LAGRANGIAN PARTICLES FOR LOCALIZED EXTREMES IN NEURAL WEATHER ODES

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural weather forecasts can lose skill on localized extremes even when global error remains low. We study LE-NODE, a continuous-time hybrid model that combines an Eulerian grid state with sparse, adaptively selected Lagrangian particles for localized anomalies. A synthetic advection probe characterizes high-wavenumber coefficient changes in Eulerian neural ODEs, motivating a particle state that follows localized structures. We evaluate the resulting model through forecast comparisons. On ERA5 at a five-day lead, LE-NODE obtains 221 m upper-tail Z500 RMSE versus 284 m for the strongest reported external baseline and 312 m for its Eulerian-only ablation under the same auxiliary objective. Recomputed lower-tail MSLP RMSE is 5.1 hPa versus 6.7 hPa for PanguLite. These results concern area-weighted pointwise tail metrics and the tested backbone. The results support a sparse particle branch as a useful complement to the tested Eulerian backbone.

1 INTRODUCTION

Learned global weather models such as GraphCast (Lam et al., 2023), Pangu-Weather (Bi et al., 2023), FourCastNet (Pathak et al., 2022), and the continuous-time ClimODE (Verma et al., 2024) now match or exceed operational numerical weather prediction on conventional skill metrics. The picture changes, however, as soon as one moves from average-case forecasting to the prediction of extreme events. Across multiple recent studies, learned forecasters systematically underestimate the central pressures of tropical cyclones, the peak temperatures of heatwaves, and the intensity of precipitation extremes (Watson, 2022; Bonavita, 2024; Ben Bouallègue et al., 2024). Accurate representation of these events is consequently an important forecast objective.

The literature offers three candidate explanations for why learned models lose tail intensity. Tail samples are scarce in training data (Rasp et al., 2020; 2024). Mean squared error optimization regresses predictions toward the conditional mean (Ravuri et al., 2021; Li et al., 2024). And localized, rapidly advecting structures are difficult to represent on fixed grids (Kochkov et al., 2024). The first two factors are widely acknowledged in the machine learning literature; the third is well established in computational fluid dynamics (LeVeque, 2002; Durran, 2010) but has received less direct empirical attention in the neural weather modeling community. Earlier work using neural differential equations for climate variables (Hwang et al., 2021; Choi et al., 2023) touched on this issue but focused on architectural design rather than diagnosing the failure mode.

We investigate the third factor as a design motivation, while recognizing that the relative effects of data scarcity, training objective, model scale, and transport representation remain unresolved on ERA5. Our synthetic advection experiment probes high-wavenumber behavior in trained Eulerian neural ODEs. Because its velocity varies spatially, transfer between Fourier modes is possible even for exact, incompressible advection. We use the probe to motivate a hybrid design, without interpreting initial-mode coefficient decay as a direct measure of numerical dissipation.

In LE-NODE, particles carry localized anomalies while a grid branch models the background field. The particle count is small compared with the number of grid cells and is adapted at fixed rollout boundaries. The method combines continuous-time particle evolution and adaptive selection with an Eulerian–Lagrangian coupling related to DeepLag (Ma et al., 2024). The present evidence concerns this implementation and scale; it does not establish that particles are necessary relative to higher-order or semi-Lagrangian Eulerian transport.

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.

We make three contributions:

- We characterize a high-wavenumber forecasting error in a controlled synthetic task and state precisely what the spatially varying-flow diagnostic can establish.
- We introduce LE-NODE, which couples a continuous-time particle state to a neural weather ODE and updates its sparse particle set between integration segments.
- We report ERA5 pointwise upper-tail results and a corrected lower-tail MSLP comparison, including an internal Eulerian-only ablation under the same auxiliary objective.

2 Related work

Data-driven weather prediction. Convolutional networks were first applied to regional nowcasting (Shi et al., 2015) and downscaling (Vandal et al., 2017), then extended to global medium-range forecasting. FourCastNet (Pathak et al., 2022) introduced adaptive Fourier neural operators, GraphCast (Lam et al., 2023) employed graph neural networks on icosahedral meshes, and Pangu-Weather (Bi et al., 2023) used Earth-specific vision transformers. ClimaX (Nguyen et al., 2023) pretrained a foundation model across heterogeneous climate datasets. GenCast (Price et al., 2025) extended deterministic approaches to probabilistic forecasting via diffusion models. These methods provide context for studying how architecture and objective affect pointwise tail errors.

Neural differential equations for atmospheric dynamics. Neural ODEs (Chen et al., 2018) parameterize continuous-time dynamics with neural networks. Early applications to climate variables include neural diffusion equations (Hwang et al., 2021) and neural advection-diffusion equations (Choi et al., 2023), which incorporated physical operators directly into the ODE drift. ClimODE (Verma et al., 2024) applied this framework to global weather with divergence-free velocity fields, while WeatherODE (Liu et al., 2024) incorporated wave equation structure. NeuralGCM (Kochkov et al., 2024) coupled differentiable dynamical cores with learned subgrid parameterizations. Physics-informed constraints have also improved turbulence modeling (List et al., 2022) and ocean simulation (Hao et al., 2025). Their different numerical and learned transport operators make a universal dissipation claim inappropriate.

Extreme event prediction. NowcastNet (Zhang et al., 2023) combined physical models with generative networks for precipitation nowcasting. ExtremeWeather (Racah et al., 2017) framed extreme detection as segmentation, while other work explored extreme value theory (Ding et al., 2019) and quantile regression (Meo et al., 2024). LE-NODE studies a complementary representation choice; objective design can also affect tail skill.

Lagrangian and hybrid representations in machine learning. Graph neural network simulators (Sanchez-Gonzalez et al., 2020) learn particle dynamics for various physical systems, with extensions incorporating smoothed particle hydrodynamics (Toshev et al., 2024) and standardized benchmarks (Toshev et al., 2023). Neural Material Point Method (Sharabi et al., 2024) and CORGI (Ji et al., 2025) couple graph-based Lagrangian simulators with grid-resident features for solid and fluid dynamics. DeepLag (Ma et al., 2024) combines Eulerian observations with Lagrangian tracking for fluid prediction using discrete recurrent updates. LE-NODE differs in its continuous-time particle evolution and adaptive spawning/removal at integration-segment boundaries. Our DeepLag comparison uses an adapted implementation with its own objective. Classical particle, semi-Lagrangian, and spectral transport methods provide further context (Harlow, 1962; Staniforth & Côté, 1991; Boyd, 2001). Cyclone-tracking studies offer a route to broader event-level validation (Neu et al., 2013; Hodges et al., 2011).

3 A synthetic advection probe and characteristic motivation

A controlled synthetic experiment probes the response of trained Eulerian neural ODEs to advected high-frequency structure. We distinguish the observed coefficient change from solver-induced dissipation and use the method of characteristics to motivate the particle branch.

3.1 A CONTROLLED DIAGNOSTIC ON SYNTHETIC ADVECTION

Consider passive scalar advection on a periodic domain $\Omega = [0, 2\pi]^2$ with prescribed divergence-free, spatially varying velocity:

$$\partial_t u + \mathbf{v} \cdot \nabla u = 0. \tag{1}$$

The exact flow preserves global L^p norms, but can redistribute energy among Fourier modes. We train Eulerian neural ODEs on on-demand synthetic samples with a full-field L^2 objective. For an initial mode $u_0(\mathbf{x}) = \cos(\mathbf{k} \cdot \mathbf{x})$, the probe records $A(\mathbf{k}, t) = |\hat{u}_\theta(\mathbf{k}, t)|/|\hat{u}_0(\mathbf{k})|$ versus $\kappa = |\mathbf{k}|\Delta x$.

