DO NEURAL PDE SOLVERS LEARN THE RIGHT DYNAMICS?

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural PDE solvers can achieve low prediction errors, but do they reproduce the dynamics of the systems they model? Prediction scores alone offer an incomplete answer: they measure agreement with reference solutions but provide limited insight into how errors accumulate, nearby states diverge, or extreme events arise. We propose an evaluation framework that directly examines these behaviors in deterministic and stochastic neural solvers. By evolving ensembles of nearby initial states and comparing them with direct numerical simulation, we assess three complementary aspects of learned dynamics: **error formation, ensemble geometry, and extreme events**. Experiments on two-dimensional Kolmogorov flow reveal limitations that conventional scores can obscure. Smaller trajectory errors can reflect weaker error amplification despite less accurate local updates. Models can match an ensemble's overall spread and effective dimension while failing to capture the spatial directions where nearby states diverge. Similarly, matching overall event frequencies can conceal failures to predict persistent extreme events. *These findings show that improved prediction accuracy does not necessarily imply greater dynamical fidelity.* Our framework makes this distinction measurable, providing concrete criteria for evaluating whether advances in neural PDE solvers better capture the underlying dynamics.

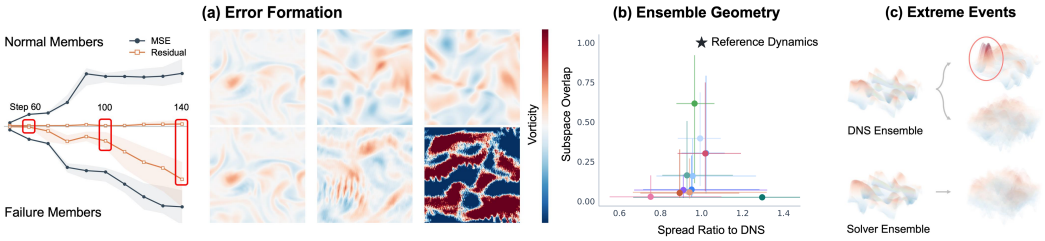

Figure 1: **Probing learned dynamics beyond prediction error.** Ensembles of nearby initial states expose three distinct failures in neural solvers of two-dimensional Kolmogorov flow. **(a) Error formation:** larger one-step dynamical errors distinguish members that later fail even while their trajectory errors remain similar to those of non-failing members. The three columns of vorticity fields on the right correspond to the boxed forecast steps (60, 100, and 140) on the left and show their subsequent divergence. **(b) Ensemble geometry:** an ensemble spread close to DNS can coexist with low overlap between the dominant solver and reference subspaces. **(c) Extreme events:** nearby initial states can lead to distinct event and non-event outcomes under the reference dynamics, while a neural solver can miss the event-producing branch.

1 INTRODUCTION

Neural partial differential equation (PDE) solvers have shown considerable success in predicting the evolution of physical systems. Recent pretrained models, including MPP, DPOT, and Poseidon, extend this capability across equations (McCabe et al., 2024; Hao et al., 2024; Herde et al., 2024). Progress is primarily measured through pointwise field errors, with studies of chaotic systems and generative models also employing distributional distances and probabilistic scores (Schiff et al.,

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.

2024; Molinaro et al., 2024; Rühling Cachay et al., 2023). These evaluations quantify agreement with reference solutions, but leave a fundamental unsolved question:

Do neural PDE solvers learn the right dynamics?

Prediction accuracy alone provides an incomplete answer. During a rollout, each model update introduces errors that interact with discrepancies inherited from earlier steps. A smaller trajectory error may therefore reflect more accurate updates, weaker amplification of existing errors, or cancellation between the two. Beyond individual trajectories, the dynamics determine how nearby initial states separate and which spatial patterns dominate their evolution. These responses are not established by matching an ensemble's overall spread. Similarly, agreement in aggregate event frequency can conceal errors under particular initial conditions, while aggregate distributional scores may inadequately expose relative errors in rare-event probabilities (Gneiting & Ranjan, 2011). Understanding learned dynamics thus requires examining how predictions evolve and which properties they preserve.

In this work, we investigate three questions: *How do model errors form and propagate? Do nearby initial states evolve along the same directions as in the reference system? Do extreme events occur with the right intensities and under the right initial conditions?* We study these questions through a common experimental protocol: densely sample perturbed initial states around the initial state of each case, evolve each ensemble using both the neural solver and direct numerical simulation (DNS), and compare their evolution. These perturbations serve as controlled probes of the learned dynamics; they do not presuppose uncertainty in the initial conditions. The protocol accommodates deterministic and stochastic solvers. For stochastic solvers, ensemble members share a random path, with fresh randomness at successive steps, so that differences across members reflect their responses to initial-state perturbations under the same realization of model randomness.

Our experiments on deterministic chaotic two-dimensional Kolmogorov flow identify three discrepancies between predictive performance and dynamical fidelity.

Error formation: lower trajectory error need not imply more accurate local updates. We decompose error at successive updates into newly introduced model residuals, propagation of existing discrepancies, and their interaction. The analysis reveals cases in which persistently larger local residuals coexist with lower trajectory errors because existing discrepancies undergo weaker dynamical amplification. The residuals also alter the collective evolution of nearby states: early in prediction, they reduce differences between ensemble members relative to DNS advancement from the same model states, while increasing the error of the centered ensemble against the DNS reference. Thus, keeping trajectories closer together can accompany a less faithful response to initial-state perturbations.

Ensemble geometry: matching ensemble spread and dimension does not ensure the correct spatial directions. As the DNS ensemble evolves, its variance first concentrates into a few directions and later spreads across more directions, while the dominant directions themselves change. Models reproduce the broad evolution of total variance and effective dimension, yet most tested configurations show substantial misalignment with the dominant DNS subspace at intermediate prediction times. This discrepancy persists even when variance and effective dimension closely match DNS. Capturing how much an ensemble spreads and how many directions it occupies therefore does not establish that it evolves along the correct directions.

Extreme events: correct aggregate event frequencies can conceal incorrect conditional probabilities. Models whose overall extreme-event frequencies closely match DNS can substantially overestimate or underestimate event probabilities within individual initial-state ensembles. These errors cancel upon aggregation, allowing apparently correct overall frequencies to conceal failures to predict extreme events that persist under the reference dynamics. Even when predicted vortices remain recognizable and have similar peak intensities, their deformation and rotation can differ substantially from the reference dynamics.

Together, these findings show how favorable prediction scores can coexist with systematic errors in learned dynamics. Our investigation complements conventional evaluation by tracing error formation, testing the geometry of nearby trajectories, and resolving extreme-event statistics by initial condition. These analyses provide concrete criteria for assessing whether improvements in neural PDE solvers correspond to a more faithful representation of the underlying system.

2 EXPERIMENTAL SETUP AND DYNAMICAL PROBES

We probe learned dynamics by evolving ensembles of nearby initial states with both direct numerical simulation (DNS) and neural solvers. This allows us to examine individual prediction errors alongside the collective response to initial-state perturbations. All three investigations share the same initial states, data split, and prediction origin.

Reference Dynamics. We consider two-dimensional Kolmogorov flow, governed by the incompressible Navier–Stokes equations with sinusoidal forcing and periodic boundaries. Although deterministic, this system exhibits chaotic evolution and develops localized intense vortices (Vela-Martín & Avila, 2024).

Dataset Construction and Initial-State Probes. We sample temporally decorrelated base states from the stationary-flow archive of Vela-Martín & Avila (2024), with each state defining a case c. Spatially structured perturbations, constructed from rescaled, randomly signed differences between archive states, define nearby ensemble members m. We denote their vorticity fields by $x^q_{c,m}(t) \in \mathbb{R}^D$, where q identifies the solver and D is the spatial dimension. Cases are split into training, validation, and test sets. Training and validation use only unperturbed trajectories; ensemble diagnostics use perturbed test members, each with a matched DNS reference. Appendix C details the numerical settings, dataset sizes, forecast initialization, and stochastic sampling protocol.

Solvers and Forecast Evaluation. We evaluate configurations of U-Net (Ronneberger et al., 2015), FNO (Li et al., 2020), AFNO (Guibas et al., 2021), DPOT (Hao et al., 2024), DySLIM (Schiff et al., 2024), PDE-Refiner (Lippe et al., 2023), ACDM (Kohl et al., 2026), PFNO, and DYffusion (Rühling Cachay et al., 2023). Configuration labels specify the input history length H, output block length P, and training unroll length K. We exclude configurations that perform poorly on conventional forecast metrics from dynamical evaluation; Appendix D provides the screening details.

