Elucidated Rolling Diffusion Models for Probabilistic Forecasting of Complex Dynamics

Salva Rühling Cachay[1], Miika Aittala[2], Karsten Kreis[2], Noah Brenowitz[2], Arash Vahdat[2], Morteza Mardani[2]*, Rose Yu[1]*

[1]UC San Diego [2]NVIDIA *Equal advising

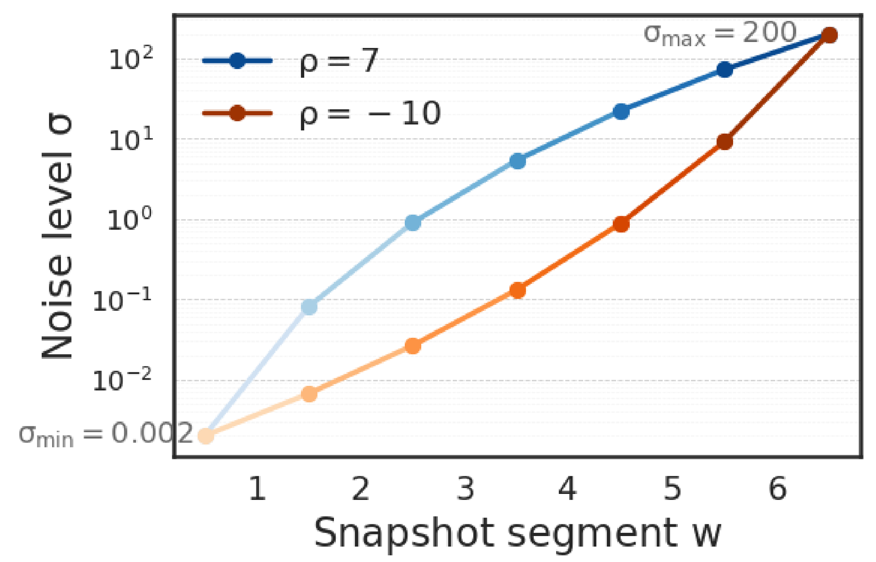
Code: https://github.com/NVlabs/ERDM

Motivation & Contributions

- **The Challenge:** Probabilistic forecasting of chaotic dynamics requires explicitly modeling the ***progressive growth of uncertainty*** over time.
- **The Gap:** Current approaches struggle to balance high-fidelity temporal modeling with efficiency:
  - ⋆ *Autoregressive models* ignore complex temporal dependencies.
  - ⋆ *Video diffusion* is often data-inefficient and computationally prohibitive.
  - ⋆ Existing *rolling diffusion frameworks* [3] lack integration with the more modern, successful diffusion paradigm EDM [1].
- **Our Solution – ERDM:** We introduce **Elucidated Rolling Diffusion Models**, the first framework to unify rolling sequence modeling with the performant design of EDM. By adapting the ***noise schedule, loss weighting, sampler,*** *and* ***spatiotemporal architecture*** to the rolling setting, ERDM achieves strong performance on Navier-Stokes and ERA5 weather forecasting with gains of up to 50% and 10%, respectively.

Rolling EDM Noise Schedule

We adapt EDM's noise schedule $\sigma(t)$ to a windowed, progressive schedule $\bar{\boldsymbol{\sigma}}(t) = (\bar{\sigma}_1(t), \ldots, \bar{\sigma}_W(t))$ for $t \in [0,1]$. Each snapshot w in the window has its own noise level $\bar{\sigma}_w(t)$, with $\bar{\sigma}_v(t) < \bar{\sigma}_w(t)$ for $v < w$ and $\bar{\sigma}_{w-1}(0) = \bar{\sigma}_w(1)$. We find a log-convex schedule ($\rho = -10$) is crucial for performance, outperforming the EDM default ($\rho = 7$). Our schedule assigns less noise to intermediate snapshots, providing more information for accurate joint denoising.