Across the tested CNN, FNO, and U-Net configurations, A decreases more at larger κ and longer rollouts. This is an observed change in the *initial-mode coefficient*, not a dissipation estimate: even the exact solution need not have $A = 1$ in this flow. Phase error, spectral transfer, optimization error, and numerical diffusion are not separated by this measurement. A constant-velocity translation test or comparison against the exact time-t spectrum and global norms would be needed to isolate solver-induced attenuation. We do not extrapolate this probe to a fraction of ERA5 forecast error.

3.2 CHARACTERISTIC TRANSPORT AND IMPLEMENTATION ERROR

For a prescribed continuous velocity, the method of characteristics gives a simple reason to carry information along trajectories:

Proposition 3.1. *Let u satisfy* (1) *with Lipschitz velocity $\mathbf{v}$, and let $\mathbf{p}_k(t)$ satisfy*

$$\frac{d\mathbf{p}_k}{dt} = \mathbf{v}(\mathbf{p}_k, t), \qquad \mathbf{p}_k(0) = \mathbf{p}_k^0. \tag{2}$$

Then $u(\mathbf{p}_k(t), t) = u(\mathbf{p}_k^0, 0)$ for all $t \geq 0$. If the prescribed velocity is evaluated exactly and is sufficiently smooth, a stable q-th order integrator gives $\mathcal{O}(\Delta t^q)$ position error on a finite interval.

The first claim is the method of characteristics; the second is the Picard–Lindelöf bound combined with a q-th order convergence result. A complete proof is in Appendix B. LE-NODE uses learned grid velocities, interpolation, evolving particle states, and Gaussian splatting. Its errors therefore include velocity, spatial interpolation, integration, and reconstruction errors. The ideal characteristic result motivates the architecture but does not give an error bound for the implemented model.

The characteristic view separates two operations that a gridded forecast performs together: moving a feature and changing its intensity. A particle stores a location and an associated state, so its position can follow the learned flow while its intensity and scale evolve through the particle dynamics network. The gridded field continues to supply the surrounding context. This division is useful for compact features whose location changes substantially over a forecast window; it does not require interpreting every particle as a physical parcel or assigning a unique meteorological object to each particle.

4 METHOD

4.1 DESIGN PRINCIPLES

The grid retains global context while particles add a compact state for localized anomalies. We select particle locations from gradients and deviations from training climatology, and use three particle scales. A sparse budget can be useful when the relevant structures occupy a small fraction of the domain; the required budget can still grow in denser or higher-resolution regimes. Continuous-time evolution applies within each fixed-size segment, with particle-set updates at segment boundaries.

4.2 ARCHITECTURE

LE-NODE decomposes the state as $u = u_{\text{grid}} + u_{\text{particle}}$, with the two components evolving under coupled neural ODEs (Figure 1).

Eulerian branch. The background field evolves on a latitude–longitude grid as $\partial u_{\text{grid}}/\partial t = \mathcal{G}_\theta(u_{\text{grid}}, \mathbf{v}_{\text{grid}}, u_{\text{particle}})$, where $\mathcal{G}_\theta$ is a U-Net (Ronneberger et al., 2015) augmented with Fourier neural operator layers (Li et al., 2021). The advecting velocity is parameterized locally from a learned

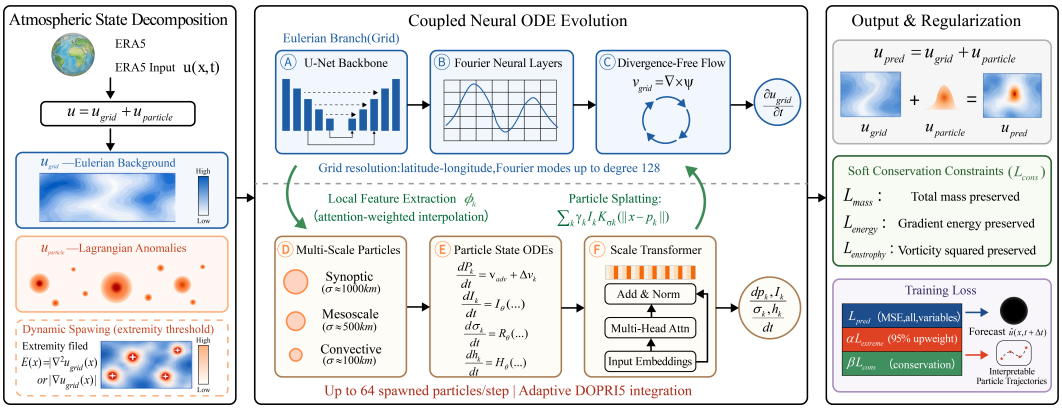


Figure 1: LE-NODE decomposes the atmospheric state into a Eulerian background field and Lagrangian particles. Particles are spawned in high-gradient regions and exchange information with the grid via attention (in) and Gaussian splatting (out).

stream function, $\mathbf{v}_{\text{grid}} = \nabla \times \mathcal{S}_\theta(u_{\text{grid}})$, giving a divergence-free horizontal velocity in the continuous formulation. This does not by itself conserve the predicted atmospheric variables. The branch ingests the splatted particle field as an auxiliary input. Backbone hyperparameters are in Appendix F.

Lagrangian branch. Each particle k carries a tuple $(\mathbf{p}_k, I_k, \sigma_k, \mathbf{h}_k)$ for position, intensity, spatial scale, and latent state. These evolve under coupled ODEs:

$$\frac{d\mathbf{p}_k}{dt} = \mathbf{v}_{\text{adv}}(\mathbf{p}_k, t) + \Delta\mathbf{v}_k, \qquad \frac{d(I_k, \sigma_k, \mathbf{h}_k)}{dt} = \mathcal{D}_\theta(I_k, \sigma_k, \mathbf{h}_k, \boldsymbol{\phi}_k), \tag{3}$$

where $\mathbf{v}_{\text{adv}}$ is interpolated from the grid, $\Delta\mathbf{v}_k$ is a learned ageostrophic correction, $\mathcal{D}_\theta$ is a shared-trunk MLP with separate heads for $(I, \sigma, \mathbf{h})$, and $\boldsymbol{\phi}_k$ is a local feature obtained by attention over grid points within a fixed radius of $\mathbf{p}_k$ (full form in Appendix F).

Spawning, removal, multi-scale. Particles are spawned at locations of high extremity,

$$E(\mathbf{x}) = w_1 \frac{\|\nabla u_{\text{grid}}(\mathbf{x})\|_2}{\|\nabla u\|} + w_2 \frac{|u_{\text{grid}}(\mathbf{x}) - \mu(\mathbf{x})|}{\sigma_{\text{clim}}(\mathbf{x})}, \tag{4}$$

The gradient term highlights sharp spatial variation; the anomaly term highlights deviations from the training climatology. At each fixed 6-hour segment boundary, discrete top-k selection spawns up to 64 particles at high-extremity locations; particles with $\gamma_k|I_k|$ below threshold or outside the domain are removed. The gate $\gamma_k = \sigma((E(\mathbf{p}_k) - \tau)/\epsilon)$ is differentiable for selected particles, while top-k indices are stop-gradient selections. Removal closes that particle's gradient path at the boundary. Particle number remains fixed within each DOPRI5 solve, which restarts after the update. Following the canonical decomposition of atmospheric motion (Orlanski, 1975; Holton & Hakim, 2013), we maintain three particle populations at synoptic ($\sigma \approx 1000$ km), meso ($\sigma \approx 500$ km), and convective ($\sigma \approx 100$ km) scales, with cross-scale interaction via a single transformer layer over particle latent states.

