CRPS-LAM: PROBABILISTIC REGIONAL WEATHER FORECASTING WITH CONTINUOUS RANKED PROBABILITY SCORE

Anonymous authors
Paper under double-blind review

ABSTRACT

Limited-Area Models (LAMs) enable weather forecasting over regional domains at higher resolutions than what is computationally feasible for global models. At such high resolutions, machine learning approaches for weather prediction increasingly rely on ensemble methods to produce probabilistic forecasts. However, existing machine learning LAMs are not scalable due to relying on computationally costly diffusion models or inefficient graph neural networks. We tackle this by introducing a new hybrid CNN/GNN architecture, tailored to the LAM weather forecasting problem. Using this architecture, we construct the DET-LAM deterministic model, producing LAM forecasts both more efficiently and accurately than its graph-based competitor. We then tackle the ensemble forecasting problem, by using this architecture as a backbone for the generative model CRPS-LAM. CRPS-LAM is trained using a Continuous Ranked Probability Score (CRPS) objective, enabling efficient training and sampling in a single forward pass. This yields a speedup of $\approx \times 39$ compared to diffusion-based baselines. We evaluate our approach on regional domains in northern Europe, demonstrating that CRPS-LAM produces skillful and well-calibrated forecasts across a range of atmospheric variables.

1 INTRODUCTION

Machine learning methods for simulating atmospheric dynamics are revolutionizing modern weather forecasting (Bi et al., 2023; Lam et al., 2023; Lang et al., 2024; Price et al., 2025; Lang et al., 2026; Bonev et al., 2025). Offering fast and accurate forecasts, these Machine Learning Weather Prediction (MLWP) methods have great potential for predicting and understanding both everyday weather conditions and extreme events. While initial MLWP methods used deterministic models to produce single forecasts (Bi et al., 2023; Lang et al., 2024), the field is increasingly moving towards more useful ensemble forecasting models (Price et al., 2025; Lang et al., 2026; Bonev et al., 2025; Alet et al., 2025). These methods use conditional generative models to sample forecasts from the distribution of possible future atmospheric states. Such sampled ensemble members can then be used to estimate probabilities of specific atmospheric conditions, or studied by meteorologists in order to understand possible future scenarios. These developments towards MLWP ensemble forecasting are happening both for the global domain, and for the regional setting considered in this work. Regional MLWP allows for building models operating at higher resolutions than what is possible in a global setting, due to excessive memory and compute costs. Regional models also enable training on region-specific datasets that provide finer-grained or locally calibrated information (Yang et al., 2026; Ridal et al., 2024).

Initial attempts at MLWP ensemble forecasting for regional domains have relied on simplified and often unrealistic problem settings, and either fail to capture high-frequency features (Oskarsson et al., 2024) or use computationally expensive diffusion models (Pathak et al., 2026; Larsson et al., 2025; Abdi et al., 2026). Furthermore, compared to previous work, we adopt a more realistic formulation for boundary conditioning following Adamov et al. (2025) and extend it to the probabilistic setting.

Building on a methodology that has been proven successful in global forecasting (Alet et al., 2025), we show that regional ensemble forecasting models can be trained effectively by matching marginal distributions using a Continuous Ranked Probability Score (CRPS) loss. We propose a probabilistic

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

regional forecasting model where stochasticity is introduced through a single latent noise vector, enabling the generation of skillful and realistic ensemble members in a single forward pass.

Limited-Area Modeling Problem Definition. In this paper, we study probabilistic MLWP in the context of Limited-Area Modeling (LAM). We adopt the problem formulation of Adamov et al. (2025) and extend it to the probabilistic forecasting setting.

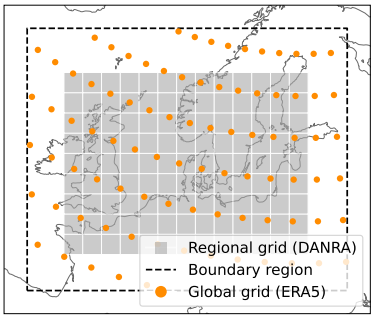


Figure 1: Regional and global grid points might not align in our setting. Low resolution example for clarity.

In LAM forecasting, the goal is to produce high-resolution forecasts over a bounded spatial domain Ω^I, which we refer to as the *interior*. While the model is restricted to a limited domain, there is also influence from large-scale atmospheric dynamics from outside the region, particularly at longer lead times. To account for this, we condition on information from a boundary region Ω^B, for example obtained from a global forecast at coarser resolution. Importantly, this conditioning is treated as a *soft* rather than a hard constraint, allowing the regional model to deviate from the global model where appropriate and thereby avoid inheriting its systematic biases. The inputs to the model are:

- Initial interior states X^I_{t-1}, X^I_t within Ω^I. These are assumed to be given, for example, from a data assimilation system in an operational setting.
- Interior forcing inputs F^I_t within Ω^I, which may include time-dependent variables (e.g., time of day) as well as static spatial features (e.g., orography and land–sea masks).
- Boundary information B_t, consisting of atmospheric states $X^B_{t-1}, X^B_t, X^B_{t+1}$ and forcing features F^B_t, defined on Ω^B. Note that Ω^B may spatially overlap with Ω^I but boundary information is typically available at a distinct and lower-resolution grid (see fig. 1).

Our objective is to generate forecasts over the interior domain by modeling the conditional distribution $p\big(X^I_{t+1} \mid X^I_t, X^I_{t-1}, F^I_t, B_t\big)$, which can be accomplished by a mean prediction from a deterministic model or an ensemble forecast by a probabilistic model. An overview of the limited-area forecasting setup and our model is illustrated in fig. 3.

Our main contributions are:

1. We propose a hybrid Graph Neural Network (GNN) and Convolutional Neural Network (CNN) architecture to be used as the backbone for LAM weather models. This design combines the best of both worlds: handling spatially irregular boundary inputs using the GNN component while leveraging the computational and memory benefits of CNNs.
2. We introduce **DET-LAM**, a deterministic forecasting model which has lower Root Mean Squared Error (RMSE) than the graph-based Graph-FM for 95% of variables and lead times.
3. We introduce **CRPS-LAM**, a probabilistic forecasting model that produces accurate ensemble forecasts with lower CRPS than the graph-based Graph-EFM in 99% of variable–lead time combinations, while generating a forecast in a single forward pass.
4. We evaluate our models for high-resolution forecasting over northern European domains, showing the advantage over purely graph-based existing methods.

2 Related Work

Probabilistic Global Weather Forecasting. Probabilistic global MLWP has seen significant progress through using generative models based on latent variable formulations (Oskarsson et al., 2024; Hu et al., 2023), diffusion models (Price et al., 2025; Andrae et al., 2025; Shi et al., 2024), and flow-matching approaches (Couairon et al., 2026). While flow-based methods have demonstrated strong performance, they typically require iterative sampling at inference time, resulting in substantial computational overhead (a factor ≈ 39 in our experiments, section B.1).

Stock et al. (2026) address this limitation by distilling a global weather diffusion model into a consistency model, capable of generating samples in a single forward pass. Others immediately train for single step forecasting, injecting random noise as a model input and training using a CRPS loss function to match the correct distribution (Pacchiardi et al., 2024; Lang et al., 2026; 2025; Bonev et al., 2025; Alet et al., 2025; Oskarsson et al., 2024; Cachay et al., 2026). These methods differ primarily in how the CRPS is estimated and how the stochasticity is incorporated into the model.

Limited-Area Modeling (LAM). In the LAM setting, regional MLWP is achieved by combining coarse resolution boundary information with dynamics simulated forward on a high-resolution grid. Deterministic machine learning LAMs include Graph-FM (Oskarsson et al., 2023; Adamov et al., 2025), using a GNN architecture, and YingLong (Xu et al., 2025), using a neural operator approach. Probabilistic LAM forecasting has so far mainly been explored using diffusion models (Pathak et al., 2026; Larsson et al., 2025; Abdi et al., 2026). As for global forecasting, diffusion approaches can achieve strong predictive performance but at a high computational cost. Oskarsson et al. (2024) propose a LAM version of the Graph-EFM latent variable model, capable of single step sampling. Graph-EFM incorporates CRPS-based regularization, but uses a more involved training objective with additional hyperparameters, compared to our approach. As shown in our experiments, Graph-EFM also tends to produce overly smooth forecast fields. While existing probabilistic methods all incorporate boundary information, they all do so at the same spatio-temporal resolution as the regional domain. In real-world scenarios, boundary data typically has a much lower resolution. Only Adamov et al. (2025) handle boundary inputs at a different resolution, but only for deterministic models. We improve on their approach with our new hybrid architecture and extend it to the probabilistic setting.

High-Resolution Regional Weather Forecasting. Apart from LAMs, alternative approaches to high-resolution regional forecasting include stretch-grid models and downscaling. Stretch-grid models (Nipen et al., 2026) combine global and regional forecasting by using a higher resolution over a specific region of the globe. Nordhagen et al. (2025) extend the stretch-grid approach to probabilistic forecasting with a similar CRPS-based training as CRPS-LAM, however the method still requires full global weather simulations, making it less scalable than a LAM approach. Downscaling instead aims to produce high-resolution weather fields by learning a mapping from a low-resolution global forecast to high-resolution regional predictions (Mardani et al., 2025; Srivastava et al., 2024; Larsson et al., 2026). While this enables generating regional predictions, downscaling methods do not explicitly simulate the underlying fine-scale physical processes at high resolution, and can thus not incorporate information from regional initial conditions. As our methods combine both regional initial conditions and coarse boundary information, they can seamlessly transition from simulating dynamics to pure downscaling as the information in the initial condition becomes less relevant.

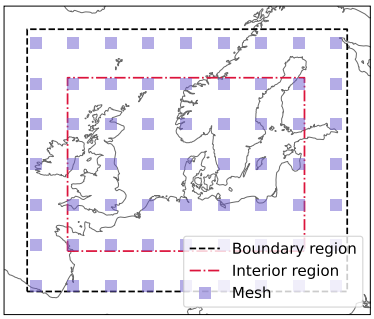


Figure 2: Example mesh covering both interior and boundary regions.

3 METHOD

Method Overview. We approach the LAM forecasting problem by constructing a hybrid CNN/GNN backbone architecture, and training it as a generative model using the CRPS scoring rule as the loss function. This backbone leads to improved memory and compute efficiency compared to previous approaches. We use the backbone model f_θ as

$$\hat{X}^I_{t+1} = f_\theta(X^I_t, X^I_{t-1}, F^I_t, B_t, z) \tag{1}$$

where z is latent Gaussian noise used to sample an ensemble member one step into the future in a single forward pass. The CRPS loss function provides a straightforward way to train such a model for calibrated forecasts. We next describe details of the different parts in this modeling framework.

Architecture. Our architecture follows an encode–process–decode framework (Battaglia et al., 2018). First, all inputs are mapped onto a shared mesh representation; processing is then performed on this mesh, and finally the forecast is decoded over the interior region of interest. We carefully

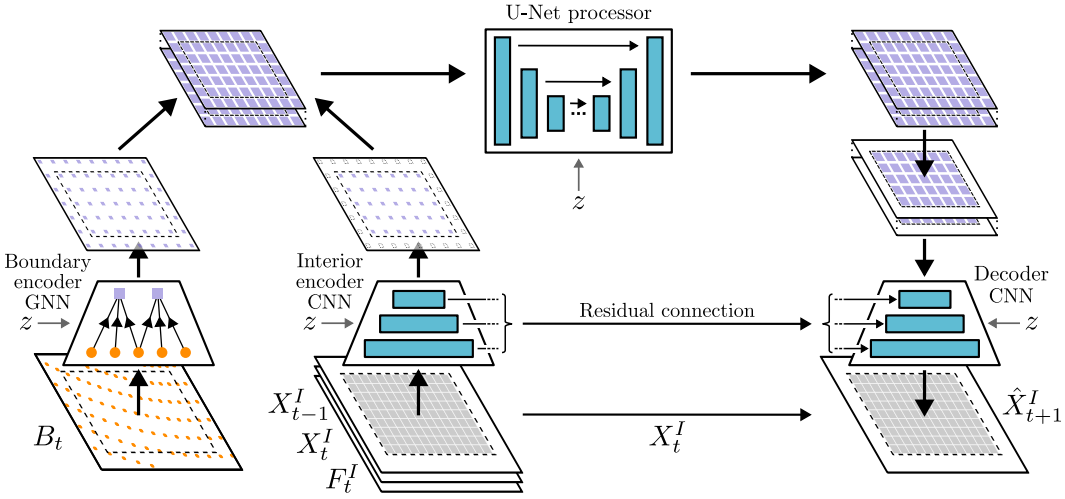


Figure 3: Overview of the CRPS-LAM backbone architecture. The same latent noise vector z is used in each component, but linearly transformed by independent parameters before each conditional normalization layer.

choose this mesh such that it lines up with a subset of points of the interior grid, but also extends out across the boundary region (see fig. 2). An overview of the full architecture is shown in fig. 3.

In the LAM setting, interior and boundary data may be defined on different grids, motivating the use of separate encoders. The interior data are typically given on a regular grid, and we therefore follow Siddiqui et al. (2024) and employ a U-Net architecture to efficiently extract spatial features. In contrast, the boundary data are defined on a lower-resolution latitude–longitude grid, for which we use a GNN encoder to flexibly map the inputs onto the regular grid mesh.

In early experiments with a purely GNN-based architecture (following Adamov et al. (2025)), we found that GNN modules incur a substantial memory footprint. This motivated a hybrid design in which CNN-based modules are used whenever the data lie on a regular grid, while GNNs are retained only for irregular boundary grids. To further reduce memory usage, we use only 32 dimensions for edge representation vectors, compared to the much higher hidden dimension used in other model components. This results in a significant reduction in memory cost, without negatively affecting performance. The mesh itself is constructed on a regular grid, which allows us to again leverage a U-Net architecture for the processing stage. After processing, the interior nodes of the mesh are decoded using a CNN-based decoder to produce forecasts over the interior domain.

Deterministic Model (DET-LAM). Our efficient backbone can directly be used as a deterministic LAM model by simply setting the noise variables z to a constant 1. We call this model DET-LAM. Following earlier works on deterministic LAMs (Adamov et al., 2025; Oskarsson et al., 2023; Wijnands et al., 2025), we train this model by minimizing a Mean Squared Error (MSE) loss function averaged over variables, grid points and lead times. DET-LAM is first trained for single-step prediction, and then finetuned using rolled out forecasts.

Probabilistic Model (CRPS-LAM). Scoring rules provide a principled framework for evaluating probabilistic forecasts and have been used to train ensemble forecasting models (Lang et al., 2026; Bonev et al., 2025; Alet et al., 2025). A more detailed introduction to CRPS as a training objective is provided in section A. Following Alet et al. (2025), we introduce stochasticity via a low-dimensional latent variable. Specifically, a noise vector $z \sim \mathcal{N}(0, I)$ of dimension 32 is transformed through a linear layer and injected into the network using conditional normalization layers (Chen et al., 2021; Song et al., 2021; Karras et al., 2022). Given interior states X^I_t, X^I_{t-1}, forcing features F^I_t and boundary conditions B_t, the model defines a conditional distribution $p\big(X^I_{t+1} \mid X^I_t, X^I_{t-1}, F^I_t, B_t\big)$. Multi-step forecasts are obtained through autoregressive rollouts, where predictions are recursively fed back as inputs. We train the model by minimizing the almost-fair CRPS estimator $\text{CRPS}_{\alpha\text{-fair}}$ (Lang et al., 2026) using two sampled ensemble members $\hat{X}^{I,(1)}_{t+1}$ and $\hat{X}^{I,(2)}_{t+1}$. The overall loss is

defined as

$$\mathcal{L} = \frac{1}{T|G|} \sum_{t=0}^{T-1} \sum_{g \in G} \sum_{d} \omega_d \sigma_d \text{CRPS}_{\alpha\text{-fair}}(\hat{X}^{I,(1)}_{t+1,g,d}, \hat{X}^{I,(2)}_{t+1,g,d}, X^I_{t+1,g,d}) \tag{2}$$

where the CRPS is computed for each lead time t, grid point g and variable d by comparing predictions from the two ensemble members and the ground truth $X^I_{t+1,g,d}$. This is then weighted by the inverse variance σ_d of the one-step-differences for the variable, as well as a manual variable-specific weight ω_d. Finally the CRPS is summed over variables and averaged across all grid points $G \subset \Omega^I$ and lead times $1, \ldots, T$. We choose the manual weights ω_d similar to existing works for the datasets used in experiments (Oskarsson et al., 2023; Adamov et al., 2025). Training follows a similar rollout schedule as the deterministic model.

While the point-wise loss primarily enforces marginal consistency at each grid point, joint dependencies are captured through the shared latent variable z, which influences all outputs simultaneously. Additionally, the use of convolutional architectures promotes spatial coherence through weight sharing. We have also experimented with adding a spectral loss component, similar to Nordhagen et al. (2025) and Bonev et al. (2025). However, this requires choosing additional weighting factors, and we found that rather than improving model performance it could introduce artifacts in the forecasts.

We also investigated dropout-based uncertainty estimation following Cachay et al. (2026), applying different dropout rates to the DET-LAM model to obtain a probabilistic forecasting model. In our setting, however, this resulted either in unstable training or pronounced spatial artifacts in the forecasts. We therefore do not include this approach as a baseline.

4 Experiments

We start by evaluating the models in a simplified setting using the MEPS dataset in section 4.1, where we compare our CRPS-based approach to Diffusion-LAM (Larsson et al., 2025), a CNN-based diffusion model, and Graph-EFM (Oskarsson et al., 2024), a graph-based latent variable approach. We then experiment with the more realistic high-resolution DANRA dataset in section 4.2. Forecasts are evaluated using point-wise metrics, including RMSE, Spread-Skill Ratio (SSR), and CRPS. In addition, we analyze the spectral characteristics of the predictions.

4.1 A Simplified Setting: the MEPS Dataset

We start by considering a previously studied simplified version of the problem defined above, in which the boundary and interior share the same spatial grid and do not overlap (Oskarsson et al., 2023; 2024; Larsson et al., 2025). In particular, we decompose the domain $\Omega = \Omega^B \cup \Omega^I$ into a non-overlapping interior domain Ω^I and a boundary domain Ω^B such that $\Omega^B \cap \Omega^I = \emptyset$, but assume that all variables are defined on a common regular grid. The boundary states X^B_t are extracted directly from the ground truth on the same grid, rather than coming from an external coarse-resolution global model. Consequently, the conditioning on B_t does not involve handling a different spatial resolution. Since both Ω^I and Ω^B lie on the same regular grid, we can use this grid as our mesh. Hence, the grid-to-mesh encoders used in the general setting can be reduced to separate multilayer perceptrons applied pointwise on the interior and boundary inputs. In particular, no graph-based operator is required to map between irregular and regular representations. An overview of the resulting limited-area setup is shown in fig. 4, and additional details are provided in section F.

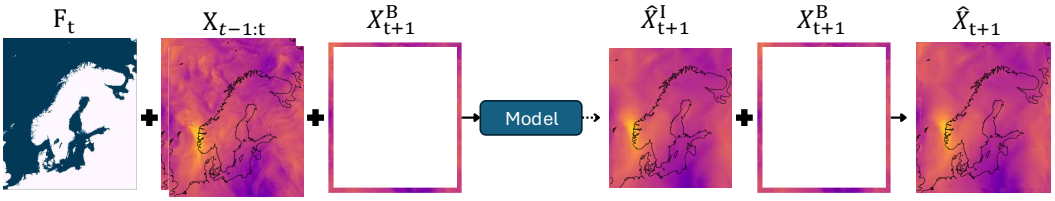


Figure 4: An overview of the forecasting process showing the inputs and outputs of the models in the MEPS problem setting (adapted from Larsson et al. (2025)).

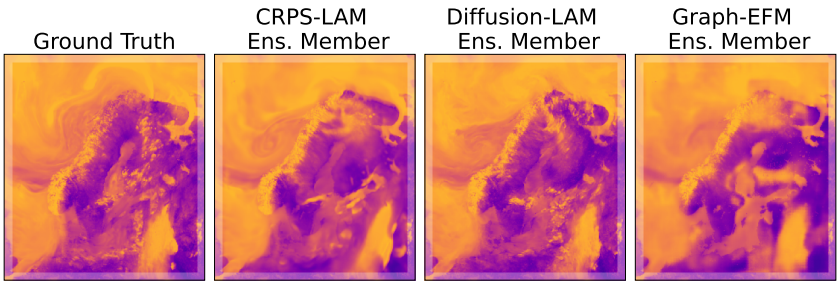
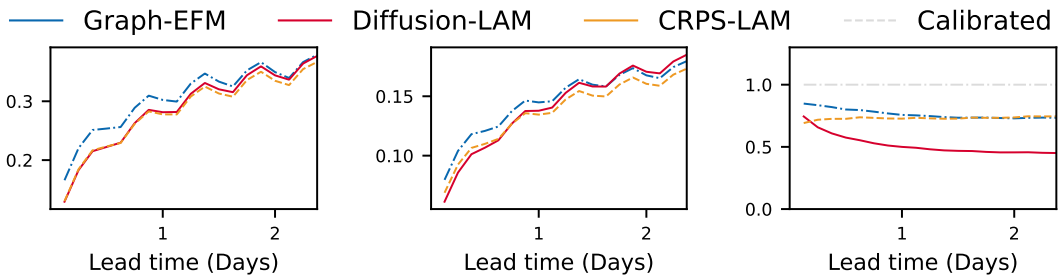


Figure 5: Forecasts at 57 h lead time for relative humidity at 2 m (`r_2`). The faded area constitutes the boundary region.

Experiments. The models are evaluated on the MEPS LAM dataset over the Nordic region. All forecast trajectories are initialized from the ground-truth states and generated autoregressively with 3 h steps up to a lead time of 57 h, with 25 ensemble members produced for each model. The results show that CRPS-LAM achieves performance competitive with both Diffusion-LAM and Graph-EFM in terms of the sample quality exemplified in fig. 5, as well as pixelwise performance metrics in fig. 6. A well-calibrated ensemble should have SSR ≈ 1, yet all models exhibit some degree of underdispersion. CRPS-LAM achieves ensemble calibration comparable to the best-performing model, Graph-EFM, whereas Diffusion-LAM shows stronger underdispersion, particularly at longer lead times. By investigating the spectra of the fields produced by the models in figs. 42 to 45 in the appendix, we also note that CRPS-LAM retains more fine-scale details than Graph-EFM, but loses some of the highest frequency components still captured by Diffusion-LAM. This can also be seen in fig. 5 and in the more detailed evaluation in section F. In this setting, we find that producing



(a) RMSE (b) CRPS (c) SSR

Figure 6: The mean of the normalized RMSE, CRPS, and SSR for all variables on the MEPS dataset.

ensemble forecasts with the diffusion approach is ≈ 39 times slower than CRPS-LAM, due to the sequential sampling. Given the encouraging results of CRPS-LAM on the MEPS data, this validates the approach as an efficient and accurate ensemble forecasting method. We next scale up the model to a more realistic LAM setting.

4.2 Realistic High-Resolution LAM Forecasting: DANRA

The simplified assumption in the MEPS experiments and in previous works (Oskarsson et al., 2024; Larsson et al., 2025), that boundary information B_t is defined on the same grid as the interior states X_t^I and is extracted from ground truth, are not realistic for an operational LAM setting. In practice, the boundary domain Ω^B is provided by an external global forecasting system and typically differs from the interior domain Ω^I in both spatial resolution, grid structure, and possibly time step. As a result, the boundary information B_t is defined on a grid that is not aligned with that of X_t^I, and we must use an encoder that can map this representation to a common representation for the processor.

In this section, we consider this more realistic formulation of the problem and demonstrate how our framework accommodates heterogeneous inputs across Ω^I and Ω^B. In particular, we consider B_t obtained either from a reanalysis dataset or from an operational global forecast, both defined on a

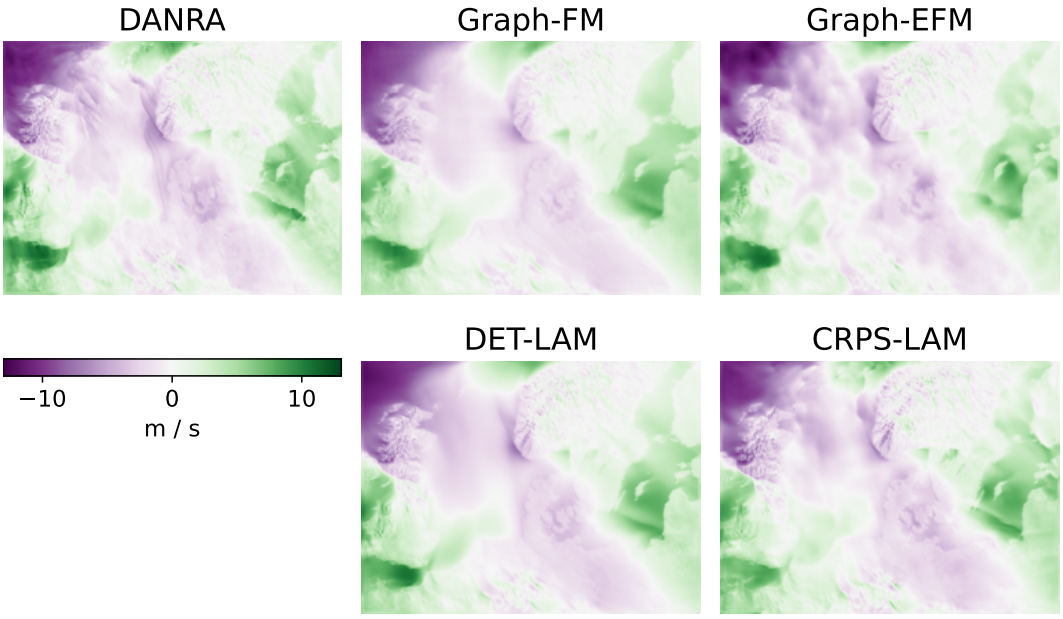


Figure 7: Forecasts at 72 h lead time for the u wind component at 10 m (`u10m`). For the probabilistic models (Graph-EFM and CRPS-LAM) we show one ensemble member.

different grid from the interior. Additionally, the interior is sampled at 3 h intervals, whereas the boundary data is available at coarser 6 h intervals.

DANRA Setting. In this experiment we train the model on the DANRA (Yang et al., 2026) high-resolution regional reanalysis dataset, aiming to learn a forecasting model for the domain shown in figs. 1 and 2. The boundary region extends 400 km beyond the interior, and we include boundary grid points overlapping with the interior, meaning that $\Omega^I \subset \Omega^B$. For a more detailed description of the data we refer the reader to section D and Adamov et al. (2025).

Interior Data. For the interior we use DANRA data covering 1472×1972 km at 3 h time steps and a spatial resolution of 2.5 km (589×789 pixels). We model 7 surface variables and 6 vertical variables on pressure levels 100, 200, 400, 600, 700, 850, 925, and 1000 hPa. The state X_t^I has a dimension of 55 at each grid point, where surface variables and pressure levels are concatenated, resulting in a total dimensionality of 25 million for X_t^I. In addition to the atmospheric states we include the forcing shown in table 4 in the appendix.

Boundary Data. We use a boundary region extending 400 km beyond the interior, which is deemed sufficient for this setting in an investigation by Adamov et al. (2025). To enable laying out the mesh as a regular grid we specify the boundary area as a rectangle in the local coordinate system, meaning that some corner points will be further than 400 km away from the interior. We consider two sources of boundary data, both at 0.25° resolution with 6 h time steps: ERA5 as provided through WeatherBench 2 (Rasp et al., 2024), and operational IFS forecasts (ECMWF, 2024) with the same variables. The spatial mismatch with the interior is handled by our hybrid architecture. For the temporal mismatch we rely on separate boundary encoders to encode the timestamp (Adamov et al., 2025).

Baselines. Apart from CRPS-LAM and DET-LAM we consider the deterministic Graph-FM model (Adamov et al., 2025), which was developed for this high-resolution DANRA setting. Graph-FM is a fully graph-based model, and we use the original implementation of Adamov et al. (2025) with an 800 km boundary region without spatial overlap with the interior. As an ensemble forecasting baseline, we also train a version of Graph-EFM (Oskarsson et al., 2024) on the DANRA setting. We also trained Diffusion-LAM with the largest backbone that fits in the 80 GB memory limit (277M parameters) and more tuning than any other model, but its samples retained substantial noise and its forecasts were not realistic, so we do not include it among the DANRA baselines (details in

section E.1). At this size, a 20-member 72 h ensemble takes more than 5 h with Diffusion-LAM on one A100 GPU, compared to about 8 min with CRPS-LAM (table 1). We compare our models also with an operational Numerical Weather Prediction (NWP) baseline (Bengtsson et al., 2017). Unfortunately, the NWP forecasts include only a limited set of variables, at lead times up to 18 h (Adamov et al., 2025).

ERA5 Training and IFS Finetuning. All models are first trained with ERA5 boundaries. As ERA5 is a reanalysis, the models then have accurate boundary information at every lead time, which would not be available in operational use. We refer to this as the *ERA5* setting and treat it as an upper bound on what each model can achieve. Operationally, the boundary instead has to come from a global forecast whose errors grow with lead time. When we ran the ERA5-trained models with IFS boundaries, skill decreased and the ensembles were clearly underdispersed. The SSR dropped further with lead time (fig. 19). We therefore finetuned all models with IFS boundaries for 10 additional epochs, using rollouts of up to 12 steps (36 h; see section B). This improved both skill and calibration (figs. 20 and 21), and all IFS results below use the finetuned models, denoted *IFS Finetuned*.

Gridded Evaluation Against Reanalysis. We evaluate against the DANRA reanalysis using the RMSE (of the ensemble mean for the probabilistic models), CRPS, SSR, and Log Spectral Distance (LSD). Example forecasts are shown in fig. 7, and per-variable results for both settings are given in tables 5 to 9 and **??**. In the IFS Finetuned setting (fig. 8), DET-LAM has lower RMSE than Graph-FM in 157 of 165 variable–lead time combinations (95%), and CRPS-LAM improves over Graph-EFM in 164/165 (99%) for CRPS and 153/165 (93%) for RMSE. The margins are larger with ERA5 boundaries (figs. 10 and 11), where the median RMSE reduction of DET-LAM over Graph-FM is 33–38% across lead times, compared to 12% at 24 h and 3% at 72 h with IFS boundaries. Similarly, the median CRPS reduction of CRPS-LAM over Graph-EFM goes from 18% to 12% over the same lead times with IFS boundaries. As the forecast progresses, errors in the IFS boundary forcing account for a growing share of the total error, and this limits how much any LAM can gain from a better interior model.

DET-LAM generally has lower RMSE than CRPS-LAM (tables 6 and 7). This is expected, since the MSE objective directly targets this metric, while the ensemble mean is only an estimate of the predictive mean that improves with more members. If only the mean forecast is of interest, DET-LAM is the more suitable model. CRPS-LAM is intended for when forecast uncertainty is also needed, which CRPS and SSR measure but RMSE does not.

CRPS-LAM is also better calibrated than Graph-EFM (fig. 12 and table 8). With IFS boundaries, its SSR is closer to 1 than that of Graph-EFM for every variable and lead time, and the ensemble is close to calibrated for the first day and a half: up to 33 h the median SSR over variables stays between 0.87 and 1.01, and about 90% of the

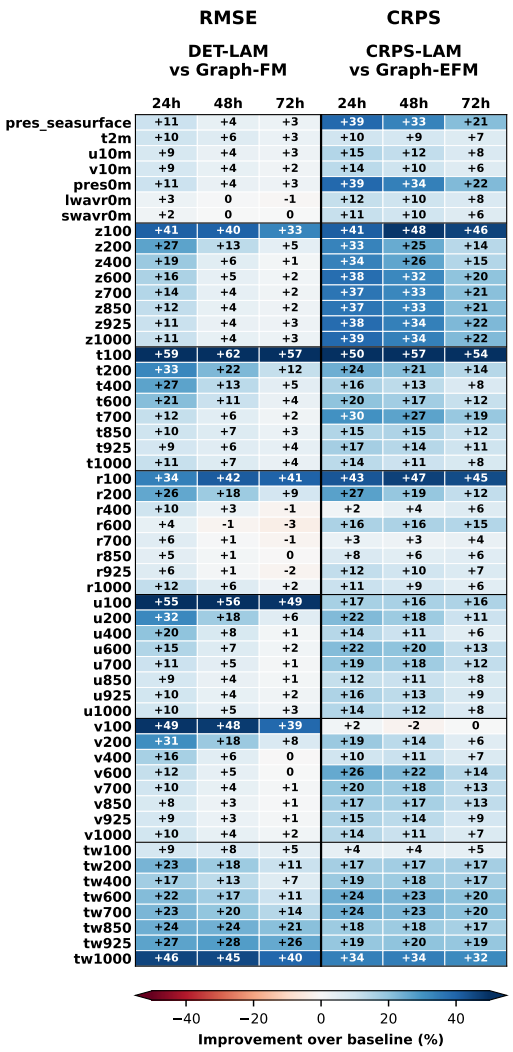


Figure 8: Relative improvement (%) in the IFS Finetuned setting: RMSE of DET-LAM over Graph-FM (left) and CRPS of CRPS-LAM over Graph-EFM (right).

variables have an SSR between 0.8 and 1.2. After this, CRPS-LAM becomes increasingly underdispersed, with a median SSR of 0.80 at 48 h and 0.69 at 72 h. The drop starts where the 36 h finetuning rollouts end, which suggests that longer rollouts could extend the calibrated range. Graph-EFM is clearly underdispersed throughout, with a median SSR of 0.64 at 3 h that falls to 0.37 at 72 h. With ERA5 boundaries, the median SSR of CRPS-LAM stays between 0.94 and 1.0 at all lead times (fig. 17). Most cases where Graph-EFM achieves better calibration are upper-level winds, for which CRPS-LAM is somewhat overdispersed.

In the IFS Finetuned setting, DET-LAM and CRPS-LAM also have lower LSD than their respective baselines in 76% and 74% of the combinations (fig. 14 and table 10), indicating better preserved spatial structure. Graph-FM shows the largest difference, over-smoothing many of the detailed fields. The main exception is 2 m temperature, where DET-LAM has the highest LSD of all models while CRPS-LAM has the lowest. We also note that Graph-EFM occasionally exhibits visual artifacts in both its predictions and estimated standard deviations (section E.2), potentially due to its graph-based architecture (Oskarsson et al., 2024).

Evaluation against in-situ observations. We further evaluate the IFS Finetuned models against in-situ observations from weather stations in Denmark, at the center of the DANRA domain. We use measurements of 2 m temperature and 10 m wind from 50 stations over the test period, and compute the RMSE, SSR, and CRPS at these locations (see fig. 9). As shown in fig. 22, the MLWP models are competitive with the NWP model in terms of RMSE. This is particularly evident for the 10 m u-wind component and for short-range forecasts (up to 6 h) of 2 m temperature. CRPS-LAM has lower CRPS than Graph-EFM, except at the shortest lead times where the two models perform very similarly. The SSR results indicate that neither CRPS-LAM nor Graph-EFM is well calibrated with respect to the observations. This is expected, as the observational distribution differs from the reanalysis data used as training target. The shift between reanalysis and observations leads to an increase in error, but the models are not trained to match this with an increase in spread. This calibration gap could be reduced by finetuning with model outputs directly compared to observations if available during training.

5 Discussion

We introduce a new backbone architecture for LAM weather forecasting that combines GNN and CNN components. The hybrid design reduces memory use enough to scale our models to ten times as many parameters as the graph-based baselines at a similar or lower inference cost (table 1). Building on this backbone, we propose DET-LAM, a deterministic model, and CRPS-LAM, a probabilistic model that samples each ensemble member in a single forward pass, about $39\times$ faster than a diffusion-based model. In high-resolution experiments over Denmark, DET-LAM and CRPS-LAM have lower errors than their graph-based counterparts, and CRPS-LAM produces better calibrated ensembles and more realistic fields than Graph-EFM.