Training

Algorithm ERDM Training

Require: Training data $\mathcal{D}_{\text{train}}$, network F_θ, $\sigma_{\text{min}}, \sigma_{\text{max}}, \rho, P_{\text{mean}}, P_{\text{std}}$

repeat

Sample $\boldsymbol{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_W) \in \mathbb{R}^{W \times D}$ from $\mathcal{D}_{\text{train}}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{W \times D}), t \sim U([0,1))$

$\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_W) \leftarrow \bar{\boldsymbol{\sigma}}(t)$ ▷ Snapshot-dependent noise levels

$\bar{\boldsymbol{x}} \leftarrow \boldsymbol{y} + \boldsymbol{\sigma} \cdot \boldsymbol{\epsilon}$ ▷ Add rolling noise to data

$\hat{\boldsymbol{y}} \leftarrow c_{\text{skip}}(\boldsymbol{\sigma})\bar{\boldsymbol{x}} + c_{\text{out}}(\boldsymbol{\sigma}) F_\theta(c_{\text{in}}(\boldsymbol{\sigma})\bar{\boldsymbol{x}}, c_{\text{noise}}(\boldsymbol{\sigma}))$ ▷ Vectorized preconditioning

Update θ using $L_\theta = \frac{1}{W}\sum_{w=1}^{W} \lambda(\sigma_w) f(\sigma_w; P_{\text{mean}}, P_{\text{std}}) \|\boldsymbol{y}_w - \hat{\boldsymbol{y}}_w\|_2^2$ ▷ $f(\cdot; P_{\text{mean}}, P_{\text{std}})$ is the pdf of `Lognormal`$(P_{\text{mean}}, P_{\text{std}})$

until Converged

Sampling

Probability flow ODE. For a noisy sequence $\bar{\boldsymbol{x}} := \bar{\boldsymbol{x}}_{1:W}$:

$$\mathrm{d}\bar{\boldsymbol{x}} = -\mathrm{diag}(\bar{\sigma}_1(t)\dot{\bar{\sigma}}_1(t)\mathbf{I}_D, \ldots, \bar{\sigma}_W(t)\dot{\bar{\sigma}}_W(t)\mathbf{I}_D)\nabla_{\bar{\boldsymbol{x}}} \log p(\bar{\boldsymbol{x}}; \bar{\boldsymbol{\sigma}}(t))\mathrm{d}t,$$

Sampling: Initialization (l. 1-4), Denoising (l. 6-11), Rolling the window (l. 12-16)

Algorithm ERDM Deterministic Sampler (Euler-only)

1: **Require:** $\hat{\boldsymbol{y}}_{1:W}, \Delta t, T_{\text{forecast}}$
2: $t_{\text{cur}} \leftarrow 0$; $\mathcal{S} \leftarrow \varnothing$ ▷ Initialize global diffusion time and empty generated sequence
3: **sample** $\bar{\boldsymbol{x}}_{\text{cur}} \sim \mathcal{N}(\hat{\boldsymbol{y}}_{1:W}, \bar{\boldsymbol{\sigma}}(t_{\text{cur}})^2 \mathbf{I}_{W \times D})$ ▷ Initialize window with snapshot-dependent rolling noise
4: **while** $|\mathcal{S}| < T_{\text{forecast}}$ **do** ▷ Predict snapshot $|\mathcal{S}|+1$
5: $\boldsymbol{\sigma}_{\text{cur}} \leftarrow \bar{\boldsymbol{\sigma}}(t_{\text{cur}})$ ▷ Current noise levels
6: $t_{\text{next}} \leftarrow t_{\text{cur}} + \Delta t$ ▷ Global diffusion time after denoising
7: $\boldsymbol{\sigma}_{\text{next}} \leftarrow \bar{\boldsymbol{\sigma}}(t_{\text{next}})$ ▷ Noise levels at the end of this iteration
8: $\hat{\boldsymbol{y}} \leftarrow D_\theta(\bar{\boldsymbol{x}}_{\text{cur}}, \boldsymbol{\sigma}_{\text{cur}})$ ▷ Denoise sequence
9: $\boldsymbol{d} \leftarrow (\bar{\boldsymbol{x}}_{\text{cur}} - \hat{\boldsymbol{y}})/\boldsymbol{\sigma}_{\text{cur}}$ ▷ Evaluate $\mathrm{d}\bar{\boldsymbol{x}}/\mathrm{d}t$ at t_{cur}
10: $\bar{\boldsymbol{x}}_{\text{next}} \leftarrow \bar{\boldsymbol{x}}_{\text{cur}} + (\boldsymbol{\sigma}_{\text{next}} - \boldsymbol{\sigma}_{\text{cur}})\boldsymbol{d}$ ▷ Euler step from t_{cur} to t_{next}
11: **if** $t_{\text{next}} \geqslant 1$ **then** ▷ Is the first snapshot "clean"?
12: **add** first snapshot of $\hat{\boldsymbol{y}}$ to $\mathcal{S}$, discard it from the active window, $\bar{\boldsymbol{x}}_{\text{next}}$, and append new noisy snapshot $\boldsymbol{x}_{\text{new}} \sim \mathcal{N}(\mathbf{0}, \sigma_{\text{max}}^2 \mathbf{I}_{1 \times D})$ to it
13: $t_{\text{next}} \leftarrow t_{\text{next}} - 1$ ▷ Re-adjust global diffusion 'time' to be in $[0, 1)$
14: $\bar{\boldsymbol{x}}_{\text{cur}} \leftarrow \bar{\boldsymbol{x}}_{\text{next}}$; $t_{\text{cur}} \leftarrow t_{\text{next}}$
15: **return** $\mathcal{S}$

