Geosci. Model Dev., 18, 5351–5371, 2025
https://doi.org/10.5194/gmd-18-5351-2025

Geoscientific Model Development
Open Access
EGU

GPTCast: a weather language model for precipitation nowcasting

Gabriele Franch[1], **Elena Tomasi**[1], **Rishabh Wanjari**[1], **Virginia Poli**[2], **Chiara Cardinali**[2], **Pier Paolo Alberoni**[2], **and Marco Cristoforetti**[1]

[1]Fondazione Bruno Kessler, Trento, Italy
[2]Arpae Emilia-Romagna, Bologna, Italy

Correspondence: Gabriele Franch (franch@fbk.eu)

Received: 25 September 2024 – Discussion started: 10 October 2024
Revised: 18 April 2025 – Accepted: 9 June 2025 – Published: 27 August 2025

Abstract. This work introduces GPTCast, a generative deep learning method for ensemble nowcasting of radar-based precipitation, inspired by advancements in large language models (LLMs). We employ a generative pre-trained transformer (GPT) model as a forecaster to learn spatiotemporal precipitation dynamics using tokenized radar images. The tokenizer is based on a Variational Quantized Autoencoder (VQ-GAN) featuring a novel reconstruction loss tailored for the skewed distribution of precipitation that promotes faithful reconstruction of high rainfall rates. This approach produces realistic ensemble forecasts and provides probabilistic outputs with accurate uncertainty estimation. The core architecture operates deterministically during the forward pass; ensemble variability arises from sampling the categorical probability distribution predicted by the forecaster during inference, rather than requiring external random inputs such as noise injection common in other generative models. All forecast variability is thus learned solely from the data distribution. We train and test GPTCast using a 6-year radar dataset over the Emilia-Romagna region in northern Italy, showing superior results compared to state-of-the-art ensemble extrapolation methods.

1 Introduction and prior work

Nowcasting – short-term forecasting up to 6 h – of precipitation is a crucial tool for mitigating water-related hazards (Werner and Cranston, 2009). Sudden precipitation can result in landslides and floods, frequently compounded by strong winds, lightning, and hailstorms, which can seriously jeopardize human safety and damage infrastructure. The foundation of very short term (up to 2 h) precipitation nowcasting systems is the application of extrapolation techniques to weather radar reflectivity sequences (Bojinski et al., 2023) that ingest current and n previous observations $T_{-n}, \ldots, T_{-1}, T_0$ with the aim to extrapolate m future time steps $T_1, T_2, \ldots, T_m$. These short-term precipitation forecasts are essential for emergency response when released timely and communicated properly via early warning systems (Göber et al., 2023).

The main contenders to extrapolation techniques are numerical weather prediction (NWP) models, which can be used to forecast the probability and estimate the intensity of precipitation across large regions, but their accuracy is limited at smaller geographical and temporal scales (Surcel et al., 2015). Convective precipitation, which produces high rainfall rates and small cells, is especially difficult to forecast correctly for NWP models (Sun et al., 2014). For these reasons, operational weather agencies recognize the great value offered by short-term extrapolation forecasts and make heavy use of statistical and, more recently, data-driven models that utilize the most recent weather radar observations for nowcasting (Woo and Wong, 2017; Turner et al., 2004).

Lagrangian extrapolation is the most well known method for nowcasting precipitation (Bellon and Austin, 1978). It generates motion vectors to forecast the future direction of precipitation systems by applying optical-flow algorithms to a series of radar-derived rain fields. However, this approach becomes less accurate for increasing lead time, particularly in convective situations where precipitation could increase or decrease quickly. Several alternative techniques have been studied to overcome these constraints, such as the seamless integration between nowcasting and NWP forecasts (Sideris et al., 2020; Bowler et al., 2006) and the integration of orography data (Foresti et al., 2018; Panziera et al., 2011).

Other, more sophisticated nowcasting methods improve the Lagrangian approach by generating ensemble nowcasts and preserving the precipitation field's structural characteristics. These sets of multiple forecasts aid in the assessment of forecast uncertainty by presenting multiple future scenarios. The most widespread example of this approach is the Short-Term Ensemble Prediction System (STEPS) (Bowler et al., 2006; Seed et al., 2013).

The most recent advancements in nowcasting precipitation have seen the application of data-driven methods and, more prominently, of deep neural networks (DNNs) and generative AI techniques to enhance forecast accuracy and realism. Deterministic DNNs have been instrumental in predicting the dynamics of precipitation, including its development and dissipation, overcoming one of the major shortcomings of extrapolation methods (Shi et al., 2015; Agrawal et al., 2019; Wang et al., 2018; Franch et al., 2020; Ayzel et al., 2020). However, deterministic models tend to produce less precise forecasts over time due to increasing uncertainty that manifests itself as a forecast field that smooths progressively with the lead time. Similarly to Lagrangian extrapolation, to overcome this limitation, ensemble deep learning methods have been introduced. Generative methods have significantly improved the generation of realistic precipitation fields beyond deterministic average predictions. The forefront of this technology is embodied in models that employ techniques, such as generative adversarial networks (GANs) (Zhang et al., 2023; Ravuri et al., 2021), which enable more accurate and detailed precipitation forecasts by learning to mimic real weather patterns closely, and more recently by latent diffusion models (Leinonen et al., 2023; Gao et al., 2023), which can not only generate realistic rainfall forecasts but also produce reliable ensembles that can provide accurate uncertainty quantification of future scenarios. Many of these techniques were originally born in the field of computer vision and have subsequently been adapted to the weather forecasting domain with resounding success (Goodfellow et al., 2014; Rombach et al., 2022).

In this study, we take inspiration from the successful trend of applying large language model (LLM) architectures (Vaswani et al., 2017; Wolf et al., 2020) born in the field of natural language processing (NLP) to other disciplines (Dosovitskiy et al., 2020; Liu et al., 2021), including the medium-range weather forecasting domain (Lang et al., 2024; Lessig et al., 2023), intending to transfer this knowledge to the nowcasting domain. To do so, in our work, we follow a strategy that mimics the setup of natural language processing: a tokenization step, where an input tokenizer splits and maps the input to a finite vocabulary, and an autoregressive model trained on the tokens produced by the tokenizer. We show that such an approach produces realistic and reliable ensemble forecasts. Given the different characteristics of our input data compared to LLMs (i.e., spatiotemporal precipitation fields vs. texts or images), our adaptation introduces several novel contributions instrumental to our task.

2 GPTCast model architecture

There are two main components of our approach, which we call GPTCast:

- *Spatial tokenizer (VQGAN).* An image compression and discretization model that learns to map patches of the radar image from/to a finite number of possible representations (tokens). The learned codebook of tokens can be used to express a compact representation of any precipitation field. The tokenizer thus has a dual role: learning how to compress and decompress the information in the input image and how to discretize the compressed information (i.e., learn an optimal codebook).

- *Spatiotemporal forecaster (generative pre-trained transformer, GPT).* A model trained on token sequences to causally learn the evolutionary dynamics of precipitation over space and time. Given a tokenized spatiotemporal context (a compressed precipitation sequence), the model outputs probabilities over the fixed codebook for the next expected token for the context. The output probabilities can be leveraged for ensemble generation.

This dual-stage architecture is an adaptation of the work of Esser et al., which we repurposed from the task of image generation to the task of precipitation nowcasting by introducing two key modifications:

- In the spatial tokenizer (VQGAN) model, we replace the standard reconstruction loss (mean absolute error, MAE) with a specific loss that helps improve the reconstruction of precipitation patterns (magnitude weighted absolute error, MWAE). Moreover, the new loss also shows a promotion of the token utilization rate, where we achieve 100 % codebook utilization.

- The token sequences used to train the GPT model represent a fixed three-dimensional context of time $\times$ height $\times$ width of precipitation patterns. This allows the model to learn spatiotemporal dynamics of the evolution of radar sequences.

The two components of the model are trained independently in cascade, starting with the tokenizer. This deliberate dual-stage architecture is crucial for achieving stable training and unlocking desirable properties for operational nowcasting run by meteorological services. Indeed, training the VQGAN and the GPT simultaneously with an end-to-end approach would introduce significant instability. As a probabilistic sequence model, the GPT relies on a fixed, finite vocabulary for stable operation: attempting to learn the token representation (vocabulary) concurrently with the complex spatiotemporal dynamics would force the GPT to learn dependencies over a constantly evolving vocabulary, likely hindering convergence. Furthermore, the fundamentally different architectures (CNN-based VQGAN with its specific loss

functions versus the autoregressive transformer GPT) and the challenges of backpropagation through the VQGAN's discrete quantization step would exacerbate training instability. By firstly establishing a robust and fixed vocabulary through the VQGAN, we create a stable foundation for the GPT to learn the spatiotemporal dynamics of precipitation. This separation allows specialized and stable optimization of each component, ultimately enabling both realistic ensemble generation and accurate uncertainty estimation at the spatiotemporal (token) level, which are instrumental in meeting the requirements of operational nowcasting systems run by meteorological services.

Another notable feature of GPTCast is that its core architecture operates deterministically, meaning it does not require stochastic elements such as injected noise during the forward pass for either training or inference. This contrasts with models such as GANs or diffusion models (Ravuri et al., 2021; Leinonen et al., 2023; Zhang et al., 2023), which often rely on random inputs to generate variability. In GPTCast, variability for ensemble generation stems from the learned data patterns: the tokenizer learns a discrete representation, allowing the forecaster to output a categorical probability distribution over the token vocabulary for each prediction step. Sampling from this distribution during autoregressive inference generates diverse ensemble members, ensuring all variability originates from the learned conditional probability of future states given the past, rather than external randomness (note: standard stochasticity in parameter initialization and optimization, e.g., stochastic gradient descent, is still employed during training).

We describe the details of the model setup and novel contributions in the following subsections.

2.1 Spatial tokenizer: VQGAN

The spatial tokenizer is a Variational Quantized Autoencoder (VQGAN) featuring an adversarial loss (Esser et al., 2021) and a novel reconstruction loss specifically tailored to improve the reconstruction of precipitation. We carefully tune the architecture of the VQGAN to obtain a model that provides the highest possible compression while maintaining a good reconstruction performance and computational complexity. The architecture of the tokenizer is visually summarized in Fig. 1.

The encoder (E) and decoder (G) of the autoencoder are symmetric in design and formed mainly by convolutional blocks, with $\alpha = 4$ steps of downsampling and upsampling, respectively. With this setup, each latent vector at the bottleneck summarizes a patch of $2^\alpha = 2^4 = 16 \times 16$ pixels of the input image. Following recent studies (Yu et al., 2022), we find it useful to set a number of channels at the bottleneck (i.e., the length of the latent vector) of 8 to obtain efficient utilization of the codebook, good training stability, and the effective capture of essential features in a space of reduced dimension. This choice was informed by the cited literature and our preliminary experiments, indicating a good balance between codebook utilization, training stability, and feature capture. The latent vectors at the bottleneck are discretized using a quantization layer that maps them to a finite codebook (Z) by finding the closest vector in the codebook. We define a codebook size of 1024 tokens in the quantization layer. The codebook vectors are initialized randomly and then learned during training.

As an example, with an input precipitation map of 192×192 pixels with a dynamic range of 601 possible values for each pixel (from 0 to 60 dBZ with a 0.1 dBZ step, as described later in Table 2), the resulting feature vector at the bottleneck will have a dimensionality of $12\,H \times 12\,W \times 8$ channels. Each 8-channel vector is then mapped to one of the possible 1024 vectors in the codebook, resulting in a compressed and discretized representation of $12\,H \times 12\,W$ with a dynamic range of 1024 values. The resulting total compression ratio of the spatial tokenizer is $\frac{192 \cdot 192 \cdot 601}{12 \cdot 12 \cdot 1024} \approx 150$ times.

To support such a high compression ratio while maintaining good reconstruction ability, especially for the extreme values, we developed a novel reconstruction loss that we use in place of commonly used reconstruction losses (l_1 or l_2, a.k.a. mean absolute error or mean squared error), defined as follows:

$$\mathrm{MWAE}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{n} |\sigma(x_i) - \sigma(y_i)| \cdot \sigma(x_i), \tag{1}$$

where σ is the sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$ and $\boldsymbol{x}$ and $\boldsymbol{y}$ are the input and output vectors of the autoencoder, respectively. We call this loss the magnitude weighted absolute error (MWAE). By giving more weight to pixels with higher rain rates (magnitude), this loss simultaneously serves two purposes: the first is to nudge the tokenizer towards reserving more learning capacity for the reconstruction of extremes, and the other is to help to rebalance the notoriously skewed distribution of precipitation data, which by nature leans towards low rain rates. While the sigmoid function can saturate for very large input values, potentially diminishing the sensitivity to differences in extreme rain rates, this effect is mitigated by our data preprocessing. The input radar reflectivity values (0–60 dBZ) are linearly rescaled to the range $[-1, 1]$ before being fed into the VQGAN. Within this range, the sigmoid function operates in a quasi-linear manner, ensuring that the absolute difference term $|\sigma(x_i) - \sigma(y_i)|$ appropriately reflects differences between the scaled true and reconstructed values, even for high rain rates within the considered 0–60 dBZ range. The primary reason for using the sigmoid, rather than a purely linear weighting, is to provide robustness against potential out-of-range predictions from the decoder during training, which can occur due to the perturbations introduced by adversarial training. The sigmoid gracefully handles such out-of-range values without assigning excessively large loss values, thereby improving training stability.