Limitations and Future Work. In our DANRA experiments, the ML models are initialized from DANRA reanalysis states rather than operational analysis states, giving them a slight advantage over the NWP baseline, although comparisons between the ML models remain fair. We also restrict the models to a subset of atmospheric variables due to data availability and prior work (Adamov et al., 2025); extending the models to a broader variable set is an important direction for future work. We were not able to train a working diffusion baseline on DANRA within the 80 GB memory limit (section E.1), so CRPS-LAM is compared only with Graph-EFM as a probabilistic baseline on this dataset. For forecasts with IFS boundary forcing, the models do not account for uncertainty in the boundary conditions, making it difficult to maintain a well-calibrated ensemble beyond the 36 h rollout horizon used during training. Using ensemble boundary forecasts could provide a way to propagate boundary uncertainty into the regional forecast. In addition, neither probabilistic model is currently calibrated against station observations, leaving potential distributional differences between reanalysis and station observations unaccounted for. Incorporating station observations during training could help address this mismatch and improve forecast reliability. More broadly, it would be valuable to investigate alternative sources of boundary information, including ML-based and ensemble forecasts, and to determine suitable adaptation or fine-tuning strategies for these settings. Finally, cross-domain transfer remains an open question, particularly whether the models can generalize zero-shot or with minimal fine-tuning to regions where high-quality reanalysis data are limited.

AI USE STATEMENT

In this work, we used generative AI tools to assist with translation. We have not used generative AI tools to generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, clean and reformat dataset, support qualitative and thematic data analysis, interpret results. Additionally, we used generative AI tools to edit the research paper to improve readability and to debug software code. We have reviewed all AI-assisted work. LLM-generated code and text suggestions were verified and tested for correctness by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Weather forecasts are used for decisions that affect safety, for example warnings of storms or extreme temperatures. The models in this paper are research prototypes and should not be used operationally without further validation in collaboration with domain experts. Training the models required substantial compute, about 23,000 H200 GPU-hours for the final runs of the five models in table 1, plus the additional Diffusion-LAM runs and smaller preliminary experiments. On the other hand, once trained, CRPS-LAM produces a 20-member 72 h ensemble in about 8 min on a single GPU, which is far cheaper than running a physics-based ensemble at the same resolution.

The authors declare no conflicts of interest.

REPRODUCIBILITY STATEMENT

The problem formulation is given in the introduction, and the architecture and training objectives of DET-LAM and CRPS-LAM are described in the method section, with the CRPS estimator detailed in section A. Section B lists hyperparameters, training schedules, optimizer settings and hardware for all models, including the ERA5 training and IFS finetuning stages, and section B.1 reports parameter counts, memory use, training cost and inference time. All evaluation metrics are defined in section C. The DANRA data, variables, forcing and data splits are described in section D, following Adamov et al. (2025), and the MEPS setup, splits and training schedule are given in section F. The DANRA and MEPS datasets are publicly available, and ERA5/IFS is available through WeatherBench 2. We will release the code, configuration files and trained model weights upon acceptance.

REFERENCES

Daniel Abdi, Isidora Jankov, Paul Madden, Vanderlei Vargas, Timothy A. Smith, Sergey Frolov, Montgomery Flora, and Corey Potvin. Hrrrcast: A data-driven emulator for regional weather forecasting at convection-allowing scales. *Artificial Intelligence for the Earth Systems*, 5(2):250061, 2026. doi: 10.1175/AIES-D-25-0061.1.

Simon Adamov, Joel Oskarsson, Leif Denby, Tomas Landelius, Kasper Hintz, Simon Christiansen, Irene Schicker, Carlos Osuna, Fredrik Lindsten, Oliver Fuhrer, and Sebastian Schemm. Building machine learning limited area models: Kilometer-scale weather forecasting in realistic settings. *arXiv preprint arXiv:2504.09340*, 2025. URL https://arxiv.org/abs/2504.09340.

Ferran Alet, Ilan Price, Andrew El-Kadi, Dominic Masters, Stratis Markou, Tom R. Andersson, Jacklynn Stott, Remi Lam, Matthew Willson, Alvaro Sanchez-Gonzalez, and Peter Battaglia. Skillful joint probabilistic weather forecasting from marginals. *arXiv preprint arXiv:2506.10772*, 2025. URL https://arxiv.org/abs/2506.10772.

Martin Andrae, Tomas Landelius, Joel Oskarsson, and Fredrik Lindsten. Continuous ensemble weather forecasting with diffusion models. In *International Conference on Learning Representations*, 2025.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al.

Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Lisa Bengtsson, Ulf Andrae, Trygve Aspelien, Yurii Batrak, Javier Calvo, Wim de Rooy, Emily Gleeson, Bent Hansen-Sass, Mariken Homleid, Mariano Hortal, Karl-Ivar Ivarsson, Geert Lenderink, Sami Niemelä, Kristian Pagh Nielsen, Jeanette Onvlee, Laura Rontu, Patrick Samuelsson, Daniel Santos Muñoz, Alvaro Subias, Sander Tijm, Velle Toll, Xiaohua Yang, and Morten Ødegaard Køltzow. The harmonie–arome model configuration in the aladin–hirlam nwp system. *Monthly Weather Review*, 145(5):1919 – 1935, 2017. doi: 10.1175/MWR-D-16-0417.1. URL https://journals.ametsoc.org/view/journals/mwre/145/5/mwr-d-16-0417.1.xml.

Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Accurate medium-range global weather forecasting with 3d neural networks. *Nature*, 619(7970):533–538, 2023.

Boris Bonev, Thorsten Kurth, Ankur Mahesh, Mauro Bisson, Jean Kossaifi, Karthik Kashinath, Anima Anandkumar, William D. Collins, Michael S. Pritchard, and Alexander Keller. FourCastNet 3: A geometric approach to probabilistic machine-learning weather forecasting at scale. *arXiv preprint arXiv:2507.12144*, (arXiv:2507.12144), 2025. doi: 10.48550/arXiv.2507.12144. URL http://arxiv.org/abs/2507.12144.

Salva Rühling Cachay, Duncan Watson-Parris, and Rose Yu. U-cast: A surprisingly simple and efficient frontier probabilistic AI weather forecaster. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=XF0wkyEbuM.

Mingjian Chen, Xu Tan, Bohan Li, Yanqing Liu, Tao Qin, Sheng Zhao, and Tie-Yan Liu. Adaspeech: Adaptive text to speech for custom voice. *arXiv preprint arXiv:2103.00993*, 2021. URL https://arxiv.org/abs/2103.00993.

Guillaume Couairon, Renu Singh, Anastase Charantonis, Christian Lessig, and Claire Monteleoni. Archesweathergen: Skillful and compute-efficient probabilistic weather forecasting with machine learning. *Science Advances*, 12(17):eadx2372, 2026. doi: 10.1126/sciadv.adx2372. URL https://www.science.org/doi/abs/10.1126/sciadv.adx2372.

ECMWF. Integrated forecasting system. https://www.ecmwf.int/en/forecasts/documentation-and-support/changes-ecmwf-model, 2024. Accessed: 2026-05-07.

C. A. T. Ferro. Fair scores for ensemble forecasts. *Quarterly Journal of the Royal Meteorological Society*, 140(683):1917–1923, 2014. doi: https://doi.org/10.1002/qj.2270. URL https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.2270.

Vincent Fortin, Mabrouk Abaza, Francois Anctil, and Raphael Turcotte. Why should ensemble spread match the rmse of the ensemble mean? *Journal of Hydrometeorology*, 15(4):1708–1713, 2014.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, pp. 359–378, 2007.

Yuan Hu, Lei Chen, Zhibin Wang, and Hao Li. Swinvrnn: A data-driven ensemble forecasting model via learned distribution perturbation. *Journal of Advances in Modeling Earth Systems*, 15, 02 2023. doi: 10.1029/2022MS003211.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 26565–26577, 2022.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, et al. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023.

Simon Lang, Mihai Alexe, Matthew Chantry, Jesper Dramsch, Florian Pinault, Baudouin Raoult, Mariana C. A. Clare, Christian Lessig, Michael Maier-Gerber, Linus Magnusson, Zied Ben Bouallègue, Ana Prieto Nemesio, Peter D. Dueben, Andrew Brown, Florian Pappenberger, and Florence Rabier. AIFS – ECMWF's data-driven forecasting system. *arXiv preprint arXiv:2406.01465*, 2024. URL https://arxiv.org/abs/2406.01465.

Simon Lang, Martin Leutbecher, and Pedro Maciel. A multi-scale loss formulation for learning a probabilistic model with proper score optimisation. *arXiv preprint arXiv:2506.10868*, (arXiv:2506.10868), 2025. doi: 10.48550/arXiv.2506.10868. URL http://arxiv.org/abs/2506.10868.

Simon Lang, Mihai Alexe, Mariana CA Clare, Christopher Roberts, Rilwan Adewoyin, Zied Ben Bouallègue, Matthew Chantry, Jesper Dramsch, Peter D Dueben, Sara Hahner, et al. AIFS-CRPS: ensemble forecasting using a model trained with a loss function based on the continuous ranked probability score. *npj Artificial Intelligence*, 2(1):18, 2026.

Erik Larsson, Joel Oskarsson, Tomas Landelius, and Fredrik Lindsten. Diffusion-LAM: Probabilistic limited area weather forecasting with diffusion. In *ICLR 2025 Workshop on Tackling Climate Change with Machine Learning*, 2025. URL https://www.climatechange.ai/papers/iclr2025/36.

Erik Larsson, Ramon Fuentes-Franco, Mikhail Ivanov, and Fredrik Lindsten. Climate downscaling with stochastic interpolants (cdsi). *arXiv preprint arXiv:2603.03838*, 2026. URL https://arxiv.org/abs/2603.03838.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Morteza Mardani, Noah Brenowitz, Yair Cohen, Jaideep Pathak, Chieh-Yu Chen, Cheng-Chin Liu, Arash Vahdat, Mohammad Amin Nabian, Tao Ge, Akshay Subramaniam, Karthik Kashinath, Jan Kautz, and Mike Pritchard. Residual corrective diffusion modeling for km-scale atmospheric downscaling. *Commun Earth Environ*, 6:124, 2025. doi: 10.1038/s43247-025-02042-5.

Malte Müller, Mariken Homleid, Karl-Ivar Ivarsson, Morten A. Ø Køltzow, Magnus Lindskog, Knut Helge Midtbø, Ulf Andrae, Trygve Aspelien, Lars Berggren, Dag Bjørge, Per Dahlgren, Jørn Kristiansen, Roger Randriamampianina, Martin Ridal, and Ole Vignes. AROME-MetCoOp: A nordic convective-scale operational weather prediction model. *Weather and Forecasting*, 2017.

Thomas Nils Nipen, Håvard Homleid Haugen, Magnus Sikora Ingstad, Even Marius Nordhagen, Aram Farhad Shafiq Salihi, Paulina Tedesco, Ivar Ambjørn Seierstad, Jørn Kristiansen, Simon Lang, Mihai Alexe, et al. Regional data-driven weather modeling with a global stretched grid. *Artificial Intelligence for the Earth Systems*, 5(2):250001, 2026.

Even Marius Nordhagen, Håvard Homleid Haugen, Aram Farhad Shafiq Salihi, Magnus Sikora Ingstad, Thomas Nils Nipen, Ivar Ambjørn Seierstad, Inger-Lise Frogner, Mariana Clare, Simon Lang, Matthew Chantry, Peter Dueben, and Jørn Kristiansen. High-resolution probabilistic data-driven weather modeling with a stretched-grid. *arXiv preprint arXiv:2511.23043*, 2025. URL https://arxiv.org/abs/2511.23043.

Joel Oskarsson, Tomas Landelius, and Fredrik Lindsten. Graph-based neural weather prediction for limited area modeling. In *NeurIPS 2023 Workshop on Tackling Climate Change with Machine Learning*, 2023.

Joel Oskarsson, Tomas Landelius, Marc Peter Deisenroth, and Fredrik Lindsten. Probabilistic weather forecasting with hierarchical graph neural networks. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Lorenzo Pacchiardi, Rilwan A. Adewoyin, Peter Dueben, and Ritabrata Dutta. Probabilistic forecasting with generative networks via scoring rule minimization. *Journal of Machine Learning Research*, 25(45):1–64, 2024. ISSN 1533-7928. URL http://jmlr.org/papers/v25/23-0038.html.

Jaideep Pathak, Yair Cohen, Piyush Garg, Peter Harrington, Noah Brenowitz, Dale Durran, Morteza Mardani, Arash Vahdat, Shaoming Xu, Karthik Kashinath, and Michael Pritchard. Kilometer-scale convection-allowing model emulation using generative diffusion modeling. *Science Advances*, 12(5):eadv0423, 2026. doi: 10.1126/sciadv.adv0423. URL https://www.science.org/doi/abs/10.1126/sciadv.adv0423.

Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, et al. Probabilistic weather forecasting with machine learning. *Nature*, 637(8044):84–90, 2025.

Stephan Rasp, Stephan Hoyer, Alexander Merose, Ian Langmore, Peter Battaglia, Tyler Russell, Alvaro Sanchez-Gonzalez, Vivian Yang, Rob Carver, Shreya Agrawal, Matthew Chantry, Zied Ben Bouallegue, Peter Dueben, Carla Bromberg, Jared Sisk, Luke Barrington, Aaron Bell, and Fei Sha. Weatherbench 2: A benchmark for the next generation of data-driven global weather models. *Journal of Advances in Modeling Earth Systems*, 16(6):e2023MS004019, 2024. doi: https://doi.org/10.1029/2023MS004019. URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2023MS004019.

Martin Ridal, Eric Bazile, Patrick Le Moigne, Roger Randriamampianina, Semjon Schimanke, Ulf Andrae, Lars Berggren, Pierre Brousseau, Per Dahlgren, Lisette Edvinsson, Adam El-Said, Michael Glinton, Susanna Hagelin, Susanna Hopsch, Ludvig Isaksson, Paulo Medeiros, Esbjörn Olsson, Per Unden, and Zheng Qi Wang. CERRA, the copernicus european regional reanalysis system. *Quarterly Journal of the Royal Meteorological Society*, 150(763):3385–3411, 2024. doi: https://doi.org/10.1002/qj.4764. URL https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.4764.

Jimeng Shi, Bowen Jin, Jiawei Han, Sundararaman Gopalakrishnan, and Giri Narasimhan. CoDiCast: Conditional diffusion model for global weather prediction with uncertainty quantification. *arXiv preprint arXiv:2409.05975*, 2024.

Shoaib Ahmed Siddiqui, Jean Kossaifi, Boris Bonev, Christopher Choy, Jan Kautz, David Krueger, and Kamyar Azizzadenesheli. Exploring the design space of deep-learning-based weather forecasting systems. *arXiv preprint arXiv:2410.07472*, 2024. URL https://arxiv.org/abs/2410.07472.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Prakhar Srivastava, Ruihan Yang, Gavin Kerrigan, Gideon Dresdner, Jeremy McGibbon, Christopher Bretherton, and Stephan Mandt. Precipitation downscaling with spatiotemporal video diffusion. *Advances in Neural Information Processing Systems*, 37:56374–56400, 2024.

Jason Stock, Troy Arcomano, and Rao Kotamarthi. Swift: An autoregressive consistency model for efficient weather forecasting. *Machine Learning: Earth*, 2(2):025004, 2026. doi: 10.1088/3049-4753/ae802a.

Jasper S Wijnands, Michiel Van Ginderachter, Bastien François, Sophie Buurman, Piet Termonia, and Dieter Van den Bleeken. A comparison of stretched-grid and limited-area modelling for data-driven regional weather forecasting. *arXiv preprint arXiv:2507.18378*, 2025.

Pengbo Xu, Xiaogu Zheng, Tianyan Gao, Yu Wang, Junping Yin, Juan Zhang, Xuanze Zhang, San Luo, Zhonglei Wang, Zhimin Zhang, et al. An artificial intelligence-based limited area model for forecasting of surface meteorological variables. *Communications Earth & Environment*, 6(1):372, 2025.

X. Yang, C. Peralta, B. Amstrup, K. S. Hintz, S. B. Thorsen, L. Denby, S. Kamuk Christiansen, H. Schulz, S. Pelt, and M. Schreiner. DANRA: the kilometer-scale danish regional atmospheric reanalysis. *Earth System Science Data*, 18(3):2251–2264, 2026. doi: 10.5194/essd-18-2251-2026. URL https://essd.copernicus.org/articles/18/2251/2026/.

Mickaël Zamo and Philippe Naveau. Estimation of the continuous ranked probability score with limited information and applications to ensemble weather forecasts. *Mathematical Geosciences*, 50(2):209–234, 2018.

A CRPS AS A TRAINING OBJECTIVE

Scoring rules provide a principled framework for evaluating probabilistic forecasts. Given a predictive distribution p and an observation $x \sim q$, a scoring rule $s(p, x)$ measures the agreement between p and the observation (Gneiting & Raftery, 2007). Defining $s(p, q) = \mathbb{E}_q[s(p, x)]$, a scoring rule is *proper* if $s(q, q) \leq s(p, q)$ for all p and q, and *strictly proper* if equality holds only when $p = q$. Thus, strictly proper scoring rules are minimized uniquely by the true data-generating distribution, making them well-suited as training objectives for probabilistic models.

A widely used strictly proper scoring rule is the Continuous Ranked Probability Score (Gneiting & Raftery, 2007), which measures the discrepancy between the predictive Cumulative Distribution Function (CDF) F and the empirical CDF of the observation. For a univariate distribution, the CRPS is defined as

$$\text{CRPS}(F, x) = \int_{\mathbb{R}} \big(F(\hat{x}) - \mathbb{1}(\hat{x} \geq x)\big)^2 \, \mathrm{d}\hat{x} = \mathbb{E}_p\big[|\hat{x} - x|\big] - \frac{1}{2}\mathbb{E}_{p,p}\big[|\hat{x} - \hat{x}'|\big], \tag{3}$$

where $\hat{x}, \hat{x}' \sim p$ are independent samples. The second expression is particularly useful for empirical estimation.

Given an ensemble forecast $\{\hat{x}_n\}_{n=1}^N \sim p$, the *fair* CRPS estimator

$$\text{CRPS}_{\text{fair}} = \frac{1}{N}\sum_{n=1}^{N}|\hat{x}_n - x| - \frac{1}{2N(N-1)}\sum_{n=1}^{N}\sum_{n^*=1}^{N}|\hat{x}_n - \hat{x}_{n^*}| \tag{4}$$

provides an unbiased estimate with respect to the ensemble size (Ferro, 2014). However, this estimator can lead to unstable training dynamics (Lang et al., 2026; Bonev et al., 2025). To mitigate this, we adopt the almost-fair variant (Lang et al., 2026),

$$\text{CRPS}_{\alpha\text{-fair}} = \alpha\text{CRPS}_{\text{fair}} + (1 - \alpha)\text{CRPS}, \tag{5}$$

which interpolates between the unbiased and standard estimators.

B IMPLEMENTATION DETAILS

Both Graph-EFM and CRPS-LAM are trained with multiple ensemble members for the CRPS loss. We train all models on H200 GPUs to speed up training, and for the ensemble models we place both ensemble members on the same GPU. The code and implementation details will be made publicly available upon acceptance of this paper.

B.1 COMPUTE RESOURCES

Table 1 lists the parameter count, peak GPU memory, training cost and inference time of each model on the DANRA dataset. Training cost is given in GPU-hours on H200 GPUs and covers all training stages, including the IFS finetuning. Memory and inference time were measured on A100 GPUs. Inference time is for a 72 h forecast (24 autoregressive steps). For the ensemble models we generate 20 ensemble members and report both the time per member and the total time for the ensemble.

The hybrid CNN/GNN backbone is more efficient in terms of memory and compute than a fully graph-based model. We did not use these savings to reduce inference time. Instead, we used them to scale DET-LAM and CRPS-LAM to roughly ten times as many parameters as Graph-FM and Graph-EFM, aiming for the best forecast performance while keeping memory use and inference time comparable to the graph-based models.

CRPS-LAM samples each ensemble member in a single forward pass per time step, so a 20-member ensemble costs about the same as 20 deterministic forecasts. Diffusion-LAM needs many solver steps per time step and is about $39\times$ slower than CRPS-LAM, the same factor as on the MEPS dataset (section F). The training cost of Diffusion-LAM is lower than that of the other models, but mainly because it is trained only for single-step prediction and so does not go through the autoregressive training stages.

Because training at full scale is expensive, we did only limited hyperparameter tuning, using small-scale preliminary experiments with reduced model and data sizes, and otherwise relied largely on

Table 1: Compute resources for each model on the DANRA dataset. Training GPU-hours are on H200 GPUs and include all training stages; peak memory and inference time are measured on A100 GPUs. Inference time is for a 72 h forecast. For the ensemble models it is the time per member when generating a 20-member ensemble, with the total time for the ensemble in brackets. The Diffusion-LAM training cost is for a single training run.

		Peak memory (GB)			
Model	Parameters (M)	Train	Inference	Train GPU-hours	Inference time (s)
Graph-FM	23.4	64	48	3556	42
DET-LAM	233	78	40	3714	30
Graph-EFM	14.5	64	48	6781	26 (527)
Diffusion-LAM	277	79	40	3022	936 (18993)
CRPS-LAM	251	79	40	5784	24 (487)

hyperparameters from previous work (Adamov et al., 2025; Oskarsson et al., 2024). DET-LAM, CRPS-LAM, Graph-FM and Graph-EFM were each trained once at full scale. Diffusion-LAM was the only model trained several times, with different hyperparameters, in an attempt to make it work on the DANRA data. None of these runs gave better results. Memory use and compute per run were the same for all of them, so table 1 reports the cost of a single run.

B.2 Models

For all machine learning models we use the same training budget as Adamov et al. (2025): 80 epochs of single-step training followed by 3 epochs with four autoregressive steps, all with ERA5 boundaries. We then finetune on IFS boundaries for 7 epochs with four autoregressive steps and 3 epochs with 12 autoregressive steps. We use the AdamW optimizer (Loshchilov & Hutter, 2019) with $\beta_1 = 0.9, \beta_2 = 0.95$ and weight decay 0.01. We required every model to be trainable within the 80 GB memory of an A100 GPU.

Numerical Model. To compare our models with an operational numerical baseline, we use forecasts produced by the HARMONIE-AROME NWP forecasting system (Bengtsson et al., 2017), run by the Danish Meteorological Institute and initialized from operational analysis. Unfortunately, we only have access to a very limited set of variables and relatively short lead times of up to 18 h.

Graph-FM. We use the exact model setup for the best performing model in Adamov et al. (2025) with a four-level hierarchical graph. The hidden dimension encoder/decoder was set to 150, and the hidden dimension for the processor was set to 300. This model uses a modified graph structure for the boundary, applying a 800 km buffer around the domain rather than extending 800 km in each direction to construct a regular grid.

Graph-EFM. For the Graph-EFM model, we use a three-level hierarchical graph, after initial experimentation with four hierarchical levels showed that it did not perform well. The encoder/decoder hidden dimension was set to 150, and the processor's hidden dimension was set to 300. We follow a training curriculum similar to Oskarsson et al. (2024) shown in table 2.

Epochs	KL β	CRPS Weight	Learning Rate	AR Steps	Boundary
35	0	0	0.001	1	ERA5
45	1	0	0.001	1	ERA5
3	1	0.00001	0.0005	4	ERA5
7	1	0.00001	0.0005	4	IFS
3	1	0.00001	0.0005	12	IFS

Table 2: Training Curriculum for Graph-EFM

DET-LAM (ours). We set the hidden dimension of the encoder, processor, and decoder to 300. We train the model with a MSE objective

$$\mathcal{L} = \frac{1}{T|G|} \sum_{t=0}^{T-1} \sum_{g \in G} \sum_{d \in D} \omega_d \sigma_d (\hat{X}_{t+1,g,d} - X_{t+1,g,d})^2, \tag{6}$$

where

- T is the number of autoregressive forecast steps during training.
- $G \subset \Omega^I$ is the set of interior grid points.
- D is the set of variables that we forecast.
- σ_d is the inverse variance of the time differences for variable d, following Lam et al. (2023). This weighting accounts for variables varying at different temporal scales.
- ω_d is a manually chosen variable-specific weighting. We follow Adamov et al. (2025) and use the same weighting scheme.
- $\hat{X}_{t+1,g,d}$ is the prediction at gridpoint g, for variable d at time step t with the corresponding ground truth $X_{t+1,g,d}$.

For the first 80 epochs we use a cosine learning rate scheduler with initial learning rate 10^{-4} and end learning rate 10^{-5}. For the autoregressive training we use a learning rate of 10^{-5}.

CRPS-LAM (ours). Training with two ensemble members introduces additional memory overhead. To accommodate this within a single H200 GPU without gradient checkpointing, we reduce the hidden dimension of the encoder and decoder to 270, while increasing the processor hidden dimension to 330. This adjustment allows us to fit the full model in memory while simplifying the training setup.

For the first 80 epochs we use a cosine learning rate scheduler with initial learning rate 10^{-4} and end learning rate 10^{-5}. For the autoregressive training we use a learning rate of 10^{-5}. We use $N = 2$ ensemble members, and $\alpha = 0.95$ when training CRPS-LAM.

C METRICS

Given an ensemble forecast $\{\hat{X}_{s,n}\}_{n=1}^N$ with N members for each forecast s in the test set S, we define the RMSE of variable $d \in D$ at lead time t as

$$\text{RMSE}_{t,d} = \sqrt{\frac{1}{|S||G|} \sum_{s \in S} \sum_{g \in G} \left(\bar{\hat{X}}_{s,t,g,d} - X_{s,t,g,d}\right)^2}, \qquad \bar{\hat{X}}_{s,t,g,d} = \frac{1}{N} \sum_{n=1}^{N} \hat{X}_{s,t,g,d,n},$$

where $G \subset \Omega^I$ are the interior grid points and $\hat{X}_{s,t,g,d,n}$ is the prediction of ensemble member n. For deterministic models the prediction replaces the ensemble mean. Following the standard convention of WeatherBench 2 (Rasp et al., 2024), the square root is applied after averaging over forecasts.

To measure the calibration of the forecast uncertainty we use the bias-corrected SSR for variable d at lead time t,

$$\text{SSR}_{t,d} = \sqrt{\frac{N+1}{N}} \frac{\text{Spread}_{t,d}}{\text{RMSE}_{t,d}}, \qquad \text{Spread}_{t,d} = \sqrt{\frac{1}{|S||G|} \sum_{s \in S} \sum_{g \in G} \frac{1}{N-1} \sum_{n=1}^{N} \left(\hat{X}_{s,t,g,d,n} - \bar{\hat{X}}_{s,t,g,d}\right)^2}.$$

For a well-calibrated ensemble, $\text{SSR}_{t,d} \approx 1$ (Fortin et al., 2014).

We also compute the CRPS (Gneiting & Raftery, 2007) for variable d at lead time t as

$$\text{CRPS}_{t,d} = \frac{1}{|S||G|} \sum_{s \in S} \sum_{g \in G} \Big(\frac{1}{N} \sum_{n=1}^{N} |\hat{X}_{s,t,g,d,n} - X_{s,t,g,d}| - \frac{1}{2N(N-1)} \sum_{n=1}^{N} \sum_{n^*=1}^{N} |\hat{X}_{s,t,g,d,n} - \hat{X}_{s,t,g,d,n^*}| \Big).$$

Note, we follow the convention of Oskarsson et al. (2024) and compute the CRPS as a finite sample estimate (Zamo & Naveau, 2018) over all ensemble members without accounting for any covariance structure.

Table 3: Overview of data products used for DANRA experiments.

Dataset	Type	Use	Resolution	Time step	Length
DANRA	Reanalysis	Training	2.5 km	3 h	21 years
ERA5	Reanalysis	Boundary	0.25°	6 h	21 years
IFS HRES	Forecasts		0.25°	6 h	2 years
DMI SYNOP	Observations	Evaluation	50 Stations	1 h	1 year
DANRA Forecasts	Forecast		2.5 km	3 h	1 year

When calculating metrics for each individual variable separately, we first unnormalize the predictions before comparing them to the ground truth. However, when evaluating the mean performance across all variables, we compute the metrics using normalized data and forecasts. In this case, we normalize the ground truth and compare it to the normalized predictions. The mean normalized score is then obtained by averaging the metric values (RMSE, CRPS, SSR) across all variables.

To evaluate how well the predicted fields preserve the spatial spectral characteristics of the target over different lead times, we compute the LSD, defined as

$$\mathrm{LSD}_{t,d} = \sqrt{\frac{1}{\Lambda}\sum_{\lambda=1}^{\Lambda}\left(\log_{10} S(\hat{X}_{t,d})_\lambda - \log_{10} S(X_{t,d})_\lambda\right)^2}. \tag{7}$$

Here, $S(X)_\lambda$ denotes the power spectrum of field X at wavenumber λ, where $\lambda = 1, \ldots, \Lambda$, and Λ is the total number of spectral components considered. Lower LSD values indicate better agreement between the predicted and target spectra, and therefore better preservation of spatial structure across scales.

D DANRA Dataset details

A list of the data used in our experiments over the DANRA[1] domain is given in table 3. All variables used as model input and forecasted in the DANRA experiments are listed in table 4. Following Adamov et al. (2025), we split the data into three splits, training (2000-01-01 to 2018-10-29), validation (2018-11-05 to 2019-10-22) and test (2019-10-29 to 2020-10-29). The locations of the in-situ observations in Denmark are shown in fig. 9.

E Additional Experiments on the DANRA dataset

Here, we provide a more detailed evaluation of the models on the DANRA dataset. Scorecards comparing DET-LAM to Graph-FM and CRPS-LAM to Graph-EFM for all variables in both settings are shown in figs. 10 to 14, and the corresponding values at 24, 48 and 72 h are given in tables 5 to 10. The RMSE, CRPS, SSR, and LSD over all lead times for the surface variables and the variables at 600 hPa with ERA5 boundaries are shown in figs. 15 to 18. Figure 19 shows the models trained only with ERA5 boundaries when evaluated with IFS boundaries. The drop in skill and calibration motivated finetuning with IFS boundaries, and the results after finetuning are shown in figs. 20 and 21. Our attempt to train Diffusion-LAM on DANRA is described in section E.1, and section E.2 shows 72 h forecasts for a subset of the variables.

[1]The DANRA dataset (`https://dmidk.github.io/danradocs/`) is openly available under a CC BY 4.0 license.

Table 4: Variables used in the DANRA experimental setup. The ERA5/IFS column describes the boundary forcing. Adapted from Adamov et al. (2025).

Variable	**Unit**	**Abbrev.**	DANRA	ERA5 / IFS
Surface variables				
10 m u-component of wind	m/s	u10m	✓	✓
10 m v-component of wind	m/s	v10m	✓	✓
2 m temperature	K	t2m	✓	✓
Mean sea level pressure	Pa	msl	✓	✓
Surface pressure	Pa	sp	✓	✓
Net short wave radiation flux	W/m^2	swavr0m	✓	
Net long wave radiation flux	W/m^2	lwavr0m	✓	
Variables on vertical levels				
Geopotential	m^2/s^2	z	✓	✓
Temperature	K	t	✓	✓
u-component of wind	m/s	u	✓	✓
v-component of wind	m/s	v	✓	✓
Vertical velocity	Pa/s	w	✓	✓
Relative humidity	-	r	✓	
Specific humidity	kg/kg	q		✓
Forcing				
Top of atmosphere radiation flux	W/m^2	toarf	✓	✓
sin-encoded time of day	-	sin_d	✓	✓
cos-encoded time of day	-	cos_d	✓	✓
sin-encoded day of year	-	sin_y	✓	✓
cos-encoded day of year	-	cos_y	✓	✓
Static fields				
Land-sea mask	-	lsm	✓	✓
Surface geopotential (orography)	m^2/s^2	oro	✓	✓

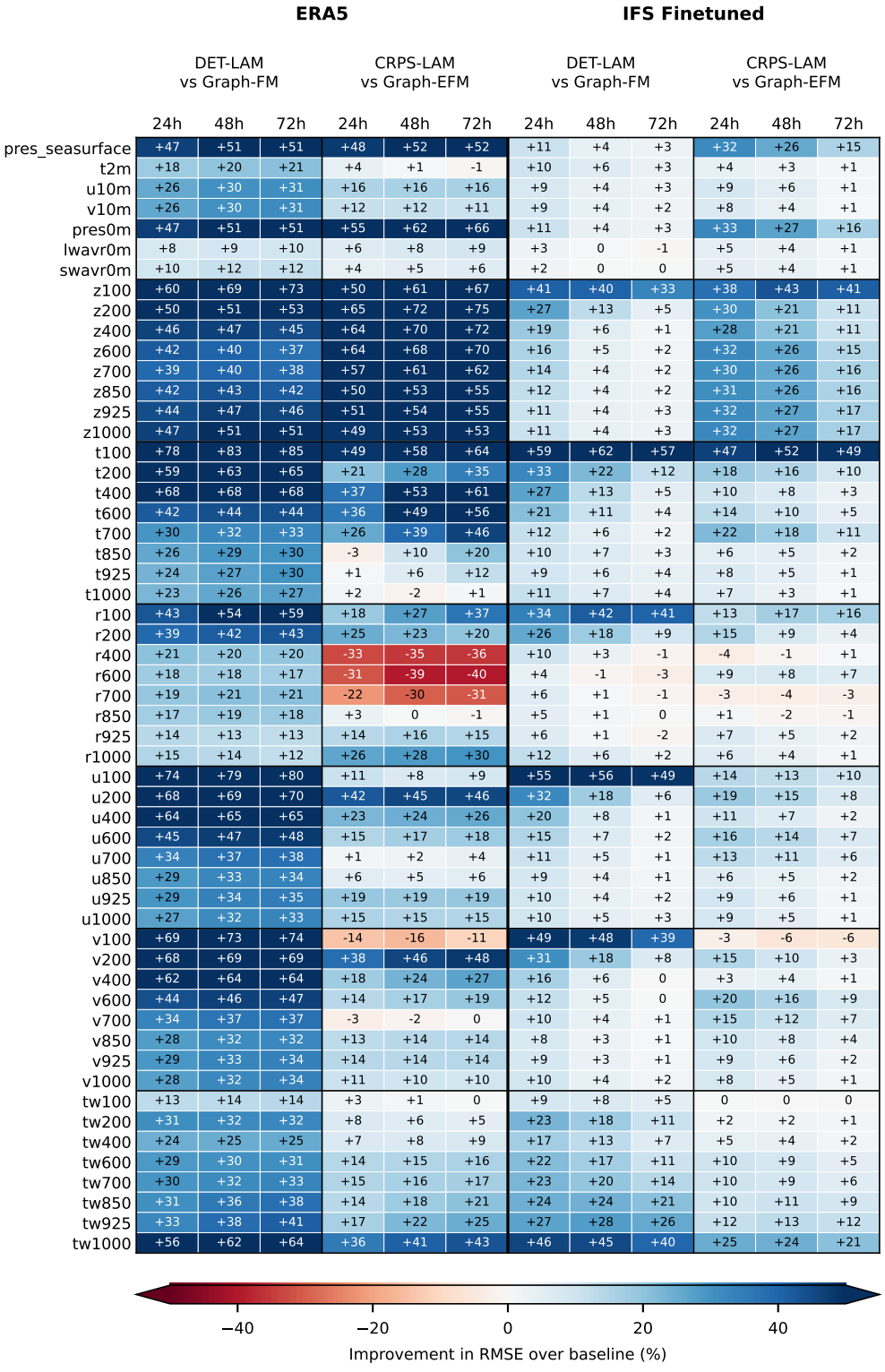


Figure 10: RMSE scorecard. Each cell shows the relative improvement (%) of DET-LAM over Graph-FM (deterministic) and of CRPS-LAM over Graph-EFM (probabilistic, RMSE of the ensemble mean), per variable, setting and lead time. Positive values (blue) mean the model is better than its baseline.



