PREDICTING UNSEEN DYNAMICS WITH NEURAL EMULATORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Can neural emulators predict dynamical regimes that are absent from their training data? Our main results concern two-dimensional Kolmogorov flow: an emulator trained only on chaotic aspect ratios predicts laminar dynamics beyond an α-bifurcation, provided the architecture does not carry a symmetry the equations lack. A standard convolution is equivariant to every y-shift, an invariance the forced equations do not possess, and drifts unphysically along that unconstrained direction; an $\mathscr{S}^2$-aware convolution, equivariant only to the discrete y-shifts the shift-reflect symmetry permits, does not. As supporting multi-system evidence, a Kuramoto–Sivashinsky emulator recovers excluded relaminarisation, warm-up, and kink transitions and extrapolates to a domain $2.4\times$ larger than any seen in training, while a partially observed beta-plane emulator recovers coalescence, nucleation, and stationary four-jet states excluded from training. Our failures are as informative as our successes: the chaotic transient approaching the Kolmogorov attractor is not recovered, and the learned map retains a dependence on the emulator timestep that a converged evolution map would not have. Both point to what matters being whether the target is represented locally in the training measure, at the scale the architecture sees, rather than whether the whole state has been seen before.

1 INTRODUCTION

Dissipative chaotic systems arise in nature in numerous forms, with turbulence and the climate being notable examples. Modelling of these systems is an important but challenging task due to the intricate geometry of their state-space manifolds Hopf (1948). Machine learning has recently shown potential in predicting short-term solutions and long-term statistics of such chaotic systems Brunton et al. (2020). A more specific but less common application of ML is cross-dynamics generalisation: predicting events that are unseen in the training set. Such a task leans towards discovery rather than interpolation, and tests whether an emulator has learned the underlying evolution rule rather than memorised observed patterns.

Our main technical focus is 2D Kolmogorov flow, with supporting evidence from the Kuramoto–Sivashinsky (KS) equation and beta-plane turbulence, using parameter-conditioned attention emulators whose architecture and training details follow Shokar et al. (2025b). Throughout, we study recovery of regimes deliberately excluded from training rather than interpolation across parameter values (table 1).

Our contributions are as follows:

- For 2D Kolmogorov flow, we show that an emulator trained only on chaotic aspect ratios can predict laminar dynamics beyond an aspect-ratio bifurcation—to our knowledge the first SciML study of generalisation across α—while exposing current limitations (spurious nearby attractors, timestep sensitivity, and unrecovered chaotic transients).
- To obtain physically admissible laminar rollouts we introduce an $\mathscr{S}^2$-aware convolution, equivariant *only* to the discrete y-shifts permitted by the shift-reflect symmetry, prove which subgroup it is equivariant to, and show qualitatively that a standard convolution's spurious continuous y-symmetry is what produces unphysical drift outside the training regime.

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.

- For the KS equation, we show that this generalisation survives when 'kink' transitions and the states preceding them are stripped from training altogether, and that the same conditioned emulator extrapolates zero-shot to $L = 400$, a domain $2.4\times$ larger than any seen during training.
- For beta-plane turbulence, a two-layer partially observed probabilistic emulator—extending the single-layer conditioned approach of Shokar et al. (2025b)—recovers coalescence, nucleation, and stationary four-jet dynamics when trained only on translating states with at most three jets and no transitions.

The remainder of the paper is organised as follows. Section 2 reviews related work, section 3 describes the emulators and the Kolmogorov $\mathscr{S}^2$-aware convolution, section 4 presents the experiments, and section 5 discusses shared implications and limitations.

2 Related Work

Emulators for dissipative chaos span echo-state networks Jaeger (2001); Pathak et al. (2018); Margazoglou & Magri (2023); Özalp & Magri (2025); Tretiak et al. (2025); Racca & Magri (2022); Pershin et al. (2023), transformers Vaswani et al. (2017); Li et al. (2023); Shokar et al. (2024); He et al. (2025), neural operators Li et al. (2020; 2021); Wu et al. (2025); Helwig et al. (2023), and diffusion models Lippe et al. (2023); Luo et al. (2024); Parker & Schneider (2026). Most of this literature targets short-term forecast skill or long-term statistics *within* the training regime, or extends forecasts across *parameter* ranges by conditioning on PDE coefficients Shokar et al. (2025b); recovering regimes excluded from the training measure is a distinct and less studied question, and is our focus.

