of numerical weather prediction (NWP) output [KB98], frequently outperforming regression/MOS and time-series baselines such as ARIMA on nonlinear, local-scale problems [KB98, SPA+06].

For what concerns statistical downscaling, early studies in this period demonstrated that multilayer perceptrons could be trained to translate large-scale atmospheric predictors from reanalyses or GCMs into local-scale climate variables. Examples include applications to temperature and precipitation extremes in the United States [SP01], seasonal precipitation and temperature in the Great Lakes Basin [CDA05], and daily weather elements for rainfall–runoff modelling in Canada [KCD06]. These works showed that even with shallow architectures and limited computing resources, ANNs could capture nonlinear relationships between coarse-resolution climate drivers and station-level observations, often outperforming conventional regression-based downscaling in representing variability and extremes.

2.2.2 2010s: Deep Learning Emerges in Climate Science

The 2010s represented a decade of profound transformation for the application of neural networks in the atmospheric sciences, marking a clear division between two distinct eras. The first half of the decade was characterized by the use of machine learning as a refined statistical tool for localized problems, in continuity with the state of the art of the late 2000s, described in Section 2.2.1. During this period, Artificial Neural Networks, typically Multi-Layer Perceptrons, were used to model nonlinear relationships in vector-based data. Research from this time commonly features models designed for specific tasks at individual locations. For instance, studies by Baboo & Shereef (2010) [BS10] and Abhishek et al. (2012) [ASGA12] showcase the use of ANNs to create efficient, automated systems for forecasting weather parameters at a single site from past measurements. This approach treated the problem as a time-series regression, a common and practical application for the models of the time.

The research landscape of the early 2010s also involved extensive comparisons between different

Year	Authors	Task / Variable	Architecture	Key Contribution
1992	McCann [McC92]	Thunderstorm forecasting (3–7 h)	MLP (BPN)	First operational ANN in US forecast center; used lifted index & moisture convergence.
1994	Navone & Ceccatto [NC94]	Seasonal Indian monsoon rainfall	MLP (BPN)	First seasonal climate forecast skill with ANNs
1996	Marzban & Stumpf [MS96]	Tornado occurrence prediction	MLP	Used Doppler radar attributes for severe weather guidance.
1997	Hsu et al. (PERSIANN) [SHG ⁺ 00]	Rainfall estimation from satellite IR	Adaptive ANN	First adaptive NN rainfall retrieval from remote sensing imagery.
2001	Schoof & Pryor [SP01]	Downscaling temperature and precipitation extremes (USA)	MLP (BPN)	Linked large-scale predictors to local extremes; selectively outperforms regression
2002	Jaruszewicz & Mandziuk [JM02]	Weather prediction	MLP + PCA	Dimensionality reduction improved ANN forecast skill.
2002	Maqsood et al. [MKA02]	Daily weather variables	MLP, Elman, RBFN, ensembles	Ensembles outperformed single models for temperature, wind, humidity.
2004	Lekkas et al. [LOLB04]	Flood forecasting (7 h)	MLP (BPN)	Added error prediction correction to improve precision.
2005	Coulibaly et al. [CDA05]	Downscaling precipitation and temperature (Great Lakes Basin)	Temporal NN	Captured nonlinear predictor–predictand relationships; improved seasonal hydrological forecasts.
2005	Li & Liu [LL05]	Weather forecasting	Rough-set-based Fuzzy NN	Hybrid for uncertainty handling and feature selection.
2006	Khan et al. [KCD06]	Downscaling daily weather for rainfall–runoff modelling (Ontario)	MLP (BPN)	Translated large-scale predictors to local daily variables for hydrological applications.
2006	Somvanshi et al. [SPA ⁺ 06]	Rainfall (Hyderabad, 104 yrs)	MLP (BPN)	ANN outperformed ARIMA in handling nonlinear patterns.
2007	Hayati & Mohebi [HM07]	Short-term temperature	MLP	Achieved lowest errors among compared models.
2009	Hung et al. [HBWT09]	Real-time rainfall & flood mgmt (Bangkok)	Generalized feedforward NN	Integrated multiple stations; better than persistence; identified key predictors.

Table 3: Selected influential neural network studies in meteorology during the 1990s and 2000s, including early applications to statistical downscaling. BPN = backpropagation network; QPF = quantitative precipitation forecast; PCA = principal component analysis; RBFN = radial basis function network.

Year	Authors	Task / Variable	Architecture	Key Contribution
2010	Baboo & Shereef [BS10]	Local weather forecasting	MLP (ANN)	Efficient, automated forecasting for a single location.
2011	Chadwick et al. [CCG11]	Downscaling GCM to RCM scale	MLP (ANN)	Used ANN as a computationally cheap emulator for a physical model (RCM).
2011	Kisi & Cimen [HWLD14]	River flow forecasting	MLP, SVM, ANFIS	Compared multiple shallow ML models.
2012	Abhishek et al. [ASGA12]	Local weather forecasting	MLP (BPN)	Standard example of localized time-series forecasting with a simple ANN.
2015	Shi et al. [SCW ⁺ 15]	Precipitation now-casting (radar)	ConvLSTM	Introduced ConvLSTM to learn spatio-temporal dependencies from image sequences.
2016	Liu et al. [LRC ⁺ 16]	Extreme weather detection	CNN	First to treat climate data as fields for automated feature detection with a CNN.
2017	Vandal et al. [VKG ⁺ 17]	Statistical downscaling (precip/temp)	CNN (SRCNN)	Framed downscaling as a single-image super-resolution computer vision problem.
2017	Shi et al. [SGL ⁺ 17]	Precipitation now-casting (radar)	TrajGRU	new model Handle location-variant motion, improving on ConvLSTM.
2017	Racah et al. [RBM ⁺ 17]	Extreme weather detection	Spatio-temporal CNN	Created a benchmark dataset and used a semi-supervised model.
2018	Rodrigues et al. [ROC18]	Statistical downscaling	Deep CNN	Validated deep learning for downscaling with a deeper, more powerful architecture.
2018	Khodayar & Wang [KW18]	Wind speed forecasting	Spatio-temporal GNN	Introduced GNNs for handling irregular spatial data in weather forecasting.
2019	Agrawal et al. [ABB ⁺ 19]	Precipitation now-casting (radar)	U-Net	Applied U-Net for dense prediction, preserving fine spatial details in rainfall.
2019	Weyn et al. [SBG ⁺ 21]	Global weather forecasting (Z500)	Deep CNN (U-Net like)	Proof-of-concept for deep learning as a potential global forecast model replacement.

Table 4: Selected influential neural network studies in meteorology during the 2010s, illustrating the shift from point-to-point MLP models to field-to-field deep learning architectures. MLP = Multi-layer Perceptron; GCM/RCM = Global/Regional Climate Model; CNN = Convolutional Neural Network; GNN = Graph Neural Network.

shallow learning models to find the optimal tool for a given task. A representative example is the work by Kisi & Cimen (2011) [HWLD14], which compared the performance of ANNs against Support Vector Machines and neuro-fuzzy systems for forecasting river flow, concluding that no single model was universally superior. Concurrently, more ambitious applications began to emerge, acting as precursors to modern deep learning. The work by Chadwick et al. (2011) [CCG11] used an ANN as a computationally inexpensive proxy model, or emulator, to learn the complex relationships involved in downscaling from a coarse Global Climate Model to a finer Regional Climate Model. While still conceptually a point-to-point model, this represented a significant step toward using machine learning to bridge scales in climate modeling.

A paradigm shift occurred around the middle of the decade, spurred by advances in computer vision and hardware. Researchers began to treat gridded meteorological data not as a collection of points, but as images, enabling the use of deep learning architectures that could process spatial information. This conceptual leap from "point-to-point" to "field-to-field" models defines the second half of the decade. Landmark work soon demonstrated that a Convolutional Neural Network (CNN) could automatically detect extreme weather events like tropical cyclones within large climate datasets, a task previously requiring laborious manual analysis [LRC+16]. This approach was further refined through the development of more advanced spatiotemporal CNNs and the creation of large, standardized benchmarks like the "ExtremeWeather" dataset, which facilitated model development and comparison [RBM+17]. These changes were further enhanced by the emergence of modern neural design libraries, such as Tensorflow and PyTorch, which streamlined the design and training processes.

The application of CNNs to statistical downscaling followed a similar path of rapid innovation. Vandal et al. (2017) [VKG+17] framed the problem as a single-image super-resolution task, a classic computer vision challenge, successfully training a CNN to generate high-resolution climate projections. This approach was further improved by implementing more sophisticated CNN designs, which yielded significant skill improvements against previous point-to-point approaches and validated deep learning as a practical tool for generating high-resolution weather forecasts [ROCN18]. The emergence of the U-Net architecture, with its broad applicability and the ability to preserve fine spatial detail, also proved highly effective for dense prediction tasks like precipitation nowcasting [ABB+19].

Alongside these spatial pattern recognition models, novel architectures were invented to handle spatio-temporal sequences. The introduction of the Convolutional LSTM (ConvLSTM) by Shi et al. (2015) was a seminal moment, providing an elegant architecture that could process both spatial and temporal dependencies simultaneously, setting a new standard for precipitation nowcasting from radar imagery [SCW+15]. The same group later identified a key limitation of the ConvLSTM—that its learned patterns are spatially uniform—and proposed the Trajectory GRU (TrajGRU) in 2017 [SGL+17]. This new model could learn location-variant motions, representing a significant step toward encoding more physical realism within deep learning models. This period also saw a diversification in architectures to handle different types of spatial arrangements. While CNN-based models excelled on regular grids, Khodayar and Wang (2018) introduced a Graph Neural Network (GNN) to forecast wind speed across the irregular layout of wind farms, demonstrating the flexibility of deep learning in adapting to the specific physical structure of a problem [KW18]. By the end of the decade, the ambition of the field had grown immensely, culminating in works like that of Weyn et al. (2019), which used a deep CNN to forecast global atmospheric fields for several days, posing a direct challenge to traditional numerical models and shifting the perception of deep learning from a subordinate tool to a potential framework for weather prediction itself [SBG+21].

2.2.3 2020s: Scale, Speed, and Probabilistic Models

The 2020s constitute a decisive period in the evolution of neural network applications in meteorology and climate science. Whereas the 2010s had already witnessed the transition from localized, point-based regression models to deep learning architectures capable of processing gridded meteorological fields, the subsequent decade was marked by the demonstration of competitiveness of machine learning systems relative to established numerical weather prediction (NWP).

The decade opened with evaluations of feasibility. New datasets with standardized metrics enabled more rigorous comparison of approaches, and early results suggested that convolutional neural networks could match coarse-resolution NWP on short-term tasks while running at much lower cost [GRT22]. Weyn et al. [WDC20, WDCCC21] extended these approaches to global domains, showing multi-day forecasts and some subseasonal skill. In parallel, MetNet [SEH+20] demonstrated operationally

Year	Authors	Task / Variable	Architecture	Key Contribution
2020	Rasp et al. [GRT22]	Benchmarking forecasting	ResNet (baseline)	Standardized evaluation for data-driven forecasts.
2020	Weyn et al. [WDC20]	Global medium-range	U-Net (cubed sphere)	Stable multi-day forecasts, outperformed low-res NWP.
2020	Sønderby et al. [SEH+20]	Nowcasting precipitation	Attention CNN	1 km nowcasts up to 8 h; beat regional NWP at 7–8 h.
2020	Stengel et al. [SGHK20]	Climate downscaling	GAN	Adversarial super-resolution for wind/-solar data.
2021	Weyn et al. [WDCCC21]	Subseasonal	CNN ensembles	ML ensembles competitive with operational S2S models.
2022	Pathak et al. [PSH+22]	Global medium-range	AFNO (FourCastNet)	Fast 0.25° forecasts; near-parity with ECMWF IFS.
2022	Bi et al. [BXZ+22]	Global medium-range	3DEST (Pangu)	Match ECMWF HRES on 1–7 d forecasts.
2022	Keisler [Kei22]	Global medium-range	GNN	Early GNN for global forecasts; precursor to GraphCast.
2022	Lam et al. [LSGW+22]	Global medium-range	GNN (GraphCast)	Outperformed ECMWF HRES on 90% of variables.
2023	Chen et al. [CZZ+23]	Subseasonal	Cascaded Transformers	Skillful out to 15 d; ensembles rivaled ECMWF EPS.
2023	Nguyen et al. [NBK+23]	Multi-task climate	Transformer (ClimaX)	First foundation model; pretrained for transfer across tasks.
2024	Rasp et al. [PSGA+23]	Ensemble medium-range	Diffusion (GenCast)	First to beat ECMWF ENS at 15-d ensembles.
2024	Bodnar et al. [Bod24]	Multi-domain Earth system	Swin Transformer (Aurora)	Foundation model spanning atmosphere, ocean, air quality.
2024	Kochkov et al. [KYL+24]	Hybrid NWP/ML	NeuralGCM	ML + differentiable core; long forecasts with physics consistency.
2025	Asperti et al. [AMP+25]	Precipitation forecasting	DDIM Diffusion	Probabilistic forecasts with ensemble diffusion models.
2025	Merizzi et al. [MEL25]	Ensemble methods	DDIM Diffusion	Advanced variance control in diffusion ensembles.
2025	Bonev et al. [BKM+25]	Long-range ensembles	Spherical operator (FourCastNet3)	60-d ensembles; surpassed IFS-ENS.

Table 5: Selected influential neural network studies in meteorology during the 2020s, including thesis contributions. The decade saw a progression from benchmarks and feasibility to operationally competitive forecasts, probabilistic diffusion ensembles, and multi-domain foundation models.

relevant skill in precipitation nowcasting, while Stengel et al. [SGHK20] applied GANs for climate downscaling. These early contributions established neural networks as credible complements to physics-based models.

By 2022–2023, the field shifted from feasibility to direct competition with operational NWP. Landmark contributions such as FourCastNet [PSH+22], Pangu-Weather [BXZ+22], and GraphCast [LSGW+22] marked a step change: each demonstrated skill that matched ECMWF’s high-resolution deterministic forecasts across a majority of variables and lead times, while reducing computational costs by several orders of magnitude. FuXi [CZZ+23] extended this paradigm to subseasonal scales, maintaining ensemble-like skill out to 15 days. Collectively, these systems transformed machine learning–based forecasting from an experimental research line into a demonstrably efficient alternative, capable of providing high-resolution, global forecasts faster and more efficiently than traditional methods. Nevertheless, important questions remain as to whether such models can truly substitute numerical approaches. They are dependent on numerical reanalyses for training data, and without continual retraining may drift out of distribution as the climate evolves. Moreover, concerns persist within the scientific community regarding whether purely neural forecasts provide sufficient physical consistency and guarantees to be relied upon in high-risk decision-making contexts.

As models achieved parity with deterministic NWP, attention increasingly turned to the ability of neural approaches to represent the probabilistic nature of the atmosphere. GenCast [PSGA+23] introduced diffusion-based generative modeling for ensemble forecasting, producing calibrated 15-day global ensembles that outperformed ECMWF’s operational ENS. Hybrid models such as NeuralGCM [KYL+24] combined machine learning parameterizations with differentiable dynamical cores, extending forecast horizons while preserving physical consistency. This thesis contributes to this emerging line of research by exploiting diffusion ensembles for both high-quality downscaling [MAC24a, MEL25] and probabilistic forecasting [AMP+25], and by advancing methodological understanding of variance control in diffusion ensembles [MEL25]. These developments marked a second turning point: machine learning systems were no longer only replicating deterministic trajectories, but were capable of generating ensembles with realistic spread, sharper spatial features, and improved calibration relative to traditional stochastic perturbation methods.

In the late 2020s, research began to diversify into several distinct but complementary directions. One line of work focuses on foundation models, inspired by large language models, which are pretrained on heterogeneous climate and weather datasets and then fine-tuned for multiple downstream applications (e.g., ClimaX [NBK+23], Aurora [Bod24]). A second direction emphasizes physics-informed approaches, such as hybrid models and PINNs, which integrate conservation laws and physical consistency into neural architectures (e.g., NeuralGCM [KYL+24]). A third strand explores multi-variable learning, motivated by the interconnected nature of the Earth system; in this thesis we demonstrate that jointly downscaling multiple variables improves overall performance [ML25]. Together, these developments highlight a shift from single-task models toward more integrated frameworks that aim to unify accuracy, physical consistency, and cross-variable coherence.

Taken together, the research trajectory of the 2020s can be described in terms of three broad phases: an initial feasibility phase, in which neural networks proved capable of reproducing weather and climate dynamics; a competitiveness phase, in which machine learning systems exceeded the skill of operational NWP; and a generalization phase, in which generative and multi-domain models extended forecasting to probabilistic ensembles and integrated Earth system applications. Within less than a decade, neural networks advanced from proof-of-concept demonstrations to operationally viable, state-of-the-art forecasting systems. Rather than being confined to a single paradigm, current research now spans multiple paths — from foundation models to physics-informed approaches and multi-variable learning — positioning machine learning not merely as an accelerator of traditional NWP, but as a potential new basis for atmospheric and climate prediction.

2.3 Evolution of Downscaling Methods

In this section, we provide an outline of the most relevant literature in the field of downscaling with deep learning. We begin with an overview of conventional approaches, including statistical and dynamical downscaling as well as simple interpolation techniques, which have long defined the methodological backbone of the field. We then turn to neural approaches, starting from relatively simple non-spatial models based on multi-layer perceptrons (MLPs), and progressing through convolutional networks, transformer-based methods, and generative architectures such as GANs and diffusion models. This

progression illustrates both the historical development of downscaling methods and the increasing sophistication of deep learning techniques applied to climate and weather data.

2.3.1 Simple Interpolation Methods

Traditional super-resolution methods relied on simple interpolation schemes such as nearest-neighbor, bilinear, or bicubic interpolation, which estimate unknown values from neighboring points according to fixed deterministic rules [Key03, PKT07]. These approaches are computationally inexpensive and reproducible, but they typically produce overly smooth fields and perform poorly at large scaling factors, since they cannot reconstruct fine-scale variability that is absent from the coarse input. Despite these limitations, interpolation methods remain important reference baselines in both computer vision and climate downscaling. They are routinely included in benchmarking studies, where their performance provides a lower bound against which more complex models can be assessed. In practice, interpolation can also serve as a diagnostic tool: if a learned model does not outperform bicubic interpolation, the downscaling task may be ill-posed or the model mis-specified, while the performance gap with respect to interpolation provides a rough indication of task difficulty.

2.3.2 Conventional downscaling frameworks

The field of climate downscaling has traditionally been organized around two complementary methodological families: dynamical and statistical downscaling [MWI+10]. Both seek to bridge the scale gap between global climate information and local impacts, but they do so in fundamentally different ways. Dynamical approaches employ limited-area numerical models to explicitly resolve atmospheric processes at higher resolution, while statistical approaches rely on empirical relationships that link large-scale predictors to local climate variables. Together, these frameworks have provided the backbone of regional climate research for several decades, underpinning both scientific assessments and climate services [WWC+98, FBT07]. They also define the reference point against which more recent neural methods are developed and evaluated.

Dynamical downscaling Dynamical downscaling relies on limited-area numerical models that are nested within global climate models (GCMs) or reanalyses [Gio90]. Regional Climate Models (RCMs) typically operate at horizontal resolutions of 10–50 km, solving the governing equations of the atmosphere and land surface over a restricted domain. This approach is firmly grounded in physical principles and provides detailed representations of processes such as orographic precipitation, land–atmosphere interactions, and coastal circulations. As such, RCMs have been central to coordinated initiatives like CORDEX [GJA+09], where widely used models include RegCM, developed at the International Centre for Theoretical Physics (ICTP), and COSMO-CLM [RWH08], maintained by the CLM-Community.

At finer scales, Convection-Permitting Models (CPMs) reach horizontal resolutions of about 1–4 km, explicitly resolving deep convective processes such as thunderstorms and heavy rainfall events that must otherwise be parameterized [PLF+15]. This allows a more realistic simulation of precipitation extremes and storm dynamics, but it comes at very high computational cost. Both RCMs and CPMs are demanding in terms of computing resources, with CPMs in particular often restricted to short simulations, small domains, or limited ensembles. Nevertheless, they represent the most physically sound downscaling methodology, and models such as ICON (Icosahedral Nonhydrostatic), developed by the German Weather Service and the Max Planck Institute for Meteorology, are increasingly used in convection-permitting climate experiments over Europe [ZRRB15].

The corpus of simulations produced with RCMs and CPMs also provides the fundamental reference data used to develop and train neural super-resolution approaches [BMMG20, VKG+17]. Because high-resolution observational records are often sparse, many neural downscaling studies rely on synthetic training pairs created by coarsening model output and learning to reconstruct the original high-resolution fields. In this sense, the availability of RCM and CPM datasets is a prerequisite for the effective training of neural models. At the same time, there is increasing interest in reducing this dependency by designing neural architectures that can transfer across domains. Approaches such as domain adaptation, distribution matching, and transfer learning aim to limit the need for local high-resolution training data and to enable application across different spatial regions and temporal periods.

Statistical downscaling Statistical downscaling refers to methods that estimate local-scale climate variables (predictands such as temperature, precipitation, or wind speed) from large-scale atmospheric predictors, typically provided by global climate models (GCMs) or reanalyses. Unlike dynamical downscaling, which explicitly solves the governing equations of the atmosphere at higher resolution, statistical approaches rely on empirical relationships derived from data [WWC+98]. These relationships are usually established using historical observations and then applied to generate local projections from coarse-scale inputs.

A key assumption underlying statistical downscaling is that the empirical relationships identified in the past remain valid under future climate conditions (the stationarity assumption) [Mar13]. The approach also depends on the availability of sufficiently long and reliable observational records to train and evaluate the models.

Classical implementations include regression-based methods such as linear and ridge regression [KCD06], as well as stochastic weather generators like SDSM [WD13] and LARS-WG [SBLW02]. More flexible machine learning methods have subsequently been incorporated, including k-nearest neighbours [GCR05], random forests [PYK+19, HCSS16], and support vector machines [PSC16]. Ensemble approaches [LYZ+16, SMMB+17] are also widely used to increase robustness and quantify uncertainty.

Compared to dynamical downscaling, statistical approaches are computationally efficient and can be rapidly deployed over large ensembles or long time periods, making them attractive for climate services, impact assessments, and bias correction pipelines. They also allow for flexible incorporation of non-linear relationships and can be adapted to specific variables or regions. However, statistical methods are less physically transparent than dynamical models and remain sensitive to the assumption of stationarity, which may not hold under strong climate change signals. Their performance is also limited by the quality and availability of observational data, which constrains their applicability in data-sparse regions.

Neural methods are often formally placed within the category of statistical downscaling [BMMG20], as they share the same principle of learning empirical relationships between coarse predictors and fine-scale targets. Nevertheless, the increasing incorporation of physics-informed architectures and constraints suggests that this classification is becoming less clear-cut. Neural approaches can therefore be seen as extending beyond traditional statistical methods, occupying an intermediate space between purely empirical and fully dynamical frameworks.

2.3.3 Non-spatial and MLP-based Approaches

The first applications of neural networks to downscaling relied on machine learning methods that treated each grid point or station independently. These approaches typically modeled coarse-to-fine relationships using simple feedforward neural networks (multi-layer perceptrons, MLPs) or other regression-style learners without explicitly encoding spatial structure.

Early applications of MLPs in climate and environmental sciences date back to the 1990s and 2000s, where they were employed for seasonal forecasting [NC94], rainfall prediction [HGS97], or statistical downscaling of GCM outputs to local variables such as temperature and wind [SHLR00, SGK00]. Subsequent studies extended these ideas to various environmental prediction problems, including air quality modeling [GD98] and hydrological forecasting [CAB00], illustrating the general appeal of MLPs as flexible, data-driven regressors capable of capturing non-linear predictor–predictand relationships more effectively than traditional linear regression models.

However, the lack of explicit spatial modeling imposed clear limitations. By treating each location independently, MLPs and related methods were unable to enforce spatial consistency, capture coherent structures such as fronts or storm systems, or extrapolate fine-scale variability beyond what was present in the training data. As a result, while they often provide skillful statistical mappings and are still widely used in operational bias correction pipelines, their overall performance was progressively considered insufficient.

This limitation motivated the transition toward spatially-aware neural networks such as CNNs and U-Nets, which could learn hierarchical patterns across space rather than fitting independent pointwise regressions. Nevertheless, non-spatial approaches remain historically important, and they continue to serve as lightweight baselines for downscaling tasks where spatial consistency is less critical or spatial training data is limited.

2.3.4 CNN-based Approaches

Convolutional Neural Networks (CNNs) provided the first major breakthrough in neural super-resolution and climate downscaling. Their capacity to learn hierarchical spatial features made them a natural successor to interpolation and regression-based approaches. Early milestones in computer vision included the Super-Resolution Convolutional Neural Network (SRCNN) [DLHT14], which demonstrated that even a shallow CNN could surpass bicubic interpolation in restoring fine-scale image structures, and the Very Deep Super-Resolution network (VDSR) [KLL16], which used a deeper architecture and residual learning to substantially improve reconstruction accuracy. The Efficient Sub-Pixel Convolutional Network (ESPCN) [SCH⁺16] further showed that real-time super-resolution was feasible by replacing traditional upsampling with learnable sub-pixel convolutions.

Building on these foundations, residual architectures became central. Enhanced Deep Super-Resolution (EDSR) [LSK⁺17] and Cascading Residual Networks (CARN) [AKS18] adapted ResNet-style skip connections [HZRS16a] to stabilize training and enable deeper, more accurate networks. Architectural innovations such as dense connections [TLLG17, HSU18], multi-branch pathways [REKL17, HGL⁺18], and attention mechanisms [ZLL⁺18, CK17] diversified the CNN landscape, pushing performance further while targeting efficiency, interpretability, or robustness.

CNNs were quickly adopted in climate applications. DeepSD [VKG⁺17] represented a turning point, showing that CNN-based downscaling of precipitation produced more accurate results than traditional statistical methods. Subsequent works extended CNNs to temperature, wind, and solar radiation fields, often training on coarsened reanalysis or regional climate model outputs [BMMG20, TSW⁺22, KVLG⁺21, PMPN22]. These studies established CNNs as the de facto baseline for neural downscaling, combining efficiency with robust deterministic mappings between coarse and fine scales.

Despite their success, CNN-based models remain limited in their ability to capture extremes, generate probabilistic outputs, or enforce physical consistency. Their deterministic nature often leads to over-smoothed reconstructions and underestimation of variability [RT21], particularly variables with high frequency components, such as precipitation or wind speed. These limitations motivated the transition to generative architectures, such as GANs and diffusion models, which aim to introduce stochasticity, enhance perceptual quality.

2.3.5 GAN-based Approaches

A notable approach in super-resolution field came with the introduction of Generative Adversarial Networks (GANs) [GPAM⁺14], which shifted the focus from minimizing pixel-wise reconstruction errors to enhancing perceptual realism. GANs consist of two competing networks: a generator that produces high-resolution images from low-resolution inputs, and a discriminator that attempts to distinguish between generated and true high-resolution samples. Through this adversarial training framework, generators are encouraged to produce outputs that are not only accurate in terms of pixel fidelity but also visually convincing.

The seminal application of GANs to super-resolution was SRGAN [LTH⁺17], which combined a perceptual loss based on high-level VGG features with adversarial training. This enabled the generation of images with sharper textures and more realistic details than those produced by CNN-based models optimized solely with mean squared error. ESRGAN (Enhanced SRGAN) [WYW⁺18] further improved the architecture by introducing a Residual-in-Residual Dense Block (RRDB) and a relativistic discriminator, leading to state-of-the-art performance on perceptual quality benchmarks.

Following these milestones, GAN-based approaches diversified into various directions. Some works emphasized lightweight architectures for real-time inference, such as Swift-SRGAN [KK21] while others integrated attention mechanisms [ZLDQ19] or multi-scale discriminators to improve robustness across different resolutions [WYW⁺18]. Beyond computer vision, GANs have also been applied to climate and weather downscaling. As an example, [CR20] introduced a GAN-based framework for statistical downscaling of precipitation, showing improved skill in capturing spatial variability compared to conventional approaches.

Despite their successes, GAN-based super-resolution models face several challenges. Training is notoriously unstable, requiring careful balancing between generator and discriminator to prevent mode collapse or divergence [GPAM⁺14, SGZ⁺16]. Moreover, improvements in perceptual quality often come at the expense of pixel-level accuracy, leading to higher RMSE or MSE compared to CNN-based models [BM18]. This well-known perception-distortion trade-off poses particular difficulties in sci-

entific applications, where accurate numerical reproduction of physical quantities is more important than visual realism. Consequently, while GANs have been successfully applied to a variety of downscaling applications, their adoption in climate and weather modeling has remained limited relative to deterministic CNNs and transformers and, more recently, diffusion-based approaches.

2.3.6 Transformer-based Approaches

While CNNs dominated super-resolution research throughout the 2010s, the field has recently witnessed a paradigm shift towards architectures based on self-attention. Transformers, first introduced in natural language processing [VSP⁺17a], rely on multi-head self-attention mechanisms that enable each token in a sequence to directly attend to all others. This design allows the model to capture long-range dependencies more effectively than convolutional filters, which are inherently local. The adaptation of transformers to the visual domain began with the Vision Transformer (ViT) [DBK⁺20], which demonstrated that images could be processed as sequences of patches. Subsequent refinements, such as the Swin Transformer [LLC⁺21], introduced hierarchical representations and shifted windows to improve efficiency for high-resolution images, paving the way for applications in super-resolution.

In the super-resolution literature, transformer-based models have rapidly established themselves as state-of-the-art. The Image Processing Transformer (IPT) [CWG⁺21] was one of the first attempts to adapt transformers to low-level vision tasks, including SR. SwinIR [LCS⁺21] further demonstrated the effectiveness of hierarchical self-attention, achieving superior results across multiple SR benchmarks. More recently, Swin2SR [CCBT22] refined this approach, improving stability and performance through architectural and training enhancements. Other variants, such as Uformer [WCB⁺22] and ITTR [ZLZ⁺22], explore fully transformer-based encoder-decoder pipelines for image restoration and SR tasks.

Transformers are particularly well-suited for super-resolution because of their ability to model non-local correlations and complex global structures, which are often missed by CNNs. In addition, they naturally scale with data and computational resources, making them attractive in the era of increasingly larger scopes. Within the climate domain, transformer architectures are beginning to be explored for downscaling tasks, as illustrated in [NJB⁺23], where a ViT-based model is benchmarked against CNN counterparts for spatial refinement of reanalysis and observational datasets. Other studies have also reported encouraging results with ViT-based approaches across precipitation [CFL⁺22], wind components [ZDC⁺24], and regional downscaling tasks [LZVO25], often outperforming CNN-based baselines.

Overall, transformer-based approaches represent a major step forward in neural downscaling. By capturing long-range dependencies and leveraging scalable architectures, they now stand at the frontier of super-resolution research in both computer vision and climate applications.

2.3.7 Diffusion-based Approaches

A more recent development in generative modeling for super-resolution is the use of diffusion models. Originally introduced in the context of probabilistic generative modeling [SDWMG15, HJA20a, SME20a], diffusion approaches simulate a gradual corruption process in which noise is incrementally added to a sample until it becomes indistinguishable from pure noise. A neural network is then trained to reverse this process step by step, effectively learning how to denoise and reconstruct the data distribution. Compared to GANs, diffusion models provide more stable training dynamics and produce higher-fidelity samples without the risk of mode collapse.

The application of diffusion models to super-resolution tasks was pioneered by SR3 [SHC⁺22], which demonstrated their ability to synthesize highly realistic high-resolution images. Subsequent works such as SRDiff [LYC⁺22] extended these ideas to more efficient and robust designs, narrowing the gap between perceptual quality and reconstruction accuracy. Diffusion approaches have quickly established themselves as state-of-the-art in image generation and enhancement, benefiting from their strong probabilistic foundation and scalability with large datasets.

In the climate and weather domain, diffusion-based downscaling is a rapidly emerging area of research. Conditional diffusion models have been proposed for a range of tasks, including weather forecasting [AHP⁺24, MBC⁺23], downscaling of wind speed [AMP⁺23, MAC24a, MEL25], and regional climate data generation [WM24]. These studies highlight two key advantages of diffusion models in climate applications: their ability to generate physically consistent fields with sharp spatial detail,

and their robustness compared to GAN-based approaches, which often suffer from instability and loss of fidelity. More broadly, diffusion models are now regarded as state-of-the-art for capturing uncertainty, generating ensembles, and improving calibration in meteorological applications [MBC⁺24, LGWZN⁺24b, LLL⁺24, LCLG⁺24, WBB⁺23, LHN⁺23].

Overall, diffusion models represent the latest frontier in neural downscaling. Their generative capacity, stability, and flexibility make them particularly well-suited to producing realistic and accurate fine-scale climate information, positioning them as a promising successor to GAN-based approaches.

3 Methodological Foundations

This chapter establishes the methodological framework common to all studies presented in this thesis. It formalizes the downscaling problem and clarifies its connection to the super-resolution task as developed in computer vision. Building on this analogy, it defines the mathematical formulation of the problem and explores the key design dimensions that shape model performance, including architectural structure, pre-processing strategies, temporal and spatial context, stochasticity, and multi-task approaches.

The chapter then focuses on neural architectures, presenting their fundamental principles, mathematical foundations, and common design practices. It begins with fully connected networks, which provide the conceptual basis of deep learning and introduced backpropagation as the key training mechanism. Convolutional neural networks (CNNs) extend this foundation by introducing spatial locality and weight sharing, representing the most successful model structure for image-based tasks. Their encoder-decoder variants, particularly U-Nets and residual designs, form the backbone of modern downscaling architectures and are extensively used in this thesis.

Transformer architectures are then discussed, replacing convolution with self-attention mechanisms and gaining traction as a new paradigm for visual representation learning. The chapter then moves to non-deterministic methods, namely Generative Adversarial Networks (GANs) and Diffusion Models. GANs frame downscaling as a competition between a generator and a discriminator, enabling sharp and realistic reconstructions through adversarial learning, though they often suffer from training instabilities and high sensitivity to problem-specific configurations. Diffusion models, representing the current state of the art in non-deterministic generation, learn to reverse a gradual noising process, iteratively transforming pure noise into structured outputs. Their stability, probabilistic nature, and ability to represent uncertainty make them a highly effective method, and they constitute the main architectural framework employed in this thesis.

For all the presented architectures, the chapter discusses the underlying mathematical formulation, design space, and common implementation techniques. The discussion then extends beyond architectural design to cover key methodological aspects of deep learning, including loss functions, training procedures, and evaluation strategies. Finally, it reviews the software frameworks commonly used for implementing deep learning models, outlining their main characteristics and differences.

3.1 Problem Definition

The concept of super-resolution originates outside of climate science, with roots in computer vision and signal processing. The central goal was to recover fine-scale details from undersampled or blurred observations, whether in industrial quality control, medical imaging, satellite remote sensing, or digital photography. Early methods were largely statistical or interpolation-based, seeking to enhance resolution through carefully designed reconstruction schemes [TH84, PPK03]. These approaches highlighted a general scientific ambition: to generate more detailed representations of a phenomenon from limited information.

With the rise of deep learning, super-resolution underwent a transformation. With the advent of Convolutional Neural Networks (CNNs) [KSH12], followed by Vision Transformers (ViT) [DBK⁺20] and diffusion architectures [HJA20b] deep learning methods demonstrated the ability to synthesize highly realistic details, far surpassing earlier methods. In computer vision, these techniques became standard tools for tasks such as face reconstruction, text and image deblurring, and restoration of natural images. Super-resolution emerged as one of the hallmark successes of deep learning in vision.

It is in this context that climate science began to recognize the close analogy between super-resolution and meteorological downscaling. Technically, the two problems are identical: both involve an image-to-image transformation where the input is a low-resolution field and the output is its high-resolution counterpart. The only difference lies in the semantics of the “image”: in climate science, the pixels correspond to physical quantities such as temperature, wind, or precipitation on a geospatial grid.

This methodological alignment facilitated a natural transfer of techniques. Architectures first developed for natural image super-resolution were adopted for climate downscaling, where they quickly proved effective. CNN-based models improved upon simple interpolation by learning localized spatial features, and diffusion models showed unprecedented capacity to capture variability. Importantly, unlike in computer vision where success is often measured by perceptual quality, in meteorology these

models are evaluated against physical consistency. Neural downscaling models have demonstrated the ability to approximate aspects of the underlying atmospheric dynamics, producing fields that are not only sharper and more realistic but also consistent with large-scale circulation and conservation principles. Across many studies, such approaches consistently outperformed traditional statistical methods and operate at a much higher efficiency compared to dynamical downscaling models.

3.1.1 Mathematical Formulation

Super-resolution is a process in digital image manipulation that enhances image resolution. This problem can be defined as an ill posed inverse problem, where the task is to invert an unknown degradation function applied to a high resolution image. Given a low resolution image denoted by y and the corresponding high resolution denoted by x , we define the degradation process as:

$$y = \Theta(x, \rho_n) \quad (1)$$

where Θ is the degradation function, and ρ_n denotes the degradation parameters (such as the scaling factor, noise, etc.). In a real-world scenario, only y is available while no information about the degradation process or the degradation parameters ρ exists. Super-resolution seeks to nullify the degradation effect and recovers an approximation $\hat{x}$ of the ground-truth image x as:

$$\hat{x} = \Theta^{-1}(y, \rho_r) \quad (2)$$

where ρ_r are the parameters for the function Θ^{-1} . The degradation process is unknown and can be quite complex. It can be affected by several factors, such as noise, compression, blur and other artefacts. Therefore, most research works [AKB20, LGDG23, BWKN22] prefer the following degradation model over that of Equation 1:

$$y = (x \otimes k) \downarrow_s + n \quad (3)$$

where k is the blurring kernel and $(x \otimes k)$ is the convolution operation between the HR image and the blur kernel, $\downarrow_s$ is a downsampling operation with a scaling factor s .

Let us consider now the task of statistical downscaling in weather sciences. Statistical downscaling is a technique employed to convert broad-scale meteorological data, typically obtained from global climate models or reanalysis products, into finer, local-scale weather information. This approach relies on establishing statistical relationships between large-scale atmospheric variables (like temperature, pressure, and wind) and local-scale weather phenomena. This process is usually implemented via a sophisticated physical models, such as the regional reanalyses presented in Section 2.1.2 or the climate projections described in Section 2.1.4. These models have as input the low resolution ground truth plus the additional information necessary to compute the downscaling result. Thus, if we represent, coherently with the Super-resolution problem, the high resolution image as x , the low resolution images as y , the statistical downscaling model as HA and the additional information as ρ_{HA} , we can define the downscaling process as:

$$x = HA(y, \rho_{HA}) \quad (4)$$

We observe that this equation is closely related to Equation 2, which establishes a relationship between high-resolution and low-resolution images. This similarity effectively equates the task of super-resolution in image processing to the problem of statistical downscaling in meteorological studies.

The primary distinction between the two problems lies in the inverse nature of their low-resolution/high-resolution relationship. Contrary to the super-resolution process described in Equation 3, where the low-resolution image is derived from an unknown high-resolution counterpart, in statistical downscaling, the situation is reversed. Here, the high-resolution image is an enhanced version of the low-resolution image.

We can thus define the new super-resolution task as the problem of approximating the behaviour of the HA model. Considering that our goal is to proceed in a data driven manner and to avoid the utilization of additional information for the downscaling process, we define our approximate model without the use of ρ_{HA} .

$$\hat{x} = \Phi(y), \hat{x} \approx x \quad (5)$$

Where Φ is our super-resolution model and $\hat{x}$ is the approximation of the ground truth x generated by the statistical downscaling model.

3.2 Architectural Design Space

In this section, we examine the design space of the super-resolution problem by focusing on its fundamental structure rather than on specific neural architectures. In other words, we treat the architecture as a “black box” that takes inputs and produces outputs, and concentrate instead on how the problem itself is framed. Even at this level of abstraction, the range of possible formulations is broad, encompassing choices in resolution upsampling, border and padding treatments, temporal dependence, domain geometry, and the inclusion of auxiliary information. These early design decisions strongly influence both the attainable performance and the eventual architecture of the downscaling model. Figure 7 illustrates the simplest setup: a low-resolution field (LR) is fed to a neural model, which returns a super-resolved high-resolution field (HR).

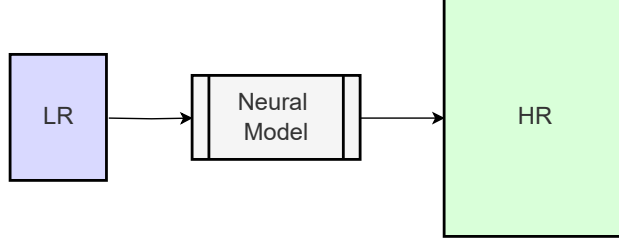


Figure 7: Super-resolution problem setup. A low-resolution field (LR, left) is fed into a neural model (center), which outputs a super-resolved field (HR, right).

3.2.1 Low-Resolution upsampling

Most neural networks for image-to-image tasks operate on inputs and outputs of the same size. This reflects the structure of encoder-decoder architectures, where the two parts largely mirror one another. In super-resolution, however, the low-resolution (LR) input and the high-resolution (HR) target differ in size. To bridge this gap, the LR field must be upsampled so that the model can operate on comparable spatial dimensions. This can be achieved in different ways:

- **Preemptive upsampling.** In this strategy, the LR data are upsampled as a preprocessing step (see Figure 8). Suppose we aim to map 64×64 inputs to 256×256 targets (a $\times 4$ scaling). Before training, we interpolate the LR data using bilinear, bicubic, or nearest-neighbor interpolation, so that the model sees a symmetric $256 \times 256 \rightarrow 256 \times 256$ mapping. This approach is simple and computationally efficient, and it makes the task compatible with standard encoder-decoder pipelines. It also provides access to the pre-upsampled fields, which can be useful for computing statistics, establishing baselines, or applying further preprocessing.
- **Model-internal upsampling.** A second option is to let the network handle the resolution mismatch directly, so that the mapping is inherently asymmetric (e.g., $64 \times 64 \rightarrow 256 \times 256$). Some architectures (such as ESPCN) are explicitly designed this way, but most encoder-decoders still require an upsampling stage at the beginning. Two main variants exist:
 - *Uninformed upsampling:* fixed operations such as bilinear or nearest-neighbor layers embedded directly in the network. This is equivalent to the preprocessing approach, but integrated inside the architecture for convenience or efficiency.
 - *Informed upsampling:* trainable operations such as transposed convolutions or sub-pixel convolutions. These not only expand the spatial dimensions but also allow the model to optimize how fine-scale structure is reconstructed. Their behavior depends strongly on kernel size, stride, and padding, which control the specifics of the upsampling operation. This flexibility may improve performance but comes at the cost of additional parameters and potential artifacts.

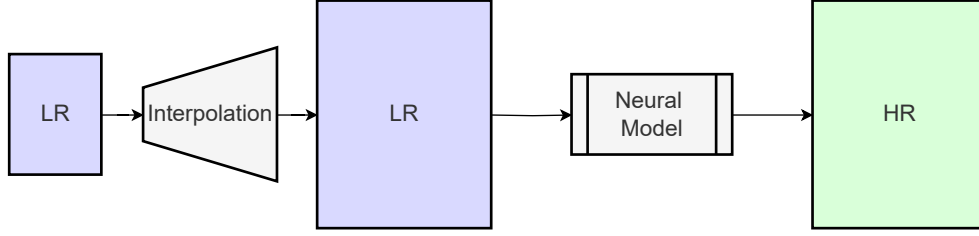


Figure 8: Super-resolution setup with external upsampling. The low-resolution input (LR) is first upsampled by interpolation (e.g., bilinear, bicubic, or nearest-neighbor), so that the neural model operates on images of the same size as the target high-resolution field (HR). This strategy is simple and efficient, and is widely used to make super-resolution compatible with standard encoder–decoder architectures.

3.2.2 Borders, padding and resolution

Border areas play a crucial role in super-resolution tasks. Convolutions are particularly sensitive to boundary conditions, and the choice of padding and stride can easily produce artifacts near the edges. Borders also matter when preparing data for encoder–decoder networks, such as U-Nets, which repeatedly halve and double spatial dimensions. Since these models typically apply downsampling 4–5 times, the input size must be divisible by 2^4 or 2^5 . This constraint is rarely satisfied by climate datasets, and various strategies are required to adjust domain sizes.

If the domain is square, the simplest option is often to resize the field to the nearest convenient dimension (e.g., resizing 253×253 to 256×256) and then invert this operation after super-resolution. This avoids information loss but may introduce slight artifacts, depending on how far the original resolution was from the target. The situation becomes more complex for rectangular domains, where naive resizing introduces distortions. In such cases, two families of strategies can be applied:

- **Preemptive Adjustments**

- *Cutting the image:* If arbitrary subdomains are acceptable, the simplest solution is to crop the image to the nearest valid size. This is common in experimental settings.
- *Adding mirrored or constant lines:* When cropping is not possible, missing rows/columns can be padded by mirroring or copying adjacent values. For large images, a handful of extra rows has negligible impact, and they can be removed after the operation.

- **Model-internal Adjustments**

- *Valid padding:* Convolution output size is given by

$$\text{Output size} = \left\lfloor \frac{W_{\text{in}} - K + 2P}{S} \right\rfloor + 1,$$

where W_{in} is the input size, K the kernel size, P the padding, and S the stride. For example, with $W_{\text{in}} = 260$, stride $S = 1$, and no padding ($P = 0$):

$$K = 3 \quad \Rightarrow \quad \text{Output size} = 258,$$

$$K = 5 \quad \Rightarrow \quad \text{Output size} = 256,$$

$$K = 7 \quad \Rightarrow \quad \text{Output size} = 254.$$

Thus, by selecting $K = 5$ we can map 260×260 inputs to 256×256 outputs without information loss, making them compatible with the remainder of the model.

- *Other custom designs:* If one is committed to handling arbitrary input sizes, custom encoder–decoders can be built. Skip connections often cause asymmetry when downsampling is imperfect, requiring explicit adjustments, and the final output layer must also be modified to match the target size. Many other ad hoc solutions exist, depending on the dataset.

Ultimately, although these adjustments may seem technical, they often determine whether a model is trainable and whether artifacts appear at the borders of the reconstructed fields.

3.2.3 Temporal independence and sequence modeling

Neural downscaling models are not restricted to the simple one-to-one mapping from a low-resolution (LR) field to a high-resolution (HR) field. Since meteorological datasets are inherently temporal sequences, additional time steps can be provided as input, offering the model more context. This is straightforward in forecasting, where past observations inform the trajectory of the field and help predict future states. In downscaling, the benefit is less obvious, but temporal information can still provide useful awareness of the system’s evolution, often leading to effective but problem-specific improvements in reconstruction quality.

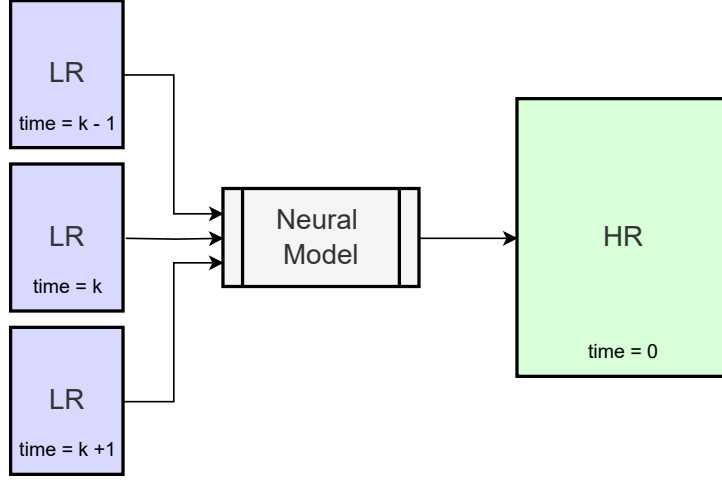


Figure 9: Sequence-to-one super-resolution setup. Multiple low-resolution fields (LR) at consecutive timesteps ($t = k - 1, k, k + 1$) are provided as input to a neural model, which outputs a single high-resolution field (HR) at the target timestep ($t = k$). This setup exploits temporal context to improve the reconstruction of the target field.

Several problem setups are possible:

- **1 to 1.** The baseline case: one LR input mapped to one HR output. This formulation is simple, intuitive, and often the first to be tested.
- **Sequence to 1.** Temporal context is included in the input, but the model still predicts a single HR field. A common strategy is to center a temporal window around the target timestamp. For instance, to predict HR at $t = 5$, the input could include LR fields at $t = 4, 5, 6$, providing information on how the field is evolving around the target. If future data are unavailable (e.g., in real-time applications), only past and present frames can be used. An example of this setup is reported in Figure 9.
- **Sequence to sequence.** The model takes multiple LR frames as input and produces multiple HR frames as output. This approach extends the sequence-to-one strategy and is especially useful when multiple target times are needed, improving efficiency by grouping predictions. Conceptually, it is related to multi-task learning, as will be discussed in the following sections.

In practice, the choice between temporal independence and sequence modeling depends on the application. Incorporating temporal information often yields incremental improvements rather than dramatic gains, but in some cases it can enhance physical consistency.

3.2.4 Spatial modeling

Designing super-resolution models for global domains or very high spatial resolutions can raise significant computational challenges. In practice, it may be infeasible to scale a model so that it maintains the same quality on a full domain as it does on a smaller region. One option is to seek more resources

(e.g., more powerful GPUs or distributed training on multiple nodes), but another is to decompose the domain into smaller subdomains and then stitch the results together. Depending on how subdomains are handled, three broad strategies can be distinguished:

- **One model per subdomain + stitching.** The domain is split into subdomains, and the same model design is trained separately on each. Transfer learning can also be used across subdomains. This approach allows larger models to be trained within limited resources and can work even with modest hardware. However, it introduces many problems: edge discontinuities, inconsistencies between subdomains, uneven performance, and difficulties in aligning the training process across all areas. Even with mitigation strategies such as overlapping patches, our experimental results suggest that this approach is rarely viable due to the cumulative issues.
- **Single model trained on all subdomains + stitching.** Instead of training separate models, a single model is trained on all subdomains concurrently. The model sees reduced spatial patches, taken (with or without conditioning) from the entire domain, and predictions are stitched together as above. Surprisingly, this setup can perform well, with the model adapting across heterogeneous patches. Still, border inconsistencies and dissimilarities at subdomain edges remain an issue.
- **Using a size-agnostic model.** In this approach, a fully convolutional architecture can be trained on reduced spatial domains and then applied to the full domain at inference. Such models are naturally agnostic to input size, since convolutions operate locally and can extend to larger grids without modification. However, they are not agnostic to scale: atmospheric phenomena differ across spatial scales, and training on small patches may limit the model’s ability to generalize to the larger-scale structures present in full-domain data.
- **Full single-domain modeling.** The simplest option is to scale the model to the full domain. While computationally heavier, this avoids stitching artifacts entirely. Code optimization and architectural improvements such as depthwise separable convolutions, as discussed in later sections, can make this feasible in practice. We consider this the most robust and practical option, as it avoids a large number of downstream issues.

In conclusion, splitting domains can appear attractive but typically hides more problems than it solves. A single full-domain model is generally preferable, and this is the approach adopted throughout this thesis.

3.2.5 Deterministic and generative outputs

Another relevant design choice is whether to model downscaling in a deterministic or probabilistic manner. Most models such as U-Nets and ViTs are intrinsically deterministic: given the same LR input they always produce the same HR output. These networks are efficient and practical, but they provide no information about uncertainty in the prediction. This can be a limitation when quantifying the likelihood of extreme events or assessing the reliability of a super-resolved field. Probabilistic approaches address this gap by explicitly modeling distributions of possible outcomes, often through the use of input noise in combination with LR information, as illustrated in Figure 10.

- **Deterministic models.** The simplest and most common approach is to use a deterministic network. These can be thought of as producing the “mean” prediction of the distribution, and they usually perform well for practical applications. However, because they are trained with point-wise losses such as MSE or MAE, they tend to smooth sharp gradients and under-represent extremes, a well-known limitation of this class of models. Most super-resolution architectures in the literature fall into this category.
- **Ensemble-style variability.** A quasi-probabilistic alternative is to generate multiple outputs from a deterministic model by perturbing the LR input (e.g., adding noise) or by varying the model itself (e.g., different hyperparameters, random seeds, or ensembles of weights/architectures). While this does not make the model intrinsically probabilistic, it provides a practical way to explore variability and uncertainty.

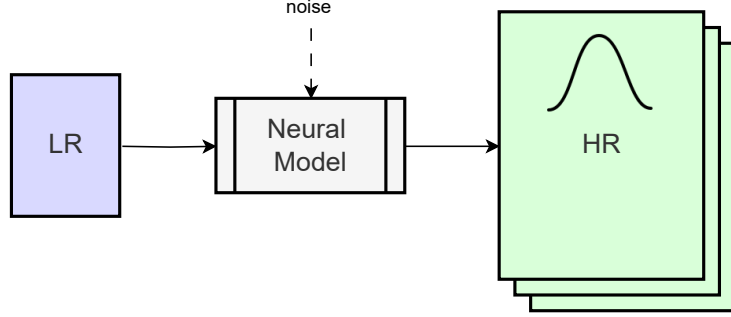


Figure 10: Non-deterministic super-resolution setup. A low-resolution field (LR) is combined with stochastic noise and processed by the neural model. Instead of a single deterministic output, the model generates multiple high-resolution samples (HR), which can be interpreted as draws from an underlying probability distribution (illustrated here by a Gaussian). This probabilistic formulation enables quantification of uncertainty and exploration of extremes beyond the mean prediction.

- **Intrinsically probabilistic models.** True probabilistic models, such as non-deterministic GANs and diffusion models, generate distributions of outcomes rather than a single deterministic result. The resulting ensemble can be used to study variance, quantify uncertainty, and examine the tails of the distribution (e.g., extreme events). These models are more complex to design and train, but they tend to produce sharper and more realistic fine-scale structures than deterministic U-Nets or ViTs, which suffer from smoothing.

In conclusion, deterministic models are efficient and effective for most applications, but when the task requires explicit information about uncertainty, probabilistic approaches are necessary.

3.2.6 Single Task, Multi-Task and Additional Elements

As discussed in Section 3.2.3, enriching the super-resolution model with additional information is a natural way to improve performance. Beyond temporal sequencing, other kinds of information can be introduced. Some of the most practical choices include static predictors such as topography or land-sea masks, dynamic variables at the same time step, or even metadata describing broader system conditions. These approaches are often designed in collaboration with climate scientists to ensure that the added fields provide positive information transfer. A related extension is multi-tasking, where the downscaling problem involves multiple variables at once. Altogether, this design space can be organized into several categories:

- **One variable at a time.** The baseline approach processes each variable separately. This avoids the risk of negative transfer between variables, but does not exploit potentially useful auxiliary information.
- **Adding auxiliary predictors (static fields, metadata).** A common extension is to include additional static fields such as orography, land-sea masks, or basin maps. These often have a strong influence on fine-scale variability: Topography (mountains, hills, lakes) becomes increasingly important as resolution increases, since fine-scale atmospheric variability is strongly shaped by local terrain. Including such information explicitly can therefore improve results beyond what the model can infer from the LR field alone. Metadata can also be incorporated. Time information (month, year) can help capture seasonal cycles or long-term changes, while climate scenario labels (e.g., CMIP trajectories) can condition the model on different forcings. Other domain-specific metadata may also be useful, such as astronomical cycles for tides.
- **Grouping dynamic variables.** Instead of static predictors, the model can be given multiple LR variables at the same time step. This provides richer physical context and can transfer useful information across fields. Two setups are possible:

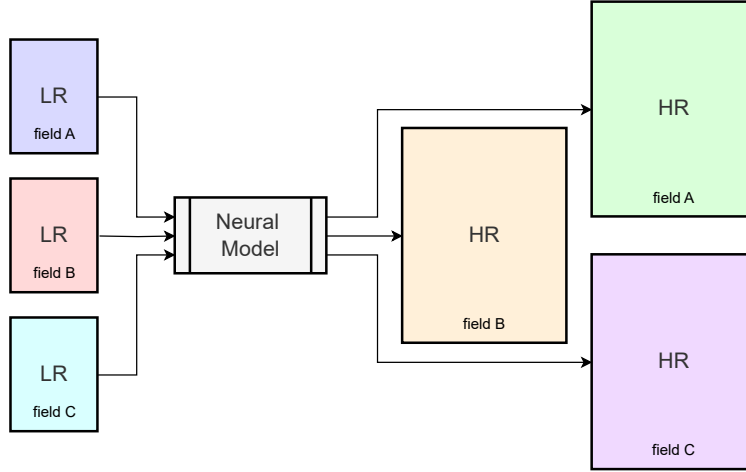


Figure 11: Multi-task super-resolution setup with three fields. At a given timestep, three low-resolution inputs (Field 1, Field 2, Field 3) are provided to the neural model. The model jointly predicts the corresponding high-resolution outputs for each field. This formulation allows the network to exploit cross-variable relationships and share information across tasks, potentially improving accuracy and efficiency compared to training separate models for each variable.