Alongside MWAE and the adversarial loss, the model incorporates the Learned Perceptual Image Patch Similarity (LPIPS) loss (Zhang et al., 2018), as shown in Fig. 1, which further encourages perceptually realistic reconstructions by comparing feature activations in a pre-trained network. In our preliminary experiments, while not affecting the final reconstruction performance, this loss term enabled a faster model convergence.

The interactions between loss terms during training follow the original VQGAN implementation (Esser et al., 2021). The total size of the VQGAN model is 90 million trainable parameters.

2.2 Spatiotemporal forecaster: GPT

Similarly to Esser et al., the core predictive component of GPTCast is an autoregressive transformer model based on the GPT-2 architecture (Radford et al., 2019). We chose this specific architecture, as it represents a well-established, robust, and widely understood foundation, allowing us to focus on the novel application of the tokenization and autoregressive generation paradigm to radar nowcasting, rather than optimizing for the latest transformer variants. GPT-2 provides a strong baseline whose components are readily adaptable for spatiotemporal forecasting tasks.

The GPTCast transformer utilizes 24 layers and 16 attention heads, resulting in a total of 304 million trainable parameters for this forecasting component. When combined with the VQGAN tokenizer (approximately 90 million parameters; see Sect. 2.1), the entire GPTCast system comprises roughly 394 million parameters. While potentially smaller than the largest models currently used in natural language processing, this scale is substantial within the atmospheric sciences. For context, it exceeds the size of ECMWF's operational AI Forecasting System (AIFS; approx. 253 million parameters according to its public checkpoint (Lang et al., 2024)), is comparable to recent diffusion models for dynamical downscaling (e.g., approx. 300 million parameters in Tomasi et al., 2025), and is significantly larger than prominent graph-based models such as GraphCast (36.7 million parameters; Lam et al., 2023). This highlights that GPTCast, despite using an established architecture, represents a large-scale deep learning approach for precipitation nowcasting. While GPT-2 serves as an effective proof of concept, future work could certainly explore the potential benefits of more recent or specialized transformer architectures (e.g., those optimized for efficiency or long-context modeling) for this task.

We train two configurations, one with a spatiotemporal context size of 8 time steps (40 min) $\times 256 \times 256$ pixels and one with 8 time steps $\times 128 \times 128$ pixels. At the token level, the two configurations amount to a context length of 2048 ($8 \times 16 \times 16$ tokens) and 512 ($8 \times 8 \times 8$ tokens), respectively. We refer to the two models as GPTCast-16x16 and GPTCast-8x8, respectively. In a GPT-like transformer model, the context size (or sequence length) does not affect the number of parameters; instead, it influences the computational complexity and memory requirements of the model during training and (more crucially) inference. For these reasons, careful consideration in balancing computational complexity and model performance should be made, since timely forecasts are crucial for nowcasting. A summary of the two GPT models' settings is reported in Table 1.

The training process of the forecaster is schematized in Fig. 2: contiguous spatiotemporal sequences of radar data are retrieved from the training dataset and encoded into codebook indices through the frozen VQGAN encoder and passed to the GPT model as training samples. The GPT forecaster is trained autoregressively to predict the probability distribution for each token z_t given the sequence of preceding tokens $z_{<t}$. The tokens are ordered starting with the oldest image using a row-first format. The ordering is instrumental to the nowcasting task: in inference, we can provide the model with a context that is pre-filled with the past seven time steps to generate the tokens for the eighth time step.

2.3 Inference

At inference time, the two models are combined in a sandwich-like configuration, with the encoding of the context input images through the VQGAN encoder, the autoregressive generation of the indices of multiple forecast steps via the transformer model, and the final decoding of the tokens back to pixel space using the VQGAN decoder (see Fig. 3). To obtain multiple ensemble members, the autoregressive generation of the indices can be repeated multiple times while applying a multinomial draw over the output probabilities to pick different tokens.

To generate forecasts for spatial domains larger than the specific training context size, we employ a sliding window inference strategy, illustrated in Fig. 4 and detailed in Algorithm 1. We process the target forecast frame sequentially, following the row-first raster scan order. To predict the token index $z_{i,j}$ for a specific spatial location (i, j) in the forecast frame, we construct an input context sequence for the transformer. This sequence comprises relevant tokens from previous time steps within a defined spatiotemporal window around (i, j), along with any tokens already predicted in the current forecast frame that precede (i, j) in the row-first sequential order. The transformer then predicts the probability distribution for the next token based on this context. Sampling from this distribution yields the predicted token $z_{i,j}$. This sequential, conditioned generation ensures that spatial and temporal consistency is learned and maintained across the domain via the transformer's attention mechanism, as each token prediction depends on its previously generated neighbors in space and time. The handling of domain edges occurs naturally as the available context within the sliding window adapts based on the target token's position.

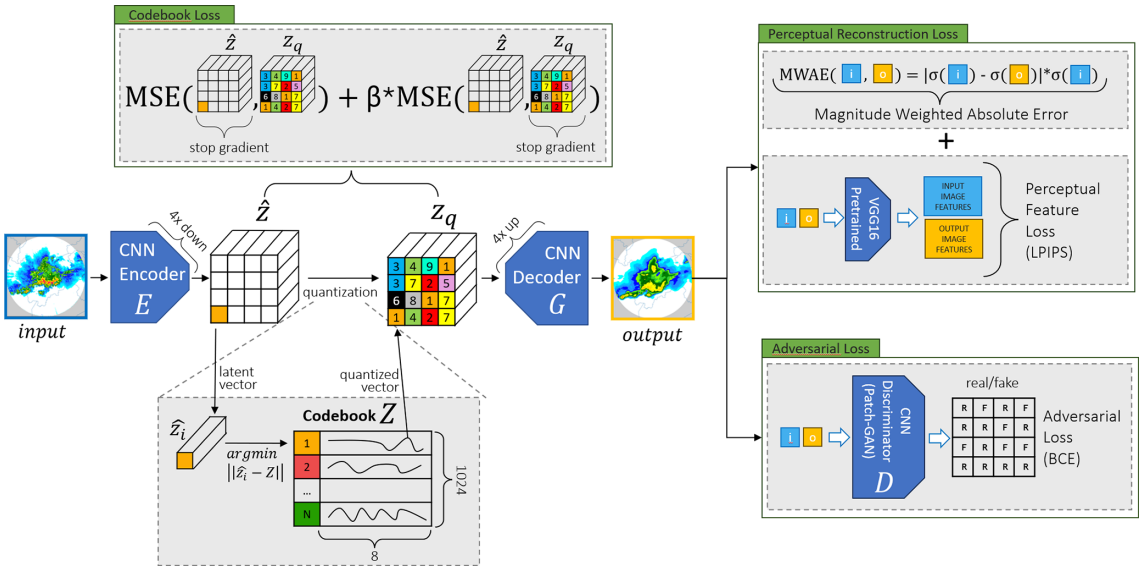


Figure 1. The spatial tokenizer architecture. The four loss terms (MWAE reconstruction loss, adversarial loss, LPIPS perceptual loss, and codebook loss) are enclosed in boxes with green borders. The blue square (***i***) is the input image, and the yellow square (***o***) is the reconstructed autoencoder output. The codebook loss is formed by two complementary parts: the gradient from the detached z_q primarily updates the codebook weights, while the gradient involving the detached z primarily updates the encoder layers preceding the quantization, encouraging them to produce easily quantizable representations.

Table 1. GPTCast model configurations with large and small spatial domain.

Configuration/model name	GPTCast-16x16	GPTCast-8x8
Vocabulary size	1024	1024
Context length	2048 ($8\,T \times 16\,H \times 16\,W$ tokens)	512 ($8\,T \times 8\,H \times 8\,W$ tokens)
Number of layers	24	24
Number of heads	16	16
Embedding dimension	1024	1024

3 Dataset

The dataset we propose for the study is the radar reflectivity composite produced by the Hydrometeorological Service of the Regional Agency for the Environment and Energy of Emilia-Romagna Region in northern Italy (Arpae Emilia-Romagna). The agency operates two dual-polarization C-band radars in the area of the Po Valley, located in Gattatico (44°47′27″ N, 10°29′54″ E) and San Pietro Capofiume (44°39′19″ N, 11°37′23″ E), respectively. The scanning strategy allows coverage of the entire region every 5 min. The area is characterized by a complex morphology, and it spans from the flat basin of the Po Valley in the north to the upper Apennines in the south and from the Ligurian coast in the west to the Adriatic Sea in the east. For the purpose of this work, scans with a radius of 125 km were chosen with a total coverage of 71 172 km^2, summarized in Fig. 5.

Arpae fully manages both the radar acquisition strategy and the data processing pipeline, including several stages of data quality control and error correction developed to reduce the effect of topographical beam blockage, ground clutter, and anomalous propagation (Fornasiero et al., 2006). Specific corrections are applied over the vertical reflectivity profile to improve precipitation estimates at the ground level (Fornasiero et al., 2008). While these quality controls mitigate major issues, residual errors inherent to radar measurements are still present, also affecting the corresponding quantitative precipitation estimation (QPE). No rain gauge correction is applied given the challenges of reconciling the two sources at the short integration time of 5 min.

The resulting product is a 2D reflectivity composite map on a 290 × 373 km grid at a resolution of 1 km^2 per pixel, with a time step of 5 min. The data are provided in units of dBZ (reflectivity factor), with original values ranging from −20 to 60 dBZ. To further minimize the presence of spurious echoes and drizzle, the reflectivity values are clipped between the range of 0 and 60 dBZ, where 0 dBZ represents no precipitation and 60 dBZ represents a rain rate of 205 mm h^{-1} (the radar saturation point). The conversion from dBZ to rain rate is done by applying the standard Marshall–Palmer Z-R

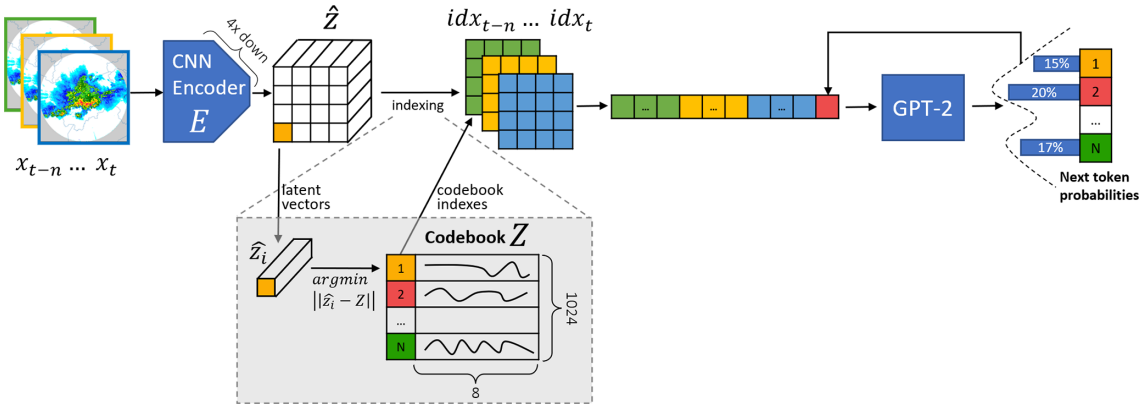


Figure 2. The spatiotemporal forecaster architecture. During the training of the forecaster, the tokenizer encoder (E) weights are frozen.

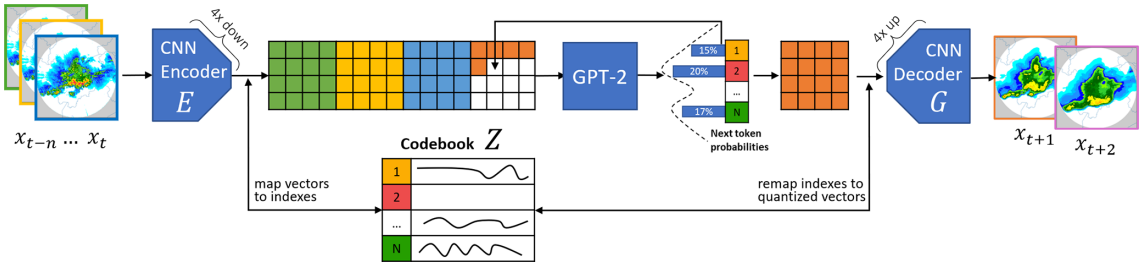


Figure 3. The GPTCast architecture during inference. The trained tokenizer and forecaster are combined (tokenizer encoder (E) → forecaster → tokenizer decoder (G)) to generate forecasts. In the standard unconditional setting, the next token is chosen by applying a multinomial draw over the codebook probabilities to generate different ensemble members.

