Diffusion generative models for weather forecasting

Relatore:
Chiar.mo Prof. **Andrea Asperti**

Correlatore:
Chiar.mo Dott. **Fabio Merizzi**

Presentata da:
Alberto Paparella

12 Ottobre 2023

Alma Mater Studiorum · Università di Bologna
Corso di Laurea Magistrale in Informatica

Sessione II
Anno Accademico 2022-23

Sommario

Introduction

Diffusion generative models

Diffusion generative models for precipitation nowcasting

Experiments

Bibliography

Backup slides

Introduction

Introduction

- The historical **datasets** commonly employed for weather forecasts are typically structured in a **regular spatial grid format** which closely resembles **images**, with each **weather variable** akin to a map or, when considering the temporal axis, as a **video**.
- Several classes of **generative models**, such as Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and Denoising Diffusion Models (DDMs), have demonstrated their effectiveness in tackling the **next-frame prediction** problem.
- Consequently, it is only natural to assess their performance in the context of **weather prediction benchmarks**.

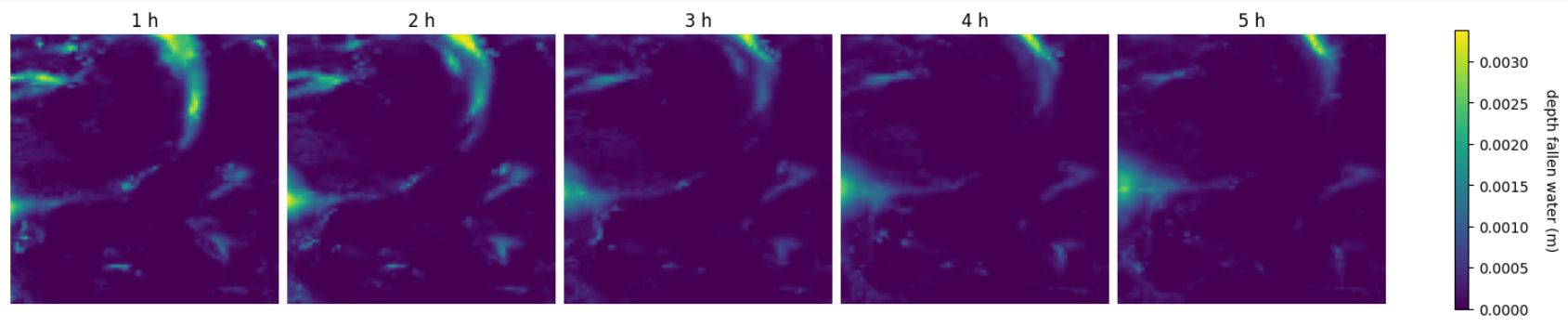


Figure 1: Example of precipitation data from the **ERA5** [1] dataset.

Introduction

- **DDMs**, in particular, hold strong appeal in this domain due to the inherently **probabilistic nature** of weather forecasting, aiming to model the **probability distribution** of **weather indicators**.
- This thesis is dedicated to investigating the application of **diffusion models** in the realm of **weather forecasting**.
- To achieve this, a specific subset of the **ERA5** [1] dataset has been leveraged, encompassing hourly data for Western Europe spanning the years 2016 to 2021.
- Within this context, the effectiveness of diffusion models has been rigorously assessed in the challenging domain of **precipitation nowcasting** in direct comparison to the well-established **U-Net models** documented in the existing literature.

Diffusion generative models

Diffusion generative models

- Essentially, a **diffusion model** leverages a single network to effectively **eliminate noise from images**, with the flexibility to parametrically adjust the level of noise to be removed.
- This network is subsequently employed to produce **novel** samples by **iteratively diminishing** noise in a designated *noisy* image.
- This iterative process commences from an entirely random noise configuration and is conventionally known as **reverse diffusion**.
- Its objective is to effectively *invert* the direct **diffusion process**, where noise is incrementally added to the source image.

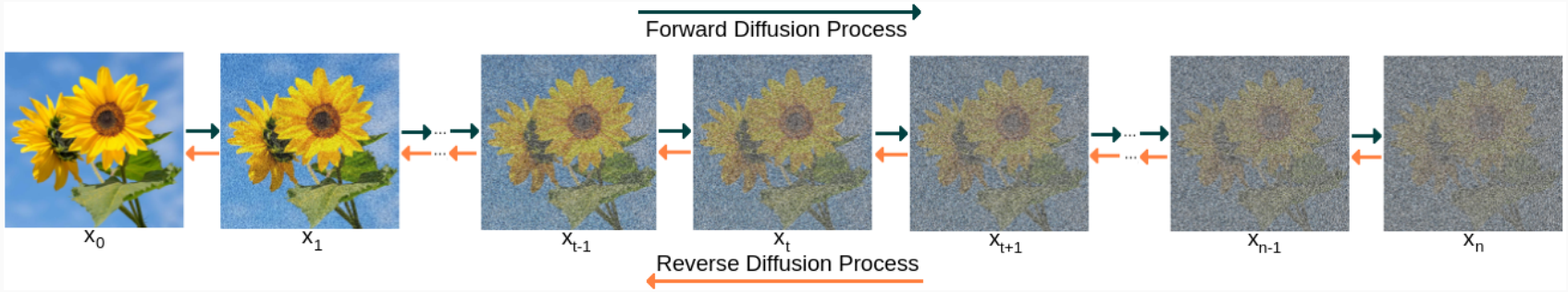


Figure 2: Forward diffusion process (from left to right) and **reverse** diffusion process (from right to left).

Denoising Diffusion Probabilistic Models (DDPM)

- From a mathematical standpoint, the **objective** of **generative models** is to identify a **parameter vector** θ utilized to shape a distribution $p_\theta(x_0)$, characterized by a **neural network**, which **closely approximate** and model the original data distribution $q(x_0)$.
- **Denoising Diffusion Probabilistic Models (DDPM)** [2] posit that the generative distribution $p_\theta(x_0)$ follows a specific form defined as:

$$p_\theta(x_0) = \int p_\theta(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) dx_{1:T} \tag{1}$$

characterizing the generative process by modeling the joint distribution over a sequence of **time steps**, where $p_\theta(x_T)$ represents the initial distribution, and the subsequent terms capture the conditional transitions from x_t to x_{t-1}.