Key Idea: Elucidated Rolling Diffusion (ERDM)

We introduce **ERDM**, which successfully unifies a **rolling forecast structure** (applying increasing noise to farther lead times) with the principled, high-fidelity design of **Elucidated Diffusion Models (EDM)**.

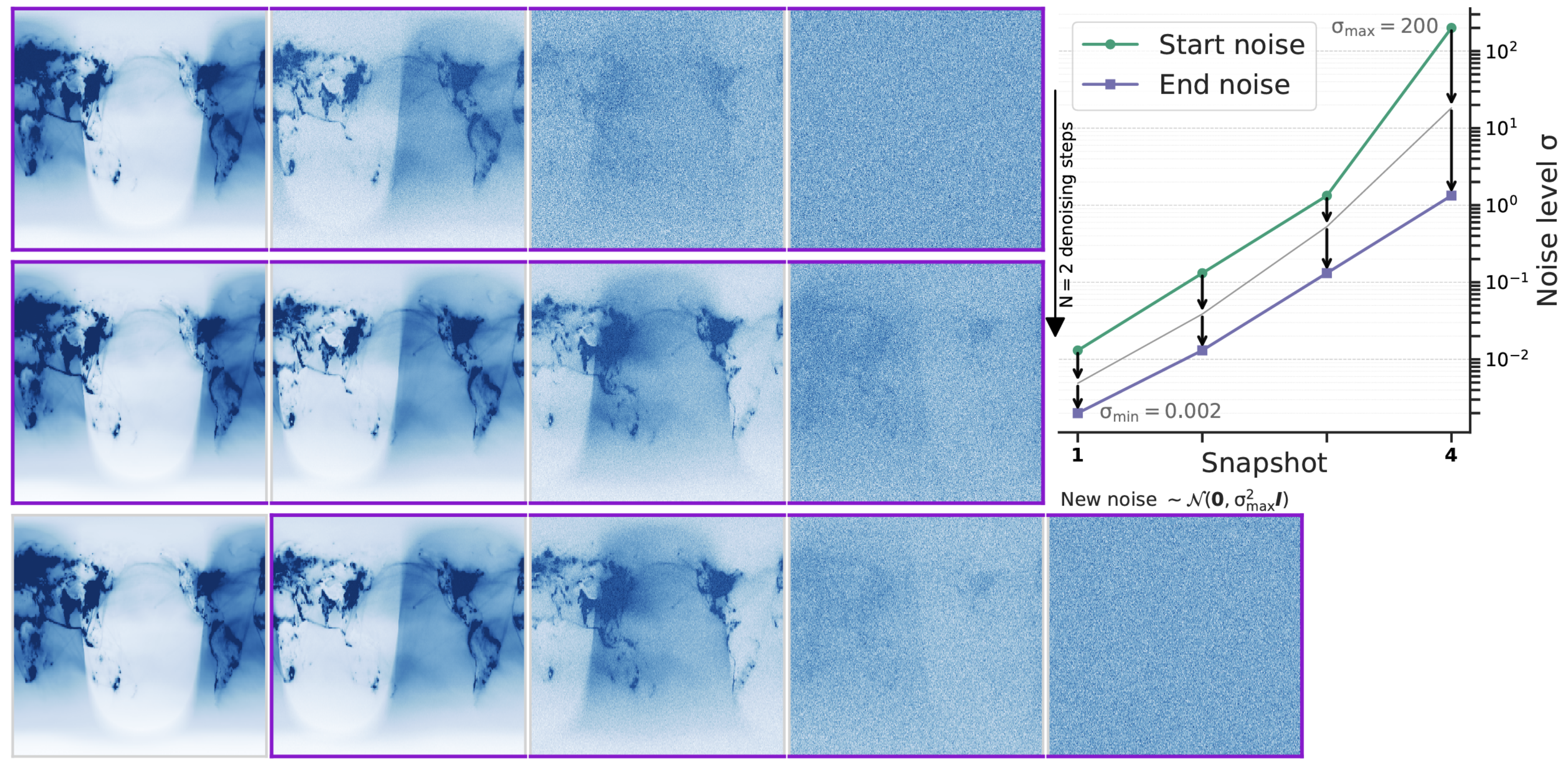
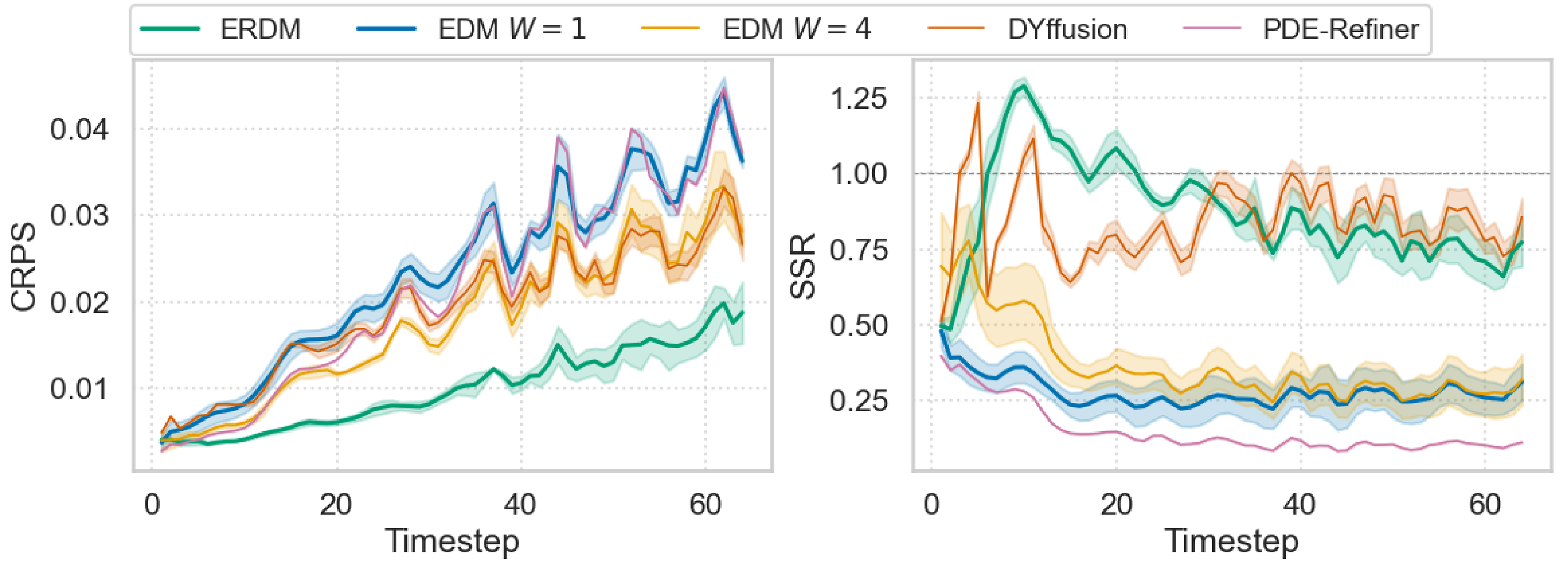