Coupling. The Lagrangian contribution to the total field is splatted using normalized Gaussian kernels (Monaghan, 2005):

$$u_{\text{particle}}(\mathbf{x}) = \sum_{k=1}^{K} \gamma_k I_k \mathcal{K}_{\sigma_k}(\|\mathbf{x} - \mathbf{p}_k\|), \quad \mathcal{K}_\sigma(r) = (2\pi\sigma^2)^{-1} \exp(-r^2/2\sigma^2). \tag{5}$$

This closes the loop: the grid sees what particles carry, and particles see the smooth context the grid maintains.

Particle-set updates. Let t_j denote consecutive 6-hour boundaries. During $[t_j, t_{j+1}]$, the particle count and state dimension are fixed, allowing the grid and particle states to be integrated jointly by

one adaptive ODE solve. At t_{j+1}, the model evaluates the extremity score on the updated grid state, retains active particles, removes particles meeting the stated criteria, and initializes newly selected particles before starting the next solve. This schedule makes adaptive selection compatible with a fixed-dimensional ODE solver. Gradients through each segment reach the continuous particle states and the soft gate of selected particles; the discrete choice of top-k locations is treated as a stop-gradient operation. The grid-to-particle attention and particle-to-grid splatting then couple the newly formed state to the following segment. The resulting trajectory is continuous within a segment, while the particle set may change at its boundaries.

4.3 Training objective

The loss combines prediction, extreme-event reweighting, and drift penalties on two-dimensional proxies,

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \alpha\mathcal{L}_{\text{extreme}} + \beta(\mathcal{L}_{\text{mass}} + \mathcal{L}_{\text{energy}} + \mathcal{L}_{\text{enstrophy}}), \tag{6}$$

where $\mathcal{L}_{\text{pred}}$ is full-field MSE, the proxy terms penalize normalized squared drift between initial and predicted states, and the extreme-event term upweights errors at grid points with observed values outside the central 90% of the climatological distribution (covering both high and low extremes symmetrically):

$$\mathcal{L}_{\text{extreme}} = \sum_{v} \mathbb{E}\Big[\mathbf{1}_{u_{\text{obs}}^{(v)} \notin [q_5^{(v)}, q_{95}^{(v)}]} \cdot \|u_{\text{pred}}^{(v)} - u_{\text{obs}}^{(v)}\|_2^2\Big]. \tag{7}$$

Within each fixed 6-hour segment, both branches are integrated jointly with DOPRI5 (Chen et al., 2018); we use $\alpha = 5$, $\beta = 0.1$, gradient clipping at norm 1.0, and no curriculum on α, β during the main training phase.

5 Experiments

5.1 Dataset and preprocessing

We use ERA5 reanalysis data (Hersbach et al., 2020) at $0.25°$ horizontal resolution, corresponding to approximately 28 km grid spacing at the equator. The domain covers the global atmosphere on a 721×1440 latitude-longitude grid. We include six primary variables: geopotential height at 500 hPa (Z500), temperature at 850 hPa (T850), 2-meter surface temperature (T2m), mean sea level pressure (MSLP), and 10-meter wind components (U10, V10). Data spans 1979 to 2021 at 6-hourly temporal resolution. We partition into training (1979 to 2015), validation (2016 to 2017), and test (2018 to 2021) periods. Each variable is normalized to zero mean and unit variance using statistics from the training period. Climatological percentiles for extreme event definition are also computed from training data.

5.2 Extreme event definition

Thresholds are computed from the training period by location and calendar month. The reported tail scores are pointwise and condition on observations: upper 95th percentile for Z500, T850, and T2m; lower 5th percentile for MSLP. Upper-tail Z500 describes high geopotential anomalies and is not a comprehensive blocking-event score. The wind-component scores in Appendix E use component-wise upper tails; they should not be read as wind-speed extremes. Event-oriented wind verification would use $\sqrt{U10^2 + V10^2}$ with a separately calibrated threshold. The model does not predict vertical velocity or 850-hPa vorticity, so we do not evaluate vertical-motion or vorticity-qualified cyclone events.

5.3 Baseline methods and evaluation

We compare LE-NODE with persistence, ClimODE (Verma et al., 2024), an ERA5 adaptation of DeepLag (Ma et al., 2024), and PanguLite, our approximately 50M-parameter reduction of Pangu-Weather (Bi et al., 2023) trained from scratch. PanguLite is not the released full Pangu-Weather model. The Eulerian-only ablation removes the particle branch while retaining the LE-NODE backbone and auxiliary losses. External methods use their own objectives, so cross-method gaps do not isolate

the representation. A previously reported zero-shot ClimaX reference is excluded from the main comparison because its checkpoint and data-overlap provenance were insufficiently documented.

Each learned method was trained with three seeds. Seed standard deviations measure training variability, not uncertainty across correlated forecast cases or years. All spatial averages and event counts use cosine-latitude weights $w_i = \cos \varphi_i$; the pointwise masked Tail-RMSE for variable v and observed tail mask M_v is

$$\mathrm{TailRMSE}_v = \sqrt{\frac{\sum_{t,i} w_i M_v(y_{t,i})(\hat{y}_{t,i} - y_{t,i})^2}{\sum_{t,i} w_i M_v(y_{t,i})}}. \tag{8}$$

Full-field RMSE uses the same area weights without M_v; ACC uses anomalies relative to training climatology. For CSI, both forecast and observation use the same variable-specific threshold, and weighted event counts form $\mathrm{TP}/(\mathrm{TP} + \mathrm{FP} + \mathrm{FN})$. We report no MSLP CSI; the intended cyclone definition requires 850-hPa vorticity, which is not among the predicted outputs. Pointwise scores can penalize displaced but otherwise sharp events twice; neighborhood and object-based verification is outside the available results.

The tail mask depends on the observed field and is identical for every method evaluated on a given variable and test case. Tail-RMSE therefore asks how accurately a forecast represents the field at locations where the verifying observation is extreme. It differs from a score conditioned on each model's own forecast extremes: a model cannot improve the denominator by predicting fewer events. The same construction also makes location errors consequential. For example, a sharp forecast minimum displaced from an observed pressure minimum incurs an error at the observed center even if its predicted central pressure is accurate. We use the pointwise score to compare the reported forecasts and reserve storm-object claims for a verification protocol that matches predicted and observed systems.

The main training times reported for ClimODE, PanguLite, and LE-NODE are 72, 84, and 108 wall-clock hours per seed on eight A100 GPUs, respectively (576, 672, and 864 GPU-hours per seed). They are not equal-compute runs. The extended 108-hour comparisons for ClimODE and PanguLite match hardware and wall time, but not FLOPs or optimization opportunity.

5.4 Main results

Table 1 reports five-day pointwise scores. LE-NODE improves upper-tail Z500 RMSE from 284 to 221 m relative to PanguLite (22.2%) and from 312 to 221 m relative to its same-objective Eulerian-only ablation. For MSLP, lower-tail scores recomputed from saved forecasts are 5.1 hPa for LE-NODE and 6.7 hPa for PanguLite. This is a pressure-minimum pointwise metric, not a cyclone-track or intensity score. Upper-tail temperature results are retained. The internal Eulerian-only comparison uses the same auxiliary objective; external comparisons span different architectures, objectives, and training costs.

