STAYING ON THE ATTRACTOR: SUPERVISING NEURAL SURROGATES OF 3D TURBULENCE WHERE THEY LEAVE IT

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural surrogates are trained to predict 3D turbulent flows in place of direct numerical simulation (DNS). For chaotic flows, the goal is short-term pointwise accuracy followed by long-term physical and statistical fidelity. However, small prediction errors can carry a surrogate away from the flow's *attractor*. Off-attractor states are poorly represented in training data, leaving their evolution weakly constrained. The learned dynamics can then amplify deviations and lead to blow-up, freezing, or statistical drift. In this paper, we propose *off-attractor supervision* (OAS) to supervise neural surrogates where they leave the attractor. OAS teaches the model how the true Navier–Stokes dynamics would evolve from these states. Each selected state is paired with its own future computed by DNS. Three generators select a few hundred states for relabeling. The first collects states from the surrogate's own rollouts. The second uses surrogate attacks to target freezing, excessive amplification, and violations of incompressibility and energy balance. The third perturbs training states along an amplified direction and a strongly damped random direction of the dynamics. All attacks run on the surrogate alone, and DNS relabeling is performed offline once per selected state. Experiments on 128^3 turbulence show that OAS increases the median time to failure from 21 to 721 steps. The compared baselines achieve medians of at most 110 steps, and the advantage holds across training seeds. OAS also achieves the lowest pointwise error at step 15 and the best long-horizon statistics among the compared methods. OAS integrates physical models into neural simulation by extending supervision from fixed reference trajectories to states where the surrogate is likely to fail. This principle can guide the development of more reliable scientific surrogates when deployment takes models beyond the coverage of their training data.

1 INTRODUCTION

Turbulence is a dominant source of simulation cost in combustion, aerodynamics, ocean hydrodynamics, and atmospheric modeling. The three-dimensional incompressible Navier–Stokes equations are nonlinear and chaotic. Unlike their two-dimensional counterparts they stretch vorticity and cascade energy toward small scales, so a simulation must resolve many scales at once. Direct numerical simulation (DNS) resolves every dynamically active scale down to the dissipation scale (Pope et al., 2000). Its cost grows steeply with Reynolds number, since a more turbulent flow demands a finer grid and a shorter time step. Flows of engineering interest lie beyond what a resolved simulation reaches, and neural surrogates are trained to replace it (Li et al., 2021; Wu et al., 2024; Pan et al., 2026; Mukhopadhyay et al., 2026; Armegioiu, 2026; Ye et al., 2025). The hardest task for a surrogate, and the standard way to test one, is autoregressive rollout. The model is applied to its own output from a single initial state without further ground truth, so its errors compound.

Chaos limits what a long prediction can mean. No rollout tracks the truth beyond a horizon the flow itself sets (Section 2.2), so pointwise accuracy is the target only up to that horizon. Beyond it, the prediction should remain a physically admissible turbulent state with the right statistics, the criterion used to evaluate long emulator rollouts in global weather forecasting (Lam et al., 2023; Price et al., 2025; Bonev et al., 2025; Mahesh et al., 2025; Watt-Meyer et al., 2025). This paper addresses both the criterion a long rollout should be evaluated against and the training that has to meet it.

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.

Evaluated that way, every baseline we compared fails, most of them within a turnover time or two (Section 4.2). They are trained on DNS trajectories. Whatever the objective, every state such a trajectory contains lies on the *attractor* of the flow, the set of states the flow itself visits (Section 2.1). Small errors displace a rollout from that set within a turnover time. The model then receives inputs it has never seen, and the energy blows up, the output freezes, or the statistics drift. The true dynamics does none of this. Over one step it does not amplify a deviation of the kind a surrogate makes, whereas the surrogate amplifies along directions that no clean training pair constrains (Section 2.3). A mean-squared-error objective also underweights the smallest scales, which degrades the statistics of a rollout without making it fail (Section 2.4).

Existing stabilization methods act on the training inputs, on the predicted state, or on the learned operator. The first family trains on deviated inputs, by noise injection (Sanchez-Gonzalez et al., 2020; Stachenfeld et al., 2022), the pushforward trick (Brandstetter et al., 2022), unrolled training (Um et al., 2020; Kochkov et al., 2021; List et al., 2025; Chakraborty et al., 2024), or sampling toward the states where a model performs poorly (Ouyang et al., 2026). The second acts on the predicted state, by denoising it (Lippe et al., 2023; Kohl et al., 2026; Dai et al., 2026c; Yoo et al., 2026) or correcting it at inference (Pedersen et al., 2025; Liu et al., 2026), and the third on the learned operator (McCabe et al., 2023; Pervez & Locatello, 2026; Nie & Suzuki, 2026; Schiff et al., 2024). Ours also trains on deviated inputs, and the difference lies in *label assignment*. In a dynamical system the target is the future of the input, so it moves when the input does. Noise injection, the pushforward trick, and unrolled training instead keep the future of the clean state as the label. The alternative, pairing a deviated state with the solver's answer at that state, appears in the diverted-chain objective of Koehler et al. (2024), the adversarial loss of Sun (2026), and the active-learning scheme of Roy et al. (2026). These approaches query the solver during training or state selection, and some require differentiating through it. Their demonstrations reach 32^3 or two dimensions. We take the solver out of the loop. A non-differentiable DNS is run once per state, offline, after the states are chosen, which leaves the choice of states open and brings three-dimensional turbulence within reach. Appendix A places this work against each family.

We therefore propose *off-attractor supervision* (OAS), a data-augmentation method combining failure-mode-guided state selection and offline DNS relabeling for three-dimensional turbulence. A deviated state is paired with its own future, computed by the DNS, so the label is exact wherever the state lies. The surrogate is then supervised where clean data leave it unconstrained, and it learns the response of the dynamics to a deviation instead of producing one of its own. Three generators, which follow the diagnosis above, select the states worth relabeling (Section 3.3). The first collects the states the surrogate reaches in its own rollouts. The second attacks the surrogate by gradient ascent on the network alone, with one objective per failure mode. The third perturbs training states along a direction the dynamics amplifies and along a random one it strongly damps. The solver labels each state once. The pairs fill a fixed fraction of fine-tuning sample slots.

We make three contributions. **(C1) Off-attractor supervision.** We pair states off the attractor with their own future under the DNS rather than with the future of the clean state they came from. The solver runs offline, once per state, and is never differentiated through, which brings the construction to 128^3. **(C2) A diagnosis, and a criterion for failure.** We measure how the dynamics and the learned map each respond to a deviation shaped like a rollout's own error. The amplification we locate in the learned map also tracks time to failure across configurations, and one-step error does not. Failure is scored by a criterion that detects freezing as well as blow-up. An on-policy-only model holds its energy in band for 674 steps but stops moving at step 86, so energy alone would rank it beside the full method. **(C3) Gain and attribution.** On 128^3 turbulence, time to failure rises from 21 steps to 721 on both criteria, at least $6.6\times$ that of the strongest method we compared, and the step-15 error is the lowest. The generators play complementary roles in addressing blow-up and freezing, and new clean DNS data at the labeled-frame count of the pool does not reproduce the gain. The benefit extends to a second regime and in part to a spectral backbone.

2 Diagnosing Long-Rollout Failure in Turbulence

2.1 The attractor, and how rollouts leave it

This study examines forced homogeneous isotropic turbulence. Energy enters at the large scales, the nonlinear term carries it toward smaller ones, and viscosity removes it at the smallest resolved

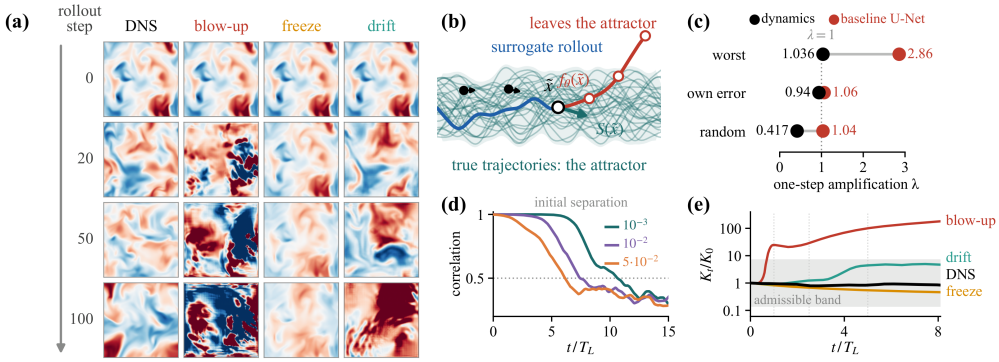

Figure 1: The three failure modes, and the one-step response behind them (Sections 2.1–2.3). (a) Mid-plane u_x of a held-out DNS trajectory and of three rollouts of models trained with clean-future labels, one column each and one row per rollout step. (b) A rollout leaving the attractor. (c) One-step amplification in the dynamics and in the learned map. (d) Divergence of two nearby DNS trajectories. (e) Total energy of the columns of (a). Dotted: the steps of (a).

scales (Pope et al., 2000). Once statistically stationary, the flow visits a particular set of velocity fields over which energy input and dissipation balance on average, which we call the *attractor* of the flow. A simulation dataset draws only from that set, so it constrains a surrogate only on the states it contains (Fig. 1b). At inference the surrogate is applied to its own output, on states no training trajectory visits. Throughout, f_θ is the surrogate's one-step map and S is one step of the reference DNS, deterministic once the forcing realization is recorded (Appendix C). One step advances the flow by $0.05\,T_L$, where T_L is the large-eddy turnover time, and we report times in T_L. States taken from DNS trajectories are *clean*, a *rollout* applies f_θ repeatedly with no further input, and a predicted state is *admissible* if it could be a state of the flow.

A rollout becomes inadmissible in one of three ways (Fig. 1a,e). (1) In a *blow-up*, the energy grows instead of settling, as the baseline U-Net of Fig. 1e does within a turnover time. (2) In a *freeze*, the learned map collapses toward the identity while the energy stays in its physical range, the shaded band of Fig. 1e (Section 4.1). This is equally unphysical, since the energy cascade is continuous in time and a map that reproduces its input transfers no energy between scales. The clean-trained FNO freezes, and its frames stop changing while its energy decays inside the band. (3) In *statistical drift*, the state keeps moving inside the energy band, but its spectrum and enstrophy depart from the reference. The third rollout in the figure, a U-Net fine-tuned on off-attractor states with clean-future labels (Section 2.3), drifts to twenty times the reference enstrophy while its energy stays in band.

2.2 The predictability horizon, and what chaos does not explain

Turbulence is chaotic. Two DNS trajectories that start 0.1% of the velocity norm apart under the same forcing separate with an e-folding time of 1.15 turnover times (Appendix C). That rate and a chosen tolerance set a *predictability horizon* of several turnover times, beyond which pointwise agreement is lost (Boffetta et al., 2002). On a held-out initial condition the correlation of such a pair stays above 0.9 for seven turnover times and falls below 0.5 after ten (Fig. 1d). Starting 5×10^{-2} apart, the size of the baseline's one-step error, the pair needs six turnover times to reach 0.5, long after the blow-up and freeze rollouts of Fig. 1e have failed. A model that minimizes pointwise error beyond the horizon is rewarded for predicting the conditional mean of the futures its input still allows, which collapses their spread onto one field (Dai et al., 2026a). Beyond it a rollout can only be asked to stay admissible with the right statistics. Chaos does not, however, explain the failures of Section 2.1. The clean-trained baseline fails within about one turnover time, roughly one e-folding time. Over that time chaos multiplies a small discrepancy by e, so a 5% discrepancy would grow to about 14%, whereas a rollout that blows up has by then multiplied its energy by e^2. Section 2.3 measures the response of the dynamics to a deviation of the kind a surrogate makes.

2.3 Measuring how the dynamics and the learned map respond to a deviation

A rollout's error is a deviation from the attractor, and its evolution over the next step depends on which map advances it. Under a recorded forcing realization, one DNS step is the map S of Sec-

tion 2.1. It is Markovian, since the next state depends only on the current state and the forcing, and it is defined at a deviated state as at any other (Appendix C). Let x denote a clean state and $\tilde{x}$ a deviated one. The Markov property fixes the training pair at $\tilde{x}$, whose future is $S(\tilde{x})$ (Fig. 1b). Noise injection, the pushforward trick, and unrolled training (Section 1) instead pair it with the clean future $S(x)$ of the state it came from, the *clean-future label*. That label is off by $S(\tilde{x}) - S(x)$, the response of the dynamics to the deviation. We call a method using this label a *clean-label method*. This response is measurable. We write λ for the factor by which one step multiplies a small perturbation of a given state along a given direction, and λ^{model} for the same quantity with f_θ in place of S. Both are measured at the same relative amplitude, 10^{-3} of the velocity norm, so that the two maps are compared on the same perturbation.