Figure 1. ERDM sampling with a window, highlighted in bold purple, of size $W = 4$. **Top row:** ERDM starts at diffusion time $t = 0$ with snapshots, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_W$, corrupted by progressively larger noise levels, $\bar{\sigma}_1(t) < \cdots < \bar{\sigma}_W(t) = \sigma_{\max}$. **Middle row:** After $N = 2$ joint denoising steps, the sequence reaches lower noise levels at $t = 1$ such that $\bar{\sigma}_1(1) = \sigma_{\min}$ and $\bar{\sigma}_w(1) = \bar{\sigma}_{w-1}(0)$ for $w > 1$, as illustrated in the right-hand panel. The now fully denoised first snapshot, $\boldsymbol{x}_1$, is returned. **Bottom row:** The rest of the sequence is shifted one slot to the right, and a fresh pure-noise snapshot is appended to the new window. The cycle then repeats.

Animation.

Results: 2D Navier-Stokes

- ERDM consistently outperforms all autoregressive EDM baselines (with $W = 1, 2, 4, 6$), as well as DYffusion [4] and PDE-Refiner [2], in probabilistic skill.
- At long horizons (>15 time steps), ERDM achieves an $\approx$ **50% improvement in CRPS** over the next-best baseline (EDM $W = 4$).
- ERDM also produces far more calibrated ensembles (Spread-Skill Ratio closer to 1).



Key Ablations

ERDM's performance relies on the synergy of its components. Removing any one is catastrophic:

- **Noise Schedule:** Using the EDM default ($\rho = 7$) schedule makes CRPS **2x worse**.
- **Loss Weighting:** Removing our loss weighting, $f(\sigma)$, causes a **>2x performance drop**.
- **Architecture:** Using a 2D-only architecture (stacking time in channels) results in a **4x performance degradation**.

Results: 1.5° ERA5 Weather Forecasting CRPS

- ERDM consistently outperforms the key autoregressive EDM baseline by up to 10% in CRPS.
- ERDM is competitive with external operational (IFS ENS) and hybrid physics-ML (NeuralGCM ENS) models, especially at mid-to-long-range lead times.
- ERDM is trained efficiently: 5 days on 4 H200 GPUs.