(Marshall and Palmer, 1948) transformation with parameters $a = 200$ and $b = 1.6$.

3.1 Data selection, preprocessing, and augmentation

For the purposes of our study, we extract all contiguous precipitating sequences in the 6 years between 2015 and 2020. Non-precipitating sequences are discarded, resulting in the selection of 179 264 time steps out of 630 720 (71.5 % of the data is discarded). Specifically, we remove all time steps where the average precipitation over all pixels in the entire domain is less than 0.01 mm h^{-1} for at least 1 h. The remaining sequences are retained only if they form a contiguous sequence of at least 3 h. This focus on precipitating events aims to concentrate the model's learning on the complex dynamics of precipitation itself. The handling of non-precipitating inputs, which are common in operational scenarios, is discussed further in Sect. 5 and addressed empirically in Sect. 4.2.5, where we test the model's behavior with entirely non-precipitating synthetic inputs.

The precipitating sequences are divided between training, validation, and test sets, and the data values are preprocessed by rounding the values to the first decimal digit, resulting in an effective dynamic range of 601 values (from 0 to 60 with a 0.1 step) per pixel.

We prepare two test sets, one for the testing of the spatial tokenizer and one for the testing of the forecaster. To test the spatial tokenizer, we isolate all time steps belonging to the days in the years 2019 and 2020 where extreme events happened by analyzing historical weather reports, resulting in a total of 21 871 radar images (time steps). We call this the *Tokenizer Test Set* (TTS). To test the forecaster, we follow the same validation approach of Pulkkinen et al. (2019), and we extract out of the TTS 10 sequences of 12 h each representative of the most relevant events. This 120 h subset, namely the *Forecaster Test Set* (FTS), is used for the testing of the forecaster.

The remaining sequences are randomly divided between training and validation, with the following final result: 149 524 steps for training, 7869 steps for validation, and 21 871 steps for the TTS including 1450 steps (12 h × 10 events) of the FTS. To further increase the training dataset size and promote generalization, we apply random cropping, random 90° rotation, and flipping to the training dataset during the training phase. The primary motivation for this augmentation strategy is pragmatic: to increase the effective size and variability of the training dataset and, crucially, to mitigate overfitting. We observed, particularly for the larger GPTCast-16x16 model, that training without augmentation led to overfitting on the validation set relatively early. In-

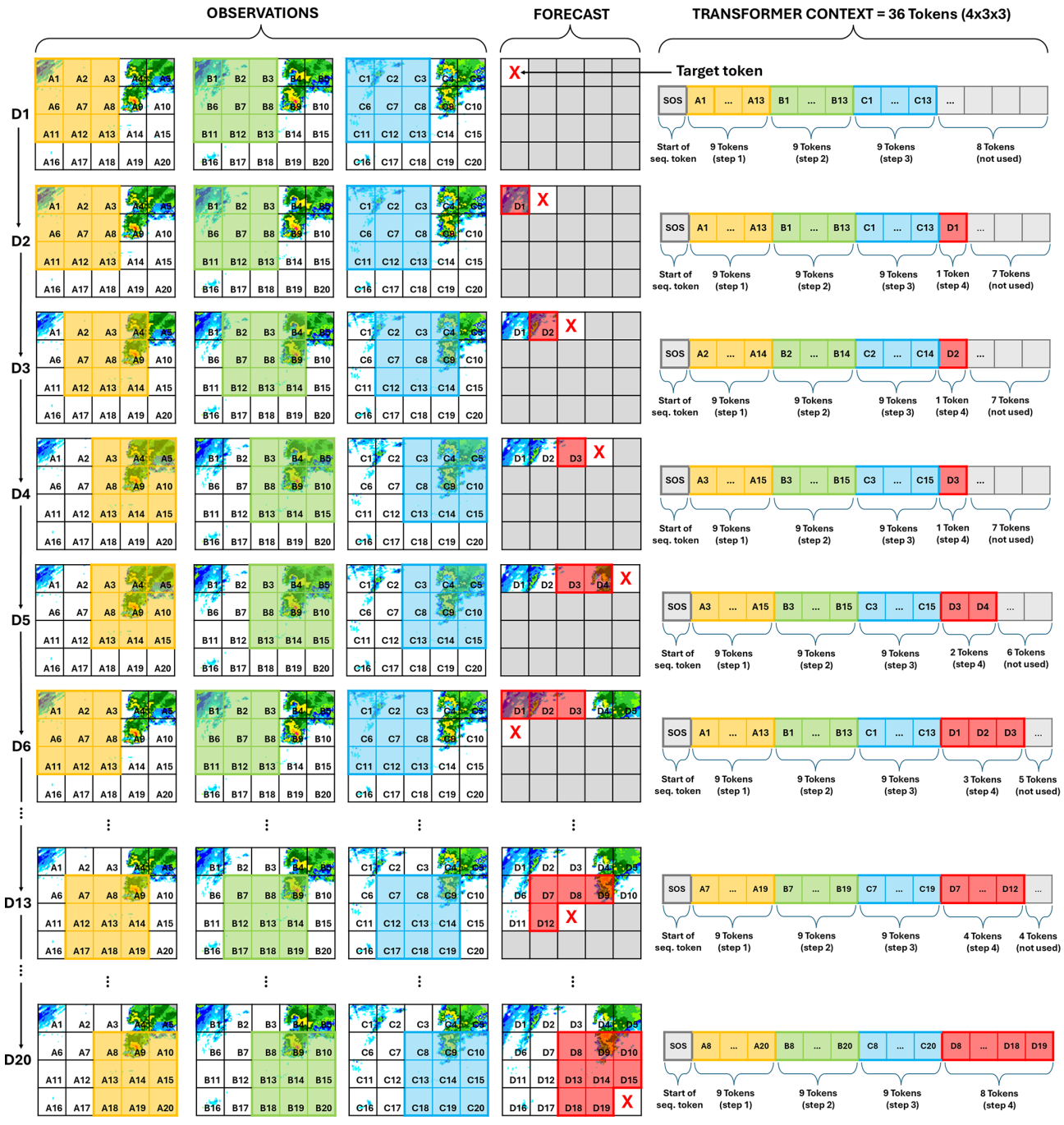


Figure 4. An illustration of the sliding window approach for a forecaster trained with a context length of 4 steps × 3 height × 3 width (36 tokens). Forecasts for domains of arbitrary sizes can be generated by moving the context window across the forecasting domain to predict a target token in the larger domain (starting with the token at the top-left position). A fixed start-of-sequence token (index 0) is prepended to the context to provide an initial conditioning for the first token.

troducing these random transformations allows significantly longer training periods, improving the model's generalization by encouraging invariance to the orientation of precipitation features.

We acknowledge that this approach has trade-offs. By making the dataset invariant to orientation, we prevent the model from explicitly learning geographically fixed patterns, such as precipitation enhancement due to specific orography or effects related to dominant wind directions within the fixed geographical domain. We do not provide additional contextual information (e.g., topography, large-scale wind fields) to the model, partly to maintain a fair comparison with the

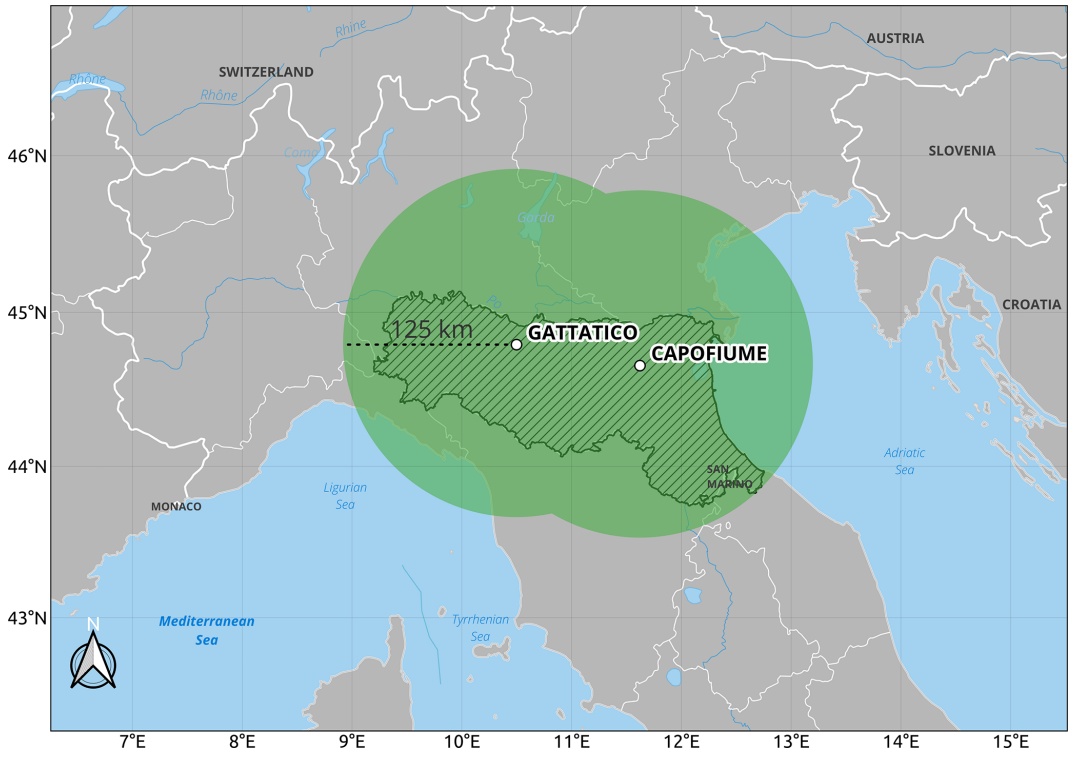


Figure 5. Extent of the dataset. Effective coverage is the composite of the 125 km range of the Gattatico and San Pietro Capofiume radars (green area). The hatched area is the Emilia-Romagna region.

baseline extrapolation methods (introduced in Sect. 4.2.1), which also operate solely on the precipitation fields. The chosen augmentation strategy therefore prioritizes learning the inherent dynamics, structure, and evolution of precipitation patterns themselves, aiming for a model that generalizes well to these dynamics regardless of their orientation within the frame, at the expense of capturing location-specific effects.

Table 2 summarizes the resulting dataset characteristics.

4 Results

Before presenting the quantitative and qualitative results, we clarify the roles of the different data subsets used throughout model development and evaluation.

All model development, hyperparameter tuning, and selection processes were performed using only the training and validation sets. This includes the selection of the final VQGAN tokenizer architecture (based on reconstruction fidelity and downstream performance on the validation set, comparing MAE and MWAE variants) and the selection of the best-performing GPTCast forecaster checkpoint (based on metrics evaluated exclusively on the validation set).

The two test sets (FTS and TTS) were used for the final evaluation presented in the following sections, after all model architectures and checkpoints were finalized based on validation performance. To further assess generalization to truly independent data beyond the scope of the original dataset, we also present an evaluation on a separate, out-of-distribution dataset over Germany in Sect. 4.2.4.

We analyze the performance of our model at two stages: firstly, we analyze the amount of information loss introduced by the data compression in the tokenizer, and then we analyze the performance of GPTCast as a whole for the nowcasting of precipitation up to 2 h in the future. All scores and measures in the Results section are computed on rain rate values (after applying Z-R conversion).

4.1 Spatial tokenizer reconstruction performance

Given the high compression ratio that we introduce in the VQGAN, it is crucial to understand how much and what type of information is lost during the compression and discretization step operated by the tokenizer. Depending on the nature of the information loss, certain phenomena may be completely lost, and this can compromise the ability of the transformer to learn and forecast some precipitation dynamics (e.g., extreme events). The new MWAE loss introduced in Sect. 2.1 is specifically built to improve the reconstruction performance of the tokenizer and reach a good level of data reconstruction while maintaining a high compression factor.

Table 3 shows the performance in reconstruction ability on the TTS between a VQGAN trained using as reconstruction loss a standard mean absolute error (MAE) and using

Table 2. Summary of dataset characteristics.

Attribute	Details
Product description	Arpae radar reflectivity composite (northern Italy)
Map size	290×373 pixels
Pixel size	1 km resolution
Time step	5 min
Reflectivity range	-20 to 60 dBZ (clipped to 0–60 dBZ, 0.1 step = 601 values of dynamic range)
Date range	Precipitation sequences in the years 2015–2020
Dataset size	630 720 total time steps (179 264 precipitating time steps selected)
Training and validation	149 524 time steps for training, 7869 for validation
Test datasets	TTS: 21 871 time steps; FTS: 1450 time steps (10 events of 12 h)

Algorithm 1 Pseudocode for sliding window prediction algorithm.