An isotropic Gaussian perturbation, whose energy lies at the smallest resolved scales where viscosity acts fastest, gives $\lambda = 0.417$, so one step removes most of it. The worst direction we found, by power iteration through the DNS (Section 3.3), gives 1.036, a single-step gain and not an asymptotic rate (Table 11). A surrogate's own error, whose energy lies mostly at the large scales where the signal is (Fig. 6d), gives 0.94, and a perturbation shaped like the energy spectrum itself gives 1.016. The dynamics therefore damps small-scale deviations and leaves large-scale ones nearly unchanged over one step, which sets how far the clean-future label is from $S(\tilde{x})$. For a small-scale deviation such as injected noise, the clean future is nearly the label the Markov property requires. For a deviation of the kind a rollout produces, the two futures differ by about the deviation itself, so the label moves with the input and cannot be kept. Along the direction a surrogate's error occupies, one step of the dynamics neither removes it nor amplifies it, so whether it grows depends on the learned map.

The same probes applied to f_θ show the learned map amplifying where the dynamics does not (Fig. 1c). The baseline U-Net amplifies its worst direction by $\lambda^{\mathrm{model}} = 2.86$ and carries small-scale content forward at 1.04, where the dynamics removes it at 0.417. Along its own error direction it amplifies at 1.06, against 0.94 for the dynamics. These directions are also not the physical ones. The cosine between a surrogate's direction of steepest growth and the worst direction of the dynamics never exceeds 0.004 in absolute value (Appendix E). Blow-up is therefore not inherited instability but amplification the map produces in directions no clean training pair constrains. Section 4.4 measures it for seven configurations, and Section 3 trains at such states with $S(\tilde{x})$ as the label.

2.4 Where the error spectrum exceeds the signal

The energy of turbulence is concentrated at large scales and falls by orders of magnitude toward the smallest resolved scales, which carry the dissipation. A mean-squared error weights every mode by its absolute error, so modes orders of magnitude below the peak are effectively unsupervised, and networks also exhibit a low-frequency bias (Rahaman et al., 2019). This small-scale deficit has been addressed by spectral and scale-resolved losses and by iterative refinement (Chakraborty et al., 2026; Lippe et al., 2023; Kohl et al., 2026; Khodakarami et al., 2026). A clean-trained U-Net shows it, with a relative error that exceeds the signal itself near the dissipation scale and reaches 304% (Fig. 6d, Appendix E). The deficit shows in the statistics of a rollout rather than in when it fails. Off-attractor supervision does not address it, since the loss family is the baseline's throughout (Section 3.2). A late failure is therefore not sufficient evidence of fidelity, which is why the drift statistics of Section 4.1 are reported with the failure times.

3 Method: Off-Attractor Supervision (OAS)

3.1 The surrogate and the reference map

We learn a deterministic one-step surrogate $u_{t+1} = f_\theta(u_t)$ of forced three-dimensional homogeneous isotropic turbulence and deploy it autoregressively, $u_{t+k} = f_\theta^k(u_t)$, for hundreds of steps from a single initial state without additional inputs. A state carries the three velocity components and a diagnostic pressure on a periodic grid, and the map is state-to-state. The surrogate therefore has neither the forcing realization nor a time index at inference. The reference dynamics is a resolved DNS, a double-precision pseudo-spectral solver with $2/3$-rule dealiasing and Eswaran–Pope stochastic forcing at the largest scales (Eswaran & Pope, 1988). It supplies the clean trajectories as well as the labels (Appendix C). One step of it is the map S of Section 2.1, deterministic once the forcing realization and the time it is replayed from are fixed, and defined off the attractor as well

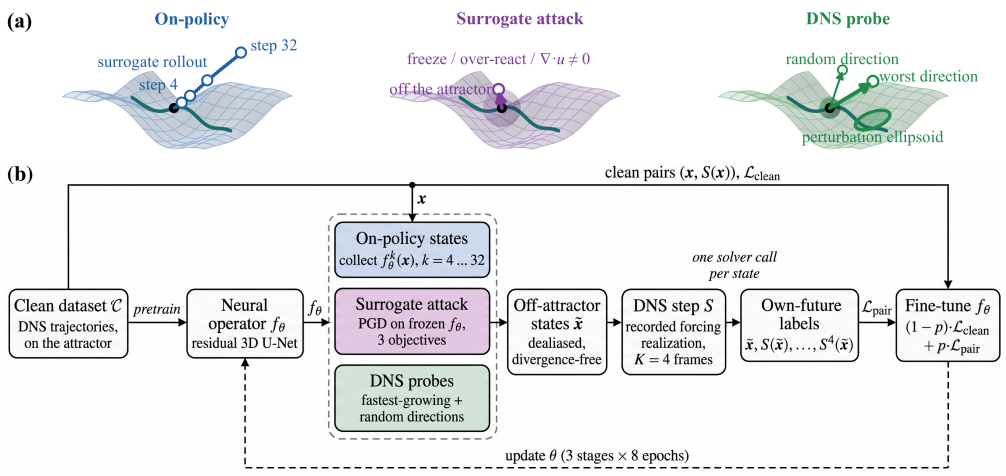

Figure 2: (a) The three state generators, drawn on the attractor. (b) The pipeline, from a generated state to a labeled pair in fine-tuning (Section 3.4).

as on it. We write S^j for j steps under the same forcing realization, so that $S(x), S^2(x), \ldots$ are consecutive frames of one reference trajectory. Clean training data are the pairs a DNS trajectory supplies. The baseline minimizes the one-step error $\mathcal{L}_{\text{clean}}(\theta) = \mathbb{E}_{x\sim\mathcal{C}}\|f_\theta(x) - S(x)\|^2$ over them, where $\mathcal{C}$ is the set of training states, all of which lie on the attractor (Section 2.1). Section 4.1 gives the dataset, backbone, and training budget.

3.2 The own-future label

Off-attractor supervision (OAS) gives a deviated state $\tilde{x}$ the *own-future label*, its own next frame $S(\tilde{x})$ computed by the DNS under a recorded forcing realization (Fig. 2b). No clean future enters the label, and as S is defined at a deviated state as at any other (Section 3.1), the label is exact wherever the deviation lands, once the state is made solver-compatible (Section 3.4). Nothing else changes. The architecture and loss family are the baseline's, and the sample and gradient budgets are those of the comparison methods (Section 4.1). Two controls separate the label from the training schedule and the amount of new DNS data. Fine-tuning on clean windows gives 38 steps, and a matched budget of new clean DNS frames gives 47, versus 721 for own-future pairs (Section 4.3).

3.3 Which states to relabel

Not every state a model might visit is worth relabeling, so three generators select states where the surrogate is likely to fail (Fig. 2a). Two of them generate their states without the solver, and the third uses it only to choose its two perturbation directions. The solver labels the states of all three.

On-policy states. We roll out the surrogate from training states and collect the states it reaches at depths of 4 to 32 steps, a range extending from a fraction of a turnover time to beyond one e-folding time (Section 2.2). These are the on-policy visits of imitation learning (Ross et al., 2011), with the DNS as the expert. They change as the model is retrained. Adding depths 64 and 96 to the full pool, however, brings the freeze back in every seed (Section 4.3), so the pool is limited to depth 32.

Surrogate attacks. The directions along which a model is fragile are not the ones that the dynamics amplifies (Section 2.3), so the states on which it fails can be found by attacking the model alone, with no solver in the attack loop. Projected gradient ascent (Madry et al., 2017) on the frozen surrogate perturbs a training state x by δ to optimize one of three objectives, one per failure mode of Section 2.1. *Freeze-inducing* drives the map toward the identity, minimizing $\|f_\theta(x+\delta) - (x+\delta)\|$. *Amplification-inducing* maximizes the local amplification $\|g_\theta(x+\delta) - g_\theta(x)\|/\|\delta\|$ of the predicted increment $g_\theta = f_\theta - \mathrm{id}$, the network's part of λ^{model} (Section 2.3). *Physics-violating* maximizes the two residuals a predicted state should not carry, the incompressibility residual of the output and the change it makes to the total energy (Appendix D). The search runs thirty gradient steps per state at step size 0.1. Between them the perturbation is projected back onto a divergence-free, dealiased sphere of fixed amplitude, 5% of the norm of the state and the size of the baseline's one-step error

(Section 2.2). The three objectives are complementary and are used together, in equal numbers. An objective that instead maximizes the one-step error, which puts the solver back inside the attack loop, is less effective (Section 4.3). The solver is called once per state, after the search, to compute the label. Conventional adversarial training on a PDE surrogate has no such separation, which is why it has been judged impractical even in two dimensions (Chen et al., 2026).

DNS probes. The two direction families that measure the dynamics in Section 2.3 also serve as generators. The first is the most amplified direction found by power iteration through the DNS. The second is the least amplified of twelve random isotropic divergence-free directions, whose energy lies at the small scales the dynamics removes. Applied to training states at the same amplitude as the attacks, they expose the model to both responses of the dynamics, the amplifying one and the contracting one. They are the pool's only model-independent states. The attacks cannot substitute for them, since a surrogate's steepest direction is nearly orthogonal to that of the dynamics.

3.4 Relabeling and training

We first make every generated state *solver-compatible* by $2/3$-rule dealiasing followed by a projection onto the divergence-free subspace, since the DNS can only start from such a state. The term describes a requirement of the solver, and we reserve *admissible* for the physical sense of Section 2.1. The DNS then advances the projected state $\hat{x}$ for $K = 4$ frames under a shared recorded forcing realization, so that one pair carries its future at four horizons, and the pressure is recomputed from the velocity. Attack and probe pairs take $\hat{x}$ as input. On-policy pairs keep the raw model output as input, so the model also learns to map its own unprojected output onto the solver's trajectory (Appendix C). The horizon $K = 4$ matches the four-step rollout of unrolled training (Section 4.1), so the two differ in their states and labels and not in how far they unroll.

Let $\mathcal{G}$ denote the generated states. A pair stores its input y_0, which is $\tilde{x}$ for an on-policy state and $\hat{x}$ otherwise, and the labels $y_j = S^j(\hat{x})$ for $j = 1, \dots, K$. It is scored by a four-step unrolled loss,

$$\mathcal{L}_{\text{pair}}(\theta; \tilde{x}) = \frac{1}{K} \sum_{j=1}^{K} \frac{\left\| f_\theta^j(y_0) - y_j \right\|^2}{\left\| y_j - y_{j-1} \right\|^2}, \tag{1}$$

which is valid at every j because y_j is the future of the pair's own state at that horizon, and the gradient is taken through the whole chain of j applications. The denominator is the change the dynamics makes at that step, which does not depend on θ and carries no gradient. It fixes the scale of each term, so that a model which returns its input scores one at the first horizon, whatever the amplitude of the deviation. The normalization also reweights the pool, since a pair the dynamics carries far in one step counts for less than one it barely moves. Pairs enter ordinary fine-tuning of the baseline model at a fixed fraction $p = 0.3$ of each epoch's sample slots, the remaining slots keeping the one-step objective, so that training minimizes $(1 - p)\, \mathcal{L}_{\text{clean}}(\theta) + p\, \mathbb{E}_{\tilde{x} \sim \mathcal{G}}\, \mathcal{L}_{\text{pair}}(\theta; \tilde{x})$. Each epoch has the length of a baseline epoch.

Because the on-policy generator follows the model, $\mathcal{G}$ is assembled in stages, and each stage trains for eight epochs. The baseline model is fine-tuned on a first pool of on-policy states with the one-step pair loss ($K = 1$). The resulting model is fine-tuned on an on-policy pool re-collected against it together with the surrogate-attack pairs, and the DNS-probe pairs are added in a third stage. This gives the pool of 242 pairs behind the main model (132 on-policy, 66 surrogate-attack, 44 DNS-probe). A one-shot variant instead builds the whole pool against the baseline model, which lowers stability and seed consistency (Appendix E). Appendix D gives the schedule, and Table 9 the sensitivity to other hyperparameters.