Figure 11: CRPS scorecard: relative improvement (%) of CRPS-LAM over Graph-EFM. Positive values (blue) mean CRPS-LAM is better.

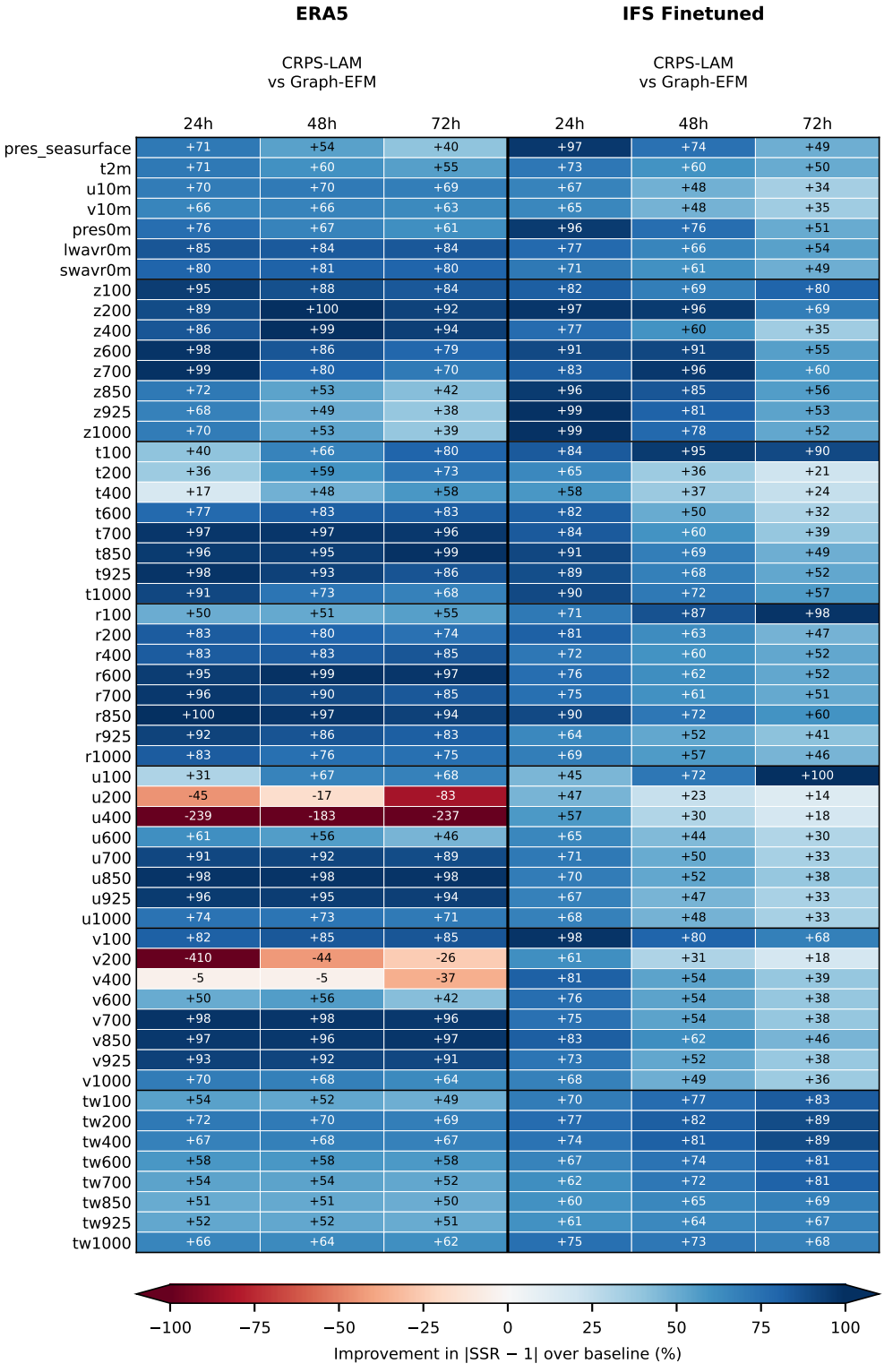


Figure 12: SSR scorecard: relative reduction (%) of $|\mathrm{SSR} - 1|$ for CRPS-LAM compared to Graph-EFM. Positive values (blue) mean CRPS-LAM is closer to a perfect spread-skill ratio of 1.

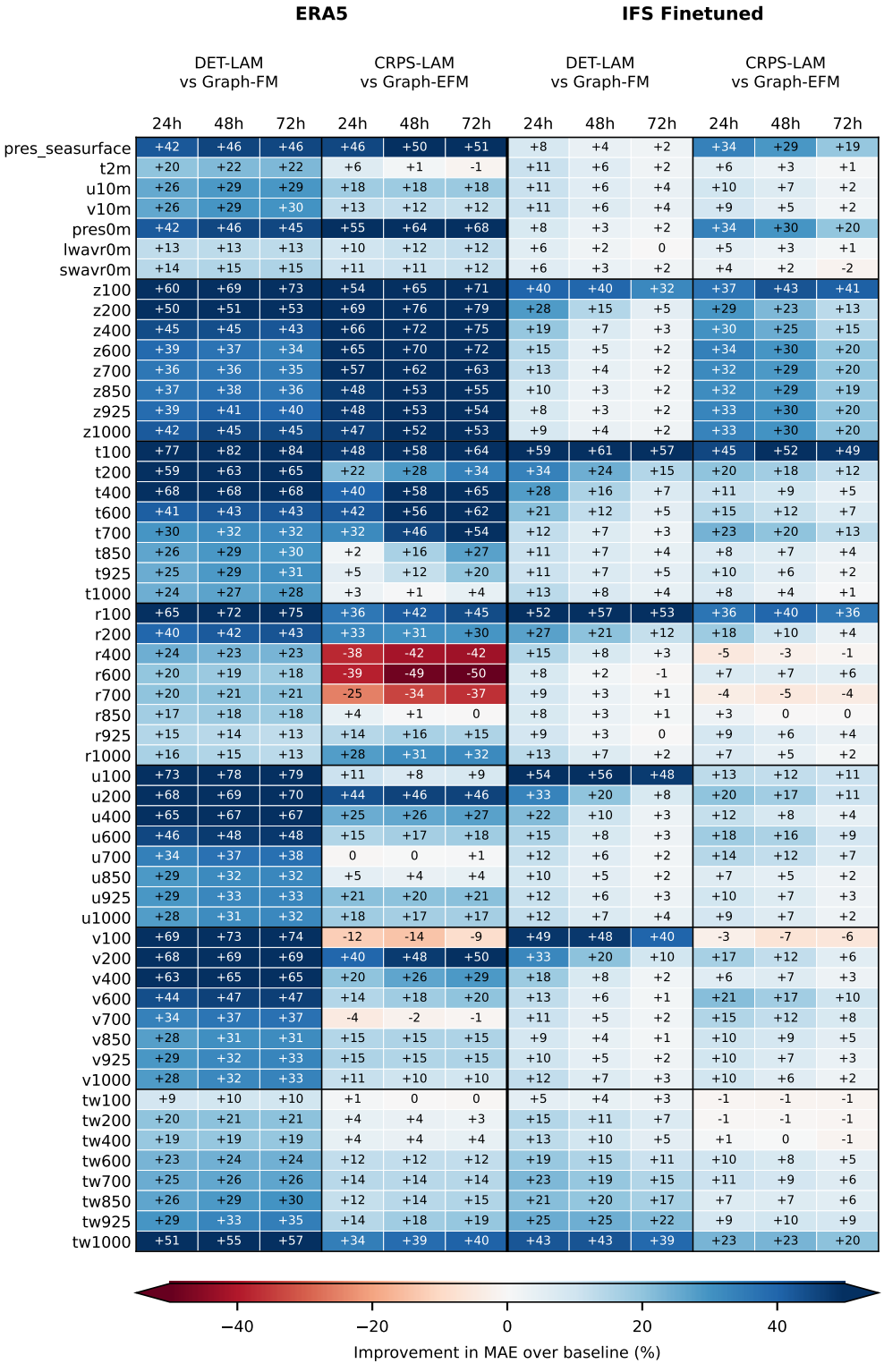


Figure 13: MAE scorecard. Each cell shows the relative improvement (%) of DET-LAM over Graph-FM (deterministic) and of CRPS-LAM over Graph-EFM (probabilistic, RMSE of the ensemble mean), per variable, setting and lead time. Positive values (blue) mean the model is better than its baseline.

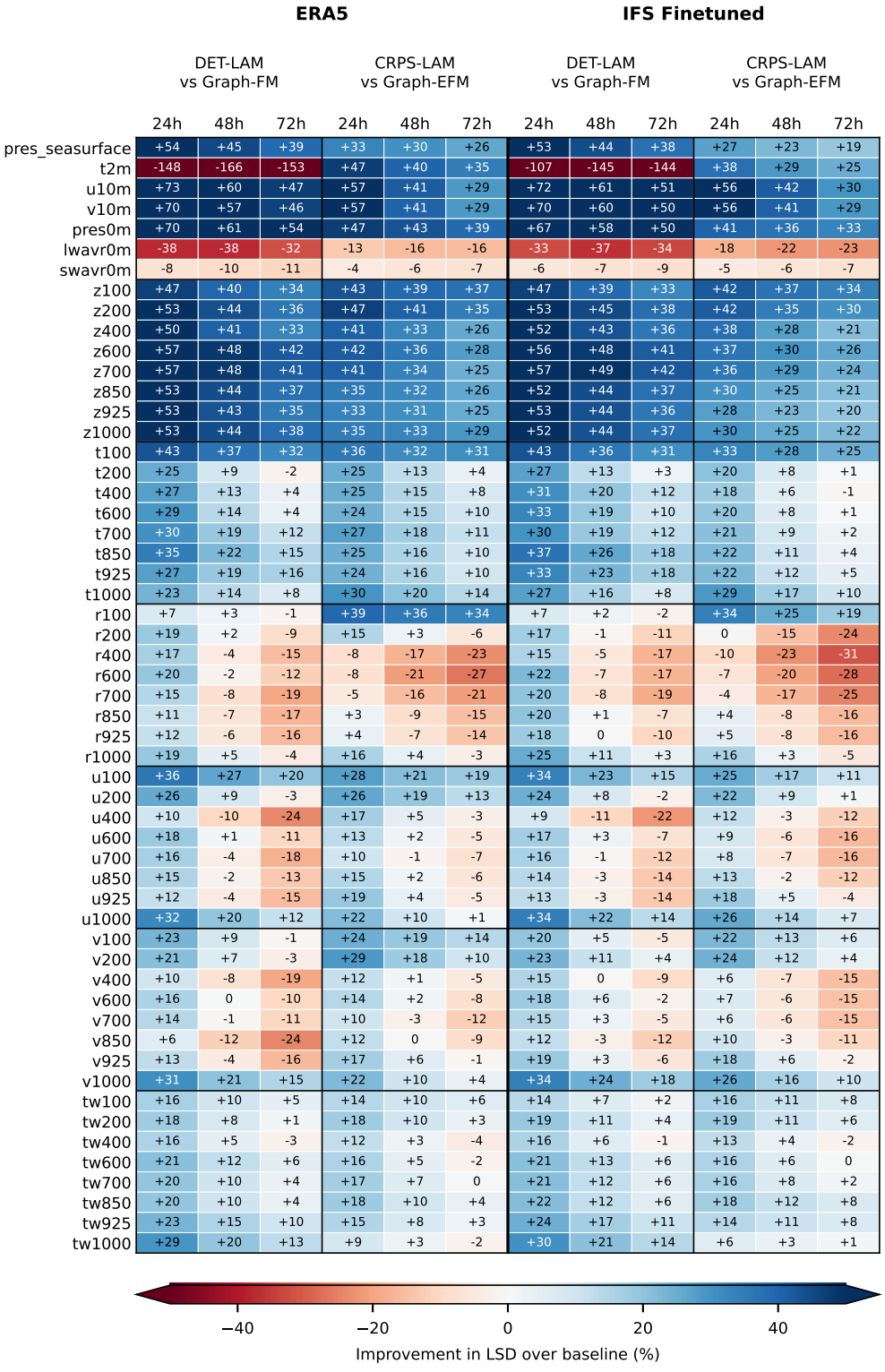


Figure 14: LSD scorecard. Each cell shows the relative improvement (%) of DET-LAM over Graph-FM (deterministic) and of CRPS-LAM over Graph-EFM (probabilistic, RMSE of the ensemble mean), per variable, setting and lead time. Positive values (blue) mean the model is better than its baseline.

Table 5: CRPS for all variables at different lead times. Mean over 3 inference seeds for CRPS-LAM (ERA5), CRPS-LAM (IFS Finetuned), Graph-EFM (ERA5), Graph-EFM (IFS Finetuned). The best deterministic and the best probabilistic model are in bold, per setting.

	ERA5						IFS Finetuned					
	Graph-EFM			CRPS-LAM			Graph-EFM			CRPS-LAM		
Variable	24 h	48 h	72 h	24 h	48 h	72 h	24 h	48 h	72 h	24 h	48 h	72 h
pres_seasurface [Pa]	30.1	39.7	45.1	**15.9**	**19.1**	**21.3**	68.1	120	178	**41.4**	**79.9**	**140**
t2m [K]	0.447	0.508	0.532	**0.402**	**0.478**	**0.513**	0.521	0.639	0.749	**0.469**	**0.584**	**0.699**
u10m [m/s]	0.576	0.601	0.609	**0.458**	**0.473**	**0.476**	0.754	0.955	1.22	**0.644**	**0.839**	**1.12**
v10m [m/s]	0.518	0.533	0.538	**0.441**	**0.454**	**0.459**	0.735	0.928	1.18	**0.636**	**0.832**	**1.11**
pres0m [Pa]	35.9	55.0	71.6	**15.5**	**18.9**	**21.3**	66.6	119	176	**40.4**	**78.2**	**137**
lwavr0m [W/m^2]	10.6	11.5	11.8	**9.14**	**9.60**	**9.70**	12.1	13.6	15.2	**10.6**	**12.2**	**14.0**
swavr0m [W/m^2]	28.3	29.6	29.8	**24.2**	**25.1**	**25.1**	25.7	28.7	31.7	**22.8**	**25.8**	**29.7**
z100 [m^2/s^2]	119	223	312	**48.7**	**66.5**	**75.1**	92.2	149	200	**54.1**	**77.1**	**108**
z200 [m^2/s^2]	127	207	264	**33.8**	**41.4**	**46.3**	81.9	131	200	**54.6**	**97.7**	**172**
z400 [m^2/s^2]	79.5	120	147	**23.7**	**28.7**	**32.0**	81.9	137	212	**54.3**	**101**	**181**
z600 [m^2/s^2]	48.5	71.1	84.7	**15.4**	**19.2**	**21.6**	60.0	103	157	**37.1**	**69.4**	**125**
z700 [m^2/s^2]	35.5	50.5	58.7	**14.3**	**17.9**	**20.2**	53.1	93.2	142	**33.7**	**62.9**	**112**
z850 [m^2/s^2]	25.7	36.3	43.6	**13.0**	**16.5**	**19.0**	50.8	91.3	137	**32.2**	**61.3**	**108**
z925 [m^2/s^2]	24.2	33.8	39.8	**12.1**	**15.4**	**17.5**	52.3	94.6	141	**32.2**	**62.3**	**110**
z1000 [m^2/s^2]	24.1	32.1	36.5	**12.5**	**15.0**	**16.6**	54.2	97.3	144	**33.2**	**64.1**	**112**
t100 [K]	0.381	0.559	0.709	**0.184**	**0.212**	**0.227**	0.543	0.761	0.911	**0.270**	**0.327**	**0.415**
t200 [K]	0.335	0.385	0.432	**0.258**	**0.268**	**0.273**	0.649	0.893	1.22	**0.490**	**0.704**	**1.05**
t400 [K]	0.273	0.417	0.540	**0.157**	**0.162**	**0.167**	0.475	0.666	0.922	**0.401**	**0.578**	**0.852**
t600 [K]	0.438	0.612	0.744	**0.221**	**0.230**	**0.233**	0.486	0.650	0.870	**0.388**	**0.538**	**0.769**
t700 [K]	0.422	0.591	0.717	**0.263**	**0.280**	**0.285**	0.577	0.734	0.924	**0.403**	**0.536**	**0.748**
t850 [K]	0.331	0.425	0.509	**0.305**	**0.323**	**0.331**	0.509	0.647	0.813	**0.432**	**0.548**	**0.718**
t925 [K]	0.339	0.423	0.503	**0.300**	**0.339**	**0.362**	0.504	0.640	0.785	**0.419**	**0.549**	**0.703**
t1000 [K]	0.347	0.408	0.460	**0.326**	**0.386**	**0.413**	0.477	0.590	0.710	**0.412**	**0.527**	**0.653**
r100 [%]	0.00366	0.00482	0.00575	**0.00208**	**0.00246**	**0.00271**	0.00426	0.00543	0.00620	**0.00243**	**0.00288**	**0.00344**
r200 [%]	0.0509	0.0522	0.0507	**0.0312**	**0.0329**	**0.0326**	0.0538	0.0604	0.0725	**0.0391**	**0.0487**	**0.0637**
r400 [%]	**0.0661**	**0.0658**	**0.0659**	0.0862	0.0881	0.0890	0.114	0.133	0.156	**0.112**	**0.127**	**0.146**
r600 [%]	**0.0628**	**0.0625**	**0.0630**	0.0797	0.0851	0.0868	0.126	0.151	0.174	**0.106**	**0.127**	**0.148**
r700 [%]	**0.0604**	**0.0606**	**0.0611**	0.0714	0.0768	0.0791	0.0987	0.119	0.141	**0.0955**	**0.116**	**0.136**
r850 [%]	0.0587	0.0611	0.0615	**0.0525**	**0.0564**	**0.0573**	0.0722	0.0837	0.0954	**0.0666**	**0.0789**	**0.0897**
r925 [%]	0.0478	0.0543	0.0557	**0.0380**	**0.0416**	**0.0429**	0.0533	0.0617	0.0687	**0.0468**	**0.0556**	**0.0637**
r1000 [%]	0.0431	0.0510	0.0533	**0.0286**	**0.0319**	**0.0325**	0.0382	0.0446	0.0501	**0.0340**	**0.0407**	**0.0472**
u100 [m/s]	0.812	0.893	0.913	**0.699**	**0.791**	**0.802**	0.971	1.10	1.26	**0.808**	**0.921**	**1.06**
u200 [m/s]	0.843	0.905	0.907	**0.470**	**0.478**	**0.482**	1.44	1.99	2.74	**1.13**	**1.63**	**2.45**
u400 [m/s]	0.900	0.936	0.942	**0.659**	**0.669**	**0.668**	2.06	2.68	3.61	**1.77**	**2.39**	**3.40**
u600 [m/s]	0.747	0.768	0.774	**0.623**	**0.624**	**0.624**	1.62	2.06	2.58	**1.27**	**1.64**	**2.26**
u700 [m/s]	0.729	0.742	0.746	**0.712**	**0.719**	**0.720**	1.48	1.86	2.32	**1.20**	**1.53**	**2.03**
u850 [m/s]	0.767	0.774	0.780	**0.668**	**0.682**	**0.684**	1.27	1.63	2.09	**1.12**	**1.45**	**1.92**
u925 [m/s]	0.829	0.849	0.856	**0.609**	**0.625**	**0.625**	1.17	1.53	1.99	**0.989**	**1.33**	**1.82**
u1000 [m/s]	0.675	0.695	0.701	**0.535**	**0.554**	**0.558**	0.898	1.14	1.45	**0.768**	**0.997**	**1.34**
v100 [m/s]	**0.709**	**0.812**	**0.881**	0.772	0.894	0.917	0.930	**1.06**	1.22	**0.915**	1.08	**1.22**
v200 [m/s]	0.825	0.968	1.04	**0.497**	**0.501**	**0.510**	1.52	2.02	2.77	**1.24**	**1.75**	**2.60**
v400 [m/s]	0.903	0.994	1.04	**0.673**	**0.677**	**0.684**	2.04	2.79	3.73	**1.84**	**2.48**	**3.46**
v600 [m/s]	0.731	0.771	0.790	**0.615**	**0.622**	**0.619**	1.68	2.07	2.54	**1.25**	**1.62**	**2.18**
v700 [m/s]	**0.693**	0.717	0.727	0.704	**0.711**	**0.713**	1.45	1.80	2.22	**1.17**	**1.48**	**1.94**
v850 [m/s]	0.858	0.882	0.888	**0.660**	**0.671**	**0.674**	1.29	1.66	2.10	**1.07**	**1.38**	**1.84**
v925 [m/s]	0.756	0.773	0.780	**0.602**	**0.616**	**0.618**	1.15	1.51	1.96	**0.976**	**1.31**	**1.78**
v1000 [m/s]	0.599	0.613	0.616	**0.518**	**0.535**	**0.541**	0.883	1.11	1.42	**0.755**	**0.989**	**1.32**
tw100 [m/s]	0.0241	0.0244	0.0244	**0.0232**	**0.0237**	**0.0238**	0.0256	0.0258	0.0260	**0.0245**	**0.0247**	**0.0249**
tw200 [m/s]	0.0269	0.0273	0.0274	**0.0222**	**0.0227**	**0.0229**	0.0293	0.0301	0.0309	**0.0243**	**0.0249**	**0.0256**
tw400 [m/s]	0.0391	0.0395	0.0396	**0.0314**	**0.0318**	**0.0318**	0.0421	0.0434	0.0450	**0.0342**	**0.0356**	**0.0374**
tw600 [m/s]	0.0361	0.0368	0.0370	**0.0269**	**0.0274**	**0.0275**	0.0394	0.0407	0.0422	**0.0298**	**0.0314**	**0.0336**
tw700 [m/s]	0.0361	0.0370	0.0373	**0.0264**	**0.0270**	**0.0271**	0.0382	0.0395	0.0409	**0.0290**	**0.0305**	**0.0325**
tw850 [m/s]	0.0359	0.0376	0.0385	**0.0281**	**0.0286**	**0.0288**	0.0365	0.0383	0.0399	**0.0299**	**0.0314**	**0.0332**
tw925 [m/s]	0.0316	0.0339	0.0350	**0.0242**	**0.0247**	**0.0249**	0.0315	0.0336	0.0354	**0.0254**	**0.0268**	**0.0286**
tw1000 [m/s]	0.0149	0.0172	0.0183	**0.00843**	**0.00897**	**0.00922**	0.0140	0.0160	0.0173	**0.00922**	**0.0105**	**0.0118**

Table 6: RMSE for all variables at different lead times with ERA5 boundary conditioning. The best deterministic and the best probabilistic model are in bold.

	Graph-FM			DET-LAM			Graph-EFM			CRPS-LAM		
Variable	24 h	48 h	72 h	24 h	48 h	72 h	24 h	48 h	72 h	24 h	48 h	72 h
pres_seasurface [Pa]	48.6	57.9	61.1	**25.9**	**28.3**	**29.9**	56.9	73.4	82.3	**29.7**	**35.5**	**39.3**
t2m [K]	0.850	0.992	1.06	**0.697**	**0.791**	**0.844**	0.850	0.946	**0.984**	**0.814**	**0.940**	0.993
u10m [m/s]	1.08	1.18	1.21	**0.800**	**0.826**	**0.838**	1.07	1.10	1.11	**0.899**	**0.925**	**0.932**
v10m [m/s]	1.07	1.17	1.20	**0.791**	**0.815**	**0.825**	0.995	1.02	1.03	**0.875**	**0.898**	**0.911**
pres0m [Pa]	47.2	56.3	60.1	**25.1**	**27.6**	**29.4**	64.5	92.7	115	**28.8**	**35.0**	**39.4**
lwavr0m [W/m^2]	20.3	21.2	21.5	**18.6**	**19.2**	**19.5**	21.4	22.6	23.0	**20.0**	**20.8**	**20.9**
swavr0m [W/m^2]	67.7	70.7	71.6	**61.1**	**62.5**	**63.0**	76.7	79.4	80.1	**73.2**	**75.1**	**75.4**
z100 [m^2/s^2]	162	258	328	**64.4**	**79.7**	**88.5**	186	320	428	**92.5**	**125**	**141**
z200 [m^2/s^2]	91.7	110	125	**45.8**	**54.1**	**59.4**	180	277	346	**63.6**	**77.5**	**87.0**
z400 [m^2/s^2]	80.5	90.5	93.8	**43.1**	**48.4**	**52.0**	124	179	214	**44.2**	**53.2**	**59.1**
z600 [m^2/s^2]	46.9	54.2	56.9	**27.3**	**32.4**	**35.6**	79.5	113	132	**28.6**	**35.7**	**40.3**
z700 [m^2/s^2]	40.6	47.1	49.7	**24.6**	**28.4**	**30.7**	61.9	85.7	98.4	**26.6**	**33.3**	**37.5**
z850 [m^2/s^2]	37.0	43.5	45.9	**21.6**	**24.8**	**26.8**	47.6	65.0	76.5	**24.0**	**30.3**	**34.6**
z925 [m^2/s^2]	36.5	43.1	45.5	**20.4**	**22.9**	**24.5**	45.7	61.4	71.2	**22.6**	**28.4**	**32.3**
z1000 [m^2/s^2]	39.3	46.9	49.6	**20.9**	**23.0**	**24.4**	45.7	59.1	66.4	**23.4**	**28.0**	**31.0**
t100 [K]	1.20	1.70	2.00	**0.262**	**0.289**	**0.303**	0.688	0.965	1.19	**0.352**	**0.402**	**0.434**
t200 [K]	1.09	1.23	1.32	**0.443**	**0.454**	**0.461**	0.611	0.701	0.787	**0.483**	**0.502**	**0.514**
t400 [K]	0.772	0.812	0.825	**0.247**	**0.257**	**0.264**	0.469	0.657	0.804	**0.297**	**0.306**	**0.314**
t600 [K]	0.701	0.755	0.769	**0.405**	**0.424**	**0.432**	0.645	0.843	0.989	**0.412**	**0.428**	**0.432**
t700 [K]	0.657	0.723	0.740	**0.462**	**0.488**	**0.497**	0.659	0.852	0.991	**0.491**	**0.522**	**0.531**
t850 [K]	0.690	0.777	0.809	**0.510**	**0.553**	**0.568**	**0.575**	0.698	0.802	0.593	**0.626**	**0.642**
t925 [K]	0.656	0.776	0.836	**0.501**	**0.564**	**0.588**	0.592	0.703	0.799	**0.589**	**0.659**	**0.700**
t1000 [K]	0.697	0.828	0.901	**0.539**	**0.614**	**0.657**	0.647	**0.733**	0.800	**0.635**	0.746	**0.790**
r100 [%]	0.0149	0.0190	0.0215	**0.00849**	**0.00883**	**0.00881**	0.0133	0.0154	0.0182	**0.0109**	**0.0112**	**0.0116**
r200 [%]	0.0957	0.102	0.105	**0.0579**	**0.0597**	**0.0601**	0.0937	0.0957	0.0937	**0.0707**	**0.0738**	**0.0747**
r400 [%]	0.203	0.206	0.207	**0.161**	**0.166**	**0.166**	**0.128**	**0.128**	**0.129**	0.171	0.174	0.175
r600 [%]	0.180	0.189	0.191	**0.148**	**0.156**	**0.158**	**0.122**	**0.121**	**0.122**	0.160	0.169	0.171
r700 [%]	0.162	0.175	0.178	**0.131**	**0.139**	**0.141**	**0.117**	**0.117**	**0.119**	0.143	0.153	0.156
r850 [%]	0.117	0.128	0.131	**0.0970**	**0.104**	**0.107**	0.111	0.115	**0.116**	**0.107**	**0.115**	0.117
r925 [%]	0.0813	0.0904	0.0935	**0.0701**	**0.0783**	**0.0817**	0.0907	0.102	0.104	**0.0779**	**0.0856**	**0.0886**
r1000 [%]	0.0585	0.0656	0.0679	**0.0498**	**0.0564**	**0.0596**	0.0765	0.0885	0.0922	**0.0564**	**0.0635**	**0.0646**
u100 [m/s]	2.51	3.15	3.45	**0.665**	**0.676**	**0.682**	1.47	1.61	1.66	**1.31**	**1.48**	**1.51**
u200 [m/s]	2.43	2.55	2.60	**0.771**	**0.784**	**0.794**	1.54	1.66	1.68	**0.888**	**0.903**	**0.914**
u400 [m/s]	3.08	3.25	3.29	**1.11**	**1.13**	**1.14**	1.68	1.74	1.77	**1.29**	**1.32**	**1.32**
u600 [m/s]	2.10	2.22	2.25	**1.16**	**1.17**	**1.17**	1.39	1.43	1.45	**1.18**	**1.18**	**1.19**
u700 [m/s]	1.90	2.02	2.05	**1.25**	**1.27**	**1.27**	1.36	1.38	1.40	**1.34**	**1.35**	**1.35**
u850 [m/s]	1.68	1.82	1.86	**1.20**	**1.23**	**1.24**	1.33	1.35	1.36	**1.26**	**1.28**	**1.28**
u925 [m/s]	1.52	1.68	1.72	**1.07**	**1.11**	**1.12**	1.43	1.46	1.47	**1.16**	**1.19**	**1.19**
u1000 [m/s]	1.28	1.41	1.45	**0.933**	**0.962**	**0.975**	1.23	1.26	1.27	**1.04**	**1.07**	**1.08**
v100 [m/s]	2.19	2.52	2.65	**0.673**	**0.684**	**0.690**	**1.28**	**1.44**	**1.54**	1.46	1.68	1.71
v200 [m/s]	2.51	2.61	2.64	**0.798**	**0.813**	**0.822**	1.52	1.75	1.87	**0.942**	**0.949**	**0.968**
v400 [m/s]	3.09	3.26	3.29	**1.16**	**1.18**	**1.19**	1.62	1.75	1.84	**1.33**	**1.34**	**1.35**
v600 [m/s]	2.07	2.19	2.21	**1.16**	**1.17**	**1.18**	1.35	1.41	1.45	**1.16**	**1.17**	**1.17**
v700 [m/s]	1.86	1.97	2.00	**1.23**	**1.25**	**1.25**	**1.27**	**1.31**	**1.33**	1.32	1.33	1.34
v850 [m/s]	1.65	1.77	1.81	**1.19**	**1.21**	**1.22**	1.43	1.46	1.48	**1.24**	**1.26**	**1.27**
v925 [m/s]	1.52	1.65	1.69	**1.07**	**1.10**	**1.12**	1.33	1.36	1.37	**1.15**	**1.17**	**1.18**
v1000 [m/s]	1.28	1.41	1.45	**0.926**	**0.952**	**0.964**	1.14	1.16	1.17	**1.02**	**1.04**	**1.06**
tw100 [m/s]	0.0523	0.0528	0.0529	**0.0454**	**0.0456**	**0.0456**	0.0497	0.0506	0.0507	**0.0484**	**0.0500**	**0.0506**
tw200 [m/s]	0.0656	0.0667	0.0671	**0.0452**	**0.0454**	**0.0455**	0.0598	0.0613	0.0619	**0.0553**	**0.0575**	**0.0586**
tw400 [m/s]	0.0891	0.0904	0.0909	**0.0675**	**0.0678**	**0.0680**	0.0828	0.0841	0.0846	**0.0767**	**0.0776**	**0.0773**
tw600 [m/s]	0.0927	0.0954	0.0966	**0.0662**	**0.0665**	**0.0668**	0.0872	0.0898	0.0905	**0.0747**	**0.0761**	**0.0763**
tw700 [m/s]	0.0952	0.0992	0.101	**0.0671**	**0.0675**	**0.0677**	0.0898	0.0931	0.0947	**0.0763**	**0.0779**	**0.0788**
tw850 [m/s]	0.0915	0.0994	0.104	**0.0633**	**0.0640**	**0.0644**	0.0851	0.0910	0.0946	**0.0732**	**0.0743**	**0.0751**
tw925 [m/s]	0.0748	0.0832	0.0876	**0.0503**	**0.0512**	**0.0516**	0.0696	0.0760	0.0796	**0.0581**	**0.0590**	**0.0598**
tw1000 [m/s]	0.0449	0.0535	0.0577	**0.0197**	**0.0205**	**0.0208**	0.0400	0.0471	0.0510	**0.0258**	**0.0279**	**0.0289**

Table 7: RMSE for all variables at different lead times for the IFS finetuned models with IFS boundary conditioning. The best deterministic and the best probabilistic model are in bold.