"Unseen" here is not one task. At fixed parameters, models may be asked to recover rare or excluded events: extremes and laminar behaviour from incomplete or turbulent-only training Racca & Magri (2022); Pershin et al. (2023), and tipping points from pre-tipping data alone Zhuge et al. (2025). Our KS and beta-plane experiments follow this axis. Closest to the KS setting, Shokar et al. (2025a) predict relaminarisation and warm-up in KS using a neighbourhood-attention transformer Hassani et al. (2023) conditioned on domain length via FiLM modulation Perez et al. (2018); we adopt that architecture and add restricted-training ablations that remove kink transitions and the states preceding them, zero-shot extrapolation to a domain $2.4\times$ the largest seen in training, and the comparison across Kolmogorov flow and beta-plane turbulence that isolates which architectural properties the generalisation depends on. Separately, models may be asked to generalise across parameters or forcing: phase-space coverage failures in low-dimensional Lorenz systems Scher & Messori (2019), out-of-distribution PDE surrogates in the small-data regime Nguyen & Sandfeld (2026), and grey-swan extremes in weather emulators Sun et al. (2025). Our Kolmogorov experiments test generalisation across aspect ratio α, well studied in dynamical systems Chandler & Kerswell (2013); Lucas & Kerswell (2014; 2015) but not, to our knowledge, in SciML.

Pershin et al. (2023) asked the same excluded-regime question first, showing that an echo state network trained only on turbulent episodes of the nine-mode Moehlis–Faisst–Eckhardt model recovers the laminar state and the statistics of transitions in both directions. Because that network's input is the entire nine-dimensional state, neither of the issues we find to matter—the scale of the effective receptive field, and spatial symmetries the architecture carries but the equations lack—can arise there, whereas our systems are spatially extended and we additionally cross a parameter bifurcation, change the domain size, and withhold half the state vector.

Symmetry-aware architectures aim to encode the true group structure of the flow rather than learn it from data. For Kolmogorov flow, prior work uses symmetry-reduced charts and latent neural ODEs Linot & Graham (2020); Linot et al. (2023); Pérez De Jesús et al. (2025), Siamese discrete-symmetry branches Kneer et al. (2021), translation-aware latent Fourier modes for invariant solutions Page et al. (2021), and group-equivariant spectral layers Helwig et al. (2023); Cohen & Welling (2016). Parker & Schneider (2026) break spurious continuous y-translation by force injection while preserving discrete shift-reflect structure, and seed a numerical solver with generated trajectories so that correctness is certified by the solver rather than the model. Our $\mathscr{S}^2$-aware convolution instead enforces equivariance *only* to the discrete y-shifts that symmetry permits, which we find more effective at suppressing unphysical continuous drift, and our prediction is the emulator's own time-marched output, with no solver in the loop.

3 MODELS AND SYSTEMS

Across all three systems, an emulator learns a discrete-time evolution map $\mathcal{F}_{\Delta t}$ from input–output snapshots of a training measure from which a regime has been deliberately removed, and is then rolled out autoregressively into that regime. Architectures, conditioning, and training objectives are system-dependent, because the systems differ in what is observable and in whether the target is predictable pointwise. Full equations, protocols, and loss formulas are collected in sections A, C, D and F.

3.1 KOLMOGOROV FLOW EMULATOR

We study 2D Kolmogorov flow at $\mathrm{Re} = 50$ and forcing wavenumber $\kappa = 4$, with aspect ratio $\alpha = L_y/L_x$ as the control parameter and vorticity as the prognostic variable (section D). Training uses chaotic trajectories for $\alpha \in [0.92, 1.00]$; inference tests α below this range. Training minimises a composite loss combining a Sobolev-weighted vorticity term, a streamfunction term enforcing the Poisson relation, and a phase term (section F). The convolutional emulator is conditioned on α and replaces standard convolutions with the $\mathscr{S}^2$-aware layer, which we outline in what follows.