Denoising Diffusion Implicit Models (DDIM)

- In **Denoising Diffusion Implicit Models (DDIM)** [3], the authors introduce a non-Markovian diffusion process defined as:

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

- The term $q_\sigma(x_T|x_0)$ is described by a Gaussian distribution.
- In practical terms, a **neural network** $\epsilon_\theta^{(t)}(x_t, \alpha_t)$ is trained to effectively map a given pair of inputs, x_t and α_t (representing the **noise rate**), into an estimate of the noise component, ϵ_t, that when added to x_0 facilitates the construction of the next time step, x_t.
- The **loss** function can be interpreted as the **weighted mean squared error** between the predicted noise and the actual noise.

Conditioning

- The process of generating data often requires a way to control the sample creation process to influence the final output: this procedure is referred to as **conditioned** or **guided** diffusion.
- In mathematical terms, **guidance** entails the conditioning of a prior data distribution $p(x)$, with specific constraints (e.g., class labels, image/text embeddings), giving a conditional distribution $p(x|y)$.
- To transform a diffusion model p_θ into a conditional diffusion model, we introduce conditioning information y at each step of the diffusion process, yielding the following formulation:

$$p_\theta(x_{0:T}|y) = p_\theta(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t, y) \tag{3}$$

- The learning of this distribution typically follows one of two approaches: the first approach relies on an auxiliary classifier [4]; the second approach operates without a classifier, offering an alternative methodology for modeling conditional diffusion.

Diffusion generative models for precipitation nowcasting

The DDIM architecture: denoising

- The **denoising network** $\epsilon_\theta(x_t, \alpha_t)$ consists of a **U-Net** architecture.
- It takes as input the *noisy* images, denoted as x_t, along with a corresponding **noise variance**, α_t, and aims to accurately estimate the level of noise affecting the image.

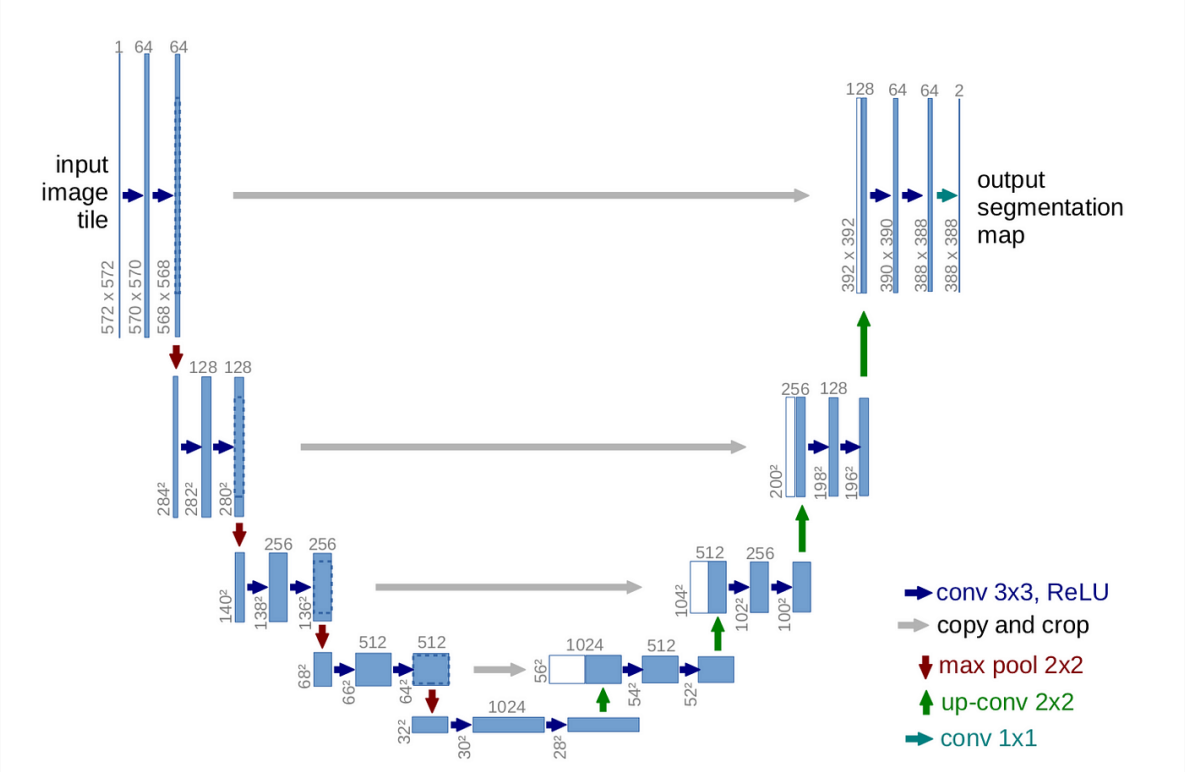


Figure 3: U-net architecture. Image taken from [5].

The DDIM architecture: conditioning

- **Conditioning** is achieved in a **classifier-free** manner [6] by directly appending the conditioning frames to the noisy images along the channel axis.