	Graph-FM			DET-LAM			Graph-EFM			CRPS-LAM		
Variable	24 h	48 h	72 h	24 h	48 h	72 h	24 h	48 h	72 h	24 h	48 h	72 h
pres_seasurface [Pa]	89.9	150	235	**80.2**	**144**	**228**	118	197	278	**80.6**	**146**	**235**
t2m [K]	0.922	1.13	1.32	**0.828**	**1.05**	**1.28**	0.990	1.19	1.37	**0.949**	**1.15**	**1.35**
u10m [m/s]	1.37	1.73	2.22	**1.25**	**1.66**	**2.16**	1.45	1.80	2.23	**1.32**	**1.70**	**2.21**
v10m [m/s]	1.36	1.73	2.19	**1.24**	**1.66**	**2.14**	1.41	1.76	2.18	**1.30**	**1.69**	**2.17**
pres0m [Pa]	88.2	147	231	**78.7**	**142**	**225**	117	196	277	**78.8**	**143**	**232**
lwavr0m [W/m^2]	21.9	24.2	**26.8**	**21.2**	**24.1**	27.0	23.6	25.8	27.9	**22.5**	**24.9**	**27.5**
swavr0m [W/m^2]	74.3	83.0	93.4	**72.7**	**82.7**	**93.4**	79.4	87.1	94.6	**75.3**	**83.7**	**93.7**
z100 [m^2/s^2]	148	213	271	**87.6**	**127**	**181**	166	263	344	**104**	**149**	**202**
z200 [m^2/s^2]	127	192	288	**92.8**	**167**	**275**	147	226	329	**103**	**180**	**292**
z400 [m^2/s^2]	125	194	303	**102**	**183**	**300**	143	229	341	**103**	**182**	**304**
z600 [m^2/s^2]	81.7	130	206	**68.9**	**124**	**202**	104	169	247	**70.3**	**126**	**210**
z700 [m^2/s^2]	73.3	118	186	**62.7**	**113**	**182**	93.2	155	225	**64.8**	**115**	**190**
z850 [m^2/s^2]	69.8	115	180	**61.2**	**110**	**176**	90.4	153	218	**62.5**	**113**	**183**
z925 [m^2/s^2]	70.0	117	184	**62.3**	**112**	**179**	92.5	157	223	**63.0**	**115**	**186**
z1000 [m^2/s^2]	72.8	121	189	**64.6**	**116**	**184**	95.5	161	227	**64.7**	**117**	**189**
t100 [K]	1.13	1.46	1.65	**0.458**	**0.555**	**0.704**	0.985	1.33	1.55	**0.524**	**0.631**	**0.788**
t200 [K]	1.29	1.58	1.99	**0.869**	**1.22**	**1.75**	1.11	1.49	1.99	**0.910**	**1.26**	**1.80**
t400 [K]	0.961	1.18	1.55	**0.702**	**1.02**	**1.47**	0.844	1.15	1.55	**0.758**	**1.06**	**1.50**
t600 [K]	0.885	1.10	1.42	**0.697**	**0.977**	**1.37**	0.845	1.11	1.45	**0.730**	**0.993**	**1.38**
t700 [K]	0.815	1.02	1.32	**0.719**	**0.960**	**1.29**	0.959	1.19	1.49	**0.748**	**0.979**	**1.33**
t850 [K]	0.847	1.06	1.32	**0.760**	**0.984**	**1.28**	0.884	1.09	1.36	**0.834**	**1.03**	**1.32**
t925 [K]	0.809	1.04	1.31	**0.736**	**0.977**	**1.25**	0.886	1.10	1.33	**0.819**	**1.05**	**1.31**
t1000 [K]	0.814	1.02	1.23	**0.721**	**0.946**	**1.19**	0.866	1.05	1.24	**0.807**	**1.02**	**1.24**
r100 [%]	0.0151	0.0178	0.0192	**0.00997**	**0.0104**	**0.0114**	0.0129	0.0144	0.0155	**0.0113**	**0.0120**	**0.0130**
r200 [%]	0.107	0.121	0.141	**0.0795**	**0.0991**	**0.128**	0.103	0.115	0.137	**0.0873**	**0.105**	**0.132**
r400 [%]	0.226	0.240	**0.260**	**0.203**	**0.232**	0.263	**0.205**	**0.234**	0.266	0.213	0.237	**0.264**
r600 [%]	0.203	**0.230**	**0.260**	**0.195**	0.233	0.269	0.225	0.260	0.290	**0.206**	**0.239**	**0.270**
r700 [%]	0.186	0.216	**0.246**	**0.175**	**0.215**	0.249	**0.181**	**0.213**	**0.244**	0.187	0.220	0.251
r850 [%]	0.132	0.152	**0.171**	**0.126**	**0.150**	0.171	0.136	**0.155**	**0.174**	**0.135**	0.157	0.176
r925 [%]	0.0915	0.107	**0.120**	**0.0857**	**0.106**	0.122	0.103	0.117	0.128	**0.0953**	**0.112**	**0.126**
r1000 [%]	0.0686	0.0810	0.0915	**0.0601**	**0.0758**	**0.0893**	0.0712	0.0821	0.0907	**0.0667**	**0.0792**	**0.0900**
u100 [m/s]	2.39	2.86	3.08	**1.09**	**1.25**	**1.57**	1.80	2.02	2.24	**1.55**	**1.76**	**2.00**
u200 [m/s]	2.90	3.44	4.34	**1.96**	**2.82**	**4.07**	2.62	3.46	4.55	**2.13**	**2.95**	**4.18**
u400 [m/s]	3.82	4.59	5.84	**3.05**	**4.23**	**5.76**	3.72	4.71	6.08	**3.33**	**4.40**	**5.94**
u600 [m/s]	2.59	3.10	3.87	**2.21**	**2.88**	**3.79**	2.82	3.47	4.23	**2.36**	**2.98**	**3.91**
u700 [m/s]	2.37	2.85	3.53	**2.11**	**2.70**	**3.49**	2.58	3.15	3.81	**2.24**	**2.80**	**3.58**
u850 [m/s]	2.18	2.71	3.44	**1.99**	**2.61**	**3.39**	2.26	2.83	3.52	**2.12**	**2.70**	**3.46**
u925 [m/s]	2.02	2.58	3.35	**1.83**	**2.48**	**3.28**	2.11	2.69	3.38	**1.92**	**2.53**	**3.33**
u1000 [m/s]	1.59	2.00	2.56	**1.44**	**1.91**	**2.49**	1.68	2.08	2.57	**1.54**	**1.97**	**2.55**
v100 [m/s]	2.10	2.43	2.67	**1.07**	**1.27**	**1.63**	**1.68**	**1.88**	**2.13**	1.73	2.00	2.25
v200 [m/s]	3.02	3.62	4.69	**2.08**	**2.95**	**4.32**	2.78	3.56	4.69	**2.37**	**3.21**	**4.53**
v400 [m/s]	3.86	4.69	**5.96**	**3.22**	**4.43**	5.97	3.65	4.83	6.22	**3.53**	**4.64**	**6.17**
v600 [m/s]	2.56	3.05	3.75	**2.26**	**2.91**	**3.74**	2.91	3.50	4.18	**2.34**	**2.96**	**3.82**
v700 [m/s]	2.32	2.78	3.42	**2.09**	**2.67**	**3.38**	2.54	3.06	3.67	**2.17**	**2.71**	**3.42**
v850 [m/s]	2.13	2.65	3.33	**1.96**	**2.57**	**3.30**	2.24	2.80	3.45	**2.02**	**2.59**	**3.32**
v925 [m/s]	2.00	2.55	3.26	**1.83**	**2.47**	**3.22**	2.08	2.65	3.31	**1.89**	**2.49**	**3.24**
v1000 [m/s]	1.59	2.00	2.52	**1.43**	**1.92**	**2.48**	1.65	2.05	2.53	**1.51**	**1.95**	**2.50**
tw100 [m/s]	0.0523	0.0527	0.0529	**0.0477**	**0.0487**	**0.0501**	0.0516	**0.0522**	**0.0527**	**0.0515**	0.0522	0.0527
tw200 [m/s]	0.0662	0.0674	0.0681	**0.0508**	**0.0551**	**0.0605**	0.0619	0.0639	0.0661	**0.0607**	**0.0627**	**0.0651**
tw400 [m/s]	0.0914	0.0931	0.0948	**0.0755**	**0.0809**	**0.0886**	0.0865	0.0893	0.0924	**0.0825**	**0.0860**	**0.0910**
tw600 [m/s]	0.0964	0.0996	0.102	**0.0757**	**0.0823**	**0.0913**	0.0917	0.0953	0.0992	**0.0822**	**0.0870**	**0.0938**
tw700 [m/s]	0.0983	0.102	0.106	**0.0758**	**0.0821**	**0.0906**	0.0926	0.0965	0.100	**0.0831**	**0.0878**	**0.0942**
tw850 [m/s]	0.0923	0.0992	0.104	**0.0699**	**0.0754**	**0.0824**	0.0860	0.0914	0.0960	**0.0775**	**0.0818**	**0.0871**
tw925 [m/s]	0.0752	0.0826	0.0872	**0.0550**	**0.0595**	**0.0650**	0.0688	0.0744	0.0787	**0.0607**	**0.0647**	**0.0694**
tw1000 [m/s]	0.0442	0.0517	0.0562	**0.0239**	**0.0283**	**0.0339**	0.0387	0.0443	0.0486	**0.0290**	**0.0336**	**0.0385**

Table 8: SSR for all variables at different lead times. The model with SSR closest to 1 is in bold, per setting.

	ERA5						IFS Finetuned					
	Graph-EFM			CRPS-LAM			Graph-EFM			CRPS-LAM		
Variable	24 h	48 h	72 h	24 h	48 h	72 h	24 h	48 h	72 h	24 h	48 h	72 h
pres_seasurface	0.741	0.700	0.689	**0.926**	**0.863**	**0.814**	0.446	0.338	0.265	**0.982**	**0.831**	**0.624**
t2m	0.652	0.601	0.587	**0.900**	**0.842**	**0.815**	0.626	0.549	0.485	**0.897**	**0.820**	**0.743**
u10m	0.663	0.644	0.638	**0.900**	**0.895**	**0.890**	0.547	0.450	0.364	**0.849**	**0.714**	**0.582**
v10m	0.709	0.696	0.691	**0.900**	**0.898**	**0.887**	0.570	0.466	0.378	**0.851**	**0.720**	**0.598**
pres0m	0.639	0.544	0.486	**0.915**	**0.850**	**0.799**	0.445	0.335	0.263	**0.977**	**0.842**	**0.636**
lwavr0m	0.707	0.677	0.669	**0.957**	**0.948**	**0.948**	0.640	0.594	0.552	**0.919**	**0.860**	**0.795**
swavr0m	0.692	0.684	0.684	**0.939**	**0.940**	**0.937**	0.597	0.549	0.503	**0.884**	**0.823**	**0.748**
z100	0.414	0.320	0.279	**1.03**	**0.916**	**0.888**	0.582	0.502	0.453	**1.07**	**1.15**	**1.11**
z200	0.341	0.291	0.270	**1.07**	**0.999**	**0.945**	0.608	0.525	0.420	**0.989**	**1.02**	**0.819**
z400	0.414	0.366	0.352	**1.08**	**1.00**	**0.964**	0.573	0.455	0.349	**0.901**	**0.780**	**0.579**
z600	0.493	0.446	0.436	**0.991**	**0.922**	**0.882**	0.564	0.448	0.350	**1.04**	**0.952**	**0.709**
z700	0.565	0.527	0.524	**0.994**	**0.906**	**0.859**	0.541	0.424	0.333	**1.08**	**0.980**	**0.734**
z850	0.713	0.655	0.626	**0.918**	**0.838**	**0.785**	0.486	0.372	0.295	**1.02**	**0.907**	**0.690**
z925	0.729	0.674	0.651	**0.912**	**0.836**	**0.785**	0.461	0.348	0.275	**1.01**	**0.874**	**0.661**
z1000	0.751	0.710	0.699	**0.925**	**0.863**	**0.815**	0.449	0.336	0.263	**0.993**	**0.856**	**0.647**
t100	0.612	0.508	0.447	**1.23**	**1.17**	**1.11**	0.524	0.444	0.403	**1.08**	**1.03**	**0.942**
t200	0.857	0.788	0.734	**1.09**	**1.09**	**1.07**	0.566	0.456	0.358	**0.847**	**0.653**	**0.490**
t400	0.702	0.536	0.473	**1.25**	**1.24**	**1.22**	0.558	0.438	0.337	**0.816**	**0.645**	**0.495**
t600	0.401	0.336	0.310	**1.14**	**1.12**	**1.12**	0.579	0.469	0.371	**0.926**	**0.737**	**0.570**
t700	0.476	0.391	0.357	**1.01**	**0.985**	**0.975**	0.492	0.420	0.348	**0.918**	**0.769**	**0.601**
t850	0.591	0.507	0.455	**1.02**	**1.02**	**1.01**	0.516	0.440	0.365	**0.956**	**0.829**	**0.675**
t925	0.585	0.514	0.472	**1.01**	**0.967**	**0.928**	0.510	0.434	0.367	**0.946**	**0.819**	**0.694**
t1000	0.704	0.644	0.607	**0.973**	**0.904**	**0.874**	0.581	0.502	0.433	**0.960**	**0.860**	**0.756**
r100	0.241	0.233	0.220	**0.619**	**0.624**	**0.651**	0.319	0.316	0.304	**0.805**	**0.912**	**1.02**
r200	0.467	0.469	0.486	**0.910**	**0.894**	**0.868**	0.438	0.400	0.340	**0.895**	**0.778**	**0.650**
r400	0.794	0.804	0.825	**1.03**	**1.03**	**1.03**	0.648	0.570	0.503	**0.902**	**0.828**	**0.760**
r600	0.585	0.593	0.598	**1.02**	**0.996**	**0.988**	0.524	0.456	0.409	**0.886**	**0.793**	**0.715**
r700	0.645	0.654	0.677	**0.987**	**0.966**	**0.951**	0.574	0.496	0.434	**0.895**	**0.801**	**0.724**
r850	0.607	0.591	0.598	**1.00**	**0.989**	**0.974**	0.611	0.549	0.491	**0.961**	**0.874**	**0.796**
r925	0.602	0.538	0.532	**0.969**	**0.936**	**0.918**	0.650	0.592	0.544	**0.876**	**0.803**	**0.730**
r1000	0.562	0.484	0.468	**0.926**	**0.874**	**0.867**	0.665	0.599	0.546	**0.897**	**0.827**	**0.756**
u100	0.716	0.702	0.698	**1.20**	**1.10**	**1.10**	0.710	0.672	0.614	**1.16**	**1.09**	**1.00**
u200	**0.914**	**0.892**	**0.936**	1.12	1.13	1.12	0.695	0.547	0.419	**0.838**	**0.651**	**0.500**
u400	**0.905**	**0.888**	**0.900**	1.32	1.32	1.34	0.686	0.550	0.427	**0.866**	**0.685**	**0.530**
u600	0.783	0.786	0.824	**1.08**	**1.09**	**1.09**	0.578	0.478	0.392	**0.853**	**0.709**	**0.573**
u700	0.745	0.755	0.801	**0.977**	**0.979**	**0.978**	0.560	0.467	0.387	**0.874**	**0.734**	**0.592**
u850	0.517	0.535	0.582	**1.01**	**1.01**	**1.01**	0.561	0.455	0.366	**0.869**	**0.738**	**0.609**
u925	0.549	0.548	0.564	**0.982**	**0.977**	**0.973**	0.517	0.416	0.331	**0.842**	**0.688**	**0.549**
u1000	0.661	0.651	0.652	**0.912**	**0.905**	**0.899**	0.555	0.460	0.372	**0.855**	**0.717**	**0.578**
v100	0.742	0.697	0.681	**1.05**	**0.954**	**0.953**	0.669	0.621	0.554	**0.995**	**0.923**	**0.859**
v200	**0.967**	**0.872**	**0.876**	1.17	1.18	1.16	0.695	0.559	0.426	**0.881**	**0.694**	**0.527**
v400	**0.655**	**0.661**	**0.738**	1.36	1.36	1.36	0.609	0.467	0.363	**0.927**	**0.752**	**0.608**
v600	0.789	0.781	0.818	**1.11**	**1.10**	**1.10**	0.561	0.473	0.397	**0.897**	**0.757**	**0.629**
v700	0.746	0.736	0.744	**0.995**	**0.994**	**0.991**	0.566	0.477	0.397	**0.890**	**0.758**	**0.628**
v850	0.477	0.483	0.517	**1.02**	**1.02**	**1.01**	0.505	0.411	0.334	**0.915**	**0.777**	**0.643**
v925	0.594	0.589	0.600	**0.972**	**0.968**	**0.963**	0.530	0.423	0.339	**0.872**	**0.721**	**0.590**
v1000	0.723	0.718	0.725	**0.916**	**0.911**	**0.901**	0.576	0.474	0.384	**0.864**	**0.729**	**0.606**
tw100	0.667	0.659	0.664	**0.847**	**0.835**	**0.829**	0.547	0.541	0.535	**0.865**	**0.894**	**0.920**
tw200	0.301	0.301	0.309	**0.802**	**0.793**	**0.786**	0.235	0.232	0.224	**0.822**	**0.865**	**0.917**
tw400	0.237	0.241	0.257	**0.750**	**0.758**	**0.758**	0.221	0.217	0.209	**0.796**	**0.853**	**0.916**
tw600	0.254	0.251	0.260	**0.686**	**0.687**	**0.687**	0.248	0.243	0.233	**0.750**	**0.802**	**0.854**
tw700	0.246	0.245	0.259	**0.650**	**0.652**	**0.647**	0.243	0.237	0.229	**0.711**	**0.786**	**0.855**
tw850	0.262	0.253	0.255	**0.635**	**0.634**	**0.627**	0.258	0.254	0.249	**0.705**	**0.739**	**0.764**
tw925	0.272	0.258	0.263	**0.648**	**0.646**	**0.636**	0.272	0.267	0.261	**0.713**	**0.738**	**0.754**
tw1000	0.262	0.235	0.233	**0.750**	**0.723**	**0.706**	0.281	0.268	0.257	**0.822**	**0.799**	**0.760**

Table 9: LSD for all variables at different lead times with ERA5 boundary. The best deterministic and the best probabilistic model are in bold.

	Graph-FM			DET-LAM			Graph-EFM			CRPS-LAM		
Variable	24 h	48 h	72 h	24 h	48 h	72 h	24 h	48 h	72 h	24 h	48 h	72 h
pres_seasurface [Pa]	1.08e3	1.20e3	1.25e3	**494**	**654**	**764**	715	855	993	**480**	**595**	**733**
t2m [K]	**2.15**	**2.22**	**2.21**	5.33	5.92	5.60	2.22	2.30	2.28	**1.17**	**1.38**	**1.48**
u10m [m/s]	3.42	3.48	3.48	**0.919**	**1.40**	**1.83**	2.82	3.02	3.10	**1.20**	**1.77**	**2.22**
v10m [m/s]	3.62	3.66	3.67	**1.08**	**1.57**	**1.97**	3.14	3.34	3.39	**1.34**	**1.98**	**2.42**
pres0m [Pa]	1.36e3	1.56e3	1.68e3	**402**	**606**	**763**	705	969	1.16e3	**371**	**552**	**707**
lwavr0m [W/m^2]	**31.4**	**31.9**	**32.1**	43.3	43.9	42.5	**26.9**	**27.5**	**27.7**	30.5	31.9	32.1
swavr0m [W/m^2]	**974**	**972**	**971**	1.05e3	1.07e3	1.08e3	**697**	**699**	**700**	728	743	750
z100 [m^2/s^2]	2.93e3	3.37e3	3.61e3	**1.57e3**	**2.04e3**	**2.37e3**	2.70e3	3.22e3	3.48e3	**1.54e3**	**1.95e3**	**2.18e3**
z200 [m^2/s^2]	2.28e3	2.58e3	2.71e3	**1.07e3**	**1.45e3**	**1.73e3**	1.88e3	2.29e3	2.46e3	**1.01e3**	**1.35e3**	**1.61e3**
z400 [m^2/s^2]	1.81e3	2.02e3	2.11e3	**900**	**1.20e3**	**1.41e3**	1.44e3	1.65e3	1.79e3	**854**	**1.10e3**	**1.33e3**
z600 [m^2/s^2]	1.29e3	1.43e3	1.48e3	**557**	**737**	**862**	891	1.03e3	1.17e3	**519**	**660**	**846**
z700 [m^2/s^2]	1.13e3	1.23e3	1.27e3	**480**	**638**	**750**	738	852	970	**437**	**565**	**731**
z850 [m^2/s^2]	930	1.01e3	1.04e3	**438**	**564**	**653**	589	714	825	**382**	**483**	**611**
z925 [m^2/s^2]	874	955	986	**410**	**544**	**637**	558	680	790	**372**	**472**	**591**
z1000 [m^2/s^2]	861	959	1.01e3	**403**	**535**	**625**	580	703	818	**376**	**471**	**581**
t100 [K]	6.05	6.87	7.34	**3.44**	**4.32**	**4.97**	5.70	6.67	7.26	**3.66**	**4.56**	**5.02**
t200 [K]	6.58	6.94	**7.08**	**4.93**	**6.28**	7.20	6.26	6.83	6.95	**4.71**	**5.96**	**6.67**
t400 [K]	6.70	6.97	7.13	**4.87**	**6.05**	**6.86**	6.62	7.11	7.30	**4.97**	**6.05**	**6.70**
t600 [K]	6.51	6.86	7.05	**4.62**	**5.90**	**6.78**	6.36	6.92	7.13	**4.83**	**5.86**	**6.42**
t700 [K]	6.24	6.55	6.74	**4.34**	**5.30**	**5.94**	6.35	6.78	7.08	**4.61**	**5.58**	**6.27**
t850 [K]	6.85	7.16	7.46	**4.46**	**5.57**	**6.34**	6.44	6.95	7.24	**4.84**	**5.86**	**6.51**
t925 [K]	7.95	8.54	9.17	**5.76**	**6.92**	**7.69**	7.27	7.67	8.06	**5.51**	**6.46**	**7.28**
t1000 [K]	8.94	9.20	9.40	**6.84**	**7.94**	**8.69**	8.40	8.71	8.98	**5.91**	**6.92**	**7.69**
r100 [%]	0.0461	0.0528	**0.0565**	**0.0428**	**0.0513**	0.0573	0.0678	0.0742	0.0781	**0.0412**	**0.0478**	**0.0519**
r200 [%]	0.314	0.316	**0.320**	**0.255**	**0.311**	0.348	0.316	0.331	**0.333**	**0.268**	**0.321**	0.353
r400 [%]	0.458	**0.423**	**0.418**	**0.381**	0.441	0.482	**0.384**	**0.392**	**0.399**	0.416	0.460	0.492
r600 [%]	0.479	**0.433**	**0.429**	**0.386**	0.442	0.480	**0.395**	**0.406**	**0.409**	0.427	0.490	0.519
r700 [%]	0.458	**0.421**	**0.417**	**0.391**	0.455	0.498	**0.404**	**0.409**	**0.420**	0.424	0.473	0.507
r850 [%]	0.324	**0.315**	**0.316**	**0.287**	0.336	0.370	0.320	**0.318**	**0.333**	**0.312**	0.346	0.383
r925 [%]	0.249	**0.242**	**0.242**	**0.219**	0.256	0.280	0.250	**0.247**	**0.256**	**0.240**	0.264	0.292
r1000 [%]	0.208	0.204	**0.205**	**0.168**	**0.195**	0.213	0.199	0.201	**0.204**	**0.168**	**0.193**	0.210
u100 [m/s]	10.1	10.9	11.3	**6.43**	**8.03**	**9.10**	9.96	10.9	11.4	**7.20**	**8.57**	**9.27**
u200 [m/s]	11.8	12.4	**12.6**	**8.80**	**11.3**	12.9	13.0	13.8	14.5	**9.61**	**11.2**	**12.6**
u400 [m/s]	12.5	**12.8**	**13.0**	**11.3**	14.2	16.1	13.2	14.1	**14.2**	**10.9**	**13.3**	14.7
u600 [m/s]	8.68	8.90	**8.98**	**7.08**	**8.79**	9.94	9.11	9.50	**9.71**	**7.92**	**9.31**	10.2
u700 [m/s]	7.76	**7.91**	**7.97**	**6.49**	8.20	9.38	8.26	**8.49**	**8.78**	**7.45**	8.61	9.43
u850 [m/s]	7.51	**7.65**	**7.71**	**6.35**	7.77	8.73	7.96	8.14	**8.37**	**6.78**	**7.99**	8.91
u925 [m/s]	7.43	**7.56**	**7.66**	**6.55**	7.89	8.83	8.13	8.26	**8.46**	**6.60**	**7.91**	8.85
u1000 [m/s]	7.26	7.35	7.44	**4.92**	**5.91**	**6.57**	7.15	7.27	7.46	**5.60**	**6.56**	**7.37**
v100 [m/s]	9.47	10.1	**10.4**	**7.27**	**9.20**	10.5	9.49	10.3	10.9	**7.17**	**8.34**	**9.33**
v200 [m/s]	13.4	14.0	**14.2**	**10.6**	**13.0**	14.7	15.1	15.9	16.6	**10.7**	**13.0**	**14.9**
v400 [m/s]	14.9	**15.1**	**15.3**	**13.4**	16.3	18.2	16.0	16.5	**17.2**	**14.1**	**16.3**	18.1
v600 [m/s]	9.74	9.92	**10.0**	**8.19**	**9.89**	11.0	10.5	10.7	**11.3**	**9.05**	**10.5**	12.2
v700 [m/s]	8.45	**8.58**	**8.66**	**7.25**	8.64	9.59	9.11	**9.16**	**9.59**	**8.19**	9.40	10.8
v850 [m/s]	7.73	**7.80**	**7.85**	**7.25**	8.74	9.75	8.47	**8.49**	**8.83**	**7.42**	8.51	9.66
v925 [m/s]	7.48	**7.58**	**7.62**	**6.52**	7.91	8.86	8.19	8.30	**8.57**	**6.80**	**7.79**	8.65
v1000 [m/s]	7.73	7.95	8.15	**5.36**	**6.30**	**6.92**	7.79	8.00	8.13	**6.06**	**7.18**	**7.83**
tw100 [m/s]	0.0742	0.0789	0.0814	**0.0621**	**0.0710**	**0.0773**	0.0728	0.0768	0.0785	**0.0624**	**0.0691**	**0.0737**
tw200 [m/s]	0.102	0.105	0.106	**0.0834**	**0.0961**	**0.105**	0.104	0.108	0.109	**0.0857**	**0.0973**	**0.105**
tw400 [m/s]	0.156	0.160	**0.161**	**0.132**	**0.152**	0.165	0.160	0.165	**0.167**	**0.141**	**0.161**	0.174
tw600 [m/s]	0.180	0.184	0.185	**0.142**	**0.161**	**0.174**	0.185	0.189	**0.191**	**0.156**	**0.179**	0.195
tw700 [m/s]	0.195	0.198	0.198	**0.157**	**0.177**	**0.191**	0.201	0.205	**0.207**	**0.167**	**0.192**	0.208
tw850 [m/s]	0.220	0.227	0.228	**0.177**	**0.203**	**0.220**	0.229	0.236	0.239	**0.188**	**0.213**	**0.230**
tw925 [m/s]	0.194	0.201	0.203	**0.148**	**0.170**	**0.183**	0.199	0.206	0.209	**0.169**	**0.190**	**0.203**
tw1000 [m/s]	0.129	0.134	0.135	**0.0911**	**0.107**	**0.118**	0.132	0.138	**0.140**	**0.121**	**0.134**	0.143

Table 10: LSD for all variables at different lead times for the IFS finetuned models with IFS boundary. The best deterministic and the best probabilistic model are in bold.

	Graph-FM			DET-LAM			Graph-EFM			CRPS-LAM		
Variable	24 h	48 h	72 h	24 h	48 h	72 h	24 h	48 h	72 h	24 h	48 h	72 h
pres_seasurface [Pa]	$1.04e3$	$1.17e3$	$1.23e3$	**488**	**649**	**763**	676	828	920	**492**	**641**	**742**
t2m [K]	**2.14**	**2.29**	**2.31**	4.44	5.60	5.63	1.89	2.01	2.04	**1.18**	**1.43**	**1.54**
u10m [m/s]	3.44	3.52	3.56	**0.962**	**1.37**	**1.76**	2.86	3.15	3.28	**1.25**	**1.84**	**2.29**
v10m [m/s]	3.68	3.77	3.80	**1.10**	**1.52**	**1.90**	3.19	3.43	3.52	**1.41**	**2.03**	**2.50**
pres0m [Pa]	$1.29e3$	$1.50e3$	$1.61e3$	**423**	**637**	**800**	676	915	$1.07e3$	**402**	**581**	**714**
lwavr0m [W/m^2]	**32.4**	**33.1**	**33.4**	43.1	45.3	44.7	**25.1**	**25.2**	**25.3**	29.6	30.8	31.1
swavr0m [W/m^2]	**993**	**991**	**989**	$1.05e3$	$1.06e3$	$1.07e3$	**998**	**997**	**997**	$1.04e3$	$1.06e3$	$1.07e3$
z100 [m^2/s^2]	$2.89e3$	$3.31e3$	$3.52e3$	**1.55e3**	**2.03e3**	**2.38e3**	$2.72e3$	$3.16e3$	$3.42e3$	**1.58e3**	**1.99e3**	**2.26e3**
z200 [m^2/s^2]	$2.21e3$	$2.54e3$	$2.69e3$	**1.04e3**	**1.39e3**	**1.66e3**	$1.86e3$	$2.16e3$	$2.34e3$	**1.08e3**	**1.41e3**	**1.63e3**
z400 [m^2/s^2]	$1.75e3$	$1.98e3$	$2.09e3$	**849**	**1.12e3**	**1.33e3**	$1.44e3$	$1.60e3$	$1.71e3$	**899**	**1.16e3**	**1.34e3**
z600 [m^2/s^2]	$1.24e3$	$1.40e3$	$1.46e3$	**551**	**730**	**857**	875	$1.03e3$	$1.13e3$	**554**	**723**	**834**
z700 [m^2/s^2]	$1.09e3$	$1.21e3$	$1.26e3$	**462**	**614**	**724**	732	858	938	**468**	**612**	**708**
z850 [m^2/s^2]	906	$1.00e3$	$1.04e3$	**431**	**559**	**649**	575	699	778	**401**	**526**	**612**
z925 [m^2/s^2]	847	939	977	**397**	**529**	**623**	536	661	739	**387**	**510**	**595**
z1000 [m^2/s^2]	828	930	981	**396**	**525**	**616**	551	677	754	**388**	**506**	**587**
t100 [K]	5.79	6.54	6.97	**3.29**	**4.17**	**4.82**	5.60	6.37	6.82	**3.75**	**4.58**	**5.15**
t200 [K]	6.36	6.79	7.03	**4.63**	**5.91**	**6.83**	6.30	6.59	6.75	**5.02**	**6.05**	**6.71**
t400 [K]	6.46	6.84	7.08	**4.47**	**5.49**	**6.21**	6.49	6.71	**6.86**	**5.31**	**6.29**	6.91
t600 [K]	6.34	6.74	6.99	**4.25**	**5.44**	**6.30**	6.23	6.47	6.63	**4.99**	**5.98**	**6.60**
t700 [K]	6.01	6.38	6.64	**4.22**	**5.17**	**5.82**	6.07	6.35	6.50	**4.81**	**5.75**	**6.34**
t850 [K]	6.58	6.94	7.15	**4.16**	**5.15**	**5.85**	6.36	6.66	6.83	**4.96**	**5.95**	**6.59**
t925 [K]	7.92	8.24	8.51	**5.30**	**6.31**	**6.98**	7.13	7.46	7.64	**5.53**	**6.60**	**7.28**
t1000 [K]	9.13	9.28	9.43	**6.62**	**7.82**	**8.65**	8.33	8.54	8.65	**5.94**	**7.05**	**7.76**
r100 [%]	0.0443	0.0506	**0.0542**	**0.0411**	**0.0495**	0.0553	0.0633	0.0650	0.0658	**0.0418**	**0.0487**	**0.0534**
r200 [%]	0.306	**0.306**	**0.313**	**0.252**	0.309	0.348	**0.301**	**0.305**	**0.308**	0.303	0.351	0.382
r400 [%]	0.435	**0.401**	**0.393**	**0.368**	0.422	0.460	**0.379**	**0.377**	**0.376**	0.416	0.464	0.494
r600 [%]	0.475	**0.390**	**0.385**	**0.368**	0.417	0.450	**0.405**	**0.404**	**0.405**	0.433	0.485	0.517
r700 [%]	0.463	**0.394**	**0.387**	**0.371**	0.424	0.460	**0.404**	**0.404**	**0.403**	0.420	0.472	0.503
r850 [%]	0.337	0.311	**0.311**	**0.269**	**0.307**	0.334	0.324	**0.325**	**0.326**	**0.310**	0.351	0.378
r925 [%]	0.255	**0.241**	**0.241**	**0.209**	0.242	0.265	0.249	**0.248**	**0.249**	**0.236**	0.268	0.288
r1000 [%]	0.219	0.213	0.212	**0.164**	**0.189**	**0.206**	0.202	0.202	**0.202**	**0.170**	**0.196**	0.213
u100 [m/s]	9.27	10.1	10.4	**6.12**	**7.74**	**8.85**	9.83	10.6	10.8	**7.35**	**8.74**	**9.63**
u200 [m/s]	11.3	12.0	**12.5**	**8.53**	**11.0**	12.8	12.6	13.2	13.4	**9.87**	**12.0**	**13.3**
u400 [m/s]	11.8	**12.2**	**12.6**	**10.7**	13.5	15.5	13.4	**13.7**	**13.8**	**11.9**	14.0	15.4
u600 [m/s]	8.04	8.45	**8.75**	**6.65**	**8.23**	9.37	8.89	**9.06**	**9.12**	**8.11**	9.64	10.6
u700 [m/s]	7.15	**7.49**	**7.77**	**5.99**	7.54	8.68	8.08	**8.21**	**8.27**	**7.40**	8.74	9.61
u850 [m/s]	7.08	**7.26**	**7.43**	**6.10**	7.46	8.43	7.92	**8.00**	**8.05**	**6.88**	8.17	9.00
u925 [m/s]	7.21	**7.32**	**7.44**	**6.28**	7.55	8.44	8.09	8.22	**8.32**	**6.60**	**7.85**	8.67
u1000 [m/s]	7.12	7.21	7.27	**4.67**	**5.64**	**6.28**	7.35	7.63	7.79	**5.47**	**6.53**	**7.23**
v100 [m/s]	8.54	9.22	**9.58**	**6.85**	**8.77**	10.1	9.29	9.92	10.2	**7.22**	**8.63**	**9.52**
v200 [m/s]	13.0	14.0	14.6	**10.0**	**12.4**	**14.0**	15.0	15.6	15.9	**11.4**	**13.7**	**15.2**
v400 [m/s]	14.6	**15.0**	**15.4**	**12.4**	15.1	16.8	15.8	**16.2**	**16.3**	**14.8**	17.3	18.8
v600 [m/s]	9.34	9.76	**10.0**	**7.64**	**9.16**	10.2	10.2	**10.4**	**10.4**	**9.42**	11.0	12.0
v700 [m/s]	8.08	8.44	**8.68**	**6.85**	**8.20**	9.14	8.94	**9.11**	**9.19**	**8.37**	9.70	10.5
v850 [m/s]	7.58	**7.74**	**7.92**	**6.66**	7.97	8.90	8.36	**8.49**	**8.59**	**7.51**	8.72	9.52
v925 [m/s]	7.64	7.78	**7.91**	**6.22**	**7.52**	8.42	8.18	8.35	**8.47**	**6.72**	**7.89**	8.65
v1000 [m/s]	7.76	7.92	8.00	**5.11**	**6.00**	**6.59**	8.01	8.32	8.51	**5.95**	**6.97**	**7.64**
tw100 [m/s]	0.0718	0.0760	0.0786	**0.0617**	**0.0707**	**0.0770**	0.0749	0.0786	0.0809	**0.0627**	**0.0696**	**0.0742**
tw200 [m/s]	0.0995	0.103	0.105	**0.0806**	**0.0921**	**0.100**	0.107	0.111	0.114	**0.0870**	**0.0989**	**0.107**
tw400 [m/s]	0.153	0.158	**0.160**	**0.129**	**0.149**	0.162	0.163	0.168	**0.171**	**0.142**	**0.162**	0.174
tw600 [m/s]	0.175	0.181	0.182	**0.139**	**0.158**	**0.171**	0.187	0.193	0.196	**0.158**	**0.180**	**0.195**
tw700 [m/s]	0.192	0.196	0.196	**0.151**	**0.171**	**0.185**	0.203	0.209	0.213	**0.170**	**0.193**	**0.209**
tw850 [m/s]	0.216	0.223	0.225	**0.170**	**0.195**	**0.211**	0.229	0.242	0.249	**0.189**	**0.214**	**0.229**
tw925 [m/s]	0.188	0.196	0.198	**0.142**	**0.163**	**0.176**	0.201	0.214	0.221	**0.172**	**0.192**	**0.204**
tw1000 [m/s]	0.125	0.130	0.132	**0.0870**	**0.103**	**0.113**	0.133	0.142	0.147	**0.125**	**0.137**	**0.145**

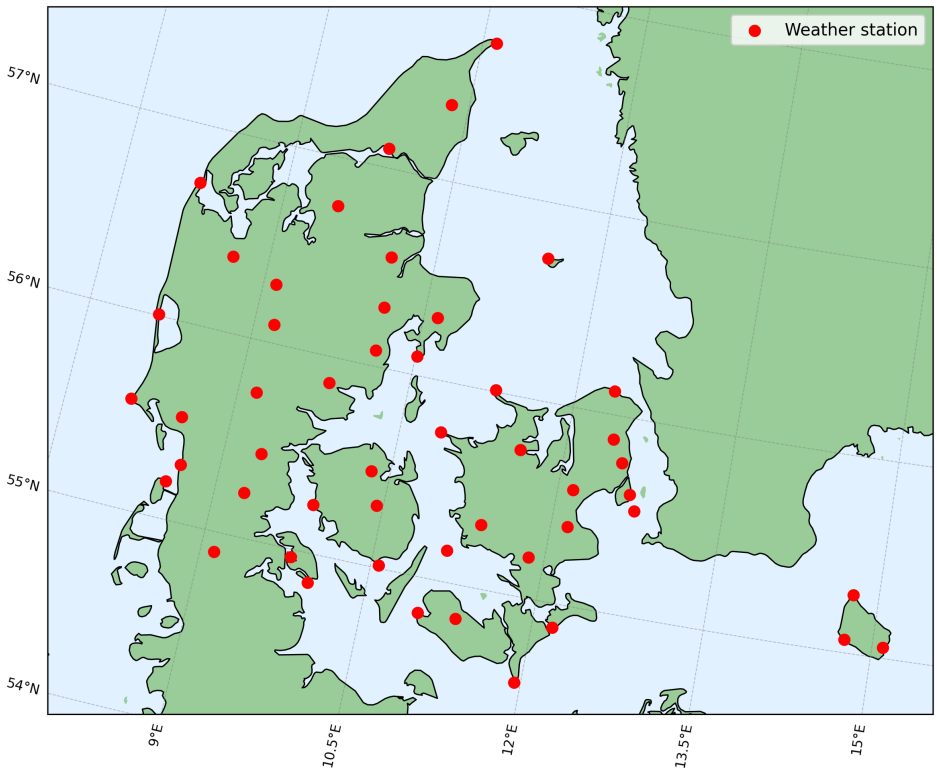
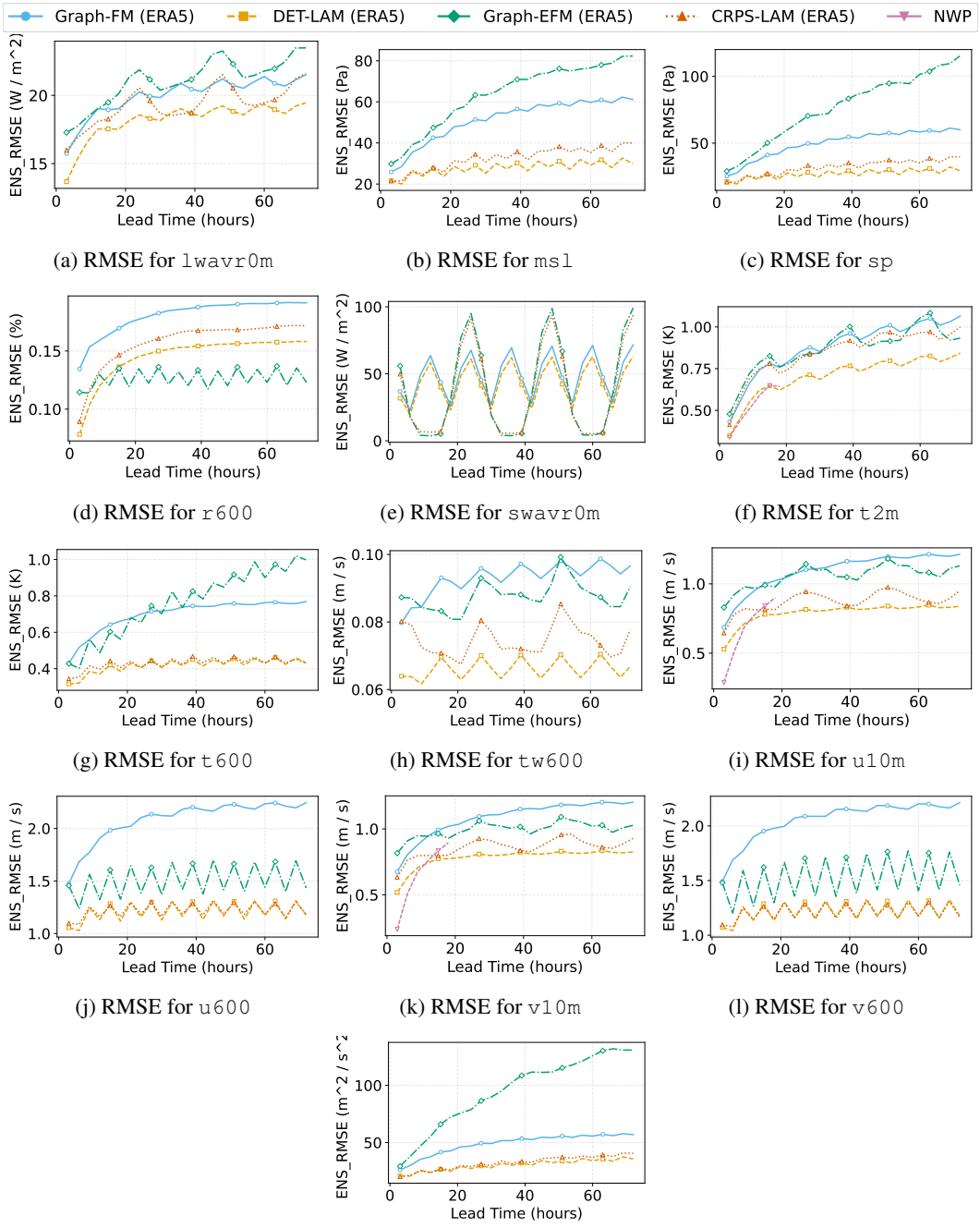


Figure 9: The locations of the in-situ observations in Denmark.