Kolmogorov flow is forced by a y-dependent body force $\propto \sin(\kappa y)$, so continuous translations in y are *not* symmetries of the PDE. The relevant discrete structure is the shift-reflect symmetry $\mathscr{S}$ Chandler & Kerswell (2013), which combines a shift by the forcing half-period π/κ with a reflection in x and sign changes in u and ω (section D); the half-period shift on its own is therefore not a symmetry. Its square $\mathscr{S}^2$ *is* a pure translation, by the full forcing period $2\pi/\kappa$—equivalently N_y/κ grid points on a grid of N_y points spanning $y \in [0, 2\pi)$. The admissible y-translations are thus exactly the κ integer multiples of N_y/κ. A standard convolution on a periodic grid is nevertheless equivariant to every pixel shift in $\mathbb{Z}_{N_y}$. If such a layer is used throughout the emulator, the architecture itself carries a spurious continuous y-symmetry that the true system does not possess.

This mismatch is especially damaging for the out-of-distribution laminar solutions we wish to predict. Laminar Kolmogorov states are steady (or nearly so) in the frame of the forcing, so an unconstrained convolutional emulator can accumulate a continuous phase drift in y while still producing a low one-step training loss on chaotic data. The drift is unphysical and prevents the model from locking onto the correct laminar attractor at reduced α.

We enforce the correct discrete equivariance by tying convolutional kernels to the forcing phase of the output row. Let $\phi_y(r) = r \bmod (N_y/\kappa)$ label the position of row r within one forcing period, so that ϕ_y takes N_y/κ values. The layer stores a bank of N_y/κ kernels $\{\mathbf{W}_0, \ldots, \mathbf{W}_{N_y/\kappa-1}\}$ and applies

$$\mathrm{Conv}_{\mathscr{S}^2}(\mathbf{h})(r, c) = \sum_{a,b} \mathbf{W}_{\phi_y(r)}(a, b)\, \mathbf{h}(r + a, c + b), \tag{1}$$

with spatial indices taken periodically. Rows separated by a multiple of N_y/κ share a kernel; rows at different positions within a forcing period may use different kernels. The layer therefore cannot treat an arbitrary y-shift as a symmetry, but it remains equivariant to the κ discrete shifts that $\mathscr{S}^2$ permits.

Proposition 1. *If $r \mapsto \mathbf{W}_{\phi_y(r)}$ has fundamental period N_y/κ, then eq.* (1) *is equivariant in y exactly to*

$$\mathcal{G}_{\mathscr{S}^2} = \left\{0, \frac{N_y}{\kappa}, 2\frac{N_y}{\kappa}, \ldots, (\kappa - 1)\frac{N_y}{\kappa}\right\} \subset \mathbb{Z}_{N_y},$$

and not to shifts outside this subgroup. The proof is given in section E.

We tie kernels to $\mathscr{S}^2$ rather than to $\mathscr{S}$ itself because $\mathscr{S}$ is not a pure translation: it reflects the x-coordinate and flips the signs of u and ω, so it cannot be imposed by tying kernels along y alone. Enforcing it would additionally require reflection-aware kernels and sign-equivariant activations Parker & Schneider (2026).

This construction is related to tiled convolutions Ngiam et al. (2010), but tiles only in y and uses the known forcing period rather than an arbitrary partition. It also differs from learning a soft partial equivariance from data Romero & Lohit (2022): here the subgroup required by the PDE is hard-coded. An alternative symmetry-breaking strategy is to concatenate the forcing field $\cos(\kappa y)$ to the

Table 1: Overview of the cross-dynamics experiments. The effective receptive field is the span of input that influences one output point once all layers are stacked: $N(K-1)+1$ grid points for N blocks of window K. The relevant comparison is with the *target* domain, which is the scale at which the coverage argument of section 5 is stated.

System	Eff. receptive field	Training restriction	Unseen target
Kolmogorov	65 pts	Chaotic $\alpha \in [0.92, 1]$	Laminar flow at lower α; $\mathscr{S}^2$ ablation
KS	65 pts	Chaos only; kink transitions excluded; $L \leq 164$	Relaminarisation, warm-up, kink transitions; $L = 400$
Beta-plane	257 pts (global)	≤ 3 translating jets, no transitions; only ψ_1 observed	Coalescence, nucleation, stationary four-jet state

convolutional input Parker & Schneider (2026). Force injection can suppress continuous drift, but it introduces an auxiliary channel whose amplitude is not constrained to match the forcing already implicit in the data; empirically this yields unsteady laminar rollouts. The ablation in fig. 3 compares $\mathscr{S}^2$-convolution, standard convolution, and force injection on the same out-of-training laminar task.