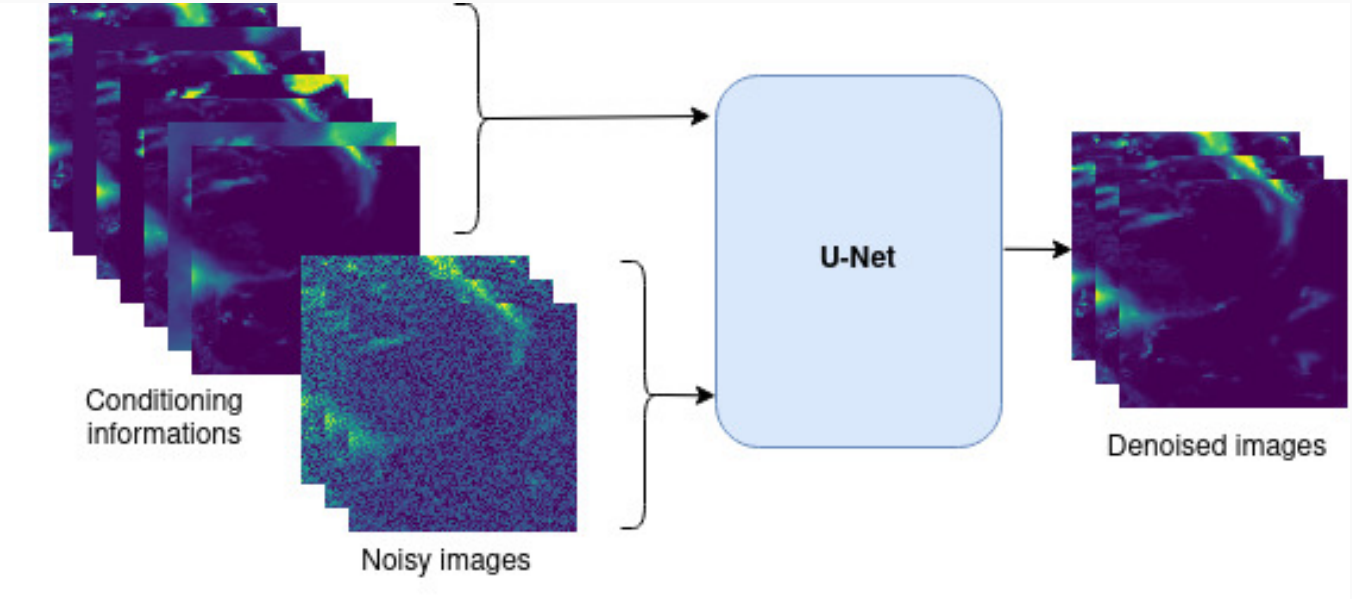


Figure 4: Conditioning is implemented by stacking **additional information** alongside the channel axis in the denoising network.

A novel approach: Generative Ensemble Diffusion (GED)

- The proposed approach, referred to as **Generative Ensemble Diffusion (GED)**, harnesses a **diffusion model** to generate a diverse **set of potential weather scenarios**.
- These scenarios are subsequently **amalgamated** into a probable prediction through the application of a sophisticated **post-processing network**.
- In direct contrast to recent deep learning models, the GED approach consistently demonstrated superior performance across multiple performance metrics, underscoring its significant advancement in the field of weather forecasting.

A novel approach: Generative Ensemble Diffusion (GED)

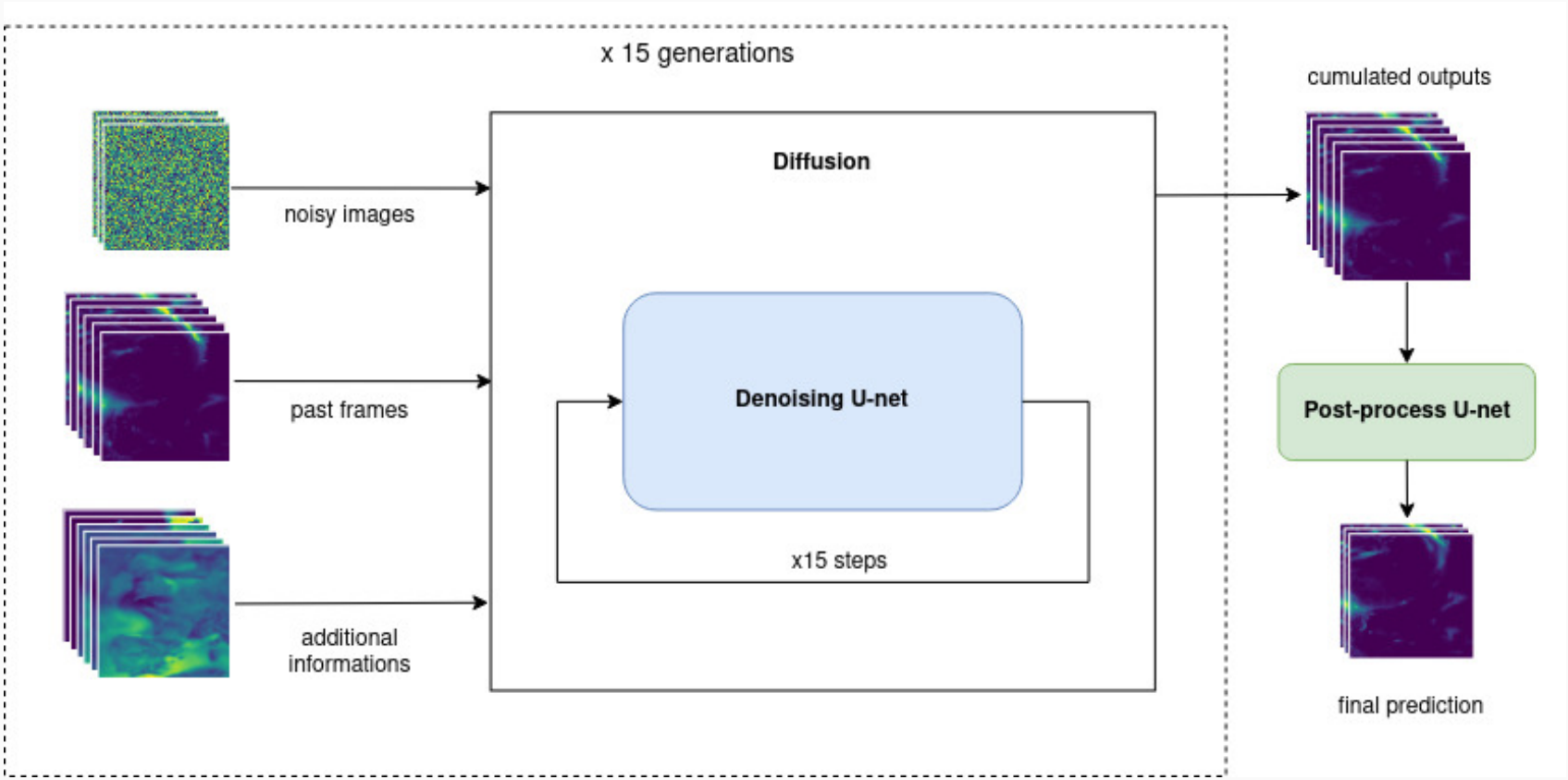


Figure 5: Generative Ensemble Diffusion (GED) prediction structure, showing the **multiple denoising cycles** and the final **post-processing** step.