Require: input_indices {Tensor of shape $[B, S, H, W]$}
Require: c_indices {Conditioning tokens (Start of Sequence)}
Require: window_size {Size of sliding context window}
Ensure: predicted_indices {Next frame token indices}
  $B, \_, H, W \leftarrow$ shape(input_indices)
  half_window $\leftarrow \lfloor$window_size$/2 \rfloor$
  predicted_indices $\leftarrow$ Tensor(B, H, W) filled with -1
  conditioning $\leftarrow$ reshape(c_indices) {Flatten conditioning}
  for $i = 0$ **to** $H - 1$ **do**
    for $j = 0$ **to** $W - 1$ **do**
      /* Calculate window boundaries with edge handling */
      $i_{\text{start}} \leftarrow \max(0, i - \text{half\_window})$
      $i_{\text{end}} \leftarrow \min(H, i_{\text{start}} + \text{window\_size})$
      $i_{\text{start}} \leftarrow \max(0, i_{\text{end}} - \text{window\_size})$ {Adjust if at bottom edge}
      $j_{\text{start}} \leftarrow \max(0, j - \text{half\_window})$
      $j_{\text{end}} \leftarrow \min(W, j_{\text{start}} + \text{window\_size})$
      $j_{\text{start}} \leftarrow \max(0, j_{\text{end}} - \text{window\_size})$ {Adjust if at right edge}
      /* Extract past context and already predicted tokens */
      past_tokens $\leftarrow$ flatten(input_indices$[:, :, i_{\text{start}} : i_{\text{end}}, j_{\text{start}} : j_{\text{end}}]$)
      pred_patch $\leftarrow$ predicted_indices$[:, i_{\text{start}} : i_{\text{end}}, j_{\text{start}} : j_{\text{end}}]$
      $\text{window\_pos}_i \leftarrow i - i_{\text{start}}$
      $\text{window\_pos}_j \leftarrow j - j_{\text{start}}$
      tokens_count $\leftarrow \text{window\_pos}_i \times (j_{\text{end}} - j_{\text{start}}) + \text{window\_pos}_j$
      pred_tokens $\leftarrow$ first tokens_count elements from flattened pred_patch
      /* Build context and predict next token */
      context $\leftarrow$ concatenate(conditioning, past_tokens, pred_tokens)
      next_token $\leftarrow$ predict_next_index(context)
      predicted_indices$[:, i, j] \leftarrow$ next_token.squeeze() {Fix shape mismatch}
    end for
  end for
  return predicted_indices

our proposed MWAE loss. We consider both global regression scores, such as the mean absolute error (MAE), the mean squared error (MSE), and the structural similarity index measure (SSIM; Wang et al., 2004), along with categorical scores computed by thresholding the precipitation at multiple rain rates (1, 10 and 50 mm h^{-1}), such as the critical success index (CSI) and the frequency bias (BIAS).

The autoencoder trained with MWAE shows significant improvements over all the considered metrics, but it is crucial to notice that the improvements are more pronounced for higher rain rates, whose frequency is almost precisely reconstructed by the autoencoder. This is clearly visible in the improvements in BIAS at 50 mm h^{-1}, which is defined as the fraction between the number of pixels in the input image over 50 mm h^{-1} and the number of pixels that surpass the same threshold in the reconstruction, where we obtain a jump in performance from 0.22 to 0.92 (where 0 is total underestimation, 1 is the perfect score, and greater than 1 is overestimation).

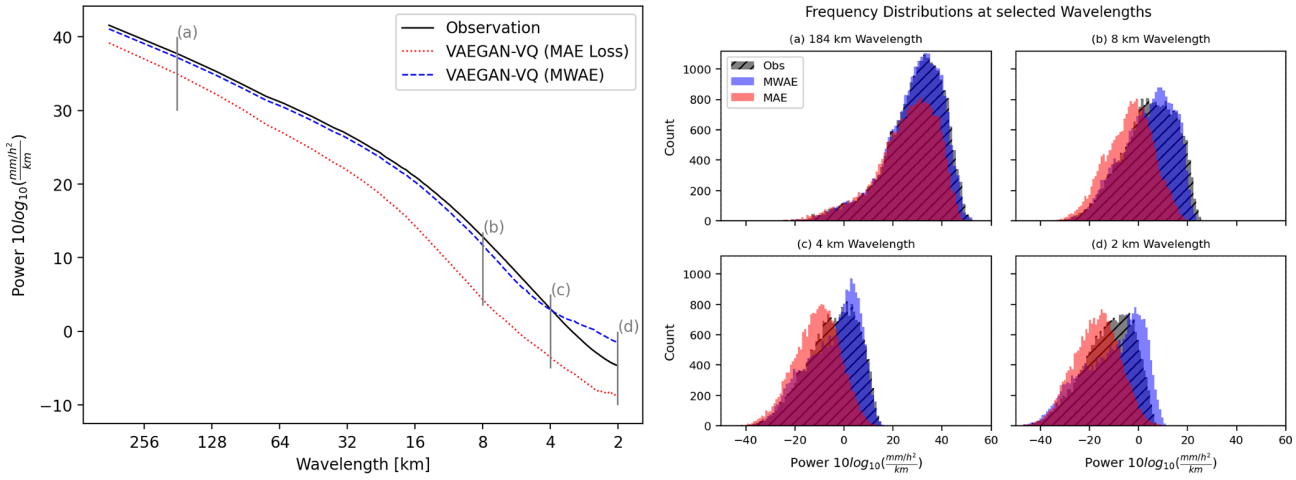
The recovery in frequency is also confirmed by analyzing the radially averaged power spectral density (i.e., the amount of energy) of the input and reconstruction: as shown in Fig. 6, the average power spectra of the MWAE autoencoder closely resemble the input (albeit with an overestimation at the smallest wavelengths), while the standard autoencoder distribution is constantly shifted and underestimated at all wavelengths.

Improvement in CSI score is also significant (at 50 mm h^{-1}, more than 3 times higher), albeit not as thorough as the frequency recovery. This implies that the remaining source of error is that the reconstructed precipitation fields have either a different structure or a different location when compared to the input (i.e., the amounts of the reconstructed precipitation are correct but misplaced at the spatial level).

To better characterize this remaining source of error, we compute the SAL measure (Wernli et al., 2008, 2009), which evaluates three key aspects of the precipitation field within a specified domain: structure (S), amplitude (A), and location (L). The amplitude component (A) measures the relative deviation of the domain-averaged reconstructed precipitation amount from the input. Positive values indicate an

Table 3. Reconstruction performance on the TTS of VQGAN trained with mean absolute error (MAE) loss and with our proposed MWAE loss. (↓) means lower is better, and (↑) means higher is better; for frequency bias (BIAS), closer to 1 is better. The best model is in bold.

Model/performance	MAE (↓)	RMSE (↓)	SSIM (↑)	CSI (↑)/BIAS @ 1 mm^{-h}	CSI/BIAS @ 10 mm^{-h}	CSI/BIAS @ 50 mm^{-h}
VQGAN MWAE	**0.204**	**2.02**	**0.988**	**0.81/1.03**	**0.56/0.94**	**0.44/ 0.92**
VQGAN MAE	0.265	2.66	0.981	0.74/0.93	0.38/0.62	0.13/0.22



(a) Average Energy spectra reconstruction performance

(b) Frequency distribution at selected wavelengths

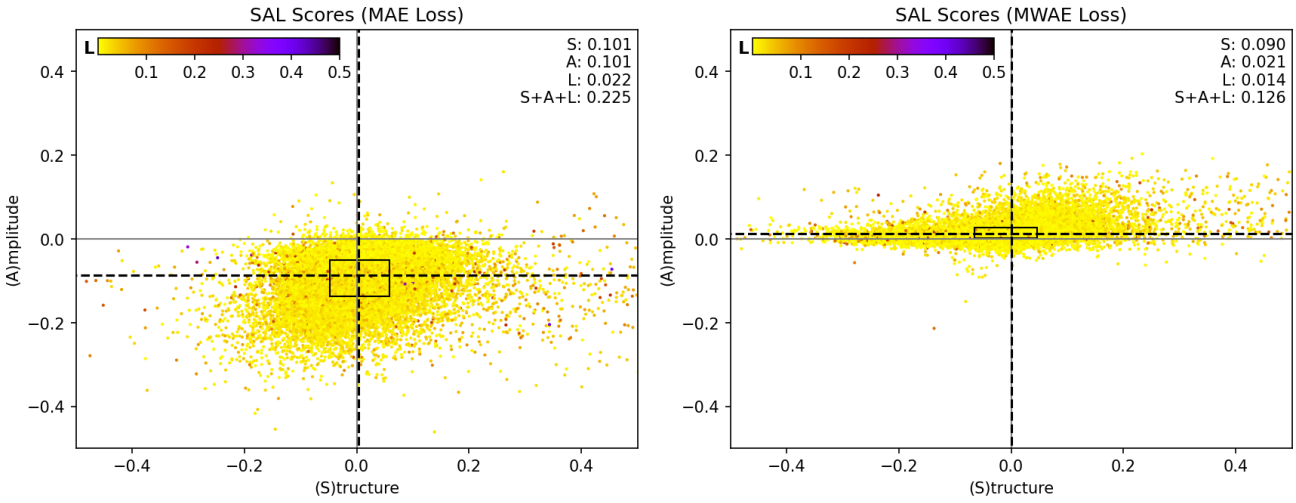
Figure 6. Comparison of radially averaged power spectral density reconstruction performance by adopting the MWAE loss function compared to MAE. The adoption of MWAE improves the ability of the autoencoder to reproduce the energy distribution of precipitation at all wavelengths.

overestimation of total precipitation, while negative values indicate an underestimation. The structure component (S) assesses the shape and size of predicted precipitation areas. Positive values occur when these areas are too large or too flat, while negative values indicate that they are too small or too peaked. The location component (L) evaluates the accuracy of the predicted location of precipitation. It combines information about the displacement of the reconstructed precipitation field's center of mass compared to the input and the error in the weighted average distance of the precipitation objects from the center of the total field. Perfect forecasts result in zero values for all three components, indicating no deviation between input and reconstructed precipitation patterns.

The SAL analysis plot for both autoencoders is shown in Fig. 7. The MWAE autoencoder improves over the baseline autoencoder on all scores, with a median value that is close to zero for all three components. A residual source of absolute error remains in the structure component, while both amplitude and location errors are negligible.

In summary, divergences in the size and shape of the reconstructed precipitation patterns account for the majority of the error for our new autoencoder, while the locations, frequencies, and energy contents of the precipitation patches are mostly accurate. Overall, this is a good compromise for the nowcasting task, since we can tolerate higher compromises for errors in structure, whereas systematic errors in amplitude, frequency, or location can seriously impair the forecaster's ability to accurately predict the evolutionary dynamics of precipitation. Some qualitative examples of the input and reconstruction from both autoencoders are presented in Fig. 8.

The last test involves an assessment of the ability of the autoencoder to reconstruct saturation-level inputs. We create a synthetic image with a saturated 64×64 km patch of 205 mm h^{-1} (60 dBZ) at the center, encode it through the tokenizer, and decode the resulting token map. The reconstruction in Fig. 9 visually confirms that end-of-scale values are much better represented in the learned codebook of the MWAE autoencoder, which is able to express rain rates up to the saturation level, although not for large extents like the one provided in the input. This limitation is expected due to the absence of such extensive saturated areas in the training data. Consequently, this could potentially affect the model's performance when encountering record-breaking extreme events that might exhibit such large areas of maximum intensity.



(a) SAL scores for Auteoencoder with MAE Loss

(b) SAL scores for Auteoencoder with MWAE Loss

Figure 7. Structure, amplitude, and location (SAL) plot that compares the performance of the MAE and MWAE autoencoders. Each dot on the plot represents the scores of one image in the TTS. Structure and amplitude are plotted on the horizontal and vertical axes, respectively, while the location component is represented by the color. The dashed vertical and horizontal lines indicate the median values of the structure (S) and amplitude (A) scores, respectively. The rectangular box represents the area between the 25th and 75th percentiles (i.e., the vertical and horizontal sides of the box contain 50 % of the points). The numbers on the top right show the mean absolute values.

4.2 GPTCast nowcasting performance

4.2.1 Baseline model: LINDA

We examine and compare GPTCast forecasting performance with that of the Lagrangian INtegro-Difference equation model with Autoregression (LINDA) (Pulkkinen et al., 2021), the state-of-the-art ensemble nowcasting model included in the pySTEPS package (Pulkkinen et al., 2019). LINDA is a nowcasting technique intended to provide superior forecast skill in situations with intense localized rainfall compared to other extrapolation methods (S-PROG or STEPS). Extrapolation, S-PROG (Seed, 2003), STEPS (Bowler et al., 2006), ANVIL (Pulkkinen et al., 2020), an integro-difference equation (IDE), and cell tracking techniques (Dixon and Wiener, 1993) are all combined in this model.

4.2.2 Verification scores

For verification assessment, we rely on the continuous ranked probability score (CRPS) and the rank histogram, which are essential tools for verifying ensemble forecasts. By showing the frequency of observed values among the forecast ranks, the rank histogram evaluates the dispersion and reliability of ensemble forecasts and highlights biases such as under- or over-dispersion. By comparing the prediction's cumulative distribution function to the actual value, CRPS calculates a numerical score for forecast skill that indicates how accurate a probabilistic forecast is. The two scores complement each other, with the CRPS providing a measure of forecast accuracy as a whole and the rank histogram emphasizing the ensemble spread and reliability.