(a) RMSE for `lwavr0m` (b) RMSE for `msl` (c) RMSE for `sp`

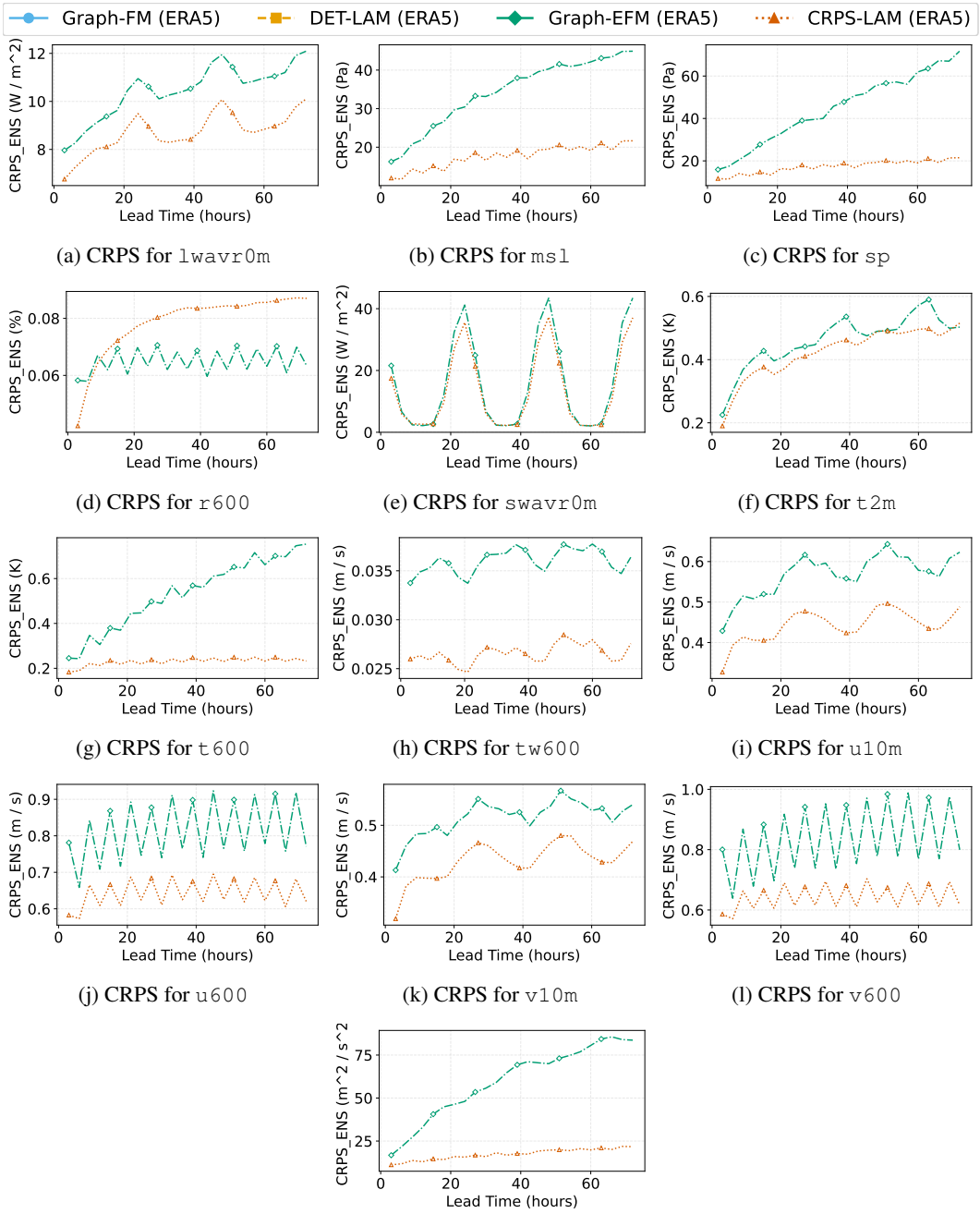
(d) RMSE for `r600` (e) RMSE for `swavr0m` (f) RMSE for `t2m`

(g) RMSE for `t600` (h) RMSE for `tw600` (i) RMSE for `u10m`

(j) RMSE for `u600` (k) RMSE for `v10m` (l) RMSE for `v600`

(m) RMSE for `z600`

Figure 15: The RMSE the DANRA dataset with ERA5 boundary conditions.



(a) CRPS for `lwavr0m` (b) CRPS for `msl` (c) CRPS for `sp`

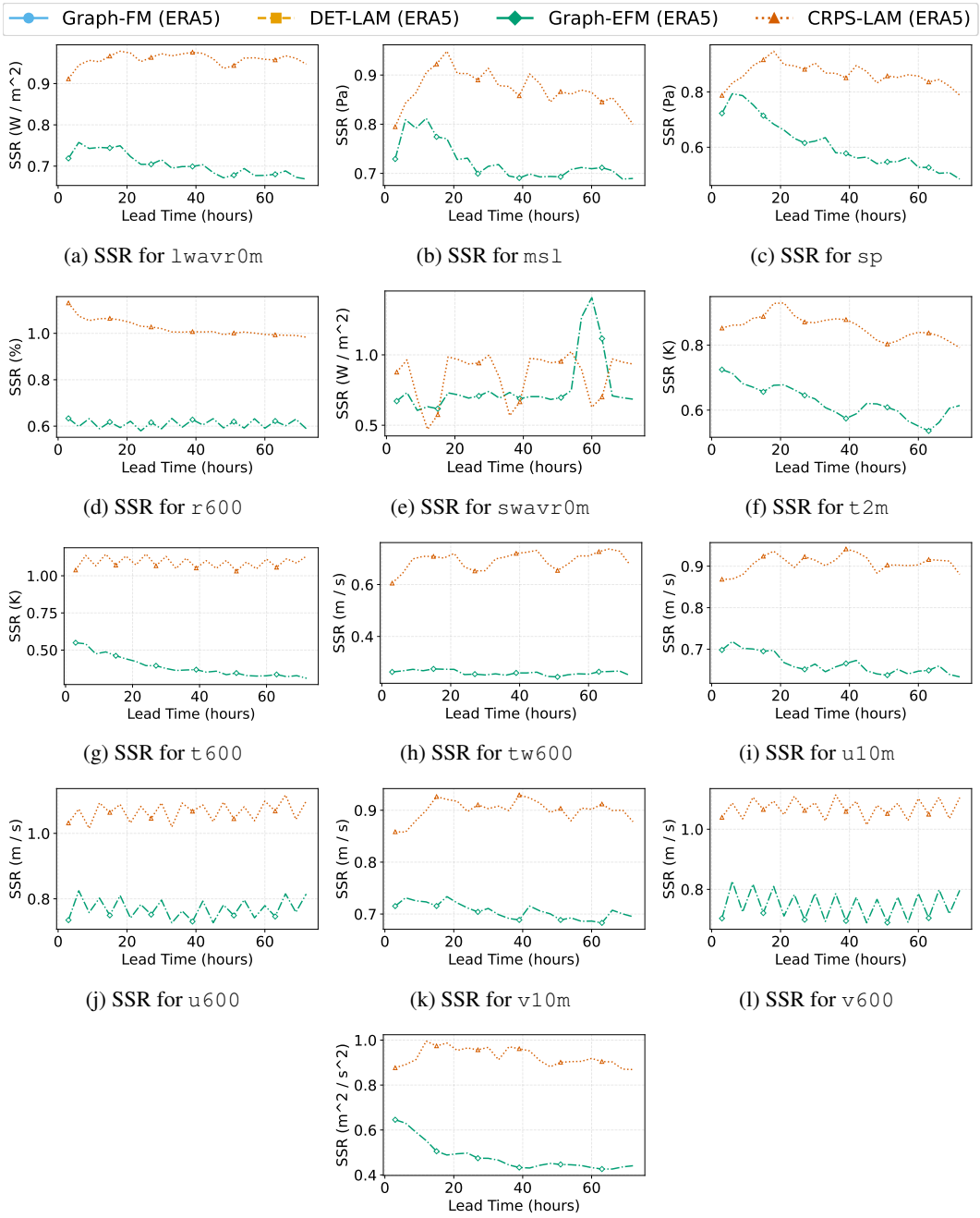
(d) CRPS for `r600` (e) CRPS for `swavr0m` (f) CRPS for `t2m`

(g) CRPS for `t600` (h) CRPS for `tw600` (i) CRPS for `u10m`

(j) CRPS for `u600` (k) CRPS for `v10m` (l) CRPS for `v600`

(m) CRPS for `z600`

Figure 16: The CRPS the DANRA dataset with ERA5 boundary conditions.



(a) SSR for `lwavr0m` (b) SSR for `msl` (c) SSR for `sp`

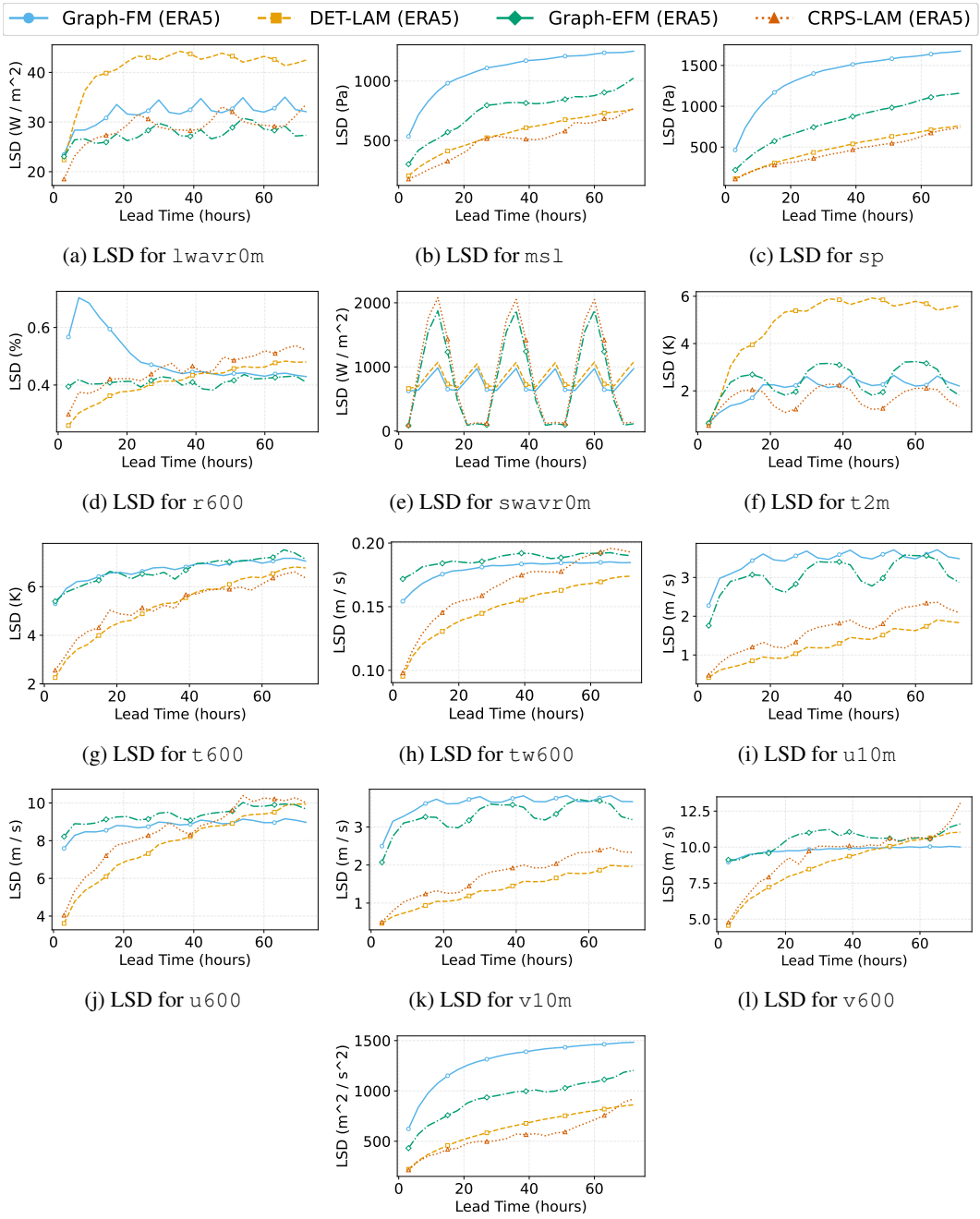
(d) SSR for `r600` (e) SSR for `swavr0m` (f) SSR for `t2m`

(g) SSR for `t600` (h) SSR for `tw600` (i) SSR for `u10m`

(j) SSR for `u600` (k) SSR for `v10m` (l) SSR for `v600`

(m) SSR for `z600`

Figure 17: The SSR the DANRA dataset with ERA5 boundary conditions.



(a) LSD for `lwavr0m` (b) LSD for `msl` (c) LSD for `sp`

(d) LSD for `r600` (e) LSD for `swavr0m` (f) LSD for `t2m`

(g) LSD for `t600` (h) LSD for `tw600` (i) LSD for `u10m`

(j) LSD for `u600` (k) LSD for `v10m` (l) LSD for `v600`

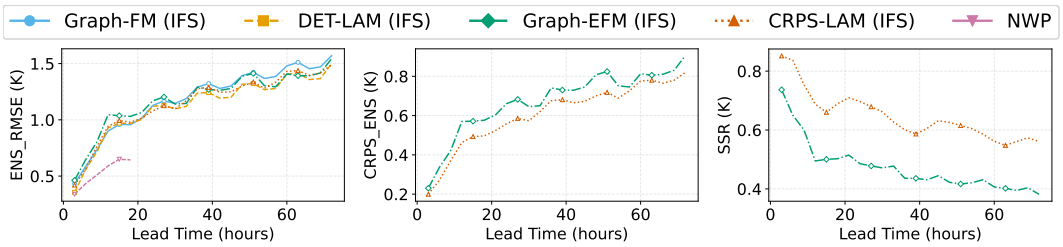
(m) LSD for `z600`

Figure 18: The LSD the DANRA dataset with ERA5 boundary conditions.

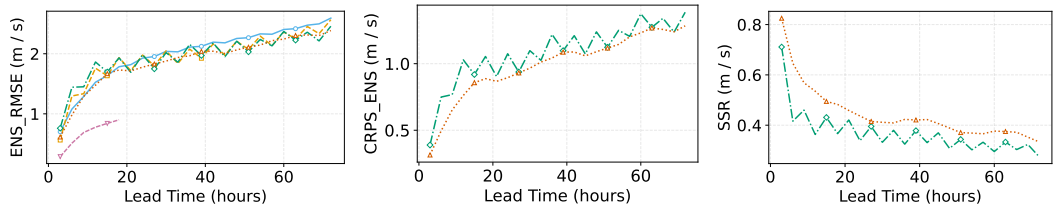
E.1 Diffusion-LAM on the DANRA dataset

We tried to include Diffusion-LAM (Larsson et al., 2025) as a probabilistic baseline on DANRA, as on MEPS (section 4.1). This section describes what we tried and why we did not include the model in the main results.

Setup. Diffusion-LAM (Larsson et al., 2025) uses a U-Net denoiser conditioned on the previous states, the forcing and the boundary. For DANRA, we replace this U-Net with the same hybrid

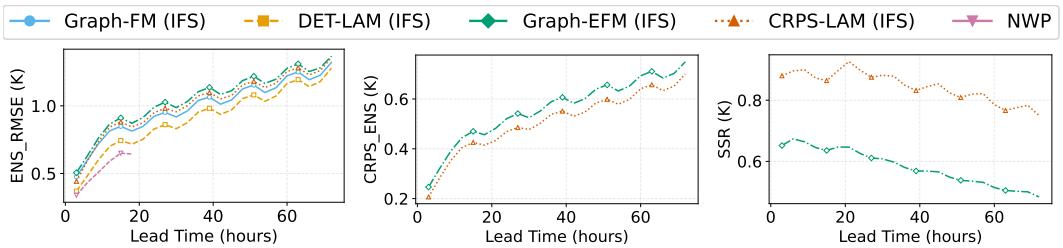


(a) RMSE for `t2m` (b) CRPS for `t2m` (c) SSR for `t2m`

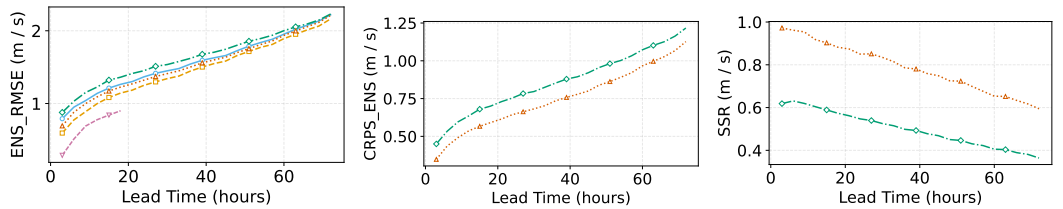


(d) RMSE for `u10m` (e) CRPS for `u10m` (f) SSR for `u10m`

Figure 19: The RMSE, CRPS, and SSR for 2 m temperature and the u wind component at 10 m on the DANRA dataset, evaluated with IFS boundary conditions for models trained only with ERA5 boundaries (before finetuning on IFS).



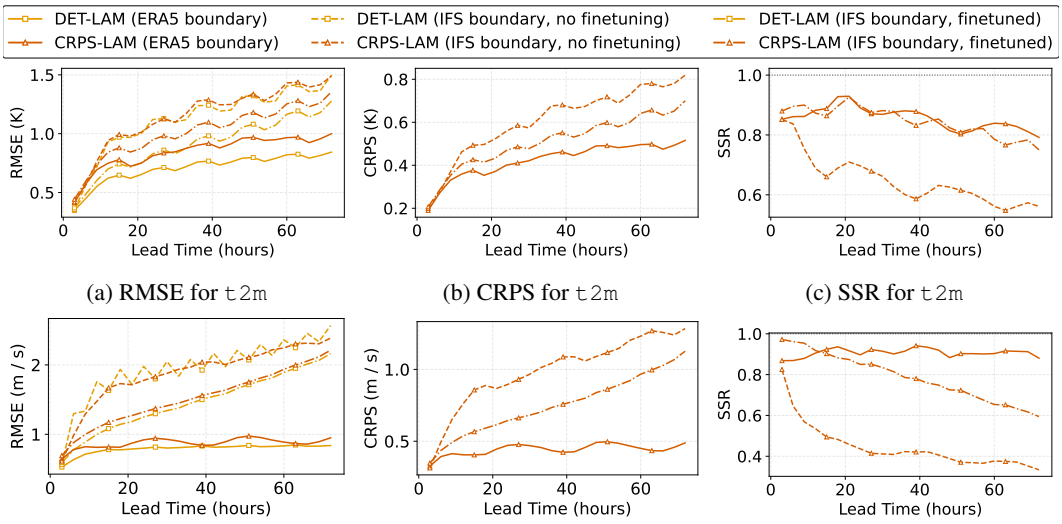
(a) RMSE for `t2m` (b) CRPS for `t2m` (c) SSR for `t2m`



(d) RMSE for `u10m` (e) CRPS for `u10m` (f) SSR for `u10m`

Figure 20: The RMSE, CRPS, and SSR for 2 m temperature and the u wind component at 10 m on the DANRA dataset with IFS boundary for the models finetuned with IFS boundary conditions.

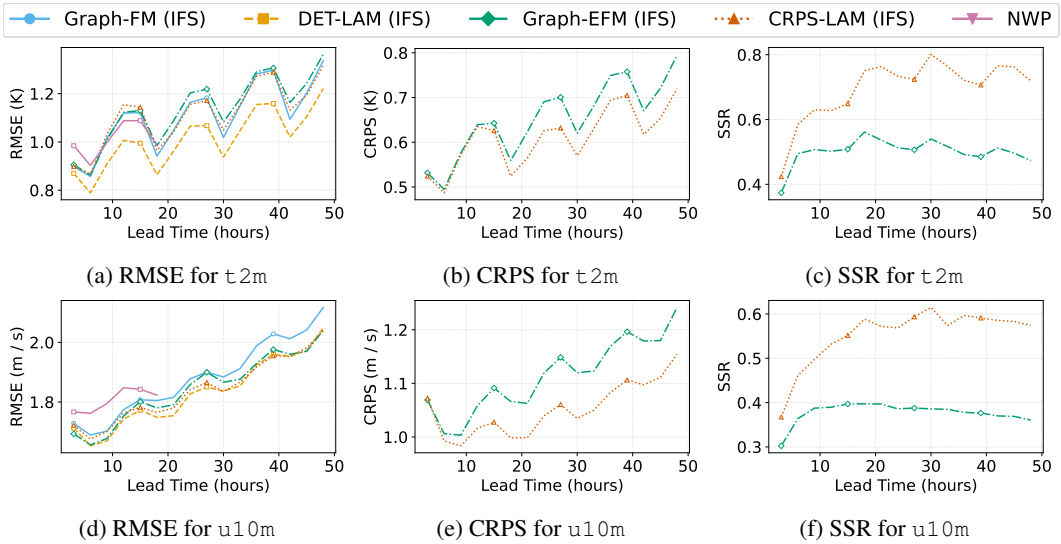
backbone as DET-LAM and CRPS-LAM, so that it can handle the boundary on a different grid and the models differ only in the training objective and sampling procedure. We increased the size of the backbone until training reached the 80 GB memory limit (section B). The largest model has 277M parameters and uses 79 GB of GPU memory during training (table 1), compared to 64M parameters for the model used on MEPS. We trained the model for single-step prediction with ERA5 boundaries for up to 100 epochs, compared to 80 single-step epochs for the other models. As the model never produced single step predictions without significant artifacts, we did not continue to the autoregressive or IFS finetuning stages, which is why its training cost in table 1 is lower than that of the other models. Diffusion-LAM was the only model tuned with full-scale training runs (section B.1): we ran multiple runs, varying, e.g., learning rate, noise schedule, and loss weighting over variables.



(a) RMSE for `t2m` (b) CRPS for `t2m` (c) SSR for `t2m`

(d) RMSE for `u10m` (e) CRPS for `u10m` (f) SSR for `u10m`

Figure 21: RMSE, CRPS, and SSR for 2 m temperature and the 10 m u-wind component on the DANRA dataset. Results are shown for DET-LAM and CRPS-LAM under different boundary-condition configurations, both with and without fine-tuning.



(a) RMSE for `t2m` (b) CRPS for `t2m` (c) SSR for `t2m`

(d) RMSE for `u10m` (e) CRPS for `u10m` (f) SSR for `u10m`

Figure 22: Results on the DANRA dataset with IFS boundary for the models finetuned with IFS boundary conditions compared against in-situ observations.

Results. Training converged in all runs, but further tuning did not give competitive forecasts. The main problem was in sampling: the generated fields contained a large amount of noise that was not removed during the reverse diffusion process. Increasing the number of solver steps from 39 to 200 did not resolve this issue or lead to meaningful improvements in forecast quality. When the noisy states were fed back as input, errors grew quickly and the autoregressive rollouts became unstable, with errors far higher than those of the other models.

Reduced-dimensional experiment. Our hypothesis is that the conditional distribution the model has to learn on DANRA is too high-dimensional for this model size. Compared to MEPS, DANRA has more variables (55 compared to 17 per grid point) and a finer grid (589×789 at 2.5 km, compared to 238×268 at 10 km), so the model has to generate more than 20 times as many values per time

step. To test this, we trained the same model to predict only three DANRA variables, keeping the other variables as inputs only. The denoising then worked much better, and the generated fields for these variables no longer had the residual noise. This suggests that the model has enough capacity for a low-dimensional target but not for the full DANRA state, and that a working model for all 55 variables would need to be substantially larger.

Why we did not scale further. A larger model does not fit in the 80 GB memory we required all models to be trainable on (section B). It could be trained with model parallelism, but a larger denoiser would also make sampling slower. Even at the current size, a 20-member 72 h ensemble forecast takes more than 5 h on a single A100 GPU, compared to about 8 min for CRPS-LAM (table 1). This sampling cost is one of the main limitations of diffusion-based forecasting, and is a large part of the motivation for single step prediction models such as CRPS-LAM.

E.2 FORECASTS ON THE DANRA DATASET

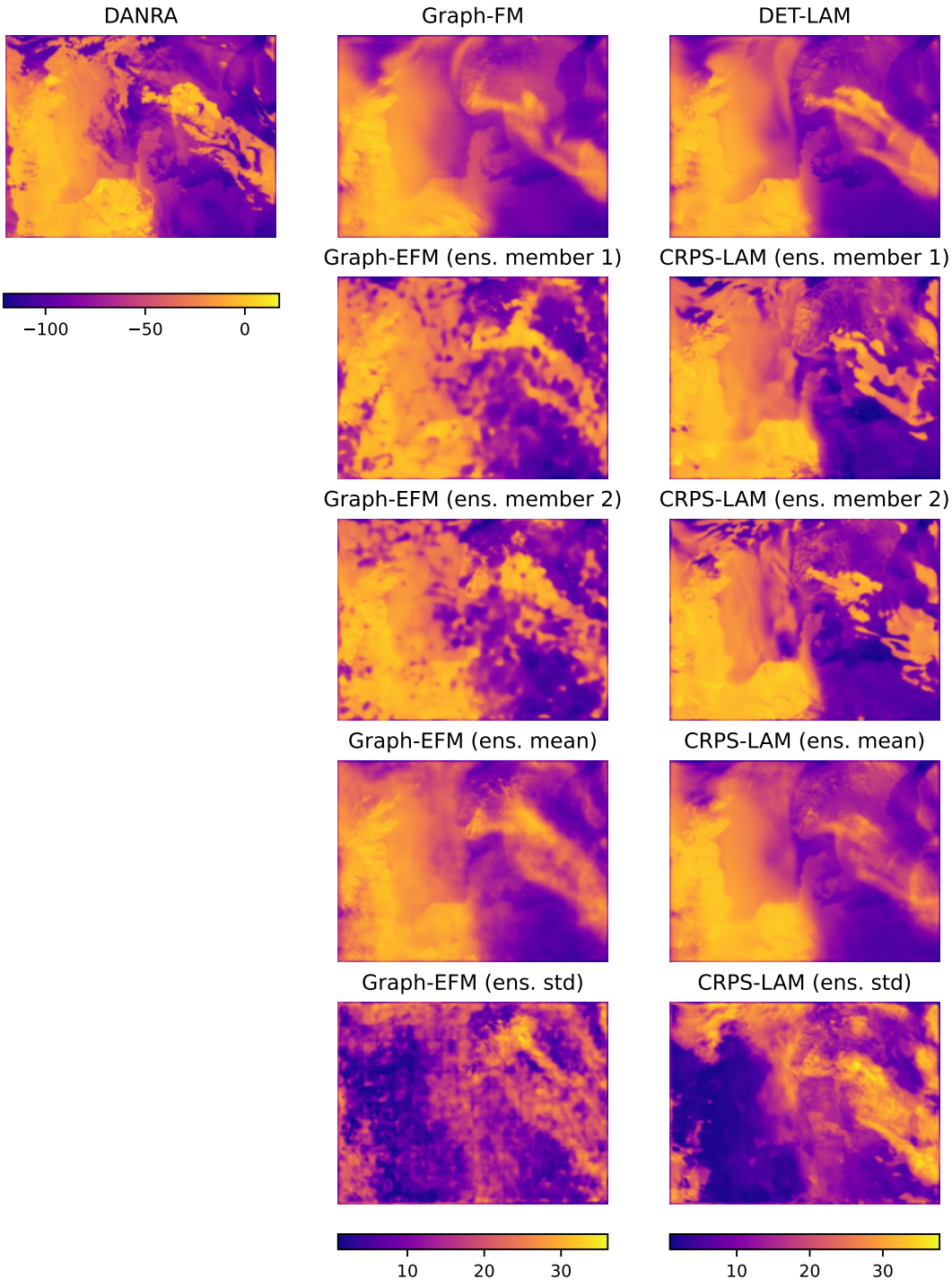


Figure 23: Forecasts of net long wave radiation flux at 72 h lead time.

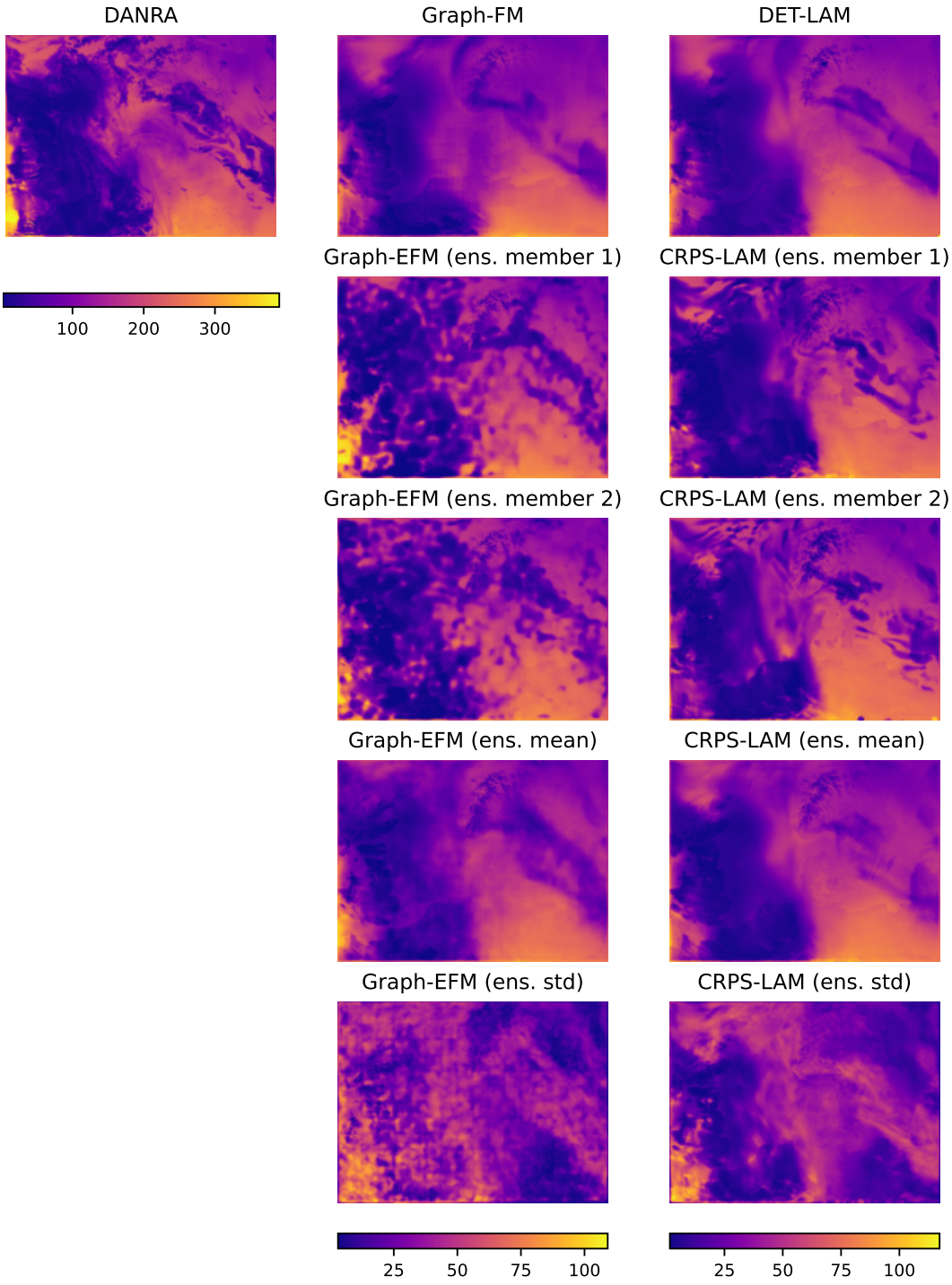


Figure 24: Forecasts of net short wave radiation flux at 72 h lead time.