- *Multi-LR to single-HR*: In this setup, several LR predictors are provided together to generate a single HR target variable. The motivation is that different meteorological fields often contain complementary information: for instance, wind and pressure fields can help inform the downscaling of precipitation. By combining these predictors, the model can build a richer internal representation of the atmospheric state, leading to more accurate reconstructions. The trade-off is that the model becomes larger and training more computationally demanding, as the input dimensionality increases compared to the simple LR-to-HR case.
- *Fully multi-task*: In this setup, several LR predictors are jointly used to produce multiple HR outputs at the same time. The rationale is that by learning the downscaling of multiple variables within a single model, the network can exploit cross-variable relationships more effectively. For example, temperature, wind, and geopotential height fields are dynamically linked, and training them together can promote positive information transfer that improves overall skill. A multi-task setup can also be more efficient, since one shared model replaces several variable-specific ones. However, the design is more complex: optimization can be harder, loss functions must balance competing objectives across variables, and there is always a risk of negative transfer, where learning on one field degrades performance on another. An example of this approach is reported in Figure 11.

In conclusion, downscaling can be framed as a single-task or multi-task problem, with additional predictors ranging from static fields to dynamic variables and metadata. Each option offers potential benefits but also introduces trade-offs, and in practice the most effective choice is highly problem-dependent. Approaches can also be combined in hybrid formulations, tailoring the model setup to the specific application.

3.2.7 Other Aspects

Alongside the architectural design space elements introduced above, several additional considerations are important when setting up a downscaling problem. Among them, we highlight:

- **Cascading super-resolution steps.** A super-resolution task can be decomposed into multiple stages. For example, an overall $8\times$ scaling can be achieved through three successive $2\times$ models, either by reusing the same model at each step or by training separate models specialized for each scale. This approach is analogous to spatial splitting (Section 3.2.4), but here the “splitting” is

applied in scale rather than in space. Cascading is mainly useful for computational efficiency or for very large scaling factors, though it adds complexity to training pipelines.

- **Physically motivated constraints (PINNs).** A significant line of research in climate and weather modeling is to enforce physical consistency in neural models, in addition to data-driven learning. For downscaling, examples include requiring conservation of integrated precipitation between LR and HR, enforcing positivity of quantities such as wind speed through non-linear activation functions (e.g., ReLU in the last layer). These constraints can be implemented as *hard* constraints (directly imposed in the architecture) or as *soft* constraints (introduced via additional loss terms). They are an active research direction to improve the physical realism of neural downscaling.
- **Data augmentation.** Data augmentation is standard in computer vision, but less common in climate downscaling. Simple transformations such as rotations or flips break geophysical consistency: for instance, they disrupt topography, land-sea configuration, or the orientation of planetary waves. Similarly, introducing out-of-distribution elements often harms performance, since super-resolution models are highly sensitive to distribution shifts. Nevertheless, augmentation can be useful in data-scarce or particularly difficult tasks, provided it is applied with domain awareness.
- **Residual learning.** Instead of predicting the HR field directly, the model can learn to predict only the residual relative to a baseline. If HR is the ground-truth high-resolution field and $BASE$ is an interpolation or the output of an existing model, then the residual is

$$RES = HR - BASE.$$

Training on RES restricts the problem to the high-frequency information missing from the baseline. This often simplifies learning and improves performance in difficult tasks. The downside is that the model is then limited to correcting an existing baseline and may not capture broader physical relationships.

- **Handling of missing data.** Many observational datasets contain missing values (e.g., radar occlusions, satellite gaps, or land-only/sea-only availability). A common approach is to exclude missing areas from the loss, allowing the model to freely generate values in those regions. If missing areas vary across the dataset, the network can often fill them convincingly. If the same areas are always missing, it may be more practical to let the model generate arbitrary values there and crop them during postprocessing. More sophisticated strategies (e.g., masked convolutions, inpainting) exist but are beyond the scope of this section.

3.3 Fully Connected Networks

The multi-layer perceptron (MLP) is the canonical form of a fully connected neural network. An MLP is composed of a sequence of layers, each consisting of a set of neurons. Every neuron in a given layer is connected to every neuron in the previous layer, hence the term *fully connected*. The output of each neuron is a weighted sum of its inputs, passed through a nonlinear activation function such as the sigmoid, hyperbolic tangent, or rectified linear unit (ReLU).

Formally, given an input vector $a^0 = x \in \mathbb{R}^d$, the computations in layer l are defined as

$$z^l = w^l a^{l-1} + b^l, \quad a^l = \sigma(z^l),$$

where w^l is the weight matrix, b^l is the bias vector, and $\sigma(\cdot)$ is a nonlinear activation function applied elementwise. Stacking multiple such layers yields a deep MLP, capable of learning highly nonlinear mappings between inputs and outputs. At the final layer L , the output of the network is

$$\hat{y} = a^L,$$

which in downscaling corresponds to a high-resolution field or a set of meteorological variables.

Historically, MLPs were among the first neural architectures explored for climate and weather applications. They demonstrated the potential of machine learning to capture nonlinear relationships

that traditional linear regression or statistical methods could not. For example, early studies used MLPs to predict rainfall, temperature anomalies, or seasonal indices, often achieving higher skill than simple empirical models, as discussed in Section 2.2.1. These applications helped establish neural networks as a viable alternative for statistical downscaling.

3.3.1 Backpropagation

The backpropagation algorithm is the backbone of neural network optimization. It provides an efficient way to compute gradients of the cost function with respect to all network parameters, making it possible to train deep architectures with a vast number of parameters. While originally derived for multi-layer perceptrons (MLPs), the same fundamental structure underlies the training of all modern neural architectures, including convolutional and transformer-based models.

We can view the backpropagation algorithm as a process composed of three flows. First, the *forward* path through the network, where the weighted inputs and activations of each layer are computed and stored, until the final output is reached and the error term can be initialized. Second, the *backprop* phase, where the error is propagated backward layer by layer to compute the error δ^l at each stage. Finally, the *gradient* step, where the derivatives of the cost function with respect to weights and biases are calculated, providing the quantities needed to optimize the network parameters.

To better understand the backpropagation algorithm, it is useful to define some relevant quantities. We define the weighted input $z^l = w^l a^{l-1} + b^l$, which mirrors the standard definition of a neuron output without the activation. The activation is $a^l = \sigma(z^l)$, and the error is δ^l . The error for neuron j in layer l is defined as

$$\delta_j^l = \frac{\partial C}{\partial z_j^l},$$

that is, the variation of the cost function with respect to the neuron's weighted input. This quantity measures how much changing z_j^l would affect the overall cost, and is central to propagating information backwards through the network. Using the chain rule,

$$\frac{dy}{dx} = \frac{dy}{du} \cdot \frac{du}{dx},$$

we can obtain the derivatives of the cost function with respect to each weight and bias in the network, which are then used to optimize the model. The set of equations that expresses these relationships is sometimes referred to as the four backpropagation equations.

Specifically, in the *forward* block (see Equation (6)), we start from the input $a^0 = x$ and, for each layer $l = 1, \dots, L$, compute the weighted input and activation, $z^l = w^l a^{l-1} + b^l$ and $a^l = \sigma(z^l)$. The pair $\{z^l, a^l\}$ is cached for later use in the backward pass, while a^l is fed to the next layer to produce $\{z^{l+1}, a^{l+1}\}$. This process is repeated until the output layer is reached, where a^L is the network output $\hat{y}$.

At the end of the chain we initialize the output-layer error term with $\delta^L = \nabla_a C \odot \sigma'(z^L)$, i.e. the gradient of the cost with respect to the output activations, scaled elementwise by the derivative of the output activation; here $\odot$ denotes the Hadamard (elementwise) product.

The *backprop* line in Equation (7) then expresses the right-to-left flow of error signals: starting from δ^L , we obtain each preceding layer's error δ^l by multiplying the next layer's error by the transpose of the next layer's weight matrix and then applying the elementwise factor from the local nonlinearity. At layer $L - 1$, for example, this recursion takes the form

$$\delta^{L-1} = (w^L)^\top \delta^L \odot \sigma'(z^{L-1}).$$

Intuitively, this is just the chain rule: the transpose appears because sensitivities are pushed backward through the linear map $z^{l+1} = w^{l+1} a^l + b^{l+1}$, and the $\sigma'(\cdot)$ term accounts for the activation at the layer whose δ is being computed.

Finally, the *gradient* block in Equation (8) gives the parameter gradients used for learning: each bias gradient equals the corresponding error, $\partial C / \partial b_j^l = \delta_j^l$, and each weight gradient is the product of the destination neuron's error and the source activation (elementwise: $\partial C / \partial w_{jk}^l = \delta_j^l a_k^{l-1}$). In words: forward computes z^l, a^l ; backprop computes δ^l by chaining local Jacobians; and the last line turns δ^l and a^{l-1} into gradients for w^l and b^l .

$$\text{forward} : \rightarrow \left\{ \begin{array}{l} z^l = w^l a^{l-1} + b^l \\ a^l = \sigma(z^l) \end{array} \right\} \rightarrow \left\{ \begin{array}{l} z^{l+1} \\ a^{l+1} \end{array} \right\} \rightarrow \dots \rightarrow \delta^{(L)} = \nabla_a C \odot \sigma'(z^{(L)}) \quad (6)$$

$$\text{backprop} : \delta^l \leftarrow \dots \delta^{L-2} \leftarrow \delta^{L-1} = ((\omega^{l+1})^T) * \delta^{l+1} \odot \sigma'(z^L) \leftarrow \quad (7)$$

$$\text{finally} : \frac{\partial C}{\partial \omega_{jk}^l} = \delta_j^l * a^{l-1}, \quad \frac{\partial C}{\partial b_j^l} = \delta_j^l \quad (8)$$

We can thus use the gradients to update the values of weights and biases of the network—an operation which, when iterated over multiple inputs x , defines what we call the training loop, in which a network learns from data.

3.3.2 MLPs for Downscaling

Multi-layer perceptrons (MLPs) were historically among the first neural architectures applied to statistical downscaling. They demonstrated the capacity to capture non-linear relationships and, in early studies, achieved results that rivalled more traditional statistical approaches. Today, however, MLPs are rarely used beyond simple baselines or initial exploratory experiments.

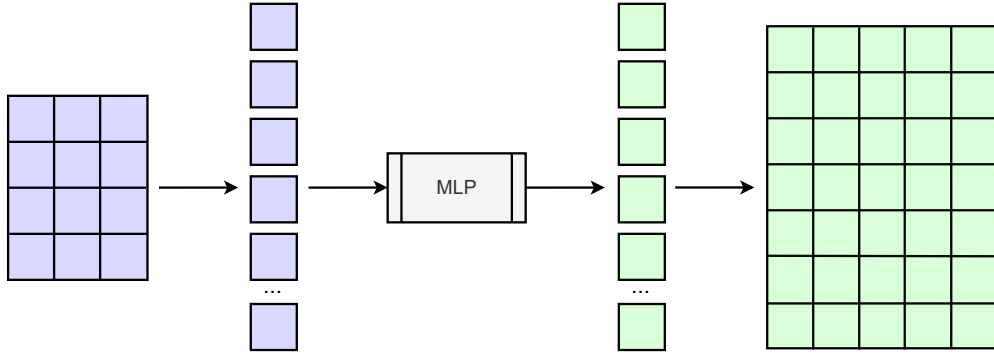


Figure 12: Illustration of an MLP-based downscaling workflow. A low-resolution field of size (3×4) is flattened into a one-dimensional vector of length 12, which is then passed through a multi-layer perceptron. The network produces an output vector of length 35, which is reshaped into the super-resolved high-resolution field of size (5×7) .

In practice, applying MLPs to downscale a spatial field requires flattening a gridded input (e.g., a 32×32 low-resolution field) into a one-dimensional vector. The network then learns a mapping from this vector to another flattened vector representing the high-resolution target field. This process treats the input as an unordered collection of values, ignoring the spatial arrangement of neighboring grid points.

The main limitations of MLPs for spatial data can be summarized as follows:

1. **No built-in notion of locality.** An MLP treats all input dimensions as independent, with no awareness of the underlying spatial grid. For example, it is not aware that pixel $(5, 5)$ is adjacent to pixel $(5, 6)$, or that nearby grid points are often correlated. As a result, local spatial structure must be inferred entirely from the data, requiring many more parameters and training samples than architectures that explicitly encode locality.
2. **Parameter inefficiency.** Fully connected networks scale very poorly with input and output size. Each input unit connects to every neuron in the first hidden layer, and this dense connectivity rapidly explodes in parameter count. For example, downscaling a 64×64 field (4,096 inputs) to a 256×256 field (65,536 outputs) with just a single fully connected layer would require more than 2.6×10^8 parameters. Adding multiple hidden layers only increases this burden. Such large parameter counts make training computationally expensive, memory intensive, and prone to overfitting, especially when compared to convolutional networks, which reuse the same filters across space.

3. **Translation sensitivity.** A spatial feature (e.g., a cyclone) shifted by one grid point corresponds to a completely different representation in the flattened vector. The MLP has no inherent mechanism to generalize across such shifts.
4. **Lack of inductive bias.** Convolutional architectures explicitly encode properties such as locality and translation equivariance. MLPs lack these priors, treating each input dimension independently despite the strong spatial structure of geophysical fields.

Despite these drawbacks, there remain scenarios where MLPs may be appropriate. For example, when downscaling a sparse set of locations that are not strongly constrained by spatial relationships, more complex architectures (such as convolutional or transformer-based models) may not provide additional benefit. In such cases, the simplicity of an MLP can make it a competitive and efficient solution.

3.4 Convolutional Architectures

Convolutional neural networks (CNNs) are the cornerstone of modern neural architectures for spatial data. Unlike fully connected layers, where each output depends on every input element, convolutions restrict the interaction to local neighborhoods. This design introduces two key advantages: *locality*, which ensures sensitivity to nearby interactions, and *weight sharing*, which applies the same filter across the entire domain. Intuitively, convolution can be seen as connecting each neuron only to a restricted region of the input field, enabling the network to specialize in detecting recurring spatial patterns such as edges, textures, or meteorological fronts, rather than redundantly processing the entire field at once.

Since their introduction, CNNs have been a major success story in machine learning, underpinning most of the progress in computer vision and the processing of structured spatial data. Their ability to extract multiscale features with relatively few parameters has made them the default choice for image analysis.

Building on these foundations, the most successful image-to-image architectures are based on *encoder-decoder* designs, of which the U-Net is the canonical example. U-Nets have become the standard backbone for dense prediction tasks because they combine the ability to capture large-scale context with the preservation of local detail. As such, they will play a central role in the approaches presented in this thesis, both as baseline models and as key components within more advanced generative architectures.

3.4.1 Foundations of Convolution

In the two-dimensional case, the convolution operation consists of sliding a kernel (i.e. a matrix of learnable parameters) over the input and computing local weighted sums. Each position of the kernel produces one element of the output, and by repeating this process across all positions, the convolution generates an entire output field. Formally, if $X \in \mathbb{R}^{H \times W}$ is the input and $K \in \mathbb{R}^{k_h \times k_w}$ is the kernel, the output at location (i, j) is given by

$$(X * K)_{i,j} = \sum_{m=1}^{k_h} \sum_{n=1}^{k_w} X_{i+m, j+n} K_{m,n}.$$

This expression describes the local weighted sum at a single location; the convolutional layer extends it to all (i, j) by shifting the kernel across the field.

To fully specify a convolutional layer, several quantities must be introduced. The *kernel size* (k_h, k_w) defines the spatial extent of the neighborhood that contributes to each output value, effectively controlling how much of the input the filter "sees" at once. The *stride* (s_h, s_w) determines how far the kernel moves across the input at each step; larger strides mean fewer evaluations and therefore a coarser output. In order to control the alignment between input and output sizes, one often introduces *padding*, which extends the input with artificial borders before applying the convolution. Padding can be also exploited to adjust the spatial dimensions of the input and output fields, as described in Section 3.2.2. A related parameter is *dilation*, which inserts gaps between kernel elements and thereby increases the receptive field without increasing the number of parameters.

Together, these quantities govern the relationship between the input and the output of a convolutional layer. For an input of height H and width W , convolved with a kernel of size (k_h, k_w) , stride (s_h, s_w) , and padding (p_h, p_w) , the output dimensions are given by

$$H_{\text{out}} = \left\lfloor \frac{H - k_h + 2p_h}{s_h} \right\rfloor + 1, \quad W_{\text{out}} = \left\lfloor \frac{W - k_w + 2p_w}{s_w} \right\rfloor + 1.$$

This expression makes explicit how design choices such as increasing the stride or changing the padding directly affect the resolution of the output field.

When extending the operation to multiple channels, each convolutional layer maps C_{in} input channels to C_{out} output channels, each associated with its own kernel. The number of learnable parameters is therefore

$$\text{num params} = C_{\text{out}} \cdot (C_{\text{in}} \cdot k_h \cdot k_w + 1) \quad (9)$$

where the additional term accounts for a bias parameter per output channel. From Equation (9) it is evident that processing large fields via convolution uses a vastly inferior number of parameters compared to fully convoluted networks.

Convolutions are translation equivariant: a shift in the input results in a corresponding shift in the output. This inductive bias makes convolutions particularly well-suited for images and geophysical fields, where the same structures may appear in different locations.

Training convolutional layers follows the same backpropagation principle described in Section 3.3.1. The main difference lies in the weight sharing mechanism: a single kernel is applied across many spatial locations, and during backpropagation its gradient is accumulated from all of these applications. In practice, this means that each kernel weight receives contributions from multiple parts of the input field. This adaptation of backpropagation not only reduces the number of parameters to be learned compared to fully connected layers, but also improves statistical efficiency by encouraging the learned filters to generalize across the entire domain.

3.4.2 Convolutional design elements

Convolutions provide the most effective way to design neural networks that operate directly on spatial fields. Unlike fully connected layers, which treat the input as a single unstructured vector, convolutional layers preserve the spatial organization of the data while remaining computationally efficient. This makes them particularly well suited for field-to-field mappings, such as climate downscaling or image-to-image translation, where the task is to transform one grid into another while retaining spatial coherence.

The design of convolutional networks involves several important choices. One is the size of the *receptive field*, which determines how much context each output unit has access to. Deeper networks naturally enlarge the receptive field, enabling the model to combine local and global information. Formally, the receptive field of a stack of convolutional layers can be computed recursively. If r_ℓ denotes the receptive field at layer ℓ , k_ℓ the kernel size, and s_ℓ the stride, then

$$r_\ell = r_{\ell-1} + (k_\ell - 1) \cdot \prod_{j=1}^{\ell-1} s_j,$$

with $r_0 = 1$. This expression shows how the receptive field grows with network depth, kernel size, and stride: larger kernels increase it additively, while strides propagate the effect multiplicatively across layers.

In the context of meteorological downscaling, the receptive field plays a major role in determining how the value at one location depends on values at others. If such interdependence is expected to be significant, the network should be designed with a sufficiently large receptive field to capture it. Conversely, if the aim is to restrict long-range interactions and ensure that dependencies remain physically consistent with the speed of atmospheric propagation, the network can be constructed with a more limited receptive field.

Another important design dimension is the trade-off between *depth* and *width*. The depth of a convolutional network refers to the number of convolutional layers stacked in sequence. Increasing

depth enlarges the effective receptive field and allows the network to capture hierarchical representations, where early layers detect simple local features and deeper layers combine them into increasingly abstract patterns. By contrast, the width of a network refers to the number of kernels (or feature maps) within each layer. A wider architecture processes more parallel features at the same level of abstraction, enabling richer representations of the input field. In practice, depth tends to enhance the model’s capacity to integrate information across scales, while width provides diversity of learned patterns at each scale.

For example, in downscaling tasks, depth may allow the model to connect large-scale circulation patterns with local variability, while width, on the other hand, enables the representation of multiple fine-scale structures simultaneously. Balancing these two dimensions is therefore a central design choice: networks that are too deep may suffer from vanishing gradients or excessive computational cost, whereas networks that are too wide may become prone to overfitting.

Convolutions can also be used to manipulate the resolution of the fields. By adjusting the stride or padding, the spatial dimensions of the output can be reduced, a process often referred to as downsampling. Conversely, when the goal is to increase spatial resolution, one may use *transposed convolutions* (sometimes called deconvolutions). The idea is to reverse the shrinking effect of a standard convolution: instead of skipping positions when the stride is larger than one, the input is conceptually expanded by inserting zeros between its elements, and a normal convolution is then applied to this expanded representation. In this way, the feature map is enlarged while the network still learns local filters to fill in the additional detail.

Formally, the output size of a transposed convolution with input size H_{in} , kernel size k , stride s , and padding p is given by

$$H_{\text{out}} = (H_{\text{in}} - 1) \cdot s - 2p + k,$$

with an analogous expression for the width. This formula highlights that strides, which reduce resolution in standard convolutions, instead stretch the feature map in the transposed case.

Convolutional blocks, analogously to fully connected blocks, are usually combined with nonlinear activation functions, such as ReLU or GELU, which allow the network to approximate complex mappings. Additionally, Normalization layers, including Batch Normalization, Layer Normalization, or Group Normalization, are often inserted after convolutions to stabilize training dynamics and improve generalization. Finally, connections across layers—whether residual shortcuts or skip links—are frequently used to facilitate optimization and to preserve spatial detail, setting the stage for the encoder–decoder architectures discussed in the following section.

Element	Description
Kernel size	Size of local neighborhood captured by the filter.
Stride	Step size of the kernel; controls downsampling.
Padding	Preserves spatial size and edge information, change field dimensions
Dilation	Enlarges receptive field without extra parameters.
Receptive field	Input region influencing each output unit.
Depth	Number of layers; captures hierarchical features.
Width	Number of filters per layer; increases feature diversity.
Normalization	Stabilizes training (e.g., BatchNorm, LayerNorm).
Activation	Adds nonlinearity (e.g., ReLU, GELU).
Upsampling	Increases resolution (transposed conv, interpolation).

Table 6: Concise summary of convolutional design elements.

3.4.3 Encoder–Decoder and U-Net Style Designs

Practically, convolutional architectures are rarely implemented as a simple sequence of layers. Instead, they commonly adopt an *encoder–decoder* structure, where the input is progressively compressed through a series of downsampling operations into a compact latent representation at the bottleneck, and

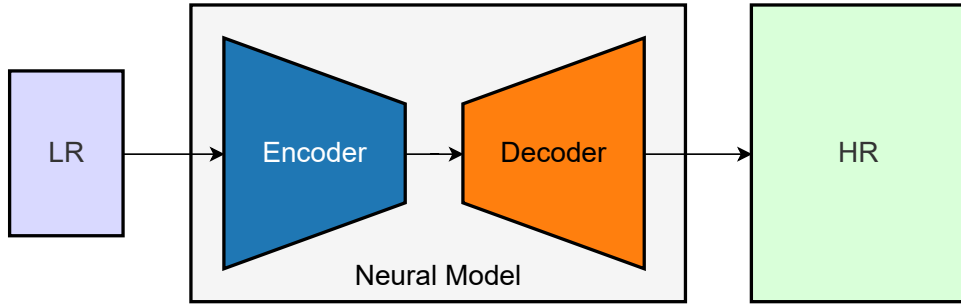


Figure 13: Schematic representation of an encoder–decoder architecture for image-to-image tasks. The low-resolution (LR) input on the left is compressed by the encoder into a compact representation, which is then expanded by the decoder to produce the high-resolution (HR) output on the right.

then expanded back to the original resolution by the decoder. This compression–expansion paradigm forces the network to learn both efficient encodings and effective reconstruction strategies, making it particularly well suited for dense prediction tasks such as forecasting and climate downscaling. By capturing multiscale features and restoring fine details, encoder–decoder designs provide a flexible foundation for modern convolutional architectures.

A canonical realization of this principle is the *U-Net* architecture. U-Nets are characterized by a symmetric encoder–decoder structure centered around a bottleneck. A key innovation is the use of *skip connections*, which directly link feature maps from the encoder to the corresponding stages of the decoder. These connections help recover fine-scale information that would otherwise be lost during downsampling, making U-Nets particularly effective for tasks requiring high spatial fidelity.

Downsampling in the encoder and upsampling in the decoder can be implemented in several ways, each with its own advantages and limitations:

- **Downsampling methods:**

- *Max pooling*: Selects the maximum value within a local window (e.g., 2×2). Simple, efficient, and widely used, non-trainable.
- *Average pooling*: Replaces each local window with its average value. Preserves more information than max pooling but may oversmooth details.
- *Strided convolutions*: Apply a convolution with stride $s > 1$, which reduces spatial resolution while still learning trainable filters. Often preferred in modern architectures as they integrate downsampling with feature extraction.

- **Upsampling methods:**

- *Nearest-neighbor interpolation*: Expands the feature map by duplicating neighboring values. Simple and fast, but can create blocky artifacts; typically followed by a convolution for refinement.
- *Bilinear interpolation*: Uses linear interpolation across both axes to produce smoother results. Also often followed by a convolution to recover detail.
- *Transposed convolutions*: Insert zeros between input elements and apply a standard convolution, producing learnable upsampling. Flexible but prone to checkerboard artifacts if not carefully designed.
- *Sub-pixel convolution*: First generates multiple feature channels, then rearranges them spatially (pixel shuffle). Efficient and widely used in super-resolution, but less common in segmentation tasks.
- *2D upsampling + convolution*: A hybrid strategy where interpolation (nearest or bilinear) is applied first, followed by a convolutional layer to refine the upscaled representation. Often avoids artifacts while retaining trainability.

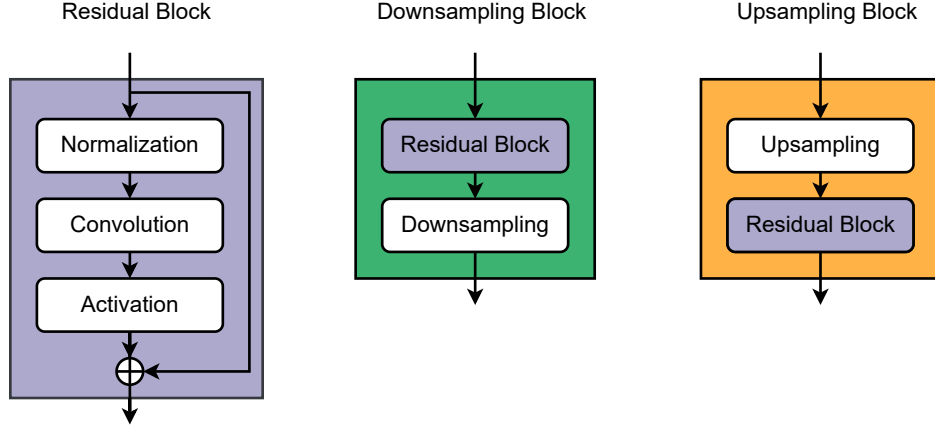


Figure 14: Illustration of the basic building blocks used in ResNet-style encoder–decoder architectures. The *Down block* (green) reduces spatial resolution while increasing feature richness, the *Residual block* (purple) serves as the fundamental processing unit, and the *Up block* (orange) restores spatial resolution while refining the representation.

Residual blocks are a powerful design element that substantially improve the performance and trainability of convolutional networks. In a plain convolutional block, the input is passed through a sequence of convolution, normalization, and activation layers, and the transformed output is then fed to the next stage. Residual blocks extend this idea by introducing a skip connection: the input is added directly to the block’s output. In its simplest form, the operation can be written as

$$y = F(x) + x,$$

where x is the block input and $F(x)$ represents the sequence of transformations (convolutions, normalizations, and activations). This formulation allows the network to learn residual mappings—corrections on top of the identity—rather than the full transformation itself. The result is easier optimization, faster convergence, and the ability to train much deeper networks without suffering from vanishing gradients.

An important property of residual blocks is their flexibility: if the learned residual $F(x)$ is close to zero, the block simply preserves its input, effectively bypassing the transformation. If, on the other hand, substantial modifications are needed, the residual path applies the required changes. In this way, residual blocks act as implicit gates, letting the network decide when to preserve information and when to transform it.

In modern ResNet-style architectures, residual blocks are typically combined with downsampling or upsampling operations to form *Down blocks* and *Up blocks*, which serve as the main structural elements of encoder–decoder designs. A simple illustration of these three components (Residual, Down, and Up blocks) is reported in Figure 14.

Several implementation variants exist. In the original formulation, the skip connection is added after a pair of convolution–activation layers, typically with Batch Normalization in between. Later refinements introduced *pre-activation*, where normalization and activation are applied before the convolutions, which further stabilizes training. Depending on the dimensionality, the skip connection can be implemented as a simple identity mapping or as a 1×1 convolution to match channel dimensions. These residual structures have become a standard component of modern CNN architectures and are often combined with encoder–decoder designs, including U-Nets, to enhance representational power and stability.

Over time, many extensions of the basic U-Net and residual designs have been proposed to improve efficiency and accuracy. A common variant is the *Residual U-Net*, which integrates residual blocks into the encoder–decoder structure. An example of a typical res-net is illustrated in Figure 15. This combination leverages the stability and depth scalability of ResNets while retaining the spatial detail recovery of U-Nets. In some implementations, *residual scaling* is also applied, where the residual branch is multiplied by a small factor before being added back to the main path, improving training stability in very deep networks.

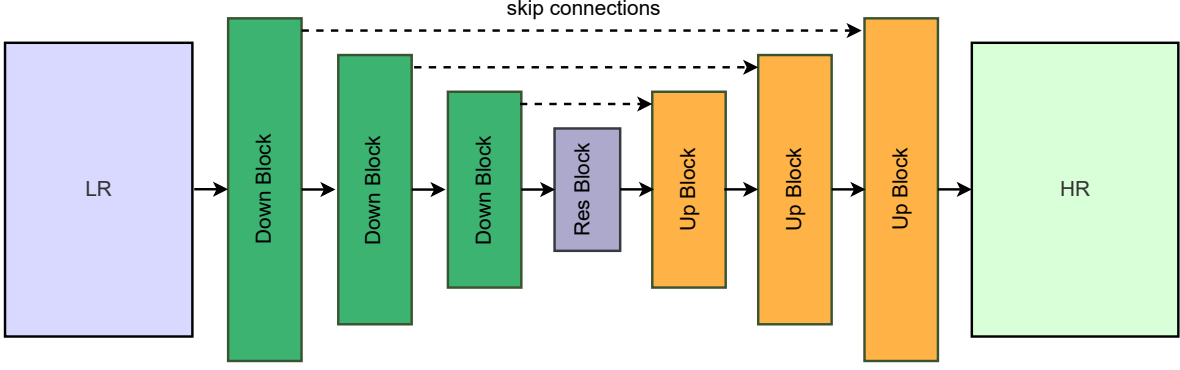


Figure 15: Schematic of a residual U-Net architecture built with standard blocks. The encoder (green Down blocks) progressively reduces spatial resolution while expanding the feature space. A bottleneck of Residual blocks (purple) processes the compressed representation. The decoder (orange Up blocks) restores spatial resolution, with skip connections linking encoder and decoder stages to recover fine-scale information.

Another important family of extensions introduces *attention mechanisms*. Modules such as squeeze-and-excite blocks adaptively recalibrate feature channels by modeling interdependencies, while CBAM (Convolutional Block Attention Module) and attention gates combine channel- and spatial-level weighting to guide the network’s focus toward the most relevant regions. These mechanisms enhance the representational capacity of convolutional architectures by allowing the model to prioritize informative features and suppress irrelevant ones.

Finally, efficiency-oriented modifications include the use of *depthwise separable convolutions*, which factorize a standard convolution into a depthwise convolution (applied independently to each channel) followed by a pointwise convolution (1×1). This dramatically reduces the number of parameters and computational cost while maintaining strong representational power, and has been widely adopted in lightweight architectures.

U-Nets have become the de facto backbone in climate and meteorological applications because their design aligns closely with the structure of geophysical fields. The encoder captures large-scale circulation patterns by progressively enlarging the receptive field, while the decoder, supported by skip connections, preserves and reconstructs local detail that is crucial for surface variables. This balance makes U-Nets particularly effective for downscaling tasks, and they will be used extensively throughout this thesis: either as baselines, as fully developed models, or as fundamental components within more advanced architectures such as diffusion models.

3.5 Transformer-Based Architectures

Transformers were introduced as a sequence-to-sequence architecture for machine translation, and they have since become the dominant paradigm in a wide range of machine learning domains. In the imaging domain, their success stems from replacing convolution with a fully attention-based mechanism, which allows efficient modeling of dependencies across sequences.

At the core of the transformer lies the *self-attention* mechanism. Given an input sequence $X \in \mathbb{R}^{n \times d}$ of n elements with embedding dimension d , each element is linearly projected into a query $Q = XW_Q$, key $K = XW_K$, and value $V = XW_V$, with learnable weight matrices $W_Q, W_K, W_V \in \mathbb{R}^{d \times d_h}$. The attention weights are computed by comparing queries and keys through a scaled dot product:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_h}}\right)V, \quad (10)$$

where d_h is the dimension of each head. The scaling factor $\sqrt{d_h}$ prevents excessively large dot products and stabilizes gradients during training. This operation produces a weighted combination of values in which every element can attend to all others, irrespective of distance. In practice, multiple attention heads are used in parallel:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W_O, \quad (11)$$

with h heads and an output projection W_O .

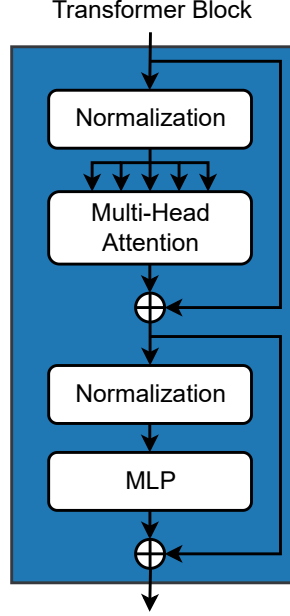


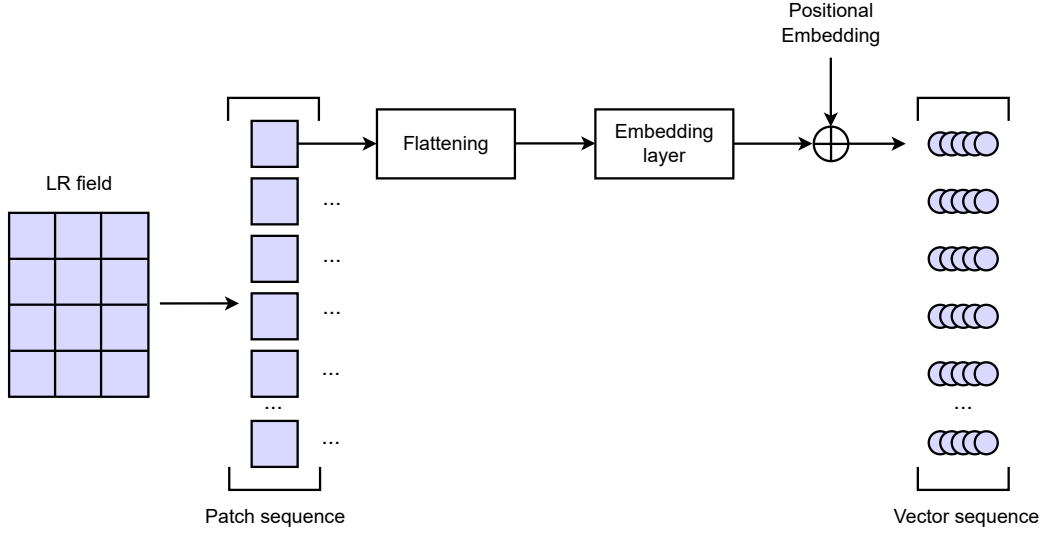
Figure 16: Structure of a standard transformer block. The input is first normalized and passed through a multi-head self-attention layer, followed by a residual (skip) connection. A second normalization precedes a position-wise feed-forward network, typically a two-layer MLP with GELU activation, and another residual connection. This sequence of normalization, attention, feed-forward processing, and skip connections constitutes the core unit that is stacked to build deep transformer architectures.

The core structural unit of transformer architectures is the *transformer block*. A standard block is composed of two main stages. In the first stage, the input is normalized and passed through a multi-head self-attention layer, after which a residual (skip) connection adds the original input back to the result. In the second stage, the output is normalized again and propagated through a position-wise feed-forward network, typically implemented as a multilayer perceptron with a GELU activation in the hidden layer. A second residual connection ensures stable optimization. This sequence of normalization, attention, feed-forward processing, and skip connections, illustrated in Figure 16, is repeated across stacked blocks, allowing the model to learn hierarchical representations. Unlike CNNs, where hierarchical features emerge from progressively larger receptive fields, transformers achieve a similar effect through successive layers of global attention, providing inductive biases that are more flexible but less spatially constrained.

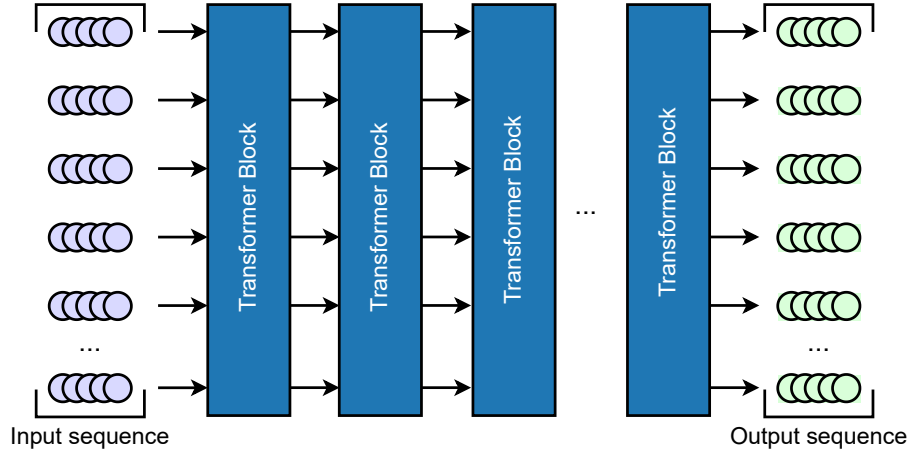
Since self-attention is inherently permutation-invariant, transformers require additional mechanisms to encode the order or structure of the input. This is typically achieved through *positional encodings*, which inject information about sequence order into the embeddings. While originally sinusoidal encodings were proposed, learned positional embeddings and relative positional encodings are now widely used.

3.5.1 Transformers for Image Super-Resolution

The application of transformers to images is built on the principle introduced by the Vision Transformer (ViT). An image is first divided into non-overlapping patches, which are flattened and embedded into vectors of fixed dimension, as illustrated in Figure 17a. The resulting sequence of patch embeddings is then propagated through a cascade of transformer blocks, as shown in Figure 17b, where self-attention captures dependencies across patches. In this way, spatial information is reformulated into a sequence representation and returned as an output sequence, enabling the network to exploit its ability to model long-range interactions.



(a) Patch embedding process in transformer-based super-resolution models. A low-resolution (LR) field is divided into non-overlapping patches, flattened, projected into embeddings, combined with positional encodings, and then passed to the transformer.



(b) Sequence-to-sequence processing with transformers. An input sequence is propagated through transformer blocks, producing an output sequence.

Figure 17: Overview of transformer-based image translation. (a) Patch embedding process. (b) Sequence-to-sequence processing.

For super-resolution, this patch-based formulation adapts naturally to the downscaling task, where a low-resolution input is mapped to a high-resolution output. Several architectural variants have been explored. A straightforward approach is the use of a pure transformer encoder–decoder, in which both the encoding and reconstruction are handled by transformer layers. More practical are hybrid designs, where a transformer encoder extracts global dependencies from the low-resolution input, and a convolutional decoder (often U-Net–like) reconstructs the high-resolution field efficiently while preserving fine-scale structure.

Further developments extend this paradigm. The Swin Transformer introduces hierarchical processing with shifted windows, reintroducing a degree of locality and improving efficiency on high-resolution inputs. Other architectures integrate transformers into U-Net designs more directly, for example by replacing the bottleneck with transformer blocks or by introducing transformer-based skip connections. These hybridizations aim to combine the strengths of global attention with the efficiency and inductive biases of convolutional layers.

Overall, transformer-based approaches provide greater flexibility in learning complex spatial dependencies, at the cost of higher data requirements and reduced inductive bias compared to purely convolutional models. When sufficient training data are available, and when the relationship between input and output fields depends strongly on long-range interactions, these models have the potential to outperform traditional CNN-based super-resolution architectures, particularly in climate downscaling where global dependencies often play a decisive role.

3.6 Generative Adversarial Networks (GANs)

The generative adversarial network (GAN) framework is based on an adversarial game between two neural networks: a generator G and a discriminator D [GPAM⁺14]. The generator is tasked with producing samples that resemble the target distribution, in the case of super-resolution, high-resolution (HR) fields, while the discriminator learns to distinguish between real HR samples from the training set and synthetic ones produced by the generator.

Architecturally, the generator is often designed as a convolutional decoder that transforms a lower-dimensional representation into a structured HR output. In the context of downscaling, G typically receives a low-resolution (LR) field as input and maps it to a plausible HR realization. Depending on the application, encoder–decoder structures such as U-Nets are commonly employed, as they allow the network to combine global context with local detail. The discriminator, on the other hand, is generally a classifier built with convolutional layers, producing a scalar output that reflects the probability that its input comes from the real dataset rather than from G . While convolutional networks are the dominant choice for both G and D , in principle any differentiable architecture can be used, including fully connected networks.

Training proceeds as a two-player min–max optimization problem. The discriminator seeks to maximize its classification accuracy, while the generator is optimized to minimize the discriminator’s ability to tell apart real and generated samples. Formally, this can be expressed as:

$$\min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}} [\log D(x)] + \mathbb{E}_{z \sim p_z} [\log (1 - D(G(z)))]. \quad (12)$$

where x denotes a real HR sample from the data distribution p_{data} , and z represents a random vector drawn from a simple prior distribution p_z (e.g., Gaussian noise). Through this adversarial process, G is encouraged to produce outputs that are not only visually convincing but also statistically consistent with the real data distribution, as judged by D .

The design space of GANs is extremely broad, with countless variants proposed to improve stability, perceptual quality, or diversity of generated samples. In the super-resolution domain, for example, adversarial losses have been combined with perceptual criteria and enhanced discriminator designs to produce sharper reconstructions. At the same time, this diversity highlights how problem dependent GANs are: adversarial training is notoriously unstable, and small architectural or training choices can determine whether a model succeeds or collapses. As a consequence, despite their flexibility and elegance, GANs have proven difficult to adapt reliably to specialized applications such as climate downscaling.

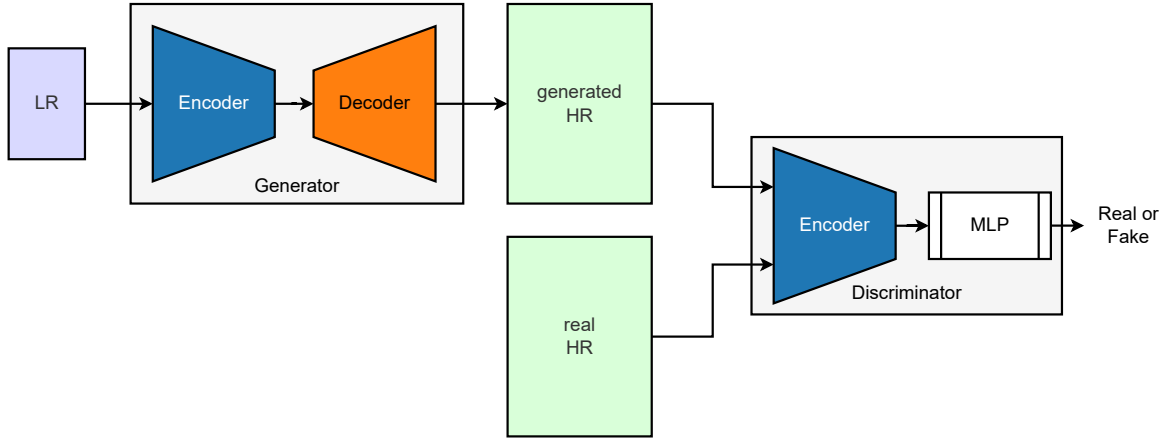


Figure 18: Schematic of a deterministic GAN for super-resolution. The generator receives a low-resolution (LR) input and, through an encoder–decoder architecture, produces a synthetic high-resolution (HR) field. Both the generated HR and the corresponding real HR are passed to the discriminator, composed of a convolutional encoder followed by a multilayer perceptron (MLP), which outputs a probability indicating whether the sample is real or generated.

3.6.1 Conditioning in GANs

In their original formulation, GANs are designed as non-deterministic generative models. The generator G maps a random vector $z \sim p_z$ to a synthetic output $G(z)$, allowing multiple plausible realizations for the same underlying condition. This stochasticity is a powerful feature: it enables the generator to represent diverse samples consistent with the target data distribution.

For super-resolution and climate downscaling, however, the setting is different. The low-resolution (LR) input already imposes strong constraints on the possible high-resolution (HR) outcomes. If the generator were driven only by random noise, it would attempt to learn the entire data distribution of atmospheric states—a task irrelevant to the conditional problem at hand. The solution is to condition the generator on the LR input, leading to the formulation of conditional GANs (cGANs) [MO14]. In this framework, the generator produces $G(y, z)$, where y denotes the LR field and z optional noise.

This conditioning introduces two main design choices:

1. Deterministic GANs, where G is conditioned solely on the LR field y (with no noise). This simplifies training and leads to stable, sharp reconstructions, but sacrifices the ability to model multiple plausible HR outcomes. The widely used SRGAN model [LTH⁺17] exemplifies this approach. An example of a deterministic conditioned GAN is illustrated in Figure 18.
2. Stochastic GANs, where G is conditioned on both the LR field and random noise, $G(y, z)$. This allows the model to capture different HR realizations consistent with the same LR input, though in practice training tends to be more difficult and less stable.

Thus, while GANs in principle provide a framework for non-deterministic super-resolution, many practical implementations in climate downscaling and computer vision favor the deterministic variant, accepting reduced variability in exchange for robustness.

3.6.2 Limitations of GANs in Practice

Despite their elegant formulation and impressive results in some domains, GANs remain difficult to use in practice. Training lacks convergence guarantees, is prone to instabilities such as mode collapse, and requires careful hyperparameter tuning. Designing an architecture that transfers smoothly across different domains or variables is rarely straightforward, and substantial effort is often needed to fine-tune models for each new downscaling application.

While GAN-based generators can produce visually sharp reconstructions, these are not always physically consistent and are prone to artefacts, which is a critical concern in operational downscaling contexts. In practice, conditioning is usually performed in a purely deterministic fashion, meaning

the model loses the stochastic advantages of probabilistic approaches and functions essentially as an adversarially trained regressor. On the other hand, GANs are typically more computationally efficient than diffusion models, offering faster training and inference once a stable configuration is achieved.

In the field of climate downscaling, GANs have been explored but have rarely become the primary tool of choice, largely due to these limitations. Their instability, difficulty of adaptation, and lack of physical reliability motivate the move toward diffusion models and other probabilistic generative methods, which provide a more robust way of representing uncertainty while maintaining consistency with large-scale dynamics. Nevertheless, GANs remain a cornerstone architecture and will be used extensively in this thesis as a point of comparison with other methods.

3.7 Denoising Diffusion Models

Diffusion models are a class of generative models that have recently achieved state-of-the-art performance in non-deterministic downscaling. Their central idea is to frame generation as the reversal of a gradual noising process. In the *forward process*, a clean data sample x_0 is progressively corrupted by Gaussian noise through a sequence of steps, producing increasingly noisy versions $x_1, x_2, \dots, x_T$ until the signal is indistinguishable from pure noise. This process is designed to be simple and tractable: each step adds only a small amount of noise according to a predefined schedule.

The learning task is to approximate the *reverse process*: starting from pure noise x_T , the model learns how to iteratively remove noise and recover a clean sample. This is achieved by training a denoiser network $\epsilon_\theta(x_t, t)$, most often a U-Net, to predict the noise component at each timestep. By chaining together many such denoising steps, the model transforms unstructured noise into structured data that matches the distribution of the training set.

Intuitively, each reverse step can be broken down into three operations:

1. Denoise the current sample using the trained network.
2. Reintroduce a controlled amount of noise (depending on the chosen formulation).
3. Update the signal-to-noise ratio according to the schedule.

Iterating this loop produces a clean sample after T steps. What distinguishes diffusion from earlier generative approaches is that the model does not attempt to generate an entire sample in a single pass. Instead, it constructs the output gradually, leveraging the stability of denoising as a learning task.

This perspective highlights two key strengths of diffusion models. First, their training is stable, since predicting noise from noisy inputs is a well-conditioned regression problem. Second, their iterative nature naturally supports probabilistic modeling: by injecting controlled noise at each step, diffusion models can generate diverse outputs consistent with the training distribution. These properties make them particularly appealing for scientific applications such as climate downscaling, where both stability and uncertainty quantification are essential.

3.7.1 Mathematical Foundations

From a mathematical perspective, considering a distribution $q(x_0)$ which generates the data, generative models aim to find a parameter vector θ such that the distribution $p_\theta(x_0)$ parameterized by a neural network approximates $q(x_0)$.

To make the problem tractable, diffusion models introduce a sequence of latent variables that represent progressively noisier versions of the data. Instead of modeling the data distribution $p_\theta(x_0)$ directly, it is expressed as the marginal of a joint distribution over the entire noise trajectory $\{x_1, x_2, \dots, x_T\}$:

$$p_\theta(x_0) = \int p_\theta(x_{0:T}) dx_{1:T}. \quad (13)$$

This formulation means that the probability of observing x_0 can be obtained by integrating over all possible intermediate states.

The key assumption of denoising diffusion probabilistic models (DDPMs) is that these intermediate states follow a Markov chain evolving backwards in time: starting from pure Gaussian noise x_T , the

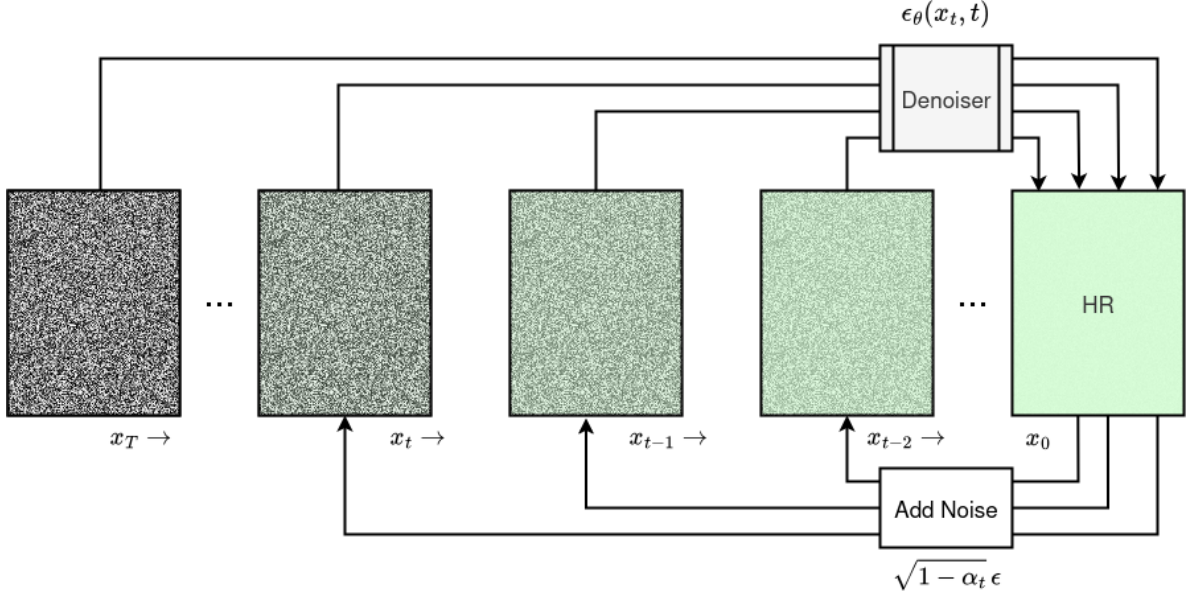


Figure 19: Illustration of the reverse diffusion process. Starting from pure Gaussian noise x_T , the model iteratively applies the neural network denoiser $\epsilon_\theta(x_t, t)$ to predict the injected noise and progressively reconstruct cleaner samples. At each step, a portion of noise proportional to $(1 - \alpha_t)$ is reintroduced, until a high-resolution approximation of the original image x_0 is obtained.

model learns to iteratively denoise until it reconstructs a clean sample x_0 . The joint distribution therefore factorizes as

$$p_\theta(x_{0:T}) = p_\theta(x_T) \prod_{t=1}^T p_\theta(x_{t-1}|x_t), \quad (14)$$

where $p_\theta(x_T)$ is a simple prior distribution (commonly $\mathcal{N}(\mathbf{0}, \mathbf{I})$), and each transition $p_\theta(x_{t-1}|x_t)$ is a learned denoising step parameterized by the neural network.

The formulation introduced above provides a principled generative model for the data: starting from Gaussian noise x_T , the model defines a reverse Markov chain that eventually produces a clean sample x_0 . However, this definition alone does not tell us how to learn the parameters θ . Training ideally requires maximizing the log-likelihood of the observed data,

$$\max_{\theta} \log p_\theta(x_0), \quad (15)$$

but computing $p_\theta(x_0)$ involves integrating over all latent states $x_{1:T}$ in Equation (13). This integral is intractable in practice, since the number of latent variables grows linearly with the number of timesteps T .

To address this challenge, diffusion models adopt a variational approach. The idea is to compare the intractable reverse process $p_\theta(x_{1:T}|x_0)$ with the known forward process $q(x_{1:T}|x_0)$, which defines how noise is added to the data. By exploiting the non-negativity of the Kullback–Leibler divergence, we obtain the following inequality:

$$D_{\text{KL}}(q(x_{1:T}|x_0) \parallel p_\theta(x_{1:T}|x_0)) \geq 0, \quad (16)$$

which implies

$$-\log p_\theta(x_0) \leq -\log q(x_0) + D_{\text{KL}}. \quad (17)$$

This expression provides an upper bound on the negative log-likelihood, known as the Evidence Lower Bound (ELBO). Expanding the terms, we obtain a tractable training objective:

$$L_\theta = \mathbb{E}_q \left[\log q(x_{1:T}|x_0) - \log p_\theta(x_{0:T}) \right], \quad (18)$$

where the expectation is taken with respect to the forward process $q(x_{1:T}|x_0)$, which is analytically defined and straightforward to sample from.

In practice, minimizing this objective enforces that the learned reverse process $p_\theta(x_{0:T})$ matches the forward noising process $q(x_{1:T}|x_0)$. Training diffusion models therefore amounts to optimizing the ELBO loss L_θ , which acts as a tractable surrogate for maximum likelihood estimation of the data distribution.