3 DOES LOWER TRAJECTORY ERROR IMPLY MORE ACCURATE LOCAL UPDATES?

Trajectory error combines errors introduced by individual updates with the subsequent evolution of existing discrepancies. We first separate these contributions to examine whether lower forecast error reflects more accurate local updates. We then ask whether training that improves forecast accuracy also preserves the response to nearby initial states.

3.1 LOWER TRAJECTORY ERROR CAN CONCEAL LARGER RESIDUALS

At each step k, we restart DNS from both the solver state $\hat{x}_k$ and its paired reference state x_k. Let f denote DNS advancement over the probed interval, with $x_{k+1} = f(x_k)$ (derivation in Appendix E.1). We define

$$e_k = \hat{x}_k - x_k, \qquad h_k = f(\hat{x}_k) - f(x_k), \qquad r_k = \hat{x}_{k+1} - f(\hat{x}_k). \tag{1}$$

Here, e_k is the current prediction error, h_k is that discrepancy after propagation under reference dynamics, and r_k compares the solver update with reference evolution from the same solver state. Thus, r_k measures local accuracy on states reached during autoregressive rollout.

Since $e_{k+1} = h_k + r_k$, the change in squared error satisfies

$$\|e_{k+1}\|^2_{\mathrm{sp}} - \|e_k\|^2_{\mathrm{sp}} = \underbrace{\|h_k\|^2_{\mathrm{sp}} - \|e_k\|^2_{\mathrm{sp}}}_{g_k} + \underbrace{\|r_k\|^2_{\mathrm{sp}}}_{j_k} + \underbrace{2\langle h_k, r_k\rangle_{\mathrm{sp}}}_{c_k}. \tag{2}$$

The spatial norm is defined by $\|v\|^2_{\mathrm{sp}} = D^{-1}\sum_{i=1}^{D} v_i^2$. The three terms quantify net propagation of existing error (g_k), squared local residual (j_k), and directional coupling between the two (c_k). Propagation can amplify or reduce an existing discrepancy, while coupling can reinforce or offset the residual contribution. Summing along each member's rollout and averaging within each case gives the cumulative contributions G, J, and C. We denote the corresponding mean squared trajectory error by E^2. This algebraic decomposition tests whether lower trajectory error E^2 accompanies smaller residuals.

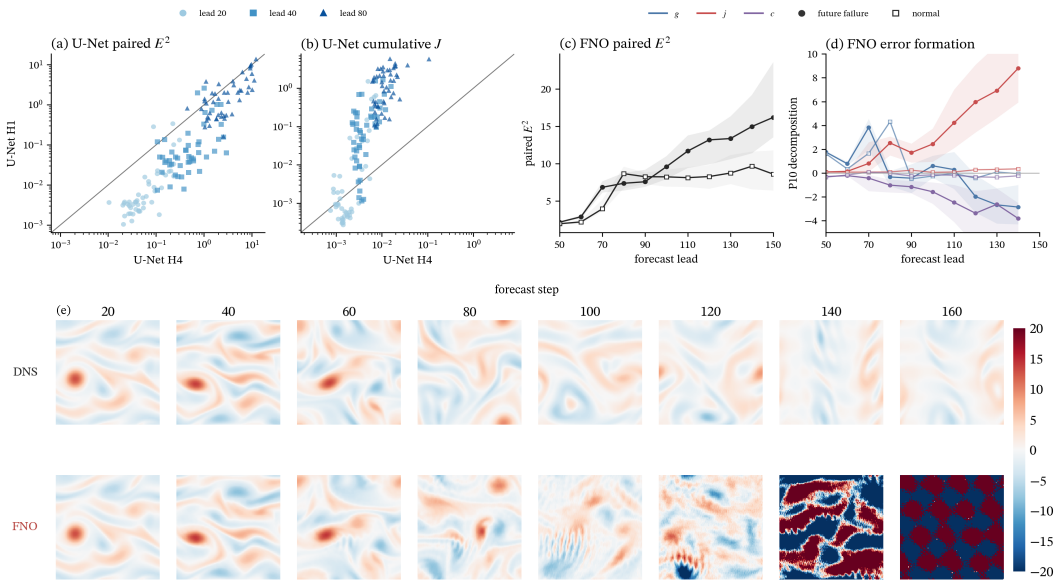

Figure 2: *Local residuals can remain hidden in trajectory error and precede rollout failure.* **(a–b)** Case-averaged trajectory error E^2 and cumulative residual injection J for U-Net H1 (vertical) versus H4 (horizontal). Markers indicate forecast steps 20, 40, and 80; diagonals indicate equality. **(c–d)** Trajectory error and per-interval error contributions g, j, and c over successive 10-step intervals for 50 later-failing and 50 non-crossing FNO trajectories in case 4232; these are not cumulative sums from the forecast origin. Curves show medians; shading shows interquartile ranges for both groups in (c) and the later-failing group in (d). The final unaudited interval is omitted. **(e)** DNS and FNO vorticity fields for one trajectory from the same case.

As shown in Figure 2a–b, U-Net H1 often achieves lower trajectory error than H4 despite larger cumulative residual injection. Its existing discrepancies undergo weaker net amplification under the reference dynamics, offsetting the larger residuals. The lower forecast error therefore does not indicate more accurate local updates.

FNO illustrates the implications for rollout reliability. Within the same case, later-failing and non-crossing groups have similar median trajectory errors (forecast steps 50–100), although the later-failing members already exhibit substantially larger residual injection (Figure 2c–d). As their rollouts progress, residual injection increases, while propagation and coupling become predominantly negative. These negative contributions partly compensate for the residuals but eventually become insufficient to offset them. The fields develop spurious oscillations followed by runaway amplitudes (Figure 2e). Local residuals reveal a difference between the groups before it becomes apparent in trajectory error.

3.2 Improved Forecast Accuracy Can Accompany Distorted Ensemble Evolution

We next examine how a training procedure that lowers trajectory error affects the evolution of nearby states. We use DPOT's autoregressive denoising strategy (Hao et al., 2024) as a controlled comparison. Because the released pretrained checkpoint performs poorly on our flow without adaptation, we fine-tune it on our training cases using the original denoising procedure. The control sets the noise coefficient to zero instead of 0.0005, leaving all other settings and clean targets unchanged.

Figure 3a–b shows that denoising lowers DPOT's trajectory error but also reduces its ensemble spread below the DNS reference. Without denoising, the early ensemble spread is closer to DNS, although trajectory error is larger. The accuracy gain therefore accompanies a suppression of differences between nearby trajectories.

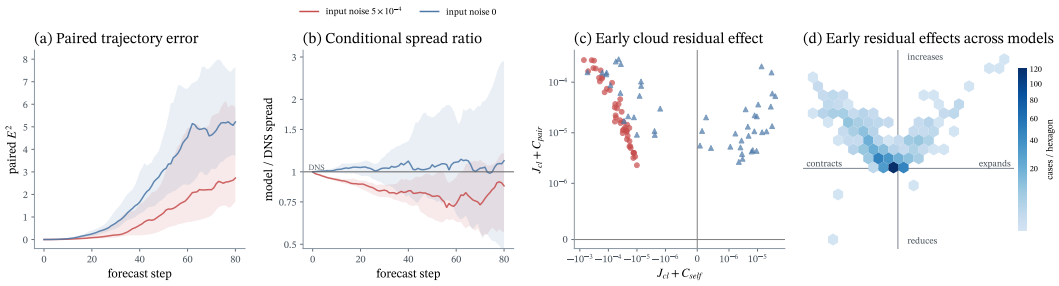

Figure 3: *Improved forecast accuracy can accompany ensemble contraction and larger centered errors.* **(a–b)** Paired trajectory error and solver-to-DNS RMS spread ratio for DPOT fine-tuned with denoising (red) and without denoising (blue). **(c)** For DPOT with denoising, cumulative residual effects on ensemble dispersion (horizontal) and centered paired error (vertical), over forecast steps 0–16. Each point is a case. Negative horizontal values indicate contraction relative to DNS advancement from solver states; positive vertical values indicate increased paired error. **(d)** The same diagnostics across 12 solvers and 48 cases; color indicates counts per hexagonal bin.