4.2.3 Performance on the forecast test set

We use the FTS for our main performance comparison. Out of the 10 events in FTS, 7 are convective events occurring in spring or summer and 3 are winter precipitation events. For each event, we produce a forecast every 30 min, and each forecast is a 20-member ensemble forecast with 5 min time steps and a maximum lead time of 2 h (i.e., 24 forecasting steps) for both LINDA and GPTCast. This results in a total of 200 forecasts (20 forecasts per event) generated per model. For GPTCast, we test both of the two model configurations, GPTCast-16x16 and GPTCast-8x8.

The CRPS score for each of the three models – LINDA, GPTCast-16x16, and GPTCast-8x8 – is displayed in Fig. 10: both variants of GPTCast outperform LINDA across all lead times, with GPTCast-16x16 outperforming all other models. This result clearly shows that the model can learn a more thorough dynamic of the evolution of precipitation patterns when the context size is more spatially extended. It is important to notice that this improvement comes with a non-negligible increase in terms of computational time at inference, which in our experiments was close to 1 order of magnitude (GPTCast-8x8 computes a time step in 2 s compared to 17 s for the larger model on an NVIDIA RTX 4090).

Figure 11 analyzes the rank histogram at different lead times for all three models, including information on the Kullback–Leibler (KL) divergence from the uniform distribution. Both versions of GPTCast provide a better overall

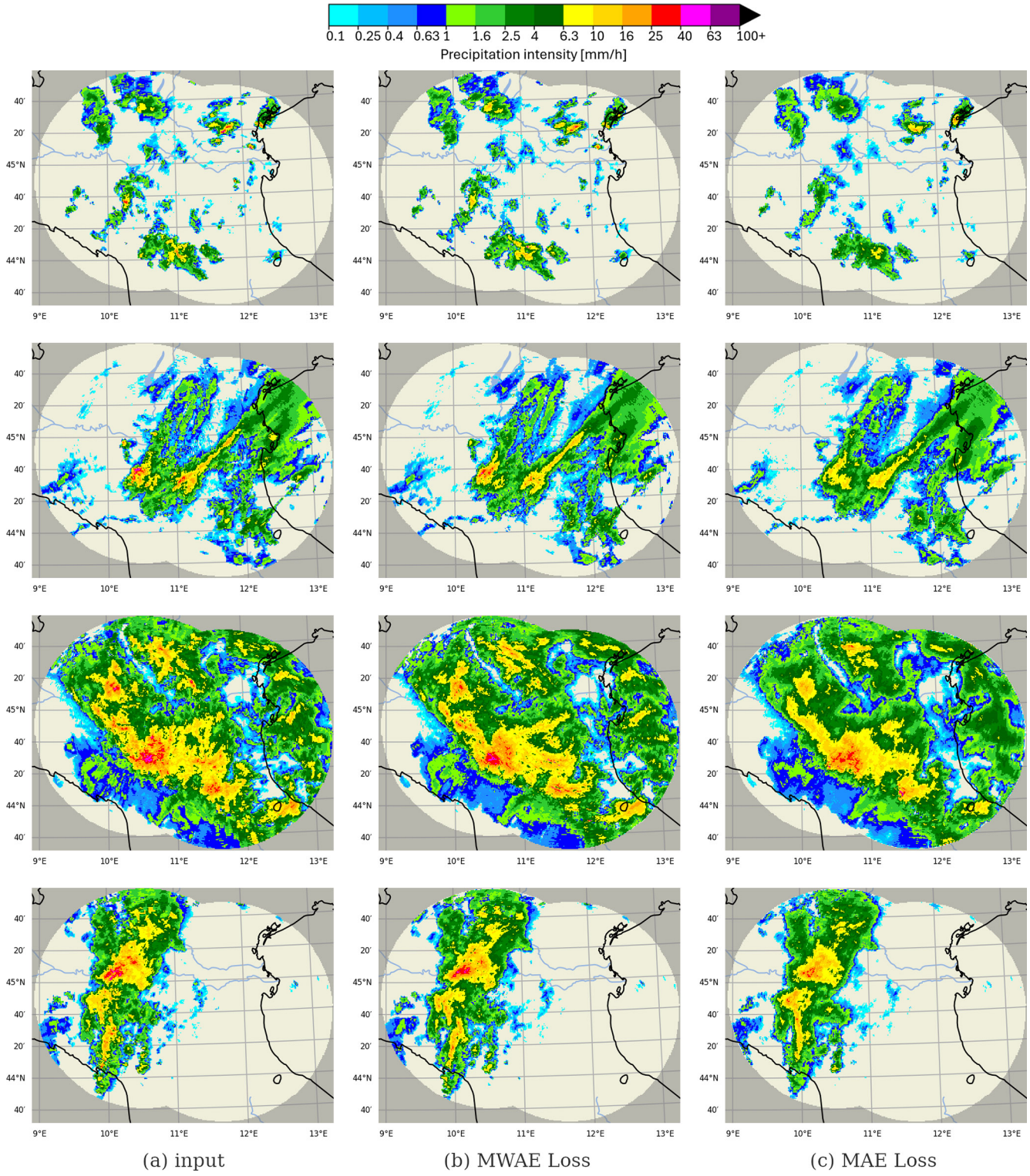


Figure 8. Qualitative comparison between precipitation snapshots reconstructed by the VQGAN autoencoder trained with MWAE loss and MAE loss, taken from the TTS. The autoencoder trained with MWAE loss shows a marked improvement in the reconstruction of precipitation, with crucial improvements in the reconstruction of higher rain rates (thunderstorms).

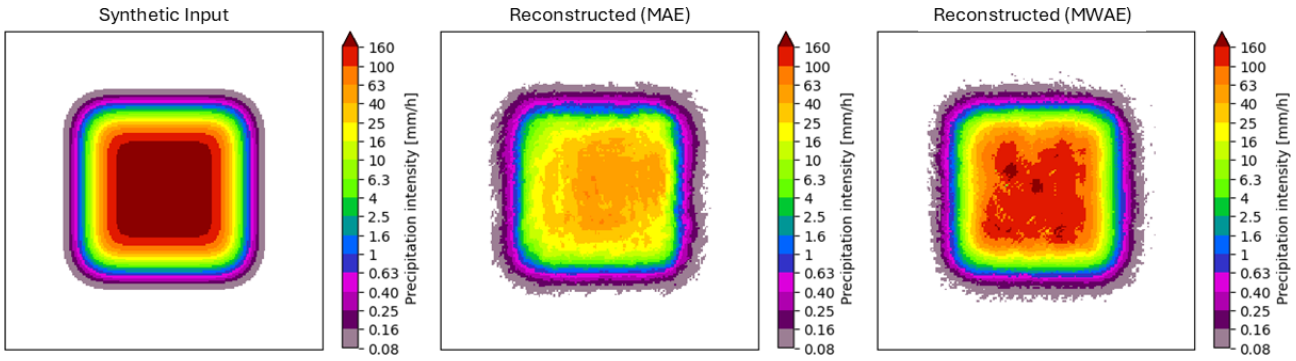


Figure 9. Qualitative comparison between precipitation snapshots reconstructed by the VQGAN autoencoder trained with MWAE loss and MAE loss on a synthetic saturated image. The MWAE-trained model can reach saturation-level intensities, although only over small areas.

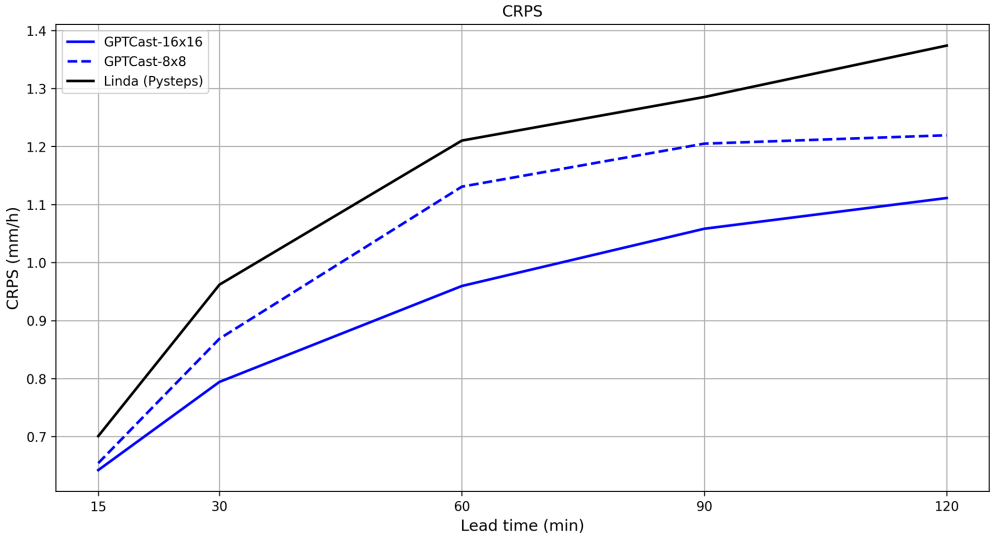


Figure 10. Continuous ranked probability score (CRPS) comparison of GPTCast and LINDA over the FTS (lower is better) at different lead times.

score than LINDA, which tends to be under-dispersed, with GPTCast-8x8 being the best model. Moreover, GPTCast-8x8 shows a rank distribution close to optimal up to the first hour, with a KL divergence from the uniform distribution of 0.006 at 60 min lead time (12 steps). GPTCast-16x16 displays an overall better rank histogram than LINDA up to the first 60 min, with a tendency to underestimate that compounds over time: we attribute this behavior to the increased ability of the GPTCast-16x16 to capture the training distribution, which has a higher ratio of dissipating precipitation events than the FTS (which is filtered to contain only extreme events).

Figure 12 shows an example of nowcast for a convective case in the FTS, with two ensemble members and the ensemble mean for both LINDA and GPTCast. GPTCast generates two realistic and diverse forecasts, with an ensemble mean that features a better location accuracy than LINDA compared to the observations.

4.2.4 Out-of-distribution evaluation on German radar data

To assess the generalization capability of GPTCast beyond the primary dataset used for training and testing, we perform an additional evaluation on an independent dataset from a different geographical region and source. We utilized the radar dataset over Germany presented alongside RainNet (Ayzel et al., 2020). From the first 150 000 time steps available in this dataset, we selected the 10 cases exhibiting the highest domain-average precipitation to focus on challenging forecasting scenarios.

For each selected case, we extracted the central 256×256 pixel domain, matching the spatial dimensions used in our primary experiments. We then generated 60 min precipitation forecasts using a 20-member ensemble for both GPTCast (specifically, GPTCast-16x16) and LINDA.

The results indicate that GPTCast achieves a lower (better) average CRPS compared to LINDA over these 10 selected

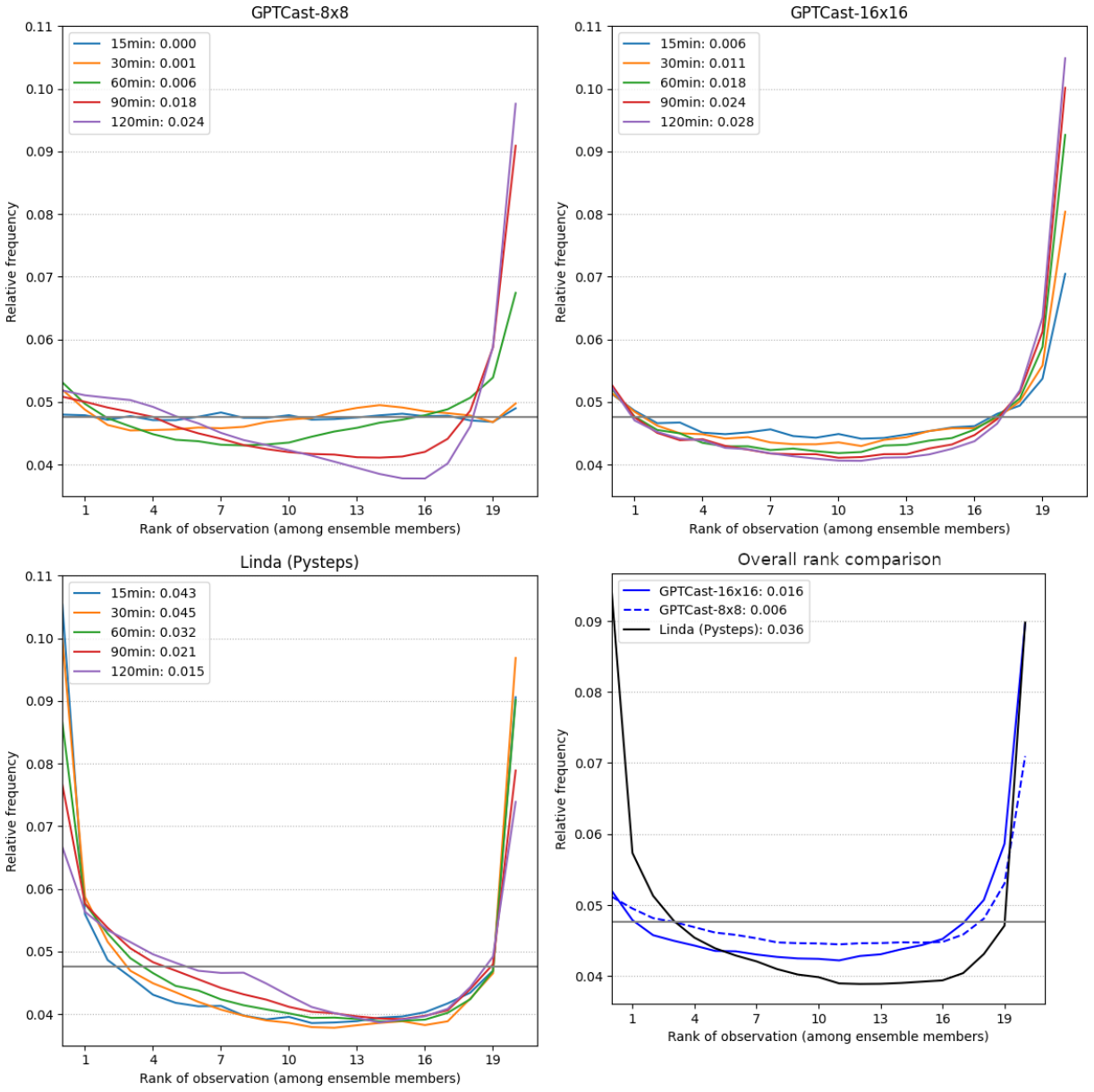


Figure 11. Rank histogram comparison of GPTCast and LINDA on the FTS. The horizontal gray line represents the ideal value (the closer the better). The numbers in the legend indicate the Kullback–Leibler divergence from the uniform distribution (lower is better).

cases, suggesting better overall probabilistic forecast skill in this out-of-distribution setting. However, the rank histogram for GPTCast still exhibited a tendency towards lower ranks, consistent with the underestimation characteristic observed in the primary evaluation (Sect. 4.2.3).