The variational loss introduced in Equation 18 can be further decomposed into a sum of per-timestep contributions [SWMG15]:

$$L_\theta = L_T + L_{T-1} + \dots + L_0, \quad (19)$$

where

$$\begin{aligned} L_T &= D_{\text{KL}}(q(x_T|x_0) \parallel p_\theta(x_T)), \\ L_t &= D_{\text{KL}}(q(x_t|x_{t+1}, x_0) \parallel p_\theta(x_t|x_{t+1})) \quad \text{for } 1 \leq t \leq T-1, \\ L_0 &= -\log p_\theta(x_0|x_1). \end{aligned}$$

Since all involved distributions are Gaussian, the KL divergences can be evaluated in closed form, leading to the simplified expression

$$L_t = \mathbb{E}_{t \sim [1, T], x_0, \epsilon_t} \left[\gamma_t \|\epsilon_t - \epsilon_\theta(x_t, t)\|^2 \right]. \quad (20)$$

In other words, the training loss reduces to the mean squared error between the predicted noise $\epsilon_\theta(x_t, t)$ and the actual noise ϵ_t used in the forward process, weighted by a factor γ_t .

In practice, empirical results show that ignoring the weighting coefficients γ_t often yields better performance, so most implementations adopt the unweighted mean squared error as the training objective.

While DDPMs assume that the forward diffusion is Markovian, Denoising Diffusion Implicit Models (DDIMs) [SME20b] relax this constraint by introducing a non-Markovian formulation. The key idea is to define a forward process that is no longer strictly Markovian, yet preserves the same marginal distributions $q(x_t|x_0)$ as in DDPM. This allows greater flexibility in the definition of the trajectory and enables deterministic sampling with fewer steps.

Formally, the DDIM forward process is written as

$$q_\sigma(x_{1:T}|x_0) = q_\sigma(x_T|x_0) \prod_{t=2}^T q_\sigma(x_{t-1}|x_t, x_0). \quad (21)$$

Here, unlike the DDPM case, the transition distribution $q_\sigma(x_{t-1}|x_t, x_0)$ is allowed to depend on both x_t and the original clean sample x_0 .

The terminal distribution is defined as a Gaussian

$$q_\sigma(x_T|x_0) = \mathcal{N}(x_T \mid \sqrt{\alpha_T}x_0, (1 - \alpha_T)\mathbf{I}), \quad (22)$$

and each intermediate transition also remains Gaussian,

$$q_\sigma(x_{t-1}|x_t, x_0) = \mathcal{N}\left(x_{t-1} \mid \mu_{\sigma_t}(x_0, \alpha_{t-1}), \sigma_t^2 \mathbf{I}\right), \quad (23)$$

with mean

$$\mu_{\sigma_t}(x_0, \alpha_{t-1}) = \sqrt{\alpha_{t-1}} x_0 + \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \frac{x_t - \sqrt{\alpha_t} x_0}{\sqrt{1 - \alpha_t}}. \quad (24)$$

By construction, the marginal distribution at any step t remains identical to the DDPM formulation:

$$q_\sigma(x_t|x_0) = \mathcal{N}(x_t \mid \sqrt{\alpha_t}x_0, (1 - \alpha_t)\mathbf{I}). \quad (25)$$

This ensures that a noisy sample x_t can always be expressed as a linear combination of the clean data and Gaussian noise:

$$x_t = \sqrt{\alpha_t} x_0 + \sqrt{1 - \alpha_t} \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \quad (26)$$

This property guarantees compatibility between DDPM and DDIM: the two models share the same forward marginals, while DDIM additionally allows for a non-Markovian trajectory that enables faster or deterministic sampling.

In order to define the reverse process, we must specify how the transition distributions $p_\theta(x_{t-1}|x_t)$ are parameterized. A common choice in diffusion models is to train a neural network to predict the noise component that was added during the forward process.

Considering that in forward dynamics a noisy sample at time t is expressed as reported in Equation (26), if the network $\epsilon_\theta(x_t, \alpha_t)$ can correctly estimate the noise ϵ_t , one can recover an estimate of the original clean sample x_0 as

$$f_\theta^{(t)}(x_t, \alpha_t) = \frac{x_t - \sqrt{1 - \alpha_t} \epsilon_\theta(x_t, \alpha_t)}{\sqrt{\alpha_t}}. \quad (27)$$

This estimated $\hat{x}_0 = f_\theta^{(t)}(x_t, \alpha_t)$ can then be injected into the DDIM reverse formulation to obtain the previous state:

$$x_{t-1} = \sqrt{\alpha_{t-1}} f_\theta^{(t)}(x_t, \alpha_t) + \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \epsilon_\theta(x_t, \alpha_t). \quad (28)$$

This parameterization highlights the practical role of the neural network: given a noisy observation x_t , it predicts the perturbation that must be removed in order to recover a clean signal. In practice, this approach has become the standard for diffusion models due to its simplicity and empirical effectiveness. It is, however, only one possible formulation: alternative parameterizations based on predicting the clean sample x_0 directly or predicting the so-called *velocity* have also been proposed. These alternatives will be discussed in subsequent sections.

Algorithms for training and sampling. The diffusion framework admits a very simple training procedure. At each iteration, a clean data point x_0 is perturbed with Gaussian noise according to a randomly chosen timestep t , yielding a noisy sample x_t . The neural network $\epsilon_\theta(x_t, \alpha_t)$ is then trained to predict the exact noise that was injected, using a mean squared error objective. This is summarized in Algorithm 1.

Algorithm 1 Training procedure for diffusion models.

```

1: repeat
2:    $x_0 \sim q(x_0)$  ▷ sample a clean datapoint
3:    $t \sim \text{Uniform}\{1, \dots, T\}$  ▷ sample a timestep
4:    $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  ▷ sample Gaussian noise
5:    $x_t = \sqrt{\alpha_t} x_0 + \sqrt{1 - \alpha_t} \epsilon$  ▷ corrupt  $x_0$ 
6:   Update  $\theta$  by minimizing  $\|\epsilon - \epsilon_\theta(x_t, \alpha_t)\|^2$ 
7: until convergence

```

Sampling from the model proceeds in the reverse direction. Starting from pure Gaussian noise $x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the network is applied iteratively to predict the noise and recover progressively denoised samples x_{t-1} until a clean signal x_0 is obtained. The process is summarized in Algorithm 2.

Algorithm 2 Sampling procedure for diffusion models.

```

1:  $x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
2: for  $t = T, \dots, 1$  do
3:    $\epsilon = \epsilon_\theta(x_t, \alpha_t)$  ▷ predict noise
4:    $\tilde{x}_0 = \frac{1}{\sqrt{\alpha_t}} (x_t - \sqrt{1 - \alpha_t} \epsilon)$  ▷ estimate clean sample
5:    $x_{t-1} = \sqrt{\alpha_{t-1}} \tilde{x}_0 + \sqrt{1 - \alpha_{t-1}} \epsilon$  ▷ construct next state
6: end for

```

3.7.2 Practical Variants: DDPM vs DDIM

While the mathematical framework of diffusion models is general, in practice two main formulations are widely adopted: Denoising Diffusion Probabilistic Models (DDPM) [HJA20b] and Denoising Diffusion Implicit Models (DDIM) [SME20b]. Both approaches rely on the same forward diffusion process, but they differ in how the reverse denoising steps are defined.

In the case of DDPM, the reverse process is modeled as a Markov chain

$$p_{\theta}(x_{0:T}) = p(x_T) \prod_{t=1}^T p_{\theta}(x_{t-1}|x_t),$$

where at each step a new random noise sample is injected. This makes the process stochastic: even with the same trained model and initial noise x_T , different noise draws at intermediate steps can lead to different samples.

DDIM instead introduces a non-Markovian formulation

$$q_{\sigma}(x_{1:T}|x_0) = q_{\sigma}(x_T|x_0) \prod_{t=2}^T q_{\sigma}(x_{t-1}|x_t, x_0),$$

which allows the reverse process to be deterministic when the variance parameter σ_t is set to zero. In this case, the trajectory from x_T back to x_0 follows a fixed path defined by the model's predictions, without injecting additional noise at each step. As a result, DDIM sampling can reconstruct realistic samples in far fewer steps compared to DDPM, while achieving similar quality.

This efficiency makes DDIM particularly well-suited for computationally intensive applications such as high-resolution climate downscaling. For this reason, throughout this thesis we rely on DDIM with deterministic sampling ($\sigma_t = 0$) as the default choice.

From an implementation perspective, the main difference between DDPM and DDIM lies in how the new noise is handled at each reverse step. In DDPM, the procedure at step t is:

1. Use the network $\epsilon_{\theta}(x_t, t)$ to predict the noise in the current sample x_t .
2. Form an estimate of the clean image $\hat{x}_0$ or the denoised sample at step $t - 1$ using the chosen parameterization.
3. Sample a new independent Gaussian noise $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
4. Combine $\hat{x}_0$ and z with the updated signal and noise rates (α_{t-1}, σ_t) to construct x_{t-1} .

Each step therefore introduces fresh randomness, and the final sample x_0 depends on both the initial noise x_T and all intermediate noise draws. In DDIM, the process is similar except for the treatment of the noise term:

1. Use the network $\epsilon_{\theta}(x_t, t)$ to predict the noise in the current sample x_t .
2. Form an estimate of the clean image $\hat{x}_0$ or the denoised sample at step $t - 1$ using the chosen parameterization.
3. Instead of sampling new noise, re-use the predicted noise $\epsilon_{\theta}(x_t, t)$, scaled according to the updated rates.
4. Construct x_{t-1} deterministically from $\hat{x}_0$ and the rescaled predicted noise.

As a result, the trajectory from x_T back to x_0 becomes deterministic when $\sigma_t = 0$, and the number of required steps can be significantly reduced without a major loss in quality. This practical difference is often summarized as: DDPM re-samples noise at every step, whereas DDIM reuses the model's predicted noise with updated coefficients.

As a compact summary, the deterministic DDIM reverse update can be written in closed form as

$$x_{t-1} = \sqrt{\frac{\alpha_{t-1}}{\alpha_t}} x_t + \left(\sqrt{1 - \alpha_{t-1}} - \sqrt{\frac{\alpha_{t-1}(1 - \alpha_t)}{\alpha_t}} \right) \epsilon_{\theta}(x_t, \alpha_t). \quad (29)$$

This expression highlights that each reverse step is a linear combination of the current sample x_t and the predicted noise $\epsilon_{\theta}(x_t, t)$, with coefficients determined entirely by the noise schedule $\{\alpha_t\}$. The stochastic DDPM update is recovered by reintroducing an additional Gaussian noise term at each step, whereas in the deterministic DDIM case ($\sigma_t = 0$) no new randomness is injected.

3.7.3 Noise and Signal Rates

As introduced in Section 3.7.1, in the standard formulation of diffusion models a corrupted sample at time t is expressed as a mixture of the clean data x_0 and Gaussian noise ϵ :

$$x_t = \sqrt{\alpha_t} x_0 + \sqrt{1 - \alpha_t} \epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

Here, α_t is the *signal power* at step t . The corresponding multipliers

$$s_t = \sqrt{\alpha_t}, \quad n_t = \sqrt{1 - \alpha_t}$$

are called the *signal rate* and *noise rate*, since they directly control the strength of the clean signal and the injected noise. Their squares satisfy $s_t^2 + n_t^2 = 1$, which reflects the *variance-preserving* (VP) assumption: at every step, the combined variance of signal and noise remains constant. Together, these quantities define the *signal-to-noise ratio* (SNR):

$$\text{SNR}_t = \frac{\alpha_t}{1 - \alpha_t}.$$

We use square roots because powers $(\alpha_t, 1 - \alpha_t)$ represent *variances*, while the multipliers in the forward process must be *standard deviations*.

A practical diffusion process, however, cannot span the entire range from perfectly clean data to pure noise. Instead, it is constrained between finite *upper and lower bounds* on the SNR, and within this interval a specific *schedule* determines how s_t and n_t evolve over time. These two design choices—the bounds and the schedule—jointly shape the difficulty of the denoising task and the overall effectiveness of the model.

Upper and lower bounds. In theory, one would like the diffusion process to begin from a perfectly clean state ($\text{SNR} \rightarrow \infty$) and end in pure noise ($\text{SNR} \rightarrow 0$). In practice, however, such extremes are problematic: if the signal rate at $t = 0$ were exactly one, the process would be deterministic and gradients would vanish; if the noise rate at $t = 1$ were exactly one, all structure would be destroyed and the reverse process would lack any foothold for denoising. For this reason, the range of α_t (equivalently the log-SNR) is restricted to *finite bounds*. By choosing upper and lower limits—typically $\log \text{SNR}_{\text{start}} \gg 0$ and $\log \text{SNR}_{\text{end}} \ll 0$ —one ensures numerical stability, well-behaved gradients, and a smoother training signal across the full trajectory.

A central design choice is therefore how to parameterize these bounds. The most common formulation is to specify them directly in log-SNR space, which provides intuitive control in terms of orders of magnitude of the signal-to-noise ratio. In cosine-based schedules, by contrast, the same role is played by selecting minimum and maximum angles, which determine the effective range of signal and noise rates. Other formulations exist as well, such as interpolating in log-noise or log-signal power. In all cases, however, explicit bounds are required to keep the process well-behaved at both ends.

In practice, the values of these bounds are treated as hyperparameters and tuned empirically. Some recent works [KAAL22] emphasize that the optimal range can be task-dependent: for example, image generation may tolerate stronger corruption, while super-resolution or conditional tasks often benefit from preserving more signal at the final step. This highlights that, while much attention is given to the functional form of the schedule, the choice of bounds can be equally critical for stable training and high-quality sampling.

Variance-preserving schedules. In the variance-preserving (VP) setting, the forward process is written as

$$x_t = s_t x_0 + n_t \epsilon, \quad s_t^2 + n_t^2 = 1,$$

so that the total variance of x_t is constant and only the balance between signal and noise evolves with time. The design freedom lies in how s_t and n_t interpolate between the near-clean start ($s_0 \approx 1, n_0 \approx 0$) and the near-noisy end ($s_1 \approx 0, n_1 \approx 1$).

Early work [HJA20b] adopted a *linear schedule* on noise power,

$$n(t)^2 = (1 - t) n(0)^2 + t n(1)^2,$$

a simple baseline that often erases signal too abruptly near $t = 1$. Nichol & Dhariwal [ND21] proposed a *cosine schedule*, where diffusion time is reparameterized as an angle

$$\theta(t) = (1 - t) \arcsin(n(0)) + t \arcsin(n(1)), \quad n(t)^2 = \sin^2 \theta(t),$$

which preserves more signal at both ends and produces smoother degradation.

A different family of schedules interpolates in logarithmic domains. In the *log-SNR linear* case [KSPH21], one sets

$$\log \text{SNR}(t) = (1 - t) \log \text{SNR}(0) + t \log \text{SNR}(1), \quad n(t)^2 = \frac{1}{1 + \text{SNR}(t)},$$

so that the SNR decays exponentially instead of collapsing suddenly at the end. Analogously, *log-noise* and *log-signal* schedules interpolate geometrically in $n(t)^2$ or $s(t)^2$,

$$n(t)^2 = n(0)^2 \left(\frac{n(1)^2}{n(0)^2} \right)^t, \quad s(t)^2 = s(0)^2 \left(\frac{s(1)^2}{s(0)^2} \right)^t,$$

ensuring constant ratio growth or decay.

Further heuristics introduce nonlinear pacing, e.g. squared-time decay

$$s(t)^2 = s(0)^2 \left(\frac{s(1)^2}{s(0)^2} \right)^{t^2},$$

which preserves more signal early on and collapses faster later. Finally, Lin et al. [LCW+24] proposed a *logistic schedule*, where cumulative signal power follows

$$\alpha(t) = \frac{1}{1 + \exp[\kappa(t - t_0)]}, \quad s(t) = \sqrt{\alpha(t)}, \quad n(t) = \sqrt{1 - \alpha(t)},$$

with parameters t_0 (midpoint) and κ (steepness) controlling the shape of the S-curve.

Across all these formulations, the endpoints remain the same, but the intermediate trajectory dictates how task difficulty is distributed: slow decay leaves denoising trivial, while fast decay makes it intractable. A comparison of all VP methods is reported in Table 8 and in Figure 20 the respective signal and noise rates are plotted.

Non variance-preserving schedules. Non-variance-preserving (non-VP) approaches lift the constraint $s_t^2 + n_t^2 = 1$, so the signal and noise coefficients are no longer tied to a fixed total variance. This decoupling enlarges the design space: the forward dynamics need not be governed solely by the signal-to-noise ratio, but can instead be shaped by other criteria. Broadly, four main directions have emerged.

The first is *variance-exploding* (VE) processes [SSDK+20, KAAL22], where the signal is left unscaled ($s_t = 1$) while the noise coefficient grows exponentially,

$$x_t = x_0 + \sigma(t) \epsilon, \quad \sigma(t) = \sigma_{\min} \left(\frac{\sigma_{\max}}{\sigma_{\min}} \right)^t.$$

Here the total variance increases as $1 + \sigma(t)^2$, leading to weak perturbations early and aggressive corruption near $t = 1$. VE schedules stabilize score matching but require very large $\sigma_{\max}$ to reach pure noise.

A second line of work defines schedules by equalizing step difficulty. Okada et al. [OYK+24] introduced *constant-rate* (CRS) schedules, which enforce a fixed distributional change per step,

$$D(q(x_t), q(x_{t+\Delta})) = \text{const},$$

with D typically the symmetric KL divergence. This yields dataset-dependent s_t, n_t trajectories that avoid trivial early steps and overly hard late ones.

A third perspective, proposed by Kahouli et al. [KRG+25], is to disentangle total variance and SNR. By introducing two independent functions, $\text{TV}(t)$ and $\text{SNR}(t)$, the forward process becomes

$$s_t^2 = \frac{\text{TV}(t) \text{SNR}(t)}{1 + \text{SNR}(t)}, \quad n_t^2 = \frac{\text{TV}(t)}{1 + \text{SNR}(t)}.$$

This framework recovers VP ($\text{TV}(t) = 1$) and VE (growing $\text{TV}(t)$) as special cases, but also allows hybrid designs such as constant-TV with tuned SNR, which were shown to outperform exploding-TV schedules.

Finally, beyond analytic formulas, several works explored *learned or adaptive* schedules [KSPH21, SGDSK24]. Here the forward process is unchanged, but the trajectories s_t, n_t (or their endpoints) are optimized during training or conditioned on the input. This adds flexibility: schedules can adapt per-instance or spatially, and the allocation of noise across timesteps can be tuned to the dataset or task. A comparison of all non-VP methods is reported in Table 8.

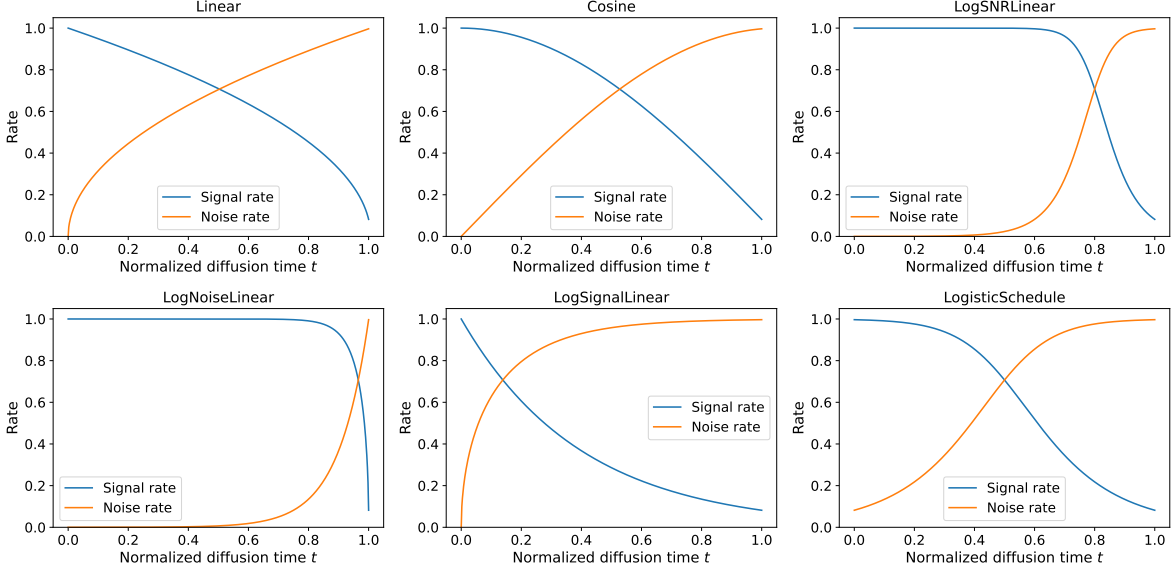


Figure 20: Comparison of common variance-preserving noise and signal rate schedules used in diffusion models. Each panel shows the evolution of the signal rate $s_t = \sqrt{\alpha_t}$ and noise rate $n_t = \sqrt{1 - \alpha_t}$ over normalized diffusion time $t \in [0, 1]$. Although all schedules interpolate between the same endpoints (near-clean at $t = 0$ and near-noise at $t = 1$), their intermediate trajectories differ substantially, affecting the difficulty of the denoising task at each step and the overall efficiency of training and sampling. For all plots the bounds are set to $\log \text{SNR}_{\text{start}} = 20.0$ and $\log \text{SNR}_{\text{end}} = -5.0$; the logistic schedule uses $\kappa = 10$ and $t_0 = 0.5$.

3.7.4 Design Space of Diffusion Models

Diffusion models involve a number of important design decisions that significantly influence their stability, efficiency, and output quality. In the following we outline the main elements of this design space.

Choice of denoiser. The denoising network $f_\theta(x_t, t)$ is the core of the model. In most implementations it takes the form of a U-Net [HJA20b], which provides a natural multi-scale decomposition of the input. Residual networks are also widely used, as skip connections facilitate optimization and allow deeper architectures. A number of architectural changes can be implemented: depthwise separable convolutions reduce computational cost while preserving accuracy, squeeze-and-excitation blocks allow dynamic channel-wise recalibration, and attention modules enable long-range interactions that are otherwise difficult to capture with convolution alone. Also transformer-based denoisers have been explored, offering potentially better scaling properties at the cost of higher training complexity. In general, the choice for denoisers for diffusion is strongly connected with denoisers research in general. A comprehensive description of denoising architectures is given in the previous sections.

Prediction targets. The denoiser can be parameterized in different but mathematically equivalent ways, depending on the choice of training target. The original formulation is *noise prediction*, where

Name	Source	Definition	Properties
Linear	Ho et al. [HJA20b] (2020)	Noise power interpolates linearly: $n(t)^2 = (1 - t)n(0)^2 + tn(1)^2$, with $n(t) = \sqrt{1 - \alpha_t}$.	Simple baseline; tends to wipe out signal too quickly near the end. Standard DDPM choice.
Cosine	Nichol & Dhariwal [ND21] (2021)	The diffusion time $t \in [0, 1]$ is first mapped to an angle between the start and end values $\theta(t) = (1-t) \arcsin(\sqrt{n(0)^2}) + t \arcsin(\sqrt{n(1)^2}),$ where $n(0)^2$ and $n(1)^2$ are the initial and final noise powers. The noise power is then defined as $n(t)^2 = \sin^2(\theta(t))$, with noise rate $n(t) = \sqrt{n(t)^2}$ and signal rate $s(t) = \sqrt{1 - n(t)^2}$.	Preserves more signal at both the start and the end of the process; produces a smoother trajectory than linear. Widely used in practice.
Log-SNR Linear	Kingma et al. [KSPH21] (2021)	Log-SNR interpolates linearly: $\log(\text{SNR}(t)) = (1 - t) \log(\text{SNR}(0)) + t \log(\text{SNR}(1))$. Equivalent to geometric interpolation of SNR, with noise power $n_t^2 = 1/(1 + \text{SNR}(t))$.	Ensures smooth exponential decay of SNR; avoids abrupt collapse near the end. Common in continuous VP-SDEs.
Log-Noise Linear	Variants	Noise power grows exponentially in t : $n(t)^2 = n(0)^2 \left(\frac{n(1)^2}{n(0)^2} \right)^t$. This can be seen as the exponential analogue of the Linear schedule, with constant growth ratio instead of constant increment.	Smooth geometric increase of noise power; avoids abrupt growth near the end. Equivalent to linear interpolation in log-noise power.
Log-Signal Linear	Variants	Signal power decays exponentially in t : $s(t)^2 = s(0)^2 \left(\frac{s(1)^2}{s(0)^2} \right)^t$, with $n(t)^2 = 1 - s(t)^2$. This can be seen as the exponential analogue of the Linear schedule applied to the signal rather than the noise.	Smooth geometric decay of signal power; emphasizes early signal retention and late rapid corruption. Equivalent to linear interpolation in log-signal power.
Signal-Step Linear	Heuristic	Signal power decays geometrically with squared time: $S(t) = S(0) \left(\frac{S(1)}{S(0)} \right)^{t^2}$, with $n(t)^2 = 1 - S(t)^2$.	Preserves more signal at early steps and collapses faster at later steps; heuristic that approximates a linear-looking decay.
Logistic	Lin et al. [LCW ⁺ 24] (2024)	Cumulative signal power follows a logistic curve: $\alpha(t) = \frac{1}{1 + \exp[\kappa(t - t_0)]}$, with $s(t) = \sqrt{\alpha(t)}$, $n(t) = \sqrt{1 - \alpha(t)}$. Here t_0 sets the midpoint (where $\alpha = 0.5$) and κ controls the steepness.	Provides a smooth S-shaped transition; avoids vanishing noise at $t = 0$; shown to improve DDIM inversion and editing stability. Proposed mainly for editing tasks.

Table 7: Variance-preserving (VP) noise and signal rate schedules in diffusion models. All schedules fit the forward process $x_t = \sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\epsilon$, with $\alpha_t + (1 - \alpha_t) = 1$.

Name	Source	Definition	Properties
Exponential / Geometric (VE)	Song et al. [SSDK ⁺ 20], Karras et al. [KAAL22]	Forward process $x_t = s_t x_0 + n_t \epsilon$, with fixed signal rate $s_t = 1$ and noise coefficient $n_t = \sigma(t) = \sigma_{\min} (\sigma_{\max}/\sigma_{\min})^t$.	Variance-exploding: total variance $1 + \sigma(t)^2$ increases over time. Aggressive late corruption; requires large $\sigma_{\max}$ to reach pure noise. Used in VE-SDEs and EDM.
Constant-Rate (CRS)	Okada et al. [OYK ⁺ 24] (2024)	Forward process $x_t = s_t x_0 + n_t \epsilon$, where s_t, n_t are defined implicitly by the condition $D(q(x_t), q(x_{t+\Delta})) = \text{const}$ for all t , typically using the symmetric KL divergence.	Ensures that each diffusion step changes the distribution by the same amount, avoiding trivial or overly hard steps. No closed form: schedule is solved numerically and becomes dataset-dependent. Reported to improve both training efficiency and sample fidelity.
TV/SNR Disentangled	Kahouli et al. [KRG ⁺ 25] (2025)	Forward process $x_t = s_t x_0 + n_t \epsilon$, with $s_t^2 = \frac{\text{TV}(t) \text{SNR}(t)}{1 + \text{SNR}(t)}$ and $n_t^2 = \frac{\text{TV}(t)}{1 + \text{SNR}(t)}$. Here $\text{TV}(t)$ (total variance) and $\text{SNR}(t)$ are chosen independently.	Framework that generalizes VP and VE: e.g. constant-TV with tuned-SNR outperforms exploding-TV. Provides new design freedom rather than a single closed-form schedule.
Learned / Adaptive	Kingma et al. [KSPH21] (2021), Sahoo et al. [SGDSK24] (2024)	Forward process $x_t = s_t x_0 + n_t \epsilon$, where s_t, n_t (or their endpoints) are optimized during training, or conditioned on the input data.	Provides flexibility beyond fixed analytic schedules: bounds can be learned, or noise allocation can vary per-instance or spatially. Reported to improve likelihoods and stability, especially in conditional or structured tasks.

Table 8: Non-variance-preserving noise and signal rate schedules. These approaches depart from the standard VP assumption $s_t^2 + n_t^2 = 1$, either by exploding variance (VE), learning dataset-specific schedules (CRS), disentangling total variance from SNR, or making schedules adaptive.

the network is trained to estimate the Gaussian noise ϵ added in the forward process, using the objective

$$L_\epsilon = \mathbb{E}_{x_0, \epsilon, t} [\|\epsilon - \hat{\epsilon}_\theta(x_t, t)\|^2]. \quad (30)$$

A natural alternative is *image prediction*, where the target is the original clean sample x_0 ,

$$L_{x_0} = \mathbb{E}_{x_0, \epsilon, t} [\|x_0 - \hat{x}_{0, \theta}(x_t, t)\|^2]. \quad (31)$$

More recently, *velocity prediction* has been proposed [SH22], in which the target is a linear combination of signal and noise defined as $v_t = \sqrt{\alpha_t} \epsilon - \sqrt{1 - \alpha_t} x_0$, with training loss

$$L_v = \mathbb{E}_{x_0, \epsilon, t} [\|v_t - \hat{v}_\theta(x_t, t)\|^2]. \quad (32)$$

Although these parameterizations differ, they are linearly related and therefore interchangeable: once the model predicts one target, the others can be recovered through deterministic transformations. In particular,

$$\hat{x}_{0, \theta} = \frac{1}{\sqrt{\alpha_t}} (x_t - \sqrt{1 - \alpha_t} \hat{\epsilon}_\theta), \quad (33)$$

$$\hat{\epsilon}_\theta = \frac{1}{\sqrt{1 - \alpha_t}} (x_t - \sqrt{\alpha_t} \hat{x}_{0, \theta}), \quad (34)$$

$$\hat{\epsilon}_\theta = \frac{1}{\sqrt{\alpha_t}} (\hat{v}_\theta + \sqrt{1 - \alpha_t} x_t), \quad \hat{x}_{0, \theta} = \frac{1}{\sqrt{\alpha_t}} (x_t - \sqrt{1 - \alpha_t} \hat{v}_\theta). \quad (35)$$

In principle, all three objectives are equivalent; the choice of target does not alter the expressive power of the model. In practice, however, their numerical conditioning differs. Noise prediction is simple and stable, image prediction can lead to sharper results but becomes unstable at large noise levels, and velocity prediction often provides the best balance by distributing information more evenly across timesteps.

Exponential Moving Average (EMA). A crucial practical technique is the use of an exponential moving average of the network weights. During training the denoiser parameters may fluctuate significantly due to stochastic optimization. The EMA maintains a smoothed copy of the parameters $\theta_k^{(\text{EMA})}$ at iteration k according to

$$\theta_k^{(\text{EMA})} = \beta \theta_{k-1}^{(\text{EMA})} + (1 - \beta) \theta_k, \quad (36)$$

where θ_k are the current parameters, and $\beta \in [0, 1)$ is the decay factor, typically chosen between 0.999 and 0.9999. This recursion ensures that the EMA weights approximate a weighted average over past parameter values with exponentially decaying importance. Concretely,

$$\theta_k^{(\text{EMA})} \approx (1 - \beta) \sum_{i=0}^k \beta^i \theta_{k-i}, \quad (37)$$

so that more recent updates contribute more strongly than older ones.

From an intuitive perspective, the EMA can be seen as a *low-pass filter* applied to the trajectory of the parameters during training: rapid, high-frequency fluctuations are suppressed, while slower, consistent trends are preserved. This smoothing greatly improves the stability of the sampling procedure, as the denoiser used at inference is less sensitive to noisy gradient updates or catastrophic jumps in parameter space.

Importantly, if training converges to a stable solution θ^* , the EMA weights will converge to the same point:

$$\lim_{k \rightarrow \infty} \theta_k = \theta^* \quad \Rightarrow \quad \lim_{k \rightarrow \infty} \theta_k^{(\text{EMA})} = \theta^*. \quad (38)$$

Thus, EMA does not alter the asymptotic solution of training but accelerates the stabilization of the model parameters along the way. For these reasons, the use of EMA has become a standard component of diffusion-based generative modeling.

Normalization. For diffusion models to function effectively, the noise distribution and the data distribution must be placed on comparable scales. In practice, this is achieved by standardizing data to zero mean and unit variance and ensuring that the predicted noise is drawn from the same distribution. If the scales are mismatched, the learning task becomes unbalanced, and the denoiser may fail to converge. Additional techniques such as clipping the outputs or reweighting the relative contribution of signal and noise have been proposed to further stabilize training, depending on the problem setting.

Stable Diffusion. A major development in the design of diffusion models has been the introduction of latent diffusion models [RBL⁺22], commonly known through the Stable Diffusion framework. Instead of operating directly in the high-dimensional pixel space, diffusion is performed in the latent space of a pretrained autoencoder. The encoder compresses the input into a lower-dimensional representation that preserves semantic content, while the decoder reconstructs the final output. By applying the diffusion process in this compressed domain, the computational burden of each denoising step is greatly reduced, enabling large-scale training and sampling at practical cost. This approach is particularly effective for general-purpose image generation, where the latent space captures perceptual structure while discarding low-level detail. For scientific applications such as climate downscaling, latent diffusion offers a potential trade-off: reduced computation and larger receptive fields at the expense of fidelity to fine-scale physical variability, which may be lost in the autoencoding stage.

Timestep sampling during training. The distribution from which timesteps t are sampled during training influences convergence. If timesteps are sampled according to the inference schedule, training dynamics match the intended sampling process, and convergence is usually faster. Alternatively, sampling timesteps uniformly across the entire range allows one to swap inference schedules without retraining. However, uniform sampling often leads to unstable training or prevents convergence altogether, an effect observed consistently in our experiments.

Number of reverse steps. The number of steps T in the reverse process directly determines the computational cost of sampling, as each step requires a forward pass of the denoiser. Increasing T generally improves sample quality, as the noise is removed more gradually, but also increases variance in the generated samples and slows inference. Conversely, reducing T speeds up generation but may lead to oversmoothing or loss of detail. The trade-off between computational cost, fidelity, and diversity is a central aspect of diffusion modeling and a recurring theme of this thesis.

Conditioning strategies. For downscaling applications, unconditional diffusion is of limited value: an unconditional model would simply generate samples from the global high-resolution distribution, ignoring the low-resolution input entirely. Conditioning is therefore essential. In diffusion models, this conditioning is not achieved by making the model “deterministic” in the way of a GAN, but by explicitly injecting the conditioning signal into the denoiser network.

Formally, this can be written as extending the denoiser to depend not only on the noisy sample x_t and the timestep information t , but also on a conditioning variable c , for example the low-resolution field. In the noise-prediction formulation, the network thus takes the form

$$\epsilon_{\theta}(x_t, t, c),$$

where $c = x^{LR}$ provides the coarse-scale guidance.

The simplest and most commonly used strategy to implement this dependence is *direct concatenation*, illustrated in Figure 21. In this scheme, the denoiser receives as inputs the noisy high-resolution sample x_t , the low-resolution conditioning field x^{LR} , and the variance information α_t (or equivalently the timestamp t). These inputs are concatenated and jointly processed by the network, which outputs a high-resolution estimate of the clean sample. This setup ensures that every denoising step is explicitly guided by the coarse-scale information, while remaining faithful to the iterative generative nature of diffusion models.

Alternative conditioning mechanisms exist, such as *cross-attention*, where the low-resolution input is separately encoded and injected via attention layers, and *encoder fusion*, where a dedicated encoder processes the low-resolution field and fuses its features into the denoiser through concatenation or modulation. In this work, we focus on the effective direct concatenation strategy.

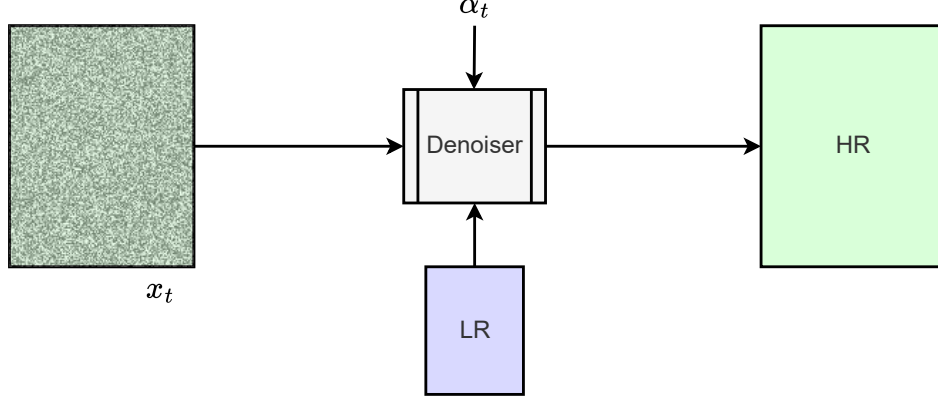


Figure 21: Illustration of a single reverse diffusion step with conditioning. The denoiser receives as input the noisy sample x_t , the conditioning low-resolution field LR, and the variance information α_t (or in a different formulation the timestamp t). It outputs a high-resolution estimate of the added noise, which is then used to reconstruct the denoised HR sample.

3.8 Loss Functions and Optimization Objectives

The choice of loss function is central to the training of super-resolution models: it defines the optimization objective, determines what aspects of the prediction are prioritized, and ultimately guides the learning dynamics of the network. Loss functions are typically computed between the ground-truth high-resolution (HR) field y and the model prediction $\hat{y}$, and may be designed to penalize pixel-wise discrepancies, structural differences, perceptual qualities, or physical inconsistencies.

For training super-resolution models, the standard choice is to minimize error metrics between the ground truth and the prediction. The most common are the mean squared error (MSE), mean absolute error (MAE):

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2, \quad (39)$$

$$\mathcal{L}_{\text{MAE}} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|, \quad (40)$$

$$(41)$$

These metrics are straightforward to compute and perform effectively in training models. Among them, MSE is particularly popular: by squaring the errors it penalizes large deviations more strongly, which stabilizes optimization but also leads to oversmoothing, a well-known limitation in imaging tasks. MAE, in contrast, is less sensitive to outliers but often results in slightly slower or less stable convergence. In practice, most architectures, ranging from MLPs and CNNs to U-Nets, vision transformers, and even diffusion models, are trained using either MSE or MAE as their primary optimization objective.

Beyond pixel-wise errors, perceptual metrics attempt to capture the similarity in terms of structure and texture. The Structural Similarity Index (SSIM) is a prominent example. For two image patches y and $\hat{y}$, SSIM is defined as

$$\text{SSIM}(y, \hat{y}) = \frac{(2\mu_y\mu_{\hat{y}} + C_1)(2\sigma_{y\hat{y}} + C_2)}{(\mu_y^2 + \mu_{\hat{y}}^2 + C_1)(\sigma_y^2 + \sigma_{\hat{y}}^2 + C_2)}, \quad (42)$$

where μ_y and $\mu_{\hat{y}}$ are the mean intensities of y and $\hat{y}$, σ_y^2 and $\sigma_{\hat{y}}^2$ are their variances, $\sigma_{y\hat{y}}$ is the covariance, and C_1, C_2 are small constants to stabilize the division. As a loss function, it is typically written as

$$\mathcal{L}_{\text{SSIM}} = 1 - \text{SSIM}(y, \hat{y}). \quad (43)$$

While SSIM has proven highly effective in computer vision for capturing perceptual quality, its application in climate downscaling remains limited, as it emphasizes structural similarity over physical

consistency. Other perception-based losses, such as the Learned Perceptual Image Patch Similarity (LPIPS), employ deep feature embeddings from pretrained networks. These are rarely used in climate applications, since they optimize for visual fidelity rather than physical correctness, but remain effective in selected applications.

In practice, pixel-wise metrics such as MAE ensure fidelity at the grid-cell level but may fail to capture general structures, while perceptual metrics such as SSIM emphasize global patterns but can overlook exact numerical accuracy. In some cases, a combined loss is used to balance these aspects, for example a combination of MAE and SSIM:

$$\mathcal{L} = \lambda \mathcal{L}_{\text{MAE}} + (1 - \lambda) \mathcal{L}_{\text{SSIM}}, \quad (44)$$

with $\lambda \in [0, 1]$ controlling the trade-off. Such hybrid formulations increase the complexity of the optimization landscape and are not always beneficial, but they may help if a model struggles to improve under a single objective.

Most super-resolution models are trained to directly predict the HR field $\hat{y}$. An alternative formulation is residual learning, already introduced in Section 3.2.7, where the model estimates the difference between the HR and the low-resolution (LR) input x_{LR} :

$$\mathcal{L}_{\text{residual}} = \ell((y - x_{\text{LR}}), f_{\theta}(x_{\text{LR}})). \quad (45)$$

This reduces the problem to modeling fine-scale corrections, which can be advantageous when computational resources are limited or when the residual signal is easier to learn than the full mapping.

Architecture-specific losses. The choice of loss function is also influenced by the architecture of the generative model. In the case of generative adversarial networks (GANs), training is formulated as a minimax game between a generator G and a discriminator D . The discriminator learns to distinguish real samples y from generated ones $G(x)$, while the generator is optimized to produce outputs that fool the discriminator. This yields adversarial losses of the form

$$\mathcal{L}_G = \mathbb{E}[\log(1 - D(G(x)))] + \alpha \mathcal{L}_{\text{perceptual}}, \quad (46)$$

$$\mathcal{L}_D = -\mathbb{E}[\log D(y)] - \mathbb{E}[\log(1 - D(G(x)))], \quad (47)$$

where the generator loss is often augmented with perceptual or pixel-wise terms to stabilize training.

Diffusion models adopt a very different training paradigm. Instead of fooling a discriminator, the model is trained to reverse a forward corruption process that gradually adds noise to the data. The standard objective is to predict the Gaussian noise ϵ injected at a given timestep t :

$$\mathcal{L}_{\text{diffusion}} = \mathbb{E}_{x,t,\epsilon} [\|\epsilon - \epsilon_{\theta}(x_t, t, c)\|^2], \quad (48)$$

where x_t is the noisy sample, ϵ_{θ} is the model prediction, and c denotes the low-resolution field conditioning. This formulation reduces to a simple mean squared error between the true and predicted noise, which explains why diffusion models, despite their complexity, are often optimized with the same basic objectives as CNNs or transformers.

Multi-task losses. In some settings, the network is trained to solve more than one task simultaneously. For example, in addition to predicting the high-resolution field, the model may be asked to predict multiple fields or estimate derived quantities such as anomalies, gradients, or physical indices. This setting requires a multi-task loss, typically formulated as a weighted sum of individual task objectives:

$$\mathcal{L}_{\text{multi}} = \sum_{k=1}^K \lambda_k \mathcal{L}_k, \quad (49)$$

where each $\mathcal{L}_k$ corresponds to the loss associated with task k , and λ_k is a weight that balances their contributions. Choosing appropriate weights is non-trivial: manual tuning is common, but adaptive weighting schemes based on task uncertainty or gradient magnitudes have also been proposed. In climate downscaling, multi-task objectives can be used to ensure that fine-scale reconstructions are not only visually accurate but also consistent with derived physical properties of the atmosphere.

Physics-informed losses. In some downscaling applications it is desirable to incorporate domain knowledge directly into the optimization. Physics-informed losses may penalize violations of conservation laws (e.g. energy, mass, momentum), spectral mismatches (e.g. differences in power spectra), or errors in derived physical indices. Such formulations remain an active research area, as they encourage models not only to match observations but also to respect fundamental physical constraints.

In addition to data-driven losses, one can introduce physics-informed losses that penalize violations of known physical relationships. These are implemented as *soft constraints*: instead of enforcing the physical law exactly, a penalty term is added to the loss function, weighted by a parameter that controls its influence.

An example is water conservation during spatial upscaling, relevant for fields such as precipitation or soil moisture. Let x be the low-resolution (LR) value of a grid cell, representing the average water content over its area, and let $\{y_i\}_{i=1}^n$ denote the corresponding n high-resolution (HR) sub-cells. Ideally, the fine-scale predictions should preserve the total water mass, meaning their average equals the LR value:

$$\frac{1}{n} \sum_{i=1}^n y_i = x. \quad (50)$$

To encourage this, a physics-informed penalty can be added to the training objective:

$$\mathcal{L}_{\text{phys}} = \left(\frac{1}{n} \sum_{i=1}^n \hat{y}_i - x \right)^2, \quad (51)$$

which is then combined with the standard data loss (e.g. MSE or MAE):

$$\mathcal{L} = (1 - \alpha) \mathcal{L}_{\text{data}} + \alpha \mathcal{L}_{\text{phys}}, \quad \alpha \in [0, 1]. \quad (52)$$

Here α is a hyperparameter that balances fidelity to observations against conservation. Larger values of α reduce water mass violation but can make optimization more difficult or slightly degrade metrics such as RMSE or SSIM.

The same principle can be applied to other balance laws, for instance energy fluxes, column-integrated water vapor, or momentum budgets, by tailoring $\mathcal{L}_{\text{phys}}$ to the relevant physical constraint.

Research on physics-informed and hybrid modeling is rapidly expanding, encompassing a variety of approaches. While these methods represent a promising path toward physically consistent downscaling, their exploration lies beyond the scope of this thesis, which instead focuses on data-driven models optimized purely through error or perceptual metrics.

3.9 Training Setup and Evaluation Strategy

The design of training and evaluation procedures in deep learning is inherently problem-dependent: different applications, models, and goals may require different approaches. The purpose of this section is therefore not to provide a definitive recipe, but rather to outline the general practices and principles followed throughout this thesis. The specific adaptations, tailored to the tasks considered, will be presented in greater detail in the individual papers and application chapters. In particular, the paper reported in Section 6 will delve into the requirements needed to assure fair comparison and effective training among different models. Within this general overview several common elements are worth highlighting, including the choice of optimizer, the scheduling of the learning rate, regularization and stability techniques, checkpointing strategies, and the evaluation protocol, as summarized in Table 9.

Optimizers are typically chosen among Adam and AdamW, which provide stable convergence with minimal tuning. In the case of AdamW, weight decay plays an important role in regularization; a common heuristic is to set it to approximately one order of magnitude smaller than the learning rate. Alternatives such as SGD with momentum or RMSProp are available but are less frequently employed in generative and downscaling settings.

Learning rate scheduling is another key component. Adaptive schemes such as ReduceLROnPlateau automatically lower the learning rate when the validation loss stagnates, but they can be unreliable if validation curves are noisy. A more controlled approach is manual scheduling: training begins with a relatively high learning rate (e.g. 10^{-4}) to encourage fast convergence, which is later reduced (e.g. 10^{-5}) once progress slows. This two-stage regime is simple, effective, and easy to reproduce. More

advanced alternatives, such as cosine annealing or warm restarts, exist but are not essential in the present setup. To improve stability, gradient clipping may be applied to prevent exploding gradients, especially in adversarial contexts such as GAN training.

Batch size is determined by hardware constraints: larger batches provide more stable gradients but require greater memory, normalization techniques such as group normalization can improve the effectiveness of low-batch training regimes. During training, model checkpoints are saved either by overwriting the best-performing model according to the validation metric, or by storing multiple snapshots at fixed intervals to provide flexibility when selecting the final model.

Aspect	Options
Optimizer	Adam, AdamW, SGD with momentum, RMSProp (Adam/AdamW most common)
Learning rate schedule	Reduce on Plateau, manual scheduling, cosine annealing, warm restarts
Regularization / stability	Weight decay, gradient clipping, dropout, normalization strategies (batch/group), batch size tuning
Checkpointing	Save best model by validation metric, periodic snapshots, hybrid strategies
Evaluation	Training/validation curves, held-out data, non-overlapping time splits (for temporal data), reporting metrics (RMSE, MAE, SSIM, PSNR, bias), selection metrics (MSE), multiple runs for robustness
Reproducibility	Fixed random seeds, dataset/code versioning, experiment logging (TensorBoard, callbacks)

Table 9: Overview of common design choices for training and evaluation

Evaluation is carried out systematically. Training and validation curves are inspected to ensure smooth convergence, with overfitting typically diagnosed by a growing gap between training and validation loss. Validation is always performed on data excluded from training. In the case of temporal datasets, non-overlapping time segments are used to avoid leakage and to mitigate confounding long-term trends such as climate change. Evaluation metrics are separated into two categories: those used for reporting (e.g. RMSE, MAE, SSIM, PSNR, and bias statistics) and those used for model selection (commonly a single validation score such as MSE). To account for stochastic variability due to random initialization and mini-batch sampling, if the available resource are sufficient multiple independent runs are compared, which can help distinguish genuine differences from random fluctuations.

Finally, reproducibility is ensured by fixing random seeds across libraries and maintaining consistent dataset and code versioning. Logging frameworks such as TensorBoard, combined with training callbacks, facilitate detailed monitoring of experiments and support systematic comparison across models and runs.

3.10 Libraries and Hardware Requirements

In practice, implementing complex neural architectures requires relying on dedicated frameworks that provide a higher level of abstraction. At their core, these frameworks are automatic differentiation libraries: they allow the forward pass of a model to be written in a natural way while automatically tracking operations and computing gradients for backpropagation. This is necessary for two main reasons. First, building models directly from basic mathematical operations would be prohibitively difficult and error-prone. Second, modern deep learning depends on efficient use of parallel hardware, primarily GPUs, which requires optimized low-level interfaces that are not practical to manage manually. A handful of frameworks are available to fulfill this role, most prominently TensorFlow, Keras, and PyTorch. Each offers distinct strengths and trade-offs, reflecting different priorities such as ease of use, research flexibility, or large-scale deployment. In this thesis, most models are implemented using the TensorFlow/Keras framework.

3.10.1 Deep Learning Frameworks

TensorFlow. TensorFlow is an open-source deep learning framework developed by Google, designed to support the entire machine learning workflow from research to large-scale deployment. Its core abstraction is the computational graph, which enables efficient execution and optimization of numerical operations across CPUs, GPUs, and TPUs. By default, TensorFlow evaluates operations eagerly in a Pythonic and imperative style, which makes model development and debugging straightforward. For performance-critical workloads, functions can be compiled into optimized static graphs using the `@tf.function` decorator. These graphs allow TensorFlow to apply kernel fusion, scheduling optimizations, and efficient execution on heterogeneous hardware. Further acceleration is available through the XLA (Accelerated Linear Algebra) compiler, which performs just-in-time compilation to reduce memory overhead and improve throughput. Beyond the computation engine, TensorFlow includes tools for dataset management (`tf.data`), visualization and profiling (TensorBoard), distributed and multi-GPU training, and deployment pipelines (TensorFlow Lite for mobile devices and TensorFlow Serving for production environments). Modern TensorFlow is now also highly integrated with Keras, which serves as an abstraction layer providing standard layer designs, model-building interfaces, and training utilities; Keras will be described in more detail in the following section. Together, these features make TensorFlow a versatile and widely adopted framework that bridges rapid prototyping and production-scale machine learning.

```
import tensorflow as tf

model = tf.keras.Sequential([
    tf.keras.layers.Dense(32, activation="relu"),
    tf.keras.layers.Dense(1)
])

optimizer = tf.keras.optimizers.Adam()
loss_fn = tf.keras.losses.MeanSquaredError()

for x, y in dataset:
    with tf.GradientTape() as tape:
        y_pred = model(x, training=True)
        loss = loss_fn(y, y_pred)
    grads = tape.gradient(loss, model.trainable_variables)
    optimizer.apply_gradients(zip(grads, model.trainable_variables))
```

Listing 1: TensorFlow model definition and training loop

In TensorFlow, models can be constructed by composing predefined layers from `tf.keras.layers`, which implement common building blocks such as dense layers, convolutions, and activations. Training relies on TensorFlow’s automatic differentiation system, which tracks all tensor operations. The central interface for this is `tf.GradientTape`, a context manager that records the forward pass and allows gradients to be computed automatically with respect to trainable variables. Once the loss has been evaluated, gradients are obtained through `tape.gradient` and applied to the model parameters with `optimizer.apply_gradients`. This explicit loop structure—forward computation, loss evaluation, gradient calculation, and parameter update—forms the basis of most custom training routines in TensorFlow. A minimal example of this workflow is shown in Listing 1.

Keras. Keras is a high-level deep learning API that was historically tied to the TensorFlow framework but can also be used on top of other automatic differentiation libraries, including PyTorch and JAX. It provides a user-friendly interface for model design and training, guided by the principles of simplicity, modularity, and rapid experimentation. This makes it particularly well suited for research and fast prototyping. Keras offers multiple modeling paradigms: the Sequential API for straightforward layer stacks, the Functional API for constructing flexible directed acyclic graphs of layers, and the Subclassing API for defining fully custom models and training loops. Beyond a large library of standard layers, losses, and optimizers, Keras also includes utilities for data preprocessing, callbacks for logging and checkpointing, and integration with TensorBoard. While prioritizing ease of use, Keras remains fully compatible with lower-level operations, enabling researchers to extend or customize any

component when needed. This combination of accessibility and extensibility has made Keras one of the most widely adopted deep learning interfaces in both academic and industrial contexts.

In Keras, models are typically defined by composing standard layers such as dense, convolutional, or transformer layers. The simplest approach is the **Sequential** API, which allows stacking layers in a linear order, while the Functional and Subclassing APIs are available for more flexible designs. Once a model is defined, it is compiled by specifying the optimizer, loss function, and evaluation metrics, after which training can be launched with a single call to `model.fit()`. This high-level interface abstracts away the details of the training loop (handling batching, shuffling, validation, and logging automatically) while still relying on the underlying engine for differentiation and GPU acceleration. A minimal example of this workflow is shown in Listing 2.

```
import keras

model = keras.Sequential([
    keras.layers.Dense(32, activation="relu", input_shape=(10,)),
    keras.layers.Dense(1)
])

model.compile(optimizer="adam", loss="mse")
model.fit(x_train, y_train, epochs=5, batch_size=32)
```

Listing 2: Keras model definition and training loop

Aspect	TensorFlow/Keras	PyTorch
Execution model	Default eager execution (since TF 2.x) with option to compile into static graphs using <code>@tf.function</code> ; XLA JIT compiler for further optimization.	Dynamic computation graph by default, with TorchScript for graph compilation and deployment.
High-level API	Keras: user-friendly, modular, with Sequential, Functional, and Subclassing APIs tightly integrated into TensorFlow.	Native <code>torch.nn.Module</code> class structure; ecosystem libraries (PyTorch Lightning, Hugging Face Accelerate) provide higher-level abstractions.
Distributed training	<code>tf.distribute</code> strategies: MirroredStrategy, MultiWorkerMirroredStrategy, TPUStrategy; seamless with <code>model.fit()</code> .	<code>torch.distributed</code> with NCCL/Gloo/MPI backends; <code>DistributedDataParallel</code> as standard; strong ecosystem support (Lightning, DeepSpeed, Accelerate).
Ease of prototyping	High, Keras abstracts much boilerplate, good for fast experimentation, but may require <code>@tf.function</code> for performance.	Very high, imperative style is coherent with python, most models behave similarly
Production deployment	Strong, TensorFlow Serving, TensorFlow Lite, TFX pipelines; wide adoption in industry and cloud.	Improving, TorchServe, ONNX export, but historically less mature than TensorFlow in production environments.
Community and adoption	Large community; dominant in industry, strong Google ecosystem integration (TPUs, cloud).	Extremely popular in academia and open research; many state-of-the-art models first appear in PyTorch.

Table 10: Comparison of TensorFlow/Keras and PyTorch

PyTorch. PyTorch is an open-source deep learning framework originally developed by Facebook’s AI Research (FAIR) group and released in 2016. It has become one of the most widely adopted

frameworks in the research community, largely due to its dynamic computation graph, which allows operations to be executed immediately in an imperative style. This design makes model development and debugging highly intuitive, while still supporting high-performance execution on GPUs via CUDA. PyTorch provides flexible abstractions for tensors, automatic differentiation, and neural network modules, enabling both rapid prototyping and the construction of complex architectures. Its ecosystem includes libraries for distributed and large-scale training (`torch.distributed`), model deployment (`TorchScript`, `ONNX`), and specialized domains such as natural language processing (`TorchText`), computer vision (`TorchVision`), and audio (`TorchAudio`). Although TensorFlow remains dominant in large-scale production environments, PyTorch is often preferred in academic research for its flexibility and ease of use, and many state-of-the-art models, including those in generative modeling, are first implemented in PyTorch before being ported elsewhere.