To determine whether this suppression corrects ensemble errors, we isolate the residual's effect on each update. Let $\mathcal{Z}$ subtract the ensemble mean at every grid point, and define

$$Y_k = \mathcal{Z} f(\hat{x}_k), \qquad X_k = \mathcal{Z} f(x_k), \qquad R_k = \mathcal{Z} r_k. \tag{3}$$

Here, f acts memberwise: Y_k is the centered ensemble obtained by advancing the current solver states with DNS, and X_k is the centered DNS reference ensemble at the next step. The actual solver update produces $Y_k + R_k$. We measure two cumulative effects over a forecast window $\mathcal{W}$:

$$J_{\mathrm{cl}} + C_{\mathrm{self}} = \sum_{k \in \mathcal{W}} \left(\|Y_k + R_k\|_M^2 - \|Y_k\|_M^2 \right), \tag{4}$$

$$J_{\mathrm{cl}} + C_{\mathrm{pair}} = \sum_{k \in \mathcal{W}} \left(\|Y_k - X_k + R_k\|_M^2 - \|Y_k - X_k\|_M^2 \right). \tag{5}$$

The norm $\|A\|_M^2 = M^{-1} \sum_m \|A_m\|_{\mathrm{sp}}^2$ averages over members and grid points. Both quantities contain the squared residual contribution $J_{\mathrm{cl}} = \sum_{k \in \mathcal{W}} \|R_k\|_M^2$. They differ in its coupling with the reference-advanced solver ensemble, $C_{\mathrm{self}} = 2 \sum_{k \in \mathcal{W}} \langle Y_k, R_k \rangle_M$, or with the centered paired discrepancy, $C_{\mathrm{pair}} = 2 \sum_{k \in \mathcal{W}} \langle Y_k - X_k, R_k \rangle_M$. Negative $J_{\mathrm{cl}} + C_{\mathrm{self}}$ indicates contraction relative to reference advancement from the current solver states; positive $J_{\mathrm{cl}} + C_{\mathrm{pair}}$ indicates increased centered paired error. They sum the residual's effects on successive updates along the observed rollout. Appendix E.2 provides the full derivation.

For DPOT with denoising, every test case satisfies $J_{\mathrm{cl}} + C_{\mathrm{self}} < 0$ and $J_{\mathrm{cl}} + C_{\mathrm{pair}} > 0$ over the early window (Figure 3c). The residuals make the ensemble more compact while moving its centered members farther from their paired DNS counterparts. This combination is also widespread across the tested solvers (Figure 3d). Because contraction is measured relative to DNS advancement from the current solver states, it can occur even when the ensemble's total spread continues to grow.

Main Findings: Lower trajectory error does not guarantee more accurate local updates: weaker dynamical amplification can offset larger residuals, and growing residuals can precede visible rollout failure. In the DPOT control, denoising improves forecast accuracy while suppressing ensemble spread. Across tested solvers, residuals commonly contract ensembles relative to DNS advancement while increasing centered paired error.

4 Do Solvers Preserve the Geometry of Nearby Trajectories?

The preceding analysis shows that solver updates can distort differences between nearby trajectories. We now examine the resulting ensemble geometry: how widely trajectories spread, how many directions account for that spread, and whether those directions agree with the reference dynamics.

For each case and solver, we center the vorticity fields across ensemble members to obtain $z^q_{c,m}$, then apply proper orthogonal decomposition (POD) to resolve their variation into orthonormal spatial modes and associated variances λ_i in decreasing order. With covariance normalized by the spatial dimension, the RMS spread is $a = (\sum_i \lambda_i)^{1/2}$, and r_{90} is the smallest number of modes accounting for 90% of this total variance. Let $U^q_r(s)$ contain the leading r modes at prediction step s, omitting the case index. We compare the solver and DNS subspaces through their normalized squared overlap,

$$O^{\text{solver,DNS}}_r(s) = \frac{1}{r}\left\|U^{\text{solver}}_r(s)^\top U^{\text{DNS}}_r(s)\right\|^2_F, \qquad r = \min\left(r^{\text{DNS}}_{90}, r^{\text{solver}}_{90}\right). \tag{6}$$

Overlap is one for identical subspaces and zero for orthogonal subspaces. Applying the same comparison to a solver's leading three-dimensional subspaces 10 steps apart gives temporal reorientation $R^q_3(s, s+10) = 1 - \frac{1}{3}\|U^q_3(s)^\top U^q_3(s+10)\|^2_F$, with larger values indicating greater reorientation. Appendix G provides the full definitions and normalization.

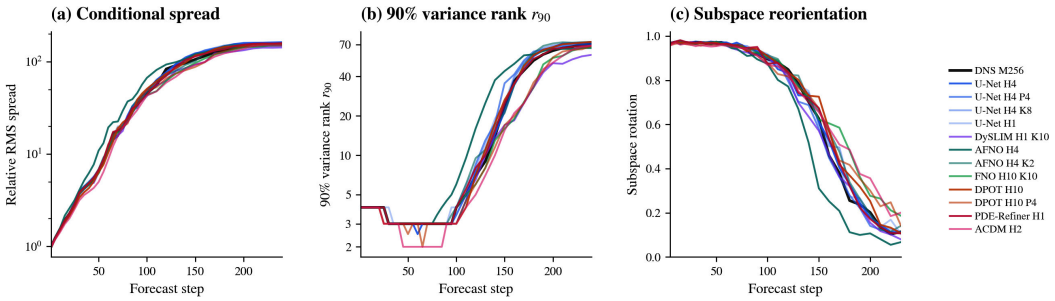

Figure 4: *Ensemble spread, effective dimension, and dominant directions evolve over time.* **(a)** RMS spread, normalized by the DNS case-median spread at the first stored time. **(b)** Number of POD modes required to explain 90% of the variance, r_{90}. **(c)** Reorientation of each solver's leading three-dimensional subspace over 10 steps, measured by $R^q_3(s, s+10)$. Larger values indicate stronger reorientation. Curves show medians across cases.

In DNS, nearby trajectories separate while their variation initially concentrates into a few dominant directions, before spreading across more directions later (Figure 4a–b). The dominant subspace also changes over time (Figure 4c). Most tested solvers reproduce the broad trends in spread and effective dimension and likewise exhibit temporal reorientation.

These similarities do not establish that the solvers preserve the directions selected by DNS. We therefore examine their same-time subspace overlap with DNS (Figure 5).

Most tested solvers exhibit low overlap with the dominant DNS subspace at intermediate prediction times, including configurations whose spread and effective dimension are close to the reference (Figure 5). The misalignment occurs while ensemble variation remains concentrated in a few modes. It therefore affects the dominant spatial patterns of trajectory separation.

Main Findings: Matching ensemble spread and effective dimension does not establish faithful perturbation dynamics. Most tested solvers reproduce their broad temporal evolution but place the dominant variation along spatial directions that differ substantially from those of the reference dynamics.

5 Do Solvers Reproduce Extreme Events Under the Right Conditions?

We next examine whether neural solvers reproduce occurrence and evolution of intense vortices. Matching overall event frequency does not establish that events arise under the correct initial conditions, since *overprediction* and *underprediction* can cancel across cases and times. We first quantify these discrepancies, then examine whether predicted vortices deform and rotate consistently with DNS.

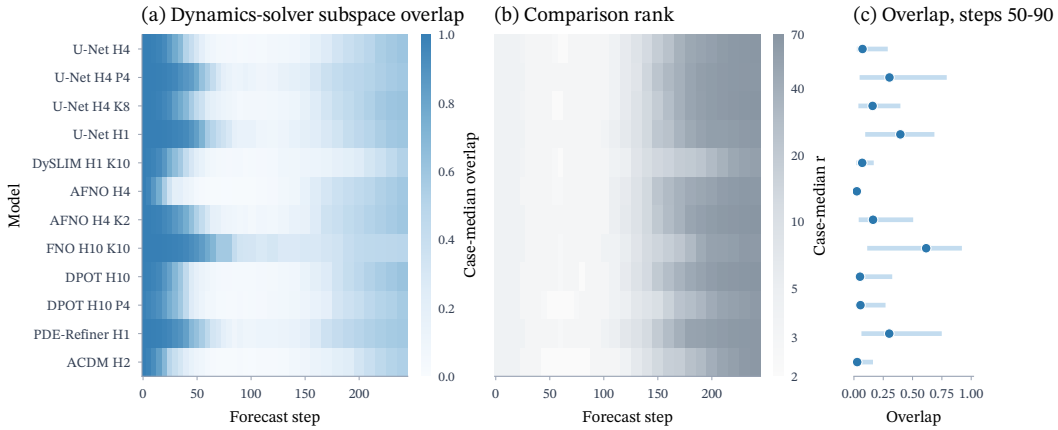

Figure 5: *Similar ensemble spread and dimension can conceal incorrect dominant directions.* **(a)** Median solver–DNS subspace overlap across cases at each prediction step. **(b)** The comparison rank r. **(c)** Overlap summarized by its temporal median over steps 50–90 for each case: dots show medians across cases, and bars show the 10th–90th percentiles.