It is important to interpret these results with caution. Firstly, the evaluation comprises only 10 cases, which limits the statistical significance of the findings. Secondly, as noted by Ritvanen et al. (2025), LINDA's performance is often optimized for and excels during high-intensity convective events. The case selection based on domain-average precipitation might not perfectly align with the scenarios where LINDA demonstrates its peak performance relative to other models. Nonetheless, this preliminary out-of-distribution evaluation provides encouraging evidence that the precipitation dynamics learned by GPTCast possess a degree of transferability to different geographical regions and data sources.

4.2.5 Behavior with non-precipitating input

To address the model's behavior when presented with input sequences entirely devoid of precipitation (a scenario excluded during training), we conduct an additional experi-

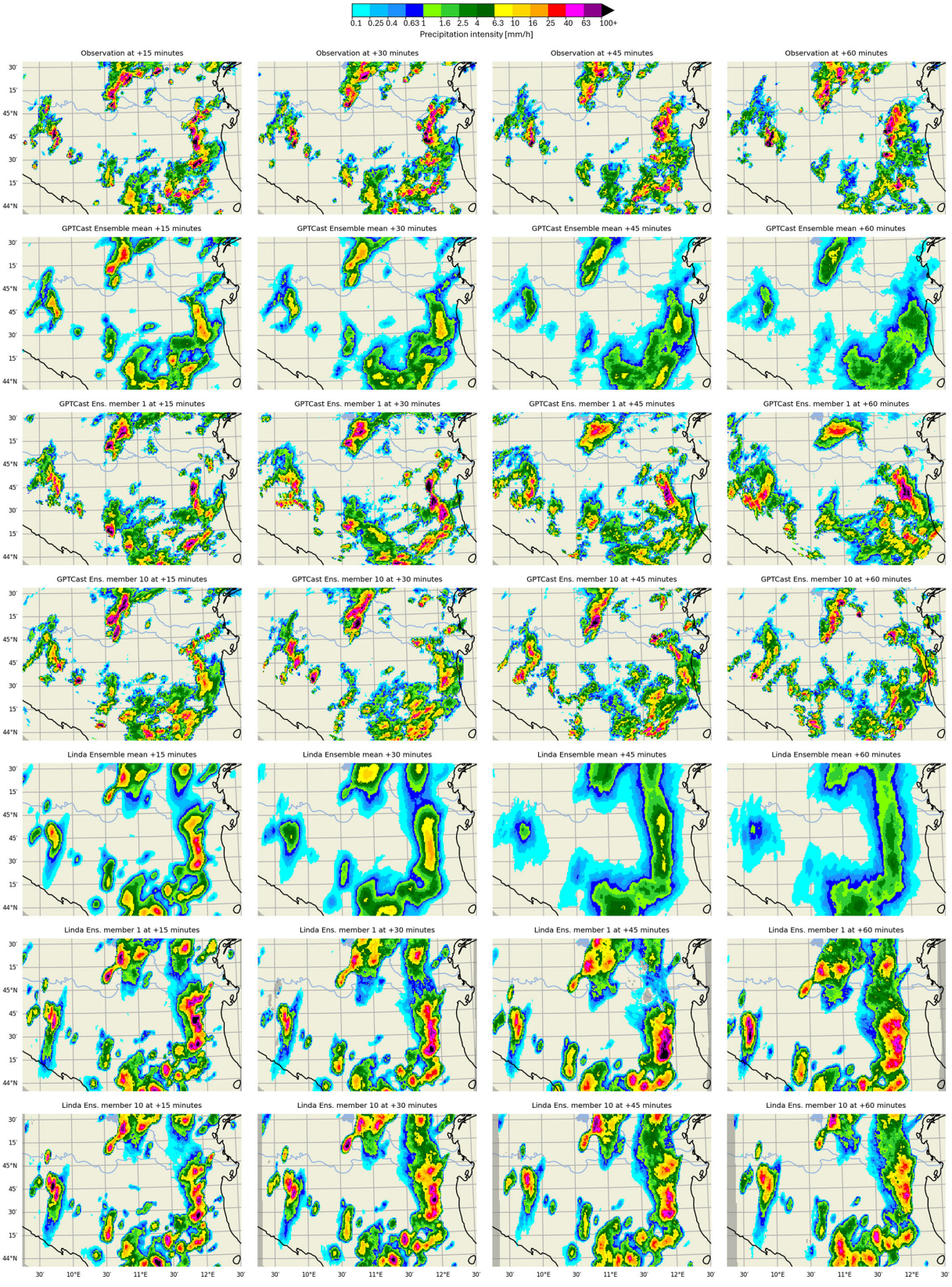


Figure 12. Example comparison of GPTCast-16x16 and LINDA nowcast on a convective case in the Forecaster Test Set (8 June 2020, 11:00 UTC). The domain is cropped on the central area for visualization convenience.

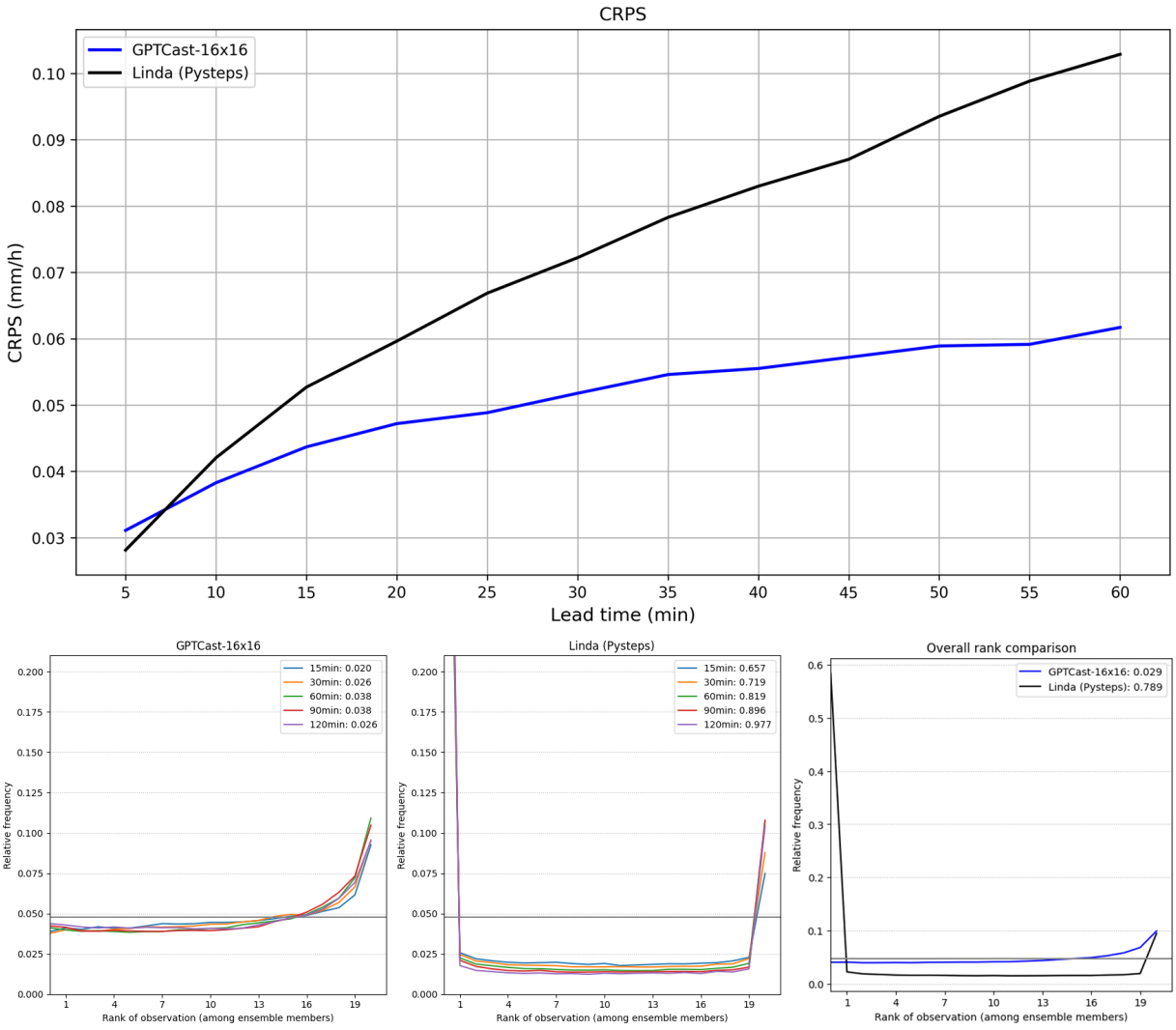


Figure 13. CRPS and rank histogram of GPTCast-16x16 and LINDA on 10 precipitation events over central Germany.

ment using synthetic data. We initialized the GPTCast-16x16 model with an input sequence consisting entirely of zero-value radar reflectivity images (representing "all clear" conditions) across the 256×256 pixel domain for the standard 7-time-step context window. We then generated an ensemble forecast of 20 members for the next time step.

The results show that most ensemble members correctly predicted continued zero (or near-zero) precipitation, consistent with a persistence forecast expected under such conditions. However, in particular, one ensemble member generates a significant spurious, albeit localized and physically plausible-looking, precipitation pattern. This highlights a potential drawback of the generative nature of the model: the possibility of "hallucinating" precipitation features when initialized with data far outside its training distribution (i.e., entirely empty sequences). While infrequent in this test (1 member out of 20), this behavior warrants consideration for operational deployment and is discussed further in Sect. 5. Figure 14 illustrates the behavior of the members and the generated pattern from the deviating ensemble member.

5 Discussion and future work

5.1 Summary and contributions

GPTCast introduces a novel approach to ensemble nowcasting of radar-based precipitation, leveraging a GPT model and a specialized spatial tokenizer to produce realistic and accurate ensemble forecasts. We show that this approach can provide reliable forecasts, outperforming the state-of-the-art extrapolation method in both accuracy and uncertainty estimation.