From a syntax point of view, in PyTorch models are defined by subclassing `nn.Module`, which provides a structured way to organize layers and parameters. A broad collection of standard components, such as linear layers, convolutions and loss functions are included in the `torch.nn` library, enabling models to be assembled from reusable building blocks. Training leverages PyTorch’s automatic differentiation engine, `autograd`, which records all operations in the forward pass. Once the loss is computed, calling `loss.backward()` automatically propagates gradients through the computational graph, and parameters are updated using `optimizer.step()`. Gradients are typically cleared at the beginning of each iteration with `optimizer.zero_grad()` to prevent accumulation across batches. A minimal example of this workflow is shown in Listing 3.

```
import torch
import torch.nn as nn
import torch.optim as optim

class SimpleNet(nn.Module):
    def __init__(self):
        super().__init__()
        self.fc1 = nn.Linear(10, 32)
        self.fc2 = nn.Linear(32, 1)

    def forward(self, x):
        return self.fc2(torch.relu(self.fc1(x)))

model = SimpleNet()
optimizer = optim.Adam(model.parameters())
criterion = nn.MSELoss()

for x, y in dataloader:
    optimizer.zero_grad()
    y_pred = model(x)
    loss = criterion(y_pred, y)
    loss.backward()
    optimizer.step()
```

Listing 3: PyTorch model definition and training loop

3.10.2 Distributed and Multi-GPU Training

Efficient use of GPUs is fundamental for training deep learning models. However, as model size and dataset complexity increase, a single GPU often becomes a limiting factor. Scaling to multiple GPUs is therefore essential. Broadly speaking, two levels of parallelism can be distinguished: multi-GPU training on a single node, and distributed training across multiple nodes, each equipped with several GPUs. Both TensorFlow and PyTorch provide built-in strategies and external libraries to support these scenarios, enabling users to extend training seamlessly from one device to large-scale clusters. Specifically, TensorFlow offers a rich set of strategies for distributed and multi-GPU training through its `tf.distribute` API. The most commonly used is `MirroredStrategy`, which replicates a model across multiple GPUs on a single machine and performs synchronous gradient updates, providing efficient data parallelism. For larger setups, `MultiWorkerMirroredStrategy` extends this approach to multiple nodes. TensorFlow also supports parameter-server strategies and asynchronous updates for workloads

that benefit from greater flexibility. These distribution mechanisms integrate with the `model.fit()` training loop in Keras, but can also be combined with custom training loops for finer control. Mixed-precision training is natively supported, reducing memory usage and accelerating computations on modern GPUs and TPUs. Also pyTorch provides extensive support for distributed and multi-GPU training. At its core, the `torch.distributed` package implements communication primitives based on backends such as NCCL, Gloo, and MPI, allowing efficient data exchange between GPUs and across nodes. The most widely used abstraction is DistributedDataParallel (DDP), which replicates the model across multiple devices and synchronizes gradients during backpropagation, offering near-linear scaling for many workloads. For simpler setups, `torch.nn.DataParallel` allows multi-GPU training on a single machine, though it is generally less efficient than DDP. PyTorch also integrates with higher-level libraries such as PyTorch Lightning, Hugging Face Accelerate, and DeepSpeed, which provide streamlined interfaces for distributed strategies, mixed-precision training, and large model optimization. Keras, as a multi-backend API, can in principle rely on the distributed strategies of its underlying engines such as TensorFlow or PyTorch. In practice, however, the TensorFlow integration is far more mature and offers straightforward configuration, making it the dominant choice for multi-GPU and distributed training with Keras.

3.10.3 Hardware

The training of deep learning models for downscaling is fundamentally dependent on GPU acceleration. In practice, most experiments rely on NVIDIA GPUs with CUDA support, which remain the dominant ecosystem for machine learning frameworks. Alternatives do exist, such as Apple Silicon accelerators or GPUs from other vendors, but they are less commonly used in research workflows. Even a single workstation equipped with one or a few GPUs can be sufficient to train moderately complex models or to prototype new approaches effectively. Large-scale applications, however, often require access to high-performance computing (HPC) infrastructure and the use of distributed training strategies. Supercomputers such as Leonardo [TAC23], with thousands of nodes and four high performance GPUs per node, enable the training of models on vast datasets and at resolutions beyond the capabilities of local machines. Besides raw compute, sufficient high-speed storage and memory bandwidth are also essential, since downscaling datasets can be terabytes in size and model training requires rapid data throughput.

4 Paper I: Wind speed super-resolution and validation, from ERA5 to CERRA via diffusion models

Abstract

The Copernicus Regional Reanalysis for Europe, CERRA, is a high-resolution regional reanalysis dataset for the European domain. In recent years, it has shown significant utility across various climate-related tasks, ranging from forecasting and climate change research to renewable energy prediction, resource management, air quality risk assessment, and the forecasting of rare events, among others. Unfortunately, the availability of CERRA is lagging two years behind the current date, due to constraints in acquiring the requisite external data and the intensive computational demands inherent in its generation. As a solution, this paper introduces a novel method using diffusion models to approximate CERRA downscaling in a data-driven manner, without additional information. By leveraging the lower resolution ERA5 dataset, which provides boundary conditions for CERRA, we approach this as a super-resolution task. Focusing on wind speed around Italy, our model, trained on existing CERRA data, shows promising results, closely mirroring the original CERRA. Validation with in-situ observations further confirms the model’s accuracy in approximating ground measurements.

4.1 Introduction

The first paper of this thesis focuses on the application of neural super-resolution methods to the downscaling of reanalysis data. Specifically, we investigate several architectures, including CNNs, GANs, Transformers, and diffusion models, for downscaling wind speed from ERA5 to CERRA. These architectures have been previously introduced and analyzed in Section 3. As discussed in Section 2, deep learning has enabled substantial progress in super-resolution across multiple domains, yet no single architecture has emerged as the optimal choice in meteorology. In this work, we assess not only their relative performance but also the broader feasibility of neural networks as a data-driven alternative to physics-based downscaling. Among the tested models, a conditioned diffusion approach achieves the best overall performance.

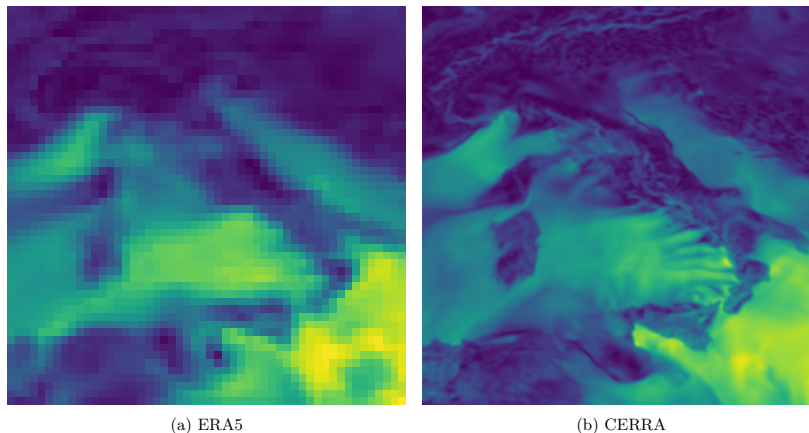


Figure 22: Comparison of wind speed over Italy between ERA5 and CERRA

In this paper, we focus on two reanalyses, ERA5 and CERRA, which have been introduced in Section 2.1. Traditional reanalysis downscaling relies on physics-based models such as HARMONIE, which are computationally intensive and depend on multiple external data sources. These requirements contribute to substantial publication delays: at the time of writing, CERRA is more than two years behind ERA5, limiting its usefulness for the study of recent events. Such delays have been particularly problematic in cases like the May 2023 Emilia-Romagna floods and the November 2023 Storm Ciarán, where timely access to high-resolution reanalysis would have supported faster scientific assessment.

In this study, we aim to demonstrate the general feasibility of neural super-resolution for data-driven downscaling. Using ERA5 as input and CERRA as reference, we focus on wind speed over Italy, training a conditioned diffusion model and benchmarking it against alternative architectures. Additionally, validation with in-situ observations ensures correspondence with ground truth measurements.

The main contributions of this paper are: (i) demonstrating that neural methods can effectively approximate physics-based downscaling, (ii) establishing a benchmark comparison of state-of-the-art architectures for this task, (iii) showing that ensemble diffusion models outperform other approaches, and (iv) releasing an approximated CERRA dataset for recent years where no official data are yet available. These results establish both the feasibility and the added value of neural downscaling, and provide the foundation for the subsequent studies presented in this thesis.

4.2 Methodology

In this section we provide a detailed exposition of our diffusion technique applied to super-resolution in statistical downscaling, which includes a comprehensive validation component. Our discussion begins with an overview of the preliminary steps required to prepare our case study, including the harmonization of features from various datasets. We then offer a detailed exploration of our primary and most effective model, the diffusion model. Following this, we compare the diffusion model with other super-resolution approaches to highlight relative strengths and weaknesses. Next, we outline the validation methods employed to ensure the accuracy of our model’s output against real-world measurements. Lastly, we evaluate our results and provide an in-depth analysis of our findings.

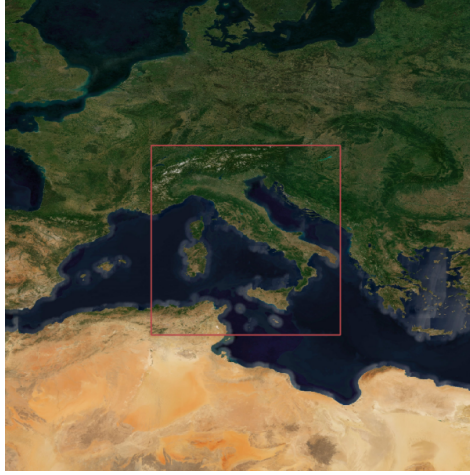


Figure 23: Selected area for our study, delimited by the following geographical coordinates: North at 47.75° , South at 35° , East at 18.75° , and West at 6° .

4.2.1 Data Selection & Preprocessing

In this section, we analyze the essential preprocessing steps required for defining the downscaling process from ERA5 to CERRA. Both ERA5 [HBB⁺18] and CERRA [SRLM⁺21] are freely available from the Copernicus Program [Cop23] website.

In our study, we have chosen to concentrate on the meteorological variable of wind speed. Several factors influenced this decision. Firstly, we aimed to simplify our experimental framework by focusing on a single variable. We selected wind speed because, compared to variables like rainfall, it is more physically self-contained and thus easier to analyze in isolation. Additionally, wind speed holds significant societal relevance, particularly in the context of extreme weather events and its implications for energy generation in wind farms.

We utilized wind speed at 10 meters above the surface. In the case of CERRA wind speed is already included as a main variable, while for ERA5 we computed it from both the zonal u and the meridional v wind components by $\sqrt{u^2 + v^2}$. The utilized values are instantaneous.

For our analysis, we extracted a specific region from the ERA5 dataset, delimited by the following geographical coordinates: North at 47.75° , South at 35° , East at 18.75° , and West at 6° . This targeted extraction ensures a focused study area for our research. This area coincides with a square containing Italy, Switzerland, Slovenia and partially Tunisia, Algeria, France, Austria, Croatia, Hungary, Bosnia & Herzegovina and Montenegro, as shown in Figure 23.

As mentioned in Section 2.1.2, ERA5 and CERRA have different projection mechanism, with ERA5 using a cylindrical projection and CERRA a Lambert Conformal Conic projection. Given that our study area is located at mid-latitudes, where distortions in cylindrical projections are relatively minimal, we chose to equalise the two projections by re-projecting CERRA using a cylindrical projection consistent with ERA5. The projection was realized utilizing the CDO [Sch23] library.

CERRA provides data samples with a 3-hour temporal resolution, whereas ERA5 offers an hourly resolution. To align these datasets, we extracted ERA5 measurements corresponding to the times 00:00, 03:00, 06:00, 09:00, 12:00, 15:00, 18:00, and 21:00, matching CERRA’s schedule. Our training set encompasses data from 2010 to 2019. For testing purposes, we selected the years 2020 and 2009. The choice of non-contiguous test years is motivated by aiming to widen our evaluation range and the availability of in-situ observations for validation.

The training sets for both ERA5 and CERRA have been normalized to a range between 0 and 1, with each entry being divided by the maximum value within its respective dataset. Similarly, the test sets for each dataset have been normalized using the same maximum value as determined from their corresponding training sets.

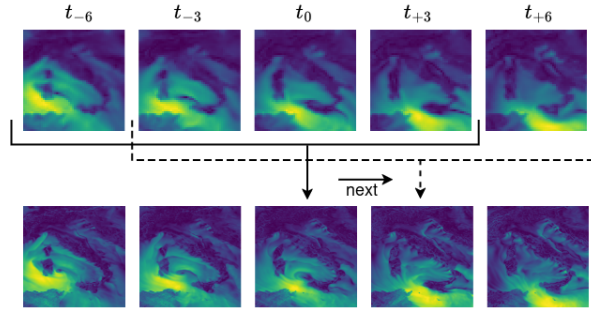


Figure 24: The downscaling diffusion process utilize as input a set of low resolution images. For predicting the high resolution image at time t_0 the conditioning information include the low resolution images at times t_{-6} , t_{-3} , t_0 , t_{+3} where each step is 3 hours long.

4.2.2 Diffusion Model Architecture

In this section, we describe our super-resolution diffusion model. This includes an examination of the super-resolution (SR) framework, the specifics of the conditioning process, ensemble methods, and various implementation details.

4.2.3 SR framework

As introduced in Section 3.2, super-resolution for neural networks implies the use of some form of upsampling, which may be implemented in different ways and in different positions within the network. Generally speaking, we can define two frameworks for SR: pre-upsampling, when the upsampling is done in the first part of the network, and post-upsampling, when the upsampling part is implemented at the very end of the network. If the upsampling is performed at the start, the spatial dimension of the features is increased, usually granting the network better overall performance at the expense of computational complexity. The first framework is generally used in setting where the main target is performance, such as SRGAN [WYW+18], the latter is more common in performance-oriented model such as ESPCN [SCH+16]. For our diffusion model, we opted for a pre-upsampling framework.

The process of upsampling can be executed using various methods. Some methods employ learnable parameters, such as transposed convolution, while others, like 2D upsampling or bilinear and bicubic upsampling, do not use learnable parameters. In our model, we opted for pre-emptive bilinear upsampling, due to its practicality. Additionally, our experiments with methods involving upsampling with learnable parameters yielded similar results.

For what concerns the super resolution task setting, we decide to utilize a sequence-to-one approach, as represented in Figure 24. In this approach, the super resolution model is trained to produce the high resolution images while having access to multiple timestamps of the low resolution counterpart.

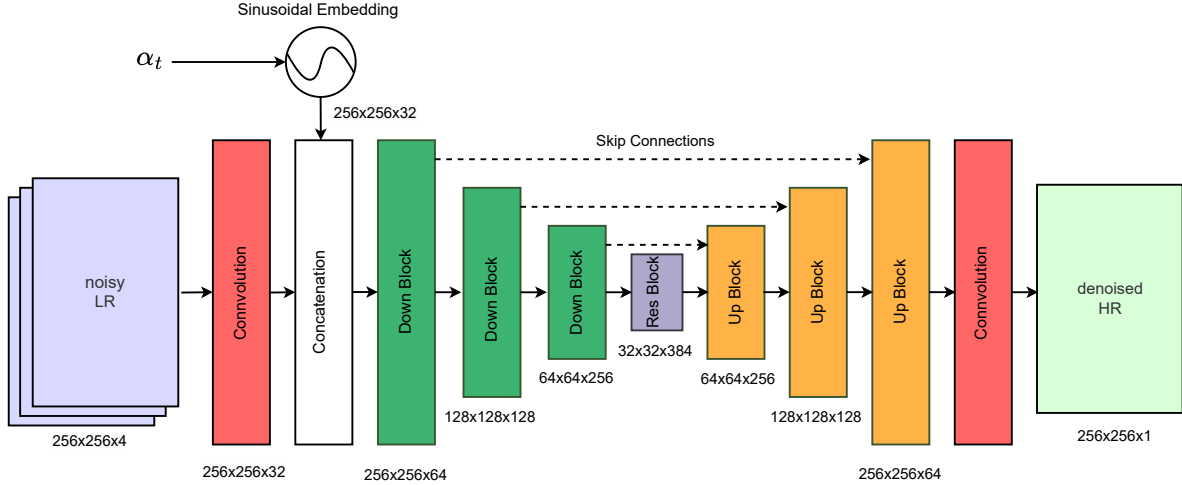


Figure 25: The denoising U-net employs a conventional architecture, consisting of three downsampling blocks implementing average pooling, followed by three upsampling blocks using bilinear interpolation. These stages are interconnected through skip connections, and each block incorporates residual connections.

Specifically, experimental results showed that the optimal sets of conditioning information for the prediction of the high resolution image at time t_0 contains the low resolution images at hours t_{-6} , t_{-3} , t_0 , t_{+3} . In our implementation, the main bulk of our data is kept on disk while a data generator feeds the model the sequences for both training and the testing operations, with the first being a random set of sequences and the second a contiguous set.

4.2.4 Denoising and conditioning

Diffusion models essentially operate as iterative denoising algorithms. Their main trainable component is the denoising network, denoted as $\epsilon_\theta(x_t, \alpha_t)$. This network receives as input the noisy input, x_t , and a corresponding noise variance, α_t , with the objective of estimating the amount of noise infiltrating the image. The training of this underlying denoising network is done conventionally. An initial sample, x_0 , is extracted from the dataset and subjected to a predefined amount of random noise. The network is then tasked with estimating the noise within these corrupted images.

Our model of choice for the denoising network is the U-net. The U-net is one of the most common architectures for denoising [GRDA21, LNU⁺20, HSB18, KG20] and it is often implemented in diffusion models [DN21]. Originally introduced for semantic segmentation [RFB15a], the U-Net architecture has gained widespread popularity and found applications in diverse image manipulation tasks. The network comprises a downsampling sequence of layers, followed by an upsampling sequence while incorporating skip connections between layers of the same size, as reported in Figure 25. Our implementation employs bilinear interpolation for the upsampling process and utilizes average pooling for the downsampling procedure.

The U-Net configuration can be determined by defining the number of downsampling blocks and the number of channels for each block. The upsampling structure follows a symmetric pattern, and the spatial dimension is dependent on the number and resolution of the input and output images. Our U-net utilize 4 downsampling blocks with increasing number of channels, with the number of channels being 64, 128, 256, 384. With the number of channels increasing as the spatial dimension decrease. This dimension of U-net proved to be the most effective experimentally, and amounts to around 20 million parameters. To improve the sensibility of the U-net to the noise variance, α_t is taken as input, which is then embedded using an ad-hoc sinusoidal transformation by splitting the value in a set of frequencies, in a way similar to positional encoding in Transformers [VSP⁺17b]. The embedded noise variance is then vectorized and concatenated to the noisy images along the channel axes before being passed to the U-Net. This helps the network to be highly sensitive to the noise level, which is crucial for good performance.

Conditioning of the model is necessary to guide the diffusion towards a forecast defined by the known previous weather conditions. Our conditioning is applied in a classifier-free manner, by concatenating the conditioning frames to the noisy images alongside the channel axis. Practically, the model $\epsilon_\theta(x_t, t, c)$ takes as input the noisy images x_t , which represents the high resolution wind speed at time t_0 . The conditioning information $c = \{w_{-6}, w_{-3}, w_0, w_{+3}\}$ contains the low resolution wind timestamps at -6 hours, -3 hours, current time and +3 hours ahead.

The input fed into our denoising U-net consists of four pre-emptively upsampled ERA5 images, along with a single image that represents the noisy input. This configuration outputs a singular, denoised image. Given that the image size is set at 256×256 pixels, the input dimension is structured as (batch_size, 256, 256, 5), while the output dimension is formulated as (batch_size, 256, 256). Our implementation directly provides the U-net with the conditioning information, specifically, each temporal slice in the input data is treated analogously to a color channel in an RGB image.

4.2.5 Ensemble Diffusion

Similarly to [AMP⁺23], in our work we leverage the ensemble properties of diffusion models. With the goals of diffusion models being the approximation of the distribution of the training data, it is expected that our generative outcome is the outcome of a probabilistic point in the data distribution. Thus, it is common that for diffusion processes with the same conditioning that the results may vary considerable with different executions. This behaviour of diffusion models may impair the prediction effort, because a single generation may diverge considerably from the mean of the distribution, producing a convincing although highly unlikely outcome inside the distribution. A possible solution to this problem lies in computing the mean of a set of diffusion generations, thus moving our generated image close to the mean value of the distribution and therefore generating a more probable outcome.

Algorithm 3 Noise Scheduler

```

1: next_noisy_images ← initial_noise
2: for step in diffusion_steps do
3:   noisy_images ← next_noisy_images
4:   noisy_variance ← diffusion_schedule(step)
5:   pred_noise, pred_image ← denoising_unet(noisy_images, noisy_variance)
6:   next_variance ← diffusion_schedule(next(step))
7:   next_noisy_images ← pred_image + next_variance * pred_noise
8: end for

```

In our experiments, we settled for an ensemble numbering 15 executions of the single diffusion model, with each diffusion being comprised of 5 steps. This combination proved to be the optimal configuration with respect to our testing years. Unlike [AMP⁺23], our approach simply calculates the mean between the predicted instances instead of utilizing a post-processing model; this simpler approach showed comparable results with a lower computational overhead.

In Figure 26 a general overview of the ensemble architecture for evaluation is reported. Starting from the left, we showcase that the inputs are the noise and 4 low-resolution conditioning images. Pre-emptive bilinear upsampling is performed on the low-resolution images before the insertion into the model. Next, the model performs 5 diffusion steps of sequential denoising, with a noise scheduler managing the re-addition of the noise and the computation of noise rates. The noise scheduling process requires the generation of a linear schedule, with denoising operation alternated to re-addition of noise at a lower rate, as reported in Algorithm 3. This operation is performed 15 times as part of the ensemble effort, using the same conditioning information but different input noise. The cumulated outputs are finally combined via a mean operation to produce the downscaled output. This operation is effectively performed in a parallelized manner with a batch size of 32.

4.2.6 Training and Evaluation

The training procedure of diffusion models, as reported in Algorithm 1, is implemented by randomly selecting an amount of noise, which is subsequently added to the input images, with the denoising U-net tasked to denoise the image. For the evaluation phase the process is reversed, applying the so called reverse-diffusion, as reported in Algorithm 2. The reverse diffusion process involves a sequence

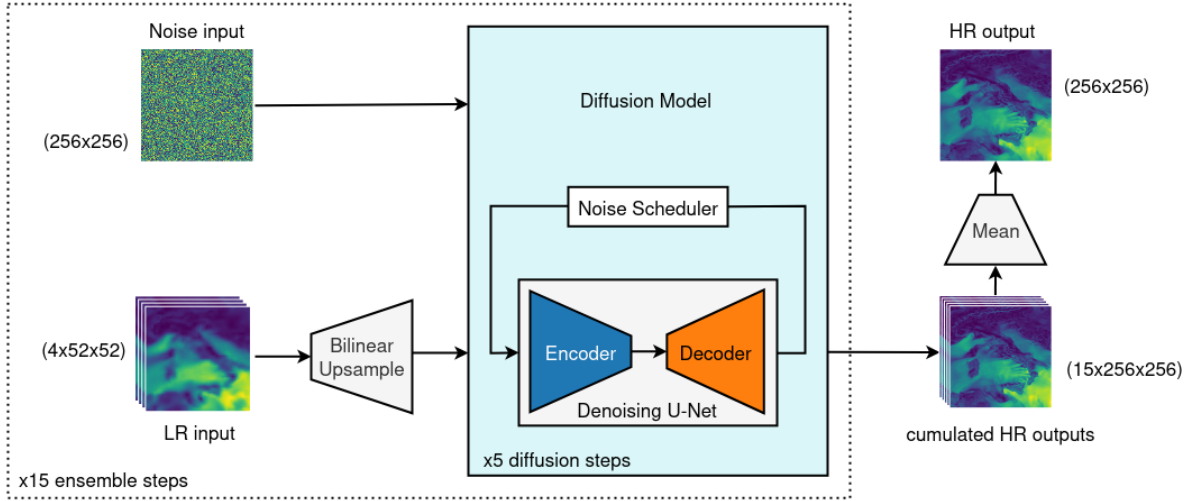


Figure 26: The super-resolution diffusion framework employs a sequence-to-one strategy, where pre-upsampling is achieved via bilinear interpolation. The diffusion process comprises five diffusion steps, using a denoising U-net to progressively denoise the input. A noise scheduler orchestrates this process as reported in Algorithm 3. This procedure is replicated 15 times, generating an ensemble of potential super-resolution outcomes. These outcomes are then amalgamated using a mean operation to produce the final output.

of steps, beginning with the introduction of input noise. This procedure alternates between denoising steps and re-injection of noise, as depicted in Figure 27. These steps are managed via a noise scheduler, which dictates a sequence of noise rates following a linear progression. For instance, in a 5-step process, the noise rates might be set as 1, 0.8, 0.6, 0.4, and 0.2, respectively. The process starts with a denoising operation at a noise rate of 1, then noise is re-injected at a rate of 0.8, followed by another denoising step at the same rate, and the pattern continues through the sequence.

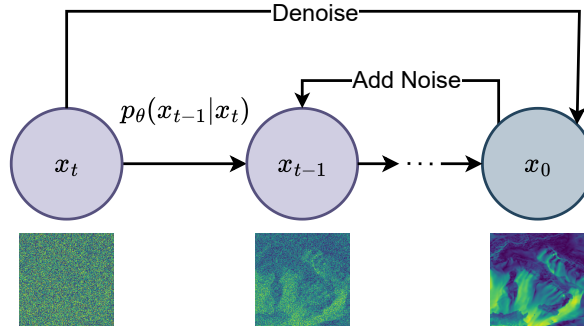


Figure 27: The reverse diffusion process is a sequential method that involves alternating between denoising steps and the re-injection of noise, progressively refining the generated image.

Our model is developed using the TensorFlow/Keras framework. We trained the model on data from 2010 to 2019, and tested it on data from 2009 and 2020, which allowed us to verify its performance across time periods before and after our training years. The training employs a data generator for efficient handling of training batches, creating custom sequences on the fly, and uses memory mapping to load only the necessary data into RAM. The selection of data batches is randomized over the ten-year training period. The training was conducted in batches of 8 over 200 epochs, using the AdamW optimization algorithm. To optimize training, we implemented a callback function that adjusts both the learning rate and weight decay. The learning rate starts at 1e-04 and is gradually reduced to 1e-05, while weight decay starts at 1e-05 and decreases to 1e-06. The training loss, coherently with most diffusion model implementation utilized MAE computed on the noise differences. In the evaluation phase, we employed a generator to produce continuous sequences for each testing year. We configured

the batch size to be 32. Taking into account the 3-hour time resolution, this approach resulted in batches representing 96-hour (4-day) periods.

4.2.7 Model Comparison

In our work, we selected a set of well known neural models to compare our diffusion architecture with. Our selection include, Efficient Sub-Pixel Convolved Neural Network (ESPCNN) [SCH⁺16], Enhanced Deep Super-Resolution Network (EDSR) [LSK⁺17], the residual U-net, Super-Resolution Generative Adversarial Network (SRGAN) [LTH⁺17], Vision Transformer (ViT) [DBK⁺20] and Swin Transformer with Long Short-Term Memory (SwinLSTM) [TLZT23].

Parameters and execution times		
Model	Parameters (M)	Time (s)
ESPCNN[SCH⁺16]	0.72	23
EDSR[LSK⁺17]	2.17	29
SRGAN[LTH⁺17]	4.31	63
ViT[DBK⁺20]	23.23	69
SwinLSTM[TLZT23]	3.41	134
U-net[RFB15a]	19.97	246
Single Diffusion	19.97	385
Ensemble Diffusion	19.97	7281

Table 11: Comparative analyses of the tested models, including the number of parameters (millions) and the execution time (seconds) for the computation of one year of data.

ESPCNN is a well known neural architecture for super-resolution, which utilize sub-pixel convolution as a form of efficient post-upsampling at the very end of the network. This design choice enables the convoluted portion of the network to function at a lower resolution feature dimension, significantly enhancing the network’s efficiency. As a result, ESPCNN maintains a low parameter count, making it not only resource-efficient but also faster in processing. This efficient architecture ensures that ESPCNN delivers high-quality super-resolution images while requiring fewer computational resources, a key advantage in practical applications.

The Enhanced Deep Super-Resolution Network (EDSR) is a well known architecture in the realm of super-resolution, renowned for its versatility and effectiveness across a broad spectrum of tasks. EDSR architecture is distinguished by its assembly of multiple residual blocks. A key innovation in EDSR is its use of constant scaling layers, which replace the traditional batch normalization layers. EDSR’s design, focused on optimizing image detail and quality, has solidified its status as a top-tier solution in the ever-evolving field of super-resolution technology.

The residual U-net is a U-net implementation which is internally used by the diffusion model to perform the denoising part. The network utilize residual blocks and implements downscaling via average pooling and upsampling via interpolative resizing. It also excels as a standalone super-resolution model. Its effectiveness in this setting is noteworthy, providing an effective benchmark against the diffusion model. This comparison is particularly significant due to the architectural parallels and similarities in parameter count between the two models. Such shared characteristics allow for a more accurate understanding of their respective strengths and capabilities in the field of super-resolution.

SRGAN [LTH⁺17] is a renowned method for super-resolution that employs the generative adversarial network (GAN) framework, featuring a generator-discriminator setup engaged in a min-max optimization game. GAN-based methods continue to maintain a significant presence in super-resolution tasks, as evidenced by numerous reviews and challenges, underscoring the necessity of including SRGAN in comparative analyses within our research context. The SRGAN study detailed a broad spectrum of architectural decisions, with the most prevalent implementation incorporating a perceptual loss component into the generator’s loss function. However, our experiments indicated that utilizing a mean squared error (MSE)-based non-perceptual loss, in conjunction with a Patch-GAN [IZZE17] discriminator, proved to be more efficacious than its perceptual counterpart. This approach enabled us to obtain optimal performance outcomes.

Vision Transformers (ViT) represent a significant advancement in the field of computer vision, primarily by adapting the transformer architecture, which has been highly successful in natural language processing, to image analysis tasks. Introduced by researchers in 2020 [DBK⁺20], ViTs approach image manipulation in a novel way by treating images as sequences of patches, similar to how transformers treat text as sequences of words. This method begins by splitting an image into fixed-size patches, then linearly embedding each of them, similar to token embedding in NLP. The sequence of embedded patches, supplemented by a learnable positional encoding, is then processed through a series of transformer layers. This process allows the model to focus on different parts of the image and learn contextual relationships between patches, leveraging the self-attention mechanism inherent to transformers. This process is then followed by a reconstruction structure which reassembles the patches into an output image. Our implementation utilize a patch size of 8×8 , which is projected into an embedding of 256 float values via a dense layer. We utilize 6 transformer layers in parallel, each with 12 heads and concatenated outputs. The reconstruction process is implemented via transposed convolution upsampling alternated with residual blocks.

SwinLSTM [TLZT23] is a state of the art transformer-based architecture that combines the strengths of Swin Transformer and Long Short-Term Memory (LSTM) networks. This architecture was successfully applied to a variety of spatio-temporal prediction tasks, where the model needs to capture both spatial and temporal dependencies in the data. This can be critical for super-resolution tasks where the model needs to understand the context and dependencies between pixels or frames. This makes this transformer based technique especially viable for setting, where using a single low-resolution image to predict the high-resolution one would not be enough to provide the information necessary to achieve optimal results.

In Table 11 we report a comparison between the tested models, reporting for each one the number of parameters and the execution time necessary to compute one year of downscaled data.

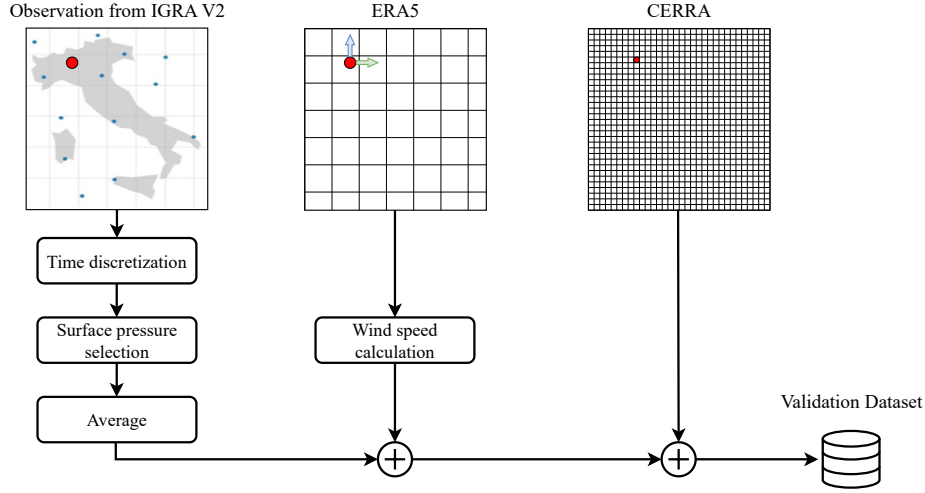


Figure 28: Construction process of the dataset used for the validation of ERA5, CERRA and the generated data. IGRA V2 observations in the reference period undergo filtering depending on the reference time and the air pressure value, after which the values selected for an observation at a given time are averaged and become part of the dataset. At this point, the corresponding values of ERA5 (after calculating the wind speed from the two u and v components) and CERRA are concatenated to each observation. The same process is implemented for the values resulting from the various generative models.

4.3 Validation

The validation process utilizes in-situ observations from weather stations as the ground truth against which the model predictions are evaluated. This process also results in an assessment of the quality of the reanalysis proposed by ERA5 and CERRA.

The IGRA V2 dataset provides for each measurement the value of the wind speed at different pressure levels, where the maximum value coincides with the altitude of the weather station. To remain consistent with the ERA5 and CERRA data, only the observation with the highest air pressure value was selected for each measurement.

Since the data available for the other datasets refer to measurements relative to the previous hour and with 3-hour intervals, the in-situ data considered must also follow a coherent structure. Therefore, given the irregular timestamp distribution of IGRA data, being mainly concentrated around midnight and midday, we decided to remove all detections that did not occur in these two time slots. For what concerns multiple measurements carried out by the same meteorological station during the hour preceding these two slots, these have been reduced to a single average value and the related timestamp rounded to the stroke of the following hour.

To allow a spatially coherent comparison we decided to map all the values to the spatial representation of the in-situ data, in order to keep track of only the values present in ERA5, CERRA and in the results of the models for which there is a correspondence in the validation dataset, thus greatly reducing the space required.

The procedure for assembling the validation dataset is described in Figure 28. The obtained dataset can be used to calculate the difference between the results obtained from the various models used to approximate CERRA and the ground truth represented by the observations. Although this does not represent our main objective, this type of analysis also allows us to obtain valuable information on the performance of ERA5 and CERRA themselves as reanalysis datasets.

4.4 Experiments and Results

In this section, we detail the experiments conducted using the models and data referenced in the preceding sections and provide an analysis of the results obtained.

For conducting our experiments, we strategically chose two distinct years, 2009 and 2020, as our testing periods. For both these years, we utilized the lower resolution ERA5 dataset as our conditioning alongside the CERRA dataset, which acted as our high-resolution ground truth. Additionally, we leveraged the IGRA V2 in-situ dataset, containing ground-based observations from meteorological stations, for validating the data generated by our experiments.

The primary objective of our experiment is to evaluate whether a neural-network-based super-resolution model can effectively replace a traditional downscaling model, like the physics-inspired HARMONIE model currently employed for the generation of CERRA. A key advantage of our neural model is its independence from supplementary data requirements, alongside significantly reduced computational demands. Our comparative analysis studies eight competing neural models: ESPCNN, EDSR, SwinLSTM, SRGAN, U-net, ViT, Single Diffusion, and Ensemble Diffusion. This general comparison includes a variety of models from small, efficient networks all the way to state of the art, high computational demand approaches. This assessment aims to determine not only the feasibility but also the relative performance of these neural models in the context of super-resolution tasks for weather variables.

All the models are trained on a 10-year period with a sequence to one approach, utilizing 4 conditioning low resolution time frames, as described in Section 4.2.1. The models are then evaluated on the two respective years. A summary of the obtained results is reported in Table 12, where we can note that all neural models have a considerably better performance with respect to our traditional downscaling baseline of bilinear interpolation. The model that achieves the best result, coherently for all three metrics, is Ensemble Diffusion. Ensemble Diffusion in comparison to bilinear interpolation improves the MSE value from $2.50\text{e-}03$ to $1.02\text{e-}03$ and provide respectively a 15% improvement in PSNR and 20% improvement in SSIM. A set of results for visual inspection is reported in Figure 29.

The data generated through our process underwent validation against in-situ measurements, providing a crucial comparison to determine if the downscaled version offered tangible improvements over real-world observations. As indicated in Table 13, the validation confirmed that our downscaled data align more closely with the actual wind speed measurements in the area, demonstrating enhanced fidelity.

For our evaluation dataset, we chose the Mean Absolute Error (MAE) as our preferred metric for assessing errors. We selected MAE due to its tangible physical interpretation; being measured in meters per second, it provides a clear indication of the magnitude of the error.

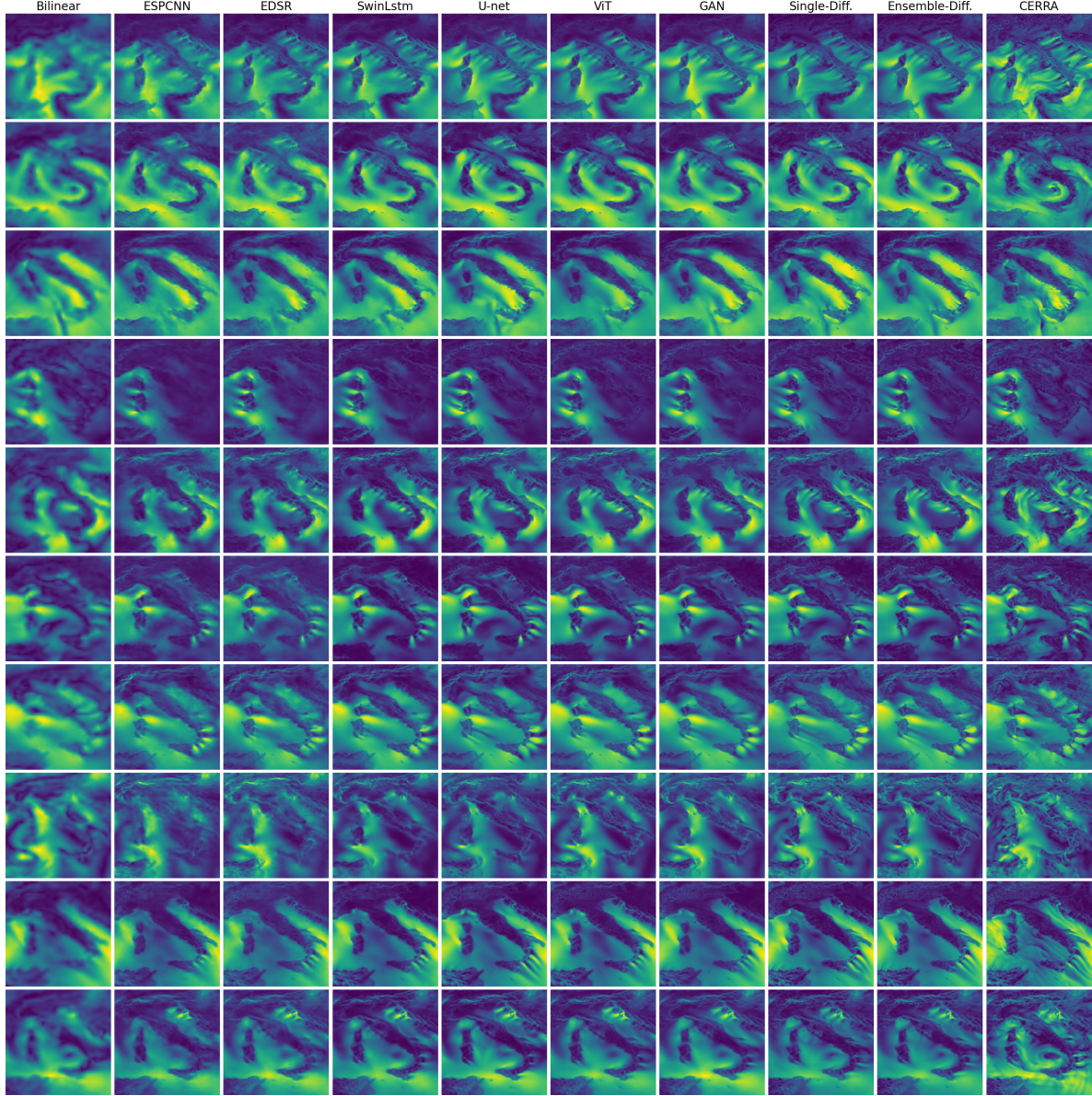


Figure 29: Visual evaluation of super-resolution methods for ten selected samples from the years 2009 and 2020 showcasing different weather conditions, including a Mediterranean cyclone. The comparison contain all tested neural downscaling methods alongside the CERRA model. Visual evaluation reveals that U-net, ViT, GAN and Diffusion methods provide the visually best results, with the diffusion models achieving the higher amount of details. The numerical results reported in Table 12 provide a more effective comparison of the respective performances.

Wind speed ERA5 to CERRA Downscaling			
Model	MSE ↓	PSNR ↑	SSIM ↑
2009			
Bilinear	2.50e-03	26.36	0.708
ESPCNN[SCH⁺16]	1.31e-03	29.11	0.773
EDSR[LSK⁺17]	1.20e-03	29.48	0.796
SwinLSTM[TLZT23]	1.14e-03	29.70	0.802
SRGAN[LTH⁺17]	1.10e-03	29.85	0.813
U-net[RFB15a]	1.16e-03	29.65	0.819
ViT[DBK⁺20]	1.09e-03	29.94	0.822
Single Diffusion	1.18e-03	29.62	0.829
Ensemble Diffusion	<u>1.06e-03</u>	<u>30.11</u>	<u>0.844</u>
2020			
Bilinear	2.38e-03	26.67	0.712
ESPCNN[SCH⁺16]	1.25e-03	29.31	0.775
EDSR[LSK⁺17]	1.14e-03	29.70	0.797
SwinLSTM[TLZT23]	1.10e-03	29.99	0.804
SRGAN[LTH⁺17]	1.05e-03	30.07	0.814
U-net[RFB15a]	1.10e-03	29.86	0.820
ViT[DBK⁺20]	1.05e-03	30.18	0.824
Single Diffusion	1.13e-03	29.84	0.831
Ensemble Diffusion	<u>1.02e-03</u>	<u>30.32</u>	<u>0.845</u>

Table 12: Results comparison for multiple models on wind downscaling for the years 2009 and 2020.

Wind speed validation for the year 2009	
Model	MAE ↓
ERA5	2.04
CERRA	<u>1.86</u>
Bilinear	1.96
ESPCNN[SCH⁺16]	1.90
EDSR[LSK⁺17]	1.92
SwinLSTM[TLZT23]	1.90
SRGAN[LTH⁺17]	<u>1.86</u>
U-net[RFB15a]	1.88
ViT[DBK⁺20]	<u>1.86</u>
Single Diffusion	1.90
Ensemble Diffusion	<u>1.86</u>

Table 13: Validation of the results obtained compared to the observations present in the IGRA V2 dataset. The comparison takes into consideration the Mean Absolute Error (MAE), measured in meters per second, and the Mean Squared Error (MSE) calculated on the in situ observations and their corresponding values generated by the models or present in ERA5 or CERRA.

Our validation analysis revealed that CERRA outperforms ERA5 in terms of real-world data correlation, achieving a 8.82% improvement in Mean Absolute Error (MAE). Among the downscaling models we tested, The Ensemble Diffusion, SRGAN, and ViT methods emerged as the most effective, demonstrating a performance enhancement of 8.82% over ERA5, which is the same level of performance demonstrated by CERRA. These results showcase the capabilities of neural models to effectively approximate the inner mechanics of state-of-the-art physics inspired model, providing both lower computational costs and avoiding the need for additional information that are often difficult to obtain, with common delays in the publication of current data. Our validation was possible only for the year 2009, as validation data was not available for the year 2020.

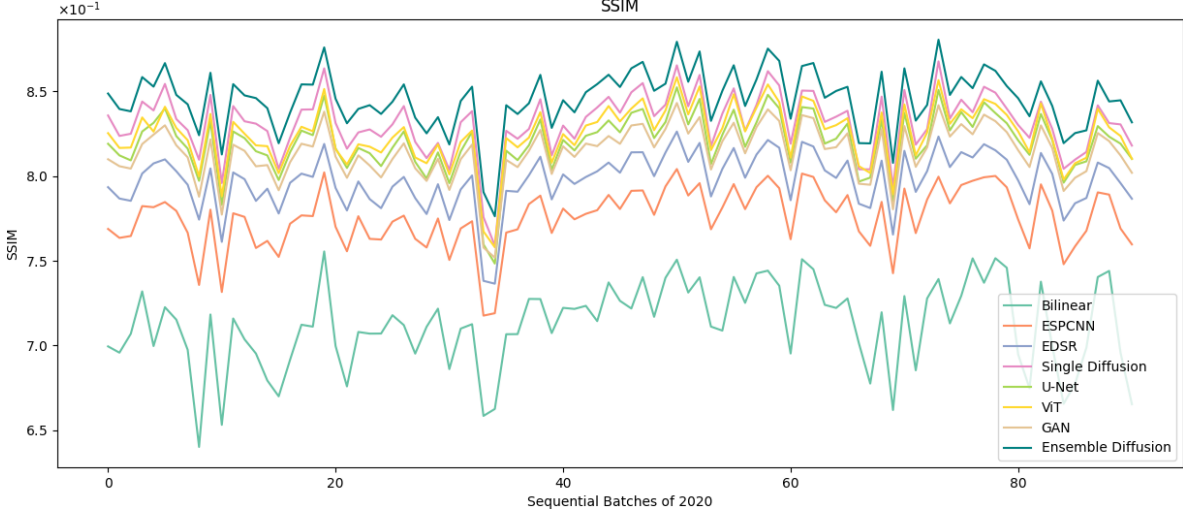


Figure 30: Graph comparison for the SSIM metric of the different tested models. SSIM is calculated for each batch sequentially in the testing year of 2020, highlighting the variability of performance at different times of the year.

The Structural Similarity Index (SSIM) is widely acknowledged as a crucial metric for evaluating super-resolution performance, and we are interested to explore the changes in its performance within the training year. Figure 30 presents a comparative analysis of SSIM metrics across the various downscaling methods for the year 2020. Notably, there is a significant fluctuation in performance throughout the year. These variations, consistently observed across all models including bilinear interpolation, suggest that temporal weather variations significantly influence the efficacy of the downscaling models. This consistent trend across different models implies that the performance changes are predominantly driven by weather-related factors rather than the intrinsic characteristics of the models themselves.

Furthermore, an in-depth geographical analysis of SSIM performance can provide insights into which areas are harder to effectively downscale. Given that SSIM is calculated over patches, it's feasible to compile a comprehensive SSIM output. This output takes the form of a matrix, where each pixel value denotes the SSIM computed over a window centered on that corresponding pixel in the input images. In Figure 31, we showcase mean images derived from averaging these full SSIM matrices for the entire testing years of 2009 and 2020, respectively. These images reveal that coastal regions typically exhibit higher error rates, with the most pronounced discrepancies observed in the high Adriatic Sea, the Strait of Messina, and the Ligurian Sea. The results also indicate that the geographical errors are remarkably similar across the two testing years, suggesting that there has been no significant change in the error distribution between these comparative periods.

As a final experiment, considering the unavailability of CERRA for the years ranging from 2021 to 2024, we also make publicly available for further studies our computed approximation of CERRA, computed via Ensemble Diffusion.

4.5 Discussion & Conclusion

In this paper, we successfully employed state of the art super-resolution diffusion models to downscale wind speed data from ERA5 to CERRA. This approach approximates the existing model used for

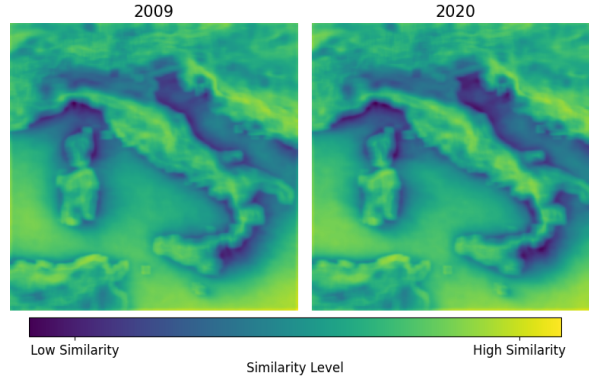


Figure 31: Mean image representing the spatial SSIM error for both testing years of 2009 and 2020, showcasing the downscaling performance relative to different geographical areas. The similarity in the results demonstrates that the errors made by the model are consistent with respect to the two years used for the test.

generating CERRA, but it innovatively does so in a data-driven manner without the need for supplementary data. Our experiment was focused on the Mediterranean area surrounding Italy and utilizing wind speed as our main weather variable.

Our experiments, conducted over two distinct testing years, demonstrated that the downscaling computed by our method aligns closely with the outcomes of traditional physics-based models. This alignment was evident both in the similarity of results and in their concordance with measurements from ground-based in-situ stations. Among all the models tested, the Ensemble Diffusion approach achieved the most accurate results. However, it's noteworthy that even less complex super-resolution models consistently outperformed basic methods like bilinear interpolation. Our findings reveal the practicality of using super-resolution diffusion models in providing timely and informative downscaled data for meteorological studies. Given the significant delays often encountered in downscaled reanalyses – for example, CERRA's data lagging over two years behind the current date at the time of this study – our findings are particularly relevant for the scientific community. These delays typically arise from the unavailability of additional information needed by physics-based models, which depend on various sources and are subject to time-consuming computational processes. Our approach, leveraging a neural model trained on previously generated data, offers a timely alternative for accessing current downscaled results. This method not only reduces computational costs but also maintains a high quality of output. Importantly, our model produces data that aligns more closely with actual measurements than the original low-resolution data, demonstrating its practical potential in enhancing real-time weather analysis.

In our study, we encountered differing time scales between the high-resolution dataset (CERRA) and the low-resolution dataset (ERA5). ERA5 offers hourly temporal resolution, whereas CERRA provides data at a 3-hour resolution. Given that our model can generate sequences conditioned only on ERA5 data, we can utilize our model to augment the temporal resolution of our downscaled data to match the temporal resolution of ERA, providing a higher temporal resolution when compared to the physics-inspired reanalyses model.

Regarding the main limitations of our approach, it's important to note that our data-driven method requires pre-existing downscaled data for training, thus relying on already established models. Additionally, our experiments represent a preliminary phase, focusing solely on a single weather variable and a relatively limited geographical area. Future developments could encompass a broader range of weather variables and potentially cover the entire region addressed by CERRA. This expanded scope could lead to improved results, as the model would gain a more comprehensive understanding of the weather system with access to diverse meteorological data. Moreover, hardware constraints limited our training dataset to only 10 years, which is noteworthy given the high quality of our results. In future research, it would be beneficial to utilize a more extensive dataset, potentially spanning the full 40 years available from CERRA, to enhance the model's accuracy and predictive capabilities.

In conclusion, our research presents a viable alternative for obtaining downscaled data of current events. We demonstrate the potential of diffusion models to effectively approximate the processes of

traditional downscaling models, avoiding delays in data availability. Furthermore, our findings suggest a promising future for reanalysis models, potentially shifting towards neural network-based approaches that leverage the advantages identified in this study. This shift could mark a significant advancement in the field, offering more efficient and accurate ways to handle meteorological data.

5 Paper II: Controlling ensemble variance in diffusion models, an application for reanalyses downscaling

Abstract

In recent years, diffusion models have emerged as powerful tools for generating ensemble members in meteorology. In this work, we demonstrate that a Denoising Diffusion Implicit Model (DDIM) can effectively control ensemble variance by varying the number of diffusion steps. Introducing a theoretical framework, we relate diffusion steps to the variance expressed by the reverse diffusion process. Focusing on reanalysis downscaling, we propose an ensemble diffusion model for the full ERA5-to-CERRA domain, generating variance-calibrated ensemble members for wind speed at full spatial and temporal resolution. Our method aligns global mean variance with a reference ensemble dataset and ensures spatial variance is distributed in accordance with observed meteorological variability. Additionally, we address the lack of ensemble information in the CARRA dataset, showcasing the utility of our approach for efficient, high-resolution ensemble generation.

5.1 Introduction

A critical aspect of generative diffusion is the statistical properties of the generated distribution, particularly the variance. In this second paper of the thesis, we focus on establishing a theoretical understanding of how the variance of diffusion ensembles relates to the number of reverse steps, and on ensuring that its spatial distribution aligns with physical expectations. Although previous works have already reported a reduction in variability when using fewer diffusion steps [SME22, ND21] to the best of our knowledge no prior study has systematically examined the relationship between ensemble variance and the number of diffusion steps.

Building on the previous paper reported in Section 4, we extend the application of diffusion-based downscaling for wind speed to the full European domain covered by CERRA, and explore the architectural adaptations required to scale diffusion models to large spatial domains.

In this paper we build upon the diffusion mathematical framework introduced in Section 3.7. Our analysis begins with a mathematical exploration of the reverse diffusion process, showing that the variance of ensemble members is directly linked to the choice of diffusion steps. We validate this relationship in the ERA5-to-CERRA task by comparing diffusion ensembles with the physically based CERRA-EDA dataset. Our experiments demonstrate that adjusting the number of diffusion steps provides effective control over the ensemble spread, allowing us to reproduce the magnitude and spatial structure of variance observed in CERRA-EDA. Finally, we propose a methodology for selecting the optimal number of diffusion steps in practical applications, and we illustrate its utility by applying it to the Arctic CARRA dataset, where no ensemble product is available.

In summary, the main contributions of this paper are: (i) extending diffusion-based downscaling from regional to continental domains, (ii) establishing a theoretical link between diffusion steps and ensemble variance, (iii) validating this link against the CERRA-EDA ensemble, and (iv) proposing a methodology to tune diffusion ensembles in domains lacking ensemble products, demonstrated on the Arctic CARRA reanalysis.

5.1.1 Related Works

A general overview of downscaling methods is provided in Section 2. Here, we focus specifically on recent developments in generative approaches for ensemble generation, with particular attention to diffusion-based methods.

In recent years an increasing number of publications have focused on using generative deep learning architectures for downscaling, taking advantage of the models' inherently probabilistic nature to generate ensemble members. In [HMC⁺22a], the authors extend generative adversarial networks (GANs) to downscale precipitation forecasts, addressing the challenges of adding fine-scale structure and correcting forecast errors. Using high-resolution radar data as ground truth, the study demonstrates that GANs and VAE-GANs can produce spatially coherent precipitation maps while maintaining statistical properties comparable to state-of-the-art post-processing methods. In [LCLG⁺24], the authors propose using deep generative diffusion models to emulate and enhance probabilistic weather forecasting. Learning from historical data, diffusion model generate realistic ensemble forecasts at a fraction of the

computational cost of traditional physics-based simulations. The generated ensembles replicate the statistical properties and predictive skill of operational forecasts, while also correcting biases, improving reliability, and better capturing probabilities of extreme weather events. This scalable approach has potential applications in both weather forecasting and large-scale climate risk assessment. In [LSR⁺22], the authors propose a statistical downscaling method for satellite sea surface wind (SSW) data using generative adversarial networks (GANs) with a dual learning scheme. The dual learning approach combines high-resolution SSW reconstruction with degradation kernel estimation in a closed loop. In [LGG24], the authors propose a cascading deep learning framework based on generative diffusion models for downscaling tropical cyclone (TC) wind fields from low-resolution ERA5 data (0.25°) to high-resolution observations (0.05°) at 6-hour intervals. The framework consists of a debiasing neural network to improve wind speed accuracy and a conditional denoising diffusion probabilistic model (DDPM) for super-resolution and ensemble generation. Applied to 85 TC events, the model achieves realistic spatial structures, enabling accurate high-resolution TC wind modeling and risk assessment. In [LGWZN⁺24a], the authors present a hybrid downscaling framework combining dynamical downscaling with generative diffusion models to enhance resolution and improve uncertainty estimates for regional climate projections. The method uses a regional climate model (RCM) to downscale Earth System Model (ESM) outputs to intermediate resolution, followed by a diffusion model for further refinement. Applied to the CMIP6 ensemble, this approach provides more accurate uncertainty bounds, lower errors than bias correction and spatial disaggregation (BCSD), and better captures spectra and multivariate correlations. The framework offers a cost-effective and accurate alternative for downscaling large climate projection ensembles.