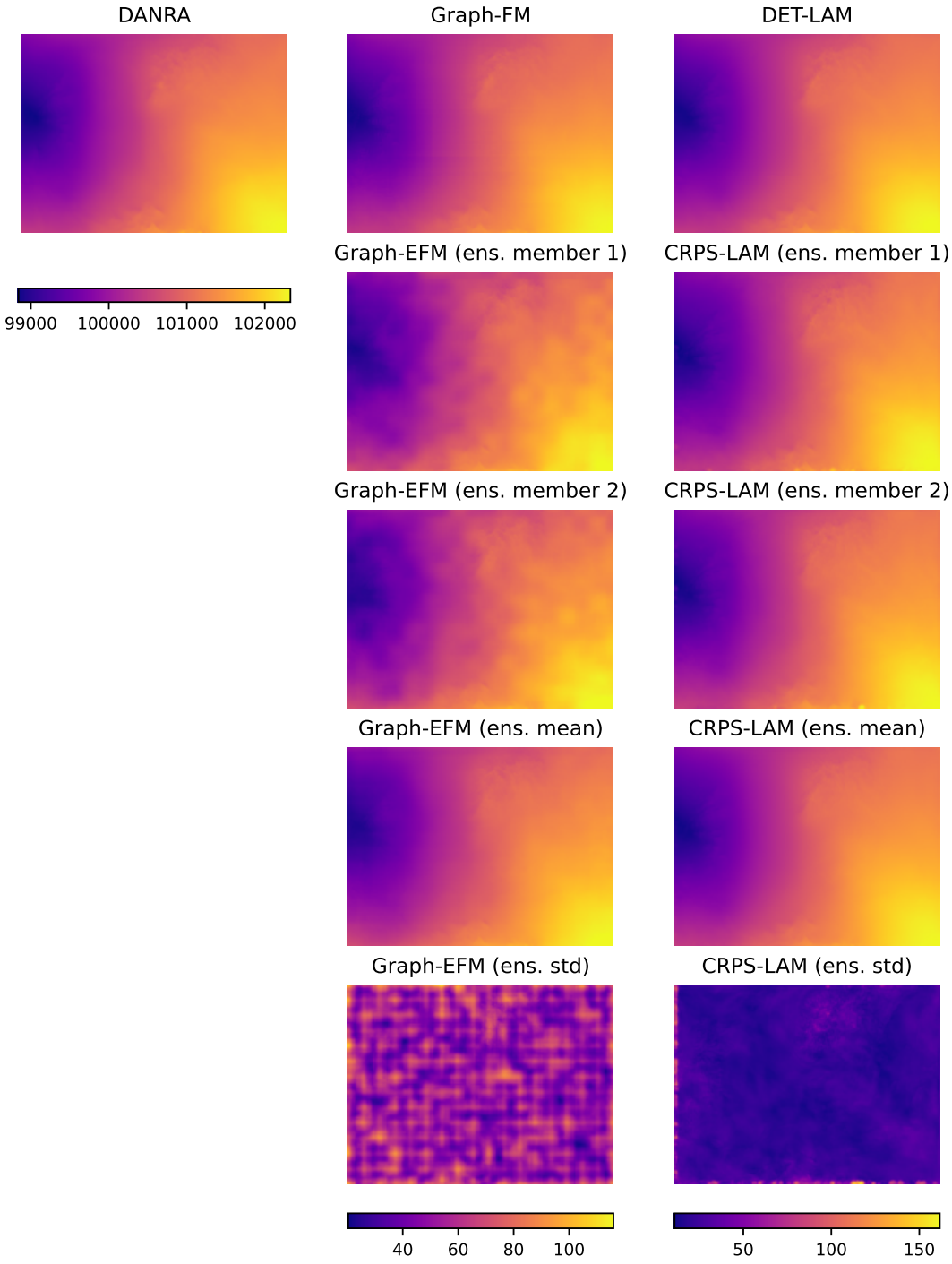


Figure 25: Forecasts of mean sea level pressure at 72 h lead time.

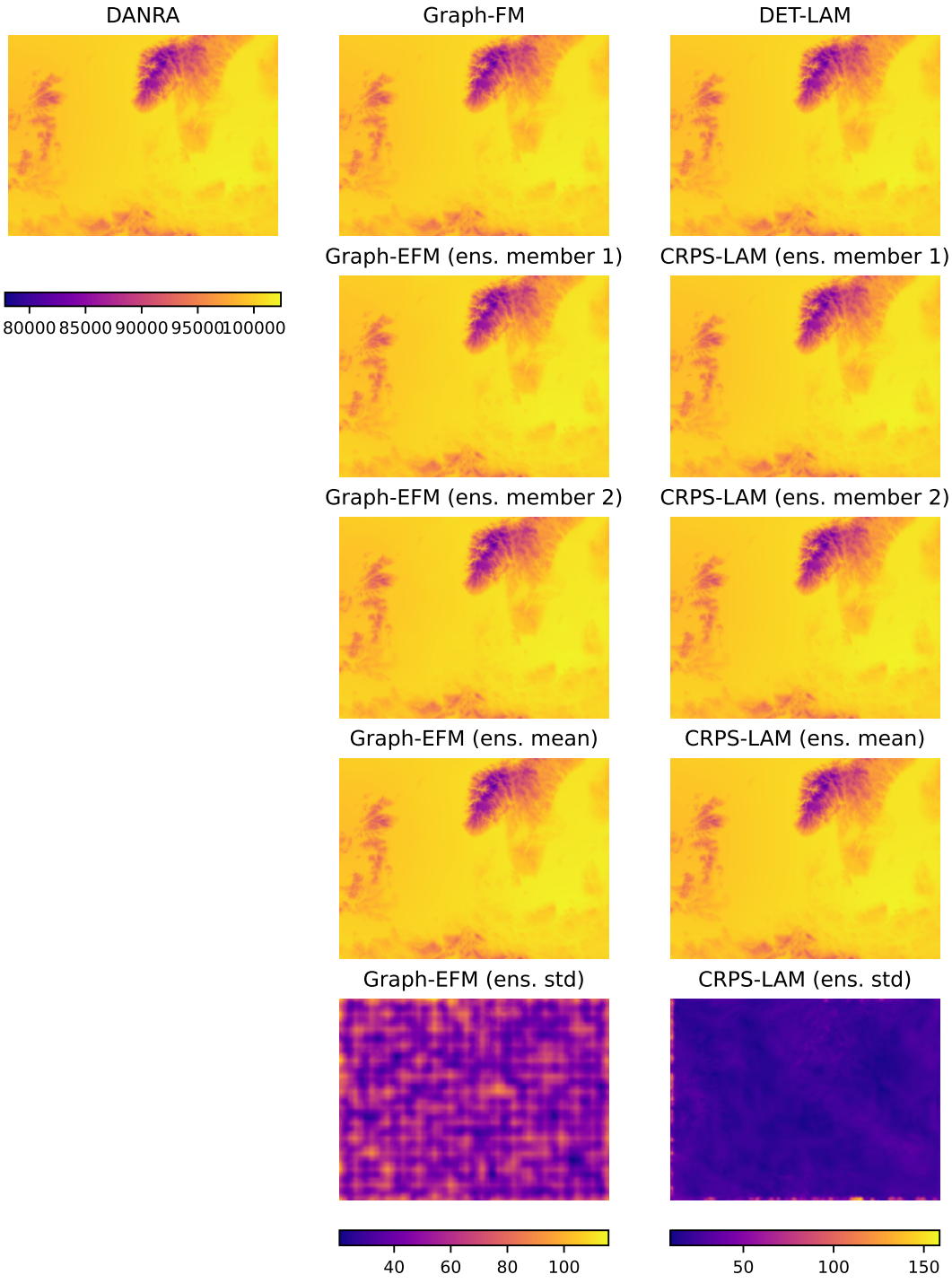


Figure 26: Forecasts of surface pressure at 72 h lead time.

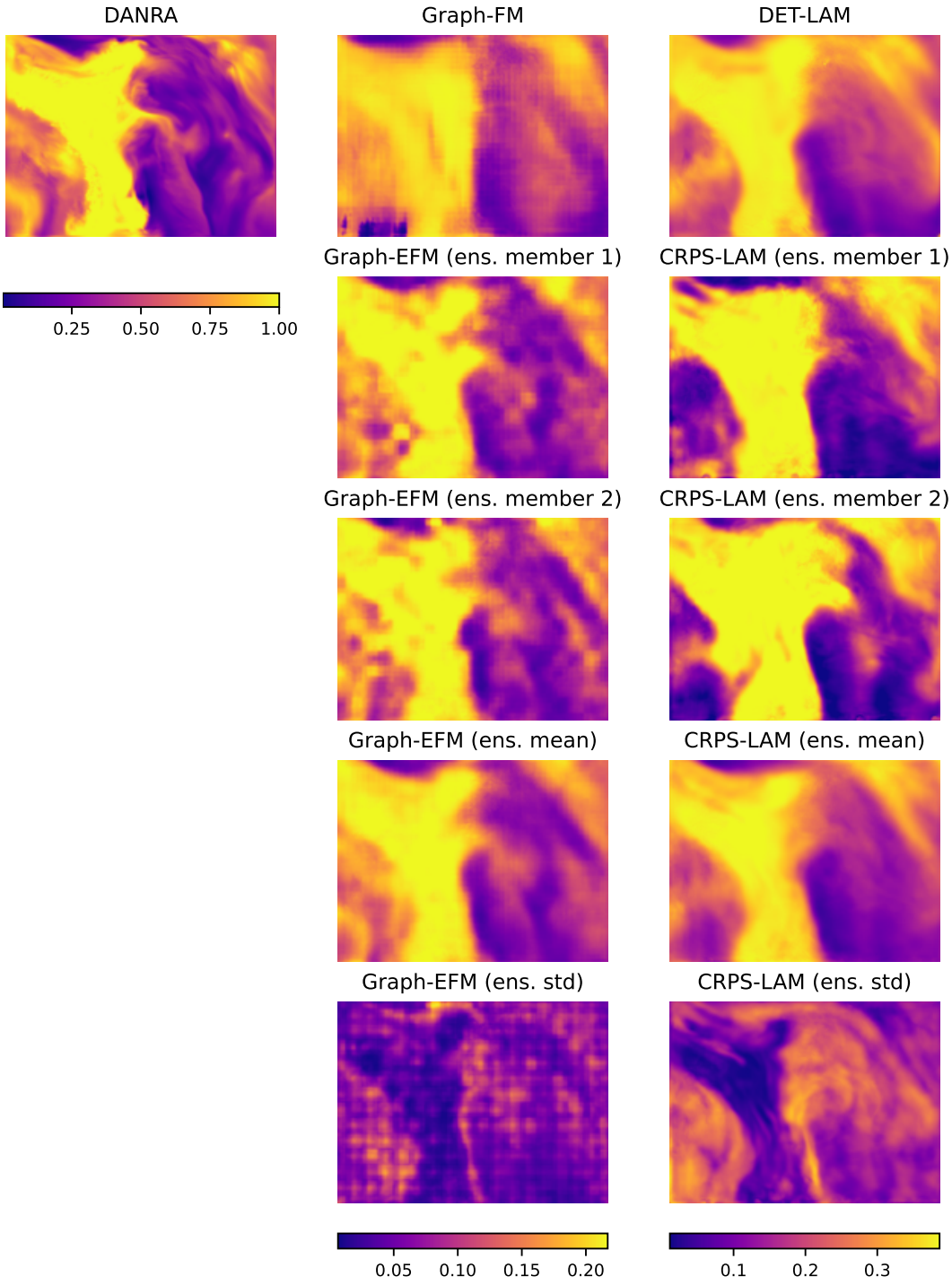


Figure 27: Forecasts of relative humidity at 600 hPa at 72 h lead time.

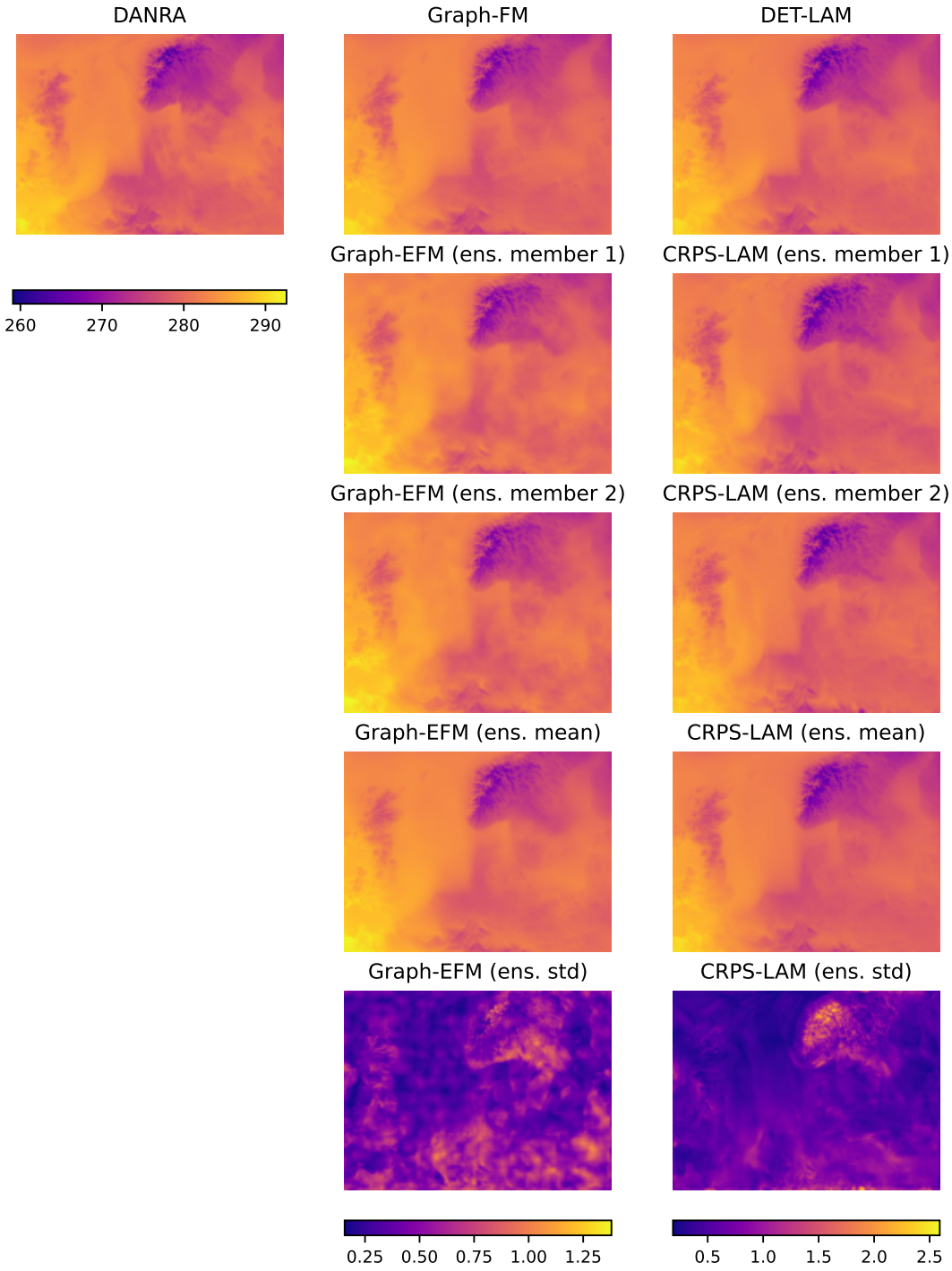


Figure 28: Forecasts of 2 m temperature at 72 h lead time.

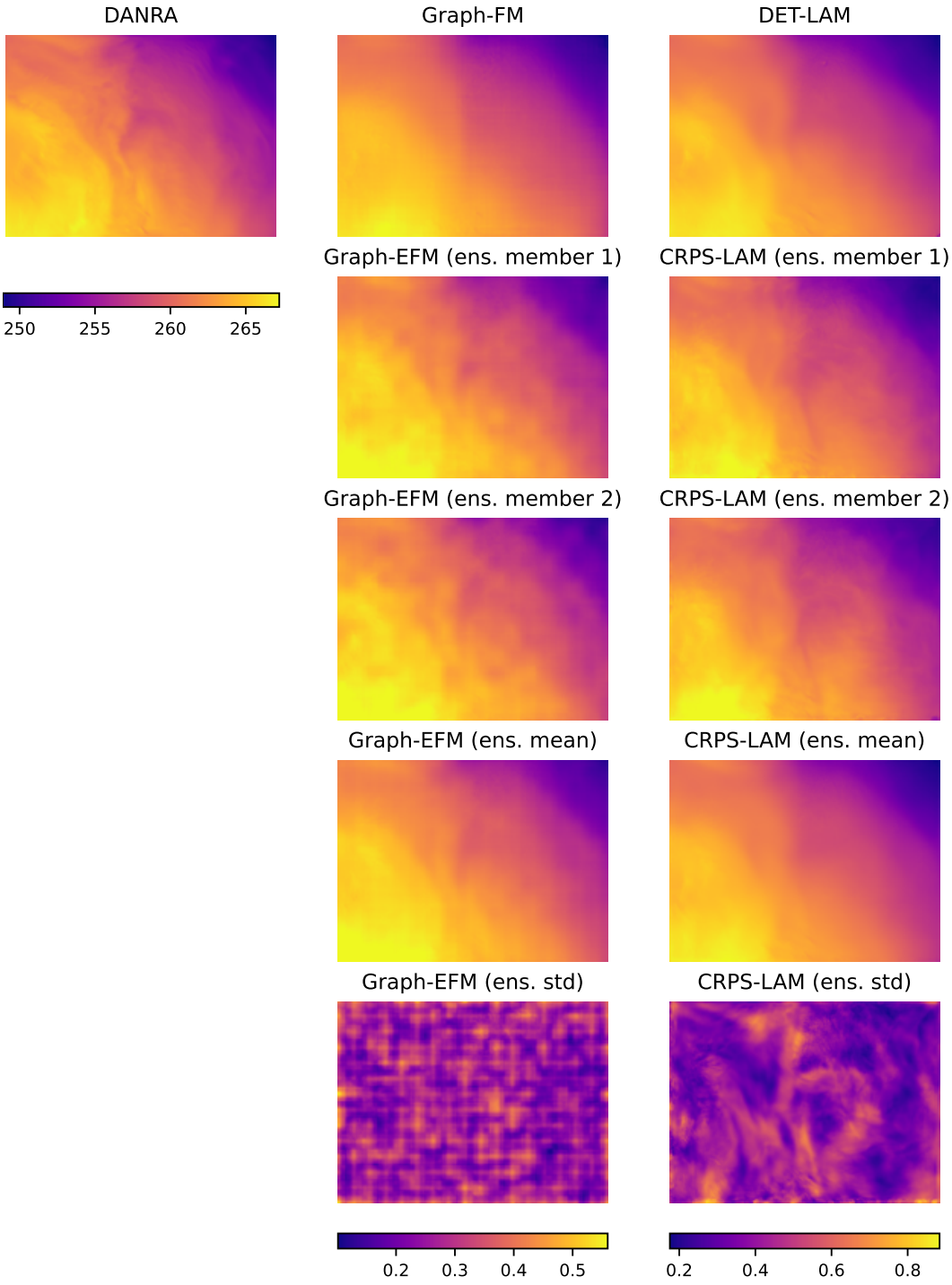


Figure 29: Forecasts of temperature at 600 hPa at 72 h lead time.

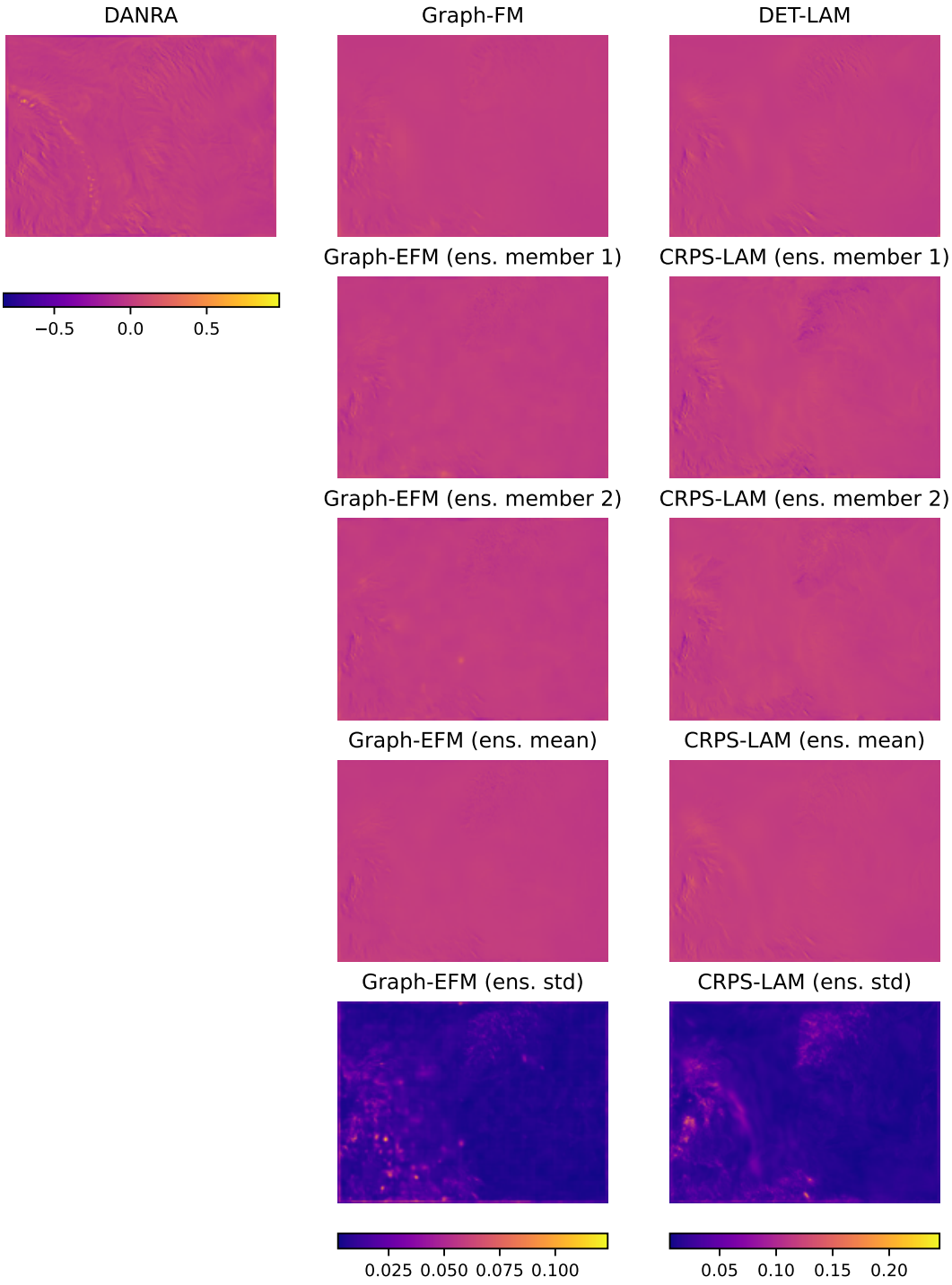


Figure 30: Forecasts of vertical velocity at 600 hPa at 72 h lead time.

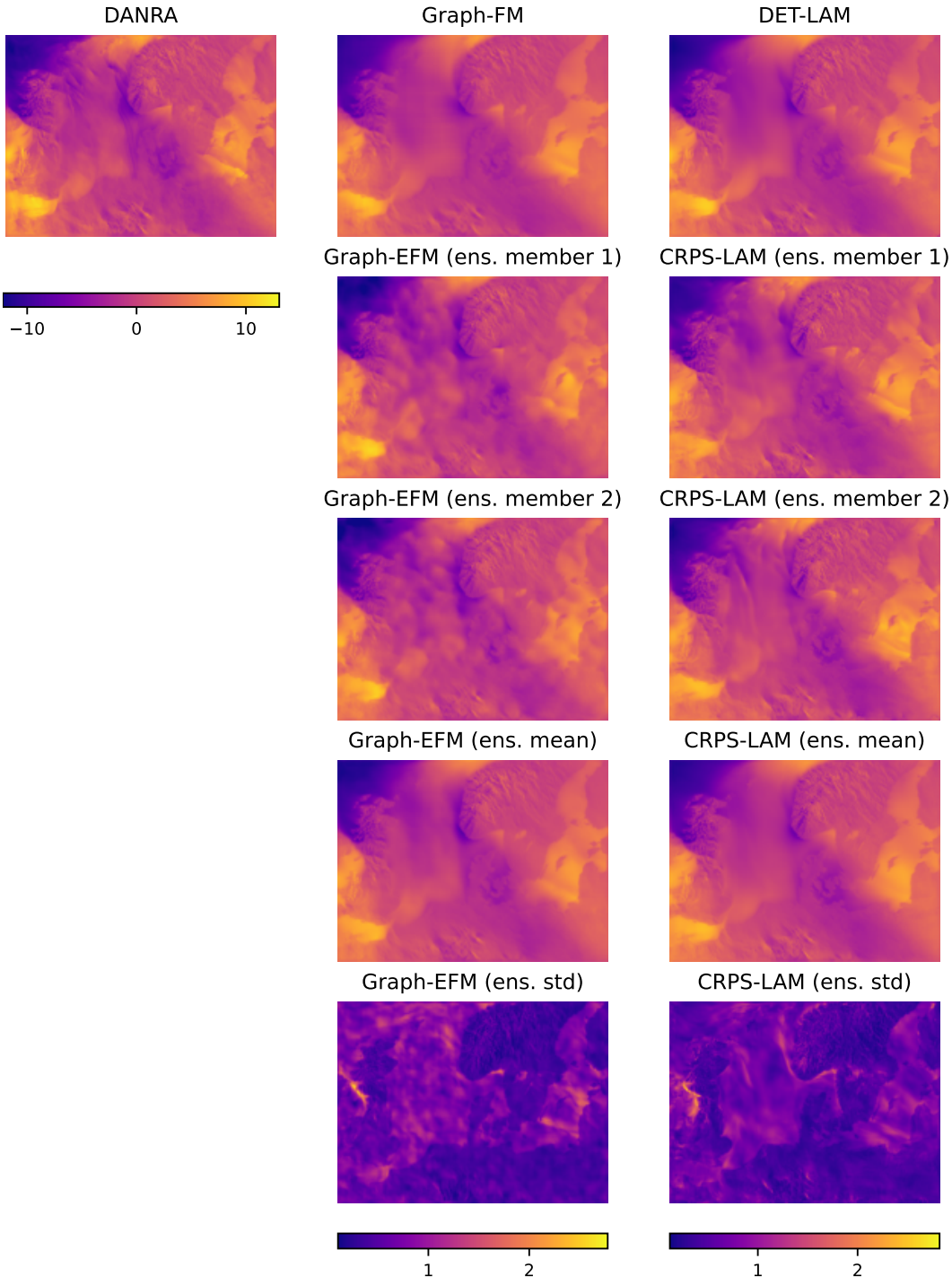


Figure 31: Forecasts of u wind component at 10 m at 72 h lead time.

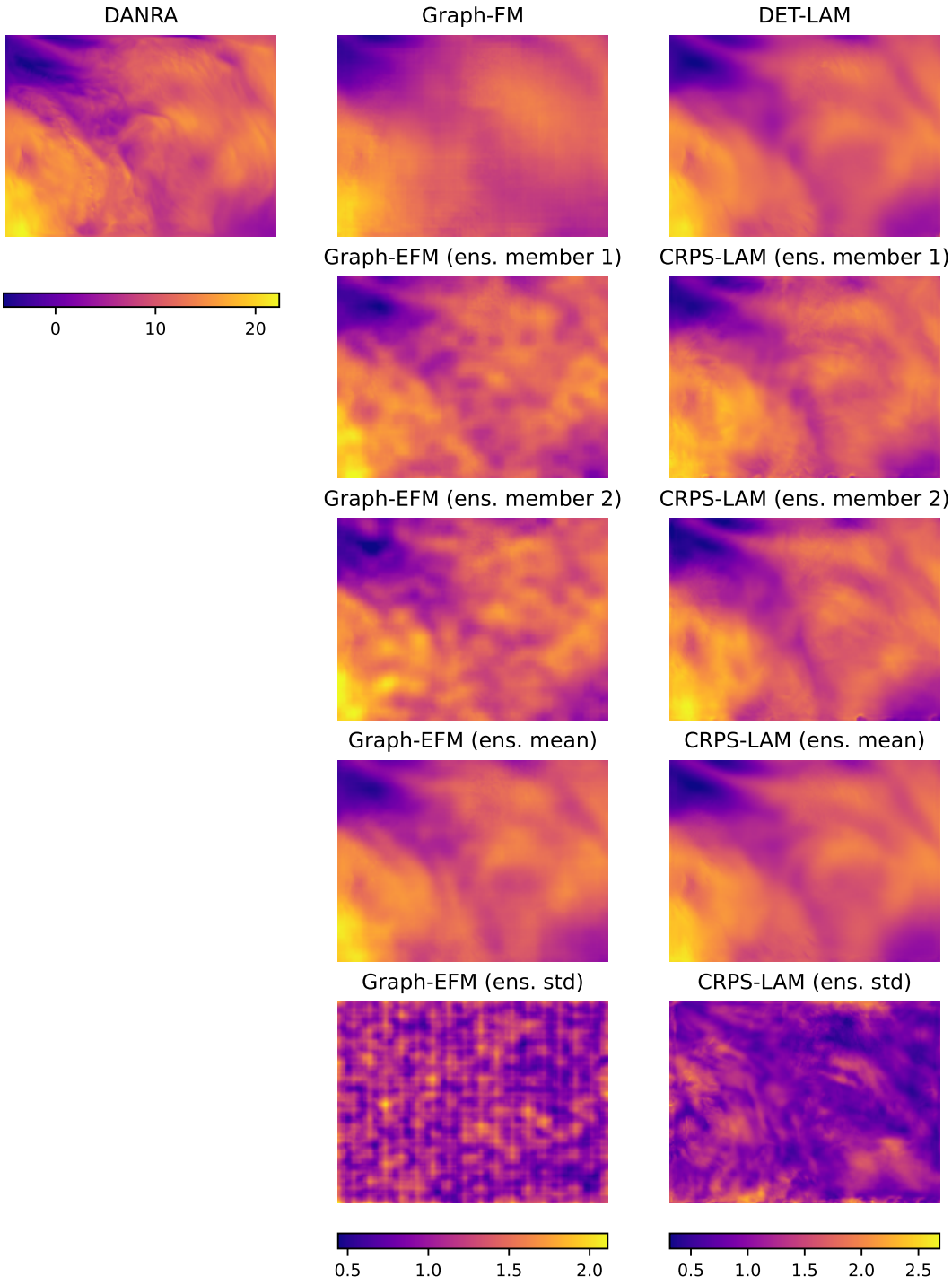


Figure 32: Forecasts of u wind component at 600 hPa at 72 h lead time.

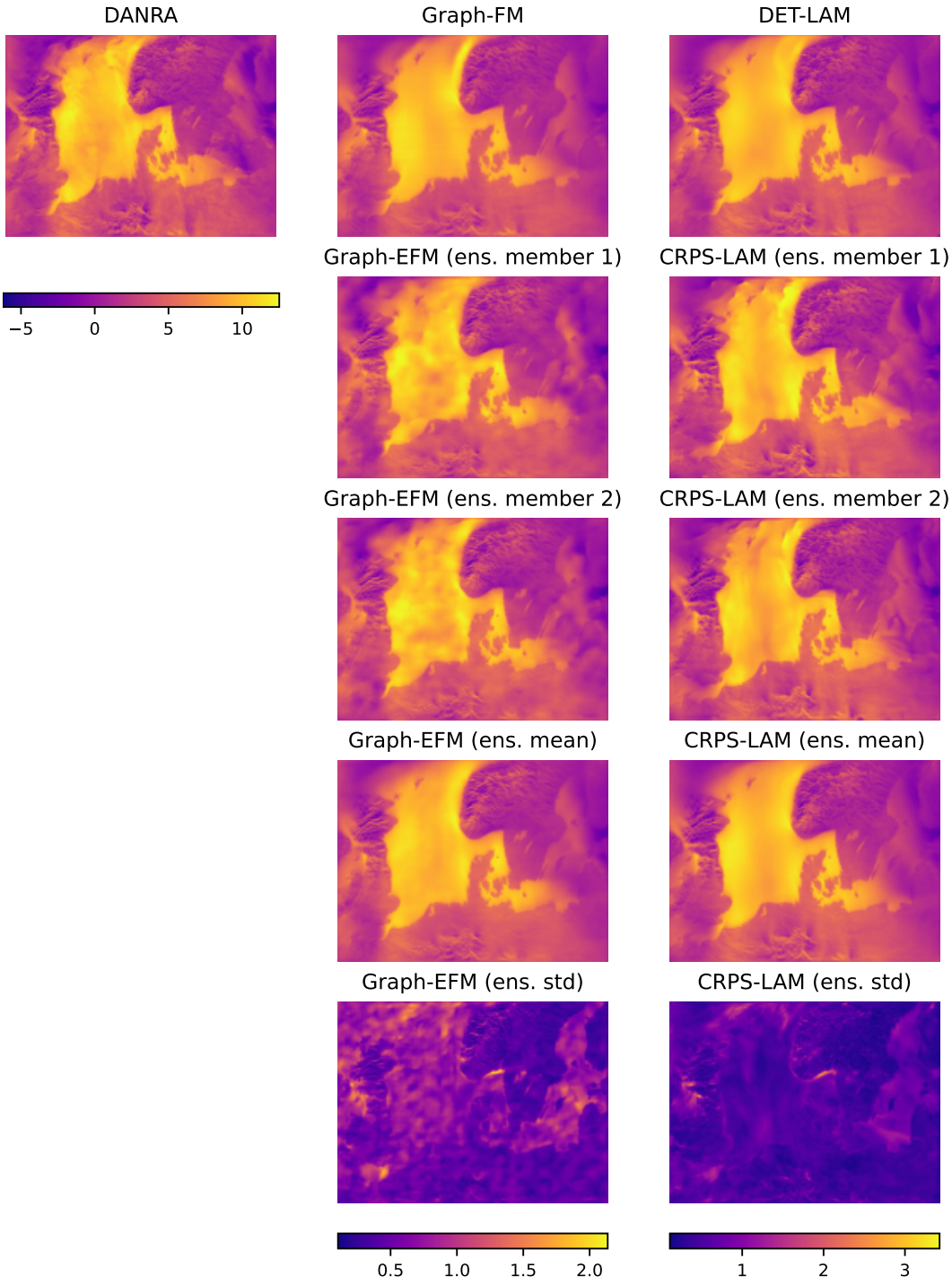


Figure 33: Forecasts of v wind component at 10 m at 72 h lead time.