The internal comparison shows a larger relative change on the selected tails than on full-field error. Adding the particle branch changes Z500 full-field RMSE from 178 to 168 m (5.6%) and upper-tail RMSE from 312 to 221 m (29.2%). For T2m, the corresponding changes are 1.76 to 1.68 K (4.5%) and 3.08 to 2.19 K (28.9%). For corrected lower-tail MSLP, the full-field error changes from 2.96 to 2.78 hPa (6.1%), while the lower-tail error changes from 7.1 to 5.1 hPa (28.2%). Thus, in this backbone and objective, the improvement is concentrated at the observed tail locations while average-field skill changes more modestly. These percentages describe the reported test scores rather than a causal decomposition of physical error sources.

5.5 Forecast skill across horizons and operating points

Figure 2 summarizes how forecast skill behaves across three orthogonal axes: lead time, the choice of representation versus auxiliary losses, and intensity preservation. Each metric is reported as mean $\pm$ standard deviation over three random seeds; error bars in the figure show one standard deviation.

The reported upper-tail Z500 advantage persists across the plotted lead times. The same-objective ablation in panel (b) improves from 312 to 221 m when the particle branch is added. This comparison also changes parameter count and training cost, so it does not by itself separate transport representation

Table 1: Five-day forecast error (three-seed means). Tail columns use observed upper 95th percentiles except MSLP, which uses the corrected lower 5th percentile from saved forecasts. All spatial scores are cosine-latitude weighted. Units: Z500 m, T2m K, MSLP hPa. A dash indicates that no corrected persistence value was supplied.

	Z500		T2m		MSLP	
Method	RMSE	Upper tail	RMSE	Upper tail	RMSE	Lower tail
Persistence	892	1247	4.21	5.83	12.4	–
ClimODE	187	342	1.87	3.42	3.12	7.9
DeepLag	179	298	1.79	2.98	2.94	6.9
PanguLite	174	284	1.72	2.84	2.89	6.7
Eulerian only	178	312	1.76	3.08	2.96	7.1
LE-NODE	**168**	**221**	**1.68**	**2.19**	**2.78**	**5.1**

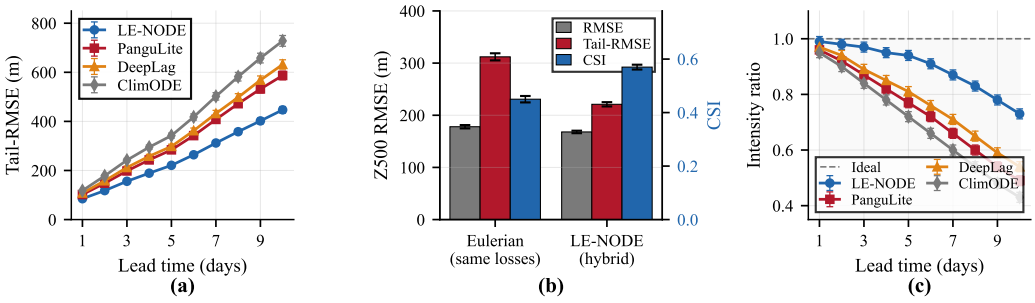


Figure 2: Forecast diagnostics, three-seed mean ± std. **(a)** Pointwise upper-tail Z500 RMSE versus lead time. **(b)** Same-objective Eulerian-only versus full LE-NODE; the 91 m gap measures the benefit of adding the particle branch within this backbone. **(c)** Pointwise upper-tail T2m intensity ratio; it is sensitive to event displacement.

from capacity and optimization. The T2m intensity ratio is consistent with improved pointwise amplitude at the tested horizons, but does not establish object-level intensity or displacement skill.

5.6 Ablation Study

The ablation compares the full model with an Eulerian-only version under the same objective and examines design choices within the hybrid model.

The hybrid model without auxiliary terms scores 254 m, while full LE-NODE scores 221 m. Comparing 254 m with the 312 m Eulerian-only model changes both architecture and objective; it cannot apportion the gain between them. Removing the extreme-event or proxy-drift penalties from the full hybrid worsens this metric to 267 m or 248 m, respectively. The discrete-RNN variant scores 263 m, suggesting continuous-time evolution is useful in this configuration; its comparison does not establish a general advantage over other hybrid methods.

The variants probe different parts of the design rather than forming an additive breakdown. The single-scale and fixed-threshold variants score 239 and 251 m, respectively, against 221 m for the full model. These results indicate that scale diversity and adaptive placement are useful within the implemented particle branch. The Lagrangian-only result, 287 m on the Z500 upper tail but 254 m on full-field RMSE, illustrates the different roles of the two states: sparse particles can carry localized information, whereas the grid retains the broad field needed for overall forecast accuracy. Because each variant is retrained, differences also include how the changed model is optimized. The most direct reported internal comparison remains the Eulerian-only versus full hybrid pair under the same auxiliary objective.

Table 2: Z500 ablation at five days (three-seed means). All tail scores refer to the observed upper Z500 tail. The left block mixes objective settings; only Eulerian-only versus full LE-NODE shares the same auxiliary objective.

Configuration	RMSE	Tail
Eulerian only	178	312
Lagrangian only	254	287
Hybrid, no aux. losses	174	254
DeepLag (discrete hybrid)	179	298
Full LE-NODE	**168**	**221**

Variant	RMSE	Tail	ΔTail
Single-scale particles	171	239	+18
Fixed spawning threshold	173	251	+30
Discrete RNN (vs. ODE)	176	263	+42
− conservation loss	172	248	+27
− extreme-event loss	169	267	+46
Full LE-NODE (ref.)	**168**	**221**	—

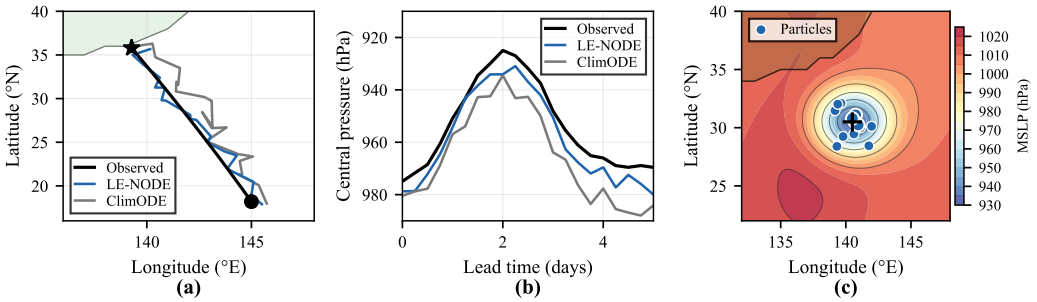


Figure 3: Typhoon Hagibis (October 2019). (a) Track: observed (black, 6-hourly), LE-NODE (blue), ClimODE (gray). (b) Central pressure. (c) Particle positions at 72 h overlaid on MSLP.

5.7 Efficiency, proxy drift, and compute

Across the tested resolutions, the particle count needed for a fixed pointwise upper-tail Z500 score increases from 192 ± 8 to 256 ± 12 as grid cell count rises roughly $30\times$. This empirical result does not guarantee the same scaling in dense convective or compound-extreme regimes.

The mass-, energy-, and enstrophy-like residuals are drift penalties on two-dimensional proxies, not full atmospheric conservation laws. They omit vertical structure, moisture, density weighting, and diabatic forcing. A divergence-free horizontal advecting velocity does not make the full learned forecast mass conserving.