Experiments

Dataset description and pre-processing

- The proposed **Generative Ensemble Diffusion (GED)** model has been compared with the state-of-the-art **Weather Fusion UNet (WF-UNet)** model introduced in [7].
- The chosen dataset comprises precipitation and wind radar images covering 14 European countries, ranging from January 2016 to December 2021, featuring a temporal resolution of 1 hour and a spatial resolution of 31 km^2, relying on the **ERA5** dataset [1].
- It's worth noting that precipitation tends to exhibit sparsity, often being absent in the analyzed region, introducing a bias toward predicting zero values [8].
- To address this issue, sequences have been filtered such that a certain percentage of rain is present, simulating the conditions outlined in the **EU-50** and **EU-20** datasets as specified in [7].

Additional features

- Additional features encompassed **wind speed**, derived from both northerly and easterly wind components, the **land-sea mask**, a **geopotential map**, and a **sinusoidal time embedding**.

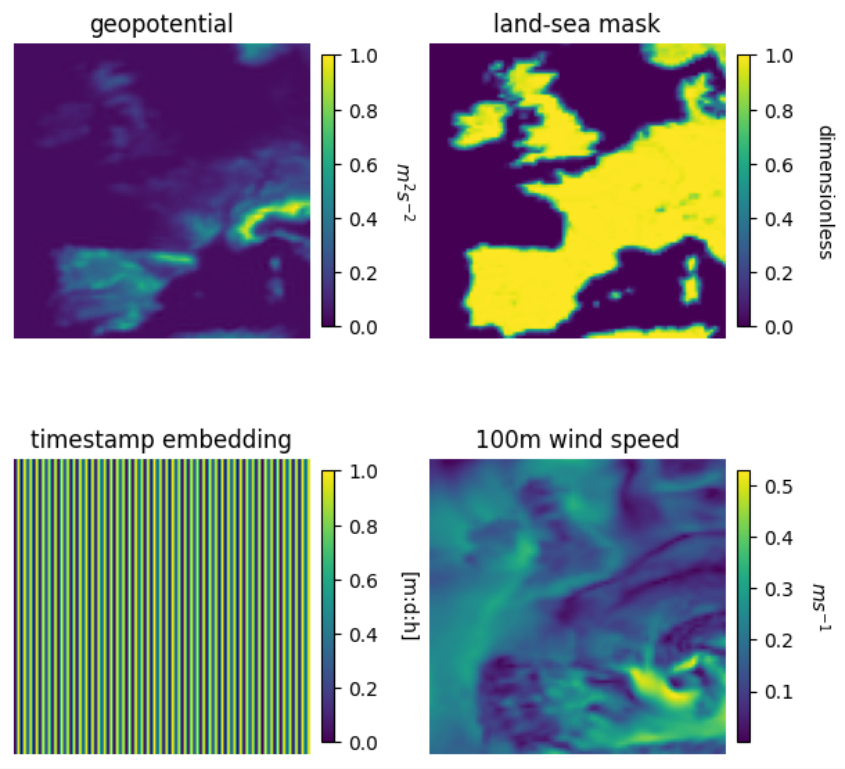


Figure 6: Visual example of the **additional features**.

Training and evaluation

- The diffusion model was trained with a batch size of 2 throughout 40 epochs using data from 2016 to 2020.
- For optimization, the **AdamW** algorithm was employed, utilizing a learning rate of 1e-04 and a weight decay of 1e-05; furthermore, a fine-tuning phase was conducted, encompassing 10 epochs with a reduced learning rate of 1e-05 and a weight decay of 1e-06.
- As with the reference model, sequences composed of more than 50% non-rain values were excluded from the training process.
- **Mean Absolute Error (MAE)** has been used as a loss function and applied to the noise difference.
- The evaluation was carried out using data from the test year of 2021, maintaining a fixed number of 15 diffusion steps and assessing performance using the **Mean Squared Error (MSE)** metric:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4}$$

Training and evaluation

- The **post-processing U-Net** takes fifteen distinct generative outputs from the diffusion model as input, each containing predictions for the subsequent three hours, and produces an output comprising three images, each predicting the rainfall for one of the upcoming three hours.
- The training of the **post-processing U-Net** utilizes **AdamW** as the optimizer with a learning rate set at 1e-4 and a weight decay of 1e-5.
- The loss function employed is the **Mean Squared Error (MSE)**, calculated as the discrepancy between the predicted images and their corresponding ground truth.

MSE values and additional metrics for EU-20 dataset

Model	MSE	Accuracy	Precision	Recall
		1 hour ahead		
WF-UNet	2.67e-04	0.933	0.790	0.847
Single Diffusion	2.86e-04	0.911	0.754	0.888
GED (mean)	2.25e-04	0.930	0.786	0.901
GED (postprocess)	2.03e-04	0.923	0.798	0.909
		2 hour ahead		
WF-UNet	4.87e-04	0.895	0.664	0.807
Single Diffusion	4.69e-04	0.886	0.705	0.831
GED (mean)	3.93e-04	0.900	0.731	0.848
GED (postprocess)	3.53e-04	0.898	0.742	0.849
		3 hour ahead		
WF-UNet	6.34e-04	0.877	0.626	0.736
Single Diffusion	6.10e-04	0.853	0.638	0.758
GED (mean)	5.20e-04	0.880	0.689	0.801
GED (postprocess)	4.70e-04	0.891	0.701	0.796