5.1 Correct Aggregate Frequencies Can Conceal Incorrect Event Conditions

We identify an event when a same-sign connected region with $|\omega| \geq 10$ occupies at least 0.5% of the periodic domain, and estimate its probability for each case and time as the fraction of ensemble members satisfying this criterion. Excess and Deficit separately average overprediction and underprediction relative to DNS, exposing errors that cancel in their signed difference, Bias. We weight cases equally and use trapezoidal time weights; all quantities are normalized by the mean DNS event frequency and reported as percentages (Appendix F.1).

Table 1: Event-frequency errors across test cases, expressed as percentages of the mean reference event frequency. For stochastic solvers, S and I denote noise shared across or sampled independently for ensemble members.

	U-Net				AFNO		DPOT		PDE-Refiner		ACDM	DySLIM
	H1	H4	H4 P4	H4 K8	H4	H4 K2	H10	H10 P4	H1 S	H1 I	H2 S	H1 K10
Bias (%)	+0.42	-60.36	+60.83	-44.33	-6.93	-16.51	-38.43	-25.85	+0.44	+4.74	-22.27	+17.16
Excess (%)	41.31	11.08	81.17	17.06	53.42	35.57	23.39	27.16	36.74	41.38	38.86	54.29
Deficit (%)	40.89	71.44	20.34	61.39	60.35	52.08	61.83	53.01	36.30	36.64	61.13	37.13

As can be observed in Table 1, substantial overprediction and underprediction occur across the tested solvers, including those with nearly correct aggregate frequencies. U-Net H1, for example, has a normalized bias of only 0.42%, yet its excess and deficit are 41.31% and 40.89%, respectively. Similarly, PDE-Refiner with shared noise has a bias of 0.44% but a combined absolute error of 73.04%. Their close aggregate agreement results from cancellation across initial-state ensembles and times, rather than consistently accurate event probabilities.

Table 2: Predicted event frequency (%) within the fixed DNS groups of case 1616, averaged over members and the evaluation window defined in Appendix F.2. DNS frequencies are given in parentheses.

	U-Net				AFNO		DPOT		PDE-Refiner	ACDM	DySLIM
	H1	H4	H4 P4	H4 K8	H4	H4 K2	H10	H10 P4	H1 S	H2 S	H1 K10
Event (100%)	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Non-event (0%)	0.00	0.00	69.58	0.00	0.00	6.99	0.00	0.00	0.00	4.90	0.00

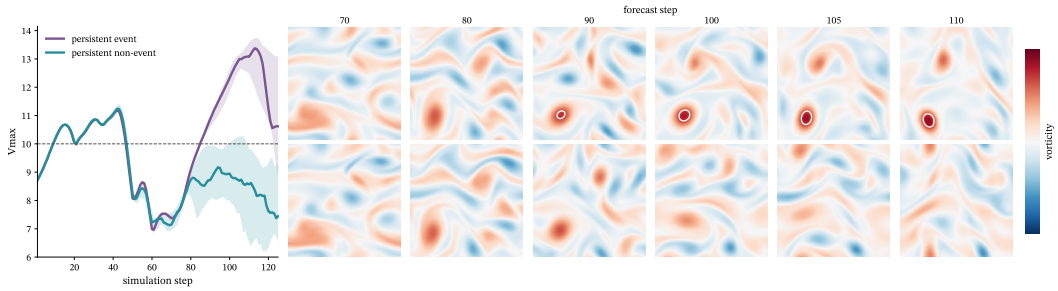

Figure 6: Nearby initial states lead to distinct persistent event outcomes in DNS. For case 1616, peak-vorticity histories and representative fields are shown for groups defined by persistent event and non-event outcomes.

For hazardous events, false negatives can be more consequential than false positives because they leave the hazard unanticipated. We therefore examine whether solvers recover extreme events that persist under the reference dynamics in case 1616 (Figure 6). We fix event and non-event groups using the DNS outcomes and evaluate each solver on the same members (Appendix F.2). Ten of the eleven compared configurations predict no events in the DNS event group (Table 2). The remaining configuration, U-Net H4 P4, recovers all events in that group but also predicts events with a frequency of 69.58% in the DNS non-event group. Thus, the dominant failure in this case is the complete omission of persistent reference events; the only configuration that recovers them also produces frequent false positives.

5.2 Recognizable Vortices Can Deform and Rotate Incorrectly

Event occurrence captures only part of the dynamics. Once a vortex forms, its deformation and rotation reflect its interaction with the surrounding flow. In the classical elliptic-vortex problem, background strain and vorticity govern changes in axis ratio and orientation (Kida, 1981). These geometric properties therefore provide diagnostics beyond peak intensity.

In case 4040, we characterize the evolution of the dominant vortex core using its peak vorticity V, the major-to-minor axis ratio χ_{70} of its 70%-of-peak core, and the accumulated change in its major-axis orientation (Appendix F.3).

Figure 7 shows that the solvers preserve recognizable intense vortices but misrepresent their subsequent geometry. AFNO retains a comparatively round core while the DNS vortex elongates. Conversely, U-Net H4 K8 and DPOT sustain elongation after the reference vortex becomes rounder. Predicted orientations also advance ahead of or lag behind DNS. These discrepancies persist even when peak intensities are similar, showing that a coherent appearance and plausible strength do not establish correct local evolution.

> **Main Findings: Near-correct aggregate event frequencies can hide substantial errors across initial conditions and times.** In the examined branch-separation case, 10/11 configurations miss the persistent event group entirely, while the remaining configuration also predicts frequent events in the non-event group. Even when vortices remain recognizable and have similar peak intensities, their deformation and rotation can differ substantially from DNS.

6 Implications for Evaluating Neural PDE Solvers

Our experiments reveal three gaps between predictive performance and learned dynamics: (1) lower trajectory error can coexist with inaccurate local updates, (2) similar ensemble spread can hide incorrect directions of evolution, and (3) matching event frequencies can conceal missed event-producing conditions. Although these gaps are only demonstrated on turbulent Kolmogorov flow (see Appendix A for limitations and future directions), these findings motivate a broader principle: ***training and evaluation should account for the physical dynamics that matter to the intended application***.

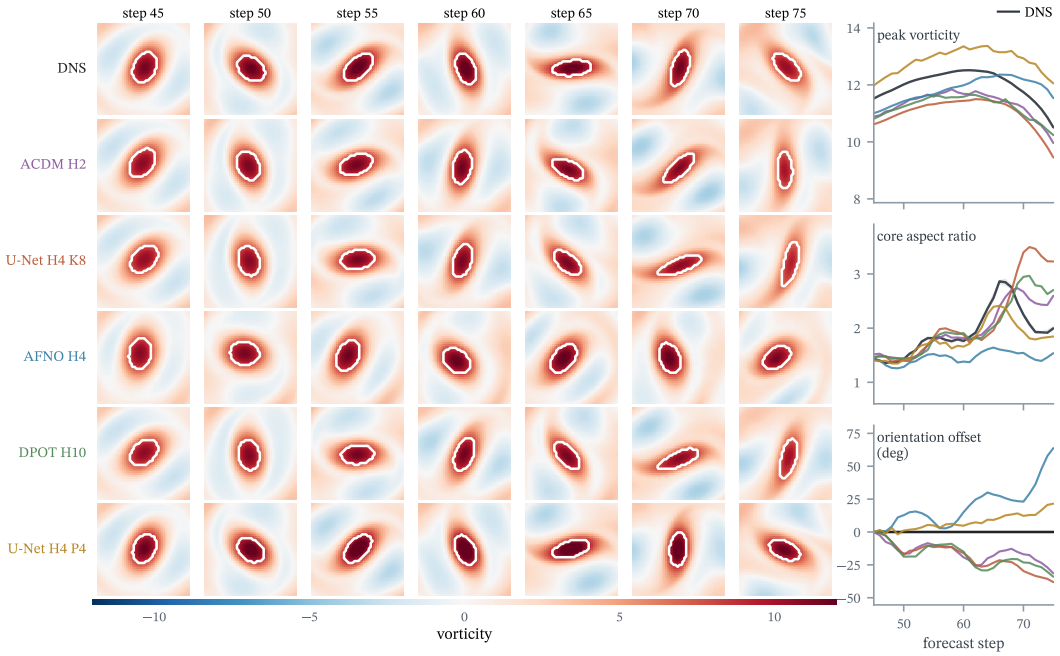

Figure 7: Predicted vortices can retain similar intensity while deforming and rotating differently from DNS. Case 4040. Left: fields evolved from the same initial perturbation, with each solver's 70%-of-peak core boundary shown in white. Right: medians across members for peak vorticity V, core axis ratio χ_{70}, and accumulated major-axis rotation relative to the DNS median.