GPTCast's deterministic architecture enhances interpretability and reliability by generating realistic ensemble forecasts without random noise inputs. The model can be

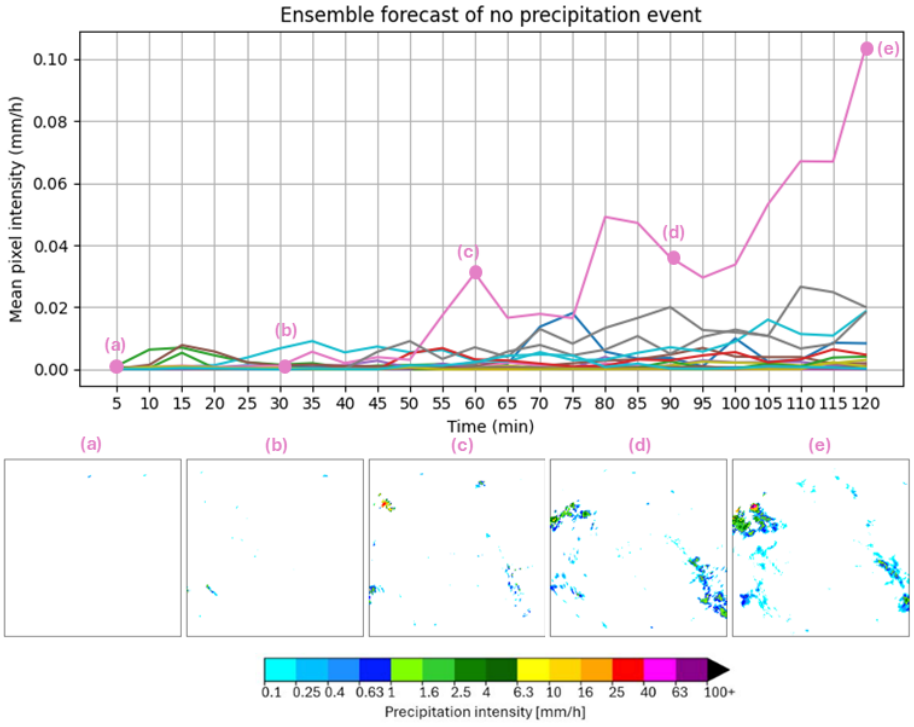


Figure 14. Behavior of the model initialized from a zero-precipitation input sequence of 256×256 pixel domain. Only 1 out of the 20 ensemble members develops a significant precipitating pattern.

scaled to different sizes, both in context length and in terms of parameters (which we postponed to future analyses), allowing a balance in the trade-off between accuracy and computational demands and providing flexibility for different operational settings.

We believe that our method, by adopting an architecture influenced by large language models (LLMs), paves the way for future promising research in precipitation nowcasting that can incorporate all the improvements and developments from the quickly developing field of LLM research. This includes more efficient architectures, improved training techniques, and better interpretability tools. Such integration can potentially enhance GPTCast's performance, scalability, and usability, ensuring that it remains a state-of-the-art nowcasting tool.

5.2 Implementation challenges

Despite its strengths, the approach poses specific challenges that must be considered for the operational usage of the model.

The approach requires the training of two models in cascade, each with its own set of challenges. In our experiments, it was hard to find a stable configuration to train the spatial tokenizer that has to balance multiple competing losses. The MWAE reconstruction loss we introduced helped substantially in terms of both convergence and stability, although at the cost of slower training induced by the smoothing effect of the sigmoid (σ) terms in the loss. On the other hand, we found the forecaster to be very stable in training (as expected by transformers) but computationally intensive in inference, especially for the long context configuration (GPTCast-16x16), making its use in a real-time application such as nowcasting challenging without significant resources.

5.3 Handling non-precipitating conditions and generative artifacts

The ability of the model to effectively capture the training distribution is both its main strength and potential pitfall. A key aspect of our training strategy was the exclusion of entirely non-precipitating sequences, representing a significant portion (71.5 %) of the raw data. This decision aimed to focus the model's learning capacity on the core challenge: capturing the complex dynamics of precipitation initiation, evolution, and decay, rather than diluting the learning signal with vast amounts of "all clear" data. Operationally, if the recent radar sequence shows no precipitation, a simple persistence forecast (predicting continued "no precipitation") is often sufficient and computationally inexpensive for the very short term, making the deployment of a complex model such as GPTCast potentially wasteful in such specific situations. Our training strategy thus aligns with a targeted use case where the model is primarily invoked when precipitation is present or developing.

However, this raises the question of how the model behaves when presented with the non-precipitating inputs it might encounter operationally. While the model learns to handle the cessation of precipitation within partly precipitating sequences present in the training data, its behavior on entirely clear inputs was not explicitly trained. Our analysis in Sect. 4.2.5, using synthetic all-zero inputs, showed that, while the model predominantly predicts continued clear conditions as expected, a small fraction of ensemble members (1 out of 20 in our test) can generate spurious precipitation patterns ("hallucinations"). This generative artifact, occurring when the input is significantly outside the training distribution, represents a potential drawback. While infrequent, this highlights the need for caution and potentially post-processing checks if the model were to be deployed in scenarios where it might frequently receive entirely non-precipitating inputs or, alternatively, highlights the need to implement a simple check to bypass the deep learning model when inputs are non-precipitating. Further investigation could explore fine-tuning strategies or architectural modifications to mitigate such behavior, although the current targeted training approach already aligns well with typical operational workflows where nowcasting models are most crucial during active precipitation events. Moreover, strategies exist to exert more control over the generation process during inference and potentially reduce the occurrence of undesirable outcomes. One common technique, adapted from natural language processing, is top-k sampling (Fan et al., 2018; Holtzman et al., 2020). Instead of sampling from the entire probability distribution over the VQGAN codebook indices predicted by the transformer, top-k sampling restricts the selection pool to only the k tokens (codebook indices) with the highest predicted probabilities at each step. By filtering out low-probability options, this can make the generated sequences more focused and less likely to contain highly improbable or spurious transitions. However, this comes at the cost of potentially reduced forecast diversity and the risk of suppressing genuinely rare but physically valid meteorological events. Choosing an appropriate value for k, or exploring related techniques such as nucleus sampling (top-p) (Holtzman et al., 2020), involves a trade-off between forecast creativity/diversity and robustness against potential hallucinations. Further investigation into optimal decoding strategies for precipitation nowcasting with GPTCast, possibly incorporating physical constraints or adaptive sampling methods, remains an area for future research to enhance reliability for operational use.

5.4 Geographical generalizability

A further consideration regarding the generalizability of GPTCast pertains to the geographical scope of the data used for training and primary evaluation. Our main experiments were conducted using radar data covering the Emilia-Romagna region, which possesses distinct topographical features and precipitation characteristics. Consequently, the model's performance might differ when applied to regions with significantly different environments, such as coastal areas or large flat plains, which exhibit distinct precipitation regimes or atmospheric dynamics.

To provide an initial assessment of the model's robustness beyond its training domain, we performed an additional evaluation on a completely independent dataset comprising recent precipitation events over Germany, a region with different geographical characteristics (as detailed in Sect. 4.2.4). The promising results obtained in this out-of-distribution setting (Sect. 4.2.4) suggest that GPTCast learns representations of precipitation dynamics that possess some degree of geographical transferability. While these findings are encouraging, they represent only a first step. More extensive validation across a wider variety of geographical regions and climatological conditions would be necessary to fully establish the broad applicability and potential regional biases of the model, representing an important avenue for future research.

5.5 Inference efficiency and optimization strategies

Another important practical consideration for deploying large autoregressive transformer models such as GPTCast in operational settings is their computational cost during inference. While powerful, the attention mechanism and the sheer number of parameters can lead to significant latency and memory requirements. However, the field has developed numerous optimization techniques specifically targeting these challenges, which could be applied to GPTCast to enhance its real-time feasibility.

One major advancement is the development of optimized attention algorithms, such as FlashAttention (Dao et al., 2022), which reduces the memory footprint and increases the speed of the attention computation by avoiding materialization of the large attention matrix. Furthermore, model quantization techniques (Gholami et al., 2021) can significantly reduce the model size and accelerate inference by representing weights and activations using lower-precision integer formats (e.g., INT8) instead of floating-point numbers, often with minimal impact on predictive performance. Relatedly, inference can be performed using reduced precision formats such as FP8 (Kuzmin et al., 2022), which speeds up matrix multiplications on hardware accelerators supporting these formats. For autoregressive generation, efficiently managing the key-value (KV) cache is crucial (Pope et al., 2023); techniques optimizing KV cache storage and retrieval avoid redundant computations for previously processed tokens, drastically speeding up the generation of subsequent forecast steps. While the implementation and evaluation of these optimizations are beyond the scope of this initial study, their successful application in other domains suggests that they represent a viable path towards deploying models like GPTCast efficiently in time-critical operational nowcasting workflows.

5.6 Future work and outlook

Finally, in future studies, we also plan to explore the interpretability of the model to control and condition the model for different tasks. The peculiar characteristics of GPTCast open the possibility of guiding the generative process of the model by combining the probabilistic output of the forecaster with the interpretability of the learned codebook in terms of physical quantities. A possibility that we envision is to leverage GPTCast for tasks such as seamless forecasting (a.k.a. blending), generation of what-if scenarios, forecast conditioning, weather generation, and observation correction capabilities.

Code and data availability. Data are from Arpae Emilia-Romagna. The full, preprocessed dataset used for the presented experiments is available on Zenodo (https://doi.org/10.5281/zenodo.13692016; Franch et al., 2024a), including the generated ensemble forecasts to reproduce the verification scores. The pre-trained models are available on Zenodo (https://doi.org/10.5281/zenodo.13594332; Franch et al., 2024c). A dedicated GitHub repository (https://github.com/DSIP-FBK/GPTCast (last access: 20 August 2025) hosts the PyTorch Lightning (https://doi.org/10.5281/zenodo.3828935; Falcon and The PyTorch Lightning team, 2019) code of the models described in this paper, based on the Lightning-Hydra-Template (https://github.com/facebookresearch/hydra; Yadan, 2019), licensed under the MIT License. The repository also hosts the code to reproduce the images shown in this paper. GPTCast v1.0 GitHub release is archived on Zenodo (https://doi.org/10.5281/zenodo.13832526; Franch et al., 2024b) and allows users to download the code to reproduce the presented experiments.

Author contributions. GF conceived and conceptualized the study, designed the GPTCast architecture, implemented the code, and ran the experiments. GF and ET performed the analysis and verification of the results and wrote the article. VP, CC, and PPA provided the data and performed the data extraction, data selection, and data quality control. RW performed data format conversion. All authors revised the results and reviewed the article. MC supervised the study from end to end.

Competing interests. The contact author has declared that none of the authors has any competing interests.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims made in the text, published maps, institutional affiliations, or any other geographical representation in this paper. While Copernicus Publications makes every effort to include appropriate place names, the final responsibility lies with the authors.

Acknowledgements. We acknowledge CINECA Consortium for providing the GPU resources for training and running the experiments presented in this study.

Review statement. This paper was edited by David Topping and reviewed by two anonymous referees.

References

Agrawal, S., Barrington, L., Bromberg, C., Burge, J., Gazen, C., and Hickey, J.: Machine Learning for Precipitation Nowcasting from Radar Images, CoRR, arXiv [preprint], https://doi.org/10.48550/arXiv.1912.12132, 2019.

Ayzel, G., Scheffer, T., and Heistermann, M.: RainNet v1.0: a convolutional neural network for radar-based precipitation nowcasting, Geosci. Model Dev., 13, 2631–2644, https://doi.org/10.5194/gmd-13-2631-2020, 2020.

Bellon, A. and Austin, G. L.: The evaluation of two years of real-time operation of a short-term precipitation forecasting procedure (SHARP), J. Appl. Meteorol., 17, 1778–1787, 1978.

Bojinski, S., Blaauboer, D., Calbet, X., de Coning, E., Debie, F., Montmerle, T., Nietosvaara, V., Norman, K., Bañón Peregrín, L., Schmid, F., Strelec Mahović, N., and Wapler, K.: Towards nowcasting in Europe in 2030, Meteorol. Appl., 30, e2124, https://doi.org/10.1002/met.2124, 2023.

Bowler, N. E., Pierce, C. E., and Seed, A. W.: STEPS: A probabilistic precipitation forecasting scheme which merges an extrapolation nowcast with downscaled NWP, Q. J. Roy. Meteor. Soc., 132, 2127–2155, https://doi.org/10.1256/qj.04.100, 2006.

Dao, T., Fu, D. Y., Ermon, S., Rudra, A., and Ré, C.: FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness, Advances in Neural Information Processing Systems (NeurIPS), https://dl.acm.org/doi/10.5555/3600270.3601459 (last access: 20 August 2025), 2022.

Dixon, M. and Wiener, G.: TITAN: Thunderstorm Identification, Tracking, Analysis, and Nowcasting – A radar-based methodology, J. Atmos. Ocean. Tech., 10, 785–797, 1993.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale, arXiv [preprint], https://doi.org/10.48550/arXiv.2010.11929, 2020.

Esser, P., Rombach, R., and Ommer, B.: Taming transformers for high-resolution image synthesis, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, Nashville, TN, USA, 20–25 June 2021, 12873–12883, https://doi.org/10.1109/CVPR46437.2021.01268, 2021.

Falcon, W. and The PyTorch Lightning team: PyTorch Lightning, Zenodo [code], https://doi.org/10.5281/zenodo.3828935, 2019.

Fan, A., Lewis, M., and Dauphin, Y.: Hierarchical Neural Story Generation, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL), https://doi.org/10.18653/v1/P18-1082, 2018.

Foresti, L., Sideris, I. V., Panziera, L., Nerini, D., and Germann, U.: A 10-year radar-based analysis of orographic precipitation growth and decay patterns over the Swiss

Alpine region, Q. J. Roy. Meteor. Soc., 144, 2277–2301, https://doi.org/10.1002/qj.3364, 2018.

Fornasiero, A., Bech, J., and Alberoni, P. P.: Enhanced radar precipitation estimates using a combined clutter and beam blockage correction technique, Nat. Hazards Earth Syst. Sci., 6, 697–710, https://doi.org/10.5194/nhess-6-697-2006, 2006.