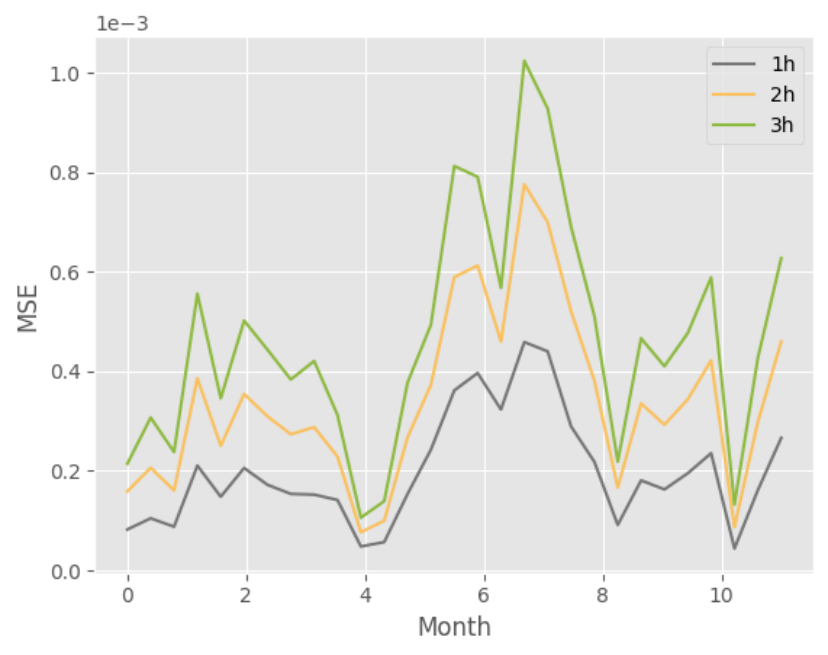
Table 1: Results comparison on the **EU-20** dataset.

MSE values and additional metrics for EU-50 dataset

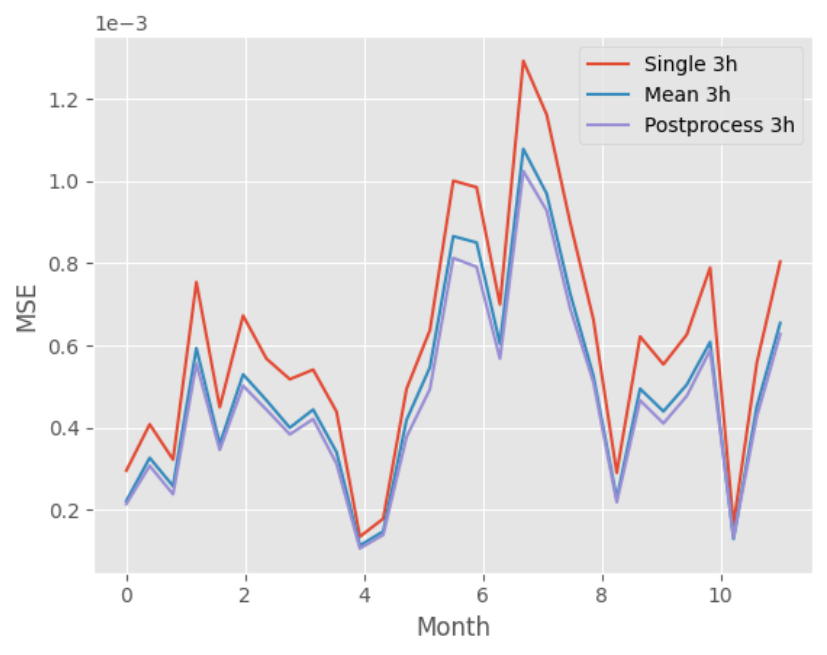
Model	MSE	Accuracy	Precision	Recall
		1 hour ahead		
WF-UNet	2.50e-04	0.921	0.803	0.849
Single Diffusion	2.59e-04	0.915	0.767	0.882
GED (mean)	2.02e-04	0.924	0.782	0.885
GED (postprocess)	1.99e-04	0.913	0.803	0.907
		2 hour ahead		
WF-UNet	4.62e-04	0.877	0.684	0.813
Single Diffusion	4.51e-04	0.875	0.699	0.844
GED (mean)	3.59e-04	0.882	0.711	0.862
GED (postprocess)	3.40e-04	0.878	0.724	0.860
		3 hour ahead		
WF-UNet	6.31e-04	0.855	0.647	0.743
Single Diffusion	6.03e-04	0.848	0.672	0.801
GED (mean)	4.92e-04	0.856	0.701	0.828
GED (postprocess)	4.65e-04	0.861	0.706	0.821

Table 2: Results comparison on the **EU-50** dataset.

Experiments



(a)



(b)

Figure 7: Single Diffusion results for the year 2021 on EU50, depicting significant score variations depending on the month of the year. (a) illustrates the **month-wise dissimilarity** in scores for each of the three predicted hours. In (b), we observe that the dissimilarity remains consistent across predictions computed with **Single Diffusion, GED (mean)**, and **GED (post-process)**.

Precipitation nowcasting with generative diffusion models[9]

- Submitted to **Neural Computing and Applications (NCAA)**
- Pre-print available on arXiv

Precipitation nowcasting with generative diffusion models

Andrea Asperti[1*†], Fabio Merizzi[1*†], Alberto Paparella[1†], Giorgio Pedrazzi[2], Matteo Angelinelli[2] and Stefano Colamonaco[1]

[1*]Department of Informatics: Science and Engineering (DISI), University of Bologna, Mura Anteo Zamboni 7, Bologna, 40126, Italy .
[2]HPC Department Cineca, Magnanelli 6/3, Casalecchio di Reno (BO), 40033, Italy .

*Corresponding author(s). E-mail(s): andrea.asperti@unibo.it; fabio.merizzi@unibo.it;
Contributing authors: alberto.paparella2@studio.unibo.it; g.pedrazzi@cineca.it;
m.angelinelli@cineca.it; stefano.colamonaco@studio.unibo.it;
†These authors contributed equally to this work.