5.1.2 Reversed DDIM formulation

In the standard formulation presented in Section 3.7.1, diffusion models are defined with x_0 as the clean data sample and x_T as pure Gaussian noise, with the forward process progressively corrupting $x_0 \rightarrow x_T$ and the reverse process reconstructing x_0 from x_T .

In this paper, we adopt an equivalent but reversed convention, in which the indexing is flipped: x_T denotes the clean sample drawn from the data distribution, while x_0 denotes pure noise. This choice does not alter the underlying theory, the resulting updates are algebraically equivalent to the standard DDIM formulation, but it simplifies subsequent derivations and notation.

Under the reversed convention, the DDIM forward process is written as a sequence $\{x_0, x_1, \dots, x_T\}$ such that x_T is clean data and x_0 is pure Gaussian noise. A noisy intermediate state x_t is expressed as

$$x_t = \sqrt{\alpha_t} x_T + \sqrt{1 - \alpha_t} \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \quad (53)$$

Here $\sqrt{\alpha_t}$ and $\sqrt{1 - \alpha_t}$ are the *signal rate* and *noise rate*, respectively, since they control the relative contribution of signal and noise at step t (see Section 5.2.1 for details). This is equivalent to the standard forward relation in Equation 26, with the roles of x_0 and x_T swapped.

Equation (53) describes how a clean image x_T is gradually corrupted into noise x_0 . To invert this process, a DDIM trains a neural network $\epsilon_\theta(x_t, t)$ to approximate the injected noise:

$$\epsilon_\theta(x_t, t) \approx \epsilon_t, \quad \forall t = T - 1, \dots, 0. \quad (54)$$

For compactness, define

$$s_t := \sqrt{\alpha_t}, \quad n_t := \sqrt{1 - \alpha_t}.$$

At step $t - 1$, we have

$$x_{t-1} = s_{t-1} x_T + n_{t-1} \epsilon_{t-1} \implies x_T = \frac{x_{t-1} - n_{t-1} \epsilon_{t-1}}{s_{t-1}}. \quad (55)$$

Substituting this into (53) at time t gives

$$\begin{aligned} x_t &= s_t \left(\frac{x_{t-1} - n_{t-1} \epsilon_{t-1}}{s_{t-1}} \right) + n_t \epsilon_t \\ &= \frac{s_t}{s_{t-1}} x_{t-1} + \left(n_t - \frac{s_t}{s_{t-1}} n_{t-1} \right) \epsilon_t + \frac{s_t}{s_{t-1}} n_{t-1} (\epsilon_t - \epsilon_{t-1}). \end{aligned} \quad (56)$$

In the deterministic DDIM setting ($\sigma_t = 0$), the path term vanishes by enforcing $\epsilon_t \equiv \epsilon_{t-1} \equiv \hat{\epsilon}$, which yields

$$x_t = \frac{s_t}{s_{t-1}} x_{t-1} + \left(n_t - \frac{s_t}{s_{t-1}} n_{t-1} \right) \hat{\epsilon}. \quad (57)$$

Re-expressing s_t, n_t in terms of α_t leads to the compact update rule

$$x_t = \sqrt{\frac{\alpha_t}{\alpha_{t-1}}} x_{t-1} + \left(\sqrt{1 - \alpha_t} - \sqrt{\frac{\alpha_t(1 - \alpha_{t-1})}{\alpha_{t-1}}} \right) \hat{\epsilon}, \quad \hat{\epsilon} \approx \epsilon_\theta(x_{t-1}, t - 1). \quad (58)$$

Starting from $x_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and iterating (58), the process converges to a synthetic sample $x_T \sim p_{gt}(x_T)$. This expression is algebraically equivalent to the standard DDIM update (Equation 29), only written under the reversed indexing.

A practical property of DDIM is that it allows *step skipping*. Setting a step-size $\Delta t \in \mathbb{N}$ generalizes (58) to

$$x_t = \sqrt{\frac{\alpha_t}{\alpha_{t-\Delta t}}} x_{t-\Delta t} + \left(\sqrt{1 - \alpha_t} - \sqrt{\frac{\alpha_t(1 - \alpha_{t-\Delta t})}{\alpha_{t-\Delta t}}} \right) \hat{\epsilon}, \quad \hat{\epsilon} \approx \epsilon_\theta(x_{t-\Delta t}, t - \Delta t). \quad (59)$$

The case $\Delta t = 1$ recovers the full generative trajectory, while $\Delta t > 1$ reduces the number of iterations $N = T/\Delta t$ and accelerates sampling, typically at the expense of output fidelity. This trade-off is particularly relevant for high-dimensional data or large denoisers, where the computational cost per step is substantial.

5.1.3 Notations

Throughout this article, we will make use of some notations, which we report here for completeness. In particular, we use lower-case non-bold latin characters to indicate scalar values, lower-case bold latin characters to indicate vectors, and upper-case bold latin characters to indicate matrices. Moreover, we will denote as $\mathbf{0}, \mathbf{1}$ the vector of all zeros and all ones, respectively, and as $\mathbf{I}$ the identity matrix. If x and y are two vectors with the same dimension, then $x \geq y$ means that $x_i \geq y_i$ for every i . The data is therefore assumed to be a vector of dimension n , which corresponds to the flattened version of the $(S \times M \times h \times w)$ -dimensional array if it is indicated with a lowercase bold letter, while it is treated as a tensor if indicated with an uppercase bold letter. For example, if $\mathbf{X} \in \mathbb{R}^{S \times M \times h \times w}$ represents the usual tensor-shaped dataset with S datapoints, M channels, and spatial dimension $h \times w$, then $x \in \mathbb{R}^n$ represents its flattened version, with $n = S \cdot M \cdot h \cdot w$.

5.1.4 Data

Our task involves using the global reanalyses ERA5 [HBB⁺20] as the low-resolution input to condition our model in predicting a high-resolution downscaled counterpart, trained on the regional reanalyses CERRA [RBLM⁺24]. The resulting ensemble members are validated against CERRA-EDA, an ensemble version of CERRA at half the temporal and spatial resolution. We will also apply our methods to CARRA, a dataset built on the same core model as CERRA and that covers the Arctic region, but lack ensemble members. The respective domains of CERRA and CARRA are reported in Figure 32.

5.2 Methodology

In this section we review our signal and noise scheduling methodology before delving into the mathematical reasoning behind the control of variance via the number of diffusion steps. In particular, we derive an explicit formula to predict the variance of the generated data, which shows how it depends on the number of diffusion steps.

5.2.1 Signal and noise schedules

As discussed in Section 3.7.3, the diffusion schedule $\{\alpha_t\}_{t=0}^T$ plays a central role in shaping the generative process. By controlling how quickly signal is attenuated and noise is introduced, the schedule directly determines the difficulty of the denoising task and the efficiency of the reverse process.

Several schedules have been proposed in the literature, including linear schedules [HJA20a], cosine-based schedules [ND21], and variants in log-SNR space [KSPH21]. Each family balances signal and

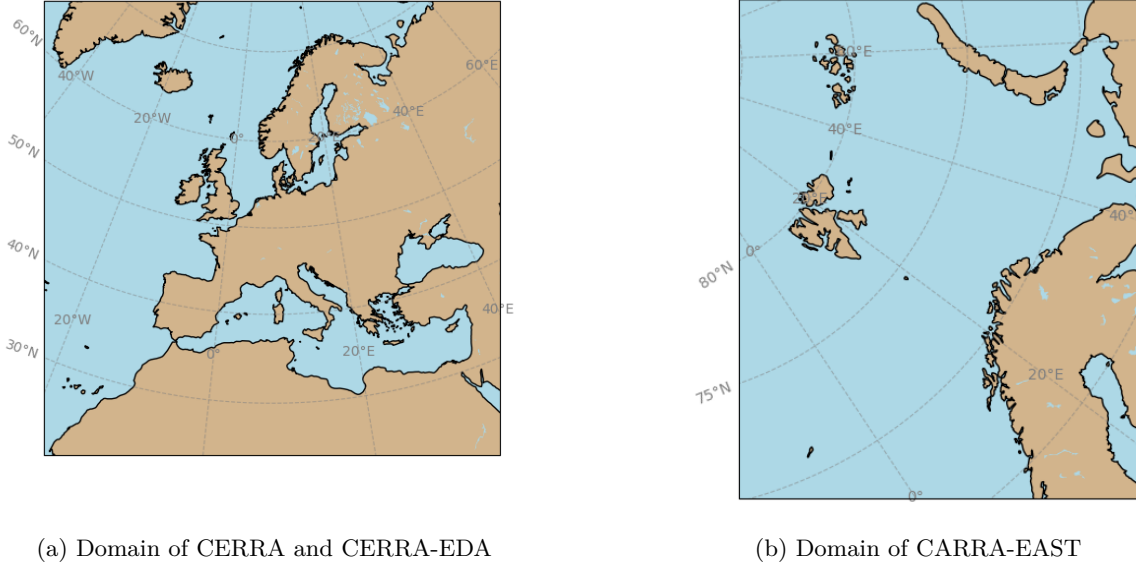


Figure 32: Comparison of CERRA and CARRA-EAST domains.

noise differently, with important implications for stability and sample quality. In this study, we adopt a sinusoidal schedule, following [AEMM23], which offers smooth transitions between signal and noise while preserving the variance-preserving (VP) property. This ensures that at every timestep the mixture of signal and noise maintains unit variance, a key requirement for stable training and effective reverse diffusion.

Concretely, the signal and noise rates are defined as

$$s_t = \sqrt{\alpha_t} = \sin\left(\pi \frac{T-t}{T}\right), \quad n_t = \sqrt{1 - \alpha_t} = \cos\left(\pi \frac{T-t}{T}\right). \quad (60)$$

It is immediate that $s_t^2 + n_t^2 = 1$, confirming the variance-preserving property. This sinusoidal interpolation avoids the abrupt decay of linear schedules while providing a controlled balance of signal and noise across timesteps, which is particularly beneficial for high-resolution generation.

As in other VP schedules, the extremes of the interval can be problematic: at $t = 0$ or $t = T$, the process would degenerate into pure noise or pure signal, causing unstable gradients or ineffective denoising. Following common practice [Che23, HHS23], we clamp the extremes of the schedule to maintain a finite signal-to-noise ratio (SNR), thereby preventing degenerate cases and ensuring stability in both training and sampling.

For the time discretization, we adopt a uniform subdivision of the interval $[0, T]$ into T steps, so that $t \in \{0, 1, \dots, T\}$. When employing step-skipping with $\Delta t > 1$ (see Equation 59), the time domain reduces to

$$t \in \{0, \Delta t, 2\Delta t, \dots, N\Delta t\}, \quad N = \frac{T}{\Delta t}, \quad (61)$$

which shortens the trajectory while retaining the same functional form of the schedule.

5.2.2 Controlling the variance

The generative process for DDIM involves a sequence of denoising operations alternated with the re-injection of the same noise. Our findings reveal that the key aspect of controlling the variance is selecting the number of diffusion steps. In this section, we analyze the evolution of the variance through the iterative diffusion process and we provide an intuition on how the number of diffusion steps N influences the variance of the generated data. We remark that the diffusion process in DDIM

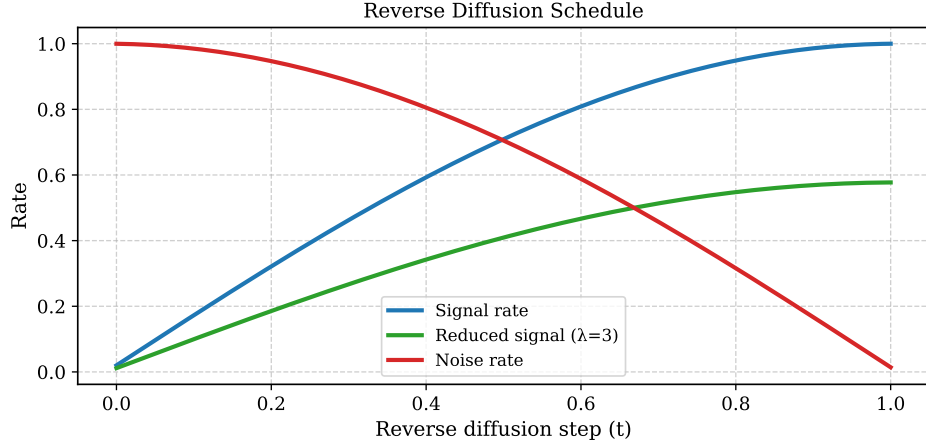


Figure 33: Reverse diffusion signal and noise ratios with cosine scheduling, including the reduced signal rate

is defined by sampling $x_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and then iterating through:

$$x_t = \sqrt{\frac{\alpha_t}{\alpha_{t-\Delta t}}} x_{t-\Delta t} + c_{t-\Delta t} \epsilon_{\Theta}^{t-\Delta t}(x_{t-\Delta t}), \quad (62)$$

where we defined $c_{t-\Delta t}$ as:

$$c_{t-\Delta t} := \sqrt{1 - \alpha_t} - \sqrt{\frac{\alpha_t(1 - \alpha_{t-\Delta t})}{\alpha_{t-\Delta t}}}, \quad (63)$$

to simplify the notation. We also recall that the diffusion schedule $\{\alpha_t\}_{t=0,\dots,T}$ is selected such that $\alpha_0 \approx 1$ and $\alpha_T \approx 0$. Note that $\alpha_T \neq 0$ as we clamped its value to avoid instabilities during training, as we already discussed in 5.2.1.

Let x_0 be an n -dimensional random variable with Gaussian distribution such that $\mathbb{E}[x_0] = \mathbf{0}$ and $\text{Var}(x_0) = \mathbf{I}$, and let $\{x_t\}_{t=0,\dots,T}$ be the stochastic process defined in (62). Since the full variance matrix $\text{Var}(x_t)$ has dimension $n \times n$, which is prohibitive from a computational point of view, in this work we limit our analysis to the element-wise variance, defined as:

$$\mathbf{v}(x_t)_i := \text{Var}(x_t)_{i,i}, \quad (64)$$

which is a vector of dimension n , representing the variance between each pixel of x_t with itself, ignoring the relationship with the adjacent pixels. Note that this choice is coherent with our task, as we want to study the pixel-wise uncertainty of DDIM downscaling, and taking into consideration the correlation between adjacent pixels is beyond the scope of this analysis.

It is not difficult to show that the element-wise variance has similar properties to the classical variance, and in particular:

$$\text{(P1)} \quad \mathbf{v}(ax_t) = a^2 \mathbf{v}(x_t),$$

$$\text{(P2)} \quad \mathbf{v}(x_t + y_t) = \mathbf{v}(x_t) + \mathbf{v}(y_t) + 2\mathbf{v}(x_t, y_t),$$

where $\mathbf{v}(x_t, y_t)$ is the element-wise covariance between x_t and y_t .

Proposition 5.1. *Let $\mathbf{m}_t := \mathbb{E}[x_t]$ and $\mathbf{v}_t := \mathbf{v}(x_t)$. Then, for any $t = \Delta t, \dots, N\Delta t$, it holds:*

$$\mathbf{v}_t \approx \mathbf{F}_t \mathbf{v}_{t-\Delta t} + \mathbf{g}_t, \quad (65)$$

where:

$$\begin{aligned}\mathbf{F}_t &:= \frac{\alpha_t}{\alpha_{t-\Delta t}} \mathbf{I} + \mathbf{J}_t, \\ \mathbf{J}_t &:= 2\sqrt{\frac{\alpha_t}{\alpha_{t-\Delta t}}} c_{t-\Delta t} \text{diag} \left(\nabla_{\mathbf{m}_{t-\Delta t}} \epsilon_{\Theta}^{t-\Delta t}(\mathbf{m}_{t-\Delta t}) \right), \\ \mathbf{g}_t &:= c_{t-\Delta t}^2 \mathbf{1}.\end{aligned}$$

Note that (65) represents a first-order linear recurrence equation of dimension n with non-constant coefficients, for which there exists a closed formula that can be used to compute $\mathbf{v}_T$, the variance of the generated data, as summarized in the following Theorem.

Theorem 5.2. *Let $\mathbf{F}_t$, $\mathbf{J}_t$, and $\mathbf{g}_t$ defined as in Proposition 5.1, and let Δt be a fixed step-size for DDIM, so that any timestep t can be written as $i\Delta t$ for $i = 0, \dots, N$. Note that $\mathbf{v}_0 = \mathbf{1}$, the vector of all ones, as $x_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Then, defining $\mathbf{F}_{(N+1)\Delta t}$ to be the identity matrix to simplify the notation, it holds:*

$$\mathbf{v}_T \approx \left(\prod_{i=1}^N \mathbf{F}_{i\Delta t} \right) \mathbf{1} + \sum_{i=1}^N \left[\left(\prod_{k=i+1}^{N+1} \mathbf{F}_{k\Delta t} \right) \mathbf{g}_{i\Delta t} \right]. \quad (66)$$

Notably, 5.2 not only allows for the explicit computation of the element-wise variance of $\mathbf{x}_T$, but it also underlines the dependence of $\mathbf{v}_T$ on the number of diffusion timesteps. Indeed, if, for example, both $\mathbf{F}_{i\Delta t} \geq \mathbf{0}$ and $\mathbf{g}_{i\Delta t} \geq \mathbf{0}$, then 66 clearly implies that $\mathbf{v}_T$ monotonically grows as a function of N . Specifically, increasing the step size Δt (and consequently reducing the number of timesteps N) results in fewer terms being included in the summation that computes $\mathbf{v}_T$, thereby decreasing its value. Refer to [MEL25] for extended proofs of the aforementioned theorem and proposition.

5.3 Experimental Setting

In this section, we describe the pre-processing, training, and evaluation strategies required to adapt the diffusion model for high-resolution downscaling across the full CERRA domain. The transition to larger spatial scales necessitated adjustments to both data handling and model architecture to ensure consistent performance. We focus on data selection and pre-processing in Section 5.3.1, followed by a detailed explanation of the modifications required for training on the full domain in Section 5.3.2.

5.3.1 Data selection and pre-processing

The diffusion model performs downscaling in an image-to-image manner, transforming low-resolution ERA5 data into high-resolution CERRA or CARRA outputs while maintaining alignment over the same spatial domain for input and output. The primary experimental framework of our study focuses on downscaling wind speed from ERA5 to CERRA over the full CERRA domain and comparing the ensemble results with the CERRA-EDA dataset. Our model will be trained solely on the CERRA data and no input of CERRA-EDA is used during training, ensuring that the model will learn the ensemble variability only from the CERRA data itself.

The CERRA domain spans a grid of 1069×1069 points across Europe, represented in a Lambert Conformal Conic (LCC) projection. The complete CERRA dataset covers the period from September 1984 to June 2021; for our experiments, we utilized data from 1985 to 2010 for training and from 2016 for ensemble testing. Additionally, we evaluated overall performance for the entire decade from 2011 to 2020, though computational constraints limited ensemble variance testing to the year 2016. In our diffusion model, ERA5 data serves as the conditioning input, re-gridded to align with CERRA's LCC grid. Wind speed, s , is derived from the two wind components, s_u and s_v , as:

$$s := \left\| \begin{bmatrix} s_u \\ s_v \end{bmatrix} \right\|_2 = \sqrt{s_u^2 + s_v^2}$$

Our diffusion model is based on a residual convolutional denoiser, which benefits from input dimensions divisible by two at multiple levels. To ensure consistency between the downsample and upsample paths, we augment the original grid size (1069×1069) with mirroring padding, resulting in

a final input size of 1072×1072 . Our conditioning is implemented via concatenation on the input of the denoising network, to equalize the dimension we opt to upscale the ERA5 data to 1069×1069 via bilinear interpolation, and then we apply the same mirroring padding to ensure spatial consistency.

To compare our diffusion-based ensembles with the existing CERRA-EDA ensemble members, we note that CERRA-EDA operates at half the spatial resolution of CERRA (565×565) and a temporal resolution of six hours, also halved relative to CERRA. For consistency, we train our model exclusively on full-resolution CERRA data with a three-hour temporal resolution. Wind speed is an instantaneous value, therefore six-hourly outputs can be generated using conditioning information at the same temporal resolution. The generated outputs retain the full CERRA resolution; for comparison with CERRA-EDA, we apply bilinear interpolation to downscale the results.

Diffusion models require standardized inputs to balance signal and noise rates effectively. We apply a standardizer that normalizes inputs using the mean and variance computed across the training set, ensuring consistent scaling across batches. It is important to note that this variance is relative to the training data and is distinct from the ensemble variance between members produced by the diffusion model. This normalization is applied to both high-resolution images (signal rates) and low-resolution conditioning data, aligning their scales and simplifying the learning task. Experimentally, this approach demonstrated improved performance and stability.

5.3.2 Training on the full domain

The general structure of our diffusion model follows the framework established in [MAC24b]. The denoiser is a residual U-Net [RFB15b, HZRS16b] implemented in TensorFlow/Keras, designed as a symmetric encoder-decoder network with skip connections and residual convolutional blocks using Swish activations and layer normalization. The architecture performs three stages of downsampling through average pooling and three corresponding upsampling stages via bilinear interpolation. Each stage (both in the encoder, decoder, and latent bottleneck) contains three residual blocks, each composed of two 3×3 convolutions and a skip connection to preserve spatial detail. The model contains approximately 22 million learnable parameters, predicting the noise component at each diffusion step.

Training is conducted by uniformly sampling signal-to-noise ratios from the sinusoidal noise scheduling functions, optimizing the mean absolute error (MAE) between the sampled noise and the predicted noise. Scaling the diffusion model to the full CERRA domain, with image sizes of 1069×1069 , required modifications to a model originally designed for downscaling images up to 256×256 . These adaptations involved adjustments to both the noise schedule and the network architecture to ensure effective performance at higher resolutions.

Recent findings [Che23, HHS23] have demonstrated that cosine noise schedules can reduce the effective noise added by the model as image size increases. This occurs because redundancy in data, such as correlations among nearby pixels, typically increases with resolution, and the independently added noise becomes easier to denoise in larger images. Consequently, noise schedules optimized for smaller resolutions may not perform as effectively at higher resolutions. To address this issue, we introduce a constant scaling factor $\lambda > 1$ to reduce the signal rate in the noise schedule. For an input $\mathbf{x}_T \sim p_{gt}(\mathbf{x}_T)$ and a scaling factor λ , the noise schedule 53 is modified as follows:

$$\mathbf{x}_t = \sqrt{\alpha_t} \frac{\mathbf{x}_T}{\lambda} + \sqrt{1 - \alpha_t} \epsilon_t \quad (67)$$

Experimentally, setting the value $\lambda = 3$ improved training efficiency and overall model performance, which is a crucial aspect for large-scale ensemble diffusion applications. A comparison of signal and noise ratios with the scaled signal rate is shown in 33. Notably, applying the same scaling to the low-resolution conditioning signal further stabilized the training process.

Experimentally, we tested several values of $\lambda \in [1, 4]$ and observed that $\lambda = 3$ provided the best trade-off between reconstruction fidelity and training stability, achieving the lowest MSE with respect to the deterministic CERRA ground truth. Full ablation at multiple λ values and full resolution would be computationally prohibitive; therefore, smaller-scale experiments were used to guide the choice, keeping the spatial resolution constant as it is the key factor influencing this effect. The selected value should thus be regarded as empirically determined. This adjustment proved crucial for large-scale ensemble diffusion applications, improving convergence and overall stability. Notably, applying the same scaling to the low-resolution conditioning signal further stabilized the training process.

While the core U-Net architecture remained largely unchanged, scaling to larger domains required increasing the number of residual blocks in the bottleneck. This adjustment improved performance when generating larger images [HHS23]. The model’s performance across the increased domain remained comparable to that observed at smaller resolutions, with the bottleneck modification enhancing its ability to process higher-resolution inputs effectively.

The diffusion model is trained on the ERA5 to CERRA task for wind speed on the full domain, utilizing data from 1985 to 2010 for training. The training was performed on a single compute node equipped with three NVIDIA A100 GPUs (64 GB VRAM each), using a *batch separation (mirroring) strategy* to distribute the workload across devices. The model was trained for 400 epochs with 500 steps per epoch, corresponding to a total training time of approximately 42 hours. Each epoch required about 300 seconds to complete, and the batch size was set to three, with one sample processed per GPU. *AdamW* was used as the optimizer, with an initial training phase of 200 epochs using a learning rate of 1×10^{-4} and a weight decay of 1×10^{-5} , followed by a fine-tuning phase of 100 epochs with a reduced learning rate of 1×10^{-5} and a weight decay of 1×10^{-6} . The model was trained to minimize the mean absolute error (MAE) between the predicted and true noise. During inference, the memory constraints are relaxed, and the model can be executed on GPUs with as little as 16 GB of VRAM.

5.4 Experiments

Our primary experimental objective is to evaluate the behavior of ensemble variance while changing the timestep Δt . While a comprehensive analysis of the ensemble diffusion model’s performance across the entire CERRA domain is beyond the scope of this paper, we ensure that the model operates within a realm of satisfactory performance. To achieve this, we begin—similar to [MAC24b]—by comparing our model’s performance against bilinearly interpolated ERA5 data and a residual U-Net model trained for the same task. The results, obtained by computing the Mean Squared Error (MSE) and SSIM between the high-resolution output produced by the model and the original CERRA, are shown in Table 14.

	Bilinear	U-Net	DDIM (Our)
MSE ($\downarrow$)	3.73e-03	2.50e-04	2.54e-04
SSIM ($\uparrow$)	0.751	0.892	0.923

Table 14: Performance of ensemble diffusion against bilinear interpolation and U-Net for the years 2011–2020.

Testing for the years 2011–2020, the ensemble diffusion model achieves an MSE of $2.54 \cdot 10^{-4}$, compared to $2.50 \cdot 10^{-4}$ for the U-Net and $3.73 \cdot 10^{-3}$ for bilinear interpolation. For SSIM, bilinear interpolation scores 0.751, the U-Net scores 0.892, and ensemble diffusion achieves the highest score of 0.923. Ensemble diffusion was executed with $N = 2$ diffusion steps and an ensemble of 10 members.

These results confirm that the diffusion model delivers satisfactory performance, with MSE values approximately an order of magnitude better than bilinear interpolation and comparable to those of the U-Net. Additionally, as expected, the diffusion model outperforms both alternatives in terms of SSIM. The proposed metrics were calculated on normalized data.

5.4.1 Quantifying variance

The first aspect of variance analysis we evaluate in our experiments is understanding how the model’s expressed variance changes with the number of diffusion steps, as detailed in Section 5.2.2. For this experiment we selected the year 2016 and computed the ensemble variance for 10 members for all the year at 6h intervals, matching the existing format of the CERRA-EDA ensemble. Furthermore, we used bilinear interpolation to reduce the spatial dimension of our output image from 1069×1069 to 565×565 , also matching the existing CERRA-EDA format. Having obtained a comparable dataset, we proceeded to calculate the pixel-wise variance for the ensemble member of each ensemble of the year, finally calculating the mean variance, a value representing the overall variance expressed by the diffusion model.

Given a dataset of ensembles $\mathbf{D} \in \mathbb{R}^{S \times M \times h \times w}$, where S represents the number of samples for 2016, $M = 10$ is the number of ensemble members for each datapoint, and $(h, w) = (565, 565)$ are the spatial

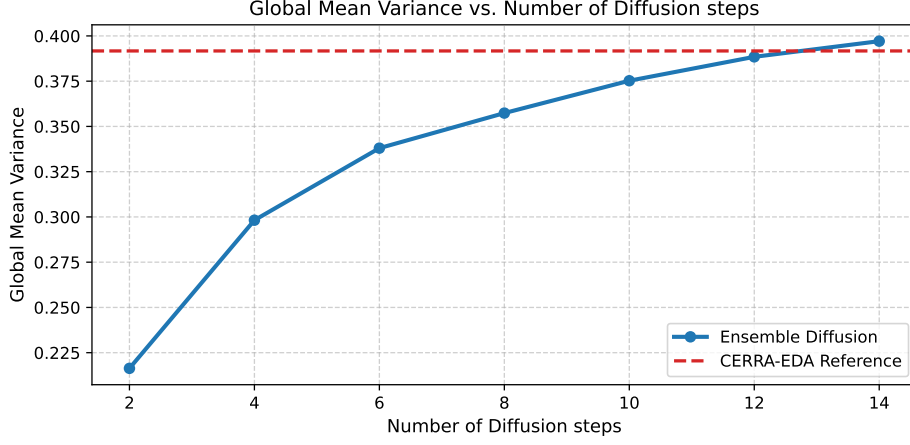


Figure 34: Plotting the global mean variance μ_V of ensemble diffusion for different number of steps against CERRA-EDA

dimensions, we compute the pixel-wise variance across the ensemble members $\mathbf{V} \in \mathbb{R}^{S \times M \times 565 \times 565}$, such that for each sample $i = 1, \dots, S$, and any spatial coordinate $(x, y) \in \{1, \dots, h\} \times \{1, \dots, w\}$,

$$\mathbf{V}_i(x, y) = \frac{1}{M} \sum_{j=1}^M \left(\mathbf{D}_{i,j}(x, y) - \frac{1}{M} \sum_{k=1}^M \mathbf{D}_{i,k}(x, y) \right)^2. \quad (68)$$

We then compute the mean variance across all samples and spatial locations as:

$$\mu_V = \frac{1}{S \cdot h \cdot w} \sum_{i=1}^S \sum_{x=1}^h \sum_{y=1}^w \mathbf{V}_i(x, y). \quad (69)$$

Here, μ_V is a scalar value representing the global mean variance across all samples and pixels. By computing μ_V for different diffusion steps, we observe that the global variance increases with the number of steps, as illustrated in Figure 34. Furthermore, we demonstrate that the variance control mechanism successfully aligns the global variance with that of CERRA-EDA, with the closest match achieved at $N = 12$ diffusion steps.

5.4.2 Analyzing variance spatially

The global mean-variance informs us that the diffusion model is able to match the overall amount of variance present in CERRA-EDA, but this gives us no information to where the variance is distributed in space. Our interest here is to examine if there is a match between the two spatial distributions of the variance.

For the following experiments, we decided to investigate the variance in a three-monthly setting, to highlight the behavior of the model in different seasons. To achieve this goal we divided the samples into four sets, including respectively January February and March (JFM), April May and June (AMJ), July August and September (JAS) and October November and December (OND).

To analyze spatial variance, we compute the pixel-wise mean of variance maps across all samples within each three-monthly subset. Let $k \in \{JFM, AMJ, JAS, OND\}$ denote the season, $S_k \subseteq \{1, \dots, S\}$ represent the subset of indices corresponding to the season k , and $|S_k|$ the number of samples corresponding to season k . The spatial mean variance $\mu_k^{spatial}$ for season k is given by:

$$\mu_k^{spatial}(x, y) = \frac{1}{|S_k|} \sum_{i \in S_k} \mathbf{V}_i(x, y), \quad (x, y) \in \{1, \dots, h\} \times \{1, \dots, w\}. \quad (70)$$

Here, $\mathbf{V}_i$ and $\mu_k^{spatial}$ are understood to represent spatial fields, where the value of each pixel is computed independently. We compute the spatial mean-variance for the reference ensemble CERRA-EDA and for Ensemble Diffusion across a range of diffusion steps $N \in \{2, 4, 8, 10, 12, 14\}$ for the year

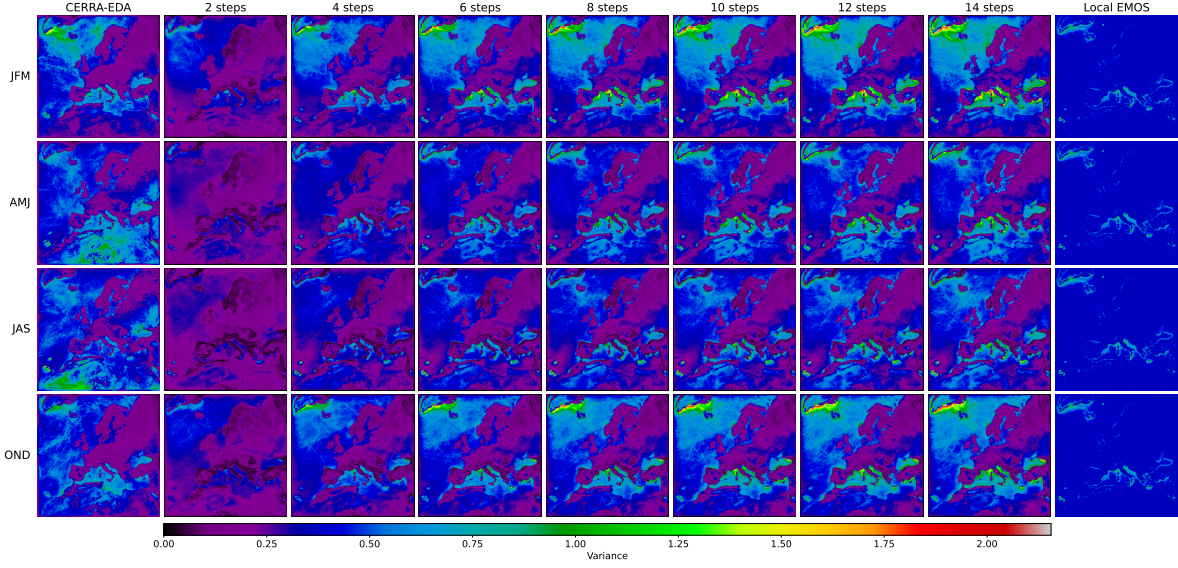


Figure 35: Comparing the spatial variance of CERRA-EDA with ensemble diffusion at different diffusion steps. The comparison relative to the testing year 2016 and is performed in a 3 monthly manner to highlight seasonal variations.

2016. This allows us to visualize how the spatial distribution of variance evolves as the number of diffusion steps increases. The obtained results are reported in Figure 35. We observe substantial overlap between the variance distributions of CERRA-EDA and Ensemble Diffusion. This indicates that the diffusion model successfully captures the variability of the data by training solely on the CERRA dataset, without requiring additional information about the underlying uncertainty. Furthermore, we note that the degree of spatial overlap varies by season, with the highest overlap observed in the JFM and OND sets, while AMJ and JAS exhibit less overlap.

To numerically quantify the overlap between variance distributions, we introduce the mean-variance discrepancy (MVD) score, a metric defined as the absolute error between the spatial mean-variance maps for each season. Let $\mu_{k,ED}^{\text{spatial}}$ denote the spatial mean-variance map of Ensemble Diffusion and $\mu_{k,CE}^{\text{spatial}}$ denote the corresponding map for CERRA-EDA. The MVD score for season k is then defined as:

$$\text{MVD}_k = \frac{1}{h \cdot w} \sum_{x=1}^h \sum_{y=1}^w \left| \mu_{k,ED}^{\text{spatial}}(x, y) - \mu_{k,CE}^{\text{spatial}}(x, y) \right|. \quad (71)$$

The numerical results for the MVD scores computed across all three-monthly periods are reported in Table 15.

From the MVD results, we observe that the optimal yearly number of diffusion steps is $N = 8$, a lower value if compared with 12, which was identified based on the global mean variance. This finding reveals that, while 12 steps yield an overall variance closest to the reference, the spatial distribution of the variance behaves differently. This underscores the importance of a tailored tuning procedure to achieve better alignment with expected uncertainties. Moreover, the MVD scores are not uniform throughout the year: the winter months achieve the lowest MVD scores with 4 diffusion steps, while the summer months perform best with 12 diffusion steps, possibly due to the model being affected by seasonal dynamics.

Alongside the full-domain analysis, we examine the behavior of the generated ensemble at a randomly selected point within the domain to provide a clearer and more readable visualization of the mean and variance ranges. For this analysis, we plot the mean value and variance range at the chosen point for both the 8-step ensemble diffusion and CERRA-EDA. To improve clarity over the high-frequency time series, we apply a moving average smoothing with a 30-sample window. The resulting plot, shown in Figure 37, reveals consistent overlap between the mean and variance ranges.

MVD Scores for Seasonal Spatial Variance Maps							
	2 steps	4 steps	6 steps	8 steps	10 steps	12 steps	14 steps
yearly	1.18e-01	1.45e-01	1.41e+01	<u>1.40e-01</u>	1.43e-01	1.45e-01	1.49e-01
JFM	1.14e-01	<u>1.02e-01</u>	1.04e+01	1.10e-01	1.19e-01	1.27e-01	1.32e-01
AMJ	2.15e-01	1.62e-01	1.42e+01	1.37e-01	1.34e-01	<u>1.33e-01</u>	1.34e-01
JAS	2.22e-01	1.74e-01	1.58e+01	1.50e-01	1.47e-01	<u>1.46e-01</u>	1.47e-01
OND	1.42e-01	<u>1.10e-01</u>	1.11e+01	1.17e-01	1.24e-01	1.29e-01	1.35e-01

Table 15: Mean variance discrepancy (MVD) scores computed between the spatial mean variance maps of Ensemble Diffusion and CERRA-EDA for each three-monthly period (JFM, AMJ, JAS, OND) and each timestamp for 2016. Lower scores indicate higher similarity between the variance distributions of the two ensembles.

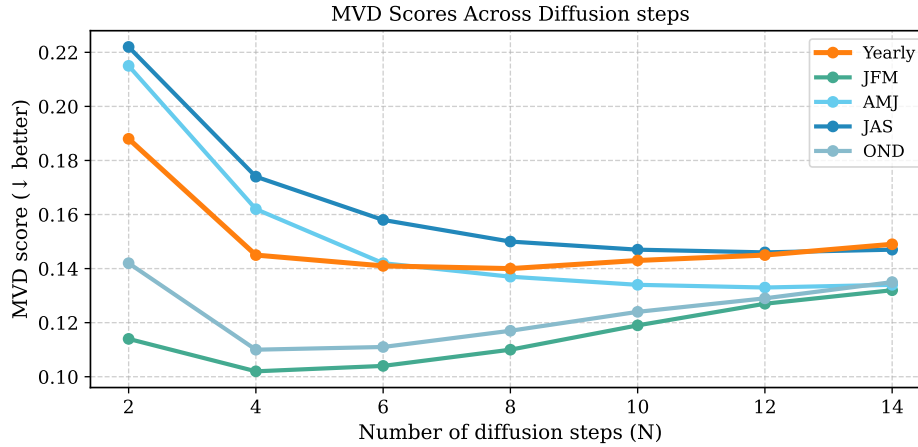


Figure 36: Evolution of the mean variance discrepancy (MVD) scores with the number of diffusion steps. The plot visualizes the values reported in Table 15 for the yearly and three-monthly periods (JFM, AMJ, JAS, OND) during 2016. Lower values indicate a closer match between the spatial variance distributions of Ensemble Diffusion and CERRA-EDA. The yearly curve (orange) highlights the global trend, while the seasonal curves illustrate the differing optimal step counts across the annual cycle.

5.4.3 Impact of Δt on performance

An interesting aspect of the variance analyses is to check how the overall performance of the Ensemble Diffusion (ED) model changes as the number of steps increase. To achieve this goal we have utilized $M = 10$ ensemble members at 565×565 spatial resolution and 6-hourly time resolution for the year 2016. We have computed the ensemble diffusion mean and compared it with the available CERRA-EDA control run, which is, as mentioned in Section 2.1.2, the member of the ensemble with no perturbation on the model input. We evaluate the comparison via MSE, calculated on normalized values. The results, reported in table 16, reveal that the performance is largely stable while changing the step size Δt , with a slight increase in the performance for larger step sizes. This should be expected as more steps allows the reverse diffusion process to make smaller, more gradual corrections at each step, leading to a more precise reconstruction of the target high-resolution image.

5.4.4 Demonstrating capabilities on CARRA

Both CERRA and CARRA, introduced in Section 6.1.1 are built upon the HARMONIE-AROME weather prediction system [BAA⁺17], adapted for reanalysis purposes. This common foundation ensures that both systems share a core methodology for atmospheric modeling and data assimilation, making them methodologically similar despite their focus on different regions.

MSE ensemble mean performance comparison							
	2 steps	4 steps	6 steps	8 steps	10 steps	12 steps	14 steps
yearly	1.67e-03	1.62e-03	1.60e-03	1.59e-03	1.58e-03	1.58e-03	<u>1.57e-03</u>
JFM	1.98e-03	1.92e-03	1.90e-03	1.88e-03	1.87e-03	1.86e-03	<u>1.86e-03</u>
AMJ	1.52e-03	1.48e-03	1.47e-03	1.46e-03	1.45e-03	1.45e-03	<u>1.44e-03</u>
JAS	1.39e-03	1.36e-03	1.34e-03	1.33e-03	1.33e-03	1.33e-03	<u>1.32e-03</u>
OND	1.79e-03	1.73e-03	1.70e-03	1.68e-03	1.68e-03	<u>1.67e-03</u>	<u>1.67e-03</u>

Table 16: Table comparing the MSE performance of the diffusion model at different numbers of reverse diffusion steps. The MSE is calculated between the mean of the 10 generated ensembles and the CERRA-EDA control run. The results are reported 3-monthly for the year 2016, along with the yearly mean across all seasons.

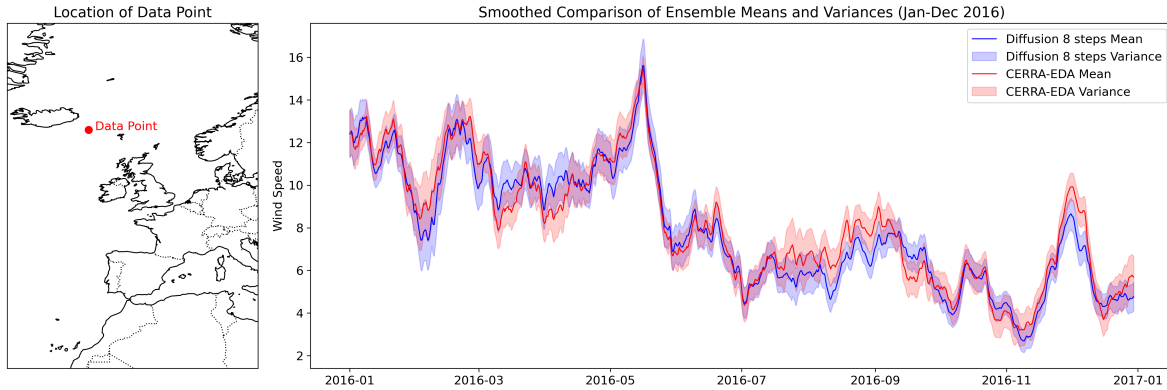


Figure 37: Mean and variance range at a randomly selected point in the domain, compared between ensemble diffusion (8 reverse diffusion steps) and CERRA-EDA. Data is smoothed using a 30 samples moving average window for readability and covers the entire year of 2016.

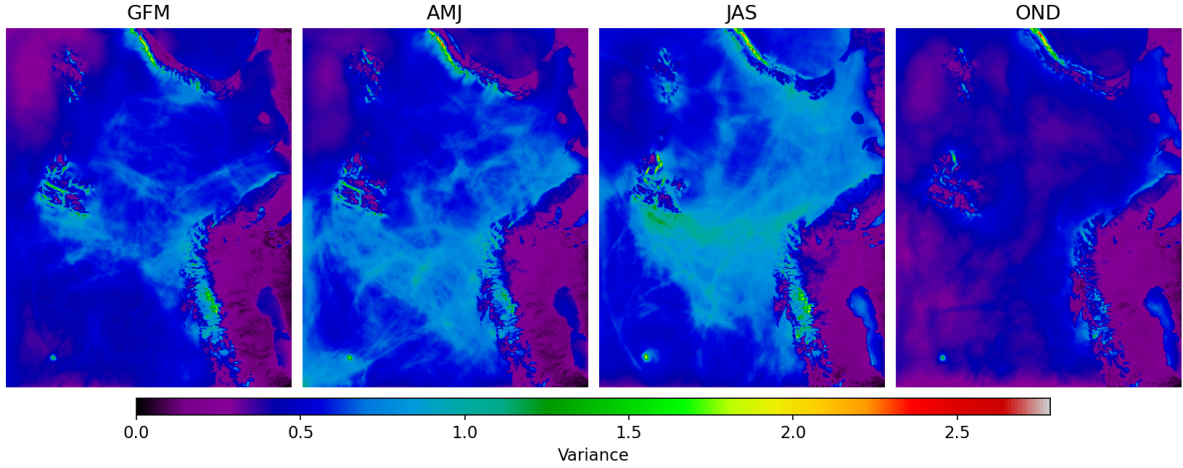


Figure 38: Revealing the spatial variance of ensemble diffusion trained on CARRA-EAST with 8 diffusion steps. The comparison is relative to the testing year 2016 and is performed in a 3 monthly manner to highlight seasonal variations.

Having validated our diffusion model’s capability to generate ensemble members for the CERRA dataset, we naturally extend this methodology to CARRA-EAST, the Arctic region dataset focusing on the eastern domain. Unlike CERRA, CARRA lacks an ensemble dataset, which limits its usability in quantifying variability, validating against real-world uncertainties, detecting rare or extreme events, performing ensemble-based statistical analyses, and estimating model robustness. Our ensemble diffusion driven ensemble dataset can be a highly efficient way to address this issues.

For CARRA-EAST, we adopt a training setup similar to the one proposed for CERRA. The downscaling process is framed as an image-to-image transformation, converting low-resolution ERA5 data into high-resolution CARRA-EAST, maintaining the same spatial domain between the two, with the resulting grid having a size of 989×789 . We regrid ERA5 wind speed data to match the projection of CARRA-EAST. The training period spans 1991–2010, while the year 2016 is reserved for ensemble testing. As with CERRA, padding is applied to adjust the grid size for compatibility with the diffusion architecture. Training was conducted following the setup detailed in Section 5.3.2.

The ensemble members are generated for the year 2016 using 8 diffusion steps, the optimal value determined through spatial validation with MVD on CERRA-EDA. Figure 38 illustrates the spatial variance for 2016, plotted within the same variance range as Figure 35. Our results reveal that the spatial variance is consistent with expected patterns.

5.4.5 Comparison with EMOS

Our method provides a direct ensemble comparison with CERRA-EDA, the numerically computed ensemble of CERRA, which can be naturally regarded as the optimal ground truth for evaluation. To ensure a comprehensive evaluation, our method can be positioned alongside other ensemble-generation approaches, including well-established statistical post-processing techniques such as EMOS. Among neural stochastic architectures, methods based on GANs have been proposed for related tasks, yet their training and tuning requirements make them impractical in our setting, as discussed in Section 5.1.1. For this reason, our comparison focuses on EMOS.

EMOS (Ensemble Model Output Statistics) is a statistical post-processing approach widely used to calibrate ensemble and deterministic forecasts [GRWG05]. Originally developed as a calibration framework for correcting bias and spread in ensemble systems, EMOS has become one of the most adopted post-processing techniques in numerical weather prediction and downscaling contexts [TG10, BL16, SGB23]. Beyond calibration, EMOS can also be applied to generate probabilistic ensembles directly from deterministic forecasts by modeling the predictive distribution of the target variable as a function of the forecast itself. This formulation allows EMOS to correct systematic biases and rescale forecast variability, producing calibrated ensemble predictions that are statistically consistent with the observed data.

In our setting, EMOS is applied to the deterministic outputs of the reference U-Net to generate prob-

abilistic ensembles. We employ a deterministic version of our downscaling model, a high-performing residual U-Net, which achieves performance comparable to the diffusion model in terms of MSE for the testing years, as reported in Table 14. This deterministic U-Net provides an ideal baseline for comparison, and our experiment focuses on generating ensemble members from the U-Net using EMOS and evaluating them against those obtained from the diffusion-based ensembles.

The objective of this experiment is to evaluate whether EMOS, when fitted to the relationship between the deterministic forecasts and the ground truth, can reproduce the ensemble variability observed in CERRA-EDA. In practice, EMOS models the conditional distribution of wind speed as the truncated normal distribution, $Y \sim \mathcal{TN}_0(\mu = a + bX, \sigma)$, where the coefficients a and b correct systematic bias and amplitude scaling, and σ controls the ensemble spread. The deterministic U-Net, trained on the 1985–2010 period (consistent with the diffusion model), was used as the forecast source. The year 2015 was used to fit the EMOS parameters against the CERRA ground truth, while 2016 was reserved for evaluation against the ten-member CERRA-EDA ensemble. All datasets were normalized to the $[0, 1]$ range using the CERRA training statistics. EMOS fitting was performed after bilinearly downscaling the raw U-Net outputs to 565×565 pixels to match the spatial resolution of CERRA-EDA. The evaluation was carried out at a 6-hourly frequency and 565×565 spatial resolution, comparing the EMOS-generated ensemble with the reference ensemble dataset.

Total Ensemble Variance Comparison	
2 steps	0.216
4 steps	0.298
6 steps	0.338
8 steps	0.357
10 steps	0.375
12 steps	0.388
14 steps	0.397
EMOS	0.392
Local EMOS	0.392
CERRA-EDA (Ref.)	0.392

Table 17: Total ensemble variance (μ_V) for Ensemble Diffusion at increasing numbers of reverse diffusion steps, compared with EMOS, Local EMOS, and the reference CERRA-EDA variance. Variance increases monotonically with diffusion steps and converges toward the reference.

Two EMOS configurations were tested. In the standard EMOS, a single set of parameters (a, b, σ) was fitted over the entire 2015 dataset, yielding $a = -7.1 \times 10^{-4}$, $b = 1.015$, and $\sigma = 0.0168$. The spread was subsequently calibrated to match the global mean variance of CERRA-EDA via $\sigma_{\text{new}} = \sigma \sqrt{\mu V_{\text{target}} / \mu V_{\text{current}}}$, resulting in $\sigma_{\text{new}} = 0.0122$. In the local EMOS, a separate regression was fitted for each grid point, $Y_t(i, j) = a_{i,j} + b_{i,j} X_t(i, j) + \varepsilon_t(i, j)$, producing spatial maps of $a_{i,j}$, $b_{i,j}$, and $\sigma_{i,j}$. To prevent over-dispersion, the per-pixel spread was globally scaled using the same variance calibration factor, $k = \sqrt{\mu V_{\text{target}} / \mu V_{\text{current}}} \approx 0.60$.

Naturally, both EMOS variants successfully reproduce the total ensemble variance of CERRA-EDA, as they are directly calibrated to match this value, as reported in Table 17. The more interesting question is whether they can approximate the *spatial distribution* of the variance. For the standard EMOS approach, which relies on a single value of σ , this is inherently impossible. However, the local EMOS formulation, which estimates parameters for each grid point, may be able to capture some spatial structures. Having computed spatial variance maps analogous to those obtained for the diffusion ensembles, the local EMOS results were added to Figure 35. As shown in the figure, EMOS identifies a few regions of high variability, but these are limited in extent and fail to represent the complex spatial patterns of variance clearly captured by both CERRA-EDA and the diffusion model.

The mean-variance discrepancy (MVD) scores computed between EMOS, local EMOS, and the best-performing diffusion configuration (8 steps), reported in Table 18, confirm this observation. In the yearly comparison, the diffusion model clearly outperforms both EMOS configurations, and the same holds for three of the four seasonal partitions. Interestingly, however, during the July–August–September

MVD Scores for Ensemble Diffusion (8 steps) vs EMOS			
	8 steps (Ours)	EMOS	Local EMOS
Yearly	<u>1.40e-01</u>	1.60e-01	1.59e-01
JFM	<u>1.11e-01</u>	1.65e-01	1.59e-01
AMJ	<u>1.37e-01</u>	1.51e-01	1.52e-01
JAS	1.51e-01	<u>1.40e-01</u>	1.44e-01
OND	<u>1.17e-01</u>	1.43e-01	1.40e-01

Table 18: Mean variance discrepancy (MVD) scores computed between the spatial mean variance maps of the 8-step Ensemble Diffusion model, EMOS, and Local EMOS for each three-monthly period (JFM, AMJ, JAS, OND) and the full year 2016. Lower scores indicate higher similarity between the variance distributions of the compared ensembles.

period, the EMOS configuration slightly outperforms diffusion. This finding is noteworthy, given that EMOS is a purely statistical method that does not explicitly model spatial variability. This outcome likely reflects a relative reduction in the diffusion model’s ability to capture uncertainty during the summer months, rather than a genuine advantage of EMOS, suggesting that ensemble diffusion may struggle to represent variance under the distinct dynamical conditions of this season.

5.5 Conclusions

In this paper, we applied an ensemble diffusion model to the full-domain ERA5-to-CERRA downscaling task and discovered a relationship between the variance exhibited by the ensemble diffusion and the number of reverse diffusion steps, giving theoretical proof of this relation. Notably, we demonstrated that the selection of the reverse diffusion step size Δt not only impacts the time efficiency of the generative process but also serves as a crucial mechanism for controlling the variance of the sampled data, a key aspect when using these models for generating ensembles. We have explored architectural improvements needed to apply ensemble diffusion models over large domains, including the use of a scaling factor to reduce the signal rate in the noise scheduler, reducing the effect that higher resolution have on denoising difficulty. Our experimental results indicate that adjusting the number of diffusion steps enables the model to operate across different ranges of ensemble variance. We found that the control over variance gained by varying the number of diffusion steps is sufficient to match the global mean variance exhibited by the available ensemble reference dataset, CERRA-EDA. We observed that increasing the number of diffusion steps leads to convergence toward a fixed variance value. Although this converged global mean variance closely matches the one exhibited by CERRA-EDA, our MVD score, which takes into account the spatial distribution of the variance, revealed that the optimal number of steps is 8, a value far from the convergence point. This demonstrates that selecting a “sufficiently high” number of steps is not necessarily an optimal strategy for tuning an ensemble diffusion model, highlighting the importance of calibration to obtain optimal results. To further contextualize the effectiveness of our approach, we have also introduced an Ensemble Model Output Statistics (EMOS) baseline as a comparison method. EMOS, a widely adopted statistical post-processing technique for ensemble calibration, provided a valuable reference for assessing the seasonal variability and overall performance of our diffusion model. While EMOS successfully reproduced global variance levels, it lacked the spatial variance coherence achieved by the diffusion-based approach, underscoring the advantages of learning-based generative methods for ensemble generation. In our work we have showcased the capability of generating ensemble members efficiently at high temporal and spatial resolutions. Specifically, our ensemble reference CERRA-EDA has a considerable reduction in spatial resolution (from 1069×1069 to 565×565) and temporal resolution (from $3h$ to $6h$). We reveal that our approach is able to generate ensemble members at full CERRA temporal and spatial resolution, while maintaining high efficiency, essentially running on workstation hardware. Considering that our model is conditioned only by ERA5 low-resolution inputs, which happens to be available at hourly intervals, we can additionally increase the ensemble temporal resolution to $1h$. Additionally, we have extended our ensemble diffusion methodology from CERRA to CARRA-EAST, demonstrating its capability to generate ensemble members for datasets lacking ensemble information, using a similar training setup

and confirming its effectiveness in reproducing spatial variance patterns through diffusion step tuning.

Since our diffusion model is trained solely on the deterministic ERA5-to-CERRA task, it is unaffected by factors such as measurement quality or other previously assessed uncertainties. As such, our approach can serve as a validation tool for existing ensembles, with discrepancies providing insights into the neural model’s understanding of the underlying physical system or serving as validation for uncertainty estimates obtained through other methods. In our experimental setting, the spatial variance reported in Figure 35 shows a clear overlap between distributions. However, during the summer months in the North African region, discrepancies between CERRA-EDA and our Ensemble Diffusion appear. This may indicate that our model struggles to capture uncertainty in this region, or that CERRA-EDA itself may be affected by data scarcity [RBLM⁺24].