4 Experiments

4.1 Protocol

Setup. The dataset is forced homogeneous isotropic turbulence on a 128^3 grid ($\nu = 0.0055$, $\mathrm{Re}_\lambda \approx 47$, $k_{\max}\eta = 1.65$) from the validated DNS of Section 3.1 (Dai et al., 2026b), split by trajectory into training, validation, and test sets (Appendix C). The surrogate is a residual three-dimensional U-Net (Ronneberger et al., 2015; Çiçek et al., 2016) with 28M parameters, trained on the one-step squared

Table 1: Main comparison. Three held-out initial conditions, medians, over rollouts continued to failure. The baseline U-Net, unrolled training, and OAS are seed medians (Table 16). The other three methods are single runs. The three statistics are read over the first 200 steps ($10\,T_L$), the spread as a single reading at its end. Criteria in Section 4.1.

	nRMSE@15 ↓	band exit ↑	strict ↑	enstrophy $\to 1$	spec. dist. ↓	spread $\to 1$
Baseline U-Net	3.50	21	21	932×	0.95	0.03
Noise injection	1.28	29	29	576×	0.85	0.01
Pushforward trick	1.14	24	24	1771×	0.97	0.02
Unrolled training	0.83	89	89	32.6×	0.50	0.45
Denoising refinement	1.03	110	88	17.7×	0.48	0.12
OAS (ours)	**0.74**	**721**	**721**	**7.4×**	**0.22**	**0.91**

error. We also run the method on the FNO, the backbone that keeps its energy in band the longest (Appendix E). We roll every model out from three held-out initial conditions until it fails, so every failure time we report is observed rather than censored. Rollouts are not bit-reproducible, and we do not interpret differences below the resulting run-to-run floor (Appendix F).

Comparison methods. We re-implement every comparison method on identical data, backbone, and evaluation setup. Each starts from the same 24-epoch one-step baseline and trains for 24 more on its own objective, as ours does, at the same sample and gradient budget. The four are *noise injection* (Sanchez-Gonzalez et al., 2020; Stachenfeld et al., 2022), the *pushforward trick* (Brandstetter et al., 2022), *unrolled training* with the gradient taken through a four-step rollout (Um et al., 2020; List et al., 2025), and *denoising refinement* (FlowRefiner, Dai et al., 2026c; Lipman et al., 2023). Denoising refinement leaves the one-step objective in place and cleans up each predicted state with a flow-matching sampler on the same backbone. Unrolled training appears in its purely unrolled form, since its solver-in-the-loop form needs a differentiable solver at this resolution. The control in its place puts the solver inside the state search instead of the training loop (Section 4.3). Appendix D gives the settings for all four.

Criteria and metrics. Let K_t be the total kinetic energy at step t and K_0 that of the initial state. The *band-exit step* is the first $t \geq 1$ at which K_t leaves $[e^{-2}, e^{2}]\,K_0$, a band far outside the energy fluctuations of the DNS (Appendix B). Energy alone does not detect a freeze, so we also record the *freeze step*, the first step at which the relative change $\|u_{t+1} - u_t\|/\|u_t\|$ stays below 3% for three consecutive steps while the energy is in band. The *strict failure step* is the minimum of the two. The freeze criterion is designed for deterministic maps and is lenient for a stochastic sampler. We report both failure steps throughout, and they disagree exactly when a model freezes before its energy leaves the band. Neither detects drift, which keeps the energy in band and the state moving. We measure drift by three statistics over the first 200 steps, the *enstrophy ratio* and the *log-spectral distance* against the DNS (Pope et al., 2000) and the *spread ratio* against a DNS ensemble (Price et al., 2025; Mahesh et al., 2025). Over the short horizon we report the normalized RMSE (nRMSE) against the DNS at the same step. Appendix B defines all four, fixes the windows they are read over, and reports sensitivity to both thresholds.

4.2 Main result

Time to failure rises from 21 to 721 steps. On both criteria, OAS raises the failure step of the baseline U-Net from 21 to 721. No other method in Table 1 exceeds 110 on either criterion, so the gain over the strongest of them is $6.6\times$ on energy alone and $8.1\times$ on the strict criterion. Over five seeds OAS spans 705 to 784 with no freeze and no overlap with any comparison method (Table 16). Every comparison method ends in blow-up or freezing (Fig. 3). OAS also ranks best on every long-horizon metric of Table 1. The late-window enstrophy ratio falls from $932\times$ to $7.4\times$, and the ensemble spread reaches 0.91 of the DNS ensemble against at most 0.45 elsewhere.

The benefit extends to a second regime and in part to a spectral backbone. A second regime at $\nu = 0.010$ ($\mathrm{Re}_\lambda \approx 35$) has its own DNS dataset (Appendix C) and 132-pair pool. One-shot OAS raises its strict failure step from 77 to 540 ($7.0\times$, band exit $7.6\times$, Table 15). With its own pool, the FNO improves from 2.120 to 1.141 in step-15 nRMSE and from 17 to 35 in strict failure step, but still freezes (Table 14).

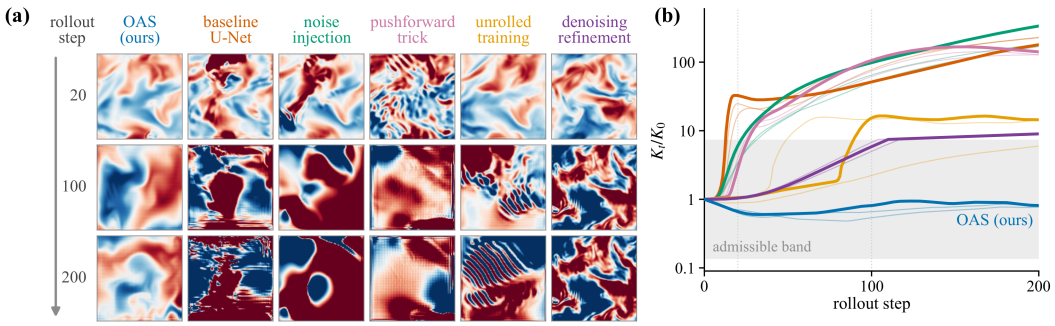

Figure 3: (a) Mid-plane u_x of OAS and of the five other methods of Table 1, one column each and one row per rollout step, from one held-out initial state on the color scale of Fig. 1a. (b) Total kinetic energy K_t/K_0, one color per method as in (a), the rollouts of (a) (thick) and two further held-out initial conditions (thin). Dotted: the steps of (a). Shaded: the band.

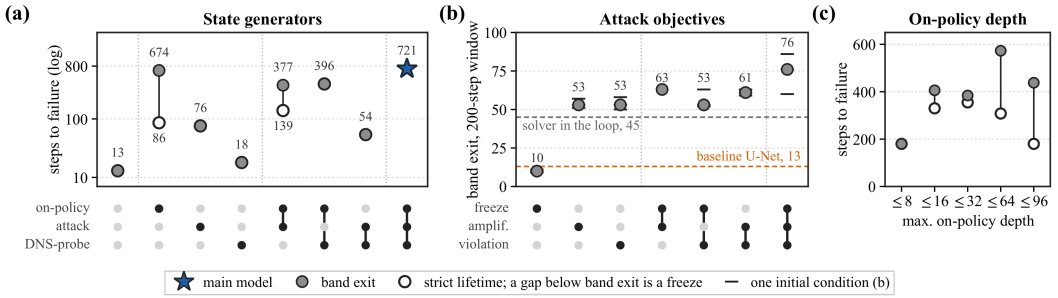

Figure 4: (a) The eight generator combinations, the matrix below marking which generators each pool holds. (b) The three attack objectives and their combinations, on attack pairs alone. (c) Maximum on-policy depth at matched budget.

4.3 Component ablations

The three generators remove different failure modes. The on-policy-only stage connects OAS to imitation learning and earlier solver-based relabeling (Ross et al., 2011; Koehler et al., 2024). It removes blow-up but freezes at step 86 despite holding energy in band for 674 steps (Table 6). The two criteria diverge there, and on the strict criterion the stage and the full method differ by $8.4\times$. Adding surrogate-attack states delays freezing to step 139, with band exit at 377. DNS-probe states remove the freeze (Fig. 4a). Every generator is needed for the full gain: removing on-policy states reduces the strict failure step from 721 to 54, and removing surrogate-attack states reduces it to 396.

The two failure modes trade against each other. No single attack objective suffices. Freeze-inducing pairs alone move the map away from the identity without limiting its increment, but cause earlier failure than the baseline. The other two objectives each reach 53 steps. Combining all three reaches 76 (Fig. 4b, Table 7), so the freeze-inducing and amplification-inducing objectives have to be combined. Maximizing one-step error, with the solver inside the attack loop, reaches 45. At matched budget, the strict failure step peaks at on-policy depth 32. Freezing rises from no initial conditions at depth 8 to all three at depths 64 and 96 (Fig. 4c, Table 8). Freezing also returns with a lower pair fraction, fewer epochs, or a one-shot pool (Tables 9, 5).

Own-future pairs drive the gain. With the same baseline and fine-tuning schedule, clean windows reach 38 steps, versus 524 on band exit and 173 on the strict criterion for one-shot OAS (Table 10). At matched training and DNS budgets (984 clean frames versus 968 OAS labels), clean augmentation reaches 47 steps. This closes only about 1% of the gap between clean-window fine-tuning and staged OAS. An unrolled-trained model fine-tuned on baseline attack pairs reaches only 99 steps, supporting model-specific state selection.

4.4 What predicts failure

One-step error does not predict failure. All methods in Table 1 use the same U-Net backbone, and deterministic rollouts initially track one another (Section 4.1, Figs. 6 and 5a). We report nRMSE at

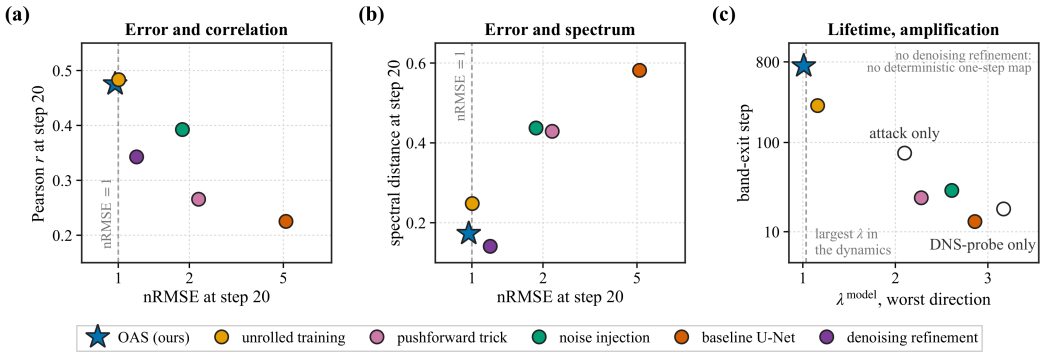

Figure 5: Step-20 nRMSE versus (a) DNS correlation and (b) log-spectral distance for the six methods in Table 1 (dashed: nRMSE of 1). (c) Band-exit step versus worst-direction amplification for seven deterministic configurations (dashed: the largest amplification found in the DNS).

step 15 ($0.75\,T_L$), once rankings separate but before one e-folding time. OAS ranks best, despite the worst one-step validation loss among U-Net models ($1.7\times$ the baseline). Even the two longest-lived methods pass an nRMSE of 1 within about one turnover time (Appendix E).

Learned amplification tracks failure time. Across seven configurations, worst-direction amplification correlates with the band-exit step (Spearman -0.93, Fig. 5c, Table 12). Seven points are too few to establish a law. Every configuration above 2 fails within 76 steps. OAS has the lowest amplification ($\lambda^{\text{model}} = 1.01$), is the only model near the DNS level (1.036), and fails latest on both criteria. Reaching that level takes states chosen against the model, and neither the attack nor the DNS-probe states get there alone. Random-direction amplification remains between 1.01 and 1.04 across configurations, versus 0.417 for the DNS. OAS removes excess amplification but does not reproduce the contraction of the dynamics.

Incompressibility violation does not explain the failure. As a control, we project every output of the baseline model onto the divergence-free subspace. On the baseline run of Table 12 this moves the band-exit step from 13 to 14, so the failure is not an accumulating violation of incompressibility.

5 Limitations and Outlook

Both regimes are 128^3 forced homogeneous isotropic turbulence at a low Reynolds number. The grid fixes it, since a resolved DNS needs $k_{\max}\eta \gtrsim 1.5$ and the dataset sits at 1.65. Finer grids, other forcing, and flows with more structure in their attractor are the natural next settings. The gains are quantified in full on a convolutional backbone. On a spectral backbone, OAS roughly doubles the strict failure step and nearly halves the short-horizon error, but freezing persists (Appendix E). Late-window statistics improve substantially but remain biased (Table 1). Failure time alone does not establish statistical fidelity. The learned map matches the dynamics along its worst direction, but along a random direction it still carries small-scale content forward where the dynamics removes it, and reproducing that contraction is an open problem. The own-future pairs also trade one-step accuracy for rollout accuracy (Yi et al., 2024), and the one-shot variant varies across seeds. Isolating individual generators' causal effects requires matched training histories and budgets.