Assess how training objectives affect dynamics. In a chaotic system, nearby states follow distinct trajectories. Training perturbed and unperturbed states toward the same next-frame target, as in denoising, encourages insensitivity that may remove observation noise but may also suppress physically meaningful differences. Multi-step rollout training presents a related tension: states displaced by earlier prediction errors are trained toward the original reference trajectory, whose continuation may differ from their own. Perturbation scales, forecast intervals, and unroll lengths should therefore be evaluated for their effects on both prediction accuracy and the state evolution during rollout.

Design benchmarks around physical systems and their uses. Datasets and benchmark platforms such as PDEBench (Takamoto et al., 2022), The Well (Ohana et al., 2024), BLASTNet (Chung et al., 2023), and PDEArena (Gupta & Brandstetter, 2022) broaden access to physical simulation data and tools for evaluating neural solvers. To make comparisons scientifically informative, benchmarks should specify whether the dynamics are deterministic or stochastic, whether the observed state is sufficient to determine its subsequent evolution, and which time scales and physical outcomes matter. These considerations should guide the inputs, forecast horizons, and evaluation criteria. For hazard prediction, for example, identifying event-producing conditions may be more important than a modest reduction in average field error. Broad coverage should therefore be accompanied by diagnostics tailored to each system and task.

Make dynamical fidelity part of the evaluation standard. Evaluation should establish whether gains in prediction scores also improve relevant dynamical behavior. Our investigation provides three complementary checks: agreement of local updates with reference evolution, preservation of ensemble geometry, and reproduction of extreme events under the appropriate conditions. Other systems may require different diagnostics, but the aim is the same: identify which physical behaviors improve and which remain distorted. For stochastic solvers, predictive-distribution quality and the dynamical fidelity of generated trajectories require separate assessment. Reporting dynamical diagnostics alongside conventional scores like prediction error would expose otherwise hidden trade-offs and provide stronger evidence of progress in neural PDE modeling.

AI USE STATEMENT

We used generative AI tools for language polishing, assistance with code development, and checking for potentially overlooked related work. The authors reviewed and revised AI-assisted text and inspected and tested AI-assisted code before use. All suggestions were evaluated against the intended methods and claims. The authors take full responsibility for the manuscript, code, experimental results, and conclusions.

REFERENCES

Gary J Chandler and Rich R Kerswell. Invariant recurrent solutions embedded in a turbulent two-dimensional kolmogorov flow. *Journal of Fluid Mechanics*, 722:554–595, 2013.

Wai Tong Chung, Bassem Akoush, Pushan Sharma, Alex Tamkin, Ki Sung Jung, Jacqueline Chen, Jack Guo, Davy Brouzet, Mohsen Talei, Bruno Savard, et al. Turbulence in focus: Benchmarking scaling behavior of 3d volumetric super-resolution with blastnet 2.0 data. *Advances in Neural Information Processing Systems*, 36:77430–77484, 2023.

Zhou Fang and Gianmarco Mengaldo. Dynamical errors in machine learning forecasts. *Chaos, Solitons & Fractals*, 201:117376, 2025.

Tilmann Gneiting and Roopesh Ranjan. Comparing density forecasts using threshold-and quantile-weighted scoring rules. *Journal of Business & Economic Statistics*, 29(3):411–422, 2011.

John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. Adaptive fourier neural operators: Efficient token mixers for transformers. *arXiv preprint arXiv:2111.13587*, 2021.

Jayesh K Gupta and Johannes Brandstetter. Towards multi-spatiotemporal-scale generalized pde modeling. *arXiv preprint arXiv:2209.15616*, 2022.

Zhongkai Hao, Chang Su, Songming Liu, Julius Berner, Chengyang Ying, Hang Su, Anima Anandkumar, Jian Song, and Jun Zhu. Dpot: Auto-regressive denoising operator transformer for large-scale pde pre-training. *arXiv preprint arXiv:2403.03542*, 2024.

Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel De Bezenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for pdes. *Advances in Neural Information Processing Systems*, 37:72525–72624, 2024.

Ruoxi Jiang, Peter Y Lu, Elena Orlova, and Rebecca Willett. Training neural operators to preserve invariant measures of chaotic attractors. *Advances in Neural Information Processing Systems*, 36: 27645–27669, 2023.

Shigeo Kida. Motion of an elliptic vortex in a uniform shear flow. *Journal of the Physical Society of Japan*, 50(10):3517–3520, 1981.

Felix Koehler, Simon Niedermayr, Rüdiger Westermann, and Nils Thuerey. Apebench: A benchmark for autoregressive neural emulators of pdes. *Advances in Neural Information Processing Systems*, 37:120252–120310, 2024.

Georg Kohl, Li-Wei Chen, and Nils Thuerey. Benchmarking autoregressive conditional diffusion models for turbulent flow simulation. *Neural Networks*, pp. 108641, 2026.

Cecil E Leith. Theoretical skill of monte carlo forecasts. *Monthly weather review*, 102(6):409–418, 1974.

Martin Leutbecher and Tim N Palmer. Ensemble forecasting. *Journal of computational physics*, 227 (7):3515–3539, 2008.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

Phillip Lippe, Bas Veeling, Paris Perdikaris, Richard Turner, and Johannes Brandstetter. Pde-refiner: Achieving accurate long rollouts with neural pde solvers. *Advances in Neural Information Processing Systems*, 36:67398–67433, 2023.

Georgios Margazoglou and Luca Magri. Stability analysis of chaotic systems from data. *Nonlinear Dynamics*, 111(9):8799–8819, 2023.

Michael McCabe, Peter Harrington, Shashank Subramanian, and Jed Brown. Towards stability of autoregressive neural operators. *arXiv preprint arXiv:2306.10619*, 2023.

Michael McCabe, Bruno Régaldo-Saint Blancard, Liam Parker, Ruben Ohana, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Siavash Golkar, Geraud Krawezik, Francois Lanusse, et al. Multiple physics pretraining for spatiotemporal surrogate models. *Advances in Neural Information Processing Systems*, 37:119301–119335, 2024.

LD Meshalkin and Ia G Sinai. Investigation of the stability of a stationary solution of a system of equations for the plane movement of an incompressible viscous liquid. *Journal of Applied Mathematics and Mechanics*, 25(6):1700–1705, 1961.

Roberto Molinaro, Samuel Lanthaler, Bogdan Raonić, Tobias Rohner, Victor Armegioiu, Stephan Simonis, Dana Grund, Yannick Ramic, Zhong Yi Wan, Fei Sha, et al. Generative ai for fast and accurate statistical computation of fluids. *arXiv preprint arXiv:2409.18359*, 2024.

Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina J Agocs, Miguel Beneitez, Marsha Berger, Blakesley Burkhart, Stuart B Dalziel, Drummond B Fielding, et al. The well: a large-scale collection of diverse physics simulations for machine learning. *Advances in Neural Information Processing Systems*, 37:44989–45037, 2024.

Jeongjin Park, Nicole T Yang, and Nisha Chandramoorthy. When are dynamical systems learned from time series data statistically accurate? *Advances in Neural Information Processing Systems*, 37:43975–44008, 2024.

Jaideep Pathak, Zhixin Lu, Brian R Hunt, Michelle Girvan, and Edward Ott. Using machine learning to replicate chaotic attractors and calculate lyapunov exponents from data. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(12), 2017.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Salva Rühling Cachay, Bo Zhao, Hailey Joren, and Rose Yu. Dyffusion: A dynamics-informed diffusion model for spatiotemporal forecasting. *Advances in neural information processing systems*, 36:45259–45287, 2023.

Yair Schiff, Zhong Yi Wan, Jeffrey B Parker, Stephan Hoyer, Volodymyr Kuleshov, Fei Sha, and Leonardo Zepeda-Núñez. Dyslim: Dynamics stable learning by invariant measure for chaotic systems. *arXiv preprint arXiv:2402.04467*, 2024.

Bharat Srikishan, Javier E Santos, Nikhil Muralidhar, and Charles D Young. Trie: An evaluation framework for stochastic pde surrogates. *arXiv preprint arXiv:2607.00196*, 2026.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. Pdebench: An extensive benchmark for scientific machine learning. *Advances in neural information processing systems*, 35:1596–1611, 2022.

Zoltan Toth and Eugenia Kalnay. Ensemble forecasting at nmc: The generation of perturbations. *Bulletin of the american meteorological society*, 74(12):2317–2330, 1993.