In conclusion, our methodology provides a practical tool for tuning ensemble diffusion models to produce the correct variance distribution. We anticipate that future implementations will consider the number of diffusion steps as a crucial tuning parameter. Furthermore, our theoretical framework opens promising directions for studying the statistical properties of ensemble members in relation to the structure of diffusion models, encouraging further research.

6 Paper III: Vision Transformers for Multi-Variable Climate Downscaling, emulating Regional Climate Models with a Shared Encoder and Multi-Decoder Architecture

Abstract

Global Climate Models (GCMs) are critical for simulating large-scale climate dynamics, but their coarse spatial resolution limits their applicability in regional studies. Regional Climate Models (RCMs) address this limitation through dynamical downscaling, albeit at considerable computational cost and with limited flexibility. Deep learning has emerged as an efficient data-driven alternative; however, most existing approaches focus on single-variable models that downscale one variable at a time. This paradigm can lead to redundant computation, limited contextual awareness, and weak cross-variable interactions. To address these limitations, we propose a multi-variable Vision Transformer (ViT) architecture with a shared encoder and variable-specific decoders (1EMD). The proposed model jointly predicts six key climate variables: surface temperature, wind speed, 500 hPa geopotential height, total precipitation, surface downwelling shortwave radiation, and surface downwelling longwave radiation, directly from GCM-resolution inputs, emulating RCM-scale downscaling over Europe. Compared to single-variable ViT models, the 1EMD architecture improves performance across all six variables, achieving an average MSE reduction of approximately 5.5% under a fair and controlled comparison. It also consistently outperforms alternative multi-variable baselines, including a single-decoder ViT and a multi-variable U-Net. Moreover, multi-variable models substantially reduce computational cost, yielding a 29–32% lower inference time per variable compared to single-variable approaches. Overall, our results demonstrate that multi-variable modeling provides systematic advantages for high-resolution climate downscaling in terms of both accuracy and efficiency. Among the evaluated architectures, the proposed 1EMD ViT achieves the most favorable trade-off between predictive performance and computational cost.

6.1 Introduction

Global Climate Models (GCMs) and Regional Climate Models (RCMs) are the primary tools for simulating the Earth’s climate system and projecting future climate change, with their respective roles and limitations described in detail in Section 2.1.4. While GCMs provide global coverage at coarse resolution, RCMs add spatial detail and realism but remain computationally demanding, which limits the number of simulations that can be performed in regional experiments [JPE⁺14].

An emerging alternative to traditional numerical downscaling is the use of deep learning models to emulate the RCM process. As outlined in Section 2.3, neural networks can learn the mapping from coarse-resolution GCM outputs to high-resolution RCM fields, providing a fast, data-driven method for producing detailed climate projections across regions and scenarios [RHG⁺24].

The GCM-to-RCM emulation task considered here is analogous to the ERA5-to-CERRA downscaling addressed in the previous papers, but applied in the context of climate projections. We approach it as a supervised learning problem using paired GCM–RCM datasets, focusing on Vision Transformer (ViT) architectures, as introduced in Section 3.5. ViTs have emerged as powerful architectures in image-to-image translation and super-resolution tasks due to their attention-based processing and ability to capture long-range dependencies [ZDC⁺24, YYWS24, RHG⁺24, LSG⁺25]. Our objective is to generate high-resolution outputs that reproduce the fine-scale structures of RCM simulations in a computationally efficient manner.

The main focus of this final paper is the exploration of joint, multi-variable processing. Typically, downscaling meteorological variables from GCM to RCM involves handling numerous diverse variables spanning atmospheric, surface, and radiation categories. Most existing deep learning approaches for climate downscaling focus exclusively on single-variable modeling, training and applying uni-variate models separately on each variable [BMMG20, VKG⁺17, OIKY23, SGIWS20a, SGIWS20b]. This practice raises several issues: cross-variable interactions remain unexploited, maintaining multiple independent models is computationally inefficient, and similar variables lead to redundant learning when modeled separately. To overcome these limitations, this study investigates multi-variable processing strategies. By modeling several variables simultaneously, we aim to exploit inter-variable relationships, improve performance, and enhance physical coherence. This approach also reduces the number of required models, lowering computational demands per variable. At the same time, multi-variable

modeling introduces new challenges, including training instabilities and cross-variable interference, which may manifest as artifacts in the outputs.

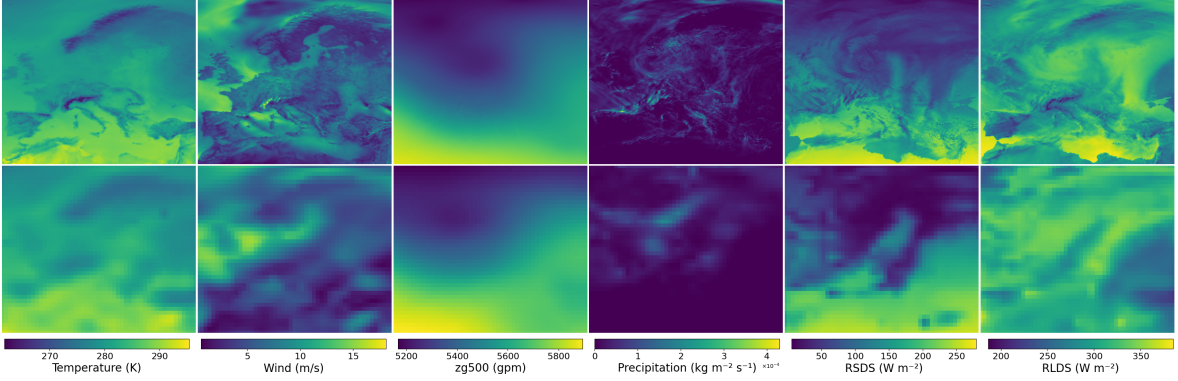


Figure 39: Visualization of a single daily sample for the six studied variables: Surface Temperature, Wind Speed, Geopotential Height, Total Precipitation, Surface Downwelling Shortwave Radiation and Surface Downwelling Longwave Radiation. Each variable is shown for both the high-resolution Regional Climate Model ICTP-RegCM4-6 (top row) and the low-resolution Global Climate Model MPI-ESM-LR (bottom row) re-gridded onto the same European domain.

Our study focuses on six atmospheric variables: wind speed (sfcWind), temperature (tas), geopotential height at 500 hPa (zg500), total precipitation (tp), surface downwelling shortwave radiation (rsds) and surface downwelling longwave radiation (rlds). An image depicting a sample of the six variables for both RCM and GCM is reported in Figure 39. We first establish baseline performance using single-variable ViT models trained independently for each variable. Subsequently, we introduce two multi-variable approaches: one treating variables as separate channels (1E1D), and a second employing a single shared encoder with multiple dedicated decoders (1EMD), one per variable. In addition, we incorporate a multi-variable U-Net model to provide an external and widely used baseline for comparison. We ensure a robust and reproducible comparison by standardizing model structure, parameter count, training epochs, learning rates, and optimizers across all experiments.

Our findings reveal that multi-variable modeling offers clear advantages in terms of computational efficiency. However, the simpler multi-channel approach (1E1D) suffers from cross-variable interference, failing to match the performance of single-variable baselines. In contrast, our structured multi-decoder design (1EMD) avoids these issues and outperforms single-variable models across all tested variables. These results highlight the effectiveness of our multi-variable architecture for high-resolution climate downscaling.

In summary, the main contributions of this paper are: (i) introducing a multi-variable Vision Transformer framework for GCM-to-RCM emulation, (ii) evaluating its performance against single-variable and alternative multi-task formulations, (iii) analyzing the benefits of shared spatial representations across variables for improving physical coherence, and (iv) discussing the challenges introduced by cross-variable interference and potential mitigation strategies.

6.1.1 Data Selection

Our study is based on the EURO-CORDEX data. EURO-CORDEX, developed as part of the Coordinated Regional Downscaling Experiment (CORDEX) under the World Climate Research Programme (WCRP), provides a consistent set of high-resolution (0.11°) RCM simulations, driven by boundary conditions from multiple GCMs and emission scenarios. These simulations are widely used for regional climate impact assessments across Europe [JPE⁺14]. We subjectively selected one GCM-RCM couple from the available simulations: the MPI-ESM-LR GCM with the ICTP-RegCM4-6 RCM simulation under emissions scenario RCP8.5.

The selection of surface air temperature (tas), surface wind speed (sfcWind), geopotential height at 500 hPa (zg500), total precipitation (tp), surface downwelling shortwave radiation (rsds), and surface downwelling longwave radiation (rlds) was motivated by both practical relevance and methodological interest. These six variables exhibit markedly different spatial and statistical characteristics

when downscaled. Temperature and geopotential height produce relatively smooth, large-scale fields, whereas wind speed contains localized high-frequency patterns. Precipitation, in contrast, is sparse, intermittent, and notoriously challenging to downscale. Radiative fluxes (rsds and rlds) reflect a mixture of smooth background structure and localized effects driven by cloud cover. This diversity offers a useful testbed for evaluating how well data-driven models handle different signal types, and whether modeling them jointly can facilitate positive transfer across variables.

The choice was also shaped by the broader goal of exploring multi-variable downscaling. By selecting a small but diverse set of variables, we can begin to assess whether downscaling performance benefits from shared representations across related but physically distinct fields. The downscaling task addresses a spatial resolution increase from 1.0° to 0.1° , representing a tenfold enhancement. This is among the largest resolution gaps available in publicly accessible GCM-RCM datasets, offering a particularly demanding test case for evaluating the capabilities of data-driven downscaling methods.

The domain of interest is Europe, selected for its climatic diversity, data availability, and relevance to policy and research communities. To ensure a direct comparison between scales, the low-resolution GCM data was interpolated onto the RCM’s 0.1° grid using bilinear interpolation. This pre-processing step ensures that GCM and RCM fields are directly comparable in both space and time. The datasets span daily time steps from 2006 to 2099.

6.1.2 Related Works

As presented in Section 2.3, in recent years, the focus of climate downscaling has increasingly shifted toward deep learning models [SDR⁺24b, BMMG21, BMMG20]. A wide range of architectures has been explored to downscale variables such as temperature, wind components, surface pressure, and precipitation, among others. These include convolutional neural networks (CNNs) [BMMG21, WTL⁺21, DL22, JLW⁺22, DIS⁺24, VKG⁺17], U-Nets [SGIWS20a, HKHW20, DSG⁺23], Vision Transformers (ViTs) [ZDC⁺24, YYWS24, LSG⁺25], generative adversarial networks (GANs) [HMC⁺22b, KAS⁺23, SGHK20, MRZ⁺25], and diffusion models [AHP⁺24, MBC⁺23, MAC24a, MEL25, WM24].

Different architectures show varying strengths depending on the target variable. Generative models like GANs and diffusion models are particularly effective at preserving high-frequency spatial detail and have shown strong performance in downscaling wind speed [TFC25]. In contrast, smoother fields such as temperature are often better handled by U-Nets [LSG⁺24]. Despite these findings, there is no clear consensus on a universally optimal architecture. Recently, ViT-based models have shown competitive and often superior performance across multiple climate variables, outperforming CNN-based baselines in several studies [LZVO25, CFL⁺22, ZDC⁺24]. As such, ViTs form the foundation of our proposed approach.

ViT-based architectures for image-to-image tasks often draw inspiration from the encoder-decoder structure of U-Nets. One common design integrates a transformer-based encoder with a convolutional decoder [CLY⁺21, RBK21, LCS⁺21], while other variants opt for fully transformer-based pipelines [WCB⁺22, ZLZ⁺22]. A key architectural consideration is the treatment of skip connections, which preserve spatial resolution and improve gradient flow. These may be preserved directly [CLY⁺21, ZGS⁺25], omitted entirely under the assumption of rich encoder representations [RBK21], or implemented through attention-based gating mechanisms [WCB⁺22].

While the majority of deep learning-based downscaling studies employ single-variable models, where each variable is modeled independently, there is growing interest in multi-variable approaches that leverage cross-variable relationships. These approaches aim to enhance physical consistency across variables and potentially improve accuracy. In this work, we design a multi-variable downscaling framework and benchmark its performance against single-variable counterparts.

Multi-variable downscaling, introduced in Section 3.2.6, naturally aligns with the framework of multi-task learning, where a single model learns to solve multiple related tasks simultaneously. In climate applications, multi-task learning offers several potential advantages: improved generalization through shared representations, reduced training and inference cost, and the possibility of positive transfer across variables [Cra20].

Several recent studies support the promise of this approach. For instance, Ling et al. [LLL⁺22a] present a multi-task model for seasonal climate prediction that jointly forecasts several indices related to the Indian Ocean Dipole. The model is trained using a loss function defined as a weighted sum of the individual task with a higher loss weight to the primary target (DMI). The model achieves improved performance compared to single-variable baselines. Similarly, Fan et al. [FZM⁺20] propose M2GSNet,

a graph-based multi-task model for wind power prediction across multiple farms. By utilizing shared encoders across tasks and training with an unweighted sum of L1 losses across farms, the model achieves superior forecasting accuracy compared to single-variable baselines.

Other examples further support the benefits of multi-task formulations. Bannai et al. [BXU+23] use a shared CNN backbone with two heads for simultaneous rain/no-rain classification and rain rate estimation, yielding improved performance through joint optimization of a combined loss function. Liu et al. [LWP+23] propose a two-stage downscaling system incorporating bias correction, terrain inputs, and a multi-task loss with physical constraints, demonstrating the potential of multi-variable models for enhancing both accuracy and realism in climate data products.

6.2 Methodology

In this section, we start by detailing our backbone implementation of a ViT-based image-to-image model. We then proceed to describe the single-variable approach, where separate encoder-decoder architectures are trained independently for each target variable. Next, introduce two multi-variable strategies. The first, a single-encoder single-decoder (1E1D) model, concatenates multiple variables along the channel dimension, treating them as multi-channel images in a unified ViT pipeline, and the second a single-encoder multi-decoder (1EMD) where a shared encoder extracts common features, while separate decoders are dedicated to reconstructing each variable individually. A visual comparison of the proposed approaches is shown in Figure 40.

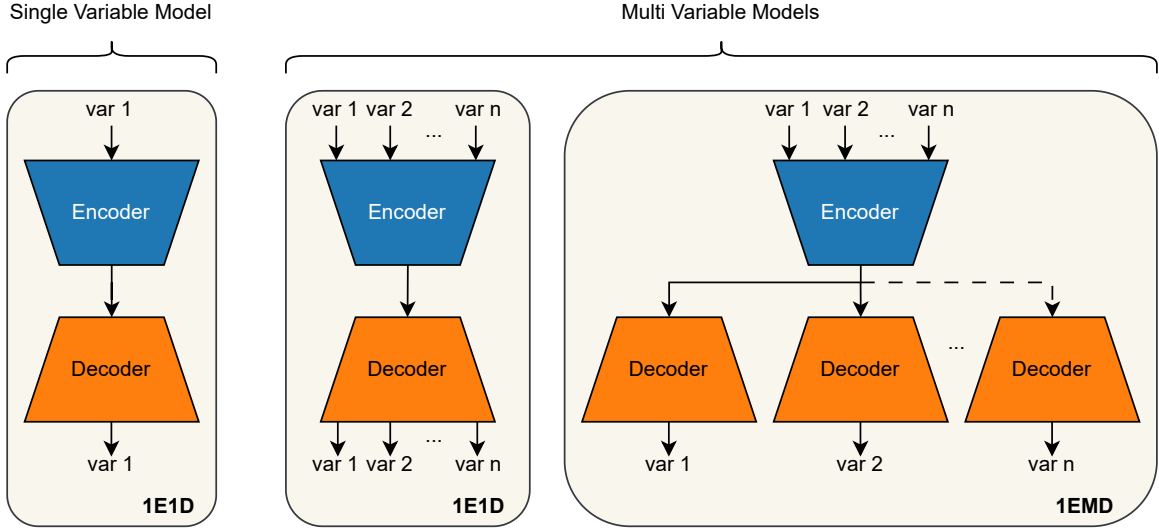


Figure 40: Comparison between a single-variable encoder-decoder model and multi-variable models using either a single shared decoder (1E1D) or multiple decoders branching from a shared encoder (1EMD).

6.2.1 Architectural Backbone: Vision Transformer (ViT)

Vision Transformers (ViTs) are neural architectures that originate in natural language processing, where self-attention is used to capture relationships within sequences. In the vision domain, ViT applies the same principle by splitting an image into fixed-size patches, which are then flattened, embedded, and treated as a sequence. This sequence is processed by transformer blocks, allowing the model to capture global image dependencies without relying on convolutions, effectively serving as an alternative to traditional encoder networks.

In recent years ViTs have been successfully applied to a variety of super-resolution tasks [LLL+22b, YZH+25, CLZVG21, LSWC23]. Our proposed ViT architecture belongs to the ViT-CNN hybrid category, combining a transformer-based encoder with a convolutional decoder, enabling the model to capture global, non-local relationships more effectively than purely convolutional networks [CLY+21,

[LCS⁺21]. This design allows the model to handle complex spatial patterns, which is especially important for tasks that require modeling long-range and cross variable dependencies.

Skip connections are a well-established component of encoder-decoder architectures, commonly used to preserve spatial detail and improve gradient flow. In ViT-CNN hybrid models, studies have shown that skip connections can enhance performance by bridging lower-level features from the encoder to the decoder [CLY⁺21]. However, in transformer-based architectures, especially those without hierarchical designs, this benefit may be less critical [RBK21, ZLZ⁺21b]. Unlike CNN encoders, which typically reduce spatial resolution through downsampling, our ViT encoder preserves a constant resolution throughout. As a result, the output tokens retain the same spatial structure as the input, reducing the need for skip connections to recover lost detail.

In our implementation, we chose to omit skip connections to prioritize modularity and simplicity, particularly in the context of supporting multiple variable-specific decoders. In a multi-decoder setting, implementing skip connections would require duplicating and routing encoder features to each decoder, increasing architectural complexity and coupling. Moreover, since each decoder specializes in a different output variable, sharing low-level features indiscriminately across all decoders could introduce unintended interactions or noise between tasks, potentially harming performance. Given that the empirical benefits of skip connections in transformer encoders remain inconclusive, we deemed the flexibility of a skip-free design—enabling easy addition or removal of decoders—more valuable than a potentially marginal performance gain.

Specifically, our architecture is designed as follows: The encoding path follows a standard ViT configuration, where the input image is divided into non-overlapping 8×8 patches, each linearly projected into a 256-dimensional vector by a dense embedding layer. To retain spatial information, learnable positional embeddings are added to the patch representations before passing them through the transformer stack. The encoder consists of six transformer layers, each following a consistent structure. Each layer applies normalization, followed by multi-head self-attention with six heads and a dropout rate of 0.1. Next, A residual connection adds the input of the attention block to its output. After attention, a second normalization is applied, followed by a multi-layer perceptron (MLP) consisting of two fully connected layers with 512 and 256 units, respectively, and GELU [HG16] activation. Another residual connection is applied at the output of the MLP block. Through this process, the transformer encoder learns both local and global relationships between spatially distributed patches. To mitigate the memory constraints that limit batch size, we use Group Normalization [WH18] with a group size of 8, which has been shown to be more stable than Batch Normalization in such settings.

Following the transformer encoder, the sequence of embeddings is reshaped into a spatial grid, initializing the decoding path. The decoder is composed of three consecutive upsampling blocks, each of which includes bilinear interpolation for upsampling, doubling the spatial resolution at each step. We utilize bilinear interpolation avoiding artifacts often associated with learned transposed convolutions [ODO16]. Each bilinear upsampling is followed by a residual block, composed of group normalization, two convolutional layers, and a residual connection that preserves input information. The first convolution in the block uses Swish activation, while the second uses no activation. The feature width begins at 256 in the first upsampling block and is halved in each subsequent block, reducing to 32 in the final block. A final convolutional layer with no activation and a kernel initializer set to zero projects the output to the target resolution, completing the decoding process. The complete architecture for a single encoder-decoder pair contains approximately 11 million learnable parameters.

6.2.2 Single-Variable Models

The most direct approach at downscaling multiple variables is based on the use of multiple single-variable models trained independently on each variable, with no sharing of weights or other information between each model.

Each individual model consists of a single encoder-decoder pair, which enables the model to focus on the individual task but doesn't enable the utilization of cross-variable information.

In the case of single variable the loss is calculated on the comparison between the generated output and the available ground truth, with our implementation using mean square error (MSE) loss. Denoting $\hat{Y}_{hw}$ as the predicted pixel at spatial position (h, w) and Y_{hw} as the corresponding ground truth pixel, the loss is defined as:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{HW} \sum_{h=1}^H \sum_{w=1}^W \left(\hat{Y}_{hw} - Y_{hw} \right)^2 \quad (72)$$

6.2.3 Multi-Variable Models

The main goal of this work is to design an architecture capable of downscaling multiple variables simultaneously. While the coarse-resolution GCM provides the primary input, we hypothesize that additional value can be gained by exploiting inter-variable relationships. Our assumption is that the atmospheric state encodes joint spatial and dynamical patterns across variables, which a shared representation can capture more effectively than isolated models. By modeling all variables together, the network can leverage these patterns, potentially leading to improved accuracy and greater physical consistency compared to independent downscaling approaches.

Building upon the ViT architecture described in Section 6.2.1, we implement two multi-variable variants: a Single-Encoder Single-Decoder (1E1D) model, where all variables share the same encoder-decoder pipeline, and a Single-Encoder Multi-Decoder (1EMD) model, where the variable are jointly encoded but decoded separately. These designs allows us to study the trade-off between fully shared architectures and variable-specific specialization via independent decoders.

Both our proposed multi-variable approaches share the same loss structure. Training is performed simultaneously across the six target variables, with the loss computed as the mean over all output channels. To make this approach viable, it is essential to normalize all variables to the same range. This ensures that the model optimizes all variables with equal priority, preventing any one variable from dominating the loss due to scale differences. With MSE being our loss function of choice, and denoting $\hat{Y}_{nhw}$ as the predicted pixel at spatial position (h, w) for variable n , and Y_{nhw} as the corresponding ground truth pixel, the joint loss is defined as:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{NHW} \sum_{n=1}^N \sum_{h=1}^H \sum_{w=1}^W \left(\hat{Y}_{nhw} - Y_{nhw} \right)^2 \quad (73)$$

Single-Encoder Single-Decoder VIT (1E1D) Our first approach to multi-variable modeling treats the different atmospheric fields as complementary features of a single spatial input by concatenating the variables along the channel dimension. This design keeps the single-variable architecture largely intact, with modifications only to the patch embedding and final convolutional layers to accommodate six input and output channels.

The input consists of six geophysical variables stacked as a six-channel image. We divide this input into non-overlapping patches of shape $8 \times 8 \times 6$, each of which is projected into a 256-dimensional vector via a dense embedding layer. These token embeddings are then processed by a Vision Transformer (ViT) encoder, following the same architecture used in the single-variable model. The decoder, composed of convolutional layers, upsamples the latent representation back to the original spatial resolution, with the final layer producing six output channels corresponding to the predicted fields.

This approach allows the encoder to capture a joint representation of the atmospheric state across multiple variables. However, it implicitly assumes that the variables are spatially aligned and semantically compatible when treated as separate channels, an assumption that may be limiting if the variables exhibit different spatial structures or temporal dynamics.

Single-Encoder Multi-Decoder VIT (1EMD) The six target variables in our downscaling task exhibit substantial differences in their spatial structures and statistical distributions. For example, temperature fields tend to show smoother gradients compared to the more intricate patterns found in wind components. When using a single decoder to reconstruct all variables jointly, these differences can cause interference: the decoder is limited to sharing the same weights for generating the different variables, leading to potential cross-variable contamination. In particular, we observed that high-frequency patterns from one variable (e.g., wind) can bleed into the smoother reconstructions of another (e.g., geopotential height), degrading performance.

To address this, we propose a novel multi-variable downscaling architecture that employs a single transformer-based encoder shared across all variables, followed by a dedicated convolutional decoder

for each output variable. This design preserves the benefit of jointly encoding spatial inter-variable dependencies while allowing each decoder to specialize in reconstructing a specific variable. By decoupling the decoding paths, we reduce interference and provide sufficient representational capacity to each output stream. We consider this architecture to be an effective compromise between the two aforementioned approaches: running completely independent encoder-decoder pipelines for each variable, and forcing all variables through a single encoder-decoder with channel concatenation. Our approach leverages shared context while controlling model complexity and avoiding cross-variable entanglement, achieving a favorable trade-off between accuracy and computational cost.

Specifically, the 1EMD architecture is structured as follows. Each low-resolution input variable is first bilinearly upsampled to the target resolution of 504×432 , matching the output grid. The resulting variables are concatenated along the channel dimension and partitioned into 3402 non-overlapping patches of size $8 \times 8 \times 6$. Each patch is linearly embedded into a 256-dimensional vector and augmented with positional information via additive encoding. The sequence of embedded patches is processed by a stack of six transformer blocks, each featuring multi-head self-attention (six heads), residual connections with group normalization, and MLP layers with GELU activations.

The final encoder output is reshaped into a spatial feature grid of size $54 \times 63 \times 256$ and passed in parallel to six independent convolutional decoders, one per variable. The decoders do not share parameters and are trained jointly using a composite loss function that accounts for all variables. Each decoder consists of a sequence of residual blocks with group normalization and Swish activations, interleaved with 2D upsampling layers. Three upsampling stages are used to restore the original resolution of 504×432 . The final output is produced by a zero-initialized convolutional layer without activation. The complete 1EMD model architecture is reported in Figure 41.

Experimental results, discussed in detail in Section 6.4, demonstrate that the proposed 1EMD model outperforms both the single-variable and 1E1D approaches.

6.2.4 Multi-Variable U-net

In imaging tasks, the U-Net architecture represents a widely adopted standard baseline, with applications spanning much of the image-to-image translation spectrum. U-Nets have also been extensively used for statistical downscaling of meteorological variables, including several recent contributions specifically addressing multivariate downscaling [ALDC25, QCBMBB23, LWH⁺25]. To provide a meaningful and broadly recognized comparison against our transformer-based single-variable and multi-variable approaches, we include a multi-variable U-Net structured in a 1E1D configuration.

The implemented residual U-Net follows a standard symmetric encoder-decoder design built around residual blocks. Each residual block applies group normalization followed by two 3×3 convolutional layers. The encoder is composed of three downsampling stages, each containing three residual blocks and a $2 \times$ spatial reduction achieved via average pooling. Feature maps from every block are stored as skip connections and later concatenated in the decoder. The bottleneck stage processes the lowest-resolution representation through multiple residual blocks at the widest channel depth. The decoder mirrors the encoder, performing $2 \times$ bilinear upsampling at each stage and concatenating the corresponding skip features, followed by residual blocks.

In the multi-variable, multi-task setting, the six variables are concatenated along the channel dimension in the input. A symmetric multi-channel output head is used to reconstruct the six variables, preserving a direct correspondence between input and output channels. The sizing of the model was designed to match the competing ViT based architectures in terms of execution and training times. The final configuration comprises approximately five million trainable parameters.

6.3 Experimental Setup

Our objective is to learn a mapping from coarse-resolution Global Climate Model (GCM) to high-resolution Regional Climate Model (RCM). Specifically, we use data from the MPI-ESM-LR GCM and the ICTP-RegCM4-6 RCM under the RCP8.5 scenario, addressing a spatial resolution shift from 1 degree (approximately 100 km) to 0.1 degree (approximately 10 km). The GCM data are re-gridded to the RCM domain using bilinear interpolation to match the spatial structure of the RCM over Europe.

We focus on downscaling six key atmospheric variables: near-surface air temperature (tas), surface wind speed (sfcWind), 500 hPa geopotential height (zg500), total precipitation (tp), surface down-

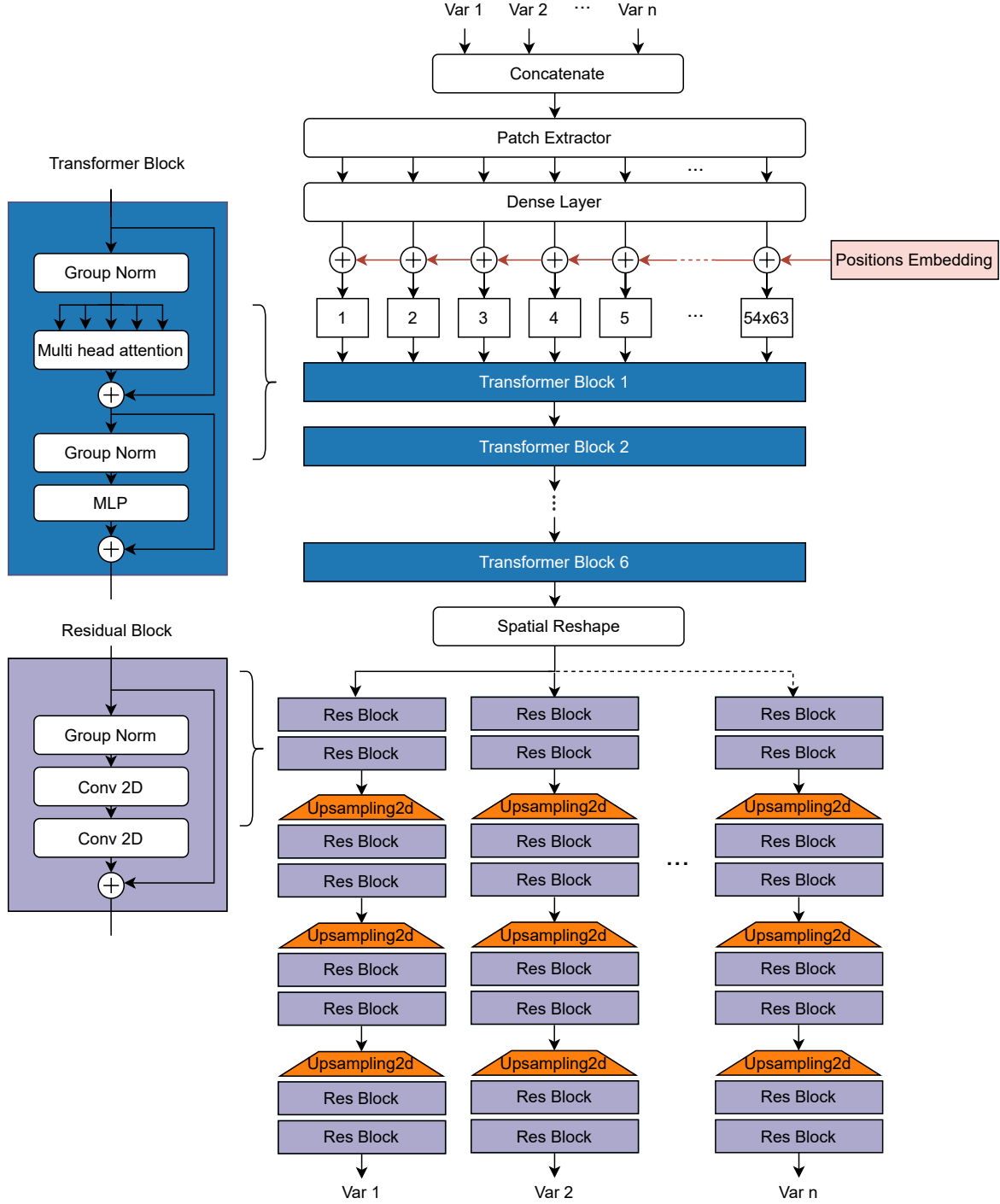


Figure 41: Architecture of the multi-variable single-encoder multi-decoder model (1EMD). Concatenated input variables are divided into patches, embedded with positional encodings, and processed as a sequence by six transformer blocks, each consisting of multi-head self-attention (six heads) followed by an MLP. The output sequence is then reshaped into a spatial grid and passed to separate CNN-based decoders, each composed of three upsampling stages interleaved with residual blocks.

welling shortwave radiation (rsds), and surface downwelling longwave radiation (rlds). From an image processing perspective, each variable is treated as a 2D field with spatial dimensions of 431×501 pixels.

Our dataset spans daily values from 2006 to 2100. We use data from 2006–2099 for training and reserve one year per decade (2010, 2020, etc.) for testing. This design avoids long-term temporal separation between training and test sets, helping prevent bias from trends such as climate change. For each variable, GCM and RCM outputs are temporally and spatially aligned so that the models can effectively learn to predict high-resolution RCM fields from corresponding low-resolution GCM inputs.

As of pre-processing, each variable is min-max normalized to the $[0, 1]$ range using the training data’s minimum and maximum values. To ensure compatibility with convolutional architectures, images are also padded via replication to 432×504 pixels to make the dimensions divisible by 8. Padding is removed before computing evaluation metrics.

Our downscaling models operate in an image-to-image manner, taking the GCM variable(s) at a given timestamp as input and predicting the corresponding RCM variable(s). In the single-variable case, the model maps one GCM field to one RCM field. In the multi-variable case, the model maps all six GCM variables to their corresponding RCM fields for the same timestamp, effectively forming a six-to-six mapping as illustrated in Figure 40.

We adopt a fair comparison framework that is coherent with standard practice in model comparison, avoiding common pitfalls such as unequal fine-tuning, training effort or model complexity [LKM⁺18, PVLS⁺21, LS19]. We compare the four modeling approaches, three based on a Vision Transformer (ViT) backbone and one based on a U-Net. First, we train and test six independent single-variable ViT models, one for each target variable. Then, we explore three multi-variable models: 1E1D, which uses a shared encoder and decoder, 1EMD, which employs a shared encoder and separate decoders for each variable, and a multi-variable U-Net providing an external, widely used reference architecture. All ViT models share the same encoder architecture. The single-variable ViT and the multi-variable ViT 1E1D also share the same parameter count, differing only in the final convolutional layer and decoder embeddings. In contrast, the 1EMD model maintains a shared encoder but uses six independent decoders (one per variable), with no weight sharing in the decoding stage.

All models share the same encoder architecture. The single-variable ViT and the multi-variable ViT 1E1D also share the same parameter count, differing only in the final convolutional layer and decoder embeddings. In contrast, the 1EMD model maintains a shared encoder but uses six independent decoders (one per variable), with no weight sharing in the decoding stage.

We use the Adam optimizer with decoupled weight decay (AdamW) [LH17], a common choice for ViT-based architectures. All models are trained for 400 epochs with 500 steps per epoch. Due to memory constraints, we use a batch size of 1. The learning rate is fixed at $1e-4$ for the first 300 epochs and reduced to $1e-5$ for the final 100 epochs. Weight decay is set to one-tenth of the current learning rate. These settings were chosen based on empirical tuning to ensure consistent convergence across all approaches. We use mean squared error (MSE) as the loss function for all models, as described in Sections 6.2.2 and 6.2.3.

Training is conducted on a NVIDIA A100 GPU using a unified pre-processing pipeline. Testing is performed on the reserved ten years of test data, spanning the full 100-year period. Each model is evaluated on the entire test set, and performance metrics are computed over all test samples. To account for variability due to random initialization and stochastic optimization, all models were trained and evaluated twice using independent random seeds. Each repetition executed the full training and testing pipeline under identical settings, and the performance values reported in the manuscript correspond to the mean across the two runs. This protocol provides a lightweight but meaningful assessment of robustness within realistic computational constraints. Notably, the relative ranking of the models is consistent across both repetitions.

6.4 Results

We evaluate the four models, Single-variable ViT, Multi-variable ViT (1E1D), Multi-variable ViT (1EMD) and Multi-variable U-Net on a test dataset comprising ten years of unseen data. The test set includes one year per decade from 2010 to 2100.

All models are tested across the six selected variables using three evaluation metrics: Mean Squared Error (MSE), Mean Absolute Error (MAE), and Structural Similarity Index (SSIM) alongside spatial and temporal Pearson correlations. These metrics offer a well-rounded perspective, balancing numerical accuracy (MSE, MAE) with perceptual similarity (SSIM), and are commonly adopted in climate

GCM to RCM Downscaling Performance (MPI to RegCM) Mean Performance Across Two Independent Runs			
Model	MSE ↓	MAE ↓	SSIM ↑
tas			
Bilinear	9.97e-04	2.27e-02	9.78e-01
Single Var. ViT	4.34e-04	1.47e-02	<u>9.89e-01</u>
Multi Var. ViT (1E1D)	4.50e-04	1.51e-02	9.84e-01
Multi Var. U-Net	4.47e-04	1.48e-02	9.83e-01
Multi Var. ViT (1EMD)	<u>4.23e-04</u>	<u>1.43e-02</u>	9.88e-01
sfcWind			
Bilinear	6.08e-03	5.81e-02	8.35e-01
Single Var. ViT	3.43e-03	4.36e-02	8.85e-01
Multi Var. ViT (1E1D)	3.33e-03	4.29e-02	8.76e-01
Multi Var. U-Net	3.35e-03	4.32e-02	8.69e-01
Multi Var. ViT (1EMD)	<u>3.23e-03</u>	<u>4.21e-02</u>	<u>8.87e-01</u>
zg500			
Bilinear	2.55e-03	3.76e-02	<u>9.95e-01</u>
Single Var. ViT	1.90e-03	3.18e-02	<u>9.95e-01</u>
Multi Var. ViT (1E1D)	1.85e-03	3.14e-02	9.94e-01
Multi Var. U-Net	1.84e-03	3.17e-02	<u>9.95e-01</u>
Multi Var. ViT (1EMD)	<u>1.80e-03</u>	<u>3.08e-02</u>	<u>9.95e-01</u>
tp			
Bilinear	6.00e-07	3.73e-04	<u>9.97e-01</u>
Single Var. ViT	4.46e-07	3.55e-04	<u>9.97e-01</u>
Multi Var. ViT (1E1D)	4.91e-07	3.94e-04	9.96e-01
Multi Var. U-Net	5.00e-07	4.09e-04	9.96e-01
Multi Var. ViT (1EMD)	<u>4.32e-07</u>	<u>3.48e-04</u>	<u>9.97e-01</u>
rsds			
Bilinear	2.24e-02	1.01e-01	7.79e-01
Single Var. ViT	5.37e-03	5.02e-02	8.80e-01
Multi Var. ViT (1E1D)	5.03e-03	4.86e-02	8.76e-01
Multi Var. U-Net	5.00e-03	4.85e-02	8.75e-01
Multi Var. ViT (1EMD)	<u>4.99e-03</u>	<u>4.83e-02</u>	<u>8.83e-01</u>
rlds			
Bilinear	6.70e-03	6.37e-02	9.17e-01
Single Var. ViT	3.06e-03	4.28e-02	9.43e-01
Multi Var. ViT (1E1D)	2.80e-03	4.03e-02	9.43e-01
Multi Var. U-Net	2.81e-03	4.05e-02	9.40e-01
Multi Var. ViT (1EMD)	<u>2.78e-03</u>	<u>4.01e-02</u>	<u>9.47e-01</u>

Table 19: Model performance comparison for downscaling from MPI-ESM-LR to RegCM4-6 for each variable, mean model performance across two independent runs. Best values for each variable are underlined, the results are computed for variables normalized between 0 and 1. Best results (including ties) are underlined.

Spatial and Temporal Correlations (MPI to RegCM)		
Model	Spatial corr ($\mu \pm \sigma$)	Temporal corr ($\mu \pm \sigma$)
tas		
Bilinear	0.910 ± 0.037	0.935 ± 0.016
Single Var. ViT	0.957 ± 0.024	0.956 ± 0.011
Multi Var. ViT (1E1D)	0.957 ± 0.024	0.956 ± 0.010
Multi Var. U-Net	0.955 ± 0.024	0.956 ± 0.011
Multi Var. ViT (1EMD)	<u>0.958 ± 0.024</u>	<u>0.957 ± 0.010</u>
sfcWind		
Bilinear	0.602 ± 0.161	0.528 ± 0.096
Single Var. ViT	0.748 ± 0.113	0.605 ± 0.083
Multi Var. ViT (1E1D)	0.759 ± 0.109	0.636 ± 0.075
Multi Var. U-Net	0.755 ± 0.111	0.631 ± 0.081
Multi Var. ViT (1EMD)	<u>0.765 ± 0.109</u>	<u>0.644 ± 0.073</u>
zg500		
Bilinear	0.929 ± 0.089	0.920 ± 0.033
Single Var. ViT	0.943 ± 0.073	0.935 ± 0.024
Multi Var. ViT (1E1D)	0.943 ± 0.070	0.937 ± 0.023
Multi Var. U-Net	<u>0.944 ± 0.070</u>	<u>0.938 ± 0.023</u>
Multi Var. ViT (1EMD)	<u>0.944 ± 0.070</u>	<u>0.937 ± 0.023</u>
tp		
Bilinear	0.339 ± 0.173	0.340 ± 0.111
Single Var. ViT	0.411 ± 0.158	0.389 ± 0.112
Multi Var. ViT (1E1D)	0.341 ± 0.121	0.321 ± 0.116
Multi Var. U-Net	0.293 ± 0.142	0.298 ± 0.128
Multi Var. ViT (1EMD)	<u>0.438 ± 0.150</u>	<u>0.407 ± 0.106</u>
rsds		
Bilinear	0.745 ± 0.165	0.826 ± 0.048
Single Var. ViT	0.885 ± 0.093	0.946 ± 0.016
Multi Var. ViT (1E1D)	<u>0.894 ± 0.088</u>	<u>0.951 ± 0.014</u>
Multi Var. U-Net	0.893 ± 0.088	0.950 ± 0.013
Multi Var. ViT (1EMD)	<u>0.894 ± 0.087</u>	<u>0.951 ± 0.014</u>
rlds		
Bilinear	0.486 ± 0.168	0.703 ± 0.078
Single Var. ViT	0.791 ± 0.089	0.811 ± 0.053
Multi Var. ViT (1E1D)	<u>0.808 ± 0.086</u>	<u>0.831 ± 0.054</u>
Multi Var. U-Net	0.805 ± 0.088	0.830 ± 0.055
Multi Var. ViT (1EMD)	<u>0.808 ± 0.088</u>	0.829 ± 0.054

Table 20: Spatial and temporal Pearson correlations between model predictions and RegCM4-6 targets for each variable. Spatial correlations are computed for each day and summarized as mean $\pm$ standard deviation over time. Temporal correlations are computed at each grid point and summarized as mean $\pm$ standard deviation over space. Best values (including ties) are underlined.

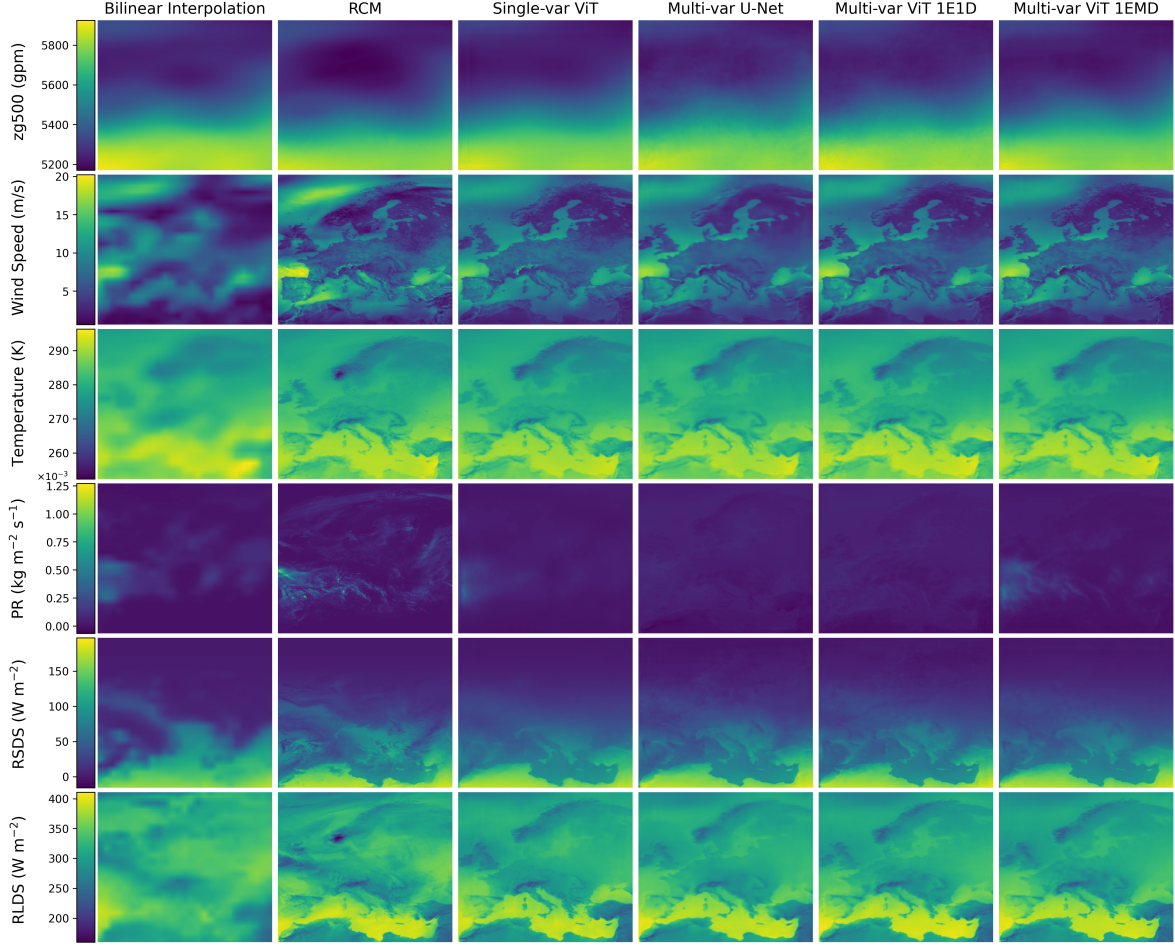


Figure 42: Comparison of the examined models (Single var ViT, multi var ViT 1E1D, Multi var ViT 1EMD), target RCM, and bilinearly interpolated GCM for a sample of the test set. For each sample, each row contains Geopotential, Wind speed, Temperature, Total Precipitation, Surface Downwelling Shortwave Radiation and Surface Downwelling Longwave Radiation for the same timestamp.

imaging studies [BMMG20, AMP+25, YHGH+24, HHGR+23, OIKY23]. All metrics are computed on normalized data to enable cross-variable comparison as the variables differ in their units and ranges.

Table 19 reports the mean performance across two independent runs for all six variables. All deep-learning models provide substantial improvements over the bilinear interpolation baseline. For example, the MSE for near-surface temperature (tas) decreases from 9.97×10^{-4} under bilinear interpolation to 4.34×10^{-4} with the Single-variable ViT. Similarly, for surface wind speed (sfcWind), the bilinear error of 6.08×10^{-3} is reduced by approximately half across all learned models. Comparable improvements are observed for the remaining variables, confirming the value of learning-based downscaling across a diverse set of atmospheric fields. A comparison of the MSE scores in a per-month plot is also reported in Figure 46.

A second consistent pattern is that the Multi-var 1E1D ViT and the Multi-var U-Net achieve nearly identical results for all variables. For instance, on rsds the two models obtain MSE values of 5.03×10^{-3} and 5.00×10^{-3} , while on rlds they reach 2.80×10^{-3} and 2.81×10^{-3} , respectively. This close alignment suggests that architectural differences between transformer-based and convolutional backbones are less influential than the broader design choice of using a single shared decoder when performing multi-variable downscaling.

The comparison with the Single-variable ViT indicates that joint modeling through a single decoder results in positive transfer for several variables. Both 1E1D and the U-Net improve upon the Single-variable ViT for sfcWind (reducing MSE from 3.43×10^{-3} to 3.33×10^{-3} and 3.35×10^{-3}), for zg500 (from 1.90×10^{-3} to 1.85×10^{-3} and 1.84×10^{-3}), for rsds (from 5.37×10^{-3} to 5.03×10^{-3} and

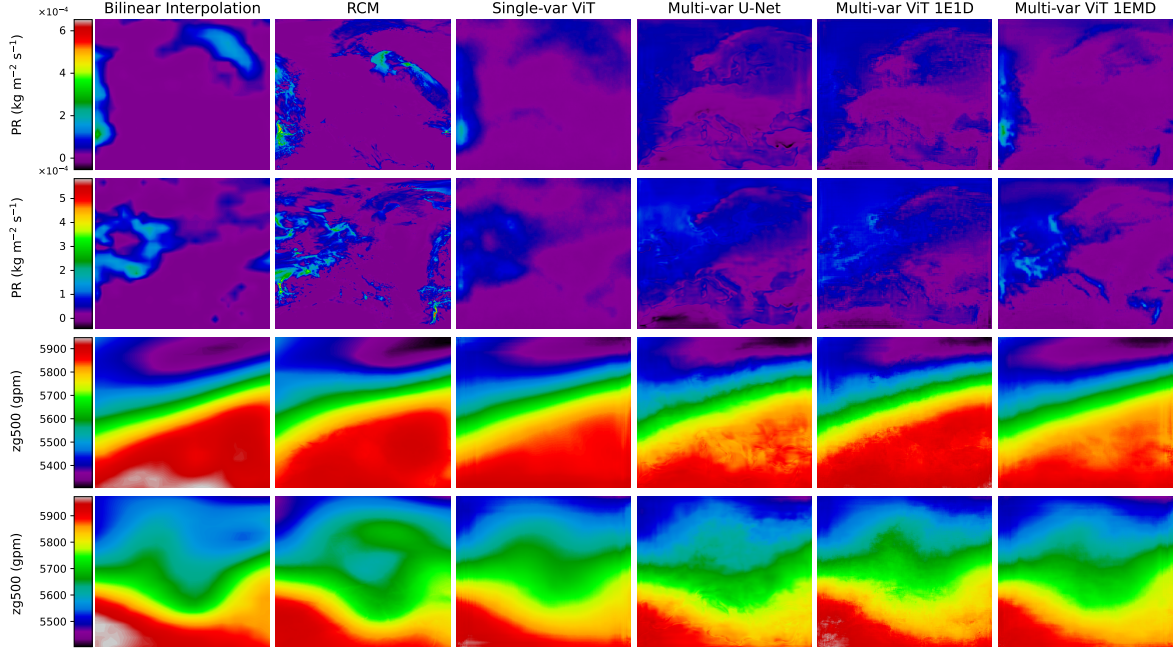


Figure 43: Visualization of cross-variable leakage in the 1E1D and multi-variable U-Net models for total precipitation and zg500. Four samples are shown, two for precipitation and two for geopotential height. Using a high contrast colormap, we can clearly see high-frequency artefacts in the multi-variable U-Net and 1E1D ViT, with faint outlines resembling European geography. This suggests contamination from temperature and wind fields. These outlines are absent from the original data, the single-variable model, and the multi-decoder 1EMD model.

5.00×10^{-3}), and for rlds (from 3.06×10^{-3} to 2.80×10^{-3} and 2.81×10^{-3}). These results indicate that several variables benefit from shared representation learning across related atmospheric patterns. In contrast, the Single-variable ViT performs better for tas and total precipitation (tp). For tas, the MSE increases from 4.34×10^{-4} when using single-variable modeling to 4.50×10^{-4} (1E1D) and 4.47×10^{-4} (U-Net). For tp, the error increases from 4.46×10^{-7} to 4.91×10^{-7} and 5.00×10^{-7} . These results illustrate negative transfer for variables whose statistical or spatial characteristics are not well accommodated by a shared decoder. Additionally, the 1E1D model exhibits cross-variable leakage in zg500, with geographic artifacts emerging in the output, as illustrated in Figure 43.

The proposed 1EMD architecture achieves the best performance for all six variables. Relative to the Single-variable ViT, the MSE decreases from 4.34×10^{-4} to 4.23×10^{-4} for tas, from 3.43×10^{-3} to 3.23×10^{-3} for sfcWind, and from 4.46×10^{-7} to 4.32×10^{-7} for tp. The improvements are consistent across the remaining variables as well. These results indicate that combining a shared encoder with variable-specific decoders successfully incorporates the benefits of multi-variable modeling while mitigating the negative transfer observed in single-decoder approaches. To further assess the performance of 1EMD, a visual comparison of mean absolute error maps between 1EMD and bilinear interpolation is reported in Figure 44.

Additionally, Table 20 reports spatial and temporal Pearson correlations for all variables and models. Spatial correlation quantifies, at each time step, the agreement between predicted and observed spatial patterns and is summarized as mean $\pm$ standard deviation across time. Temporal correlation analogously measures, at each grid point, the agreement between predicted and observed time series and is summarized across space. Across variables, the spatial and temporal correlations reveal two broad regimes of model behaviour. For tas and zg500, all learned models perform similarly, with minimal differences between the single-variable and multivariate configurations. For rsds and rlds, learned models also converge to closely aligned performance, though a modest improvement is still observed when moving from single-variable to multivariate setups. In contrast, wind speed and total precipitation (tp) exhibit much larger performance spreads. For wind speed, the advantage of 1EMD is substantial: it clearly outperforms both the single-variable model and the other multivariate

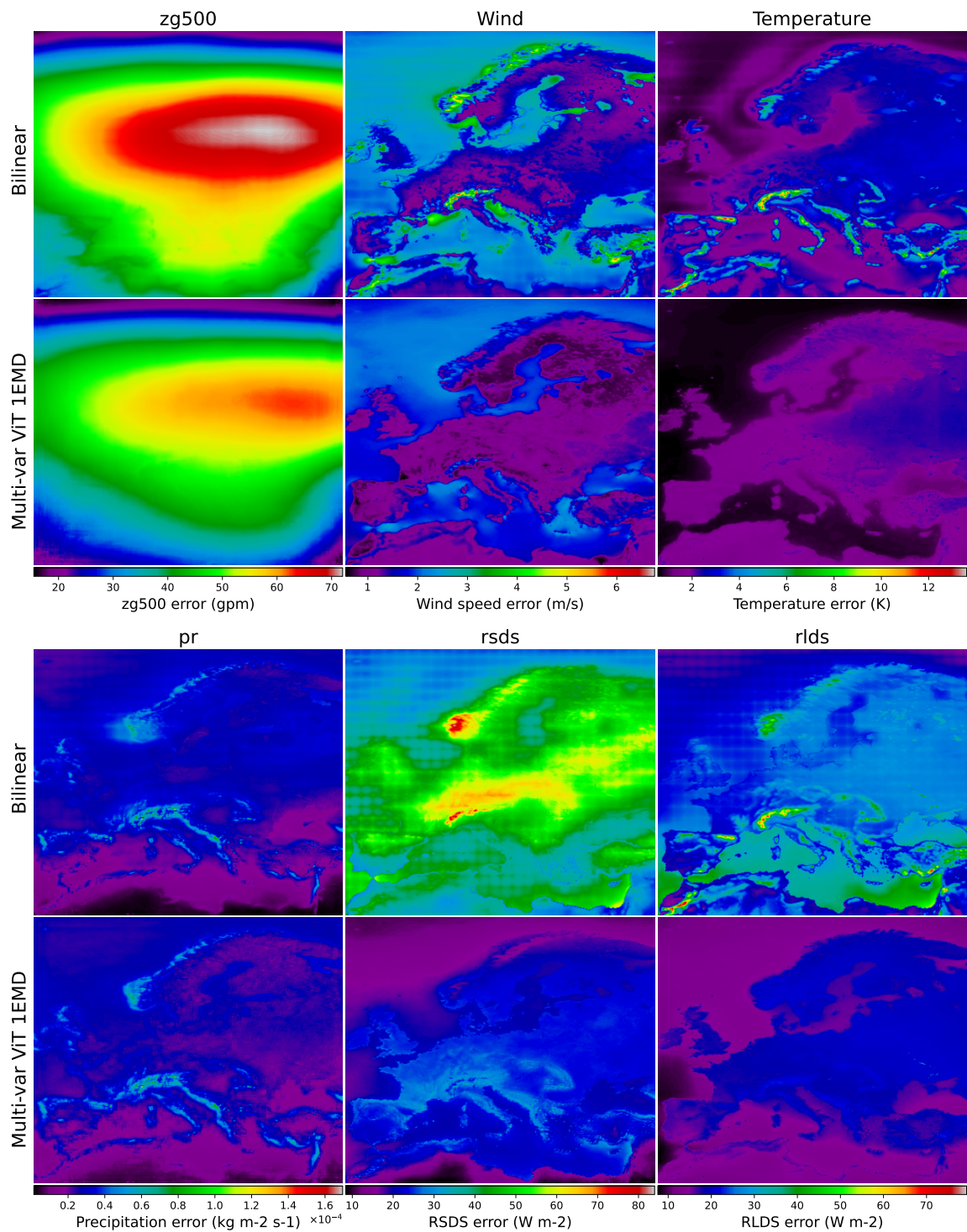


Figure 44: Mean absolute error maps for bilinear interpolation (top rows) and the 1EMD multi-variable ViT (bottom rows). The top panel shows zg500, wind speed, and temperature; the bottom panel shows precipitation, shortwave downwelling radiation, and longwave downwelling radiation. Errors are computed over the full test set, with per-variable color scales.

variants, achieving the highest spatial and temporal correlations (up to 0.76 and 0.64, respectively), whereas the alternative multivariate designs repeatedly underperform. Precipitation displays even more pronounced differences: most multivariate models deteriorate markedly (e.g., temporal correlation dropping to 0.32 in 1E1D), while 1EMD reaches 0.41, surpassing both the single-variable model

(0.39) and all other multivariate approaches. Overall, 1EMD emerges as the most robust multivariate design, with the largest gains observed for the more challenging variables.