6 Conclusion

In this paper, we proposed off-attractor supervision (OAS) for improving the long-term stability of neural surrogates for three-dimensional turbulence. OAS addresses insufficient supervision outside the training distribution through three state generators and offline DNS relabeling, pairing each selected state with its own physical future. Experiments show that OAS increases the median time to failure from 21 to 721 steps, reaching $8.1\times$ that of the strongest baseline under a criterion that detects both freezing and blow-up. Controlled experiments support the contribution of the selected states and their labels beyond changes in training schedule or additional DNS data. These findings motivate extending physically consistent supervision beyond reference trajectories as a strategy for stabilizing neural surrogates.

AI USE STATEMENT

Generative AI tools were used for coding and for writing in this work. Large language model coding assistants helped implement parts of the training, state-generation, and evaluation code. Large language models also assisted with translation and were used to edit the text of this paper for academic style and grammar. They were not used for any other research tasks requiring disclosure under the ICLR AI policy, including generating data, developing the conceptual framework or mathematical claims, formulating hypotheses, designing experiments, or interpreting results. AI-assisted code was read and tested by the authors, and every number in the paper was computed by scripts from the per-rollout reports. We take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

The stability criteria were fixed before the main comparison and are defined in Section 4.1 and Appendix B, which also reports the energy band preregistered before the experiments. The solver, the dataset generator, the state generators, and the training and evaluation code are available at https://anonymous.4open.science/r/oas-turbulence-5B16, together with a pipeline that regenerates the corpus, the pair pools, and the trained models from scratch. The noise-injection, pushforward, and unrolled-training baselines are options of the same trainer. The forcing realizations are recorded from fixed seeds, so the relabeling is reproducible. Run-to-run variance is reported in Appendix F and seed variance in Section 4.2.

REFERENCES

Victor Armegioiu. Memory-conditioned flow-matching for stable autoregressive pde rollouts. *arXiv preprint arXiv:2602.06689*, 2026.

Alex Beatson, Jordan Ash, Geoffrey Roeder, Tianju Xue, and Ryan P Adams. Learning composable energy surrogates for pde order reduction. *Advances in neural information processing systems*, 33:338–348, 2020.

Guido Boffetta, Massimo Cencini, Massimo Falcioni, and Angelo Vulpiani. Predictability: a way to characterize complexity. *Physics reports*, 356(6):367–474, 2002.

Boris Bonev, Thorsten Kurth, Ankur Mahesh, Mauro Bisson, Jean Kossaifi, Karthik Kashinath, Anima Anandkumar, William D Collins, Michael S Pritchard, and Alexander Keller. Fourcastnet 3: A geometric approach to probabilistic machine-learning weather forecasting at scale. *arXiv preprint arXiv:2507.12144*, 2025.

Johannes Brandstetter, Daniel E. Worrall, and Max Welling. Message passing neural PDE solvers. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=vSix3HPYKSU.

Yuan-dong Cao, Chi Chiu So, Jun-Min Wang, and He Wang. When and why adversarial training improves pinns: A neural tangent kernel perspective. *arXiv preprint arXiv:2605.15959*, 2026.

Pierre Cesar, Sofya Dymchenko, Abhishek Purandare, and Bruno Raffin. Learning where to simulate: Generative active sampling for online pde surrogate training. *arXiv preprint arXiv:2606.09949*, 2026.

Dibyajyoti Chakraborty, Seung Whan Chung, Ashesh Chattopadhyay, and Romit Maulik. Improved deep learning of chaotic dynamical systems with multistep penalty losses. *arXiv preprint arXiv:2410.05572*, 2024.

Dibyajyoti Chakraborty, Arvind T Mohan, and Romit Maulik. Binned spectral power loss for improved prediction of chaotic systems. *Journal of Computational Physics*, pp. 114866, 2026.

Jian-Nan Chen, Xiao-Yan Cao, and Jun-Jie Zhang. Adversarial vulnerability of machine-learning surrogates for computational fluid dynamics. *Physical Review Research*, 8(3):033002, 2026.

Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pp. 424–432. Springer, 2016.

Yilong Dai, Shengyu Chen, Xiaowei Jia, and Runlong Yu. Flow learners for pdes: Toward a physics-to-physics paradigm for scientific computing. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 13164–13169, 2026a.

Yilong Dai, Yiming Sun, Yiheng Chen, Shengyu Chen, Peyman Givi, Xiaowei Jia, and Runlong Yu. Tide: A physically diverse 3d turbulence benchmark dataset for advancing scientific machine learning. *arXiv preprint arXiv:2608.04222*, 2026b.

Yilong Dai, Yiming Sun, Yiheng Chen, Shengyu Chen, Xiaowei Jia, and Runlong Yu. Flowrefiner: Flow matching-based iterative refinement for 3d turbulent flow simulation. *arXiv preprint arXiv:2604.17149*, 2026c.

Karthik Duraisamy. Predictivity and utility of neural surrogates of multiscale pdes. *arXiv preprint arXiv:2604.20061*, 2026.

Vinayak Eswaran and Stephen B Pope. An examination of forcing in direct numerical simulations of turbulence. *Computers & Fluids*, 16(3):257–278, 1988.

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. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=X7UnDevHOM.

Siavash Khodakarami, Vivek Oommen, Nazanin Ahmadi Daryakenari, Maxim Beekenkamp, and George Em Karniadakis. Spectral bias in physics-informed and operator learning: Analysis and mitigation guidelines. *Computer Methods in Applied Mechanics and Engineering*, 461:119156, 2026.

Dmitrii Kochkov, Jamie A Smith, Ayya Alieva, Qing Wang, Michael P Brenner, and Stephan Hoyer. Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21):e2101784118, 2021.

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.

Jean Kossaifi, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, and Anima Anandkumar. Multi-grid tensorized fourier neural operator for high- resolution PDEs. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=AWiDlO63bH.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Oriol Vinyals, Jacklynn Stott, Alexander Pritzel, Shakir Mohamed, and Peter Battaglia. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023.

Zijie Li, Dule Shu, and Amir Barati Farimani. Scalable transformer for pde surrogate modeling. *Advances in Neural Information Processing Systems*, 36:28010–28039, 2023.

Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=c8P9NQVtmnO.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

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.

Bjoern List, Li-Wei Chen, Kartik Bali, and Nils Thuerey. Differentiability in unrolled training of neural physics simulators on transient dynamics. *Computer Methods in Applied Mechanics and Engineering*, 433:117441, 2025.

Chaoyu Liu, Chris Budd, and Carola-Bibiane Schönlieb. Inverse evolution data augmentation for neural pde solvers. *arXiv preprint arXiv:2501.14604*, 2025.

Qi Liu, Laure Zanna, and Joan Bruna. Towards infinitely long neural simulations: Self-refining neural surrogate models for dynamical systems. *arXiv preprint arXiv:2603.17750*, 2026.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Ankur Mahesh, William D Collins, Boris Bonev, Noah Brenowitz, Yair Cohen, Joshua Elms, Peter Harrington, Karthik Kashinath, Thorsten Kurth, Joshua North, et al. Huge ensembles–part 1: Design of ensemble weather forecasts using spherical fourier neural operators. *Geoscientific Model Development*, 18(17):5575–5603, 2025.

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

Payel Mukhopadhyay, Stefan S Nixon, Romain Watteaux, Michael McCabe, Alberto Bietti, Kyunghyun Cho, Cristiana Diaconu, Irina Espejo Morales, David Fouhey, Siavash Golkar, et al. Emergent transfer of a physics foundation model from simulation to laboratory turbulence. *arXiv preprint arXiv:2606.01470*, 2026.

Fengxiang Nie and Yasuhiro Suzuki. Jacobian-adaptive weighting for stability: Enhancing long-term rollout of neural partial differential equation solvers via spatially-adaptive regularization. *arXiv preprint arXiv:2603.05538*, 2026.

Weihang Ouyang, Min Zhu, Wei Xiong, Si-Wei Liu, and Lu Lu. Rams: Residual-based adversarial-gradient moving sample method for scientific machine learning in solving partial differential equations. *Advanced Intelligent Discovery*, pp. e202500214, 2026.

Guanyu Pan, Huiyu Yang, Yunpeng Wang, Zikun Xu, Jianchun Wang, and Nianyu Yi. Stable fine-time-step long-horizon turbulence prediction with a multi-stepsize mixture-of-experts neural operator. *arXiv preprint arXiv:2604.12794*, 2026.

Christian Pedersen, Laure Zanna, and Joan Bruna. Thermalizer: Stable autoregressive neural emulation of spatiotemporal chaos. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=73zrUyI5kB.

Adeel Pervez and Francesco Locatello. Controlling transient amplification improves long-horizon rollouts. *arXiv preprint arXiv:2605.08856*, 2026.

Stephen B Pope et al. *Turbulent flows*, volume 20. Cambridge university press Cambridge, 2000.

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

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International conference on machine learning*, pp. 5301–5310. PMLR, 2019.

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.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

Samrendra Roy, Souvik Chakraborty, and Syed Bahauddin Alam. Beyond uniform sampling: Synergistic active learning and input denoising for robust neural operators. *arXiv preprint arXiv:2604.13316*, 2026.

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

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. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 43649–43684. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/schiff24b.html.

Kimberly Stachenfeld, Drummond B. Fielding, Dmitrii Kochkov, Miles Cranmer, Tobias Pfaff, Jonathan Godwin, Can Cui, Shirley Ho, Peter Battaglia, and Alvaro Sanchez-Gonzalez. Learned coarse models for efficient turbulence simulation, 2022. URL https://arxiv.org/abs/2112.15275.

Yifei Sun. Solver-integrated adversarial attacking and training of neural operators, 2026. URL https://arxiv.org/abs/2510.18989.

Ashwin Suriyanarayanan, Dibyajyoti Chakraborty, and Romit Maulik. Deep learning of solver-aware turbulence closures from nudged les dynamics. *arXiv preprint arXiv:2604.23874*, 2026.

Kiwon Um, Robert Brand, Yun Raymond Fei, Philipp Holl, and Nils Thuerey. Solver-in-the-loop: Learning from differentiable physics to interact with iterative pde-solvers. *Advances in neural information processing systems*, 33:6111–6122, 2020.

Oliver Watt-Meyer, Brian Henn, Jeremy McGibbon, Spencer K Clark, Anna Kwa, W Andre Perkins, Elynn Wu, Lucas Harris, and Christopher S Bretherton. Ace2: accurately learning subseasonal to decadal atmospheric variability and forced responses. *npj Climate and Atmospheric Science*, 8 (1):205, 2025.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for PDEs on general geometries. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53681–53705. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/wu24r.html.

Zaijun Ye, Chen-Song Zhang, and Wansheng Wang. Recurrent neural operators: stable long-term pde prediction. *arXiv preprint arXiv:2505.20721*, 2025.

Bongsoo Yi, Rongjie Lai, and Yao Li. Improving clean accuracy via a tangent-space perspective on adversarial training. *arXiv preprint arXiv:2408.14728*, 2024.

Seungwoo Yoo, Juil Koo, Daehyeon Choi, and Minhyuk Sung. Diffusionrollout: Uncertainty-aware rollout planning in long-horizon pde solving. *arXiv preprint arXiv:2602.13616*, 2026.

A EXTENDED DISCUSSION OF RELATED WORK

Section 1 groups the stabilization literature by where a method acts, on the training inputs, on the predicted state, or on the learned operator. Three further distinctions organize the comparison drawn in this paper: which states the model is trained on, which future each of those states is labeled with, and where the reference solver is called, if it is called at all. Table 2 arranges the families by those three.

Deviated inputs with clean-future labels. Noise injection perturbs the input with a random field (Sanchez-Gonzalez et al., 2020; Stachenfeld et al., 2022), the pushforward trick perturbs it with the model's own one-step output and detaches the gradient through the perturbation (Brandstetter et al., 2022), and unrolled training exposes the model to the states its own rollout reaches over a short window (Um et al., 2020; Kochkov et al., 2021; List et al., 2025; Chakraborty et al., 2024). The three differ in how the deviation is produced but agree on what it is paired with. Noise injection adjusts its target so that the absolute one is unchanged: Sanchez-Gonzalez et al. (2020) subtract the injected velocity noise from the acceleration target, so that integrating the prediction recovers the unperturbed next state, and Stachenfeld et al. (2022) subtract it from the target increment to the same effect. The pushforward loss is $\mathcal{L}(f_\theta(u^k + \epsilon), u^{k+1})$, with u^{k+1} read from the stored reference trajectory. No solver is re-run at the perturbed state in any of the three, and Appendix D gives the implementations used here. The error that choice leaves is $S(\tilde{x}) - S(x)$, and Section 2.3 measures it at both amplitudes that matter. For a small-scale perturbation, the dynamics removes most of the difference within one step, so the clean future is close to the correct label and the choice costs little. For a deviation of the size and spectrum a rollout actually produces, the two futures differ by about the deviation itself, so the label is wrong by roughly the quantity the model is being trained to handle.