Alberto Vela-Martín and Marc Avila. Large-scale patterns set the predictability limit of extreme events in kolmogorov flow. *Journal of Fluid Mechanics*, 986:A2, 2024.

APPENDIX CONTENTS

A Discussion and Future Work **13**

B Related Work **13**

B.1 Ensemble Forecasting 13

B.2 Evaluation of Neural PDE Solvers 13

C DNS, Dataset Construction, and Forecast Protocol **14**

D Solver Configurations and Forecast Evaluation **15**

E Derivation of the Error Decomposition **17**

E.1 Trajectory Error 17

E.2 Residual Effects on Ensemble Evolution 17

F Event and Vortex Diagnostics **17**

F.1 Event Definition and Aggregate Probability Errors 17

F.2 Persistent-Event Groups 18

F.3 Vortex Geometry 18

G Ensemble Geometry Diagnostics **18**

A Discussion and Future Work

Our experiments are limited to two-dimensional Kolmogorov flow. This spatially periodic shear flow provides a simple setting for studying hydrodynamic instability (Meshalkin & Sinai, 1961) and the recurrent flow structures underlying turbulent motion (Chandler & Kerswell, 2013). Its tractable formulation makes it useful for examining learned dynamics, but results from one system do not establish how broadly the observed failures extend. Tests on additional dynamical systems may yield a more comprehensive understanding and reveal different relationships between prediction accuracy and dynamical fidelity.

Our diagnostics identify how predicted dynamics depart from DNS, but do not establish which architectural or training choices cause these departures. Prior studies show that both model design and training objectives can affect rollout behavior (McCabe et al., 2023; Lippe et al., 2023). Establishing which factors account for the failures observed here requires further controlled comparisons. More broadly, future work should examine whether improvements in dynamical fidelity persist across systems and operating conditions, alongside gains in conventional predictive performance.

B Related Work

B.1 Ensemble Forecasting

Forecasting with perturbed initial states. Leith (1974) studied finite Monte Carlo ensembles in a test-field model of two-dimensional turbulence to assess forecast skill when the atmosphere is incompletely observed. Ensemble-based estimates improved mean-square wind forecasts, in part by filtering unpredictable small-scale structure; the useful forecast range depended on observational resolution. In operational weather prediction, Toth & Kalnay (1993) used ensembles at NMC to represent growing analysis errors and obtain useful forecasts with few members. Their preliminary experiments found that two bred-perturbation forecasts could outperform substantially larger random or lagged-forecast ensembles. Ensembles also support investigations of dynamical sensitivity: Vela-Martín & Avila (2024) used DNS ensembles with nearby initial states in Kolmogorov flow to study the predictability of extreme dissipation bursts, connecting event predictability to large-scale circulation patterns.

What the perturbed states represent. In operational forecasting, perturbed states primarily represent uncertainty about the actual initial state. This uncertainty arises from measurement errors, incomplete observational coverage, and errors in the background forecast used in data assimilation (Leutbecher & Palmer, 2008). Perturbations therefore represent plausible errors of the estimated state, rather than measurement noise alone. Forecast-model errors can also affect that estimate through assimilation; unresolved processes, uncertain parameterizations, and numerical approximations contribute further uncertainty during forecast evolution. Ensemble-based assimilation seeks to sample the initial uncertainty distribution, while dynamically constrained approaches select forecast-sensitive directions when that distribution cannot be reliably estimated (Leutbecher & Palmer, 2008).

Perturbed states as dense probes of learned dynamics. Many neural PDE solvers are trained and evaluated on numerical simulation data, as in PDEBench and The Well (Takamoto et al., 2022; Ohana et al., 2024). When complete simulated states are supplied without observation noise, incomplete observations and measurement errors are absent from the learning problem. A numerical simulator may differ from the physical system it approximates, but that discrepancy does not itself make the simulator's specified initial state or deterministic evolution uncertain. Errors in learning that evolution are a separate issue. Our motivation is to probe this learned evolution: we deliberately construct dense ensembles of nearby initial states and compare their evolution under each solver and the reference dynamics. *These perturbations provide controlled probes of local dynamical behavior, rather than estimates of uncertainty about an unknown initial state.* Their evolving errors, geometry, and extreme-event behavior reveal how the learned dynamics respond to nearby states.

B.2 Evaluation of Neural PDE Solvers

Large datasets and benchmarks. Evaluation suites examine neural solvers across equations, parameters, and prediction horizons. PDEBench includes field errors, errors in conserved quantities

and boundary values, and errors separated into Fourier bands (Takamoto et al., 2022). PDEArena compares neural architectures and their generalization across PDE parameters and time scales (Gupta & Brandstetter, 2022). The Well reports variance-normalized errors for individual steps and rollout windows, and provides spectral diagnostics to resolve performance across spatial scales (Ohana et al., 2024). BLASTNet complements forecasting benchmarks with three-dimensional turbulence data and a volumetric super-resolution study (Chung et al., 2023). APEBench provides a configurable benchmark across PDE dynamics, architectures, and unrolled training procedures, including experiments connecting rollout accuracy to the learned numerical update (Koehler et al., 2024).

Long-term statistics and dynamical invariants. Learned models have also been evaluated beyond trajectory agreement. Park et al. (2024) provide an ergodic-theoretic account of why small prediction errors can coexist with inaccurate invariant statistics, and study how incorporating Jacobian information during training improves statistical accuracy. Pathak et al. (2017) used reservoir computing to reproduce the attractor statistics and estimate Lyapunov exponents of the Lorenz system and the Kuramoto–Sivashinsky equation. Jiang et al. (2023) studied neural operators trained on noisy Lorenz-96 and Kuramoto–Sivashinsky data using distributional and contrastive objectives. Their evaluation includes summary-statistic distributions and energy spectra, together with Lyapunov and attractor-dimension diagnostics for Lorenz-96; the results show improved long-term behavior relative to training only on prediction error. DySLIM evaluates learned chaotic dynamics using distributional distances, energy spectra, spatial covariance, and temporal correlations (Schiff et al., 2024). Its invariant-measure regularization improves rollout stability and statistical agreement in the tested systems. For stochastic PDEs, Srikishan et al. (2026) introduce TRIE, which evaluates invariant-measure fidelity, predictive uncertainty calibration, and computational efficiency on stochastic Kuramoto–Sivashinsky and Kolmogorov systems. These studies evaluate learned dynamics through long-term statistics, dynamical invariants, and predictive distributions.

Local dynamical properties of forecasts. Margazoglou & Magri (2023) evaluate whether echo state networks recover tangent-space dynamics, including covariant Lyapunov vectors, finite-time Lyapunov exponents, and angles between stable, neutral, and unstable subspaces. Fang & Mengaldo (2025) directly assess dynamical discrepancies in machine-learning forecasts of Lorenz, Kuramoto–Sivashinsky, and Kolmogorov systems. They estimate instantaneous dimension and inverse persistence from recurrences in a training-data reference attractor, relate these quantities to conventional forecast errors, and measure errors in the dynamical indices themselves. States with higher dimension and lower persistence tend to have larger forecast errors, while recursive forecasts can progressively distort these local properties. These approaches characterize tangent-space stability and local attractor properties. Our ensemble diagnostics instead follow finite initial-state perturbations and compare their evolving spread and dominant spatial directions with those of the reference dynamics.

C DNS, Dataset Construction, and Forecast Protocol

DNS Settings. DNS uses a 128×128 Fourier pseudospectral discretization on $[0, 2\pi)^2$, with viscosity $\nu = 0.01$ and an internal time step of 0.002. The spectral update applies the two-thirds de-aliasing rule, removing modes with $|k_x| > 42$ or $|k_y| > 42$. Vorticity fields are saved at intervals of $\Delta t = 0.2$, giving $D = 128^2$ spatial degrees of freedom. In this two-dimensional incompressible periodic flow, the spatial mean velocity is fixed at zero, and the velocity field is recovered from vorticity by solving the streamfunction Poisson equation (Vela-Martín & Avila, 2024). With the forcing and viscosity fixed, the full vorticity field therefore specifies the dynamical state and determines its subsequent reference evolution.

Dataset and Perturbations. The archive provided by Vela-Martín & Avila (2024) contains $8,192$ temporally decorrelated states of statistically stationary Kolmogorov flow. We sample 512 base initial states, each defining a case c. Around each base state, we generate spatially structured perturbations from randomly signed differences between sampled archive states, rescaled to a spatial mean square of 10^{-5}. Each perturbed initial state defines an ensemble member m and is shared across DNS and the neural solvers; member 0 denotes the unperturbed trajectory.