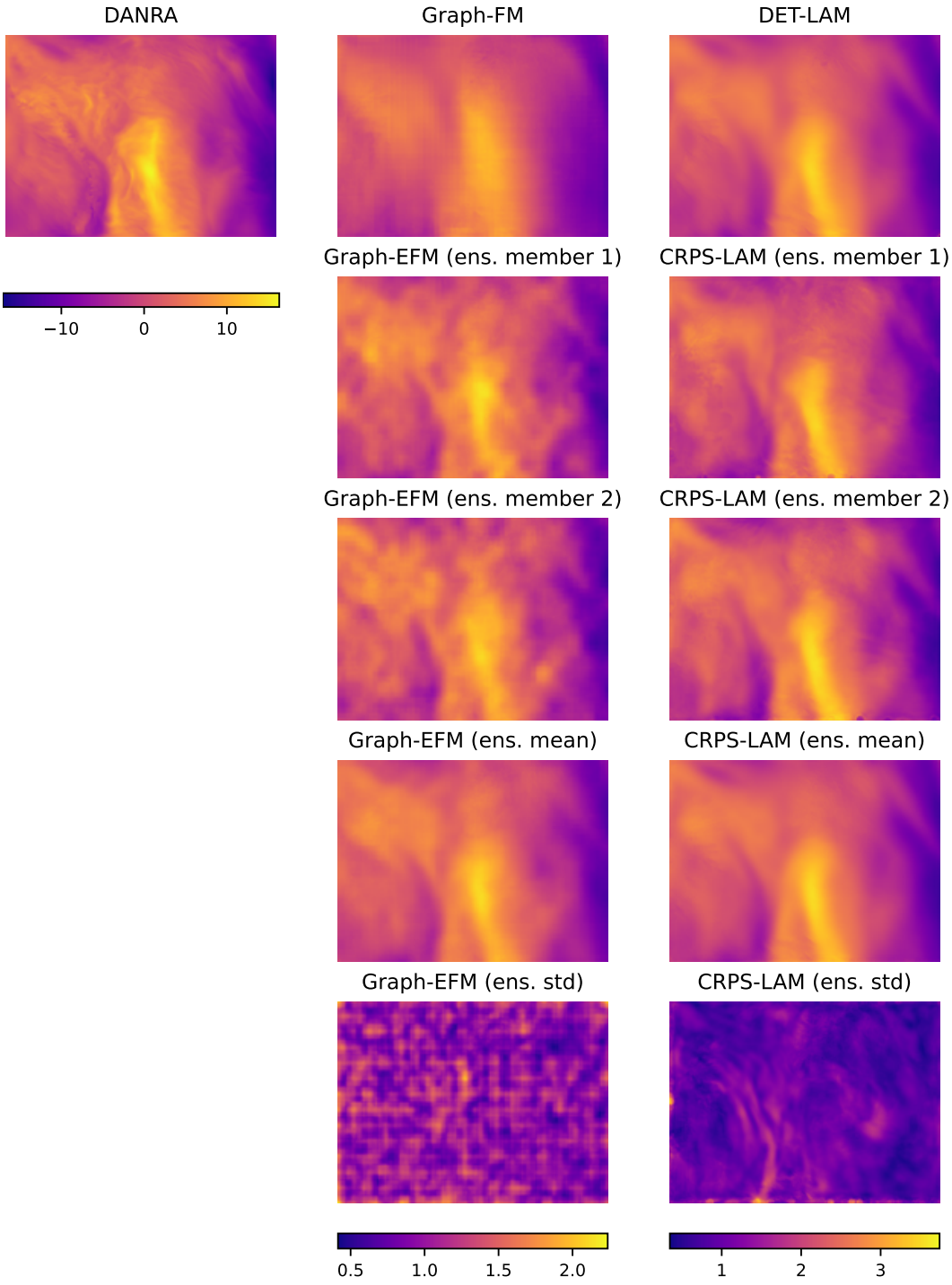


Figure 34: Forecasts of v wind component at 600 hPa at 72 h lead time.

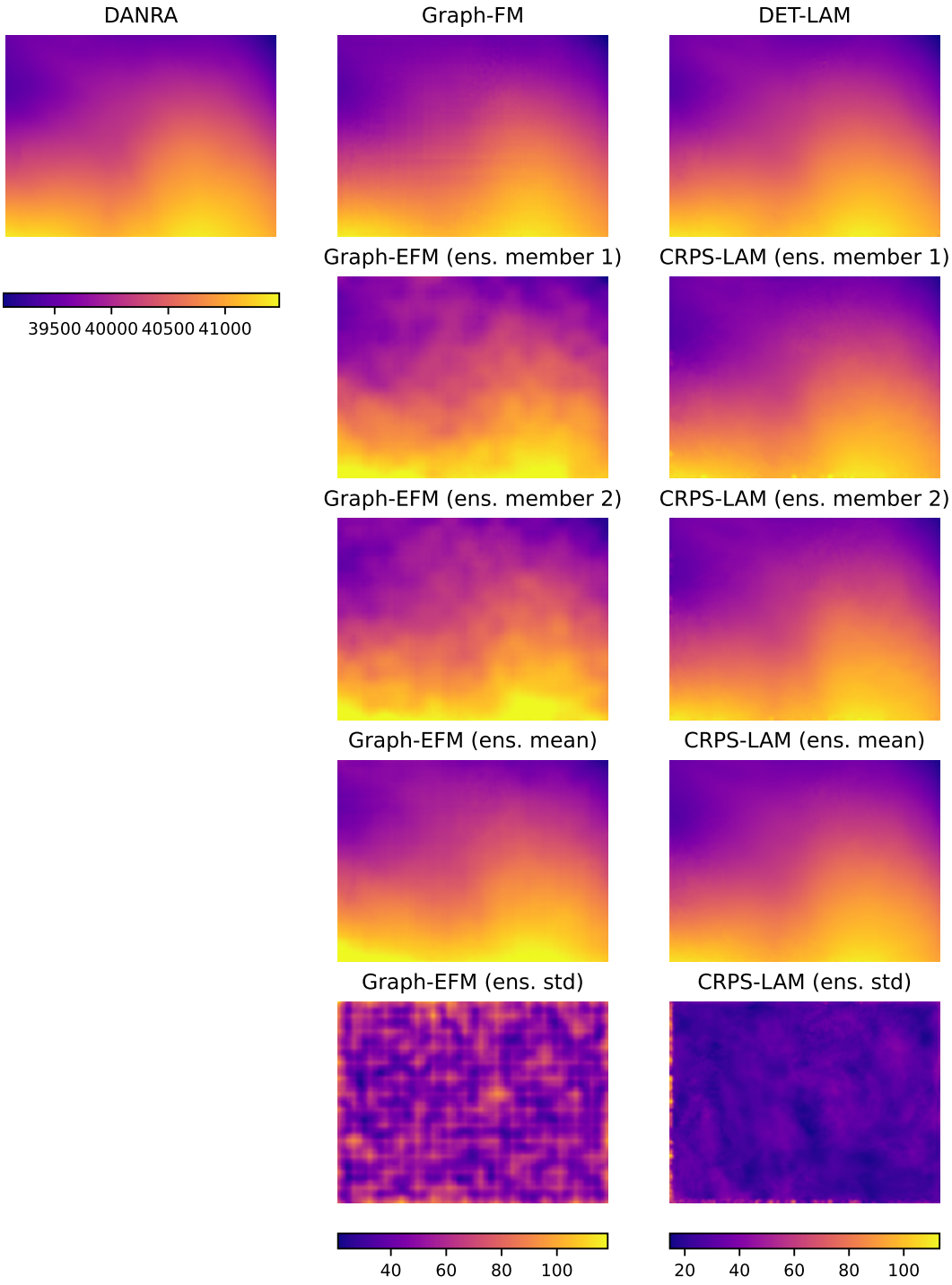


Figure 35: Forecasts of geopotential at 600 hPa at 72 h lead time.

F Additional Experiments on the MEPS dataset

Here we present details of the experiments[2] in a simplified setting using the MEPS dataset[3], including a comparison with a diffusion-based baseline (Larsson et al., 2025) and Graph-EFM. The problem definition is slightly different compared to the DANRA setting in section 4.2 as the boundary comes from the ground truth on the same grid and is not overlapping with the interior. An overview of the limited-area MEPS forecasting setup is illustrated in fig. 4.

Since the boundary and interior are on a regular grid, the grid-to-mesh encoders reduce to an MLP (with different weights for the interior and boundary), since there is no need for a GNN to map an irregular grid to the mesh.

F.1 The MEPS Dataset

Since the training objective is based on forecasts rather than actual observations, the objective is to develop an emulator model for MEPS. The 6069 forecasts in the dataset are from the time period April 2021 to March 2023. For simplicity and consistency with Oskarsson et al. (2024) we use the same training, validation and test split. We use forecasts from April 2021 to June 2022 for training (2713 samples) and validation (678 samples), and forecasts from July 2022 to March 2023 for testing (2678 samples).

Table 11: Variables in the MEPS dataset. *Level 65 in the MEPS system is approximately 12.5 m over the ground (Müller et al., 2017).

Description	Abbreviation	Unit	Residual standard deviation
Net longwave solar radiation flux at the surface	nlwrs	W/m^2	0.0583
Net shortwave solar radiation flux at the surface	nswrs	W/m^2	0.0583
Atmospheric pressure at ground level	pres_0g	Pa	0.6399
Atmospheric pressure at sea level	pres_0s	Pa	0.7608
Relative humidity at 2 m	r_2	[0, 1]	0.5534
Relative humidity at level 65*	r_65	[0, 1]	0.5371
Temperature at 2 m	t_2	K	0.2197
Temperature at level 65*	t_65	K	0.1950
Temperature at 500 hPa	t_500	K	0.1319
Temperature at 850 hPa	t_850	K	0.1294
u-component of wind at level 65*	u_65	m/s	0.3885
u-component of wind at 850 hPa	u_850	m/s	0.3530
v-component of wind at level 65*	v_65	m/s	0.3815
v-component of wind at 850 hPa	v_850	m/s	0.3861
Water vapor for the full integrated column	wvint_0	kg/m^2	0.2473
Geopotential at 1000 hPa	z_1000	m^2/s^2	0.1202
Geopotential at 500 hPa	z_500	m^2/s^2	0.0720

F.2 Experiment Details

All forecast trajectories are initialized from the ground-truth states and generated autoregressively up to a lead time of 57 h, with 25 ensemble members produced for each model. The model is trained to perform 3 h forecasts (see fig. 4) and is rolled out autoregressively to produce longer forecasts. CRPS-LAM can generate 57 h forecasts in approximately 0.5 s per ensemble member on a single A100 GPU. Moreover, an arbitrary number of ensemble members can be sampled in parallel through batched inference on one or multiple GPUs. This corresponds to a sampling speed comparable to that of Graph-EFM or a deterministic model (on a per-member basis) and approximately 39× faster than Diffusion-LAM, depending on the number of solver steps that are used.

[2]The code and implementation details will be made publicly available upon acceptance of this paper.

[3]The MEPS dataset is openly available at `https://nextcloud.liu.se/s/meps` under a CC BY 4.0 license.

Table 12: Forcing features in the MEPS dataset.

Description	Abbreviation	Unit
Solar radiation flux at the top of the atmosphere	toa	W/m^2
Fraction of open water at the surface	water	[0, 1]
Sine-encoded time of day	sin_tod	[0, 1]
Cosine-encoded time of day	cos_tod	[0, 1]
Sine-encoded time of year	sin_toy	[0, 1]
Cosine-encoded time of year	cos_toy	[0, 1]

Table 13: Static features for each grid position in the MEPS dataset.

Description	Abbreviation	Unit
Topology (geopotential at the surface)	topology	m^2/s^2
x-coordinate in the MEPS projection	x_coord	[0, 1]
y-coordinate in the MEPS projection	y_coord	[0, 1]
Boundary mask (indicating which pixels belong to the border)	border_mask	0/1
Interior mask (indicating which pixels belong to the interior)	interior_mask	0/1

An overview schematic of our architecture of our backbone model is shown in fig. 36 with the details of each U-Net shown in fig. 37. Note that the MLP blocks are different from the U-Net blocks and consist of a two-layer MLP followed by a conditional layer normalization, where the latent variable z is included as an additional conditioning input.

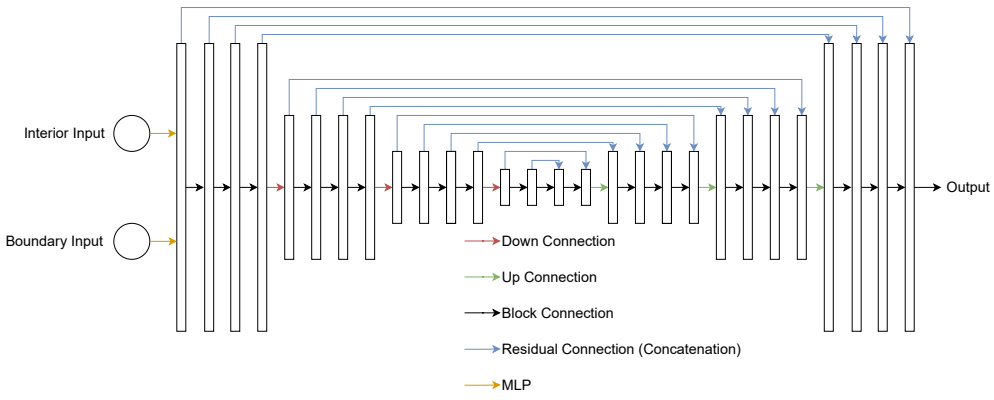


Figure 36: An overview of the backbone model.

The training procedure for CRPS-LAM is summarized in table 14. We begin by training the model on single-step forecasts until convergence, after which we extend the training to two-step autoregressive rollouts. In this setup, we did not observe any additional benefits from training with more autoregressive steps.

During the initial training phase, we sometimes observe instability when training with the unbiased CRPS estimator. Similar behavior has been reported by Lang et al. (2026), who proposed a modified training objective to address it, and by Bonev et al. (2025), who mitigated the issue by using a biased CRPS estimator with a large ensemble early in training, later transitioning to a smaller ensemble and the unbiased fair CRPS estimator.

Prior to introducing autoregressive training with two-step trajectories, we observed artifacts in a small subset of ensemble members at longer lead times. However, these artifacts disappeared once

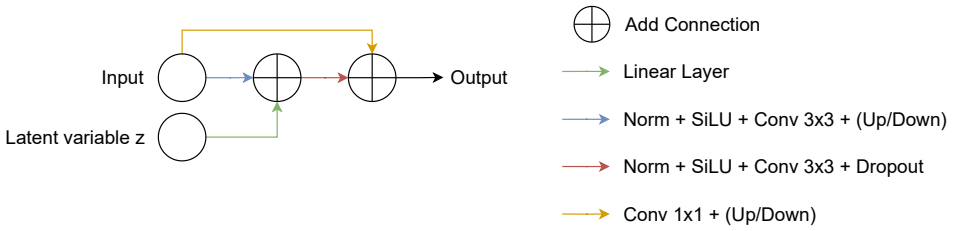


Figure 37: A description of each U-Net block.

Table 14: Training schedule.

Epochs	Learning Rate	Autoregressive Steps
600	0.001	1
400	0.0001	1
200	0.00001	2

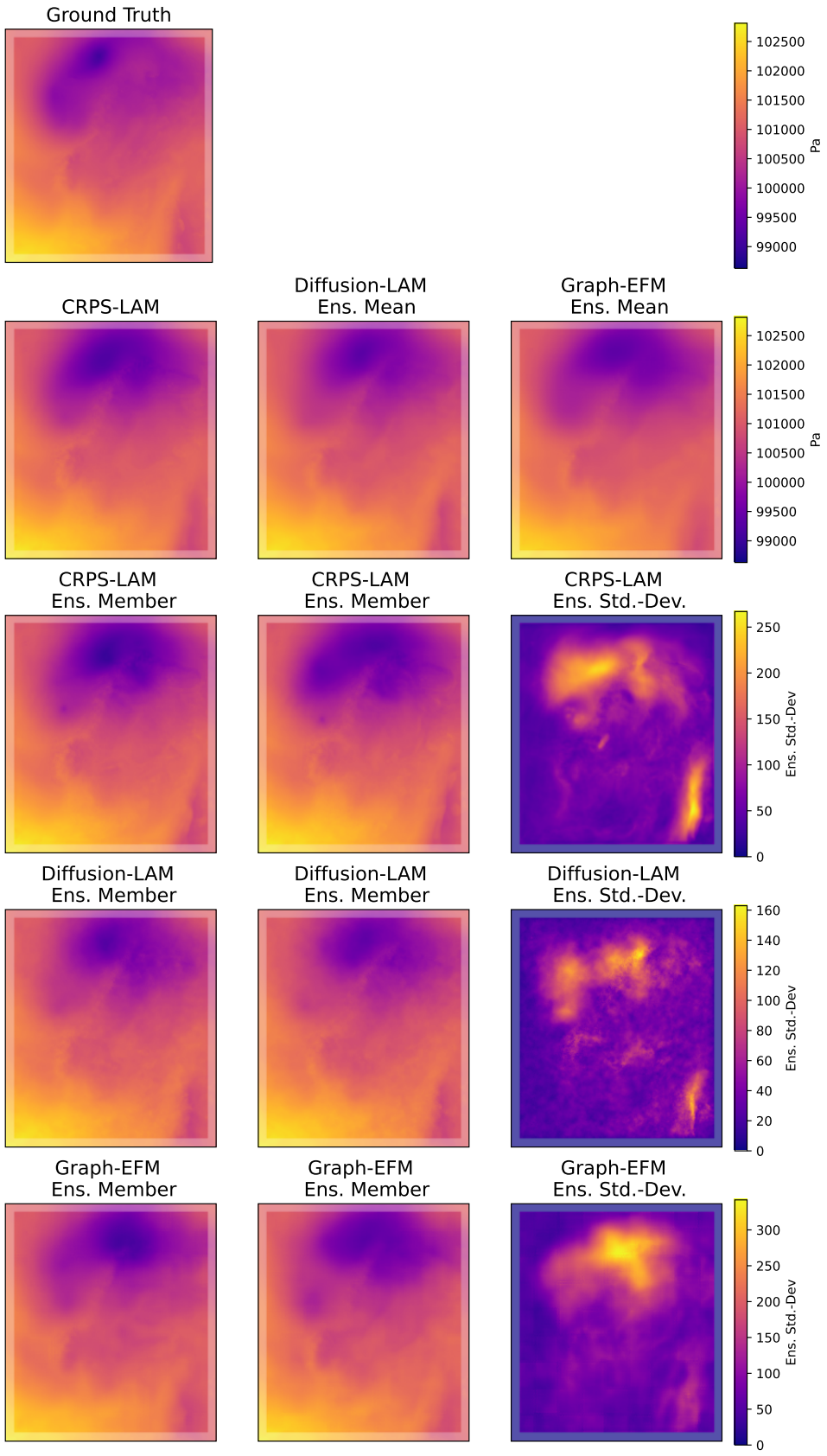
the model was trained with autoregressive forecasting steps, suggesting that training with multiple autoregressive steps helps stabilize the usage of latent variables at longer lead times.

For the MEPS experiments we set all manual weights $\omega_d = 1$, in contrast to Graph-EFM and Diffusion-LAM, as we found variable- or level-specific weighting unnecessary to achieve competitive results. For DANRA we use the weights of Adamov et al. (2025) (section B).

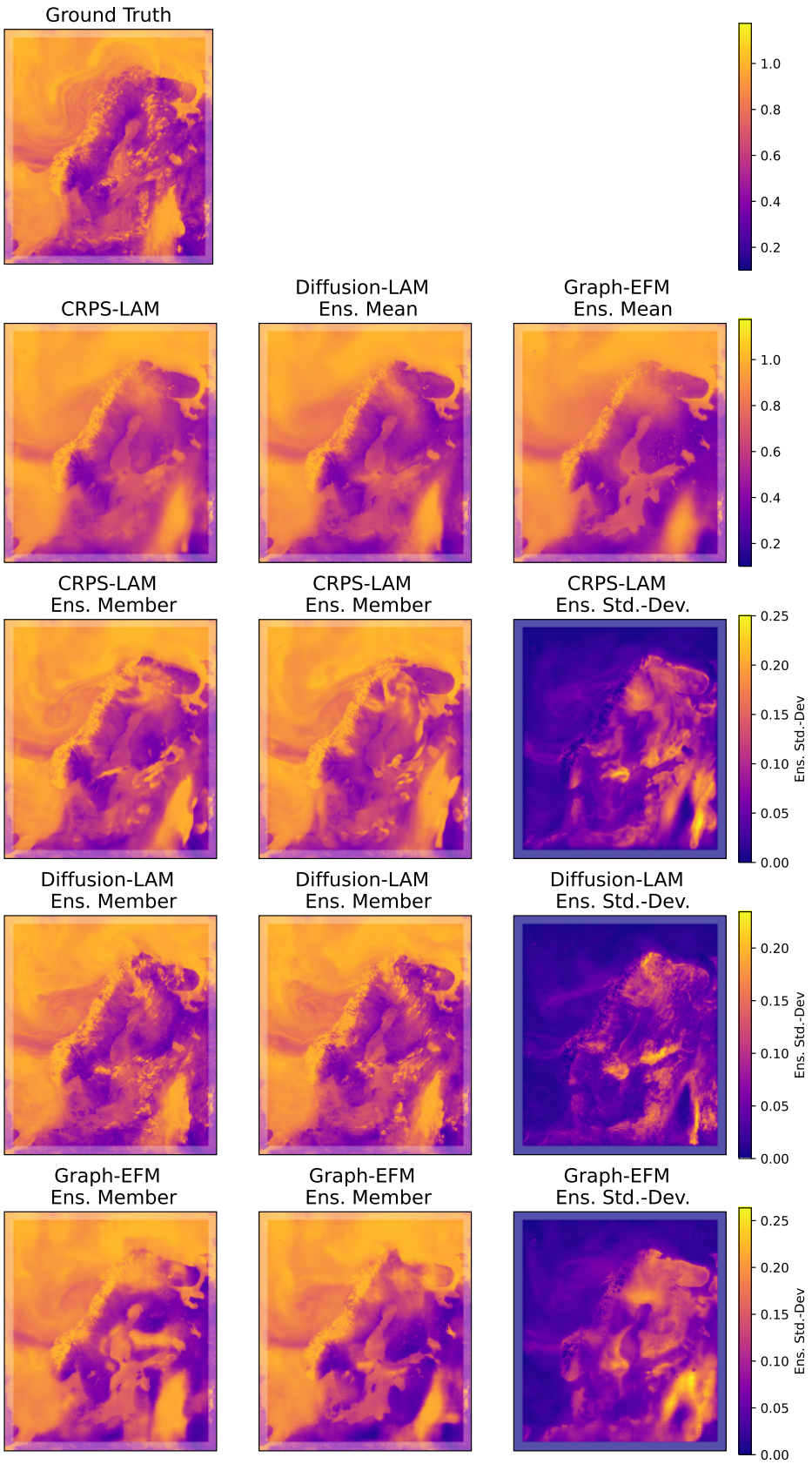
When preparing the dataset, we subsampled every fourth grid point, reducing the spatial resolution from the native 2.5 km of MEPS to 10 km (yielding 238×268 pixels). Consequently, spectral components at wavenumbers above 10^2 are not physically meaningful, since the models cannot represent scales finer than those resolved by the input data. In an operational setting, such unresolved scales should be suppressed by applying a suitable low-pass filter.

F.3 Detailed Evaluation

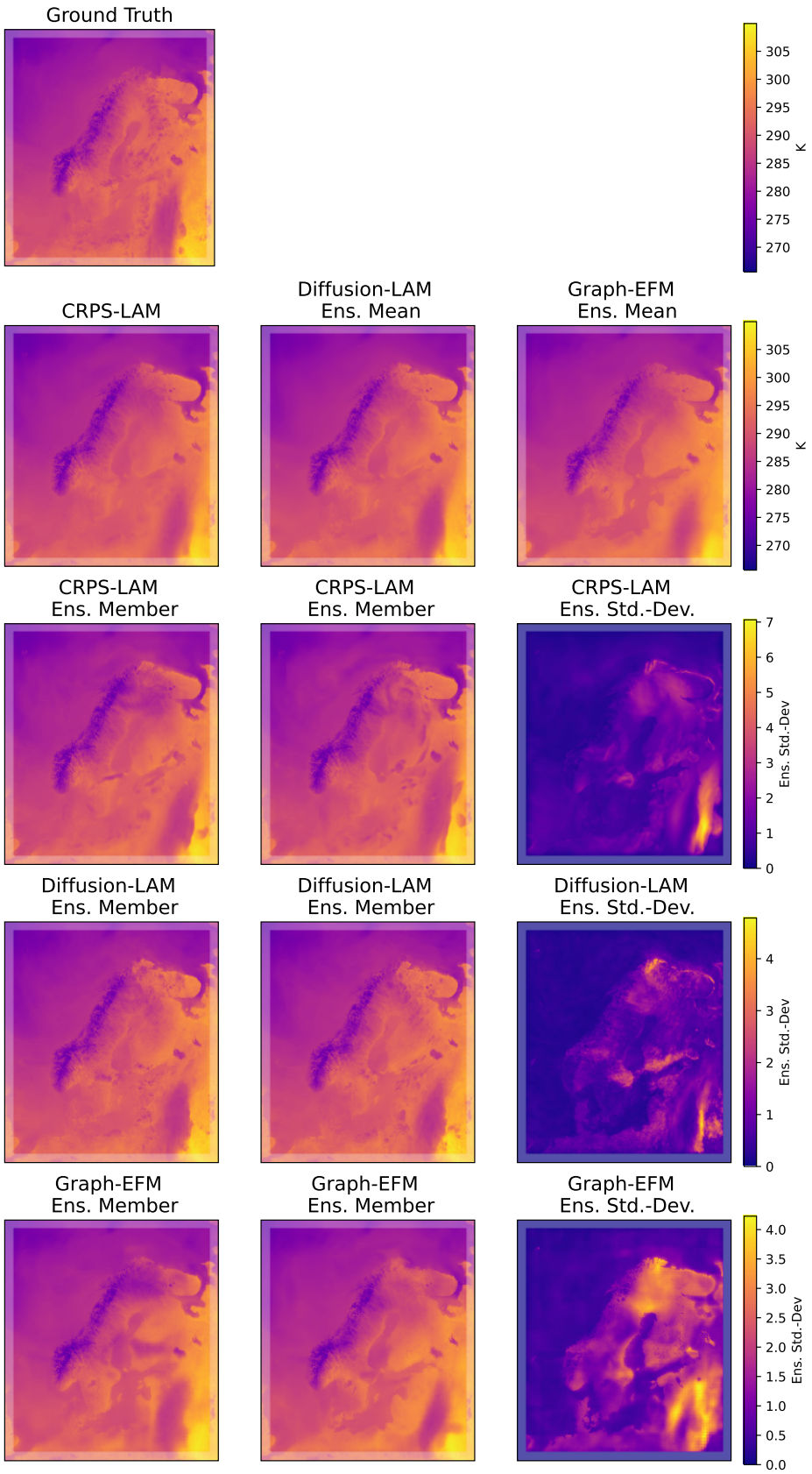
We provide detailed per-variable evaluation results in fig. 39, fig. 40, and fig. 41, together with a 57 h forecast example for a randomly selected test case for a subset of the variables in fig. 38. The corresponding energy spectra for all variables are shown in figs. 42 to 45. For an evaluation of deterministic models and less competitive probabilistic baselines on the MEPS dataset, we refer the reader to Oskarsson et al. (2023; 2024).


Ground Truth
102500
102000
101500
101000
100500
100000
99500
99000
Pa
CRPS-LAM
Diffusion-LAM
Ens. Mean
Graph-EFM
Ens. Mean
102500
102000
101500
101000
100500
100000
99500
99000
Pa
CRPS-LAM
Ens. Member
CRPS-LAM
Ens. Member
CRPS-LAM
Ens. Std.-Dev.
250
200
150
100
50
0
Ens. Std.-Dev
Diffusion-LAM
Ens. Member
Diffusion-LAM
Ens. Member
Diffusion-LAM
Ens. Std.-Dev.
160
140
120
100
80
60
40
20
0
Ens. Std.-Dev
Graph-EFM
Ens. Member
Graph-EFM
Ens. Member
Graph-EFM
Ens. Std.-Dev.
300
250
200
150
100
50
0
Ens. Std.-Dev

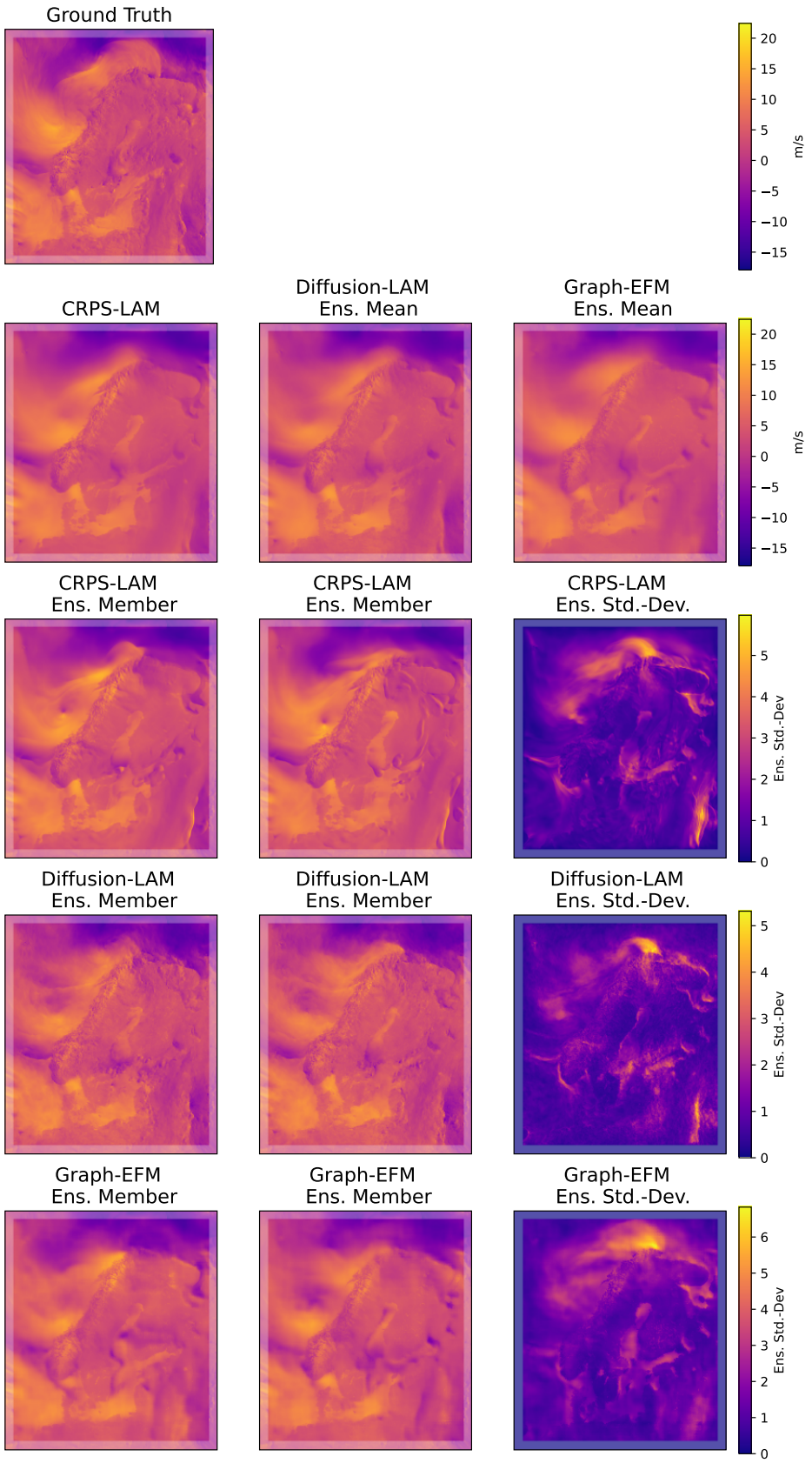
(a) `pres_0s`


Ground Truth
CRPS-LAM
Diffusion-LAM
Ens. Mean
Graph-EFM
Ens. Mean
CRPS-LAM
Ens. Member
CRPS-LAM
Ens. Member
CRPS-LAM
Ens. Std.-Dev.
Diffusion-LAM
Ens. Member
Diffusion-LAM
Ens. Member
Diffusion-LAM
Ens. Std.-Dev.
Graph-EFM
Ens. Member
Graph-EFM
Ens. Member
Graph-EFM
Ens. Std.-Dev.
1.0
0.8
0.6
0.4
0.2
0.25
0.20
0.15
0.10
0.05
0.00
Ens. Std.-Dev

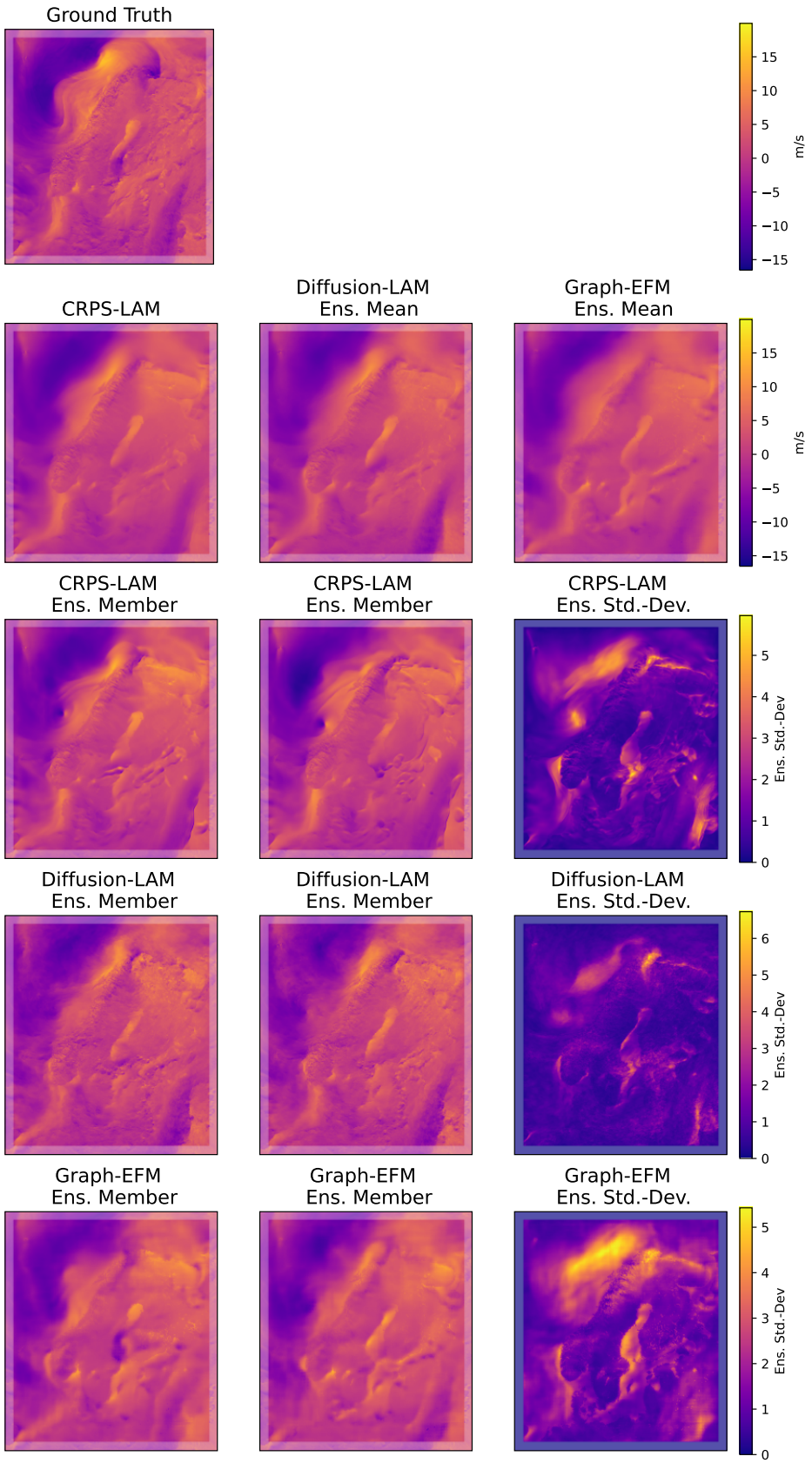
(b) r_2


Ground Truth
CRPS-LAM
Diffusion-LAM
Ens. Mean
Graph-EFM
Ens. Mean
CRPS-LAM
Ens. Member
CRPS-LAM
Ens. Member
CRPS-LAM
Ens. Std.-Dev.
Diffusion-LAM
Ens. Member
Diffusion-LAM
Ens. Member
Diffusion-LAM
Ens. Std.-Dev.
Graph-EFM
Ens. Member
Graph-EFM
Ens. Member
Graph-EFM
Ens. Std.-Dev.
K
Ens. Std.-Dev