The reported 72, 84, and 108 training hours for ClimODE, PanguLite, and LE-NODE are wall-clock hours per seed on eight A100 GPUs and correspond to 576, 672, and 864 GPU-hours. At equal 108-hour hardware time, ClimODE and PanguLite reported upper-tail Z500 RMSE of 339 ± 5 m and 279 ± 4 m, versus 221 ± 4 m for LE-NODE. This matches elapsed time on the specified hardware, while architectural and optimization differences remain.

5.8 Particle trajectory analysis

We use Typhoon Hagibis (October 2019) as an illustrative case study of learned particle behavior. Figure 3 compares predicted and observed tracks and intensities over the 5-day forecast window. LE-NODE spawns a cluster of 8–12 convective-scale particles near the typhoon center, which persist throughout the forecast. Particles track the observed position with mean error of 87 ± 6 km (3 seeds), compared to 142 ± 9 km for ClimODE. Central pressure stays within 6 hPa of observations at day 5, while ClimODE underestimates the cyclone depth by 18 hPa, consistent with the intensity-ratio analysis. The plotted trajectories illustrate that some particles follow this event; one case cannot establish population-level cyclone detection, track, or intensity skill.

6 DISCUSSION

What the comparisons establish. The strongest internal evidence is the pair that shares the LE-NODE backbone, training data, and auxiliary objective: Eulerian-only versus the full hybrid. The particle branch improves the reported upper-tail Z500, upper-tail T2m, and lower-tail MSLP errors by larger proportions than it improves full-field RMSE. The other ablations show how this branch behaves when its scale hierarchy, placement rule, temporal update, or loss terms are changed. DeepLag and PanguLite place the result in the context of other architectures, while their different objectives and costs make those comparisons broader performance benchmarks. These complementary views locate the gain within the tested system without assigning an independent percentage to each possible cause of extreme forecast error. Comparator choice changes the numerical headline: the five-day Z500 tail gap is 91 m against the internal Eulerian-only ablation and 63 m against PanguLite. The first contrast holds the auxiliary objective fixed within the same backbone; the second compares complete systems with different architectures and training costs. Reporting both makes clear which conclusion each comparison supports.

Why the evaluation uses several views. Full-field RMSE measures the average error budget across the global grid. Observation-conditioned Tail-RMSE emphasizes the grid locations and times at which a specified variable is unusually high or low. CSI adds a thresholded occurrence view, and the intensity-ratio and Hagibis displays illustrate amplitude and trajectory behavior in their stated settings. Agreement across these views is more informative than any single score because an apparent intensity improvement can coincide with a spatial offset, while a smooth forecast may obtain a favorable global RMSE by suppressing sharp features. The corrected lower-tail MSLP calculation is particularly important: high pressure and cyclone-core low pressure occupy opposite sides of the variable's climatology. Keeping the tail direction explicit allows the pressure result to be interpreted consistently with the event motivating the model.

Use of a sparse state. The grid and particles serve different parts of the forecast. A full grid is needed to represent large-scale circulation and to provide local context for particle updates; the particle state concentrates additional computation around selected anomalies. In the tested resolution sweep, the particle count required for a fixed tail score grows much more slowly than the grid cell count. This is an empirical operating point rather than a guarantee about arbitrary resolutions or event densities. The additional training and inference cost should be weighed against the tail-score improvement for the forecast task at hand. These observations motivate evaluating the hybrid representation where localized sharp structures matter, while preserving the grid pathway that carries the rest of the atmospheric state. Selection is refreshed at each segment boundary, allowing the limited particle budget to follow changes in where anomalies occur during a rollout. A structure missed by the extremity score remains represented by the grid until a later update.

7 CONCLUSION

LE-NODE combines a grid-based neural ODE with a sparse, adaptive particle state. In the tested ERA5 setting it improves pointwise upper-tail Z500 and temperature scores and the corrected lower-tail MSLP score. The same-objective Eulerian-only ablation supports a benefit from adding the particle branch to this backbone. Across the tested variables, tail-error reductions are larger than the corresponding full-field reductions. This pattern makes the sparse branch relevant when a forecast task places extra weight on localized errors while still requiring a global field. Refreshing the continuous-time particle state between forecast segments provides a concrete way to allocate that additional capacity. The synthetic initial-mode probe does not identify numerical dissipation under spatially varying flow, and existing experiments do not determine whether improved Eulerian transport, a different objective, or a larger backbone could reproduce the gains.

Limitations. The model is two-dimensional and deterministic. Pointwise tail metrics can double-penalize displacement; the Hagibis visualization is a single case. External baselines have different objectives and training costs, three seeds do not measure forecast-case uncertainty, and high-order/semi-Lagrangian Eulerian controls, neighborhood verification, and all-storm cyclone evaluation are not available. The required particle budget may change in denser extreme-event regimes.

AI USE STATEMENT

A generative AI assistant helped revise manuscript wording, check numerical statements against the existing manuscript and author response, and convert the document to the ICLR template. No experimental outputs were generated by the assistant. The authors are responsible for verifying all assisted changes and the final submission.

REPRODUCIBILITY STATEMENT

The data split, predicted variables, architecture, training schedule, evaluation masks, and limitations of the available comparisons are specified in Sections 4–5 and Appendix F.

REFERENCES

Zied Ben Bouallègue, Mariana CA Clare, Linus Magnusson, Estibaliz Gascon, Michael Maier-Gerber, Martin Janoušek, Mark Rodwell, Florian Pinault, Jesper S Dramsch, Simon TK Lang, et al. The rise of data-driven weather forecasting: A first statistical assessment of machine learning–based weather forecasts in an operational-like context. *Bulletin of the American Meteorological Society*, 105(6):E864–E883, 2024.

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.

Massimo Bonavita. On some limitations of current machine learning weather prediction models. *Geophysical Research Letters*, 51(12):e2023GL107377, 2024.

John P Boyd. *Chebyshev and Fourier Spectral Methods*. Dover Publications, 2001.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *NeurIPS*, volume 31, 2018.

Hwangyong Choi, Jeongwhan Choi, Jeehyun Hwang, Kookjin Lee, Dongeun Lee, and Noseong Park. Climate modeling with neural advection–diffusion equation. *Knowledge and Information Systems*, 65(6):2403–2427, 2023.

Daizong Ding, Mi Zhang, Xudong Pan, Min Yang, and Xiangnan He. Modeling extreme events in time series prediction. In *KDD*, pp. 1114–1122, 2019.

Dale R Durran. *Numerical methods for fluid dynamics: With applications to geophysics*, volume 32. Springer Science & Business Media, 2010.

Ernst Hairer, Gerhard Wanner, and Syvert P Nørsett. *Solving ordinary differential equations I: Nonstiff problems*. Springer, 1993.

Rixu Hao, Yuxin Zhao, Shaoqing Zhang, and Xiong Deng. Deep learning for ocean forecasting: A comprehensive review of methods, applications, and datasets. *IEEE Transactions on Cybernetics*, 2025.

Francis H Harlow. The particle-in-cell method for numerical solution of problems in fluid dynamics. Technical report, Los Alamos Scientific Lab., N. Mex., 1962.

Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, et al. The era5 global reanalysis. *Quarterly journal of the royal meteorological society*, 146(730):1999–2049, 2020.

Kevin I Hodges, Robert W Lee, and Lennart Bengtsson. A comparison of extratropical cyclones in recent reanalyses era-interim, nasa merra, ncep cfsr, and jra-25. *Journal of Climate*, 24(18): 4888–4906, 2011.

James R Holton and Gregory J Hakim. *An Introduction to Dynamic Meteorology*. Academic Press, 5 edition, 2013.