The perturbation generator forms $\delta\omega = s(\omega_a - \omega_b)$, where $s \in \{-1, +1\}$ and the two distinct archive states exclude the base state. It rescales each difference by $\sqrt{10^{-5}/\|\delta\omega\|^2_{\mathrm{sp}}}$ before adding

it to the base state, so the prescribed scale is a spatial mean square, not an RMS amplitude. The generation configuration uses balanced archive-state pairing and signs; ensemble diagnostics use the matched subset of 256 perturbed members rather than all generated members.

We divide the cases into 416 for training, 48 for validation, and 48 for testing. Training and validation use only unperturbed trajectories. Dynamical evaluation uses $M = 256$ perturbed members per test case, with a complete DNS reference trajectory for every member. The unperturbed member is excluded from ensemble statistics and evaluated separately using conventional forecast metrics.

Forecast Origin and Stochastic Sampling. Perturbed initial states first evolve under DNS for ten saved time steps to provide input histories. Each solver receives its required input history from data steps 1–10 and predicts from data step 11 onward, using its own outputs for subsequent updates. Predictions are compared with DNS at matching times. Throughout the paper, prediction step 1 corresponds to data step 11. For stochastic solvers, all members within a case share a random path, with fresh draws at successive updates and different paths across cases, so that sampling differences do not confound responses to initial-state perturbations. We generate one predicted trajectory per initial state; experiments using independent randomness across members are identified separately.

D Solver Configurations and Forecast Evaluation

We screen configurations using conventional relative L_2 forecast error, excluding PFNO H10 P10, DYffusion H1 P16, FNO H1, and DPOT H10 trained from scratch and retaining 12 configurations for dynamical evaluation. Table 3 records the training configurations; Table 4 reports their parameter counts and forecast errors. Scores are case medians of relative L_2 field error on the unperturbed test trajectories, using the common forecast origin. Stochastic solvers use one fixed noise realization per case. Training objectives differ across configurations; controlled interventions hold the remaining training settings fixed.

Architectures and learning procedures. U-Net uses a convolutional encoder–decoder with skip connections between corresponding resolutions (Ronneberger et al., 2015). FNO learns convolution kernels through Fourier-mode weights (Li et al., 2020); AFNO uses structured channel mixing in Fourier space (Guibas et al., 2021). DPOT combines a Fourier-based operator architecture with autoregressive denoising pretraining (Hao et al., 2024). Our denoising control is described in the main text. DySLIM is a training framework that regularizes learned dynamics through distribution matching, rather than a distinct network architecture (Schiff et al., 2024).

U-Net has base width 64, channel multipliers $(1, 2, 4, 4)$, two residual blocks per level, and circular padding. AFNO uses 8×8 patches, embedding width 384, eight layers, eight Fourier channel blocks, an MLP expansion factor of four, and a soft-threshold parameter of 0.01. FNO H10 K10 uses width 32, five layers, 32 retained modes in each spatial direction, and positional grid inputs. DPOT uses the tiny configuration with 8×8 patches, width 512, four layers, four blocks, and 32 modes; the operator blocks, positional embeddings, and temporal aggregation weights are initialized from the pretrained checkpoint.

The U-Net H4 K8 configuration starts from the H4 checkpoint and advances seven updates without gradients before differentiating the eighth-step loss. AFNO H4 K2 starts from its one-step H4 checkpoint and differentiates through both updates. AFNO and FNO use relative L_2 loss; their multi-step configurations sum the equally weighted losses without dividing by the unroll length. These settings distinguish pushforward training from full differentiation through a rollout.

DySLIM uses a periodic convolutional network of width 200, with four levels, four processors, 3×3 kernels, and an input residual connection. Its loss combines squared Sobolev H^1 error with conditional MMD. MMD uses a biased multiscale inverse-multiquadric estimator on a stride-two view of the fields. The rollout curriculum samples lengths up to ten, with gradients stopped through preceding updates. PDE-Refiner uses a conditioned U-Net of base width 32, channel multipliers $(1, 1, 2, 4)$, two residual blocks per level, and circular padding; it predicts state differences.

Stochastic forecasting models. PDE-Refiner uses successive denoising refinements to recover components that can be underrepresented by a conventional prediction loss (Lippe et al., 2023). ACDM generates the next flow state by conditional diffusion given preceding states (Kohl et al.,

2026). DYffusion uses learned temporal interpolation within a diffusion-based forecasting procedure (Rühling Cachay et al., 2023); the evaluated configuration is included below among the excluded settings. Their original evaluations address different targets: PDE-Refiner emphasizes accurate rollout duration, while ACDM examines flow statistics and stability, and DYffusion reports probabilistic forecast scores. Here, their sampled trajectories follow the common initialization and randomness protocol in Appendix C. These descriptions distinguish the published methods from the particular configurations evaluated in this study.

Table 3: Training configurations. $H/P/K$ gives history, prediction block, and training unroll length. Validation losses select checkpoints within each training setup and are not comparable forecast scores across objectives.

Configuration	$H/P/K$	Training objective and procedure
U-Net H1	1/1/1	MSE, one-step training
U-Net H4	4/1/1	MSE, one-step training
U-Net H4 P4	4/4/1	MSE, direct four-step targets
U-Net H4 K8	4/1/8	MSE, eight-step pushforward training
AFNO H4	4/1/1	Relative L_2, one-step training
AFNO H4 K2	4/1/2	Relative L_2, two-step full unroll
FNO H10 K10	10/1/10	Relative L_2, summed over ten unrolled steps
DPOT H10	10/1/1	Fine-tuned from the released pretrained checkpoint
DPOT H10 P4	10/4/1	Fine-tuned from the released pretrained checkpoint
DySLIM H1 K10	1/1/10	Sobolev H^1 plus conditional MMD; pushforward curriculum
PDE-Refiner H1	1/1/1	Diffusion denoising MSE
ACDM H2	2/1/1	DDPM denoising MSE; 20 sampling steps

Table 4: Solver configurations and conventional forecast accuracy. Columns report parameter counts and case-median relative L_2 field error, $\|\hat{x} - x\|_2/\|x\|_2$, on test trajectories.

Configuration	Parameters (M)	Step 20	Step 80
Retained for dynamical analysis			
U-Net H4	11.76	0.123	0.857
U-Net H4 P4	11.76	0.049	0.656
U-Net H4 K8	11.76	0.093	0.730
U-Net H1	11.76	0.037	0.670
DySLIM H1 K10	10.09	0.164	0.863
AFNO H4	10.29	0.334	1.106
AFNO H4 K2	10.29	0.076	0.770
FNO H10 K10	10.49	0.030	0.316
DPOT H10	7.51	0.141	0.912
DPOT H10 P4	7.51	0.155	0.883
PDE-Refiner H1	9.47	0.072	0.627
ACDM H2	11.75	0.289	0.973
Not retained			
PFNO H10 P10	10.51	0.619	6.374
DYffusion H1 P16	11.21	0.962	1.755
FNO H1	10.49	0.756	1.197
DPOT H10 (scratch)	7.51	0.870	1.385

Excluded configurations. PFNO H10 P10 diverged under sampled autoregressive recursion. DYffusion H1 P16, FNO H1, and DPOT H10 trained from scratch had weaker rollout performance than the retained settings. FNO H10 K10 remains in the geometry comparison but is excluded from physical-event comparisons because of its numerical instability.

E Derivation of the Error Decomposition

E.1 Trajectory Error

Using the definitions of e_k, h_k, r_k in Equation 1 with $x_{k+1} = f(x_k)$, we add and subtract DNS advancement from the solver state:

$$e_{k+1} = \hat{x}_{k+1} - x_{k+1} \tag{7}$$

$$= \big[f(\hat{x}_k) - f(x_k)\big] + \big[\hat{x}_{k+1} - f(\hat{x}_k)\big] = h_k + r_k. \tag{8}$$

Expanding the squared spatial norm and subtracting the current squared error gives

$$\|e_{k+1}\|_{\mathrm{sp}}^2 - \|e_k\|_{\mathrm{sp}}^2 = \|h_k + r_k\|_{\mathrm{sp}}^2 - \|e_k\|_{\mathrm{sp}}^2 \tag{9}$$

$$= \underbrace{\|h_k\|_{\mathrm{sp}}^2 - \|e_k\|_{\mathrm{sp}}^2}_{g_k} + \underbrace{\|r_k\|_{\mathrm{sp}}^2}_{j_k} + \underbrace{2\langle h_k, r_k\rangle_{\mathrm{sp}}}_{c_k}. \tag{10}$$

Here $\langle u, v\rangle_{\mathrm{sp}} = D^{-1}\sum_i u_i v_i$, consistent with the norm in the main text. The identity requires neither linearization nor independence of h_k and r_k. In particular, the nonnegative squared residual j_k can be offset by negative coupling c_k.