Selecting which states to train on. A second line chooses training points by a criterion rather than uniformly. Ouyang et al. (2026) treat the sample locations as parameters and move them by ascent on the PDE residual, which needs no labels in the physics-informed setting and calls a solver at the moved inputs in the data-driven one. Cesar et al. (2026) train a generative sampler, though over configuration parameters rather than over states, and simulate fresh trajectories from the parameters it proposes, so the pairs it produces lie on the attractor. Cao et al. (2026) analyze a discriminator-based form of adversarial training for physics-informed networks, which is residual-based and needs no labels, so the difficulty discussed here does not arise there. The surrogate-attack and DNS-probe generators of Section 3.3 belong to this line and share its premise, that a labeling budget is better spent where a model is weak than spread over the attractor.

Solver-computed labels. Three methods pair a deviated state with the solver's answer at that state, which is the labeling rule of Section 3.2. Koehler et al. (2024) define a *diverted chain* objective in which the reference simulator branches off after each network prediction, so the target at a state the model rolled out to is the simulator advanced from that same state. The setup requires that simulator to be differentiable, and the benchmark reports it up to 32^3. Sun (2026) states the distinction explicitly at adversarially found states, contrasting a loss against the solver's answer at the perturbed input with one against the clean answer and ablating the two, with the solver inside the attack loop and differentiated through, at about one solver forward and backward per attack step, on one- and two-dimensional problems. Roy et al. (2026) select perturbed inputs by a gradient-free attack and label them with the solver at those inputs, on one-dimensional Burgers. Three further methods are adjacent without sharing the rule. Solver-in-the-loop places a differentiable solver inside the training loop so that its response to the corrected state enters the gradient, and keeps the reference trajectory as the target, so its deviated states carry clean-future labels (Um et al., 2020). Liu et al. (2025) build pairs through the evolution operator, consistent by construction and independent of any model. Suriyanarayanan et al. (2026) record the synchronization forcing of a nudged coarse run and learn it offline as a closure term rather than as a surrogate.

Off-attractor supervision differs from the first three in where the solver runs. It calls a non-differentiable DNS once per selected state, offline, after the states have been chosen, so that neither the attack nor the rollout collection touches it. The choice of states is then unconstrained by the cost of labeling, which admits a generator needing only gradients of the network (Section 3.3), and the procedure reaches three-dimensional turbulence at 128^3, where the three works above stop at 32^3 or at two dimensions. Grid size is not itself the obstacle, since Suriyanarayanan et al. (2026) train

offline against 128^3 DNS for a closure term. The number of solver calls is. Fixing the pool between stages rather than refreshing it at every step is the cost of the arrangement. Section 3.4 offsets part of it by re-collecting the on-policy states against the model the first stage produces, and Table 5 reports what a pool built in one pass gives up.

Adversarial training of physics surrogates. Chen et al. (2026) audit FNO and U-Net surrogates on three two-dimensional flows and find them adversarially fragile. They also note why the usual remedy is hard to apply. A perturbed input changes the PDE solution, so a valid label needs a new simulation, which makes conventional adversarial training impractical for these surrogates. That is the mismatch of Section 2.3, seen from the cost side, and it binds harder in three dimensions, where the resimulation is more expensive. The attack used here needs no solver. The three objectives of Section 3.3 are functions of the network alone, so the search costs forward and backward passes through it, and the solver is called once per state at the end, to supply the label.

On-policy states and imitation learning. Collecting the states a policy visits and labeling them with an expert is DAGGER (Ross et al., 2011), with the DNS as the expert. Beatson et al. (2020) apply it in a PDE setting, querying a finite-element solver at the states a learned energy surrogate reaches. Their surrogate is static, so no question of which future to use arises. The first stage of Section 3.4 is that algorithm, which is why Section 4.3 reports it on its own: it holds the energy in band for 674 steps and stops moving at step 86, removing the blow-up mode and leaving the freeze. The limitation is structural rather than a matter of tuning. A rollout visits the states the model reaches, not the ones it would fail on and has not reached, and deeper rollouts do not fix this, since extending the pool to depths 64 and 96 makes the freeze more frequent rather than less (Table 8). The generator ablation of Section 4.3 attributes the removal of the freeze to the other two generators.

Post-hoc correction and operator regularization. Two families act elsewhere and are in principle composable with this one. Post-hoc correction leaves the one-step objective in place and cleans up the predicted state, by iterative refinement or a diffusion or flow-matching sampler (Lippe et al., 2023; Kohl et al., 2026; Dai et al., 2026c; Yoo et al., 2026), or moves it back toward the attractor at inference (Pedersen et al., 2025; Liu et al., 2026). The last of these is the complement of what is done here: it returns a rollout to the set that training covers, rather than extending training to cover where rollouts go. Operator regularization acts on the learned map rather than on the data it is fitted on, by bounding the spectral norm of the convolution in the architecture itself (McCabe et al., 2023), by a spatially adaptive penalty (Nie & Suzuki, 2026), by penalizing the non-normality and non-commutativity of successive Jacobians (Pervez & Locatello, 2026), or by matching an invariant measure (Schiff et al., 2024). The demonstrations are on two-dimensional flows, global weather fields, and chaotic systems of a few thousand state dimensions. This family addresses the failure mode of Section 4.4 by a different route. The penalties of Pervez & Locatello (2026) constrain the *structure* of the Jacobians, since normal and commuting Jacobians bound the gain of a product by the spectral radius rather than by the larger non-normal bound, whereas λ^{model} is a finite-amplitude magnitude. Neither quantity bounds the other. Here it falls from 2.86 to 1.01 with no term in the objective referring to it or to any Jacobian. Whether supplying the labels and constraining the map are redundant or complementary is not answered by any experiment reported here.

Evaluating a long rollout. Reporting pointwise skill over a short horizon together with spectra and ensemble spread beyond it is established practice in operational weather and climate emulation (Lam et al., 2023; Price et al., 2025; Bonev et al., 2025; Mahesh et al., 2025; Watt-Meyer et al., 2025), and Section 4.1 follows it, with two additions specific to this setting. The failure time detects freezing as well as blow-up, since a map collapsing toward the identity keeps its energy in band indefinitely and is ranked well by an energy criterion alone, as the clean-trained FNO is in Table 13. Drift statistics are read beside it, since a late band exit leaves the statistics unconstrained, as the first training stage shows (Appendix E). Duraisamy (2026) argues that part of what a surrogate of a multiscale system loses is not recoverable, since spectral bias and coarse-graining discard high-frequency content that no architecture or training procedure puts back. The deficit of Section 2.4 is an instance of the spectral half of that argument. The failures studied here are of a different kind, occurring within a turnover time or two at the resolution of the reference itself, and Section 2.3 locates their cause in the learned map.

Table 2: Where off-attractor supervision sits. *States*: which inputs the model is trained on. *Label*: which future each is paired with, $S(x)$ being the future of the clean state it was displaced from and $S(\tilde{x})$ its own. *Solver*: where the reference solver runs, if at all. Rows are families rather than individual papers, and the text qualifies them. The own-future label appears in four of them, and the second and fourth columns are where those four differ.

family	states	label	solver
One-step supervised	attractor only	clean future	offline, dataset only
Noise injection, pushforward	randomly perturbed	clean future $S(x)$	none
Unrolled training	on-policy, short window	clean future $S(x)$	none
Solver-in-the-loop	on-policy, solver-corrected	clean future $S(x)$	differentiable, in the training loop
Diverted chain	on-policy	own future $S(\tilde{x})$	differentiable, in the training loop
Solver-integrated attacks	adversarial	own future $S(\tilde{x})$	in the attack loop, differentiated through
Active learning with solver labels	attack-selected	own future $S(\tilde{x})$	online, once per selected input
Inverse-evolution augmentation	analytically constructed	consistent by construction	evolution operator, analytic
Post-hoc correction	attractor only	clean future	none, or at inference
Operator regularization	attractor only	clean future	none
OAS (ours)	**on-policy + attack + DNS probe**	own future $S(\tilde{x})$	**offline, once per selected state, not differentiated through**

B EVALUATION CRITERIA, PROVENANCE, AND SENSITIVITY

Definitions. Steps are counted from the first predicted frame (step 1). A rollout is continued until it fails, with no fixed evaluation horizon, so no failure time reported here is censored. In the freeze criterion of Section 4.1, the in-band condition prevents a blow-up from being recorded as a freeze through its inflated denominator. A stochastic sampler draws fresh noise at every step, so its relative change stays above the freeze floor even when its mean prediction has stopped moving, and the strict failure step of such a model is an upper bound. The failure-mode label in the tables records how a rollout ends. Within the first 200 steps the priorities are blow-up (energy exits upward or a non-finite value appears), then freeze. A rollout still in band and moving after 200 steps takes the label of its eventual terminal event, marked *late* for the longest-running configurations here. Drift is not a terminal event. A rollout drifts when it stays in band and moving while the late-window energy or enstrophy ratio to the reference lies outside $[0.5, 2]$ or the time-averaged log-spectral distance exceeds 0.5, and the three statistics below report it. The enstrophy ratio divides the rollout's mean enstrophy over the late window, the last third of the 200 steps, by the stationary DNS value, the log-spectral distance is the relative L^2 distance between the logarithms of the shell-averaged energy spectra over the resolved shells, taken against the time-averaged reference spectrum, and the spread ratio divides the eight-member ensemble spread at a given step by the spread of the DNS ensemble started from the same eight perturbed states. None of the three depends on how far the DNS trajectory of an evaluation initial condition runs: they are a stationary value, a time average, and a separately integrated ensemble (Appendix C). The pointwise error nRMSE at step t is $\|u_t - u_t^{\mathrm{DNS}}\|_2 / \|u_t^{\mathrm{DNS}}\|_2$, taken over the normalized state, the three velocity components and pressure. It reaches 1 when the prediction is no better than the zero field and $\sqrt{2}$ for two decorrelated fields of matching variance. It is defined only while the reference exists, about 150 frames after an evaluation initial condition.

Provenance of the band. Energy is quadratic in amplitude, so $K_t/K_0 = e^{\pm 2}$ corresponds to a root-mean-square velocity one e-fold away from its initial value. On six long DNS records of about 125 turnover times each (Appendix C), the energy fluctuates with a standard deviation of 12% of its mean (per trajectory 10.5 to 19.9%) inside an envelope of $[0.70, 1.36]$ times the mean. The band

edges lie about 7 and 53 standard deviations from the mean, and DNS trajectories never trigger the criterion.

Sensitivity of the two thresholds. The freeze threshold beside it is fixed from the same records: the DNS changes by 10 to 25% of the velocity norm per frame, and the 3% threshold lies a factor of about three below the slowest of those. Sweeping it needs no retraining, since the strict failure step can be recomputed from the stored per-step change rates. Over $\varepsilon \in \{1, 2, 3, 5\}\%$ and persistence $k \in \{3, 5, 10\}$ steps, the main model's strict failure step equals its band-exit step at every cell with $\varepsilon \leq 3\%$, at any k: no seed freezes anywhere in that range, which is the criterion-independent form of the statement in Section 4.3. Its ratio to the first training stage stays between $4.3\times$ and $8.4\times$ across the grid and takes the value quoted in Section 4.3 at the setting fixed in advance. Only at $\varepsilon = 5\%$, which counts a state still moving by a twentieth of its norm each step as frozen, does the main model register a freeze, at step 516, and the ratio is then $8.3\times$. The variant carrying depths 64 and 96 is what the grid separates: its strict failure step falls by two fifths over the same range, from the band-exit step at $\varepsilon = 1\%$ to 479 at $\varepsilon = 5\%$. Rescaling the band to $[1/b, b]\, K_0$ for $b \in \{1.5, 2, 3, 5, 10\}$ and recomputing every configuration's band-exit step leaves the ranking unchanged for $b \geq 2$. The value $b = 1.5$ is excluded because it lies close to the observed envelope and produces false positives on rollouts that remain admissible. For a single seed, the ratio of the main model's band-exit step to that of unrolled training ranges from 2.9 to 8.5 over $b \in [2, 10]$, and the value at $b = e^2 \approx 7.4$ lies near the lower end of that range. The band preregistered before the experiments was $[0.2, 5]\, K_0$, under which the same two methods give 770 and 239 steps ($3.2\times$). The switch to the e^2 band was made for its provenance and is conservative. Energy is also the last of the checks to trigger. The baseline model violates gradient-statistics and incompressibility checks within its first four steps, so the band-exit step is a late estimate of the failure time, not an artifact of a sensitive threshold.