Jeehyun Hwang, Jeongwhan Choi, Hwangyong Choi, Kookjin Lee, Dongeun Lee, and Noseong Park. Climate modeling with neural diffusion equations. In *ICDM*, pp. 230–239. IEEE, 2021.

Ethan Ji, Yuanzhou Chen, Arush Ramteke, Fang Sun, Tianrun Yu, Jai Parera, Wei Wang, and Yizhou Sun. Corgi: Gnns with convolutional residual global interactions for lagrangian simulation. *arXiv preprint arXiv:2511.22938*, 2025.

Dmitrii Kochkov, Janni Yuval, Ian Langmore, Peter Norgaard, Jamie Smith, Griffin Mooers, Milan Klöwer, James Lottes, Stephan Rasp, Peter Düben, et al. Neural general circulation models for weather and climate. *Nature*, 632(8027):1060–1066, 2024.

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.

Randall J LeVeque. *Finite volume methods for hyperbolic problems*, volume 31. Cambridge university press, 2002.

Lizao Li, Robert Carver, Ignacio Lopez-Gomez, Fei Sha, and John Anderson. Generative emulation of weather forecast ensembles with diffusion models. *Science Advances*, 10(13):eadk4489, 2024.

Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, Anima Anandkumar, et al. Fourier neural operator for parametric partial differential equations. In *ICLR*, 2021.

Björn List, Li-Wei Chen, and Nils Thuerey. Learned turbulence modelling with differentiable fluid solvers: physics-based loss functions and optimisation horizons. *Journal of Fluid Mechanics*, 949: A25, 2022.

Peiyuan Liu, Tian Zhou, Liang Sun, and Rong Jin. Mitigating time discretization challenges with weatherode: A sandwich physics-driven neural ode for weather forecasting. *arXiv preprint arXiv:2410.06560*, 2024.

Qilong Ma, Haixu Wu, Lanxiang Xing, Shangchen Miao, and Mingsheng Long. Deeplag: Discovering deep lagrangian dynamics for intuitive fluid prediction. In *NeurIPS*, volume 37, pp. 122833–122866, 2024.

Cristian Meo, Ankush Roy, Mircea Lică, Junzhe Yin, Zeineb Bou Che, Yanbo Wang, Ruben Imhoff, Remko Uijlenhoet, and Justin Dauwels. Extreme precipitation nowcasting using transformer-based generative models. *arXiv preprint arXiv:2403.03929*, 2024.

Joe J Monaghan. Smoothed particle hydrodynamics. *Reports on progress in physics*, 68(8):1703–1759, 2005.

Urs Neu, Mirseid G Akperov, Nina Bellenbaum, Rasmus Benestad, Richard Blender, Rodrigo Caballero, Angela Cocozza, Helen F Dacre, Yang Feng, Klaus Fraedrich, et al. Imilast: A community effort to intercompare extratropical cyclone detection and tracking algorithms. *Bulletin of the American Meteorological Society*, 94(4):529–547, 2013.

Tung Nguyen, Johannes Brandstetter, Ashish Kapoor, Jayesh K Gupta, and Aditya Grover. Climax: A foundation model for weather and climate. In *ICML*, pp. 25904–25938. PMLR, 2023.

Isidoro Orlanski. A rational subdivision of scales for atmospheric processes. *Bulletin of the American Meteorological Society*, 56:527–530, 1975.

Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, et al. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, et al. Gencast: Diffusion-based ensemble forecasting for medium-range weather. In *105th AMS Annual Meeting*. AMS, 2025.

Evan Racah, Christopher Beckham, Tegan Maharaj, Samira Ebrahimi Kahou, Mr Prabhat, and Chris Pal. Extremeweather: A large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events. In *NeurIPS*, volume 30, 2017.

Stephan Rasp, Peter D Dueben, Sebastian Scher, Jonathan A Weyn, Soukayna Mouatadid, and Nils Thuerey. Weatherbench: a benchmark data set for data-driven weather forecasting. *Journal of Advances in Modeling Earth Systems*, 12(11):e2020MS002203, 2020.

Stephan Rasp, Stephan Hoyer, Alexander Merose, Ian Langmore, Peter Battaglia, Tyler Russell, Alvaro Sanchez-Gonzalez, Vivian Yang, Rob Carver, Shreya Agrawal, et al. 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.

Suman Ravuri, Karel Lenc, Matthew Willson, Dmitry Kangin, Remi Lam, Piotr Mirowski, Megan Fitzsimons, Maria Athanassiadou, Sheleem Kashem, Sam Madge, et al. Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 597(7878):672–677, 2021.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pp. 234–241. Springer, 2015.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *ICML*, pp. 8459–8468, 2020.

Omer Rochman Sharabi, Sacha Lewin, and Gilles Louppe. A neural material point method for particle-based emulation. *arXiv preprint arXiv:2408.15753*, 2024.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NeurIPS*, volume 28, 2015.

Andrew Staniforth and Jean Côté. Semi-lagrangian integration schemes for atmospheric models—a review. *Monthly Weather Review*, 119(9):2206–2223, 1991.

Artur Toshev, Gianluca Galletti, Fabian Fritz, Stefan Adami, and Nikolaus Adams. Lagrangebench: A lagrangian fluid mechanics benchmarking suite. In *NeurIPS*, volume 36, pp. 64857–64884, 2023.

Artur P Toshev, Jonas A Erbesdobler, Nikolaus A Adams, and Johannes Brandstetter. Neural sph: Improved neural modeling of lagrangian fluid dynamics. In *ICLR*, pp. 48428–48452, 2024.

Thomas Vandal, Evan Kodra, Sangram Ganguly, Andrew Michaelis, Ramakrishna Nemani, and Auroop R Ganguly. Deepsd: Generating high resolution climate change projections through single image super-resolution. In *KDD*, pp. 1663–1672, 2017.

Yogesh Verma, Markus Heinonen, and Vikas Garg. Climode: Climate and weather forecasting with physics-informed neural odes. In *ICLR*, pp. 1–23, 2024.

Peter AG Watson. Machine learning applications for weather and climate need greater focus on extremes. *Environmental Research Letters*, 17(11):111004, 2022.

Yuchen Zhang, Mingsheng Long, Kaiyuan Chen, Lanxiang Xing, Ronghua Jin, Michael I Jordan, and Jianmin Wang. Skilful nowcasting of extreme precipitation with nowcastnet. *Nature*, 619 (7970):526–532, 2023.

A Diagnostic experiment: setup and additional results

This appendix details the controlled diagnostic of Section 3.1.

A.1 Synthetic dataset

We generate divergence-free velocity fields $\mathbf{v}(\mathbf{x}, t)$ on $\Omega = [0, 2\pi]^2$ as the curl of a stream function $\psi(\mathbf{x}, t)$ that is itself a sum of Fourier modes with random amplitudes and slowly drifting phases. This gives a spatially varying incompressible flow; exact trajectories can be approximated numerically for reference, but individual Fourier-mode amplitudes are not invariant. Initial scalar fields u_0 are sampled from one of two distributions: (i) single-mode probes $u_0 = \cos(\mathbf{k} \cdot \mathbf{x})$ used for the wavenumber sweep, and (ii) multi-mode random fields with prescribed power spectra used for training. Reference fields are generated by integrating trajectories backward from query points and evaluating u_0 at their starting positions. Their numerical accuracy is separate from the initial-mode coefficient diagnostic.