Summing over a window from a to b yields

$$\|e_b\|_{\mathrm{sp}}^2 - \|e_a\|_{\mathrm{sp}}^2 = \sum_{k=a}^{b-1} g_k + \sum_{k=a}^{b-1} j_k + \sum_{k=a}^{b-1} c_k. \tag{11}$$

Averaging these memberwise quantities gives $E_b^2 - E_a^2 = G + J + C$, with the notation of the main text. When the initial states coincide, $E_a^2 = 0$.

For Figure 2(c–d), a trajectory is classified as later-failing if its peak absolute vorticity exceeds ten times the maximum peak absolute vorticity of the aligned DNS ensemble for case 4232. The comparison uses all 50 threshold-crossing members and a fixed control group of 50 non-crossing members.

E.2 Residual Effects on Ensemble Evolution

Using Y_k, X_k, R_k and the member-averaged inner product from the main text, the centered solver update is $Y_k + R_k$, and its difference from the DNS-advanced reference ensemble is $Y_k - X_k + R_k$. Expanding their squared norms gives

$$\|Y_k + R_k\|_M^2 - \|Y_k\|_M^2 = \|R_k\|_M^2 + 2\langle Y_k, R_k\rangle_M, \tag{12}$$

$$\|Y_k - X_k + R_k\|_M^2 - \|Y_k - X_k\|_M^2 = \|R_k\|_M^2 + 2\langle Y_k - X_k, R_k\rangle_M. \tag{13}$$

Summing over $k \in \mathcal{W}$ recovers $J_{\mathrm{cl}} + C_{\mathrm{self}}$ and $J_{\mathrm{cl}} + C_{\mathrm{pair}}$ in Equations 4–5. Their difference is

$$C_{\mathrm{pair}} - C_{\mathrm{self}} = -2\sum_{k\in\mathcal{W}} \langle X_k, R_k\rangle_M. \tag{14}$$

The same residual can therefore reduce ensemble dispersion while increasing paired error: the two effects use different reference vectors in the coupling term. These sums isolate the residual's effect relative to DNS advancement from the current states. They are not changes between the window's endpoints, which also include the propagation of existing ensemble differences.

F Event and Vortex Diagnostics

F.1 Event Definition and Aggregate Probability Errors

An event occurs when the largest same-sign connected component of $|\omega| \geq 10$, with connectivity across periodic boundaries, occupies at least 0.5% of the domain. Positive and negative threshold masks are labeled separately using four-neighbor connectivity, with labels merged across opposite domain edges. Diagonal contact alone does not join components. Component area is its grid-point

count divided by 128^2, so the event threshold corresponds to at least 82 grid points. The indicator $I(x)$ equals one when this criterion is satisfied and zero otherwise. For each case and stored time, event probability is the fraction of ensemble members satisfying this criterion. In Table 1, Excess and Deficit average the positive and negative parts of the solver-minus-reference probability difference separately; Bias is their signed difference. Cases receive equal weight, and the 35 stored times spanning forecast steps 1–240 receive trapezoidal weights. All three quantities are normalized by the mean reference event frequency using the same weights.

F.2 Persistent-Event Groups

The groups in case 1616 are fixed by persistent event or non-event outcomes under the reference dynamics over forecast steps 90–100. Each solver is evaluated on these same groups. For a fixed group of N trajectories and L stored times, the frequency reported in Table 2 is $(NL)^{-1}\sum_{m,j} I(x_m(t_j))$. This averages event occupancy equally over members and stored times, rather than counting trajectories with at least one event. It is a within-group statistic over the selected window, distinct from the time-weighted aggregate probability errors above.

F.3 Vortex Geometry

For the local comparison in Figure 7, we characterize vortex strength, elongation, and orientation using peak absolute vorticity V, core axis ratio χ_{70}, and accumulated major-axis rotation. The latter describes the geometry of an Eulerian vorticity region, not fluid-particle rotation.

Core identification and periodic coordinates. For each field, $V = \max_{\boldsymbol{x}} |\omega(\boldsymbol{x})|$, and σ is the vorticity sign at that maximum. We select the largest periodic four-neighbor connected component satisfying $\sigma\omega \geq 0.7V$. This is an area-based selection among components of the peak's sign; the selected component need not contain the absolute maximum itself. Before computing geometric moments, we obtain the circular mean of the core coordinates separately in each periodic direction and map each coordinate to its minimum-image offset from that mean. This avoids treating points adjacent across a periodic boundary as widely separated. The construction describes a localized core for which this periodic coordinate chart is well defined.

Axis ratio and orientation. Let $\boldsymbol{\xi}_i$ be the resulting coordinates of the N core grid points and $\bar{\boldsymbol{\xi}}$ their arithmetic mean. We form the area-weighted second-moment matrix

$$S = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{\xi}_i - \bar{\boldsymbol{\xi}})(\boldsymbol{\xi}_i - \bar{\boldsymbol{\xi}})^\top, \qquad \chi_{70} = \sqrt{\frac{\eta_1}{\eta_2}}, \tag{15}$$

where $\eta_1 \geq \eta_2 > 0$ are the eigenvalues of S. Equal grid spacing makes grid-coordinate and physical-coordinate axis ratios identical. The eigenvector associated with η_1 defines the major-axis angle θ modulo π. All core points have equal weight; vorticity magnitudes do not weight the moments. Geometry is undefined for a zero field, fewer than three core points, or a nonpositive minor eigenvalue; non-finite fields are also excluded from this calculation.

Accumulated rotation. For successive valid orientations, we remove the sign ambiguity of the major eigenvector using

$$\Delta\theta_j = \tfrac{1}{2}\,\mathrm{atan2}(\sin[2(\theta_j - \theta_{j-1})], \cos[2(\theta_j - \theta_{j-1})])\,. \tag{16}$$

We accumulate these increments and report angles in degrees. Orientations with $\chi_{70} < 1.05$ are omitted because near-circular cores have poorly determined major axes. An invalid orientation interrupts accumulation: the next valid segment starts at zero, and increments are not inferred across the gap. No such reset occurs over the interval shown in Figure 7. Interpreting the accumulated angle as continuous rotation requires a consistently identified core and inter-frame angular changes smaller than $\pi/2$. Figure 7 reports rotation relative to the DNS median.

G Ensemble Geometry Diagnostics

For each case and solver, centered fields are $z^q_{c,m}(t) = x^q_{c,m}(t) - M^{-1}\sum_{m'} x^q_{c,m'}(t)$. Centering separates variation across members from ensemble-mean bias. For the $D \times M$ matrix

Z of centered fields, we use the spatially normalized covariance $C = ZZ^\top/[D(M-1)]$. Its Euclidean-orthonormal eigenvectors are the POD modes, and its eigenvalues satisfy $\sum_i \lambda_i = (M-1)^{-1}\sum_m \|z_m\|_{\mathrm{sp}}^2 = a^2$. Thus spread is the square root of the total mode variance, rather than the RMS of the eigenvalues. We define RMS spread, the rank needed to capture 90% of the variance, and temporal reorientation as

$$a_c^q = \left[\frac{1}{M-1}\sum_m \|z_{c,m}^q\|_{\mathrm{sp}}^2\right]^{1/2}, \qquad r_{90} = \min\left\{r : \frac{\sum_{i=1}^r \lambda_i}{\sum_i \lambda_i} \geq 0.9\right\}, \tag{17}$$
$$R_3^q(s, s+10) = 1 - \frac{1}{3}\left\|U_3^q(s)^\top U_3^q(s+10)\right\|_F^2.$$

Here, λ_i are the centered covariance eigenvalues in decreasing order, and $U_r^q(s)$ contains the leading r Euclidean-orthonormal POD modes at forecast step s; time and case indices are suppressed where unambiguous. Temporal reorientation R_3 is one minus the normalized squared overlap of subspaces 10 steps apart: zero indicates identical subspaces, and larger values indicate greater reorientation. Figure 4 normalizes RMS spread by the DNS case-median spread at the first stored time and reports case medians of all three metrics.

For solver–DNS comparison, Equation 6 uses the smaller of the two r_{90} values at each case and time, whereas temporal reorientation uses a fixed rank of three. Comparing subspaces rather than individual POD modes avoids interpreting changes of basis within a shared subspace as physical disagreement. Figure 5 reports case-median overlaps and comparison ranks at each step; its summary over steps 50–90 first takes the temporal median within each case, then reports the median and 10th–90th percentiles across cases.