C DNS, FORCING, AND DATASET

Solver. The reference dynamics is an incompressible pseudo-spectral Navier–Stokes solver in double precision on a triply periodic 128^3 grid with $2/3$-rule dealiasing, low-storage third-order Runge–Kutta time stepping with the viscous term integrated exactly by an integrating factor, and Eswaran–Pope stochastic (Ornstein–Uhlenbeck) forcing (Eswaran & Pope, 1988) confined to the lowest wavenumber shell ($k < 2$, correlation time 2.0, variance 0.0555). The corpus trajectories use a CFL-adaptive time step (CFL number 0.4, $\Delta t \leq 0.008$) and export a frame at the first step past each multiple of $0.05\, T_L$ ($T_L = 2.47$). Relabeling uses the same scheme at a fixed $\Delta t = 3.98 \times 10^{-3}$, so one model frame is 31 solver steps. Every label is driven by one recorded forcing realization per time-step size, started from a zero forcing state and replayed from its first step, so all pairs share the same forcing. The forcing state therefore reaches about 34% of its stationary amplitude after one frame and 62% after four, whereas the clean trajectories are sampled after a spin-up under stationary forcing. Replaying the recorded realization from the same state reproduces the trajectory with zero difference. Any state injected into the solver (on-policy, surrogate-attack, or DNS-probe) is first dealiased and projected onto the divergence-free subspace, and the pressure is recomputed from the velocity by the pressure Poisson equation when frames are exported. Surrogate-attack and DNS-probe pairs store that projected state as their input, so the model is trained at exactly the state its label was computed from. On-policy pairs store the raw rollout state, velocity and pressure as the model produced them, so they also supervise the projection. Collected states with large velocities are relabeled with a reduced time step ($\Delta t/m$, $m \leq 8$) to keep the CFL number below 0.45, and states beyond that limit are discarded.

Validation. The production dataset passes a validation protocol: $k_{\max}\eta = 1.65$ (mean over its 2553 frames, with 99.9% of the dissipation resolved), monotone spectral tail, box-to-integral-scale ratio 5.3, stationarity drift 0.024% per turnover time, energy-budget closure 0.07%, component and gradient isotropy within 1% and 5%, incompressibility residual $\langle(\nabla\cdot u)^2\rangle/\langle|\nabla u|^2\rangle = 10^{-29}$, velocity-derivative skewness -0.51, and a momentum-residual check of the Navier–Stokes equations on stored triples of frames with second-order convergence under time-step halving. Six long 256^3 records of about $125\, T_L$ supply independent validation at higher $k_{\max}\eta$ and the energy-band statistics of Appendix B. They enter neither training nor evaluation. The second regime ($\nu = 0.010$) passes the same checks ($\mathrm{Re}_\lambda = 35.0 \pm 1.3$, $k_{\max}\eta = 2.59 \pm 0.06$).

Dataset. Sixteen DNS trajectories with independent forcing realizations make up the corpus. Each is spun up for 30 time units (about 12 turnover times) and then sampled at $0.05\,T_L$ for about eight turnover times, some 160 frames. The split is trajectory-atomic, eleven trajectories for training, two for validation, and three for evaluation, and normalization statistics are computed on the eleven training trajectories only. This gives about 1800 training frames, or 450 stride-4 rollout windows. The three evaluation initial conditions are frames of the three test trajectories, which span the same window as the training ones. The dispersion reference and the eight-member DNS ensemble behind the spread ratio at step 200 are produced separately, by re-integrating the eight perturbed states from a test-trajectory frame for 200 frames under a shared forcing realization.

Hidden forcing. The surrogate never sees the OU forcing. To measure what the unobserved realization costs, we start from frame 10 of the reference trajectory, replay the recorded forcing for 20 frames so that the forcing state has reached its stationary amplitude, and from that single state advance one frame (31 solver steps) three times: under the recorded realization and under two fresh realizations of the same OU process (same forcing state, independent increments). The three next states differ by 0.80 to 0.89% of the velocity norm (the two fresh ones by 0.76%), whereas the state itself moves by 13.7% in that step, and energies agree to 0.3%. Two trajectories started 10^{-3} of the velocity norm apart under the same forcing realization separate with an e-folding time of about 23 steps, the $1.15\,T_L$ quoted in Section 2.2. On a held-out initial condition, eight trajectories under one forcing realization, one unperturbed and seven perturbed by isotropic Gaussian fields (dealiased and projected), give 1.1 to $1.2\,T_L$ by a log-linear fit over the first two to three turnover times, at amplitudes 10^{-3} and 10^{-2} alike. At 5×10^{-2} the growth departs from exponential within the first turnover time, so no e-folding time is fitted there (Fig. 1d).

D Training and generator settings

Backbone and training. The surrogate is a residual three-dimensional U-Net with 2.83×10^7 parameters that predicts the increment to the next frame from the current frame (four channels: three velocity components and pressure). The baseline model trains for 24 epochs with batch size 1 and gradient accumulation over 8 samples, learning rate 10^{-3} with cosine decay, on windows sampled with stride 4 along each trajectory. OAS fine-tuning keeps that batch size and accumulation. It fills a fraction 0.3 of the sample slots of each epoch with pairs, trained on the change-normalized pair loss of Section 3, and the rest with clean samples, trained on the one-step squared error. Each epoch has the length of a baseline epoch, and the clean windows and the pairs are each cycled through a fixed permutation across epochs. The main model's three stages run eight epochs each, 24 in total, the first at learning rate 3×10^{-4} with $K = 1$ and the other two at 10^{-4} with $K = 4$, each with cosine decay. Unrolled training applies the four-step loss to clean windows from the same pretrained baseline, on its own schedule rather than the OAS fine-tuning schedule above. Noise injection adds Gaussian noise of standard deviation 0.01 in normalized units to the input and subtracts it from the target increment, and the pushforward trick unrolls two steps and detaches the first. These follow the original descriptions: the pushforward loss is $\mathcal{L}(f_\theta(u^k+\epsilon), u^{k+1})$ with u^{k+1} the reference state, and unrolled training compares each unrolled prediction with the reference state at that step. Denoising refinement wraps the same U-Net in a decoupled flow-matching sampler with two refinement levels and two Euler substeps per level, at noise scales log-spaced in $[10^{-3}, 10^{-2}]$, so one rollout step costs five forward passes with fresh noise and a reported rollout is a single sample path. Half of each training batch trains the base predictor at refinement level $k = 0$, which maps an all-zeros input to the clean next frame conditioned on the previous frame, and the other half trains the flow velocity $v_\theta(x_\tau, x_{\mathrm{prev}}, k)$ with target $-\epsilon$. Optimizer, batch size, and gradient accumulation are those of the deterministic methods, and the residual connection is disabled because the base branch predicts absolute frames. At rollout step 1 on the evaluation initial conditions the five deterministic methods lie between 0.049 and 0.056 nRMSE. Denoising refinement predicts absolute frames from an all-zeros input and is not comparable there.

Generators. On-policy states are collected from rollouts of an intermediate fine-tuning stage at depths 4, 8, 16 and 32 steps and relabeled for four frames (the deep variant of Table 8 adds 64 and 96). Surrogate-attack states use 30 steps of projected gradient ascent (step size 0.1) on the frozen surrogate with the perturbation held on the sphere of relative amplitude 0.05 after dealiasing and divergence-free projection, one objective per state, at states of training trajectories only. The one-step-error objective of Section 4.3 recomputes the solver's answer at every ascent step

and treats it as a constant, so its gradient also runs through the network alone. The two residuals entering the physics-violating objective are the relative incompressibility residual of the output, $r_\nabla = \langle(\nabla\cdot u)^2\rangle / \langle|\nabla u|^2\rangle$, and the relative change the step makes to the total energy E. They enter with equal weight and no further scaling, the objective being $\log r_\nabla\big(f_\theta(x+\delta)\big) + \big|\log\big(E(f_\theta(x+\delta))/E(x+\delta)\big)\big|$. Table 3 gives the order in which the pool is assembled. DNS-probe directions are obtained by power iteration through the DNS at relative amplitude 10^{-3} (fourteen iterations) for the growing direction and, for the contracting direction, by drawing twelve isotropic Gaussian directions (dealiased and projected), advancing each through one frame of the DNS, and keeping the least amplified. The resulting states use relative amplitude 0.05. The main pool holds 132 on-policy, 66 surrogate-attack (22 per objective), and 44 DNS-probe (22 per family) pairs.

Table 3: Off-attractor supervision: pool assembly and staged fine-tuning. Inputs are the clean training states $\mathcal{C}$, the solver S with its recorded forcing realizations, the baseline model f_{θ_0} trained on $\mathcal{L}_{\text{clean}}$, the pair fraction p, and the label horizon K. The three stages of Section 3.4 differ in which generators have been added, and the first also in its horizon and learning rate (Appendix D). The on-policy states are re-collected once, against the model that the first stage produces.

Input: $\mathcal{C}$, S, f_{θ_0}, p, K; **Output:** θ_3

$\mathcal{G} \leftarrow \varnothing$
for stage $m = 1, 2, 3$ **do**
  if $m \le 2$: $\mathcal{G} \leftarrow \big(\mathcal{G} \setminus \text{on-policy}\big) \cup \{f^k_{\theta_{m-1}}(x) \,:\, x \in \mathcal{C},\ k \in \{4, 8, 16, 32\}\}$
  if $m = 2$: $\mathcal{G} \leftarrow \mathcal{G} \cup \{x + \delta^\star(x)\}$, $\delta^\star$ from PGD on frozen $f_{\theta_{m-1}}$, one objective per state
  if $m = 3$: $\mathcal{G} \leftarrow \mathcal{G} \cup \{x + \epsilon v\}$, v from power iteration, or least amplified of 12 random directions
  for each state $\tilde{x}$ newly added to $\mathcal{G}$ **do**
    $\hat{x} \leftarrow \Pi_{\text{div}}\big(\text{dealias}(\tilde{x})\big)$ *solver-compatible start*
    store $\big(y_0,\ S(\hat{x}), \ldots, S^K(\hat{x})\big)$, $y_0 = \tilde{x}$ if on-policy, else $\hat{x}$ *one solver call per state, offline*
  $\theta_m \leftarrow$ 8 epochs of $(1-p)\,\mathcal{L}_{\text{clean}}(\theta) + p\,\mathbb{E}_{\tilde{x}\sim\mathcal{G}}\,\mathcal{L}_{\text{pair}}(\theta; \tilde{x})$ from θ_{m-1} *Eq. 1*

E Extended results

The figures and tables of this section back the statements of Section 4 and follow its order.

Short rollout and error spectrum (Fig. 6). Panels (a) and (b) follow the six methods of Table 1 over the first twenty steps, (c) all but denoising refinement, and Fig. 5a,b is the cross-section of (a) to (c) at step 20. At step 20 OAS has the lowest pointwise error, and its correlation with the DNS field is within 0.01 of unrolled training, the second-best method on that error. Every method but OAS and unrolled training has passed an nRMSE of 1 by step 15, and those two reach it near step 20, before one e-folding time of the dynamics has passed. At step 15 the error of unrolled training exceeds OAS's by 12%. OAS then plateaus near 1.3 for as long as the DNS reference runs, just below the $\sqrt{2}$ level of two decorrelated fields with matching variance, while unrolled training (2.3) and the baseline U-Net (10) pass through that level as their energy inflates. Panel (d) is the measurement behind Section 2.4. The DNS spectrum falls by orders of magnitude from the forced shell to the dissipation scale, and the shell energy of the baseline U-Net's one-step error falls more slowly, so the relative error rises with wavenumber and reaches 304% at $k = 40$. Every step of a clean-trained rollout therefore injects small-scale content that the dynamics would dissipate.