Thanks for your attention!

Bibliography

Bibliography i

Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, Adrian Simmons, Cornel Soci, Saleh Abdalla, Xavier Abellan, Gianpaolo Balsamo, Peter Bechtold, Gionata Biavati, Jean Bidlot, Massimo Bonavita, Giovanna De Chiara, Per Dahlgren, Dick Dee, Michail Diamantakis, Rossana Dragani, Johannes Flemming, Richard Forbes, Manuel Fuentes, Alan Geer, Leo Haimberger, Sean Healy, Robin J. Hogan, Elías Hólm, Marta Janisková, Sarah Keeley, Patrick Laloyaux, Philippe Lopez, Cristina Lupu, Gabor Radnoti, Patricia de Rosnay, Iryna Rozum, Freja Vamborg, Sebastien Villaume, and Jean-Noël Thépaut.
The era5 global reanalysis.
Quarterly Journal of the Royal Meteorological Society, 146(730):1999–2049, 2020.

Bibliography ii

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.

Jiaming Song, Chenlin Meng, and Stefano Ermon.
Denoising Diffusion Implicit Models.
arXiv e-prints, page arXiv:2010.02502, October 2020.

Bibliography iii


Augustus Odena, Christopher Olah, and Jonathon Shlens.
Conditional image synthesis with auxiliary classifier gans.
In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 2642–2651, 2017.


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.


Jonathan Ho and Tim Salimans.
Classifier-free diffusion guidance.
CoRR, abs/2207.12598, 2022.

Bibliography iv


Christos Kaparakis and Siamak Mehrkanoon.
Wf-unet: Weather fusion unet for precipitation nowcasting.
CoRR, abs/2302.04102, 2023.


Kevin Trebing, Tomasz Stanczyk, and Siamak Mehrkanoon.
Smaat-unet: Precipitation nowcasting using a small attention-unet architecture, 2021.


Andrea Asperti, Fabio Merizzi, Alberto Paparella, Giorgio Pedrazzi, Matteo Angelinelli, and Stefano Colamonaco.
Precipitation nowcasting with generative diffusion models, 2023.

Bibliography v


Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra.
Stochastic backpropagation and approximate inference in deep generative models.
In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1278–1286. JMLR.org, 2014.


Diederik P. Kingma and Max Welling.
An introduction to variational autoencoders.
Found. Trends Mach. Learn., 12(4):307–392, 2019.


Andrea Asperti, Davide Evangelista, and Elena Loli Piccolomini.
A survey on variational autoencoders from a green AI perspective.
SN Comput. Sci., 2(4):301, 2021.

Bibliography vi

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli.
Deep unsupervised learning using nonequilibrium thermodynamics.
37:2256–2265, 2015.

Prafulla Dhariwal and Alexander Quinn Nichol.
Diffusion models beat gans on image synthesis.
In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 8780–8794, 2021.

Backup slides

Denoising Diffusion Probabilistic Models (DDPM) - Details

Denoising Diffusion Probabilistic Models (DDPM) [2] posit that the generative distribution $p_\theta(x_0)$ follows a specific form defined as:

$$p_\theta(x_0) = \int p_\theta(x_{0:T}) dx_{1:T} \tag{5}$$

Here, the time horizon extends to $T > 0$, and $p_\theta(x_{0:T})$ can be further expressed as:

$$p_\theta(x_{0:T}) = p_\theta(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) \tag{6}$$

In this formulation, DDPM characterizes the generative process by modeling the joint distribution over a sequence of time steps, where $p_\theta(x_T)$ represents the initial distribution, and the subsequent terms capture the conditional transitions from x_t to x_{t-1} as the process evolves.

Denoising Diffusion Probabilistic Models (DDPM) - Details

Training in diffusion models traditionally relies on a **variational lower bound of the negative log-likelihood**:

$$
\begin{aligned}
&-\log p_\theta(x_0) \\
&\quad \leq -\log p_\theta(x_0) + D_{\mathsf{KL}}(q(x_{1:T}|x_0) \| p_\theta(x_{1:T}|x_0)) \\
&\quad = -\log p_\theta(x_0) + \mathbb{E}_q\left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})/p_\theta(x_0)}\right] \\
&\quad = -\log p_\theta(x_0) + \mathbb{E}_q\left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})} + \log p_\theta(x_0)\right] \\
&\quad = \mathbb{E}_q\Big[\log q(x_{1:T}|x_0) - p_\theta(x_{0:T})\Big] = \mathcal{L}(\theta) \qquad (7)
\end{aligned}
$$

What sets diffusion models apart from typical latent variable models like Variational Autoencoders (VAEs) [10, 11, 12] is that they employ a fixed, non-trainable inference procedure denoted as $q(x_{1:T}|x_0)$.

Denoising Diffusion Implicit Models (DDIM) - Details

In **Denoising Diffusion Implicit Models (DDIM)** [3], the authors introduce a non-Markovian diffusion process defined as:

$$q_\sigma(x_{1:T}|x_0) = q_\sigma(x_T|x_0) \prod_{t=2}^{T} q_\sigma(x_{t-1}|x_t, x_0) \tag{8}$$

Here, the term $q_\sigma(x_T|x_0)$ is described by a Gaussian distribution:

$$q_\sigma(x_T|x_0) = \mathcal{N}(x_T|\sqrt{\alpha_T}x_0, (1-\alpha_T)\cdot I) \tag{9}$$

Additionally, the conditional distribution $q_\sigma(x_{t-1}|x_t, x_0)$ takes the form:

$$q_\sigma(x_{t-1}|x_t, x_0) = \mathcal{N}\left(x_{t-1}\middle|\mu_{\sigma_t}(x_0, \alpha_{t-1}); \sigma_t^2 \cdot I\right) \tag{10}$$

with

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

Denoising Diffusion Implicit Models (DDIM) - Details

The choice of $q(x_{t-1}|x_t, x_0)$ is strategically made to fulfill two critical aspects of the DDPM diffusion process: the Gaussian nature of $q(x_{t-1}|x_t, x_0)$ when conditioned on x_0, and the ability to recover the same marginal distribution as in DDPM, where:

$$q_\sigma(x_t|x_0) = \mathcal{N}(x_t|\sqrt{\alpha_t}x_0; (1-\alpha_t)\cdot I. \tag{11}$$