Multi-variable models offer clear efficiency advantages by consolidating multiple downscaling tasks into a single model, utilizing a single shared encoder thus reducing the number of computations and requiring fewer model initializations overall. To quantify this, we measure the inference time for each model to generate predictions over the 10-year test dataset. All models are run on the same hardware and share a unified pre-processing pipeline.

In Table 21 we report the number of parameters, execution time, and training time for all models. Although the 1EMD model has a substantially larger number of trainable parameters due to its six independent decoders, its inference time remains similar to that of the 1E1D model. This is expected: using six decoders instead of one increases the parameter count but does not significantly change the number of arithmetic operations, as each decoder performs the same sequence of computations with different weights. Moreover, the transformer encoder dominates the total computational cost, meaning that differences in decoder size contribute only marginally to overall runtime.

Figure 45 presents the total runtime per model (in seconds), along with per-variable time for multi-variable models which process all six variables simultaneously.

From an inference cost standpoint, executing six independent single-variable ViT models over the 10-year test period would require a cumulative runtime of $6 \times 817 \text{ s} = 4,902 \text{ s}$. In contrast, the multi-variable architectures complete the same six-variable downscaling task in 3,336–3,456 s. This corresponds to a reduction in execution time of approximately 29%–32%, demonstrating that multi-variable processing yields a substantial improvement in inference efficiency. It is important to note that these timings exclude the additional overhead associated with loading and managing six separate models (such as repeated weight initialization, memory allocation, and session handling) all of which would further amplify the real-world efficiency gains of the multi-variable approach.

Training costs follow a similar pattern. Training six single-variable ViT models independently would require $6 \times 11,200 \text{ s} = 67,200 \text{ s}$, whereas the multi-variable models achieve full six-variable training in 58,400–61,320 s. This represents an overall reduction of approximately 8%–13% in wall-clock training time. As with inference, these estimates do not account for overheads inherent to conducting six separate training runs (such as checkpoint management, logging, scheduling, and data loading) which suggests that the effective operational efficiency of the multi-variable configurations is likely even greater in practice.

Model	Exec. Time (s)	Train Time (s)	Params (M)
Single-variable ViT	817	11 200	11.63
1E1D ViT	3 409	58 400	15.39
Multi-variable U-Net	3 336	61 320	5.20
1EMD ViT	3 456	60 200	32.25

Table 21: Computational cost and model complexity for the three examined ViT architectures. Execution time is measured over the entire 10-year test dataset. Training time corresponds to 400 training epochs under identical hardware and hyperparameter settings.

6.5 Conclusion

In this paper, we have explored multi-variable deep learning approaches for climate downscaling, specifically targeting the GCM-to-RCM emulation task. While most deep learning approaches focus on modeling single variables, we have argued that jointly processing multiple variables can improve both efficiency and predictive performance. Redundant modeling across variables can be avoided, and information from one variable can enhance the prediction of another by capturing inter-variable dependencies. To test this hypothesis, we designed three model configurations using a shared Vision Transformer (ViT)-based encoder-decoder architecture. These configurations explore different strategies for handling multiple variables, ranging from fully independent models to joint encoding with separate decoders. This progression allowed us to assess the trade-offs between variable separation and shared representation, culminating in our proposed Multi Var. ViT (1EMD) model, which combines a shared encoder with dedicated decoders for each variable. To rigorously evaluate our methods,

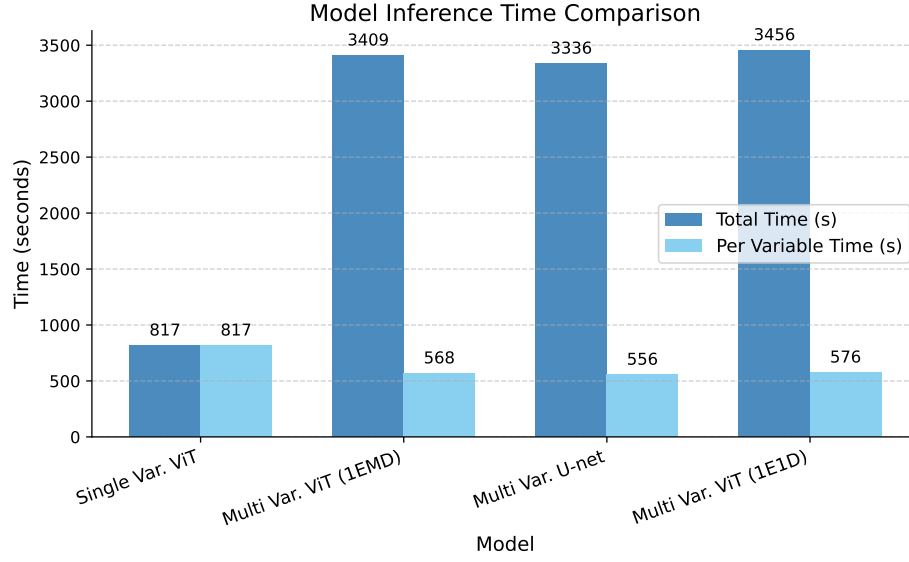


Figure 45: Inference time comparison across model architectures. The Total inference time (dark blue bars) and per-variable inference time (light blue bars) are shown for four model configurations. All models were evaluated on identical hardware (NVIDIA A100), with the same input data pipeline. The measurements correspond to the inference time required to process 10 years of daily testing data. For the multi-variable models, which predict six variables simultaneously, the per-variable inference time was calculated by dividing the total inference time the number of variables.

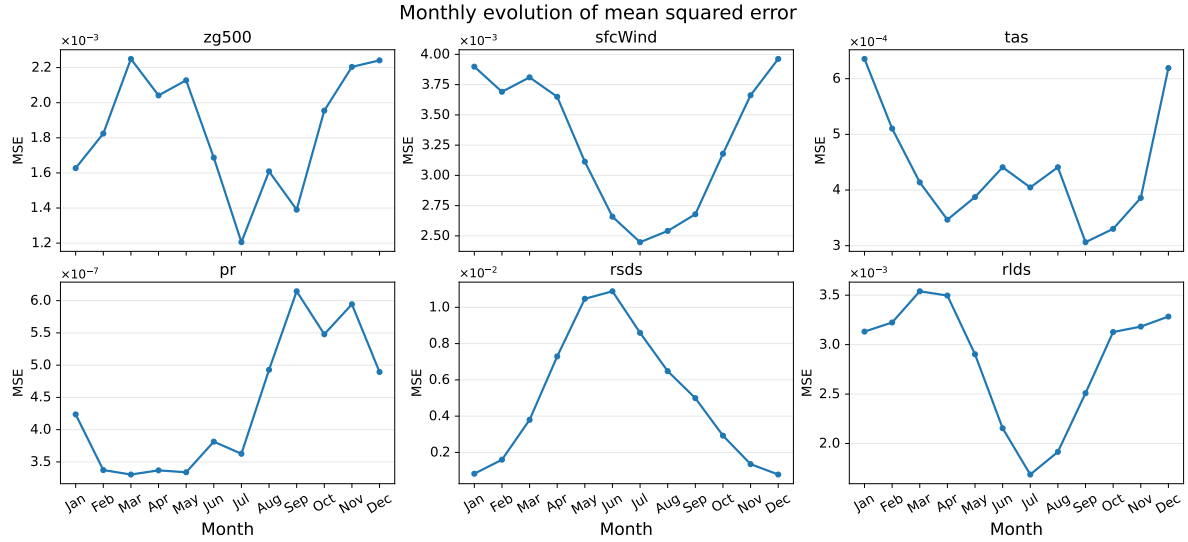


Figure 46: Monthly evolution of the mean squared error (MSE) for the six downscaled variables, computed over ten climatological years and averaged over all spatial grid points.

we established a consistent and fair comparison framework. All models share the same architecture, differing only in the components necessary to support multi-variable processing. Training hyperparameters, learning rate scheduling, pre-processing pipelines, and hardware resources were kept identical across all experiments. Our results, based on a 10-year test set of held-out data, demonstrate that the proposed Multi Var. ViT (1EMD) model consistently outperforms all other tested configurations. This suggests that combining shared representations with variable-specific decoders is an effective strategy for multi-variable downscaling. Overall, we have introduced an effective and straightforward method for climate downscaling in a multi-variable context. By modeling variables jointly, our results suggest a substantial gain in both performance and computational efficiency. Several open questions remain, including how many variables can be effectively modeled together, which variables should be selected to maximize performance, and whether more sophisticated loss functions could further enhance the model's capabilities.

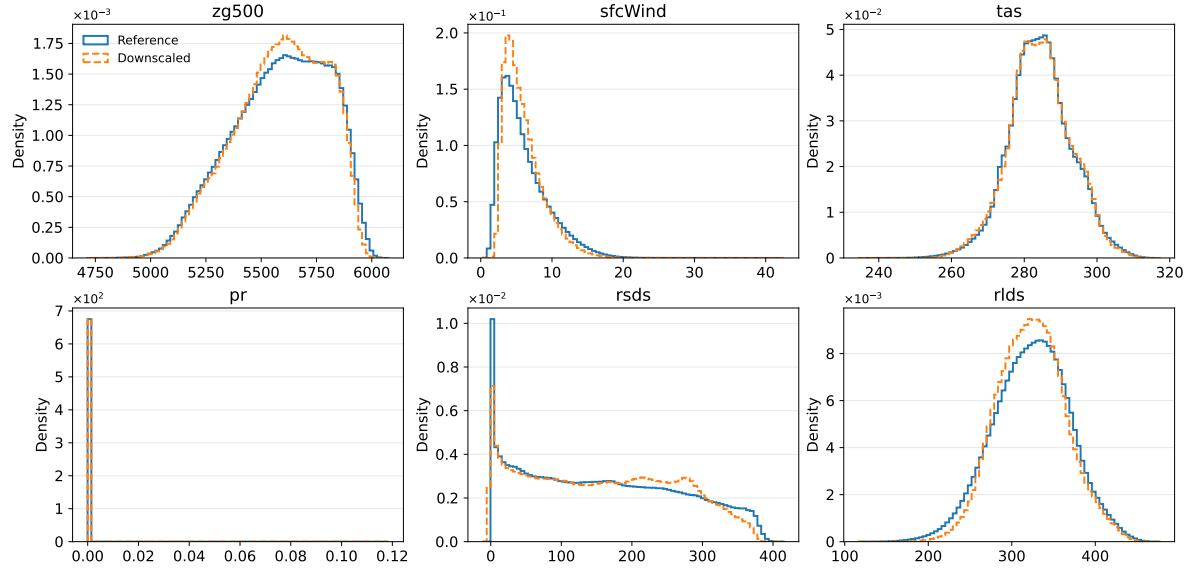


Figure 47: Empirical probability density functions (PDFs) of the six downscaled variables computed over the test set and aggregated across all spatial grid points. Histograms are normalized such that the total area equals one and are computed using common bin edges for the reference and downscaled datasets.

7 Synthesis and Discussion

This chapter synthesizes the main findings of the thesis and places them within the broader context of meteorological downscaling. It connects the results of the three studies, highlighting how methodological choices influence the model capabilities. The discussion begins by revisiting the main downscaling paradigms and examining how neural approaches extend beyond dynamical and statistical frameworks. It then compares the behavior of different neural architectures and explores two key themes that emerged from this work: the ability of generative models to infer uncertainty from deterministic data, and the trade-offs involved in multi-variable learning for efficient and scalable modeling.

7.1 Downscaling Paradigms: Dynamical, Statistical, and Neural

Downscaling methods refine coarse global climate information to regional scales. They are usually divided into two families: dynamical approaches, which resolve physical processes, and statistical approaches, which learn empirical relationships between large and local-scale variables.

Dynamical downscaling relies on regional or convection-permitting climate models nested within global drivers to explicitly solve the governing equations of the atmosphere at higher resolution. This approach is physically consistent and provides detailed process representations, but it comes at a very high computational cost and is typically limited in domain size, ensemble spread, or simulation length.

Statistical downscaling, in contrast, establishes empirical relationships between large-scale predictors and local climate variables. It is computationally efficient and widely used in climate services and bias correction pipelines, but its validity depends on the assumption that relationships derived from past data remain stationary under climate change, and on the availability of sufficiently long observational records.

Neural models are often grouped under the umbrella of statistical downscaling. Yet this classification is increasingly inadequate. Neural networks do more than replicate past relationships: they can generalize patterns to new conditions, approximate physical processes from data alone, and, when informed by physics-inspired architectures, embed explicit physical consistency into their outputs. In this sense, neural approaches share properties with both statistical and dynamical models, bridging the gap between them.

Thus, it may be argued that the traditional division between dynamical and statistical downscaling is no longer sufficient. Neural methods are carving out a new methodological category of their own, combining the efficiency of statistical models with the capacity to approximate and emulate physical systems in ways that bring them closer to dynamical downscaling.

7.2 Broader Perspectives on Neural Downscaling

Neural downscaling, much like data-driven forecasting, is currently framed as an emulation task. Models are trained to reproduce the behavior of existing numerical systems, relying on their outputs as training data. This dependency imposes a fundamental limitation: the achievable quality of neural downscaling is bounded by the fidelity of the underlying numerical models. In practice, neural approaches inevitably inherit the deficiencies and biases present in the datasets that generated their training targets. As a result, neural downscaling models are not yet positioned as complete replacements for traditional numerical models, but rather as complementary companions. They rely on physically based simulations to provide training data, yet they can efficiently emulate their outputs once trained. This symbiosis highlights an interesting dynamic: while neural methods can dramatically accelerate and simplify the production of high-resolution data, they still depend on physics-based models to define what “realistic” means in the first place.

From a computational standpoint, neural networks occupy an entirely different efficiency regime compared to traditional numerical models. Running high-resolution reanalyses or climate simulations remains extremely resource-intensive, with only a handful of research centers worldwide possessing the infrastructure and expertise to do so. In this context, neural approaches act as democratizing technologies, enabling high-resolution downscaling on inexpensive hardware. This shift is likely to lead, in the coming years, to a proliferation of deep learning-based downscaled datasets, broadening scientific access to fine-scale climate information.

In terms of performance, image-based neural architectures, inspired by tasks such as denoising, segmentation, and super-resolution, have proven remarkably effective for emulating downscaling trans-

formations. Our experiments demonstrate that, when properly trained, such models can reproduce the accuracy of the original high-resolution data when compared against in-situ observations.

A common concern lies in the dependence on historical training data. Since neural models are trained on a fixed time period, their performance may degrade when applied to conditions that diverge from the training distribution, for instance, due to long-term climate trends or changes in forcing scenarios. While this limitation is conceptually valid, it may be mitigated by conditioning the models on external predictors that encode slow-varying climate signals or scenario information. Moreover, the observed ability of neural networks to capture physically consistent relationships suggests that, with further development, they could eventually evolve into standalone downscaling systems requiring minimal reliance on traditional models.

Finally, an active line of research concerns the generalization capabilities of neural downscaling methods. This includes assessing their performance under domain shifts, such as transferring models across regions (spatial generalization), across time periods (temporal generalization), or across data sources (e.g., from reanalyses to climate simulations). These experiments are critical to understanding the true robustness of neural downscaling and will likely define the next stage of research in this emerging field.

7.3 Architectures for Neural Downscaling

In this thesis, a wide spectrum of neural architectures has been considered in order to cover the design space of deep-learning approaches to downscaling. Each family of models offers distinct advantages and limitations, reflecting different balances between efficiency, accuracy, and physical realism. Below, we summarize the main characteristics of these architectures and discuss their role in the experiments presented in this work.

MLPs. MLPs represent the earliest application of neural networks to downscaling and are included here primarily for their historical and methodological significance. While they do not scale well to large spatial domains and cannot capture spatial dependencies, they can still be practical in small or sparse settings where spatial context is of limited importance. Beyond serving as simple baselines, MLPs are also useful as diagnostic tools: by comparing their performance with spatially aware architectures, researchers can quickly assess the extent to which spatial information contributes to improvements in a given downscaling task.

CNNs. CNNs (e.g., ESPCNN, EDSR) provide satisfactory performance, particularly when compared with trivial baselines such as direct interpolation. They are computationally inexpensive, can be trained and run efficiently on modest hardware, and thus remain attractive in applications where cost and efficiency are the primary constraints. However, they are easily outperformed by more advanced architectures such as U-Nets and are rarely the method of choice when accuracy is the main objective. In this thesis, CNNs were therefore employed primarily as benchmarking baselines.

U-Nets. U-Nets have become the standard architecture for image-to-image downscaling tasks. They are versatile and generally robust, offering a strong balance between training stability, predictive performance, and computational efficiency. Their main limitation lies in a tendency to produce overly smoothed outputs, which can suppress high-frequency variability. In this thesis, we explored several U-Net variants, including residual U-Nets, depthwise convolutions, and squeeze-excite blocks. Across experiments, U-Nets consistently proved to be reliable baselines and effective backbones, confirming their role as a safe and widely applicable first choice for neural downscaling.

ViTs. Vision Transformers (ViTs) are increasingly applied to imaging tasks and offer a range of formulations, including hybrid convolution-transformer designs and sliding-window approaches. They often achieve performance comparable to U-Nets while providing lower inductive bias and greater flexibility, particularly in capturing long-range dependencies and multi-variable relationships. In this thesis, ViTs were used both as benchmarks and as the backbone for our multi-variable downscaling architecture. By adopting a skip-connection-free design, we were able to extend the model into a

straightforward and effective multi-task framework, demonstrating strong performance and computational efficiency. These results suggest that ViTs may emerge as the next standard for deterministic neural downscaling.

GANs. Generative Adversarial Networks (GANs) represented the state of the art in non-deterministic generation prior to the emergence of diffusion models. They remain attractive for their efficiency, which often surpasses diffusion approaches at inference, but their practical use has been hindered by well-known issues of training instability and susceptibility to artifacts. As a result, GANs have largely been displaced by diffusion models for ensemble downscaling. Nevertheless, they continue to serve as valuable benchmarks, particularly given the limited number of non-deterministic neural methods available. In this thesis, we employed GANs primarily in this comparative role, using a conditioned variant of DC-GAN.

Diffusion Models. Diffusion models are among the most advanced and promising architectures in the downscaling landscape. Their mathematical foundations are relatively complex, and many aspects of their optimal configuration (such as noise scheduling, normalization, and conditioning) remain the subject of ongoing research. Despite this, they have surpassed GANs in terms of stability and reliability, offering predictable training dynamics while enabling high-fidelity generation across a wide range of domains. In this thesis, we applied diffusion models in multiple ways: first in a deterministic setting, where averaging across multiple samples yielded results that outperformed competing methods, although at higher computational cost; and subsequently in a non-deterministic ensemble framework, where we introduced a theoretical basis for variance tuning. Overall, diffusion models emerge as powerful, modern architectures that set the current state of the art for uncertainty-aware downscaling.

Model type	Key characteristics	Strengths	Limitations	Typical use cases
MLPs	Fully connected, non-spatial	Simple, fast, easy to train	Cannot capture spatial structure, bound to small size	Early baselines; conceptual benchmark, non-spatial circumstances
CNNs	Convolutional filters, local receptive fields	Efficient, stable, capture spatial patterns	Limited long-range dependency modeling; oversmoothing	Largely superseded by U-Net designs, only valuable for high efficiency applications
U-Nets	Encoder-decoder with skip connections	Versatile, combines ease of training with reliable performance and efficiency	May blur high-frequency details, deterministic modelling	Widely used, the standard for neural downscaling
GANs	Generator-discriminator training with adversarial loss	Generates sharp details; non-deterministic model	Training instability, no guarantee of convergence; risk of artifacts	Rarely used, high efficiency non-deterministic generation
ViTs	Self-attention across image patches	Captures long-range dependencies; better suited for multi-variable inputs	Weaker architectural priors than U-Nets, often leading to reduced training stability	Promising for multi-variable downscaling; strong state-of-the-art results
Diffusion Models	Iterative denoising, probabilistic formulation	Stable training; controllable ensembles uncertainty	Computationally costly at inference; still new in climate	Suitable for ensemble generation and uncertainty-aware downscaling

Table 22: Overview of main neural architectures applied to meteorological downscaling, with key characteristics, strengths, limitations, and typical use cases.

7.4 Learning Uncertainty from Deterministic Data

A particularly intriguing outcome of this work lies in the behavior of the ensemble diffusion model. As reported in Section 5, despite being trained solely on deterministic CERRA data and without access to CERRA-EDA or any explicit information on spatial or temporal uncertainty, the model successfully learns to generate ensemble members that exhibit realistic variability. In doing so, it infers the underlying structure of uncertainty directly from the data itself, rather than learning it from predefined examples.

This result challenges the conventional view of neural models as simple emulators of existing numerical models. Instead, it demonstrates that generative diffusion models are capable of *creating* new information, capturing statistical properties that are not explicitly encoded in the training data.

One of the most remarkable findings of this thesis is that the variance produced by the diffusion model can be quantitatively aligned with reference ensemble datasets such as CERRA-EDA, despite never having seen them during training. This confirms that the uncertainty expressed by the generated ensembles is physically meaningful and statistically consistent with observed variability.

More broadly, this capability may be interpreted as a form of generalization beyond the deterministic domain. The model’s ability to infer uncertainty from single-sample time series raises interesting questions: if a neural system can infer ensemble statistics from deterministic data, could similar mechanisms be used to learn domain or temporal generalization? This perspective opens new research directions at the intersection of generative modeling and physical uncertainty representation.

In conclusion, ensemble generation emerges as one of the most promising and impactful applications of neural downscaling models. Recent research has begun to recognize and exploit these capabilities, suggesting that diffusion-based approaches could play a central role in the next generation of data-driven ensemble systems.

7.5 The Trade-offs of Multi-Variable Approaches

Multi-variable approaches likely represent the next development in neural downscaling. Modeling each meteorological variable as an isolated field introduces inconsistencies among variables, as independently trained models may diverge in their spatial structures. This separation offers no guarantee of overall physical continuity or inter-variable coherence, two essential aspects for realistic downscaled datasets. It is therefore difficult to imagine the future of data-driven downscaling continuing to rely on entirely independent models, each responsible for a single variable, as has been common practice until now. Multi-variable multi-task frameworks appear to be a natural evolution toward more integrated and physically consistent modeling. However, agreeing on the need for multi-tasking is only the beginning; implementing it effectively is a far more complex challenge. Multi-variable training introduces a range of practical and theoretical difficulties. Larger models require higher computational costs, and their unified structure removes the compartmentalization that makes single-variable models straightforward to design and optimize. A multi-task model must perform satisfactorily across all variables simultaneously, which complicates loss design, increases training instability, and often impedes fine-tuning for individual variables. Moreover, such models typically require all input variables to be available at once, reducing operational flexibility, although this constraint can be relaxed with specific architectural formulations. Despite these drawbacks, multi-variable modeling remains a compelling direction. The key to success lies in maximizing positive transfer (the beneficial exchange of shared information among variables) while minimizing negative transfer, where interference between variables degrades performance. In this thesis, we addressed these challenges through a shared-encoder, multi-decoder architecture. The encoder jointly processes all variables, learning a shared latent representation that captures their interdependencies, while separate decoders reconstruct each variable individually at high resolution. This design leverages shared information during encoding without forcing the variables to interfere in the decoding phase. To assess the benefits of multi-tasking objectively, we conducted controlled experiments ensuring a fair comparison between single-variable and multi-variable models. Architectures, hyperparameters, and computational resources were carefully aligned across experiments. The results clearly indicate that multi-variable learning yields measurable performance improvements, confirming that cross-variable information can indeed be exploited effectively. In conclusion, multi-variable approaches are both conceptually natural and empirically beneficial. While their implementation remains challenging, they represent a promising foundation for future neural downscaling frameworks. The proposed shared-encoder, multi-decoder structure offers a

practical backbone upon which more sophisticated, physically informed designs can be developed.

8 Conclusions and Future Work

This chapter concludes the thesis by summarizing its main contributions and outlining research directions. It consolidates the results of the three studies, highlighting their methodological advances and collective impact on neural meteorological downscaling. The first part provides an overview of the contributions and limitations of each paper, situating them within the broader context of data-driven downscaling. The second part discusses the implications of these findings for the field and identifies the emerging role of diffusion-based and multi-variable architectures. Finally, the chapter presents future research opportunities aimed at expanding and refining the proposed methodologies toward more scalable, stable and effective neural frameworks.

8.1 Summary of the Thesis Contributions

This thesis has sought to advance the state of the art in meteorological downscaling using neural techniques. Alongside providing a general overview of the field, including a historiographic perspective and a proposed design space of downscaling architectures, the thesis presents three original research papers. Each addresses a distinct challenge in data-driven downscaling: the first applies and benchmarks diffusion models for super-resolution downscaling of wind speed, introducing an ensemble formulation that outperforms competing approaches; the second develops a theoretical framework for variance control in diffusion ensembles and demonstrates its practical use for generating calibrated ensemble members; and the third explores multi-variable, multi-task Vision Transformer architectures, showing that under fair comparison conditions, multi-variable modeling yields consistent advantages over single-variable alternatives. Beyond these individual contributions, the thesis also offers a broader discussion of the field, argues for methodological shifts in perspective, and provides insights on current limitations and future research directions. The following sections summarize in detail the specific contributions of each paper.

8.1.1 Paper I

This paper [4] demonstrated the application of diffusion models for meteorological downscaling, focusing on wind speed over the Italian peninsula. In addition to benchmarking multiple neural architectures, the study introduced an ensemble diffusion formulation and evaluated its performance against both reanalysis products and ground observations.

- Applied state-of-the-art super-resolution diffusion models to downscale wind speed from ERA5 to CERRA in a fully data-driven manner, without requiring supplementary data, and designed experiments centered on the Mediterranean region around Italy.
- Trained a diverse set of models for comparison, including ESPCNN, EDSR, SwinLSTM, SRGAN, U-Net, ViT, and both Single and Ensemble Diffusion.
- Proposed an Ensemble Diffusion formulation where multiple generations are combined via the mean to provide more accurate downscaling.
- Showed that even simpler neural super-resolution models consistently outperform classical interpolation techniques such as bilinear upscaling.
- Demonstrated strong alignment between data-driven downscaling and traditional physics-based reanalyses, validated both by similarity to CERRA outputs and by comparison with in-situ ground station observations.
- Identified the Ensemble Diffusion with mean output as the most accurate among the tested methods, yielding outputs of higher quality than ERA5 and more closely aligned with real-world measurements.
- Highlighted the advantage of neural models in addressing the long delays of physics-based reanalysis products (e.g., CERRA lagging over two years at the time of the study), since only ERA5 is required to generate timely outputs.

- Established the computational efficiency of neural models relative to traditional physics-based approaches.
- Exploited the hourly availability of ERA5 to augment temporal resolution, producing hourly downscaled outputs compared to CERRA’s 3-hour resolution. We not only can generate crucial missing data, but we can do it with higher temporal resolution.

8.1.2 Paper II

This paper [5] advanced the theoretical understanding of ensemble diffusion by formally linking variance to the number of reverse diffusion steps. Building on this insight, the study introduced a practical variance calibration strategy and architectural improvements to scale diffusion models across the full ERA5-to-CERRA domain. The approach was validated against the CERRA-EDA ensemble and extended to CARRA-EAST, positioning ensemble diffusion as both a tool for efficient ensemble generation and a framework for diagnosing limitations in existing ensemble datasets.

- Provided a formal proof of the relationship between ensemble variance and the number of reverse diffusion steps, establishing the reverse step size Δt as both an efficiency factor and the key mechanism for variance control.
- Highlighted the overlooked effect of variance convergence at high diffusion steps, a common assumption in prior work, and showed that optimal results are achieved well before convergence (at around 8 steps).
- Applied an ensemble diffusion model to the full-domain ERA5-to-CERRA downscaling task at high spatial and temporal resolution, introducing architectural improvements to scale diffusion models across large domains, including a scaling factor in the noise scheduler to mitigate denoising difficulty.
- Demonstrated that variance tuning via step count enables the model to match the global mean variance and mean variance discrepancy of CERRA-EDA, while also producing efficient ensembles at full CERRA resolution (3h temporal extended to 1h, and full spatial coverage), outperforming the reduced-resolution CERRA-EDA and remaining feasible on workstation hardware.
- Extended the ensemble diffusion methodology from CERRA to CARRA-EAST, confirming robustness and applicability to datasets without existing ensemble information.
- Positioned the method as a validation tool for existing ensembles, with discrepancies offering insights into model limitations or potential weaknesses in ensemble datasets, and identified regional discrepancies (e.g., North Africa in summer) likely due to either modeling challenges or data scarcity.

8.1.3 Paper III

This paper [6] investigated multi-variable multi-task climate downscaling using Vision Transformers, introducing a shared-encoder, multi-decoder architecture (1EMD) to emulate Regional Climate Models from coarse GCM inputs. By jointly modeling surface temperature, wind speed, 500 hPa geopotential height, total precipitation, surface downwelling shortwave radiation, and surface downwelling longwave radiation, the study evaluated the benefits and challenges of multi-variable processing, demonstrating improved accuracy, efficiency, and cross-variable consistency compared to single-variable and simpler multi-variable alternatives.

- Proposed a multi-variable Vision Transformer (ViT) architecture for climate downscaling, using a shared encoder with variable-specific decoders (1EMD), designed to emulate Regional Climate Models (RCMs) from coarse Global Climate Model (GCM) inputs.
- Focused on six diverse climate variables over the European domain: surface air temperature (tas), surface wind speed (sfcWind), 500 hPa geopotential height (zg500), total precipitation (tp), surface downwelling shortwave radiation (rsds), and surface downwelling longwave radiation (rls), capturing both smooth and high-frequency spatial structures.

- Established single-variable ViT baselines and introduced two multi-variable transformer variants: 1E1D (shared encoder-decoder) and 1EMD (shared encoder with dedicated decoders), enabling a controlled and reproducible comparison under identical training conditions. In addition, a multi-variable U-Net was included as a widely adopted convolutional baseline, providing architectural diversity in the evaluation.
- Demonstrated that both the simpler 1E1D transformer and the multi-variable U-Net suffer from cross-variable interference, including visible artifacts and degraded spatial and temporal correlations for precipitation and geopotential height. These findings highlight the risks of treating heterogeneous physical variables as independent image channels within a fully shared decoding pipeline.
- Showed that the proposed 1EMD model mitigates interference while exploiting shared spatial representations, consistently outperforming single-variable ViTs, 1E1D, and multi-variable U-Net baselines across all six variables, with particularly strong gains for wind speed and precipitation.
- Quantified the computational benefits of joint modeling, achieving approximately 29–32% lower inference time per variable compared to executing six independent single-variable models, while also reducing overall training cost.
- Positioned multi-variable ViTs as an effective and scalable framework for high-resolution GCM-to-RCM emulation, demonstrating gains in both accuracy and efficiency while highlighting open questions on scalability to larger variable sets and more complex physical constraints.

8.2 Limitations

This section summarizes the main limitations emerging from this thesis. The discussion is organized in three layers. First, general limitations that affect the entire field of neural downscaling are outlined, providing a contextual framework for interpreting the results presented in this thesis. The following two subsections then focus on specific methodological and empirical constraints encountered in the ensemble diffusion and multi-variable vision transformer frameworks developed in this work. This distinction allows for a clear separation between the structural challenges of neural downscaling as a discipline and the more concrete limitations specific to the implementations proposed here.

General Limitations of Neural Downscaling

- **Dependence on physics-driven datasets:** Neural downscaling currently depends on training data generated by numerical, physics-based models (e.g., reanalyses or RCMs). Decoupling from these sources, by learning directly from observations or self-consistent physical constraints, remains an open and largely unresolved challenge. This dependency limits the independence of neural models and risks propagating existing model biases.
- **Stationarity assumption:** Most models implicitly assume that statistical relationships between coarse and fine scales remain stationary across time, space, and climate regimes. This assumption may break under climate change, where physical processes can shift significantly.
- **Neglect of temporal dynamics:** Most neural downscaling models operate on temporally independent samples, ignoring the sequential and dynamical nature of the climate system. This can limit their ability to represent persistence, feedbacks, and compound or extreme events.
- **Lack of physical interpretability:** Neural networks often function as black boxes, providing accurate predictions but limited insight into the underlying physical mechanisms. This hampers scientific understanding, model trust, and adoption in regulatory or decision-making contexts.
- **Limited physical consistency:** Purely data-driven models may violate conservation laws (e.g., mass, energy, water), physical balances (e.g., hydrostatic or geostrophic), or thermodynamic relationships, leading to physically implausible outputs even when statistical scores are high.

- **Limited transferability:** Neural downscaling models are often highly sensitive to the resolution, region, and temporal domain they were trained on. This hinders their generalization and forces repetitive retraining for similar tasks across datasets, variables, or domains. Such redundancy leads to wasteful computational expenditure and hinders scalability.
- **Under-representation of global south:** Relying on existing dataset, like reanalyses and simulations, reflects clearly the fact that most available data, especially high resolution data, is located in europe, east asia and north america. This creates an under representation of the global south, which translates to uneven performance from neural models.
- **Uncertainty representation:** As of now, most operational neural models are deterministic, producing a single best estimate rather than an ensemble or probability distribution.
- **Evaluation bias toward image-based metrics:** Current assessment practices rely heavily on statistical or perceptual similarity measures (e.g., MSE, SSIM), which may not reflect physically meaningful accuracy in terms of extremes, trends, or conservation properties.
- **Lack of fair comparison:** Ensuring fairness in model comparison is often as demanding as developing new models themselves. Differences in tuning, computational requirements, and implementation details can heavily bias results, making many reported comparisons only partially valid.

Ensemble Diffusion Limitations

- **Single-variable focus:** Our ensemble diffusion approaches are applied only to wind speed, leaving open its generalization to other variables with different spatial statistics.
- **Step-count calibration sensitivity:** The optimal number of diffusion steps varies by region and season, suggesting that a universal calibration rule may be infeasible without further theoretical insight.
- **Computational cost:** Ensemble diffusion models require multiple reverse diffusion steps for each generated sample, and multiple such samples to form an ensemble. This iterative process results in substantially higher computational demands compared to alternative downscaling approaches.
- **Theoretical limitation:** Although variance tuning is a valuable feature, the lack of convergence toward the expected variance with increasing diffusion steps is not yet theoretically understood.

Multi-Variable and Multi-Task Limitations

- **Variable selection scope:** The study focuses on six key variables (tas, sfcWind, zg500, tp, rsds, rlds), providing a diverse but limited representation of atmospheric processes. Expanding the approach to include additional variables would further test its generality.
- **Fixed model architecture:** Only a ViT backbone was explored. Other multi-task architectures (e.g. Swin Transformer) may obtain different results.
- **Simplified training loss:** The uniform MSE loss treats all variables equally, overlooking variable-specific dynamics and potential physical interdependencies.

8.3 Broader Impact on Neural Downscaling

The research presented in this thesis has had a considerable impact on the field of neural downscaling. In our first paper, we proposed one of the earliest applications of ensemble diffusion to meteorological downscaling, showing that a mean-ensemble formulation can outperform competing approaches. This finding sparked increased interest in diffusion-based techniques for downscaling [LGG24, TFC25, TXY⁺25, SHP⁺24, ZXYW25], which have since been extended to a variety of domains. In our second paper, we pioneered variance control for ensemble diffusion calibration, enabling the matching of generated uncertainty to existing reference ensembles. This development provided

the research community with a valuable tool for tuning generative variance as needed. Notably, the previous paradigm in diffusion modeling assumed that the generative variance converged to that of the training dataset. Our work refined this understanding by showing that diffusion variance instead tends toward a fixed value (potentially distinct from the target) and that it can be systematically adjusted through the number of diffusion steps. While the broader implications of this formulation are still being investigated, it has already inspired several studies exploring diffusion-powered approaches to uncertainty calibration. Finally, our multi-variable, multi-task formulation revealed that, under fair and consistent comparisons, jointly modeling several meteorological variables allows the model to exploit inter-variable interactions and outperform single-variable alternatives. The core innovation lies in a simple yet effective architectural modification: instead of training a separate encoder–decoder model for each variable, a single shared encoder can be coupled with multiple decoders, one per variable. This adjustment requires minimal changes to existing encoder–decoder pipelines, making it straightforward for researchers already using such architectures to experiment with multi-variable learning. Despite its simplicity, the approach consistently yields performance gains and offers a practical entry point for testing joint modeling without added complexity. It is therefore foreseeable that future work in neural downscaling will increasingly adopt such multi-variable extensions, building upon the methodological foundation established here.

8.4 Future Work and Developments

Building on the methods and findings presented in this thesis, numerous research directions emerge, both within the proposed frameworks and toward broader methodological advancements.

Variance Calibrated Ensembles

Several open questions remain regarding the variance-controlled ensemble diffusion framework:

- **Extension to other variables.** The present work focused primarily on wind speed. It remains to be explored whether variables with smoother spatial behavior (e.g., temperature, geopotential height) or highly intermittent distributions (e.g., precipitation) exhibit similar variance dynamics. Preliminary experiments suggest so, but systematic validation is needed.
- **Robustness across domains.** Our analysis concentrated on the ERA5–CERRA domain. Future experiments could assess whether the same behavior holds in other geographical or climatological contexts. Further, datasets such as ERA5–Ensemble offer valuable opportunities for expanded validation, including on extreme events and the tails of uncertainty distributions.
- **Extension to forecasts and simulations.** The methodology could be tested beyond reanalyses, for instance on operational forecasting systems or long-term climate simulations, to evaluate whether automated ensemble generation maintains its effectiveness across data regimes and temporal scales.
- **Architectural refinements.** The diffusion model design itself is not definitive. Potential improvements include the use of depthwise separable convolutions or transformer-based denoisers, as well as the systematic exploration of reverse diffusion schedules and their effect on variance dynamics and image quality.
- **Linking variance to physical uncertainty decomposition.** Current work calibrates variance empirically, but it could be insightful to study whether physical sources of uncertainty (model error, initial conditions, boundary conditions) map in identifiable ways to the diffusion variance. This could bridge machine-learning uncertainty with classical ensemble meteorology.
- **Uncertainty as diagnostic.** The proposed calibration method could serve as a diagnostic tool for datasets lacking explicit uncertainty information, helping identify regions of high variance and inconsistencies in data collection or model formulation.
- **Consistency and distillation models.** Recent “consistency models” offer a promising avenue to distill multi-step diffusion processes into single-step approximations, dramatically improving efficiency. Their applicability to meteorological downscaling remains untested but highly relevant.

- **Latent diffusion.** Stable diffusion employs a variational autoencoder to perform diffusion in latent space, trading fidelity for efficiency. While meteorological downscaling prioritizes accuracy over variability, it remains worthwhile to experimentally verify the limits of this approach.

Multi-Task Downscaling

Future developments can also enhance and generalize the proposed multi-variable formulation:

- **Integration with ensemble diffusion.** A natural next step would be to integrate the multi-task architecture as a denoiser within the ensemble diffusion framework, producing multi-variable, internally consistent ensemble members.
- **Advanced loss design.** The initial formulation used a simple mean of normalized losses. More complex variable combinations will require carefully designed, weighted or physically informed loss functions to ensure balanced optimization across variables.
- **Physical constraints.** Incorporating soft or hard physical constraints could improve operational reliability, ensuring that the model adheres to conservation laws and dynamic consistency, especially under extreme conditions.
- **Architectural extensions.** The current 1EMD model deliberately excludes skip connections to prevent variable interference. Future work could explore ways to safely reintegrate skip connections or adopt transformer-based decoders (e.g., Swin architectures) to further enhance performance.
- **Scalability to many variables.** The current experiments used six representative variables. Expanding to larger sets could reveal how performance scales and where multi-task learning begins to saturate or degrade.

Broader Research Directions

Finally, several cross-cutting research opportunities are worth exploring:

- **Cross-domain interoperability.** A key question is whether models trained on one dataset type (e.g., high-resolution reanalyses) can generalize or be adapted to others (e.g., operational forecasts) through transfer learning or distribution alignment.
- **Toward foundational models for downscaling.** With growing interest in foundational models for meteorology, future work may investigate whether a general-purpose backbone trained on diverse datasets can be fine-tuned for specific downscaling tasks.
- **Domain adaptation under climate change.** Study whether models trained on historical data generalize under future climate distributions. Initial experiments reveal surprising results, as climate change induced smoothing may actually improve the quality of predictions.
- **Operational integration.** Bridging the gap between experimental and operational settings remains a major challenge. Future studies should address model deployment, optimization for cloud environments, and data management pipelines to translate neural downscaling into viable alternatives to traditional numerical methods.
- **Energy and carbon footprint.** Evaluate how ensemble diffusion can be made energy-efficient through distillation or quantization, given the high computational costs of inference, especially for diffusion models.
- **Additional evaluation metrics** Error metrics such as MSE, RMSE, and SSIM provide a solid indication of the relative performance of the neural models. However, assessing their applicability to real-world climate applications requires additional diagnostics, including regime-dependent analyses (e.g., North Atlantic Oscillation phases), percentile-based metrics, and quantile–quantile plots, among others.

In closing, I hope this thesis has effectively conveyed that deep learning for meteorological downscaling is poised to reshape our approaches to climate modeling in ways that are not yet fully understood. Throughout this thesis, we explored the vast design space of deep learning and showed that it can not only emulate existing climate models but also transcend them, for example by learning uncertainty from deterministic data. I hope that this work will stand as a small step toward extending the capabilities of neural downscaling and contributing to a more comprehensive understanding of Earth's climate. A knowledge that, I hope, will ultimately serve to safeguard and appreciate our beautiful planet.

Acknowledgments

I thank all my colleagues, especially Salvatore and Davide.

I thank my supervisor, Andrea.

I thank the masters of language, human and not, that participated in the interesting dialogue that is science.

I thank all the inhabitants of Casale. I thank my friends, my family, my love.

References

- [ABB⁺19] Shreya Agrawal, Luke Barrington, Carla Bromberg, John Burge, Cenk Gazen, and Jason Hickey. Machine learning for precipitation nowcasting from radar images. *arXiv preprint arXiv:1912.12132*, 2019.
- [AEMM23] Andrea Asperti, Davide Evangelista, Samuele Marro, and Fabio Merizzi. Image embedding for denoising generative models. *Artificial Intelligence Review*, 56(12):14511–14533, 2023.
- [AHP⁺24] Michael Aich, Philipp Hess, Baoxiang Pan, Sebastian Bathiany, Yu Huang, and Niklas Boers. Conditional diffusion models for downscaling & bias correction of earth system model precipitation. *arXiv preprint arXiv:2404.14416*, 2024.
- [AKB20] Saeed Anwar, Salman Khan, and Nick Barnes. A deep journey into super-resolution: A survey. *ACM Comput. Surv.*, 53(3), may 2020.
- [AKS18] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *Proceedings of the European conference on computer vision (ECCV)*, pages 252–268, 2018.
- [ALDC25] Maryam Alipourhajiagha, Pierre-Louis Lemaire, Youssef Diouane, and Julie Carreau. A probabilistic u-net approach to downscaling climate simulations. *arXiv preprint arXiv:2511.03197*, 2025.
- [AMP⁺23] Andrea Asperti, Fabio Merizzi, Alberto Paparella, Giorgio Pedrazzi, Matteo Angelinelli, and Stefano Colamonaco. Precipitation nowcasting with generative diffusion models. *arXiv preprint arXiv:2308.06733*, 2023.
- [AMP⁺25] Andrea Asperti, Fabio Merizzi, Alberto Paparella, Giorgio Pedrazzi, Matteo Angelinelli, and Stefano Colamonaco. Precipitation nowcasting with generative diffusion models. *Applied Intelligence*, 55(2):1–21, 2025.
- [ASB⁺19] W. B. Anderson, R. Seager, W. Baethgen, M. Cane, and L. You. Synchronous crop failures and climate-forced production variability. *Science Advances*, 5(7):eaaw1976, 2019.
- [ASGA12] Kumar Abhishek, Maheshwari Prasad Singh, Saswata Ghosh, and Abhishek Anand. Weather forecasting model using artificial neural network. *Procedia Technology*, 4:311–318, 2012.
- [ASJF⁺24] Ali Salem Al-Sakkaf, Zhang Jiahua, Yao Fengmei, Mohammed Magdy Hamed, Claudien Habimana Simbi, Arslan Ahmed, and Shamsuddin Shahid. Assessing exposure to climate extremes over the arabian peninsula using era5 reanalysis data: Spatial distribution and temporal trends. *Atmospheric Research*, page 107224, 2024.
- [BAA⁺17] Lisa Bengtsson, Ulf Andrae, Trygve Aspelien, Yurii Batrak, Javier Calvo, Wim de Rooy, Emily Gleeson, Bent Hansen-Sass, Mariken Homleid, Mariano Hortal, et al. The harmonie–arome model configuration in the aladin–hirlam nwp system. *Monthly Weather Review*, 145(5):1919–1935, 2017.

- [BKM⁺25] Boris Bonev, Thorsten Kurth, Ankur Mahesh, Mauro Bisson, Jean Kossaifi, Karthik Kashinath, Anima Anandkumar, William D Collins, Michael S Pritchard, and Alexander Keller. Fourcastnet 3: A geometric approach to probabilistic machine-learning weather forecasting at scale. *arXiv preprint arXiv:2507.12144*, 2025.
- [BL16] Sándor Baran and Sebastian Lerch. Mixture emos model for calibrating ensemble forecasts of wind speed. *Environmetrics*, 27(2):116–130, 2016.
- [BM18] Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6228–6237, 2018.
- [BMMG20] Jorge Baño-Medina, Rodrigo Manzananas, and José Manuel Gutiérrez. Configuration and intercomparison of deep learning neural models for statistical downscaling. *Geoscientific Model Development*, 13(4):2109–2124, 2020.
- [BMMG21] Jorge Baño-Medina, Rodrigo Manzananas, and José Manuel Gutiérrez. On the suitability of deep convolutional neural networks for continental-wide downscaling of climate change projections. *Climate Dynamics*, 57(11):2941–2951, 2021.
- [Bod24] Cristian Bodnar. Aurora: A foundation model of the atmosphere. In *AGU Fall Meeting Abstracts*, volume 2024, pages GC21C–03, 2024.
- [BS10] S Santhosh Baboo and I Kadar Shereef. An efficient weather forecasting system using artificial neural network. *International journal of environmental science and development*, 1(4):321, 2010.
- [BSC⁺16] V. Banzon, T. M. Smith, T. M. Chin, C. Liu, and W. Hankins. A long-term record of blended satellite and *in situ* sea-surface temperature for climate monitoring, modeling and environmental studies. *Earth System Science Data*, 8:165–176, 2016.
- [BTB15] Peter Bauer, Alan Thorpe, and Gilbert Brunet. The quiet revolution of numerical weather prediction. *Nature*, 525(7567):47–55, 2015.
- [BWKN22] Syed Muhammad Arsalan Bashir, Yi Wang, Mahrukh Khan, and Yilong Niu. A comprehensive review of deep learning-based single image super-resolution, 2022.
- [BXU⁺23] Takumi Bannai, Haoyang Xu, Nobuyuki Utsumi, Eunho Koo, Keming Lu, and Hyungjun Kim. Multi-task learning for simultaneous retrievals of passive microwave precipitation estimates and rain/no-rain classification. *Geophysical Research Letters*, 50(7):e2022GL102283, 2023.
- [BXZ⁺22] Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Pangu-weather: A 3d high-resolution model for fast and accurate global weather forecast. *CoRR*, abs/2211.02556, 2022.
- [CAB00] Paulin Coulibaly, François Anctil, and Bernard Bobée. Daily reservoir inflow forecasting using artificial neural networks with stopped training approach. *Journal of Hydrology*, 230(3-4):244–257, 2000.
- [CAU⁺16] Met Office Hadley Centre, Irish Climate Analysis, Research Units, National Center for Atmospheric Research, and National Centers for Environmental Information NOAA. Hadisd: global sub-daily station data for climate extremes. Centre for Environmental Data Analysis, 2016.
- [CB21] Yiling Cai and François-Marie Bréon. Wind power potential and intermittency issues in the context of climate change. *Energy Conversion and Management*, 240:114276, 2021.
- [CCBT22] Marcos V Conde, Ui-Jin Choi, Maxime Burchi, and Radu Timofte. Swin2sr: Swinv2 transformer for compressed image super-resolution and restoration. In *European Conference on Computer Vision*, pages 669–687. Springer, 2022.

- [CCC22] Francesco Chiaravalloti, Tommaso Caloiero, and Roberto Coscarelli. The long-term era5 data series for trend analysis of rainfall in italy. *Hydrology*, 9(2):18, 2022.
- [CCDP21] Matthew Chantry, Hannah Christensen, Peter Dueben, and Tim Palmer. Opportunities and challenges for machine learning in weather and climate modelling: hard, medium and soft ai. *Philosophical Transactions of the Royal Society A*, 379(2194):20200083, 2021.
- [CCG11] R Chadwick, E Coppola, and F Giorgi. An artificial neural network technique for downscaling gcm outputs to rcm spatial scale. *Nonlinear Processes in Geophysics*, 18(6):1013–1028, 2011.
- [CCRA07] Jens H Christensen, Timothy R Carter, Markku Rummukainen, and Georgios Amanatidis. Evaluating the performance and utility of regional climate models: the prudence project. *Climatic Change*, 81(Suppl 1):1–6, 2007.
- [CDA05] Paulin Coulibaly, Yonas B Dibike, and François Anctil. Downscaling precipitation and temperature with temporal neural networks. *Journal of Hydrometeorology*, 6(4):483–496, 2005.
- [CDC⁺07] Ole Bøssing Christensen, Martin Drews, Jens Hesselbjerg Christensen, Klaus Dethloff, Klaus Ketelsen, Ines Hebestadt, and Anette Rinke. The hirham regional climate model. version 5 (beta). 2007.
- [CDCG⁺22] Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. Scientific machine learning through physics-informed neural networks: Where we are and what’s next. *Journal of Scientific Computing*, 92(3):88, 2022.
- [CFL⁺22] Xuanhong Chen, Kairui Feng, Naiyuan Liu, Bingbing Ni, Yifan Lu, Zhengyan Tong, and Ziang Liu. Rainnet: A large-scale imagery dataset and benchmark for spatial precipitation downscaling. *Advances in Neural Information Processing Systems*, 35:9797–9812, 2022.
- [Che23] Ting Chen. On the importance of noise scheduling for diffusion models. *arXiv preprint arXiv:2301.10972*, 2023.
- [CK17] Jae-Seok Choi and Munchurl Kim. A deep convolutional neural network with selection units for super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 154–160, 2017.
- [CLY⁺21] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*, 2021.
- [CLZVG21] Jiezhong Cao, Yawei Li, Kai Zhang, and Luc Van Gool. Video super-resolution transformer. *arXiv preprint arXiv:2106.06847*, 2021.
- [Cop23] Copernicus Climate Change Service (C3S). ERA5 hourly data on single levels from 1940 to present, 2023. Accessed on 22-11-2023.
- [Cop24] Copernicus Climate Change Service. Copernicus arctic regional reanalysis service, 2024. Accessed: 2024-12-11.
- [CR20] C. Chaudhuri and Colin Robertson. Cligan: A structurally sensitive convolutional neural network model for statistical downscaling of precipitation from multi-model ensembles. *Water*, 2020.
- [Cra97] Arthur P Cracknell. *Advanced very high resolution radiometer AVHRR*. Crc Press, 1997.
- [Cra20] Michael Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020.

- [CSX⁺08] M. Chen, W. Shi, P. Xie, V. B. S. Silva, V. E. Kousky, R. W. Higgins, and J. E. Janowiak. Assessing objective techniques for gauge-based analyses of global daily precipitation. *Journal of Geophysical Research: Atmospheres*, 113:D04110, 2008.
- [CWG⁺21] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12299–12310, 2021.
- [CZZ⁺23] Lei Chen, Xiaohui Zhong, Feng Zhang, Yuan Cheng, Yinghui Xu, Yuan Qi, and Hao Li. Fuxi: a cascade machine learning forecasting system for 15-day global weather forecast. *npj climate and atmospheric science*, 6(1):190, 2023.
- [DBK⁺20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [DD21] Tianyun Dong and Wenjie Dong. Evaluation of extreme precipitation over asia in cmip6 models. *Climate Dynamics*, 57(7-8):1751–1769, 2021.
- [Deu23] Deutscher Wetterdienst (DWD). Icon (icosahedral nonhydrostatic) numerical weather prediction model. DWD model documentation (nextGEMS cycle 3 ICON output), 2023. Operational NWP model used in global and regional forecast systems.
- [DIS⁺24] Alessandro Damiani, Noriko N Ishizaki, Hidetaka Sasaki, Sarah Feron, and Raul R Cordero. Exploring super-resolution spatial downscaling of several meteorological variables and potential applications for photovoltaic power. *Scientific Reports*, 14(1):7254, 2024.
- [DL22] Jérôme Dujardin and Michael Lehning. Wind-topo: Downscaling near-surface wind fields to high-resolution topography in highly complex terrain with deep learning. *Quarterly Journal of the Royal Meteorological Society*, 148(744):1368–1388, 2022.
- [DLHT14] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13*, pages 184–199. Springer, 2014.
- [DLHT15] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.
- [DMBA20] Job CM Dullaart, Sanne Muis, Nadia Bloemendaal, and Jeroen CJH Aerts. Advancing global storm surge modelling using the new era5 climate reanalysis. *Climate Dynamics*, 54:1007–1021, 2020.
- [DN21] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.
- [DSG⁺23] Antoine Doury, Samuel Somot, Sebastien Gadat, Aurélien Ribes, and Lola Corre. Regional climate model emulator based on deep learning: Concept and first evaluation of a novel hybrid downscaling approach. *Climate Dynamics*, 60(5):1751–1779, 2023.
- [DYV⁺16] Imke Durre, Xungang Yin, Russell S. Vose, Scott Applequist, Jeff Arnfield, Bryant Korzeniewski, and Bruce Hundermark. Integrated global radiosonde archive (igra), version 2. NOAA National Centers for Environmental Information, 2016.
- [EBM⁺16a] Veronika Eyring, Sandrine Bony, Gerald A Meehl, Catherine A Senior, Bjorn Stevens, Ronald J Stouffer, and Karl E Taylor. Overview of the coupled model intercomparison project phase 6 (cmip6) experimental design and organization. *Geoscientific Model Development*, 9(5):1937–1958, 2016.