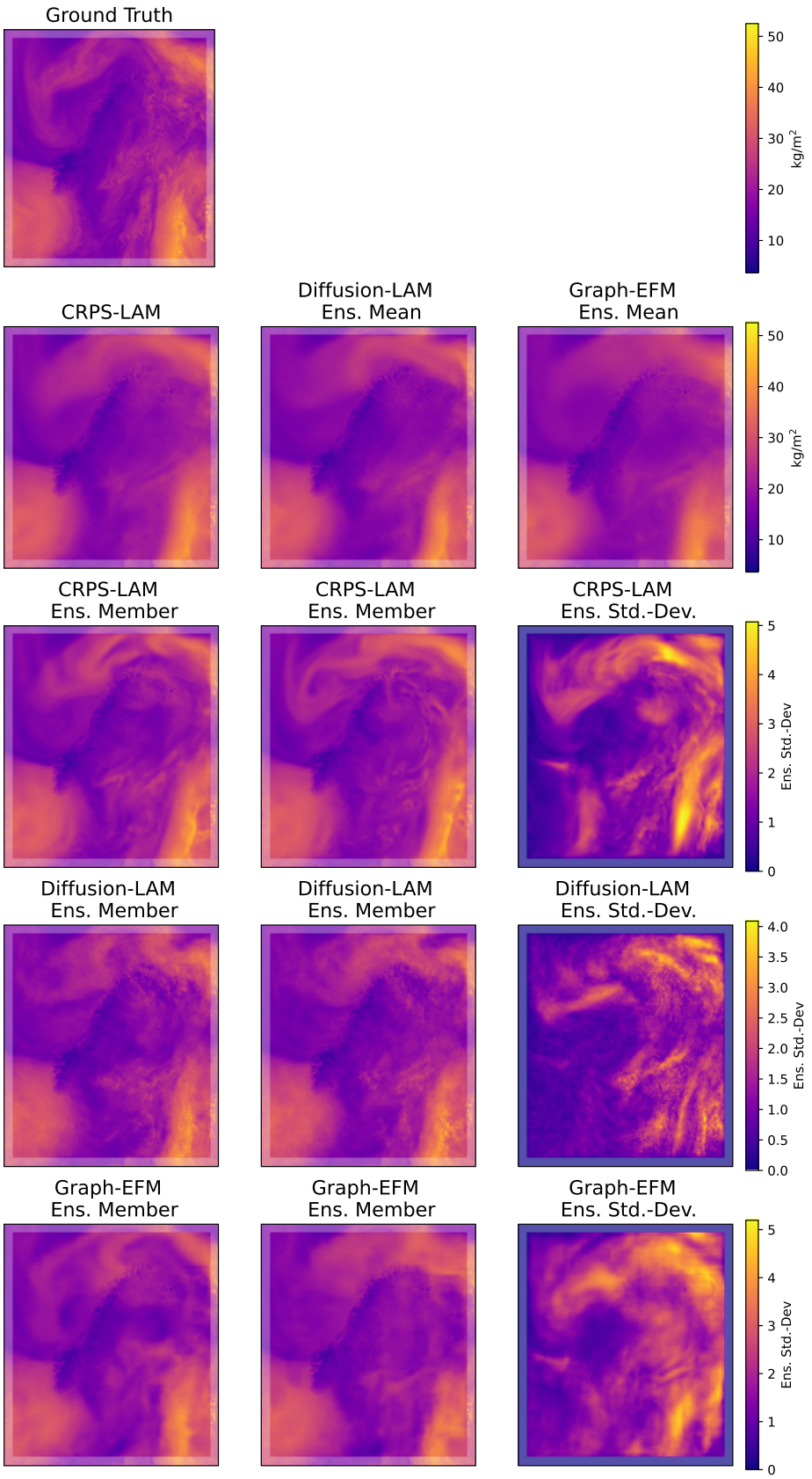
(c) t_2


Ground Truth
CRPS-LAM
Diffusion-LAM
Ens. Mean
Graph-EFM
Ens. Mean
CRPS-LAM
Ens. Member
CRPS-LAM
Ens. Member
CRPS-LAM
Ens. Std.-Dev.
Diffusion-LAM
Ens. Member
Diffusion-LAM
Ens. Member
Diffusion-LAM
Ens. Std.-Dev.
Graph-EFM
Ens. Member
Graph-EFM
Ens. Member
Graph-EFM
Ens. Std.-Dev.
m/s
Ens. Std.-Dev

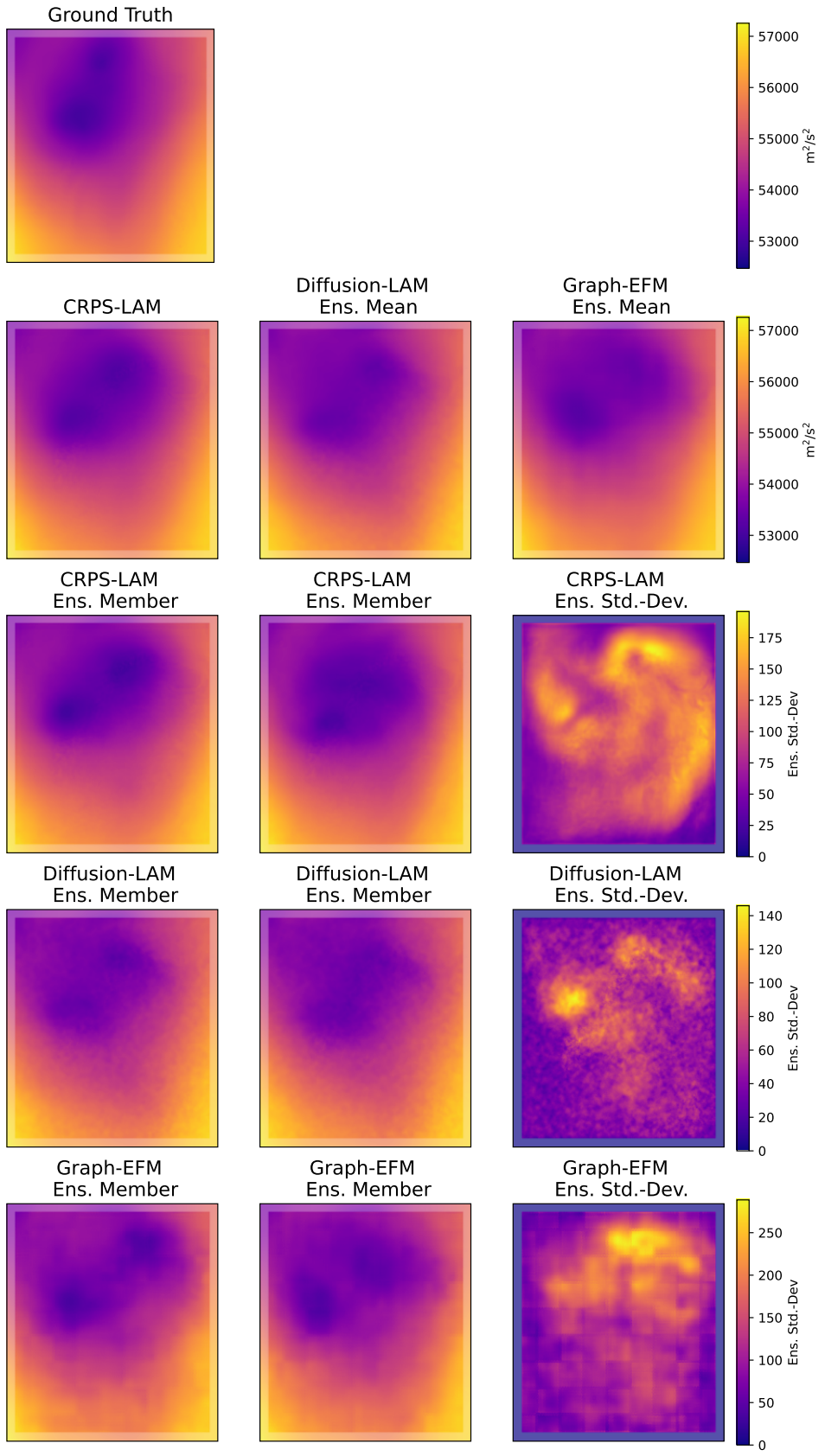
(d) u_65


Ground Truth
CRPS-LAM
Diffusion-LAM
Ens. Mean
Graph-EFM
Ens. Mean
CRPS-LAM
Ens. Member
CRPS-LAM
Ens. Member
CRPS-LAM
Ens. Std.-Dev.
Diffusion-LAM
Ens. Member
Diffusion-LAM
Ens. Member
Diffusion-LAM
Ens. Std.-Dev.
Graph-EFM
Ens. Member
Graph-EFM
Ens. Member
Graph-EFM
Ens. Std.-Dev.
m/s
Ens. Std.-Dev

(e) v_65


Ground Truth
CRPS-LAM
Diffusion-LAM
Ens. Mean
Graph-EFM
Ens. Mean
CRPS-LAM
Ens. Member
CRPS-LAM
Ens. Member
CRPS-LAM
Ens. Std.-Dev.
Diffusion-LAM
Ens. Member
Diffusion-LAM
Ens. Member
Diffusion-LAM
Ens. Std.-Dev.
Graph-EFM
Ens. Member
Graph-EFM
Ens. Member
Graph-EFM
Ens. Std.-Dev.
kg/m²
Ens. Std.-Dev

(f) `wvint_0`



(g) z_500

Figure 38: An ensemble forecast for a few variables at 57 h.

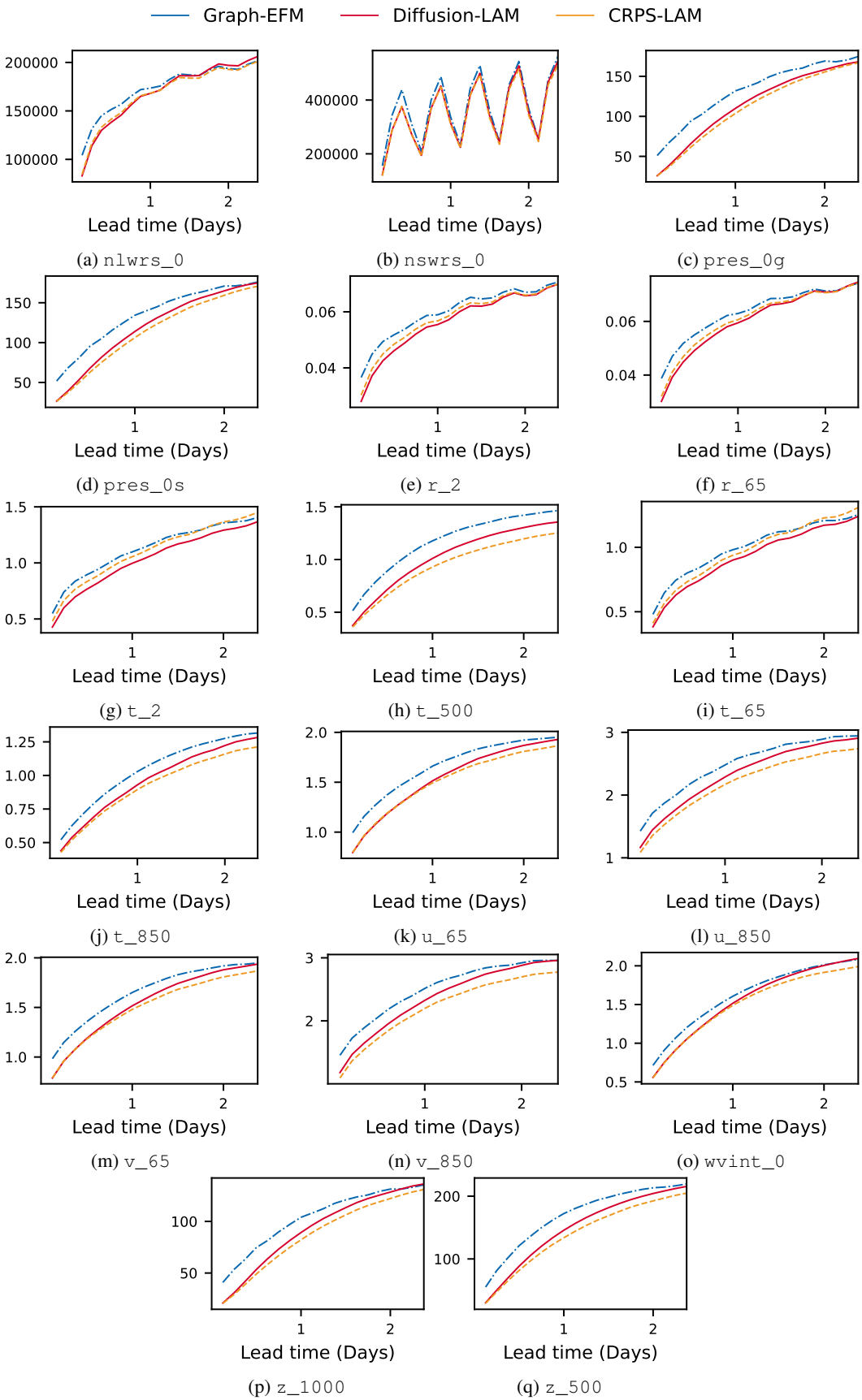


Figure 39: The RMSE results for each variable.

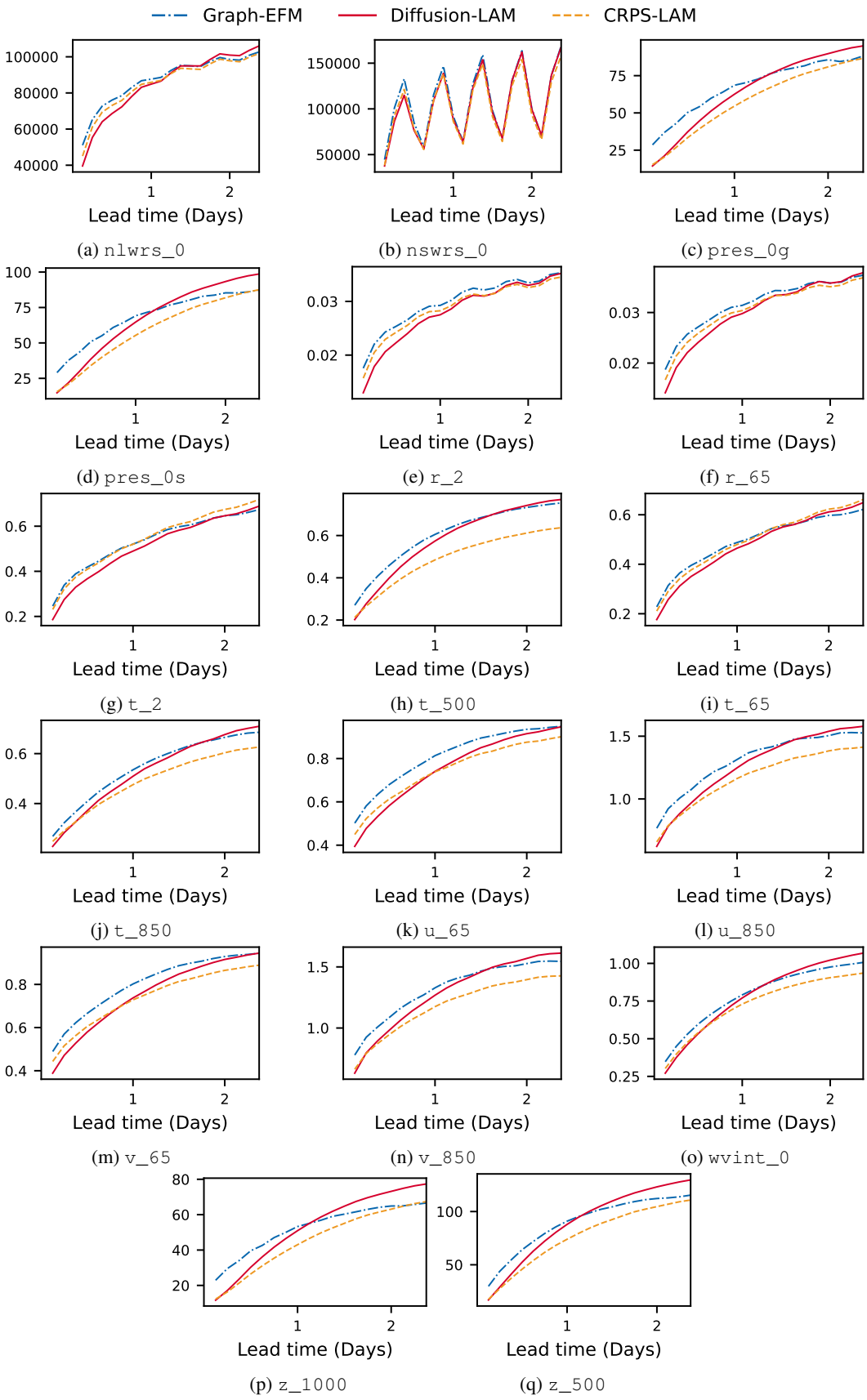


Figure 40: The CRPS results for each variable.

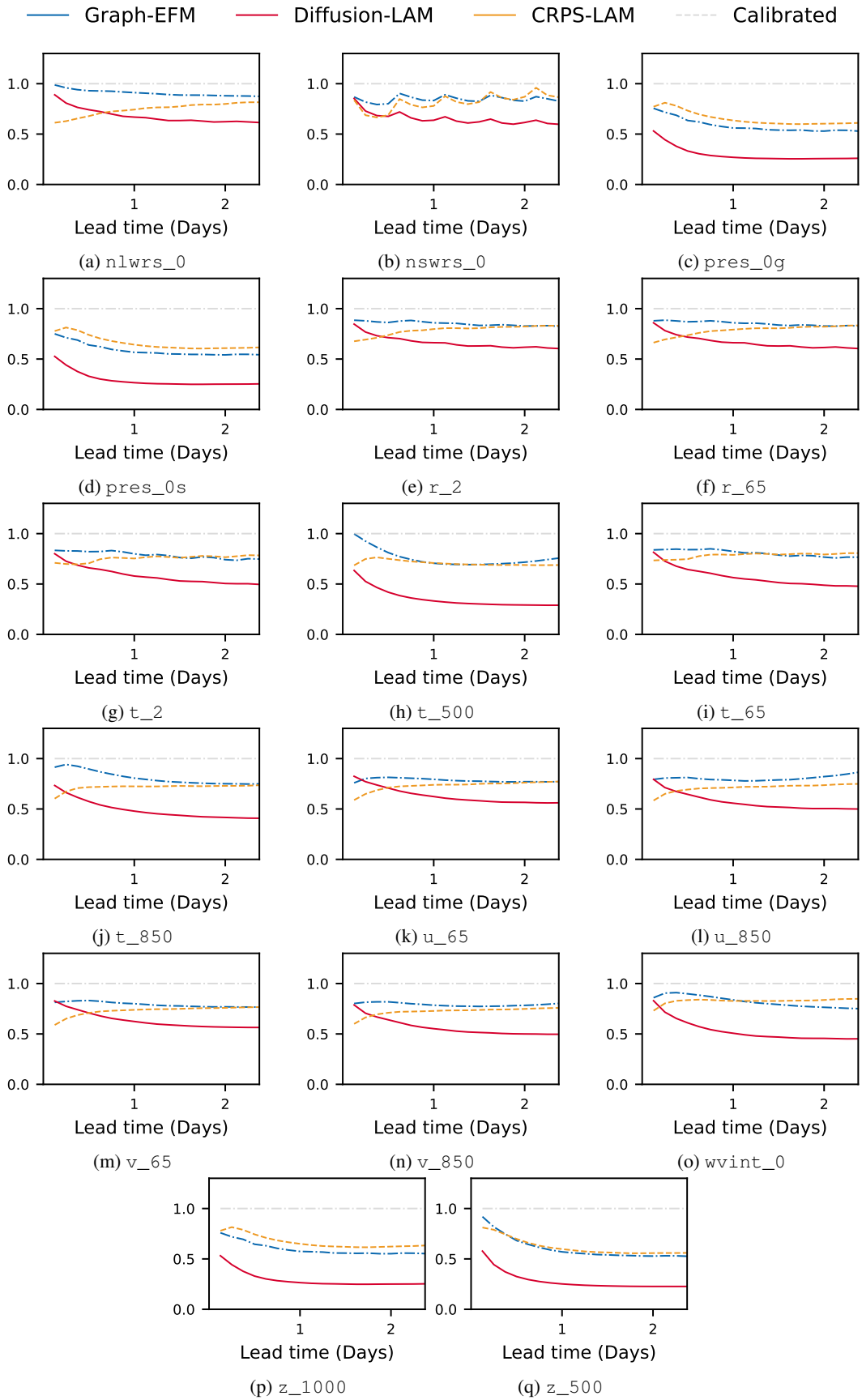
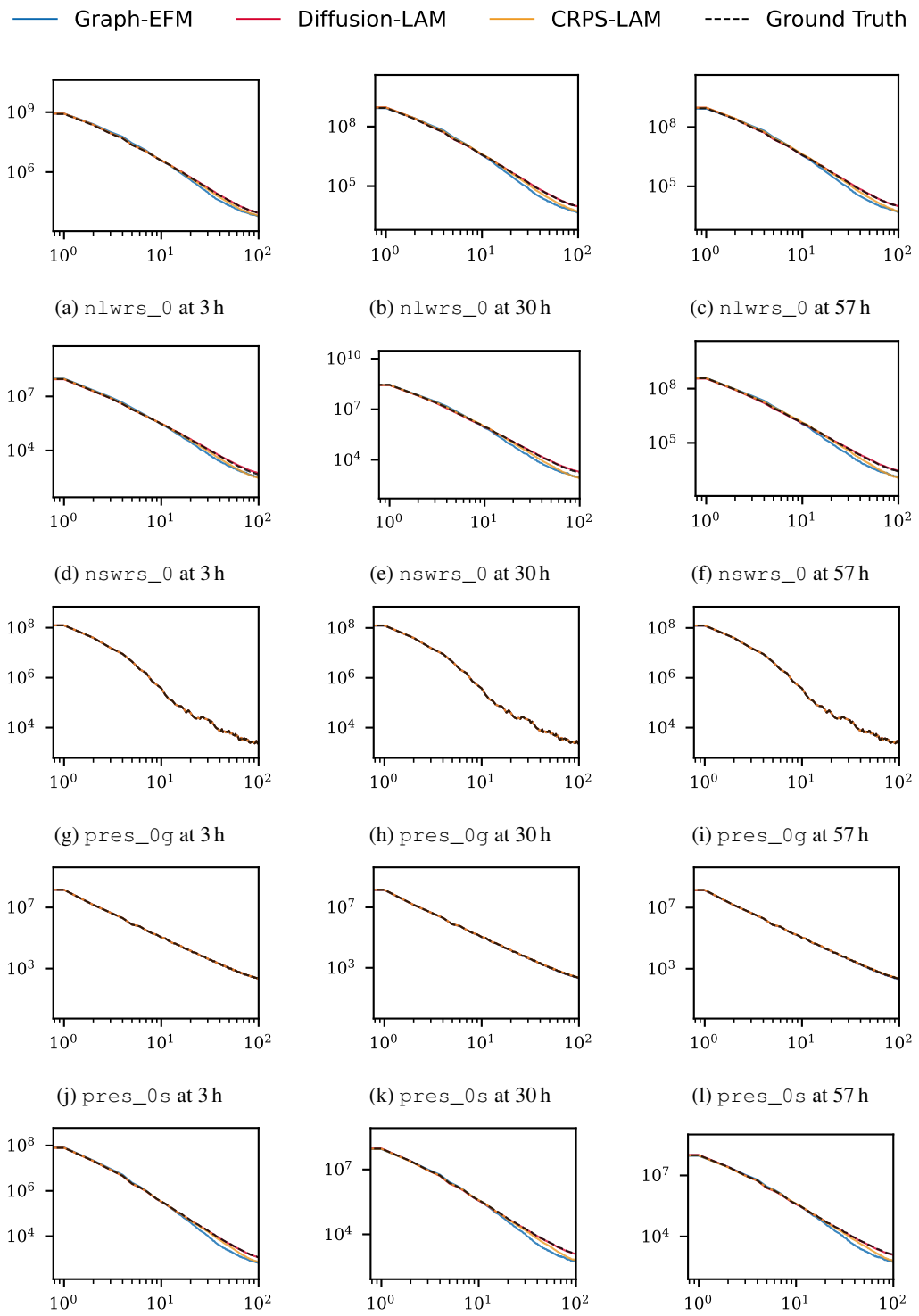


Figure 41: The SSR results for each variable.



(a) `nlwrs_0` at 3 h (b) `nlwrs_0` at 30 h (c) `nlwrs_0` at 57 h

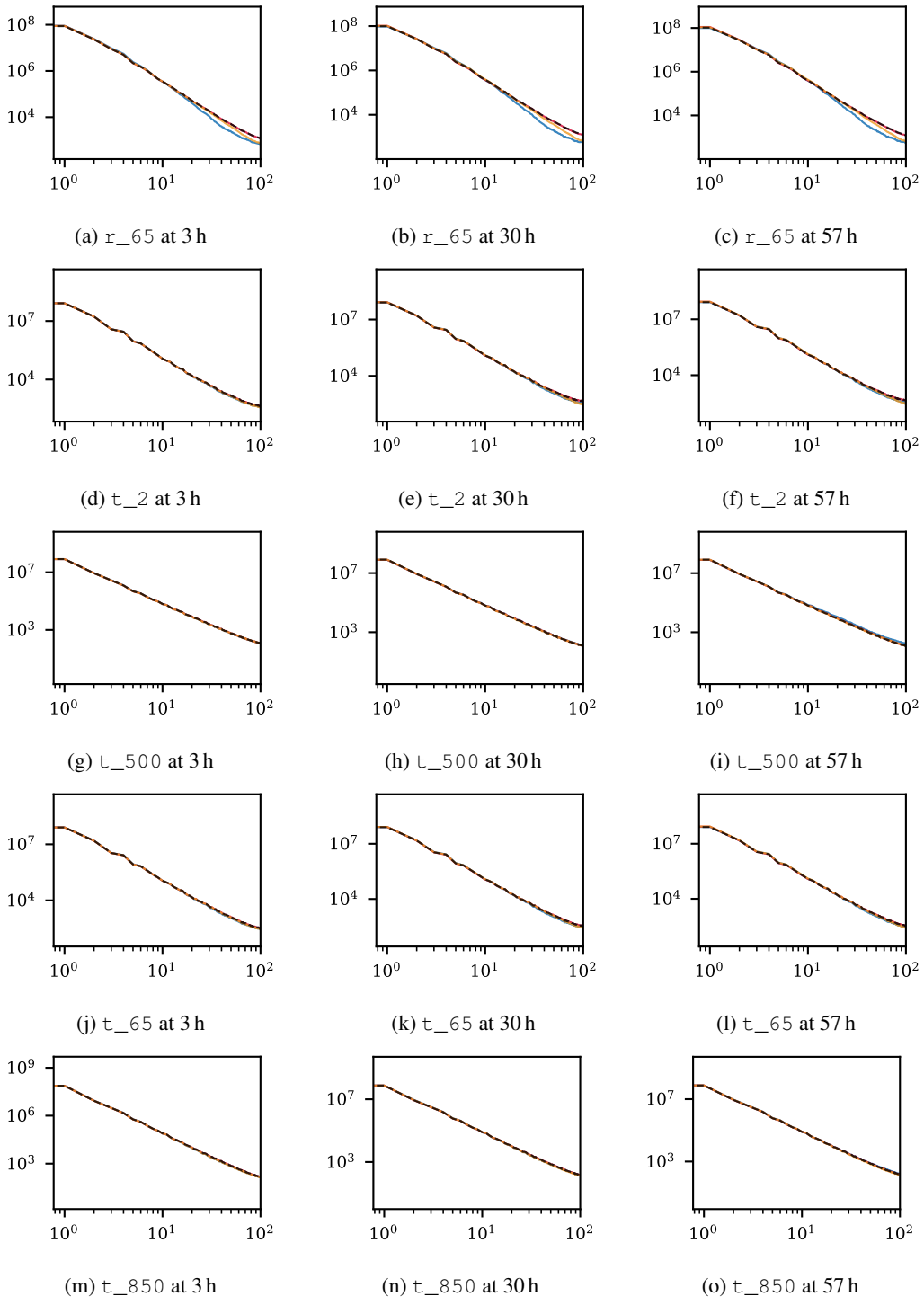
(d) `nswrs_0` at 3 h (e) `nswrs_0` at 30 h (f) `nswrs_0` at 57 h

(g) `pres_0g` at 3 h (h) `pres_0g` at 30 h (i) `pres_0g` at 57 h

(j) `pres_0s` at 3 h (k) `pres_0s` at 30 h (l) `pres_0s` at 57 h

(m) `r_2` at 3 h (n) `r_2` at 30 h (o) `r_2` at 57 h

Figure 42: The energy spectra for each variable at the lead times 3 h, 30 h, and 57 h.



(a) r_65 at 3 h (b) r_65 at 30 h (c) r_65 at 57 h

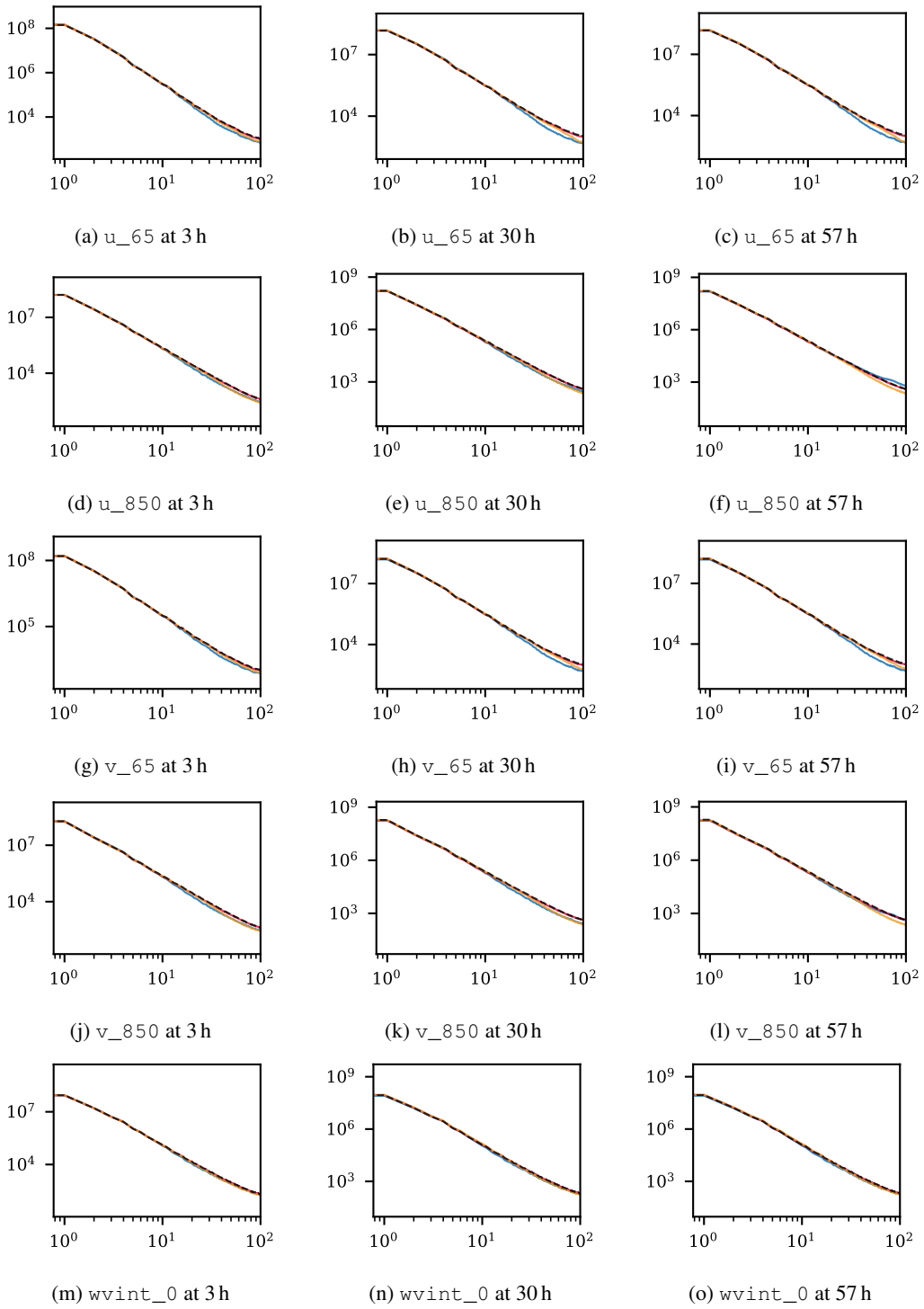
(d) t_2 at 3 h (e) t_2 at 30 h (f) t_2 at 57 h

(g) t_500 at 3 h (h) t_500 at 30 h (i) t_500 at 57 h

(j) t_65 at 3 h (k) t_65 at 30 h (l) t_65 at 57 h

(m) t_850 at 3 h (n) t_850 at 30 h (o) t_850 at 57 h

Figure 43: The energy spectra for each variable at the lead times 3 h, 30 h, and 57 h.



(a) u_65 at 3 h (b) u_65 at 30 h (c) u_65 at 57 h

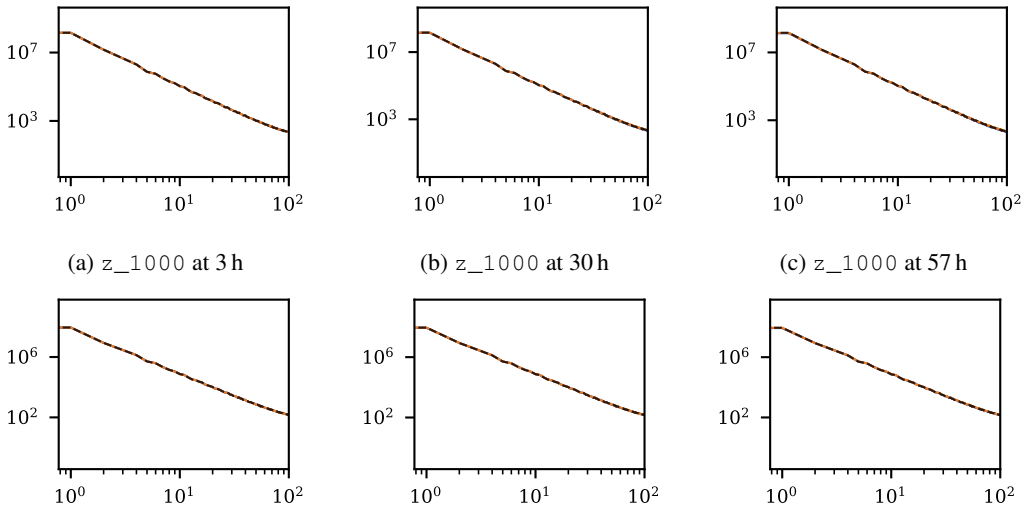
(d) u_850 at 3 h (e) u_850 at 30 h (f) u_850 at 57 h

(g) v_65 at 3 h (h) v_65 at 30 h (i) v_65 at 57 h

(j) v_850 at 3 h (k) v_850 at 30 h (l) v_850 at 57 h

(m) wvint_0 at 3 h (n) wvint_0 at 30 h (o) wvint_0 at 57 h

Figure 44: The energy spectra for each variable at the lead times 3 h, 30 h, and 57 h.



(a) `z_1000` at 3 h (b) `z_1000` at 30 h (c) `z_1000` at 57 h

(d) `z_500` at 3 h (e) `z_500` at 30 h (f) `z_500` at 57 h

Figure 45: The energy spectra for each variable at the lead times 3 h, 30 h, and 57 h.