This property allows us to represent x_t as a linear combination of x_0 and a noise variable $\epsilon_t \sim \mathcal{N}(\epsilon_t|0; I)$:

$$x_t = \sqrt{\alpha_t}x_0 + \sqrt{1-\alpha_t}\epsilon_t. \tag{12}$$

Next, our task is to define a trainable generative process denoted as $p_\theta(x_{0:T})$, where the conditional distribution $p_\theta(x_{t-1}|x_t)$ is crafted to incorporate the structure from $q_\sigma(x_{t-1}|x_t, x_0)$. The concept is that when provided with a noisy observation x_t, the process begins by predicting x_0 and then employs this prediction to derive x_{t-1} according to equation 10.

Denoising Diffusion Implicit Models (DDIM) - Details

In practical terms, a neural network denoted as $\epsilon_\theta^{(t)}(x_t, \alpha_t)$ is trained to effectively map a given pair of inputs, x_t and α_t (representing the noise rate), into an estimate of the noise component, ϵ_t. This estimated noise, when added to x_0, facilitates the construction of the next time step, x_t. Consequently, the **conditional distribution** $p_\theta(x_{t-1}|x_t)$ is approximated as a Dirac delta function $\delta_{f_\theta^{(t)}}$, where:

$$f_\theta^{(t)}(x_t, \alpha_t) = \frac{x_t - \sqrt{1-\alpha_t}\epsilon_\theta(x_t, \alpha_t)}{\sqrt{\alpha_t}}. \tag{13}$$

Using $f_\theta^{(t)}(x_t, \alpha_t)$ as an approximation of x_0 at timestep t, x_{t-1} is subsequently calculated as follows:

$$x_{t-1} = \sqrt{\alpha_{t-1}} \cdot f_\theta^{(t)}(x_t, \alpha_t) + \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \epsilon_\theta(x_t, \alpha_t) \tag{14}$$

.

Denoising Diffusion Implicit Models (DDIM) - Details

Regarding the **loss** function, the term in equation 7 can be further decomposed into the sum of the following terms [13]:

$$L_\theta = L_T + L_{t-1} + \cdots + L_0 \tag{15}$$

where

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

This breakdown of the loss function provides a more granular perspective on the optimization process.

Denoising Diffusion Implicit Models (DDIM) - Details

All the aforementioned distributions take the form of Gaussians, facilitating the calculation of their KL divergences in a closed-form manner, following a Rao-Blackwellized approach. After a series of manipulations, we arrive at the following formulation:

$$L_t = \mathbb{E}t \sim [1, T], x_0, \epsilon_t \Big[\gamma_t |\epsilon_t - \epsilon\theta(x_t, t)|^2 \Big] \tag{16}$$

This expression can be interpreted as the **weighted mean squared error** between the predicted noise and the actual noise at time t. In practice, the weighting parameters are often omitted, as experimental evidence suggests that the training process tends to perform better without them.

Denoising Diffusion Implicit Models (DDIM) - Details

Algorithm 1 Training

1: **repeat**
2: $x_0 \sim q(x_0)$
3: $t \sim$Uniform(1,..,T)
4: $\epsilon \sim \mathcal{N}(0; I)$
5: $x_t = \sqrt{\alpha_t} x_b + \sqrt{1-\alpha_t}\epsilon$
6: Backpropagate on $||\epsilon - \epsilon_\theta(x_t, \alpha_t)||^2$
7: **until** converged

Denoising Diffusion Implicit Models (DDIM) - Details

Sampling is an iterative process, starting from a purely noisy image $x_T \sim \mathcal{N}(0, I)$. The denoised version of the image at timestep t is obtained using equation 14.

Algorithm 2 Sampling

1: $x_T \sim \mathcal{N}(0, I)$
2: **for** $t = T, ..., 1$ **do**
3: $\quad \epsilon = \epsilon_\theta(x_a, x_t, \alpha_t)$
4: $\quad \tilde{x}_0 = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_t}}\epsilon)$
5: $\quad x_{t-1} = \sqrt{\alpha_{t-1}}\tilde{x}_0 + \sqrt{1-\alpha_{t-1}}\epsilon$
6: **end for**

Diffusion classifier guidance

- The concept underpinning classifier guidance is as follows: the objective is to learn the gradient of the logarithm of the conditional density $p_\theta(x_t|y)$.
- By applying Bayes' rule, this can be expressed as:

$$\nabla_{x_t} \log p_\theta(x_t|y) = \nabla_{x_t} \log \left(\frac{p_\theta(y|x_t) \cdot p_\theta(x_t)}{p_\theta(y)} \right) \tag{17}$$

- Given that the gradient operator solely pertains to x_t, the term $p_\theta(y)$ can be eliminated; after simplification, we arrive at:

$$\begin{aligned} \nabla_{x_t} \log p_\theta(x_t|y) = & \nabla_{x_t} \log p_\theta(x_t) \\ & + s \cdot \nabla_{x_t} \log p_\theta(y|x_t) \end{aligned} \tag{18}$$

  Here, the scalar term s assumes the role of modulating the strength of the guidance component.

Diffusion classifier guidance

- As elucidated in [14], one approach to guide the diffusion process during generation entails the utilization of a **classifier**, denoted as $f_\phi(y|x_t, t))$.
- This method involves the training of a classifier $f_\phi(y|x_t, t)$ using a *noisy* image x_t to predict its corresponding **class label** y.
- Subsequently, the gradient $\nabla_x \log f_\phi(y|x_t)$ can be harnessed to steer the diffusion sampling process towards the targeted conditioning information y through adjustments to the noise prediction.
- This technique is particularly well-suited for scenarios involving discrete labels.