Energy trajectories and dispersion (Fig. 3). The energy trajectories show blow-up for the baseline U-Net and the input-perturbation methods within the first 200 steps, delayed blow-up for unrolled training, a freeze at step 88 for denoising refinement, and for OAS, beyond the window of Fig. 3b, a slow drift out of the band at step 721 with no freeze before it. Fig. 3a shows the corresponding slices of every method on one initial condition. The input-perturbation methods and the baseline U-Net saturate the color scale within two turnover times, unrolled training leaves the band at step 90 on that condition, denoising refinement barely changes between the last two rows, and OAS stays in it to the end of the 200-step window, carrying large-scale structure of the kind the DNS column of Fig. 1a shows. Its slices nonetheless smooth as the rollout proceeds, while its late-window enstrophy runs well above the reference (Table 1): the rollout has not failed, and its small-scale statistics have drifted, which is the distinction the drift metrics record. The DNS-ensemble dispersion agrees with the energy view. OAS overshoots the DNS ensemble spread by about $3\times$ at step 100 and returns

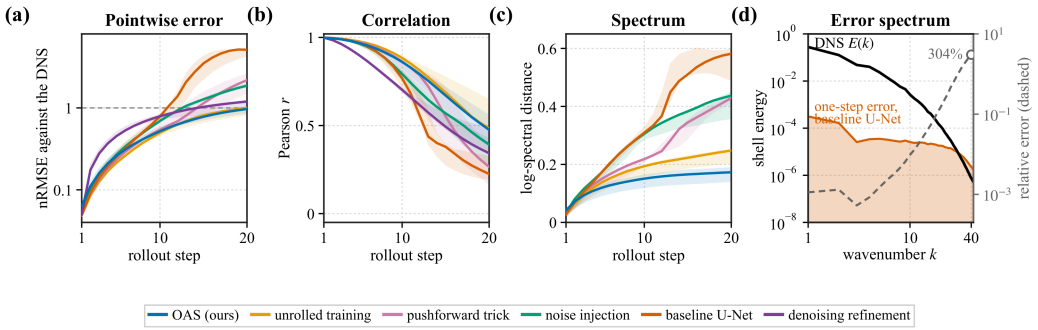

Figure 6: (a–c) Methods of Table 1 over steps 1 to 20, medians over three initial conditions (shaded: min to max): nRMSE against the DNS (dashed: 1), Pearson correlation, and log-spectral distance, with denoising refinement omitted from (c). (d) DNS energy spectrum $E(k)$ (black, time average), shell energy of the baseline U-Net's one-step error (filled), and their ratio (dashed, right axis).

to 0.91 at step 200, the only method that converges back toward the reference. Unrolled training overshoots by $4\times$ and blows up, and the input-perturbation methods collapse to a few percent of the reference. A longer reading gives the growth of the spread rather than its level at one step. Eight members perturbed at 10^{-3} of the velocity norm along a spectrum-shaped direction and followed to step 700 on one initial condition disperse exponentially for all three models measured, with a log-linear e-folding time of $1.1\,T_L$ for unrolled training, $1.6\,T_L$ for OAS, and $10\,T_L$ for the baseline model, against the 1.1 to $1.2\,T_L$ of DNS pairs (Appendix C). The rate does not separate the two survivors. Where the spread settles does. Over the last third of that window it holds at 0.62 of the velocity norm for OAS and runs to 0.99 for unrolled training as its energy inflates, while the baseline model's ensemble never leaves 0.09, the dispersion of a map collapsing toward a single field. This is one initial condition and eight members, so we read it for the shape of the three curves.

The onset of failure falls between the rows of Fig. 3 (Fig. 7). Fig. 3 samples three rollout steps, which is coarse where the deterministic methods break. The baseline U-Net is still at $K_t/K_0 = 1.01$ at step 5 and reaches 32.6 by step 20, so its whole blow-up falls between two of those rows. Fig. 7 keeps its layout, adds the clean-trained FNO, and samples thirteen steps on one held-out initial condition, six of them in the first ten. The three methods that blow up early develop grid-scale structure within the first turnover time, while their large scales are still in place, which is the order the diagnosis of Section 2.3 predicts. The FNO freezes at step 17, and its field changes little after that. Off-attractor supervision and unrolled training hold their large-scale structure to step 45, and past it the two part, unrolled training into a striped pattern and off-attractor supervision into progressively smoother fields. This is one rollout of each method.

Pool size (Table 4). Four pool sizes between 60 and 296 pairs, fine-tuned from the one-step baseline for eight epochs each, place the largest pool first on both criteria and above the smallest by 70% on band exit and 55% on the strict criterion. The two intermediate sizes do not fall between them, and the per-initial-condition spread is wide enough that the comparison carries only its endpoints. The reduced pools were drawn as random subsets of the pool carrying depths 64 and 96 rather than in the proportions of Section 3.4, so composition varies with size and the sizes are not comparable to the one-shot family below. In a single stage of eight epochs the largest of them reaches 79% of the staged main model's strict failure step, which is the sense in which a few hundred states are enough. Across the same four, the one-step validation loss rises as the pool grows, so the endpoints repeat along this axis the inversion Section 4.4 reports across methods.

One-shot family (Table 5). This family holds the base fixed and changes only the pool, fine-tuning the baseline model directly for the same 24 epochs on each reduced pool. Built in one pass this way the method is weaker than the staged main model and more seed dependent. Removing the on-policy states again collapses it, while removing the attack pairs or the two deepest depths leaves both criteria inside or at the edge of the five-seed range of the full pool (181 to 774 on band exit, 173 to 617 on the strict criterion), so in this setting, where configurations fail by freezing and its onset is seed dependent, those two contributions are not resolved. The staged training resolves all three.

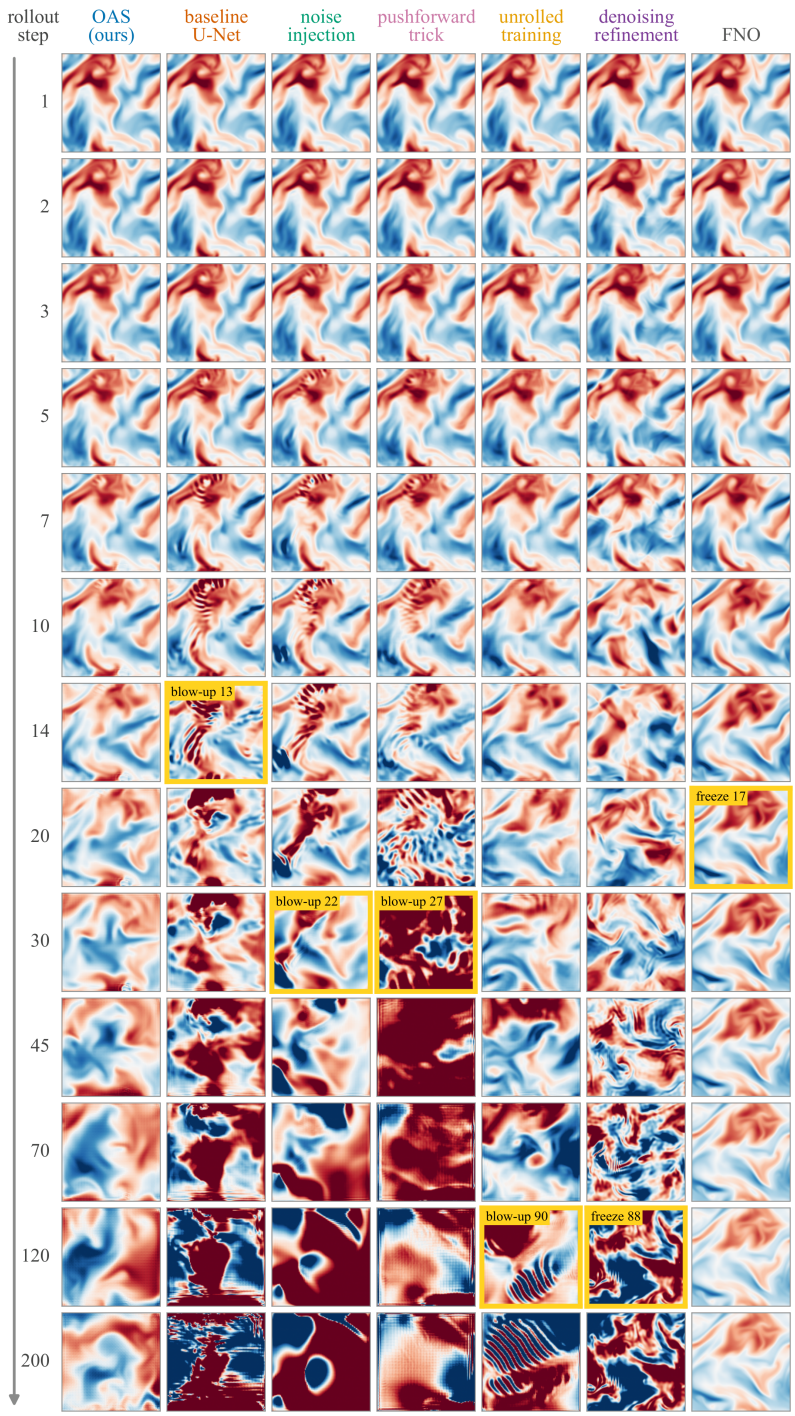

Figure 7: Onset of failure. Mid-plane u_x of the six methods of Table 1 and of the clean-trained FNO of Table 13 at thirteen rollout steps, six of them in the first ten, one column each and one row per step, from one held-out initial condition and on the color scale of Fig. 1a. Framed: the first step shown after the rollout fails, with its failure mode and failure step.

Generator combinations (Table 6, Fig. 4a). Table 6 lists the eight combinations of Section 4.3 with their freeze steps and failure modes. Rows 2 to 4 are the stages of the staged training, and row 5 adds depths 64 and 96 to the main pool. The strict criterion makes the sequence legible, since on energy alone the first stage (674) would place second in Table 1. The leave-one-out cells are built differently. Without DNS-probe states the model is the stage before they were added (139), whereas

Table 4: Pool size at a matched fine-tuning budget: the one-step baseline, eight epochs, and random subsets of the 296-pair pool that carries depths 64 and 96. Medians over three initial conditions with the per-condition strict values in brackets. Single runs.

pairs	one-step val. loss	band exit ↑	strict ↑	per condition
60	9.7×10^{-5}	388	368	[278, 368, 371]
120	1.05×10^{-4}	317	236	[219, 236, 448]
242	1.21×10^{-4}	650	319	[140, 319, 431]
296	1.21×10^{-4}	**661**	**569**	[558, 569, 613]

Table 5: One-shot family. Every configuration is fine-tuned from the baseline model on a reduced pool (rollouts to failure, three initial conditions, medians). The full-pool row is the seed whose band exit equals the five-seed median.

on-policy	surrogate-attack	DNS-probe	band exit	freeze	strict	failure mode
✓	✓	✓	524	173	173	freeze
✓	–	✓	639	636	636	freeze (late)
✓ (−depths 16, 32)	✓	✓	534	294	294	freeze
–	✓	✓	54	—	54	blow-up

without surrogate-attack states or without on-policy states the final stage is retrained on the reduced pool (396 and 54). Measured on single runs, the first stage's late-window enstrophy is $48\times$ the DNS value and its log-spectral distance 0.42, against $9.2\times$ and 0.24 for one run of the main model, whose seed medians Table 1 reports.

Table 6: Component ablation, all eight combinations of the three generators, plus the main pool extended to depths 64 and 96 in row 5 (rollouts to failure, three initial conditions, medians). Row 1 is the single-seed baseline run, the bold row and row 5 are five-seed medians (Table 16), and the rest are single runs. The *base* column gives the model each row was fine-tuned from, since the pool is assembled in stages and the leave-one-out rows below the rule therefore do not share one: rows 2 to 4 are those stages in order, the row without DNS-probe states is row 3 itself, and the two remaining leave-one-out rows retrain the last stage on the reduced pool. Table 5 repeats the comparison from a single fixed base.

on-policy	surrogate-attack	DNS-probe	base	band exit	freeze	strict	failure mode
–	–	–	one-step	13	—	13	blow-up
✓	–	–	row 1	674	86	86	freeze
✓	✓	–	row 2	377	139	139	freeze
✓	✓	✓	row 3	**721**	—	**721**	band exit (late)
✓ (+depths 64, 96)	✓	✓	row 3	796	635	635	freeze (late)
–	✓	–	row 1	76	—	76	blow-up
–	–	✓	row 1	18	—	18	blow-up
✓	–	✓	row 2	396	—	396	band exit (late)
–	✓	✓	row 2	54	—	54	blow-up

Inside the attack (Table 7, Fig. 4b). The full 2^3 grid of Section 4.3 appears in Table 7, with per-initial-condition values and the solver-in-the-loop control. Each cell is fine-tuned from the same base on 66 pairs and read as the band-exit step, so the grid compares objectives at a fixed small budget. No configuration in the grid freezes, and the strict failure step coincides with band exit. The freeze-plus-amplification cell was evaluated on a single initial condition. The freeze-inducing objective is the instructive case. It costs three steps against the baseline's 13 when used alone, adds nothing beside the physics-violating objective (53), and adds fifteen on top of the other two (61 to 76). Pairs that only push the model away from the identity train it to move without constraining how far it moves, and the amplification-inducing pairs supply the bound. The solver-in-the-loop control maximizes the one-step error with about thirty solver calls per state. These are single-seed configurations and the pairs of objectives lie within a few steps of one another, so the grid is read for its pattern and not its ordering.