- [EBM⁺16b] Veronika Eyring, Sandrine Bony, Gerald A. Meehl, Catherine A. Senior, Bjorn Stevens, Ronald J. Stouffer, and Karl E. Taylor. Overview of the coupled model intercomparison project phase 6 (CMIP6) experimental design and organization. *Geoscientific Model Development*, 9(5):1937–1958, 2016.
- [EUM10] EUMETSAT/OSI SAF. Metop-a ascat level 2 ocean surface wind vectors optimized for coastal ocean, version operational/near-real-time. PO.DAAC, CA, USA, 2010. Accessed: 2025-08-08.
- [FBT07] Hayley J Fowler, Stephen Blenkinsop, and Claudia Tebaldi. Linking climate change modelling to impacts studies: recent advances in downscaling techniques for hydrological modelling. *International Journal of Climatology: A Journal of the Royal Meteorological Society*, 27(12):1547–1578, 2007.
- [FMA⁺14] Gregory Flato, Jochem Marotzke, Babatunde Abiodun, Pascale Braconnot, Sin Chan Chou, William Collins, Peter Cox, Fatima Driouech, Seita Emori, Veronika Eyring, et al. Evaluation of climate models. In *Climate change 2013: the physical science basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*, pages 741–866. Cambridge University Press, 2014.
- [FWW⁺16] E. Freeman, S. D. Woodruff, S. J. Worley, S. J. Lubker, E. C. Kent, W. E. Angel, D. I. Berry, P. Brohan, R. Eastman, L. Gates, W. Gloeden, Z. Ji, J. Lawrimore, N. A. Rayner, G. Rosenhagen, and S. R. Smith. International comprehensive ocean-atmosphere data set (icoads) release 3.0 - monthly summary groups (msg), revision 3.0. NOAA National Centers for Environmental Information (NCEI), 2016.
- [FZM⁺20] Hang Fan, Xuemin Zhang, Shengwei Mei, Kunjin Chen, and Xinyang Chen. M2gsnet: Multi-modal multi-task graph spatiotemporal network for ultra-short-term wind farm cluster power prediction. *Applied Sciences (Switzerland)*, 10:1–15, 11 2020.
- [GCR05] Subhrendu Gangopadhyay, Martyn Clark, and Balaji Rajagopalan. Statistical downscaling using k-nearest neighbors. *Water Resources Research*, 41(2), 2005.
- [GD98] Matt W Gardner and Stephen R Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14-15):2627–2636, 1998.
- [GG15] Filippo Giorgi and William J. Gutowski. Regional dynamical downscaling and the CORDEX initiative. *Wiley Interdisciplinary Reviews: Climate Change*, 6(4):447–457, 2015.
- [GGA⁺23] Elissavet Galanaki, Christos Giannaros, Ilias Agathangelidis, Constantinos Cartalis, Vassiliki Kotroni, Konstantinos Lagouvardos, and Andreas Matzarakis. Validating the copernicus european regional reanalysis (cerre) dataset for human-biometeorological applications. *Environmental Sciences Proceedings*, 26(1):111, 2023.
- [GGJ15] Filippo Giorgi and William J Gutowski Jr. Regional dynamical downscaling and the cordex initiative. *Annual review of environment and resources*, 40(1):467–490, 2015.
- [Gio90] Filippo Giorgi. Simulation of regional climate using a limited area model nested in a general circulation model. *Journal of climate*, 3(9):941–963, 1990.
- [Gio19] Filippo Giorgi. Thirty years of regional climate modeling: where are we and where are we going next? *Journal of Geophysical Research: Atmospheres*, 124(11):5696–5723, 2019.
- [GJA⁺09] Filippo Giorgi, Colin Jones, Ghassem R Asrar, et al. Addressing climate information needs at the regional level: the cordex framework. *World Meteorological Organization (WMO) Bulletin*, 58(3):175, 2009.

- [GKDR⁺24] Emily Gleeson, Ekaterina Kurzeneva, Wim De Rooy, Laura Rontu, Daniel Martín Pérez, Colm Clancy, Karl-Ivar Ivarsson, Bjørg Jenny Engdahl, Sander Tijm, Kristian Pagh Nielsen, et al. The cycle 46 configuration of the harmonie-arome forecast model. *Meteorology*, 3(4):354–390, 2024.
- [GLM⁺23] Naomi Goldenson, L. Ruby Leung, Linda O. Mearns, David W. Pierce, Kevin A. Reed, Isla R. Simpson, Paul Ullrich, Will Krantz, Alex Hall, Andrew Jones, and Stefan Rahimi. Use-inspired, process-oriented gcm selection: Prioritizing models for regional dynamical downscaling. *Bulletin of the American Meteorological Society*, 104(9):E1619 – E1629, 2023.
- [GPAM⁺14] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
- [GRDA21] Javier Gurrola-Ramos, Oscar Dalmau, and Teresa E. Alarcón. A residual dense u-net neural network for image denoising. *IEEE Access*, 9:31742–31754, 2021.
- [GRT22] Sagar Garg, Stephan Rasp, and Nils Thuerey. Weatherbench probability: A benchmark dataset for probabilistic medium-range weather forecasting along with deep learning baseline models. *CoRR*, abs/2205.00865, 2022.
- [GRWG05] Tilmann Gneiting, Adrian E Raftery, Anton H Westveld, and Tom Goldman. Calibrated probabilistic forecasting using ensemble model output statistics and minimum crps estimation. *Monthly Weather Review*, 133(5):1098–1118, 2005.
- [HBB⁺18] H. Hersbach, B. Bell, P. Berrisford, G. Biavati, A. Horányi, J. Muñoz Sabater, J. Nicolas, C. Peubey, R. Radu, I. Rozum, D. Schepers, A. Simmons, C. Soci, D. Dee, and J-N. Thépaut. ERA5 hourly data on single levels from 1940 to present. *Copernicus Climate Change Service (C3S) Climate Data Store (CDS)*, 2018. Accessed on 22-11-2023.
- [HBB⁺20] Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, et al. The era5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020.
- [HBWT09] Nguyen Q Hung, Mukand S Babel, S Weesakul, and NK Tripathi. An artificial neural network model for rainfall forecasting in bangkok, thailand. *Hydrology and Earth System Sciences*, 13(8):1413–1425, 2009.
- [HCSS16] Xiaogang He, Nathaniel W Chaney, Marc Schleiss, and Justin Sheffield. Spatial downscaling of precipitation using adaptable random forests. *Water resources research*, 52(10):8217–8237, 2016.
- [HG16] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.
- [HGL⁺18] Yanting Hu, Xinbo Gao, Jie Li, Yuanfei Huang, and Hanzi Wang. Single image super-resolution via cascaded multi-scale cross network. *arXiv preprint arXiv:1802.08808*, 2018.
- [HGSG97] Kou-lin Hsu, Xiaogang Gao, Soroosh Sorooshian, and Hoshin V Gupta. Precipitation estimation from remotely sensed information using artificial neural networks. *Journal of applied meteorology*, 36(9):1176–1190, 1997.
- [HHGR⁺23] Paula Harder, Alex Hernandez-Garcia, Venkatesh Ramesh, Qidong Yang, Prasanna Sattigeri, Daniela Szwarcman, Campbell Watson, and David Rolnick. Hard-constrained deep learning for climate downscaling. *Journal of Machine Learning Research*, 24(365):1–40, 2023.

- [HHS23] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, pages 13213–13232. PMLR, 2023.
- [HJA20a] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.
- [HJA20b] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc’Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.
- [HKHW20] Kevin Höhlein, Michael Kern, Timothy Hewson, and Rüdiger Westermann. A comparative study of convolutional neural network models for wind field downscaling. *Meteorological Applications*, 27(6):e1961, 2020.
- [HM07] Mohsen Hayati and Zahra Mohebi. Application of artificial neural networks for temperature forecasting. *World Academy of Science, Engineering and Technology*, 28(2):275–279, 2007.
- [HMC⁺22a] Lucy Harris, Andrew T. T. McRae, Matthew Chantry, Peter D. Dueben, and Tim N. Palmer. A generative deep learning approach to stochastic downscaling of precipitation forecasts. *Journal of Advances in Modeling Earth Systems*, 14(10):e2022MS003120, 2022. e2022MS003120 2022MS003120.
- [HMC⁺22b] Lucy Harris, Andrew TT McRae, Matthew Chantry, Peter D Dueben, and Tim N Palmer. A generative deep learning approach to stochastic downscaling of precipitation forecasts. *Journal of Advances in Modeling Earth Systems*, 14(10):e2022MS003120, 2022.
- [HSB18] Mattias P Heinrich, Maik Stille, and Thorsten M Buzug. Residual u-net convolutional neural network architecture for low-dose ct denoising. *Current Directions in Biomedical Engineering*, 4(1):297–300, 2018.
- [HSU18] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1664–1673, 2018.
- [HT98] William W Hsieh and Benyang Tang. Applying neural network models to prediction and data analysis in meteorology and oceanography. *Bulletin of the American Meteorological Society*, 79(9):1855–1870, 1998.
- [HWLD14] Zhibin He, Xiaohu Wen, Hu Liu, and Jun Du. A comparative study of artificial neural network, adaptive neuro fuzzy inference system and support vector machine for forecasting river flow in the semiarid mountain region. *Journal of Hydrology*, 509:379–386, 2014.
- [HZRS16a] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [HZRS16b] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [IZZE17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

- [Jap25] Japan Meteorological Agency (JMA). Jma global spectral model (gsm) forecasts. JMA High-Resolution GSM Data Service, 2025. Operational global NWP at 0.125° resolution, forecasts up to 132 h four times daily, and up to 264 h twice daily.
- [JLW⁺22] Weixin Jin, Yong Luo, Tongwen Wu, Xiaomeng Huang, Wei Xue, and Chaoqing Yu. Deep learning for seasonal precipitation prediction over china. *Journal of Meteorological Research*, 36(2):271–281, 2022.
- [JM02] Marcin Jaruszewicz and Jacek Mandziuk. Application of pca method to weather prediction task. In *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP’02.*, volume 5, pages 2359–2363. IEEE, 2002.
- [JPE⁺14] Daniela Jacob, Juliane Petersen, Bastian Eggert, Antoinette Alias, Ole Bøssing Christensen, Laurens M. Bouwer, Alain Braun, Augustin Colette, Michel Déqué, Goran Georgievski, Elena Georgopoulou, Andreas Gobiet, Laurent Menut, Grigory Nikulin, Andreas Haensler, Nils Hempelmann, Colin Jones, Klaus Keuler, Sari Kovats, Nico Kröner, Sven Kotlarski, Arne Kriegsmann, Eric Martin, Erik van Meijgaard, Christopher Moseley, Susanne Pfeifer, Swantje Preuschmann, Christine Radermacher, Kai Radtke, Diana Rechid, Mark Rounsevell, Patrick Samuelsson, Samuel Somot, Jean-Francois Soussana, Claas Teichmann, Riccardo Valentini, Robert Vautard, Björn Weber, and Pascal Yiou. Euro-cordex: new high-resolution climate change projections for european impact research. *Regional Environmental Change*, 14(2):563–578, Apr 2014.
- [KAAL22] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.
- [KAS⁺23] Bipin Kumar, Kaustubh Atey, Bhupendra Bahadur Singh, Rajib Chattopadhyay, Nachiketa Acharya, Manmeet Singh, Ravi S Nanjundiah, and Suryachandra A Rao. On the modern deep learning approaches for precipitation downscaling. *Earth Science Informatics*, 16(2):1459–1472, 2023.
- [KB98] Robert J Kuligowski and Ana P Barros. Experiments in short-term precipitation forecasting using artificial neural networks. *Monthly weather review*, 126(2):470–482, 1998.
- [KBR⁺17] Elizabeth J Kendon, Nikolina Ban, Nigel M Roberts, Hayley J Fowler, Malcolm J Roberts, Steven C Chan, Jason P Evans, Giorgia Fossler, and Jonathan M Wilkinson. Do convection-permitting regional climate models improve projections of future precipitation change? *Bulletin of the American Meteorological Society*, 98(1):79–93, 2017.
- [KCD06] Mohammad Sajjad Khan, Paulin Coulibaly, and Yonas Dibike. Uncertainty analysis of statistical downscaling methods. *Journal of Hydrology*, 319(1-4):357–382, 2006.
- [Kei22] Ryan Keisler. Forecasting global weather with graph neural networks. *arXiv preprint arXiv:2202.07575*, 2022.
- [Key03] Robert Keys. Cubic convolution interpolation for digital image processing. *IEEE transactions on acoustics, speech, and signal processing*, 29(6):1153–1160, 2003.
- [KG20] Rina Komatsu and Tad Gonsalves. Comparing u-net based models for denoising color images. *AI*, 1(4):465–486, 2020.
- [KK21] Koushik Sivarama Krishnan and Karthik Sivarama Krishnan. Swiftsrgan-rethinking super-resolution for efficient and real-time inference. In *2021 International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA)*, pages 46–51. IEEE, 2021.

- [KLL16] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654, 2016.
- [KLS⁺23] Kai Kornhuber, Corey Lesk, Carl Schleussner, Jonas Jägermeyr, Peter Pfleiderer, and Radley Horton. Risks of synchronized low yields are underestimated in climate and crop model projections. *Nature Communications*, 14, 07 2023.
- [KRG⁺25] Khaled Kahouli, Winfried Ripken, Stefan Gugler, Oliver T Unke, Klaus-Robert Müller, and Shinichi Nakajima. Enhancing diffusion models efficiency by disentangling total-variance and signal-to-noise ratio. *arXiv preprint arXiv:2502.08598*, 2025.
- [KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.
- [KSPH21] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.
- [KVLG⁺21] Rupa Kurinchi-Vendhan, Björn Lütjens, Ritwik Gupta, Lucien Werner, and Dava Newman. Wisosuper: Benchmarking super-resolution methods on wind and solar data. *arXiv preprint arXiv:2109.08770*, 2021.
- [KW18] Mahdi Khodayar and Jianhui Wang. Spatio-temporal graph deep neural network for short-term wind speed forecasting. *IEEE Transactions on Sustainable Energy*, 10(2):670–681, 2018.
- [KYL⁺24] Dmitrii Kochkov, Janni Yuval, Ian Langmore, Peter Norgaard, Jamie Smith, Griffin Mooers, Milan Klöwer, James Lottes, Stephan Rasp, Peter Düben, et al. Neural general circulation models for weather and climate. *Nature*, 632(8027):1060–1066, 2024.
- [LCLG⁺24] Lizao Li, Robert Carver, Ignacio Lopez-Gomez, Fei Sha, and John Anderson. Generative emulation of weather forecast ensembles with diffusion models. *Science Advances*, 10(13):eadk4489, 2024.
- [LCS⁺21] Jingyun Liang, Jiezhong Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1833–1844, 2021.
- [LCW⁺24] Haonan Lin, Yan Chen, Jiahao Wang, Wenbin An, Mengmeng Wang, Feng Tian, Yong Liu, Guang Dai, Jingdong Wang, and Qianying Wang. Schedule your edit: A simple yet effective diffusion noise schedule for image editing. *Advances in Neural Information Processing Systems*, 37:115712–115756, 2024.
- [LD24] Jangho Lee and Andrew E Dessler. Improved surface urban heat impact assessment using goes satellite data: A comparative study with era-5. *Geophysical Research Letters*, 51(1):e2023GL107364, 2024.
- [LGDG23] Dawa Chyophel Lepcha, Bhawna Goyal, Ayush Dogra, and Vishal Goyal. Image super-resolution: A comprehensive review, recent trends, challenges and applications. *Information Fusion*, 91:230–260, 2023.
- [LGG24] Joseph W. Lockwood, Avantika Gori, and Pierre Gentine. A generative super-resolution model for enhancing tropical cyclone wind field intensity and resolution. *Journal of Geophysical Research: Machine Learning and Computation*, 1, 12 2024.
- [LGWZN⁺24a] Ignacio Lopez-Gomez, Zhong Yi Wan, Leonardo Zepeda-Núñez, Tapio Schneider, John Anderson, and Fei Sha. Dynamical-generative downscaling of climate model ensembles. *arXiv preprint arXiv:2410.01776*, 2024.

- [LGWZN⁺24b] Ignacio Lopez-Gomez, Zhong Yi Wan, Leonardo Zepeda-Núñez, Tapio Schneider, John Anderson, and Fei Sha. Dynamical-generative downscaling of climate model ensembles, 2024.
- [LH17] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [LHL20] Jiang Liu, Daniel Fiifi Tawia Hagan, and Yi Liu. Global land surface temperature change (2003–2017) and its relationship with climate drivers: Airs, modis, and era5-land based analysis. *Remote Sensing*, 13(1):44, 2020.
- [LHN⁺23] Jussi Leinonen, Ulrich Hamann, Daniele Nerini, Urs Germann, and Gabriele Franch. Latent diffusion models for generative precipitation nowcasting with accurate uncertainty quantification. *arXiv preprint arXiv:2304.12891*, 2023.
- [LKM⁺18] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study. *Advances in neural information processing systems*, 31, 2018.
- [LL05] Kan Li and Yu-Shu Liu. A rough set based fuzzy neural network algorithm for weather prediction. In *2005 international conference on machine learning and cybernetics*, volume 3, pages 1888–1892. IEEE, 2005.
- [LLC⁺21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.
- [LLL⁺22a] Fenghua Ling, Jing-Jia Luo, Yue Li, Tao Tang, Lei Bai, Wanli Ouyang, and Toshio Yamagata. Multi-task machine learning improves multi-seasonal prediction of the indian ocean dipole. *Nature Communications*, 13(1):7681, 2022.
- [LLL⁺22b] Zhisheng Lu, Juncheng Li, Hong Liu, Chaoyan Huang, Linlin Zhang, and Tiejong Zeng. Transformer for single image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 457–466, 2022.
- [LLL⁺24] Fenghua Ling, Zeyu Lu, Jing-Jia Luo, Lei Bai, Swadhin K Behera, Dachao Jin, Baoxiang Pan, Huidong Jiang, and Toshio Yamagata. Diffusion model-based probabilistic downscaling for 180-year east asian climate reconstruction. *npj Climate and Atmospheric Science*, 7(1):131, 2024.
- [LNU⁺20] Sehyung Lee, Makiko Negishi, Hidetoshi Urakubo, Haruo Kasai, and Shin Ishii. Munet: Multi-scale u-net for two-photon microscopy image denoising and restoration. *Neural Networks*, 125:92–103, 2020.
- [LOLB04] DF Lekkas, C Onof, MJ Lee, and EA Baltas. Application of artificial neural networks for flood forecasting. *Global Nest Journal*, 6(3):205–211, 2004.
- [LRC⁺16] Yunjie Liu, Evan Racah, Joaquin Correa, Amir Khosrowshahi, David Lavers, Kenneth Kunkel, Michael Wehner, William Collins, et al. Application of deep convolutional neural networks for detecting extreme weather in climate datasets. *arXiv preprint arXiv:1605.01156*, 2016.
- [LS19] Zachary C Lipton and Jacob Steinhardt. Troubling trends in machine learning scholarship: Some ml papers suffer from flaws that could mislead the public and stymie future research. *Queue*, 17(1):45–77, 2019.
- [LSG⁺24] Jieli Liu, Chunxiang Shi, Lingling Ge, Ruian Tie, Xiaojian Chen, Tao Zhou, Xiang Gu, and Zhanfei Shen. Enhanced wind field spatial downscaling method using unet architecture and dual cross-attention mechanism. *Remote Sensing*, 16(11):1867, 2024.

- [LSG⁺25] Jieli Liu, Chunxiang Shi, Lingling Ge, Ruian Tie, Tao Zhou, Xiaojian Chen, Xiang Gu, Yue Wu, and Zhanfei Shen. A spatial downscaling approach for enhanced accuracy in high wind speed estimation using hybrid attention transformer. *Journal of Meteorological Research*, 39(2):303–321, 2025.
- [LSGW⁺22] Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirsberger, Meire Fortunato, Alexander Pritzel, Suman Ravuri, Timo Ewalds, Ferran Alet, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Jacklynn Stott, Oriol Vinyals, Shakir Mohamed, and Peter Battaglia. Graphcast: Learning skillful medium-range global weather forecasting, 2022.
- [LSGW⁺23] Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, et al. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023.
- [LSK⁺17] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017.
- [LSR⁺22] Jia Liu, Yongjian Sun, Kaijun Ren, Yanlai Zhao, Kefeng Deng, and Lizhe Wang. A spatial downscaling approach for windsat satellite sea surface wind based on generative adversarial networks and dual learning scheme. *Remote Sensing*, 14(3), 2022.
- [LSVR22] David A Lavers, Adrian Simmons, Freja Vamborg, and Mark J Rodwell. An evaluation of era5 precipitation for climate monitoring. *Quarterly Journal of the Royal Meteorological Society*, 148(748):3152–3165, 2022.
- [LSWC23] Huan Liu, Mingwen Shao, Chao Wang, and Feilong Cao. Image super-resolution using a simple transformer without pretraining. *Neural Processing Letters*, 55(2):1479–1497, 2023.
- [LTH⁺17] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.
- [LWH⁺25] Haijie Li, Ya Wang, Gang Huang, Weichen Tao, and Pengfei Lin. Generative downscaling and bias correction of multivariable earth system model simulations. *Geophysical Research Letters*, 52(18):e2025GL117397, 2025.
- [LWP⁺23] Pengwei Liu, Wenwei Wang, Bingqing Peng, Binqing Wu, and Liang Sun. Dsaf: A dual-stage adaptive framework for numerical weather prediction downscaling. *arXiv preprint arXiv:2312.12476*, 2023.
- [LYC⁺22] Haoying Li, Yifan Yang, Meng Chang, Shiqi Chen, Huajun Feng, Zhihai Xu, Qi Li, and Yueting Chen. Srdiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing*, 479:47–59, 2022.
- [LYZ⁺16] Jiaming Liu, Di Yuan, Liping Zhang, Xia Zou, and Xingyuan Song. Comparison of three statistical downscaling methods and ensemble downscaling method based on bayesian model averaging in upper hanjiang river basin, china. *Advances in Meteorology*, 2016:1–12, 2016.
- [LZVO25] Parthiban Loganathan, Elias Zea, Ricardo Vinuesa, and Evelyn Otero. Regional climate projections using a deep-learning-based model-ranking and downscaling framework: Application to european climate zones. *arXiv preprint arXiv:2502.20132*, 2025.
- [MAC24a] Fabio Merizzi, Andrea Asperti, and Stefano Colamonaco. Wind speed super-resolution and validation: from era5 to cerra via diffusion models. *Neural Computing and Applications*, 36(34):21899–21921, 2024.

- [MAC24b] Fabio Merizzi, Andrea Asperti, and Stefano Colamonaco. Wind speed super-resolution and validation: from era5 to cerra via diffusion models. *Neural Computing and Applications*, Sep 2024.
- [Mar13] Douglas Maraun. Bias correction, quantile mapping, and downscaling: Revisiting the inflation issue. *Journal of Climate*, 26(6):2137–2143, 2013.
- [MBC⁺23] Morteza Mardani, Noah D Brenowitz, Yair Cohen, Jaideep Pathak, Chieh-Yu Chen, Cheng-Chin Liu, Arash Vahdat, Karthik Kashinath, Jan Kautz, and Mike Pritchard. Generative residual diffusion modeling for km-scale atmospheric downscaling. *CoRR*, 2023.
- [MBC⁺24] Morteza Mardani, Noah Brenowitz, Yair Cohen, Jaideep Pathak, Chieh-Yu Chen, Cheng-Chin Liu, Arash Vahdat, Mohammad Amin Nabian, Tao Ge, Akshay Subramaniam, Karthik Kashinath, Jan Kautz, and Mike Pritchard. Residual corrective diffusion modeling for km-scale atmospheric downscaling, 2024.
- [McC92] Donald W McCann. A neural network short-term forecast of significant thunderstorms. *Weather and Forecasting*, 7(3):525–534, 1992.
- [MDK⁺12] Matthew J. Menne, Imke Durre, Bryant Korzeniewski, Shelley McNeill, Kristy Thomas, Xungang Yin, Steven Anthony, Ron Ray, Russell S. Vose, Byron E. Gleason, and Tamara G. Houston. Global historical climatology network - daily (ghcn-daily), version 3. NOAA National Climatic Data Center, 2012.
- [MDKL23] Ioannis Maslounidis, Stavros Dafis, Georgios Kyros, and Konstantinos Lagouvardos. Snow depth trends of european ski resorts. *Environmental Sciences Proceedings*, 26(1), 2023.
- [MDV⁺12a] M. J. Menne, I. Durre, R. S. Vose, B. E. Gleason, and T. G. Houston. An overview of the global historical climatology network-daily database. *Journal of Atmospheric and Oceanic Technology*, 29:897–910, 2012.
- [MDV⁺12b] Matthew J. Menne, Imke Durre, Russell S. Vose, Byron E. Gleason, and Tamara G. Houston. An overview of the global historical climatology network-daily database. *Journal of Atmospheric and Oceanic Technology*, 29:897–910, 2012.
- [MEL25] Fabio Merizzi, Davide Evangelista, and Harilaos Loukos. Controlling ensemble variance in diffusion models: An application for reanalyses downscaling. *arXiv preprint arXiv:2501.14822*, 2025.
- [MFK19] Andreise Moreira, Denise Cybis Fontana, and Tatiana Mora Kuplich. Wavelet approach applied to evi/modis time series and meteorological data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 147:335–344, 2019.
- [MKA02] Imran Maqsood, Muhammad Riaz Khan, and Ajith Abraham. Neurocomputing based canadian weather analysis. In *Computational Intelligence and applications*, pages 39–44. 2002.
- [ML25] Fabio Merizzi and Harilaos Loukos. Vision transformers for multi-variable climate downscaling: Emulating regional climate models with a shared encoder and multi-decoder architecture. *arXiv preprint arXiv:2506.22447*, 2025.
- [MLMM⁺13] Valéry Masson, Patrick Le Moigne, Eric Martin, Stéphanie Faroux, A Alias, Ramdane Alkama, Sophie Belamari, Alina Barbu, A Boone, François Bouyssel, et al. The surfexv7. 2 land and ocean surface platform for coupled or offline simulation of earth surface variables and fluxes. *Geoscientific Model Development*, 6(4):929–960, 2013.
- [MO14] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

- [MRZ⁺25] Ilenia Manco, Walter Riviera, Andrea Zanetti, Marco Briscolini, Paola Mercogliano, and Antonio Navarra. A new conditional generative adversarial neural network approach for statistical downscaling of the era5 reanalysis over the italian peninsula. *Environmental Modelling & Software*, 188:106427, 2025.
- [MS96] Caren Marzban and Gregory J Stumpf. A neural network for tornado prediction based on doppler radar-derived attributes. *Journal of Applied Meteorology and Climatology*, 35(5):617–626, 1996.
- [MSJ⁺09] James M Murphy, DMH Sexton, Geoff J Jenkins, BB Booth, CC Brown, Robin T Clark, M Collins, GR Harris, EJ Kendon, RA Betts, et al. Uk climate projections science report: climate change projections. 2009.
- [MSL⁺13] LO Mearns, Steve Sain, LR Leung, MS Bukovsky, Seth McGinnis, S Biner, Daniel Caya, RW Arritt, William Gutowski, E Takle, et al. Climate change projections of the north american regional climate change assessment program (narccap). *Climatic Change*, 120(4):965–975, 2013.
- [MWI⁺10] Douglas Maraun, Frederick Wetterhall, Anderson M Ireson, Richard E Chandler, Elizabeth J Kendon, Martin Widmann, S Brien, Henning W Rust, Tobias Sauter, Matthias Themeßl, et al. Precipitation downscaling under climate change: Recent developments to bridge the gap between dynamical models and the end user. *Reviews of geophysics*, 48(3), 2010.
- [Nat07] National Centers for Environmental Prediction (NCEP), National Oceanic and Atmospheric Administration (NOAA), U.S. Department of Commerce. Ncep global forecast system (gfs) analyses and forecasts, 0.25° grid. Research Data Archive, dataset d084001, 2007. Accessed via NCAR Research Data Archive.
- [Nat25] National Centers for Environmental Prediction (NCEP), NOAA. Noaa high-resolution rapid refresh (hrrr) model output. NOAA/HWRF data via NOAA Open Data Program and AWS Registry of Open Data, 2025. 3 km, hourly-updated, convection-allowing operational forecast product.
- [NBK⁺23] Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K Gupta, and Aditya Grover. Climax: A foundation model for weather and climate. *arXiv preprint arXiv:2301.10343*, 2023.
- [NC94] HD Navone and HA Ceccatto. Predicting indian monsoon rainfall: a neural network approach. *Climate Dynamics*, 10(6):305–312, 1994.
- [ND21] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.
- [NJB⁺23] Tung Nguyen, Jason Jewik, Hritik Bansal, Prakhar Sharma, and Aditya Grover. Climatelearn: Benchmarking machine learning for weather and climate modeling. *arXiv preprint arXiv:2307.01909*, 2023.
- [NJG⁺12] Grigory Nikulin, Colin Jones, Filippo Giorgi, Ghassem Asrar, Matthias Büchner, Ruth Cerezo-Mota, Ole Bøssing Christensen, Michel Déqué, Jesus Fernandez, Andreas Häsler, et al. Precipitation climatology in an ensemble of corDEX-africa regional climate simulations. *Journal of Climate*, 25(18):6057–6078, 2012.
- [NPR⁺23] Gianfranco Nicodemo, Gaetano Pecoraro, Guido Rianna, Alfredo Reder, Davide Luongo, Dario Peduto, and Michele Calvella. Remote sensing meteorological and dinsar historical data to analyse the kinematic behaviour of slow-moving landslides at municipal scale. In *National Conference of the Researchers of Geotechnical Engineering*, pages 242–250. Springer, 2023.
- [ODO16] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 1(10):e3, 2016.

- [OIKY23] Norihiro Oyama, Noriko N Ishizaki, Satoshi Koide, and Hiroaki Yoshida. Deep generative model super-resolves spatially correlated multiregional climate data. *Scientific Reports*, 13(1):5992, 2023.
- [OYK⁺24] Shuntaro Okada, Ryota Yoshihashi, Hirokatsu Kataoka, Tomohiro Tanaka, et al. Constant rate scheduling: Constant-rate distributional change for efficient training and sampling in diffusion models. *arXiv preprint arXiv:2411.12188*, 2024.
- [Pal00] Tim N Palmer. Predicting uncertainty in forecasts of weather and climate. *Reports on progress in Physics*, 63(2):71, 2000.
- [PBSG09] David W. Pierce, Tim P. Barnett, Benjamin D. Santer, and Peter J. Gleckler. Selecting global climate models for regional climate change studies. *Proceedings of the National Academy of Sciences*, 106(21):8441–8446, 2009.
- [PCMG24] Antonio Pérez, Mario Santa Cruz, Daniel San Martín, and José Manuel Gutiérrez. Transformer based super-resolution downscaling for regional reanalysis: Full domain vs tiling approaches. *arXiv preprint arXiv:2410.12728*, 2024.
- [Pel23] A. Pelosi. Performance of the copernicus european regional reanalysis (cerra) dataset as proxy of ground-based agrometeorological data. *Agricultural Water Management*, 289:108556, 2023.
- [PKT07] J Anthony Parker, Robert V Kenyon, and Donald E Troxel. Comparison of interpolating methods for image resampling. *IEEE Transactions on medical imaging*, 2(1):31–39, 2007.
- [PLF⁺15] Andreas F Prein, Wolfgang Langhans, Giorgia Fosser, Andrew Ferrone, Nikolina Ban, Klaus Goergen, Michael Keller, Merja Tölle, Oliver Gutjahr, Frauke Feser, et al. A review on regional convection-permitting climate modeling: Demonstrations, prospects, and challenges. *Reviews of geophysics*, 53(2):323–361, 2015.
- [PMPN22] Linsey S Passarella, Salil Mahajan, Anikesh Pal, and Matthew R Norman. Reconstructing high resolution esm data through a novel fast super resolution convolutional neural network (fsrnn). *Geophysical Research Letters*, 49(4):e2021GL097571, 2022.
- [PPK03] Sung Cheol Park, Min Kyu Park, and Moon Gi Kang. Super-resolution image reconstruction: a technical overview. *IEEE signal processing magazine*, 20(3):21–36, 2003.
- [PSC16] Sahar Hadi Pour, Shamsuddin Shahid, and Eun-Sung Chung. A hybrid model for statistical downscaling of daily rainfall. *Procedia Engineering*, 154:1424–1430, 2016.
- [PSGA⁺23] Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, et al. Gencast: Diffusion-based ensemble forecasting for medium-range weather. *arXiv preprint arXiv:2312.15796*, 2023.
- [PSH⁺22] Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Aizzadenesheli, Pedram Hassanzadeh, Karthik Kashinath, and Animashree Anandkumar. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators, 2022.
- [PVLS⁺21] Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d’Alché Buc, Emily Fox, and Hugo Larochelle. Improving reproducibility in machine learning research (a report from the neurips 2019 reproducibility program). *Journal of machine learning research*, 22(164):1–20, 2021.
- [PYK⁺19] Quoc Bao Pham, Tao-Chang Yang, Chen-Min Kuo, Hung-Wei Tseng, and Pao-Shan Yu. Combining random forest and least square support vector regression for improving extreme rainfall downscaling. *Water*, 11(3):451, 2019.

- [QCBMBB23] Dánnell Quesada-Chacón, Jorge Baño-Medina, Klemens Barfus, and Christian Bernhofer. Downscaling cordex through deep learning to daily 1 km multivariate ensemble in complex terrain. *Earth's Future*, 11(8):e2023EF003531, 2023.
- [RBK21] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021.
- [RBL⁺22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.
- [RBLM⁺24] Martin Ridal, Eric Bazile, Patrick Le Moigne, Roger Randriamampianina, Semjon Schimanke, Ulf Andrae, Lars Berggren, Pierre Brousseau, Per Dahlgren, Lisette Edvinsson, et al. Cerra, the copernicus european regional reanalysis system. *Quarterly Journal of the Royal Meteorological Society*, 2024.
- [RBM⁺17] Evan Racah, Christopher Beckham, Tegan Maharaj, Samira Ebrahimi Kahou, Mr Prabhat, and Chris Pal. Extremeweather: A large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events. *Advances in neural information processing systems*, 30, 2017.
- [RBM⁺24] Martin Ridal, Eric Bazile, Patrick Le Moigne, Roger Randriamampianina, Semjon Schimanke, Ulf Andrae, Lars Berggren, Pierre Brousseau, Per Dahlgren, Lisette Edvinsson, Adam El-Said, Michael Glinton, Susanna Hagelin, Susanna Hopsch, Ludvig Isaksson, Paulo Medeiros, Esbjörn Olsson, Per Uden, and Zheng Qi Wang. Cerra, the copernicus european regional reanalysis system. *Quarterly Journal of the Royal Meteorological Society*, 2024.
- [RCVS⁺19] Markus Reichstein, Gustau Camps-Valls, Bjorn Stevens, Martin Jung, Joachim Denzler, Nuno Carvalhais, and F Prabhat. Deep learning and process understanding for data-driven earth system science. *Nature*, 566(7743):195–204, 2019.
- [RDK⁺22] David Rolnick, Priya L Donti, Lynn H Kaack, Kelly Kochanski, Alexandre Lacoste, Kris Sankaran, Andrew Slavin Ross, Nikola Milojevic-Dupont, Natasha Jaques, Anna Waldman-Brown, et al. Tackling climate change with machine learning. *ACM Computing Surveys (CSUR)*, 55(2):1–96, 2022.
- [REKL17] Haoyu Ren, Mostafa El-Khamy, and Jungwon Lee. Image super resolution based on fusing multiple convolution neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 54–61, 2017.
- [RFB15a] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [RFB15b] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.
- [RHG⁺24] Neelesh Rampal, Sanaa Hobeichi, Peter B. Gibson, Jorge Baño-Medina, Gab Abramowitz, Tom Beucler, Jose González-Abad, William Chapman, Paula Harder, and José Manuel Gutiérrez. Enhancing regional climate downscaling through advances in machine learning. *Artificial Intelligence for the Earth Systems*, 3, 3 2024.
- [RLR⁺21] Xiaoli Ren, Xiaoyong Li, Kaijun Ren, Junqiang Song, Zichen Xu, Kefeng Deng, and Xiang Wang. Deep learning-based weather prediction: a survey. *Big Data Research*, 23:100178, 2021.

- [ROCn18] Eduardo Rocha Rodrigues, Igor Oliveira, Renato Cunha, and Marco Netto. Deep-downscale: A deep learning strategy for high-resolution weather forecast. In *2018 IEEE 14th International Conference on e-Science (e-Science)*, pages 415–422. IEEE, 2018.
- [RSL⁺07] R. W. Reynolds, T. M. Smith, C. Liu, D. B. Chelton, K. S. Casey, and M. G. Schlax. Daily high-resolution-blended analyses for sea surface temperature. *Journal of Climate*, 20:5473–5496, 2007.
- [RSM⁺18] Christopher D Roberts, Retish Senan, Franco Molteni, Souhail Boussetta, Michael Mayer, and Sarah PE Keeley. Climate model configurations of the ecmwf integrated forecasting system (ecmwf-ifs cycle 43r1) for highresmp. *Geoscientific model development*, 11(9):3681–3712, 2018.
- [RT21] Stephan Rasp and Nils Thuerey. Data-driven medium-range weather prediction with a resnet pretrained on climate simulations: A new model for weatherbench. *Journal of Advances in Modeling Earth Systems*, 13(2):e2020MS002405, 2021.
- [RTC22] Matteo Rolle, Stefania Tamea, and Pierluigi Claps. Climate-driven trends in agricultural water requirement: an era5-based assessment at daily scale over 50 years. *Environmental Research Letters*, 17(4):044017, 2022.
- [RWH08] Burkhard Rockel, Andreas Will, and Andreas Hense. The regional climate model cosmo-clm (ccm). *Meteorologische zeitschrift*, 17(4):347, 2008.
- [SBF⁺11] Udo Schneider, Andreas Becker, Peter Finger, Anja Meyer-Christoffer, Bruno Rudolf, and Markus Ziese. Gpcc full data reanalysis version 6.0 at 0.5 degree: Monthly land-surface precipitation from rain-gauges built on gts-based and historic data, 2011.
- [SBG⁺21] Martin G Schultz, Clara Betancourt, Bing Gong, Felix Kleinert, Michael Langguth, Lukas Hubert Leufen, Amirpasha Mozaffari, and Scarlet Stadtler. Can deep learning beat numerical weather prediction? *Philosophical Transactions of the Royal Society A*, 379(2194):20200097, 2021.
- [SBLW02] Mikhail A Semenov, Elaine M Barrow, and A Lars-Wg. A stochastic weather generator for use in climate impact studies. *User Man Herts UK*, pages 1–27, 2002.
- [SBM⁺89] V.V. Salomonson, W.L. Barnes, P.W. Maymon, H.E. Montgomery, and H. Ostrow. Modis: advanced facility instrument for studies of the earth as a system. *IEEE Transactions on Geoscience and Remote Sensing*, 27(2):145–153, 1989.
- [SCH⁺16] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016.
- [Sch23] Uwe Schulzweida. Cdo user guide, October 2023.
- [SCW⁺15] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.
- [SDR⁺24a] Yongjian Sun, Kefeng Deng, Kaijun Ren, Jia Liu, Chongjiu Deng, and Yongjun Jin. Deep learning in statistical downscaling for deriving high spatial resolution gridded meteorological data: A systematic review, 2 2024.
- [SDR⁺24b] Yongjian Sun, Kefeng Deng, Kaijun Ren, Jia Liu, Chongjiu Deng, and Yongjun Jin. Deep learning in statistical downscaling for deriving high spatial resolution gridded meteorological data: A systematic review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 208:14–38, 2024.

- [SDWMG15] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015.
- [SEH⁺20] Casper Kaae Sønderby, Lasse Espeholt, Jonathan Heck, Mostafa Dehghani, Avital Oliver, Tim Salimans, Shreya Agrawal, Jason Hickey, and Nal Kalchbrenner. Metnet: A neural weather model for precipitation forecasting. *arXiv preprint arXiv:2003.12140*, 2020.
- [SGB23] Marianna Szabó, Estíbaliz Gascón, and Sándor Baran. Parametric postprocessing of dual-resolution precipitation forecasts. *Weather and Forecasting*, 38(8):1313–1322, 2023.
- [SGDSK24] Subham Sahoo, Aaron Gokaslan, Christopher M De Sa, and Volodymyr Kuleshov. Diffusion models with learned adaptive noise. *Advances in Neural Information Processing Systems*, 37:105730–105779, 2024.
- [SGHK20] Karen Stengel, Andrew Glaws, Dylan Hettinger, and Ryan N King. Adversarial super-resolution of climatological wind and solar data. *Proceedings of the National Academy of Sciences*, 117(29):16805–16815, 2020.
- [SGIWS20a] Yingkai Sha, David John Gagne II, Gregory West, and Roland Stull. Deep-learning-based gridded downscaling of surface meteorological variables in complex terrain. part i: Daily maximum and minimum 2-m temperature. *Journal of Applied Meteorology and Climatology*, 59(12):2057–2073, 2020.
- [SGIWS20b] Yingkai Sha, David John Gagne II, Gregory West, and Roland Stull. Deep-learning-based gridded downscaling of surface meteorological variables in complex terrain. part ii: Daily precipitation. *Journal of Applied Meteorology and Climatology*, 59(12):2075–2092, 2020.
- [SGK00] Seth E Snell, Sucharita Gopal, and Robert K Kaufmann. Spatial interpolation of surface air temperatures using artificial neural networks: Evaluating their use for downscaling gcms. *Journal of Climate*, 13(5):886–895, 2000.
- [SGL⁺17] Xingjian Shi, Zhihan Gao, Leonard Lausen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. Deep learning for precipitation nowcasting: A benchmark and a new model. *Advances in neural information processing systems*, 30, 2017.
- [SGZ⁺16] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.
- [SH22] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.
- [SHC⁺22] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE transactions on pattern analysis and machine intelligence*, 45(4):4713–4726, 2022.
- [SHG⁺00] Soroosh Sorooshian, Kuo-Lin Hsu, Xiaogang Gao, Hoshin V Gupta, Bisher Imam, and Dan Braithwaite. Evaluation of persiann system satellite-based estimates of tropical rainfall. *Bulletin of the American Meteorological Society*, 81(9):2035–2046, 2000.
- [SHLR00] DJ Sailor, T Hu, X Li, and JN Rosen. A neural network approach to local downscaling of gcm output for assessing wind power implications of climate change. *Renewable energy*, 19(3):359–378, 2000.
- [SHP⁺24] Rahul Sundar, Yucong Hu, Nishant Parashar, Antoine Blanchard, and Boyko Dodov. Taudiff: Highly efficient kilometer-scale downscaling using generative diffusion models. *arXiv preprint arXiv:2412.13627*, 2024.

- [SJPB⁺17] Gail Skofronick-Jackson, Walter A Petersen, Wesley Berg, Chris Kidd, Erich F Stocker, Dalia B Kirschbaum, Ramesh Kakar, Scott A Braun, George J Huffman, Toshio Iguchi, et al. The global precipitation measurement (gpm) mission for science and society. *Bulletin of the American Meteorological Society*, 98(8):1679–1695, 2017.
- [SME20a] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- [SME20b] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. *arXiv e-prints*, page arXiv:2010.02502, October 2020.
- [SME22] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022.
- [SMMB⁺17] Daniel San-Martín, Rodrigo Manzananas, Swen Brands, Sixto Herrera, and José M Gutiérrez. Reassessing model uncertainty for regional projections of precipitation with an ensemble of statistical downscaling methods. *Journal of Climate*, 30(1):203–223, 2017.
- [SP01] Justin T Schoof and SC Pryor. Downscaling temperature and precipitation: A comparison of regression-based methods and artificial neural networks. *International Journal of Climatology: A Journal of the Royal Meteorological Society*, 21(7):773–790, 2001.
- [SPA⁺06] VK Somvanshi, OP Pandey, PK Agrawal, NV Kalanker, M Ravi Prakash, and Ramesh Chand. Modeling and prediction of rainfall using artificial neural network and arima techniques. *J. Ind. Geophys. Union*, 10(2):141–151, 2006.
- [SRLM⁺21] S. Schimanke, M. Ridal, P. Le Moigne, L. Berggren, P. Undén, R. Randriamampianina, U. Andrea, E. Bazile, A. Bertelsen, P. Brousseau, P. Dahlgren, L. Edvinsson, A. El Said, M. Ginton, S. Hopsch, L. Isaksson, R. Mladek, E. Olsson, A. Verrelle, and Z.Q. Wang. CERRA sub-daily regional reanalysis data for Europe on single levels from 1984 to present. *Copernicus Climate Change Service (C3S) Climate Data Store (CDS)*, 2021. Accessed on 22-11-2023.
- [SSDK⁺20] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.
- [SSZH23] Xueshun Shen, Yong Su, Hongliang Zhang, and Jianglin Hu. New version of the cma-gfs dynamical core based on the predictor–corrector time integration scheme. *Journal of Meteorological Research*, 37(3):273–285, 2023.
- [SWMG15] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. 37:2256–2265, 2015.
- [SWZ⁺22] Liucheng Shen, Jiahong Wen, Yuqing Zhang, Safi Ullah, Jingya Cheng, and Xiangchun Meng. Changes in population exposure to extreme precipitation in the yangtze river delta, china. *Climate Services*, 27:100317, 2022.
- [SZZ⁺20] Hao Sun, Baichi Zhou, Chuanjun Zhang, Hongxing Liu, and Bo Yang. Dscale_mod16: A model for disaggregating microwave satellite soil moisture with land surface evapotranspiration products and gridded meteorological data. *Remote Sensing*, 12(6), 2020.
- [TAC23] Matteo Turisini, Giorgio Amati, and Mirko Cestari. Leonardo: A pan-european pre-exascale supercomputer for hpc and ai applications. *arXiv preprint arXiv:2307.16885*, 2023.
- [TBA20] M. Tarek, F. Brissette, and R. Arsenault. Evaluation of the era5 reanalysis as a potential reference dataset for hydrological modelling over north america. *Hydrology and Earth System Sciences*, 24:2527–2544, 2020.

- [TFB⁺18] P. Termonia, C. Fischer, E. Bazile, F. Bouyssel, R. Brožková, P. Bénard, B. Bochenek, D. Degrauwe, M. Derková, R. El Khatib, R. Hamdi, J. Mašek, P. Pottier, N. Pristov, Y. Seity, P. Smolíková, O. Španiel, M. Tudor, Y. Wang, C. Wittmann, and A. Joly. The aladin system and its canonical model configurations arome cy41t1 and alaro cy40t1. *Geoscientific Model Development*, 11(1):257–281, 2018.
- [TFC25] Elena Tomasi, Gabriele Franch, and Marco Cristoforetti. Can ai be enabled to perform dynamical downscaling? a latent diffusion model to mimic kilometer-scale cosmo5.0.clm9 simulations. *Geoscientific Model Development*, 18(6):2051–2078, 2025.
- [TFS15] Kevin E Trenberth, John T Fasullo, and Theodore G Shepherd. Attribution of climate extreme events. *Nature climate change*, 5(8):725–730, 2015.
- [TG10] Thordis L Thorarinsdottir and Tilmann Gneiting. Probabilistic forecasts of wind speed: ensemble model output statistics by using heteroscedastic censored regression. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 173(2):371–388, 2010.
- [TH84] Roger Y Tsai and Thomas S Huang. Multiframe image restoration and registration. *Multiframe image restoration and registration*, 1:317–339, 1984.
- [TLLG17] Tong Tong, Gen Li, Xiejie Liu, and Qinquan Gao. Image super-resolution using dense skip connections. In *Proceedings of the IEEE international conference on computer vision*, pages 4799–4807, 2017.
- [TLZT23] Song Tang, Chuang Li, Pu Zhang, and RongNian Tang. Swinlstm: Improving spatiotemporal prediction accuracy using swin transformer and lstm. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13470–13479, 2023.
- [TSM12] Karl E. Taylor, Ronald J. Stouffer, and Gerald A. Meehl. An overview of CMIP5 and the experiment design. *Bulletin of the American Meteorological Society*, 93(4):485–498, 2012.
- [TSW⁺22] Ruian Tie, Chunxiang Shi, Gang Wan, Xingjie Hu, Lihua Kang, and Lingling Ge. Cldassd: reconstructing fine textures of the temperature field using super-resolution technology. *Advances in Atmospheric Sciences*, 39(1):117–130, 2022.
- [TXY⁺25] Siwei Tu, Jingyi Xu, Weidong Yang, Lei Bai, and Ben Fei. Mods: Multi-source observations conditional diffusion model for meteorological state downscaling. *arXiv preprint arXiv:2506.14798*, 2025.
- [VGBM21] A. Verrelle, M. Glinton, E. Bazile, and Patrick Le Moigne. Cerra-land : A new land surface reanalysis at 5.5 km resolution over europe. 2021.
- [VKG⁺17] Thomas Vandal, Evan Kodra, Sangram Ganguly, Andrew Michaelis, Ramakrishna Nemani, and Auroop R Ganguly. Deepspd: Generating high resolution climate change projections through single image super-resolution, 2017.
- [VSP⁺17a] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [VSP⁺17b] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.
- [WBB⁺23] Zhong Yi Wan, Ricardo Baptista, Anudhyan Boral, Yi-Fan Chen, John Anderson, Fei Sha, and Leonardo Zepeda-Núñez. Debias coarsely, sample conditionally: Statistical downscaling through optimal transport and probabilistic diffusion models. *Advances in Neural Information Processing Systems*, 36:47749–47763, 2023.

- [WBD⁺15] NP Wedi, P Bauer, W Denoninck, M Diamantakis, M Hamrud, C Kuhnlein, S Malardel, K Mogensen, G Mozdzyński, and PK Smolarkiewicz. The modelling infrastructure of the integrated forecasting system: Recent advances and future challenges. 2015.
- [WCB⁺22] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17683–17693, 2022.
- [WD13] Robert L Wilby and Christian W Dawson. The statistical downscaling model: insights from one decade of application. *International Journal of Climatology*, 33(7):1707–1719, 2013.
- [WDC20] Jonathan A Weyn, Dale R Durran, and Rich Caruana. Improving data-driven global weather prediction using deep convolutional neural networks on a cubed sphere. *Journal of Advances in Modeling Earth Systems*, 12(9):e2020MS002109, 2020.
- [WDC⁺21] Jonathan A Weyn, Dale R Durran, Rich Caruana, and Nathaniel Cresswell-Clay. Sub-seasonal forecasting with a large ensemble of deep-learning weather prediction models. *Journal of Advances in Modeling Earth Systems*, 13(7):e2021MS002502, 2021.
- [WH18] Yuxin Wu and Kaiming He. Group normalization. 3 2018.
- [WM24] Robbie A Watt and Laura A Mansfield. Generative diffusion-based downscaling for climate. *arXiv preprint arXiv:2404.17752*, 2024.
- [WMS⁺20] A. Wootten, E. Massoud, A. Sengupta, D. Waliser, and Huikyo Lee. The effect of statistical downscaling on the weighting of multi-model ensembles of precipitation. *Climate*, 2020.
- [WTL⁺21] Fang Wang, Di Tian, Lisa Lowe, Latif Kalin, and John Lehrter. Deep learning for daily precipitation and temperature downscaling. *Water Resources Research*, 57(4):e2020WR029308, 2021.
- [WWC⁺98] Robert L Wilby, TML Wigley, D Conway, PD Jones, BC Hewitson, J Main, and DS Wilks. Statistical downscaling of general circulation model output: A comparison of methods. *Water resources research*, 34(11):2995–3008, 1998.
- [WYW⁺18] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018.
- [WZD⁺23] Wei Wei, Shan Zou, Weili Duan, Yaning Chen, Shuai Li, and Yiqi Zhou. Spatiotemporal variability in extreme precipitation and associated large-scale climate mechanisms in central asia from 1950 to 2019. *Journal of Hydrology*, 620:129417, 2023.
- [XYC⁺07] P. Xie, A. Yatagai, M. Chen, T. Hayasaka, Y. Fukushima, C. Liu, and S. Yang. A gauge-based analysis of daily precipitation over east asia. *Journal of Hydrometeorology*, 8:607–626, 2007.
- [YHA⁺21] Peng Yuan, Addisu Hunegnaw, Fadwa Alshawaf, Joseph Awange, Anna Klos, Felix Norman Teferle, and Hansjörg Kutterer. Feasibility of era5 integrated water vapor trends for climate change analysis in continental europe: An evaluation with gps (1994–2019) by considering statistical significance. *Remote Sensing of Environment*, 260:112416, 2021.
- [YHGH⁺24] Qidong Yang, Alex Hernandez-Garcia, Paula Harder, Venkatesh Ramesh, Prasanna Sattigeri, Daniela Szwarcman, Campbell D Watson, and David Rolnick. Fourier neural operators for arbitrary resolution climate data downscaling. *Journal of Machine Learning Research*, 25(420):1–30, 2024.

- [Yil23] Meric Yilmaz. Accuracy assessment of temperature trends from era5 and era5-land. *Science of The Total Environment*, 856:159182, 2023.
- [YYWS24] Fan Yang, Qiaolin Ye, Kai Wang, and Le Sun. Successful precipitation downscaling through an innovative transformer-based model. *Remote Sensing*, 16(22):4292, 2024.
- [YZH⁺25] Yifeng Yang, Hengqian Zhao, Xiadan Huangfu, Zihan Li, and Pan Wang. Vit-isrgan: A high-quality super-resolution reconstruction method for multispectral remote sensing images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 18:3973–3988, 2025.
- [ZCL⁺24] Xiaohui Zhong, Lei Chen, Hao Li, Jun Liu, Xu Fan, Jie Feng, Kan Dai, Jing-Jia Luo, Jie Wu, and Bo Lu. Fuxi-ens: A machine learning model for medium-range ensemble weather forecasting. *arXiv preprint arXiv:2405.05925*, 2024.
- [ZDC⁺24] Xiaohui Zhong, Fei Du, Lei Chen, Zhibin Wang, and Hao Li. Investigating transformer-based models for spatial downscaling and correcting biases of near-surface temperature and wind-speed forecasts. *Quarterly Journal of the Royal Meteorological Society*, 150(758):275–289, 2024.
- [ZGS⁺25] Lin Zhang, Xinyu Guo, Hongkun Sun, Weigang Wang, and Liwei Yao. Alternate encoder and dual decoder cnn-transformer networks for medical image segmentation. *Scientific Reports*, 15(1):8883, 2025.
- [ZLDQ19] Wenlong Zhang, Yihao Liu, Chao Dong, and Yu Qiao. Ranksrgan: Generative adversarial networks with ranker for image super-resolution. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3096–3105, 2019.
- [ZLL⁺18] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 286–301, 2018.
- [ZLZ⁺21a] Ruqing Zhang, Lu Li, Yeping Zhang, Feini Huang, Jianduo Li, Wei Liu, Taoning Mao, Zili Xiong, and W. Shangguan. Assessment of agricultural drought using soil water deficit index based on era5-land soil moisture data in four southern provinces of china. *Agriculture*, 11:411, 2021.
- [ZLZ⁺21b] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021.
- [ZLZ⁺22] Wanfeng Zheng, Qiang Li, Guoxin Zhang, Pengfei Wan, and Zhongyuan Wang. Ittr: Unpaired image-to-image translation with transformers. *arXiv preprint arXiv:2203.16015*, 2022.
- [ZRRB15] Günther Zängl, Daniel Reinert, Pilar Rípodas, and Michael Baldauf. The icon (icosahedral non-hydrostatic) modelling framework of dwd and mpi-m: Description of the non-hydrostatic dynamical core. *Quarterly Journal of the Royal Meteorological Society*, 141(687):563–579, 2015.
- [ZSV20] Harald Zandler, Thomas Senftl, and Kim André Vanselow. Reanalysis datasets outperform other gridded climate products in vegetation change analysis in peripheral conservation areas of central asia. *Scientific Reports*, 10(1):22446, 2020.
- [ZXYW25] Yicheng Zhang, Dexi Xia, Fei Yuan, and Zhengjun Wang. Typhoon forecasting based on remote sensing satellite data and two-stage super-resolution diffusion (tsrd) model. In *Journal of Physics: Conference Series*, volume 3055, page 012012. IOP Publishing, 2025.