A.2 Architectures swept

We train three architectures with comparable parameter counts:

- CNN: a 7-block residual convolutional network with kernel size 5 and 128 channels.
- FNO: 4 Fourier layers with 32 retained modes per dimension and width 64.
- U-Net: 4 encoder/decoder levels with channel progression 64–128–256–512.

Each is trained as a neural ODE with DOPRI5 integration (tol 10^{-5}) on 50k random initial conditions sampled freshly each epoch (reducing finite-sample scarcity in this task). The loss is full-field MSE.

A.3 Initial-mode coefficient behavior

For the three architectures at $\Delta x = 2\pi/64$, the reported ratio A stays near one for $\kappa < 0.3$ and falls below 0.5 at $\kappa = 0.8$ after ten advection times. Doubling width or adding two residual blocks changes this coefficient ratio by less than 0.05 in the reported sweep. Repeating at $\Delta x \in \{2\pi/32, 2\pi/64, 2\pi/128\}$ yields similar curves against $\kappa = |\mathbf{k}|\Delta x$. Since the reference field can transfer spectral energy away from the initial mode, these observations do not quantify numerical dissipation or predict ERA5 tail error. A valid error diagnostic would compare predictions with the reference at the same time, including global norm and shell-energy differences.

B Proof of Proposition 3.1

Conservation along characteristics. Let $\mathbf{p}_k(t)$ satisfy $\dot{\mathbf{p}}_k = \mathbf{v}(\mathbf{p}_k, t)$ with $\mathbf{p}_k(0) = \mathbf{p}_k^0$. Define $f(t) = u(\mathbf{p}_k(t), t)$. By the chain rule,

$$\frac{df}{dt} = \nabla u \cdot \dot{\mathbf{p}}_k + \frac{\partial u}{\partial t} = \nabla u \cdot \mathbf{v} + \frac{\partial u}{\partial t} = 0$$

where the last equality uses (1). Therefore $f(t) = f(0)$, that is, $u(\mathbf{p}_k(t), t) = u(\mathbf{p}_k^0, 0)$.

Numerical integration error. Let $\mathbf{v}$ be Lipschitz with constant L_v on the relevant domain. The Picard–Lindelöf theorem guarantees a unique solution to $\dot{\mathbf{p}}_k = \mathbf{v}(\mathbf{p}_k, t)$ over any finite interval $[0, T]$. For a q-th order Runge–Kutta scheme with step Δt, classical results (Hairer et al., 1993, §5.3) give the global error bound

$$\|\hat{\mathbf{p}}_k(T) - \mathbf{p}_k(T)\| \leq C(L_v, T)\Delta t^q,$$

under the standard additional smoothness and stability assumptions for the chosen integrator. The constant is independent of spatial grid spacing only when the prescribed velocity is evaluated independently of a grid.

When $\mathbf{v}$ is itself given on a grid and interpolated to $\mathbf{p}_k$, an interpolation error of order $\mathcal{O}(\Delta x^p)$ enters; this is a separate effect from the integration error and must be controlled separately. For LE-NODE, learned-velocity error, interpolation error, evolving intensity, and splatting enter in addition

to integration error. Appendix C reports an interpolation sensitivity check, not a bound on their combined effect.

C INTERPOLATION SENSITIVITY

On the synthetic advection task, changing velocity interpolation from linear to cubic reduces the reported particle-position error at $T = 5\tau$ by roughly fourfold; changing from cubic to quintic gives less than 5% further reduction. The experiment supports the choice of cubic interpolation at this tested setting. It does not identify the remaining error as solely ODE integration error or remove learned-velocity and splatting errors.

D TOWARD A 3D EXTENSION

The current implementation treats horizontal advection on isobaric surfaces. A 3D extension would add a vertical particle position z_k and a vertical velocity w_k governed by the omega equation or a learned drift. The kernel would become anisotropic (σ_h, σ_v). Spawning would need to identify deep convective columns rather than pixel-level extremes, possibly using vertical integrals of moisture or buoyancy as the extremity signal. We do not run this experiment in the present paper; we sketch it here to indicate that the framework extends naturally.

E ADDITIONAL EXPERIMENTAL RESULTS

E.1 PER-VARIABLE BREAKDOWN

Table 3 reports five-day pointwise scores for the six predicted variables. Z500, temperatures, and wind components use upper-tail thresholds; MSLP uses the corrected lower-tail threshold. Wind-component tails do not measure wind-speed extremes.

The upper-tail Z500 and temperature columns use the original pointwise results. The MSLP entries use lower-tail scores recomputed from saved forecasts. U10 and V10 columns retain their original component-wise upper-tail definitions; they should not be interpreted as severe-wind verification.

E.2 SEASONAL BREAKDOWN

Table 4 reports Tail-RMSE for Z500 stratified by meteorological season, enabling assessment of model performance under varying synoptic regimes.

Across the reported seasonal upper-tail Z500 scores, LE-NODE has lower error than PanguLite in each season. These are pooled pointwise scores; case-level uncertainty and event-type stratification were not computed.

E.3 HYPERPARAMETER SENSITIVITY

Table 5 reports sensitivity of Z500 Tail-RMSE to two key hyperparameters: maximum particle count and extreme event loss weight α.

The particle count analysis reveals a clear relationship between representational capacity and extreme event skill. Increasing the maximum particle count from 128 to 512 yields monotonic improvement in Tail-RMSE, with the most substantial gain occurring between 128 and 256 particles (16 m reduction). This pattern suggests that a minimum particle budget is necessary to adequately cover the spatial distribution of extreme events at any given time. With 128 particles distributed across three scales, the model has approximately 40 particles per scale, which may be insufficient to track multiple simultaneous extreme systems across the global domain.

Beyond 512 particles, diminishing returns are evident: doubling the count to 1024 yields only 3 m additional improvement while substantially increasing computational cost. The small further change in this metric suggests diminishing returns at the tested setting. The required particle count may differ at higher resolution or with denser extreme-event occurrence.

Table 3: Five-day pointwise metrics. Tail means observed upper 95th percentile except MSLP (lower 5th percentile). MSLP CSI is unavailable under the corrected event definition. Wind entries are component-wise, not wind-speed, scores.

Variable	Method	RMSE	ACC	Tail-RMSE	CSI
Z500 (m)	ClimODE	187	0.923	342	0.41
	DeepLag	179	0.931	298	0.47
	PanguLite	174	0.936	284	0.48
	LE-NODE	**168**	**0.942**	**221**	**0.57**
T850 (K)	ClimODE	2.14	0.887	3.78	0.37
	DeepLag	2.02	0.901	3.34	0.42
	PanguLite	1.96	0.908	3.21	0.44
	LE-NODE	**1.89**	**0.914**	**2.54**	**0.53**
T2m (K)	ClimODE	1.87	0.891	3.42	0.38
	DeepLag	1.79	0.904	2.98	0.44
	PanguLite	1.72	0.911	2.84	0.45
	LE-NODE	**1.68**	**0.917**	**2.19**	**0.55**
MSLP (hPa)	ClimODE	3.12	0.912	7.9	–
	DeepLag	2.94	0.924	6.9	–
	PanguLite	2.89	0.929	6.7	–
	LE-NODE	**2.78**	**0.938**	**5.1**	–
U10 (m/s)	ClimODE	2.43	0.842	4.21	0.34
	DeepLag	2.31	0.858	3.87	0.39
	PanguLite	2.26	0.864	3.72	0.41
	LE-NODE	**2.18**	**0.873**	**2.98**	**0.49**
V10 (m/s)	ClimODE	2.38	0.848	4.12	0.35
	DeepLag	2.27	0.861	3.79	0.40
	PanguLite	2.21	0.868	3.64	0.42
	LE-NODE	**2.14**	**0.876**	**2.91**	**0.50**

Table 4: Seasonal Tail-RMSE for Z500 at 5-day lead time.