On-policy depth and training hyperparameters (Table 8, Table 9, Fig. 4c). At the matched budget of Table 8 (same base, 8 epochs, 176 pairs, with only the maximum rollout depth of the on-policy states varying) the band-exit view is inflated by freeze survivorship. The strict failure step peaks at a maximum depth of 32 (355) and declines as deeper states re-introduce freezing, with the freeze count rising from none at 8 to all three initial conditions at 64 and 96, while the two shallowest depths alone take the base from 377 to 180 on band exit and from 139 to 180 on the strict criterion. Table 9 varies one hyperparameter at a time from the defaults of the staged main model (Appendix D). Band exit falls by 15 to 40% under the three changes and the strict failure step by 47 to 88%, since each change lets late freezing return.

Table 7: Attack objectives (2^3 grid, 66 pairs per cell) and the solver-in-the-loop control. Band-exit step over a 200-step window. †single initial condition.

objectives	band exit (3 ICs)	median
freeze-inducing	[9, 11, 10]	10
amplification-inducing	[51, 53, 57]	53
physics-violating	[50, 53, 58]	53
freeze + amplification†	[63]	63
freeze + violation	[52, 53, 63]	53
amplification + violation	[59, 61, 63]	61
all three	[60, 76, 86]	**76**
error maximization (solver in loop)	[45, 45, 70]	45

Table 8: On-policy depth ablation. Top: matched budget, only the maximum rollout depth of the on-policy states varies. Bottom: full budget. Single runs, except the two main-model rows (five-seed medians, Table 16).

max. on-policy rollout depth	band exit	strict	freeze events (of 3)
≤ 8	180	180	0
≤ 16	406	330	1
≤ 32	384	**355**	2
≤ 64	573	308	3
≤ 96 (all depths)	438	180	3
16 and 32 only	421	358	1
fine-tuning base (no on-policy pairs added)	377	139	3
main model (depths to 32)	**721**	**721**	0
main model + depths 64, 96	796	635	3
one-shot, full pool	524	173	3
one-shot − depths 16, 32	534	294	3

Table 9: Training hyperparameters of the staged main model (721 on both criteria), one varied at a time (rollouts to failure, three initial conditions, medians). Single runs.

hyperparameter	default	variant	band exit	strict
learning rate	10^{-4}	5×10^{-5}	436	379
pair-batch ratio	0.3	0.2	613	218
fine-tuning epochs	24	4	465	86

Label control (Table 10). The two rows of Table 10 marked † share the baseline model and an identical fine-tuning schedule, the 24 epochs, the same learning rate, and the four-step unrolled loss of Appendix D, so at a matched total budget of 48 epochs the training pairs are the only difference between 38 and 524 steps. The clean-window configuration also leaves the one-step validation loss at the base value. The 38-step and 259-step rows share their states, labels, loss, and base and differ only in schedule. The unrolled-training row is the single seed whose amplification Table 12 measures (259), and its three-seed median is 89 (Table 16). The matched-budget row controls for

the amount of new DNS data. Six fresh forcing realizations supply 246 clean windows and 984 new frames, against the 968 the pool holds, at the $0.05\,T_L$ frame spacing the pair pool uses, and they enter the same pair recipe from the same base model on the staged schedule of the main model, so the control matches OAS in samples and optimizer steps as well as in DNS frames. It fails by blow-up on every initial condition, at 47 steps on both criteria (47, 50, 39 by initial condition). Clean data at the same budget therefore reaches nine steps beyond the clean-window row, about one percent of the 683 steps between that row and the full method.

Table 10: Label control (rollouts to failure, three initial conditions, medians). †same baseline model and fine-tuning schedule, only the training pairs differ. ‡matched DNS budget, on the staged schedule of row 1. Row 1 is the five-seed median of Table 16. The rest are single runs.

states	**labels**	**loss protocol**	band exit	strict
off-attractor (OAS staged)	own future $S(\tilde{x})$	4-step unrolled on pairs	**721**	**721**
off-attractor (OAS one-shot)†	own future $S(\tilde{x})$	4-step unrolled on pairs	524	173
clean windows, fine-tuning schedule†	clean future $S(x)$	4-step unrolled	38	38
clean windows, 984 new DNS frames‡	clean future $S(x)$	4-step unrolled on pairs	47	47
clean windows, unrolled schedule	clean future $S(x)$	4-step unrolled	259	259
unrolled base + attack pairs	mixed	4-step unrolled	99	99

Amplification (Table 11, Table 12). Table 11 lists the one-step gains of the dynamics along the four directions of Section 2.3, all finite-amplitude single-step gains and not asymptotic rates. Table 12 lists the gains of the learned map for the seven configurations of Fig. 5c, each measured on the single run whose failure times the table reports, so they differ from the seed medians of Table 16. Along a random isotropic direction every learned map stays between 1.01 and 1.04, and the ordering of the configurations is set by the worst direction alone. That direction is not the one the dynamics amplifies: the cosine of Section 2.3 stays below 0.004, which is why the model-independent DNS-probe states alone leave the amplification above the baseline value (3.17) while the attack states alone bring it to 2.10.

Table 11: One-step amplification λ of the dynamics along four directions, measured with the DNS at relative amplitude 10^{-3}.

direction	λ
worst (power iteration)	1.036
random isotropic	0.417
model's actual error direction	0.94
spectrum-shaped	1.016

Table 12: One-step amplification of the learned map along its worst direction and along a random one, with the failure times of the same run, for the seven configurations of Fig. 5c. Each is the single run whose failure times the table reports. Both directions are probed at relative amplitude 10^{-3}, as in Table 11. The random-direction column is not bolded: it is flat across configurations and far from the 0.417 of the dynamics, so it distinguishes nothing (Section 4.4).

configuration	worst dir.	random dir.	band exit	strict
DNS-probe states only	3.17	1.03	18	18
Baseline U-Net	2.86	1.04	13	13
Noise injection	2.61	1.03	29	29
Pushforward trick	2.28	1.03	24	24
Surrogate-attack states only	2.10	1.03	76	76
Unrolled training	1.16	1.03	259	259
OAS (ours)	**1.01**	1.01	**721**	**721**

Backbones (Table 13). FNO and TFNO are spectral backbones, DPOT and FactFormer attention backbones, all trained on the one-step objective on the same data, every one a single run. All four

carry at least twice the U-Net's one-step error, from $2.1\times$ for FNO to $3.3\times$ for DPOT, and fail within 6 to 19 steps on the strict criterion, so one-step error does not order the five by when they fail. Neither of the two error columns orders the other either. FactFormer is the more accurate of the two attention models at step 1 and the earlier of the two to fail, and both spectral backbones reach step 15 with less pointwise error than the U-Net whose one-step error is less than half of theirs, though every backbone is past the $\sqrt{2}$ level of two decorrelated fields by then. The two criteria separate FNO from the rest. It keeps its energy in band thirty times longer than the U-Net, 408 steps against 13, and has stopped moving by step 17. That combination is what spectral truncation would produce. Its spectral convolutions keep a fixed set of low wavenumbers, which can bias the map toward a smoothed field, and the large scales it retains hold the energy in band long after the map has stopped transferring any. A long band-exit step is therefore not stability here.

Table 13: Clean-trained backbones. Rollouts to failure from three initial conditions, medians with the per-condition values in brackets. Single runs.

backbone	nRMSE@1	nRMSE@15	band exit	strict
U-Net (baseline)	0.049	3.498	13 [12, 13, 15]	13
FNO (Li et al., 2021)	0.102	2.120	408 [352, 408, 466]	17 [11, 17, 18]
TFNO (Kossaifi et al., 2024)	0.128	2.941	33 [31, 33, 34]	19 [18, 19, 23]
DPOT (Hao et al., 2024)	0.164	5.831	11 [7, 11, 12]	11
FactFormer (Li et al., 2023)	0.119	5.274	6 [5, 6, 7]	6

Choosing the backbone of the main study (Table 14). Table 13 separates two criteria a backbone could be chosen by, and they disagree. The U-Net has the lowest error at one step, and the FNO keeps its energy in band the longest. Neither criterion is safe on its own. The U-Net's advantage at step 1 has inverted by step 15, where both spectral backbones are more accurate than it despite carrying two to three times its one-step error, and the FNO's 408 steps in band are those of a frozen map. We therefore ran the method on both candidates rather than settling the question by either number.

Off-attractor supervision on a spectral backbone. The staged recipe of Section 3.4 was applied to the FNO as it was to the U-Net, in the same three stages of eight epochs, with its own pool collected against it. Step-15 error and strict failure step each improve by about a factor of two, against $4.7\times$ and $34\times$ for the same method on the U-Net (Table 14). The strict criterion is the one that reads this backbone, whose clean-trained band exit of 408 sits far beyond its freeze at 17, so energy alone measures a map that has already stopped moving. The failure mode is also unchanged. The FNO still freezes, later, whereas the U-Net under the same treatment freezes in no seed at all. The U-Net is the backbone of the main study for that reason, and not for its one-step error, which Section 4.4 shows does not order the rollout. It is the backbone the method has the most room to repair. The full gains are therefore quantified on the convolutional family, and a fraction of them carries to a spectral one.

Table 14: Off-attractor supervision on the two candidate backbones, at rollout step 15 and on the strict criterion, over rollouts to failure from three initial conditions. The clean-trained rows are the single run of Table 13 (FNO) and the three-seed median of Table 1 (U-Net). Fine-tuned rows are seed medians. Brackets show individual seed results or their range.

backbone		nRMSE@15 ↓	strict ↑
FNO	clean-trained	2.120	17
	+ OAS (3 seeds)	1.141 [1.077, 1.141, 1.232]	35 [34, 35, 37]
	gain	$1.9\times$	$2.1\times$
U-Net	clean-trained	3.498	21
	+ OAS (5 seeds)	**0.740**	**721** [705–784]
	gain	$\mathbf{4.7}\times$	$\mathbf{34}\times$

Second regime (Table 15). The dataset at $\nu = 0.010$ passes the validation checks of Appendix C. Its pool of 132 pairs (54 on-policy, 54 surrogate-attack, 24 DNS-probe) was built in one shot against

that regime's baseline U-Net, and the fine-tune runs the same 24 epochs as in the main regime. The per-initial-condition values spread by 25 steps for the baseline model and by 86 steps for OAS on band exit.

Table 15: Second regime, $\nu = 0.010$ ($\mathrm{Re}_\lambda \approx 35$), with its own dataset and pool. Rollouts to failure from three initial conditions, medians with the per-condition values in brackets. Single runs.

	pairs	epochs	band exit ↑	strict ↑
Baseline U-Net	0	24	77 [59, 77, 84]	77
OAS (one-shot)	132*	24	**583** [503, 583, 589]	**540** [485, 540, 545]
gain over baseline			7.6×	7.0×

*54 on-policy, 54 surrogate-attack, 24 DNS-probe pairs.

Seeds (Table 16). Over five seeds the main model spans 705 to 784 on both criteria, the variant with depths 64 and 96 added 685 to 833 on band exit and 583 to 718 on the strict criterion, and the one-shot variant 181 to 774 and 173 to 617. Staging therefore narrows the seed spread as well as raising the median. Unrolled training spans 52 to 259 over three seeds and the baseline model 13 to 22. Row 1 of Table 6 is the single-seed baseline run at 13, whose three-seed median is 21 (Table 1).

Table 16: Seed robustness. Rollouts continued to failure, from three initial conditions. Median over seeds, with the per-seed medians in brackets.

	seeds	band exit ↑	strict ↑
Baseline U-Net	3	21 [13, 22, 21]	21
Unrolled training	3	89 [259, 52, 89]	89 [259, 52, 89]
OAS one-shot	5	524 [524, 399, 181, 774, 617]	247 [173, 399, 181, 247, 617]
OAS staged (main)	5	**721** [705, 719, 721, 723, 784]	**721** [705, 719, 721, 723, 784]
OAS staged + depths 64, 96	5	796 [685, 794, 796, 820, 833]	635 [583, 593, 635, 647, 718]

F Reproducibility details

Training and evaluation ran on two NVIDIA RTX 5090 (32 GB) workstations and, for five of the one-shot ablation configurations, on cloud A100 (40 GB) instances. Building and labeling the whole pool takes about as much wall-clock time as two of the 48 training epochs behind the main model, most of it the attack search on the network rather than the solver calls. The configurations of Table 16 and the FNO under off-attractor supervision (Table 14) were trained with several seeds, and the ablation configurations, the amplification probes, and the clean-trained backbones are single runs. Rollouts on GPUs are not bit-reproducible, which sets the evaluation noise floor: band-exit steps carry a run-to-run floor of 3 to 6% and strict failure steps one of 6 to 11%. Relabeling is bit-reproducible because it is driven by a recorded forcing realization at a fixed time step.