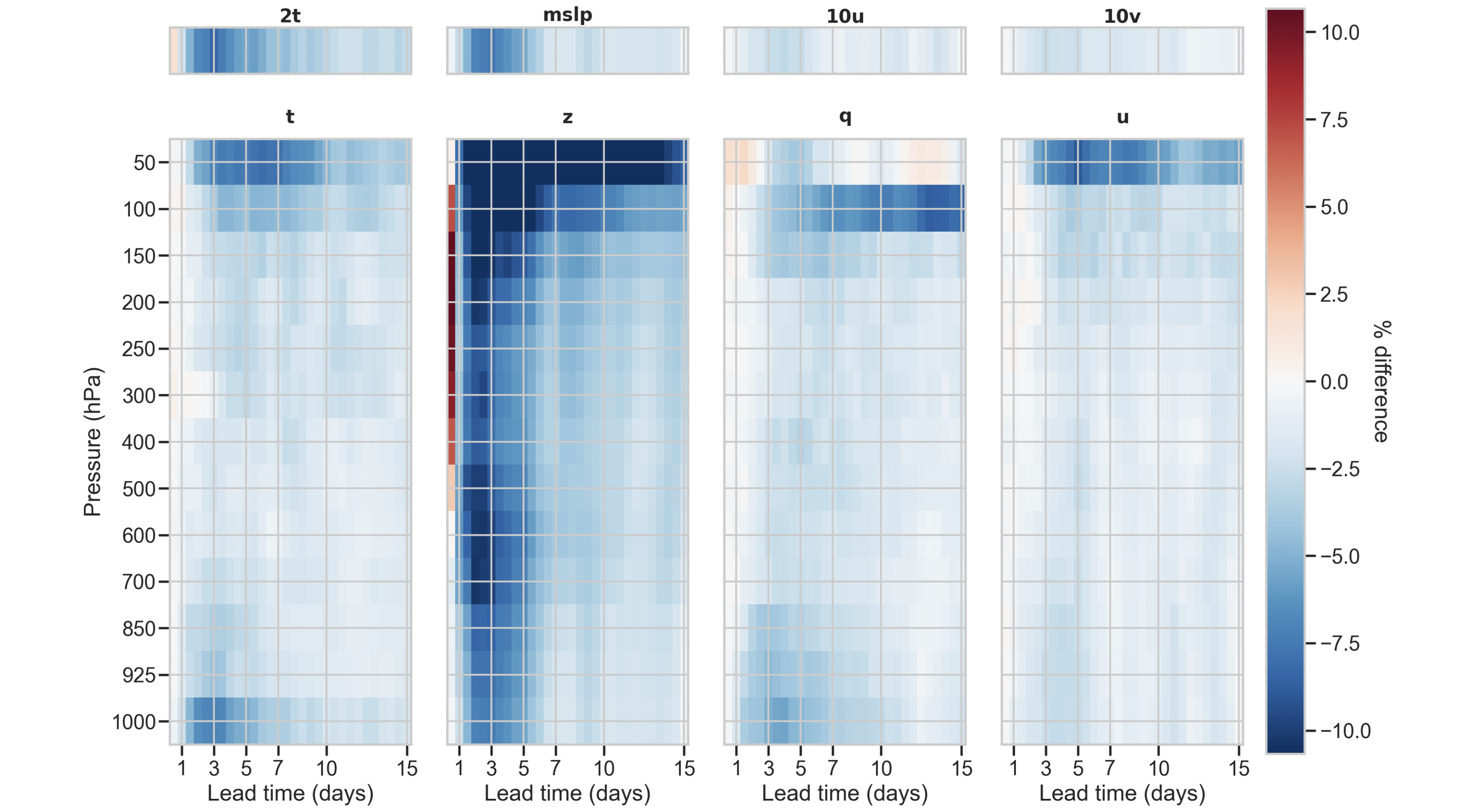
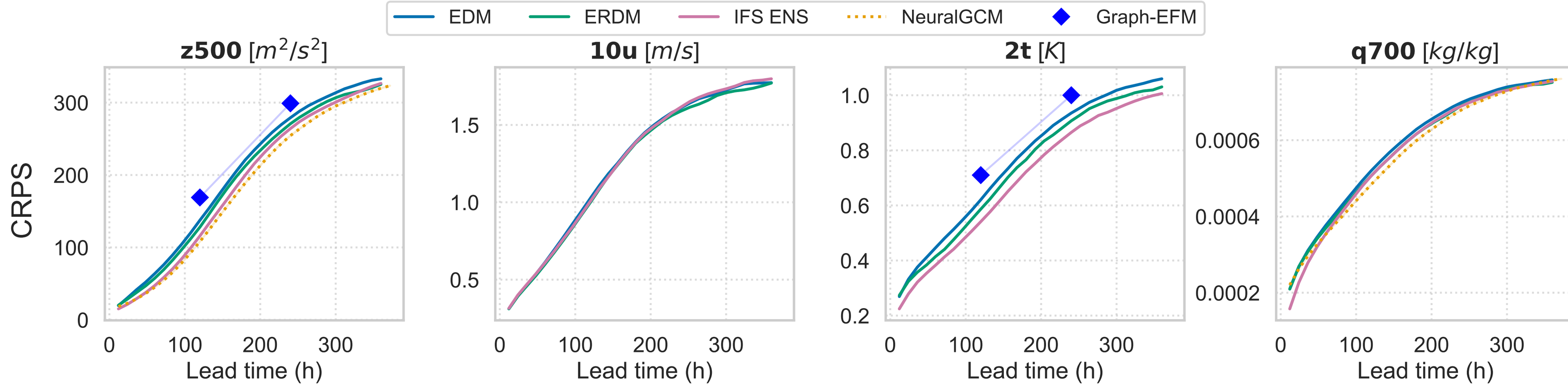