Method	DJF	MAM	JJA	SON
ClimODE	378	324	298	367
DeepLag	329	281	261	321
PanguLite	312	268	254	302
LE-NODE	**243**	**208**	**197**	**236**

In this sweep, the extreme weight α gives the lowest reported upper-tail Z500 RMSE at $\alpha = 5.0$. Lower values ($\alpha = 1.0, 2.0$) provide insufficient emphasis on extreme events during training, resulting in models that optimize primarily for average conditions. Higher values ($\alpha = 10.0$) overcorrect, potentially causing the model to sacrifice accuracy on moderate anomalies to focus excessively on the most extreme cases. The slight performance degradation at $\alpha = 10.0$ may also reflect increased gradient variance during training, as the loss becomes dominated by rare, high-magnitude samples. The default value $\alpha = 5.0$ achieves an effective balance, providing strong extreme event emphasis while maintaining stable optimization dynamics.

F Implementation details

F.1 Network architectures

Eulerian U-Net. The encoder follows a standard contracting path with four resolution levels. Each encoder block applies two consecutive convolutional layers with kernel size 3 and padding 1, where each convolution is followed by GroupNorm with 8 groups and GELU activation. Spatial

Table 5: Hyperparameter sensitivity for Z500 Tail-RMSE at 5-day lead time.

Max particles	Tail-RMSE	Extreme weight α	Tail-RMSE
128	248	1.0	254
256	232	2.0	238
512 (default)	221	5.0 (default)	221
1024	218	10.0	224

downsampling is performed via average pooling with kernel size 2. The channel dimensions progress as 64, 128, 256, and 512 through the encoder stages, providing increasing representational capacity at coarser spatial resolutions where the receptive field encompasses larger atmospheric structures.

The decoder mirrors the encoder architecture with bilinear upsampling replacing pooling operations. Skip connections concatenate encoder features at matching resolutions, preserving fine-grained spatial information that would otherwise be lost through the bottleneck. The final output layer projects to the number of predicted variables using a 1x1 convolution.

Fourier neural operator layers are inserted at the two coarsest resolution levels (corresponding to spatial dimensions of approximately 4 and 8 degrees). These layers transform the spatial representation to spherical harmonic coefficients up to degree 128 using the `torch-harmonics` library, apply learnable spectral filtering in the coefficient domain, and transform back to physical space. This design enables efficient modeling of global-scale teleconnections that would require prohibitively large convolutional kernels to capture directly.

Particle networks. The particle dynamics networks share a common trunk architecture to promote feature reuse and reduce parameter count. The trunk consists of three fully-connected layers with 256 hidden units each. Layer normalization is applied after each linear transformation, followed by ReLU activation. The input to the trunk concatenates the particle state components (intensity, scale, flattened latent vector) with the local feature vector extracted from the Eulerian grid.

Three separate output heads branch from the shared trunk to predict the time derivatives of intensity, scale, and latent state respectively. The intensity head produces a single scalar output without activation, allowing both positive and negative intensity changes. The scale head also produces a single scalar but applies softplus activation to ensure non-negative scale values. The latent head outputs a 64-dimensional vector matching the latent state dimension.

Cross-scale attention. Information exchange between particles at different scales is mediated by a single transformer layer. The attention mechanism operates over the union of all particle latent states, with queries, keys, and values computed via learned linear projections. We use 4 attention heads with embedding dimension 64, yielding 16-dimensional projections per head. The feedforward sublayer consists of a two-layer MLP with hidden dimension 256 and GELU activation. Layer normalization and residual connections follow the standard transformer architecture.

Local feature extraction. The local feature vector $\boldsymbol{\phi}_k$ summarizes Eulerian information near particle k:

$$\boldsymbol{\phi}_k = \sum_{i \in \mathcal{N}(\mathbf{p}_k)} \alpha_{ki} \, \mathrm{MLP}\big(u^{(i)}_{\text{grid}}, \mathbf{x}_i - \mathbf{p}_k\big), \tag{9}$$

where $\mathcal{N}(\mathbf{p}_k)$ contains grid points within a fixed radius (six grid cells in our experiments) and the attention weights α_{ki} are computed via softmax over inverse distances. The MLP is two layers with hidden dimension 64 and GELU activation. The same MLP is shared across particle scales.

ODE solver. Both Eulerian and Lagrangian ODEs are integrated jointly using the Dormand–Prince adaptive method (DOPRI5) implemented in `torchdiffeq`, with the adjoint method for memory-efficient backpropagation. Tolerances are 10^{-5} (relative) and 10^{-6} (absolute). During training we integrate from $t = 0$ to target lead times of 24 to 120 hours, sampled uniformly per batch.

F.2 TRAINING PROCEDURE

Training proceeds in three phases designed to promote stable convergence and strong extreme event performance. The warmup phase spans 5 epochs during which the learning rate increases linearly from 10^{-6} to 10^{-4}. This gradual warmup prevents early training instabilities that can arise from large gradient magnitudes during ODE integration with randomly initialized networks.

The main training phase continues for 90 epochs using cosine annealing to decay the learning rate from 10^{-4} to 10^{-6}. This schedule provides sustained learning capacity during the middle epochs while gradually reducing the learning rate to enable fine-grained optimization in later stages. The extreme event loss weight is fixed at $\alpha = 5.0$ throughout this phase.

The final fine-tuning phase spans 5 epochs at a fixed learning rate of 10^{-6} with increased extreme weight $\alpha = 10.0$. This phase refines the model specifically for extreme event prediction after the overall dynamics have stabilized. The elevated α value during fine-tuning provides additional emphasis on tail performance without the training instabilities that would result from using this value throughout training.

Gradient clipping with maximum norm 1.0 is applied throughout all training phases to prevent instabilities arising from the ODE integration. Such instabilities can occur when the learned dynamics produce rapid state changes that cause the adaptive integrator to take many small steps, amplifying gradient magnitudes during backpropagation through the solve.

We use the AdamW optimizer with weight decay 10^{-5} and default momentum parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$). Training is distributed across 8 NVIDIA A100 GPUs using PyTorch DistributedData-Parallel with a per-GPU batch size of 4, yielding an effective batch size of 32. The complete training procedure requires approximately 100 hours.

F.3 DATA AUGMENTATION

We apply two data augmentation strategies during training to improve generalization and reduce overfitting. Random horizontal flips reverse the longitude dimension with probability 0.5, effectively mirroring the atmospheric state about a meridian. This augmentation exploits the approximate zonal symmetry of large-scale atmospheric dynamics while doubling the effective training set size. Wind components are negated appropriately to maintain physical consistency under reflection.

Random temporal shifts perturb the initial condition time by up to 6 hours in either direction, sampled uniformly. This augmentation reduces sensitivity to the specific 6-hourly sampling of the ERA5 data and encourages the model to learn dynamics that generalize across the diurnal cycle. Target fields are adjusted correspondingly to maintain the specified lead time.

Latitude flipping is deliberately excluded from the augmentation pipeline. Unlike longitude, latitude reversal would invert the Coriolis parameter, fundamentally altering the rotational dynamics of the atmosphere. Cyclonic systems rotate counterclockwise in the Northern Hemisphere and clockwise in the Southern Hemisphere due to the sign of the Coriolis force, and this asymmetry must be preserved during training.