Fornasiero, A., Amorati, R., and Alberoni, P. P.: Radar Quantitative Precipitation Estimation at Arpa-Sim: A Critical Approach to Retrieve the Rainfall Rate at the Ground Level, in: Proceedings of the 5th European Radar Conference, Helsinki, vol. 30, ISBN 9789516976764, 2008.

Franch, G., Nerini, D., Pendesini, M., Coviello, L., Jurman, G., and Furlanello, C.: Precipitation Nowcasting with Orographic Enhanced Stacked Generalization: Improving Deep Learning Predictions on Extreme Events, Atmosphere, 11, 267, https://doi.org/10.3390/atmos11030267, 2020.

Franch, G., Tomasi, E., Cardinali, C., Poli, V., Alberoni, P. P., and Cristoforetti, M.: Dataset for "GPTCast: a weather language model for precipitation nowcasting", Zenodo [data set], https://doi.org/10.5281/zenodo.13692016, 2024a.

Franch, G., Tomasi, E., and Cristoforetti, M.: Code for "GPTCast: a weather language model for precipitation nowcasting", Zenodo [code], https://doi.org/10.5281/zenodo.13832526, 2024b.

Franch, G., Tomasi, E., and Cristoforetti, M.: Pretrained models for "GPTCast: a weather language model for precipitation nowcasting", Zenodo [code], https://doi.org/10.5281/zenodo.13594332, 2024c.

Gao, Z., Shi, X., Han, B., Wang, H., Jin, X., Maddix, D., Zhu, Y., Li, M., and Wang, Y.: PreDiff: precipitation nowcasting with latent diffusion models, in: Proceedings of the 37th International Conference on Neural Information Processing Systems (NeurIPS '23), New Orleans, LA, USA, 10–16 December 2023, Curran Associates, Inc., Red Hook, NY, USA, 3439, 36 pp., ISBN 9781713899921, 2023.

Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M. W., and Keutzer, K.: A Survey of Quantization Methods for Efficient Neural Network Inference, arXiv [preprint], https://doi.org/10.48550/arXiv.2103.13630, 2021.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y.: Generative adversarial nets, Association for Computing Machinery, New York, NY, USA, 139–144, https://doi.org/10.1145/3422622, 2014.

Göber, M., Christel, I., Hoffmann, D., Mooney, C. J., Rodriguez, L., Becker, N., Ebert, E. E., Fearnley, C., Fundel, V. J., Geiger, T., Golding, B., Jeurig, J., Kelman, I., Kox, T., Magro, F.-A., Perrels, A., Postigo, J. C., Potter, S. H., Robbins, J., Rust, H., Schoster, D., Tan, M. L., Taylor, A., and Williams, H.: Enhancing the Value of Weather and Climate Services in Society: Identified Gaps and Needs as Outcomes of the First WMO WWRP/SERA Weather and Society Conference, B. Am. Meteor. Soc., 104, E645–E651, https://doi.org/10.1175/BAMS-D-22-0199.1, 2023.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y.: The Curious Case of Neural Text Degeneration, in: International Conference on Learning Representations (ICLR), ISBN 979-8-3313-2198-7, 2020.

Kuzmin, A., Van Baalen, M., Ren, Y., Nagel, M., Peters, J., and Blankevoort, T.: FP8 quantization: the power of the exponent, in: Proceedings of the 36th International Conference on Neural Information Processing Systems (NeurIPS '22), New Orleans, LA, USA, 28 November–9 December 2022, Curran Associates, Inc., Red Hook, NY, USA, 1065, 12 pp., ISBN 9781713871088, 2022.

Lam, R., Pascanu, R., Puigdomènech Gimenez, M., Agrawal, S., Dapogny, C., Schmidt, M., Keck, T., Mudigonda, M., Brutlag, P., Wang, J., Chantry, M., Norman, C., Dudhia, A., Clark, R., Otte, N., Tirilly, P., Wiklendt, S., Zimmer, A., Merose, A., Petersen, S., Visram, R., Valter, D., Hess, F., See, A., Fritz, F., Bodin, T., Untema, B., Thurman, R., Targett, P., Ravenscroft, A., McGuire, P., Kabra, M., Keeling, J., Gopal, A., Cheng, H., Piotrowski, T., Battaglia, P., Kohli, P., Heess, N., and Hassabis, D.: GraphCast: AI model for faster and more accurate global weather forecasting, Science, 382, 1416–1421, https://doi.org/10.1126/science.adi2336, 2023.

Lang, S., Alexe, M., Chantry, M., Dramsch, J., Pinault, F., Raoult, B., Clare, M. C. A., Lessig, C., Maier-Gerber, M., Magnusson, L., Ben Bouallègue, Z., Prieto Nemesio, A., Dueben, P. D., Brown, A., Pappenberger, F., and Rabier, F.: AIFS-ECMWF's data-driven forecasting system, arXiv [preprint], https://doi.org/10.48550/arXiv.2406.01465, 2024.

Leinonen, J., Hamann, U., Nerini, D., Germann, U., and Franch, G.: Latent diffusion models for generative precipitation nowcasting with accurate uncertainty quantification, arXiv [preprint], https://doi.org/10.48550/arXiv.2304.12891, 2023.

Lessig, C., Luise, I., Gong, B., Langguth, M., Stadler, S., and Schultz, M.: AtmoRep: A stochastic model of atmosphere dynamics using large scale representation learning, arXiv [preprint], https://doi.org/10.48550/arXiv.2308.13280, 2023.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE/CVF international conference on computer vision, 11–17 October 2021, 10012–10022, https://doi.org/10.1109/ICCV48922.2021.00986, 2021.

Marshall, J. S. and Palmer, W. M. K.: The distribution of raindrops with size, J. Atmos. Sci., 5, 165–166, https://doi.org/10.1175/1520-0469(1948)005<0165:TDORWS>2.0.CO;2, 1948.

Panziera, L., Germann, U., Gabella, M., and Mandapaka, P. V.: NORA – Nowcasting of Orographic Rainfall by means of Analogues, Q. J. Roy. Meteor. Soc., 137, 2106–2123, https://doi.org/10.1002/qj.878, 2011.

Pope, R., Douglas, S., Chowdhery, A., Devlin, J., Bradbury, J., Levskaya, A., Heek, J., Xiao, K., Agrawal, S., and Dean, J.: Efficiently scaling transformer inference, arXiv [preprint], https://doi.org/10.48550/arXiv.2211.05102, 2023.

Pulkkinen, S., Nerini, D., Pérez Hortal, A. A., Velasco-Forero, C., Seed, A., Germann, U., and Foresti, L.: Pysteps: an open-source Python library for probabilistic precipitation nowcasting (v1.0), Geosci. Model Dev., 12, 4185–4219, https://doi.org/10.5194/gmd-12-4185-2019, 2019.

Pulkkinen, S., Chandrasekar, V., von Lerber, A., and Harri, A.-M.: Nowcasting of Convective Rainfall Using Volumetric Radar Observations, IEEE T. Geosci. Remote S., 58, 7845–7859, https://doi.org/10.1109/TGRS.2020.2984594, 2020.

Pulkkinen, S., Chandrasekar, V., and Niemi, T.: Lagrangian Integro-Difference Equation Model for Precipitation Nowcasting, J. Atmos. Ocean. Tech., 38, 2125–2145, https://doi.org/10.1175/JTECH-D-21-0013.1, 2021.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I.: Language Models are Unsupervised Multitask Learners, OpenAI Blog, 1, https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf (last access: 20 August 2025), 2019.

Ravuri, S., Lenc, K., Willson, M., Kangin, D., Lam, R., Mirowski, P., Fitzsimons, M., Athanassiadou, M., Kashem, S., Madge, S., Prudden, R., Mandhane, A., Clark, A., Brock, A., Simonyan, K., Hadsell, R., Robinson, N., Clancy, E., Arribas, A., and Mohamed, S.: Skilful precipitation nowcasting using deep generative models of radar, Nature, 597, 672–677, 2021.

Ritvanen, J., Pulkkinen, S., Moisseev, D., and Nerini, D.: Cell-tracking-based framework for assessing nowcasting model skill in reproducing growth and decay of convective rainfall, Geosci. Model Dev., 18, 1851–1878, https://doi.org/10.5194/gmd-18-1851-2025, 2025.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B.: High-resolution image synthesis with latent diffusion models, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 10684–10695, https://doi.org/10.1109/CVPR46437.2021.01268, 2022.

Seed, A. W.: A Dynamic and Spatial Scaling Approach to Advection Forecasting, J. Appl. Meteorol., 42, 381–388, https://doi.org/10.1175/1520-0450(2003)042<0381:ADASSA>2.0.CO;2, 2003.

Seed, A. W., Pierce, C. E., and Norman, K.: Formulation and evaluation of a scale decomposition-based stochastic precipitation nowcast scheme, Water Resour. Res., 49, 6624–6641, https://doi.org/10.1002/wrcr.20536, 2013.

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-C.: Convolutional LSTM network: A machine learning approach for precipitation nowcasting, Adv. Neur. In., 28, 802–810, ISBN 9781510825024, 2015.

Sideris, I. V., Foresti, L., Nerini, D., and Germann, U.: NowPrecip: localized precipitation nowcasting in the complex terrain of Switzerland, Q. J. Roy. Meteor. Soc., 146, 1768–1800, https://doi.org/10.1002/qj.3766, 2020.

Sun, J., Xue, M., Wilson, J. W., Zawadzki, I., Ballard, S. P., Onvlee-Hooimeyer, J., Joe, P., Barker, D. M., Li, P.-W., Golding, B., Xu, M., and Pinto, J.: Use of NWP for Nowcasting Convective Precipitation: Recent Progress and Challenges, B. Am. Meteorol. Soc., 95, 409–426, https://doi.org/10.1175/BAMS-D-11-00263.1, 2014.

Surcel, M., Zawadzki, I., and Yau, M. K.: A Study on the Scale Dependence of the Predictability of Precipitation Patterns, J. Atmos. Sci., 72, 216–235, https://doi.org/10.1175/JAS-D-14-0071.1, 2015.

Tomasi, E., Franch, G., and Cristoforetti, M.: Can AI be enabled to perform dynamical downscaling? A latent diffusion model to mimic kilometer-scale COSMO5.0_CLM9 simulations, Geosci. Model Dev., 18, 2051–2078, https://doi.org/10.5194/gmd-18-2051-2025, 2025.

Turner, B. J., Zawadzki, I., and Germann, U.: Predictability of Precipitation from Continental Radar Images. Part III: Operational Nowcasting Implementation (MAPLE), J. Appl. Meteorol., 43, 231–248, https://doi.org/10.1175/1520-0450(2004)043<0231:POPFCR>2.0.CO;2, 2004.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I.: Attention is all you need, Adv. Neur. In., 30, 5999–6009, ISBN 9781510860964, 2017.

Wang, Y., Gao, Z., Long, M., Wang, J., and Philip, S. Y.: Predrnn++: Towards a resolution of the deep-in-time dilemma in spatiotemporal predictive learning, in: International conference on machine learning, PMLR, 5123–5132,ISBN 9781510867963, 2018.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P.: Image quality assessment: from error visibility to structural similarity, IEEE T. Image Process., 13, 600–612, 2004.

Werner, M. and Cranston, M.: Understanding the value of radar rainfall nowcasts in flood forecasting and warning in flashy catchments, Meteorological Applications: A journal of forecasting, practical applications, Training Techniques And Modelling, 16, 41–55, 2009.

Wernli, H., Paulat, M., Hagen, M., and Frei, C.: SAL – A novel quality measure for the verification of quantitative precipitation forecasts, Mon. Weather Rev., 136, 4470–4487, 2008.

Wernli, H., Hofmann, C., and Zimmer, M.: Spatial forecast verification methods intercomparison project: Application of the SAL technique, Weather Forecast., 24, 1472–1484, 2009.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A.: Transformers: State-of-the-Art Natural Language Processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, edited by: Liu, Q. and Schlangen, D., Association for Computational Linguistics, Online, 38–45, https://doi.org/10.18653/v1/2020.emnlp-demos.6, 2020.

Woo, W.-C. and Wong, W.-K.: Operational Application of Optical Flow Techniques to Radar-Based Rainfall Nowcasting, Atmosphere, 8, 48, https://doi.org/10.3390/atmos8030048, 2017.

Yadan, O.: Hydra – A framework for elegantly configuring complex applications, Github [code], https://github.com/facebookresearch/hydra (last access: 20 August 2025), 2019.

Yu, J., Li, X., Koh, J. Y., Zhang, H., Pang, R., Qin, J., Ku, A., Xu, Y., Baldridge, J., and Wu, Y.: Vector-quantized Image Modeling with Improved VQGAN, in: International Conference on Learning Representations, https://openreview.net/forum?id=pfNyExj7z2 (last access: 20 August 2025), 2022.

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 586–595, https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00068 (last access: 20 August 2025), 2018.

Zhang, Y., Long, M., Chen, K., Xing, L., Jin, R., Jordan, M. I., and Wang, J.: Skilful nowcasting of extreme precipitation with NowcastNet, Nature, 619, 526–532, 2023.