Classifier-Free Diffusion Guidance

- The concept of **conditioned diffusion**, devoid of reliance on an external classifier, has been extensively explored in [6].
- This approach involves the training of both a **conditional** diffusion model, denoted as $\epsilon_\theta(x_t, t, y)$, and an **unconditional model**, denoted as $\epsilon_\theta(x_t, t, 0)$.
- In many cases, the same neural network architecture can serve both models: during training, the class label y is randomly set to 0, thereby exposing the model to both conditional and unconditional scenarios.
- The estimated noise, represented as $\hat{\epsilon}_\theta(x_t|t, y)$ at time step t, subsequently emerges as a judiciously **weighted combination** of the conditional and unconditional predictions:

$$\hat{\epsilon}_\theta(x_t, t, y) = \epsilon_\theta(x_t, t, y) + s \cdot \epsilon_\theta(x_t, t) \tag{19}$$

U-Net [5]

- The network structure consists of a **downsampling** sequence of layers, succeeded by an **upsampling** sequence, all while incorporating **skip connections** linking layers of equivalent dimensions.
- Typically, the configuration of the U-Net is defined by specifying the number of **downsampling blocks** and the **channel count** for each block.
- The upsampling component mirrors a symmetrical pattern, and the spatial dimensions align with the image resolution.
- The entire structure of a U-Net can be encoded in a single list (e.g., [32, 64, 96, 128]), denoting both the number of downsampling blocks (in this case, 4) and the associated channel counts, which typically increases as the spatial dimensions decrease.
- For the experiments, a U-Net configuration of [64, 128, 256, 384] has been used, which empirically proved to be the most effective.

U-Net [5]

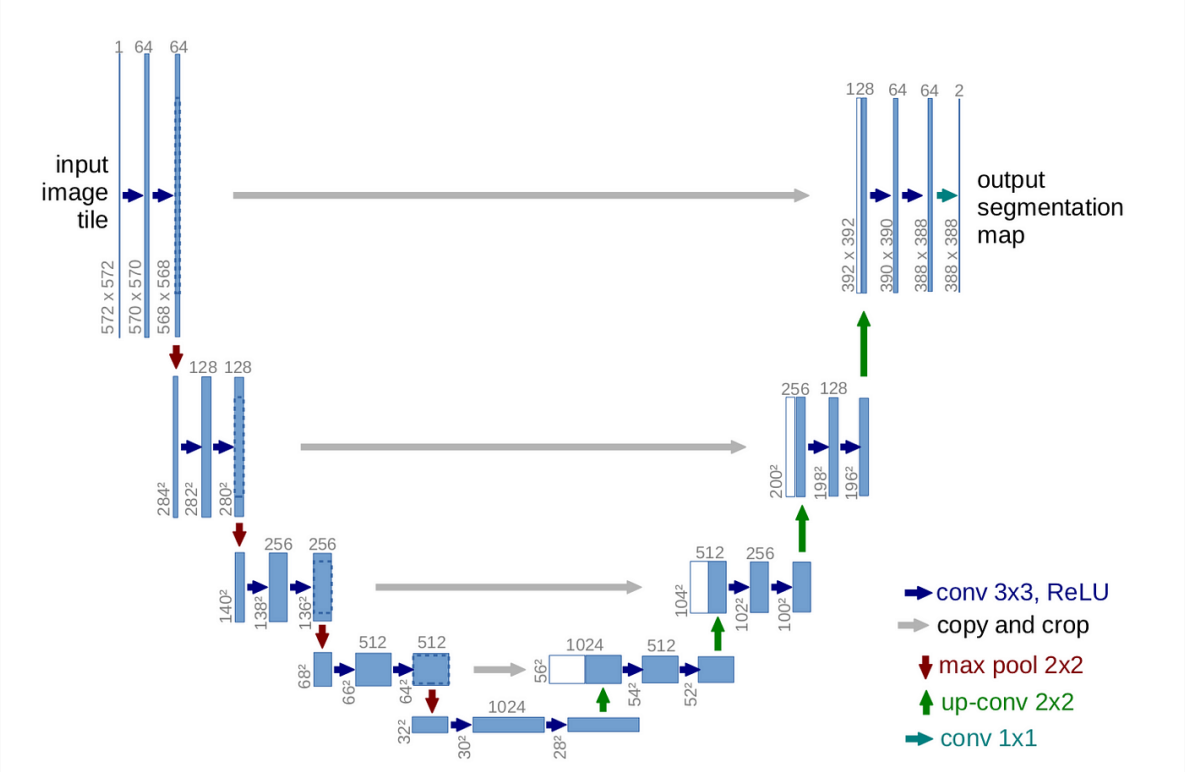


Figure 8: U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations. Image taken from [5].

Additional features

Name	Units	Description
100m wind speed	ms^{-1}	Wind speed of air at a height of 100 meters above the surface of the Earth, given easterly and northerly components u and v the speed is obtained by $\sqrt{(u^2+v^2)}$.
Timestamp	[m,d,h]	Timestamp including month, day, and hour of the start of the given sequence, tile encoded into a 96x96 array.
Land-sea mask	dimensionless	Proportion of land, as opposed to ocean or inland waters in a grid box.
Geopotential	m^2s^{-2}	Gravitational potential energy of a unit mass, at a particular location at the surface of the Earth, relative to mean sea level.

Table 3: Additional features units and details.

Single diffusion with different inputs on EU-50

Inputs	MSE 1h	MSE 2h	MSE 3h
8 rain	2.62e-04	4.60e-04	6.21e-04
8 rain + lsm + geopot	2.60e-04	4.61e-04	6.23e-04
8 rain + lsm + geopot + time	2.60e-04	4.56e-04	6.16e-04
8 rain + lsm + geopot + time + 2 wind speed	2.59e-04	4.51e-04	6.03e-04

Table 4: Results comparison on the **EU-50** dataset using different sets of additional features with the **Single Diffusion** model. All sets include 8 frames representing total precipitation (*rain*). *lsm* and *geopot* stand for land-sea mask and geopotential map, respectively. *time* represents the timestamp embedding.