Figure 2. Score card of ERDM's CRPS relative to next-step autoregressive EDM baseline (in %; lower is better) for up to 15-day ahead forecasts. ERDM consistently outperforms the EDM baseline, especially for geopotential and high-altitude levels.



Results: Physical Realism (ERA5)

- Many ML models produce blurry forecasts that lack physical realism.
- ERDM produces physically consistent power spectra, **matching the operational, physics-based IFS ENS model** and outperforming other ML-based models like NeuralGCM.

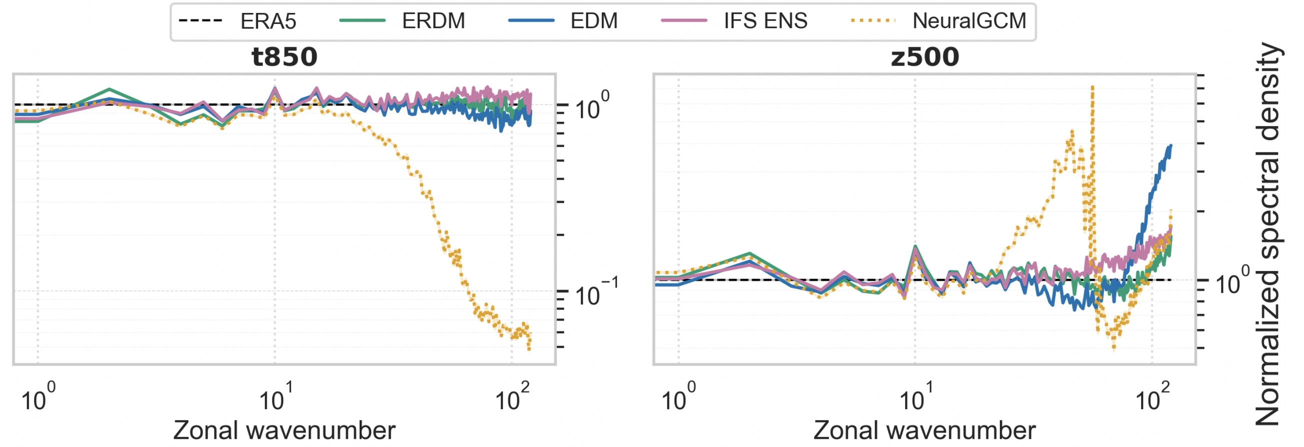


Fig. 3: Normalized spectral density of 14-day forecasts, averaged over high latitudes, $[60°, 90°]$. Spectra are divided by the target ERA5 reanalysis spectra.

References

[1] T. Karras, M. Aittala, T. Aila, and S. Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 2022.
[2] P. Lippe, B. S. Veeling, P. Perdikaris, R. E. Turner, and J. Brandstetter. PDE-Refiner: Achieving accurate long rollouts with temporal neural pde solvers. *Advances in Neural Information Processing Systems*, 2023.
[3] D. Ruhe, J. Heek, T. Salimans, and E. Hoogeboom. Rolling diffusion models. *International Conference on Machine Learning*, 2024.
[4] S. Rühling Cachay, B. Zhao, H. Joren, and R. Yu. DYffusion: A dynamics-informed diffusion model for spatiotemporal forecasting. *Advances in Neural Information Processing Systems*, 2023.

Acknowledgments S.R.C. acknowledges generous support from the summer internship and collaboration at NVIDIA. This work used resources from NERSC and was supported in part by the U.S. Army Research Office, DOE, IARPA, NSF, CDC, and DARPA.