3.2 Kuramoto–Sivashinsky emulator

The KS equation for $u(x,t)$ on a periodic domain of length L is $\partial_t u = -u\partial_x u - \partial_{xx} u - \partial_{xxxx} u$. As L increases the system undergoes a sequence of bifurcations toward spatiotemporal chaos. We approximate the discrete evolution operator $\mathcal{F}_{\Delta t}$ with a neighbourhood-attention transformer conditioned on L via FiLM modulation Hassani et al. (2023); Perez et al. (2018). Local attention imposes a translation-equivariant stencil that can be applied on grids of any length, so the same weights handle different L without fixing the number of Fourier modes a priori. The model is pretrained at $L = 22$ (5,000 snapshots) and fine-tuned on $L \in \{48, 64, 96, 128, 164\}$ (500 snapshots each), then tested on held-out regimes including relaminarisation at $L = 56$ and zero-shot extrapolation to $L = 400$ (section A). Training minimises one-step MSE; rollouts are autoregressive and scored with MAE/RMSE.

Eight blocks of window $K = 9$ give an effective receptive field of 65 grid points, or 26 length units at the fixed grid spacing $\Delta x \approx 0.4$. Because this is fixed in physical units while L varies, the same stencil spans the whole $L = 22$ pretraining domain but only 6.5% of $L = 400$, moving from global to strictly local as the target grows (section A).

3.3 Beta-plane emulator

We study two-layer quasi-geostrophic beta-plane turbulence, a standard idealisation of atmospheric and oceanic jets. Writing the upper-layer evolution schematically as $\partial_t \psi_1 = F(\psi_1; \beta) + G(\psi_1, \psi_2; \beta, R_d)$, the unresolved baroclinic interaction G couples the layers through the deformation radius R_d. Following the parameter-conditioned probabilistic approach of Shokar et al. (2025b), which studied a single-layer model, we use a neighbourhood-attention transformer conditioned on (β, R_d) via adaptive layer normalisation, with latent Gaussian sampling and a fair-CRPS objective with spectral regularisation to produce ensemble forecasts. Here we extend that setting to the harder two-layer problem under partial observation: the emulator receives only upper-layer history $(\psi_1(t-1), \psi_1(t))$ and must infer G without ever seeing the lower layer ψ_2. Unlike the KS emulator this network is not local at the scale of its target—sixteen blocks of window $K = 17$ span 257 grid points against a 128-point domain—so what this experiment isolates is partial observation and an excluded large-scale configuration, not locality (section C).

4 Experiments

4.1 Kolmogorov flow

Training uses nine chaotic DNS trajectories for $\alpha \in [0.92, 1.00]$ with transients removed; Lyapunov exponents, spectra used to choose Δt, and the training phase-space density are reported in section G. Autoregressive rollouts are initialised from a noise-perturbed laminar state (section H) and evaluated at $\alpha \in \{0.8250, 0.8500, 0.8750\}$, spanning the DNS bifurcation from chaos ($\alpha = 0.8750$) to laminar

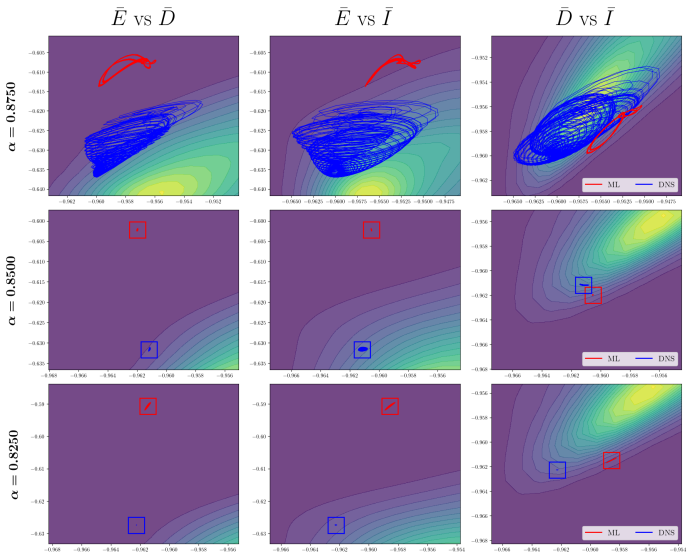

Figure 1: Phase-space trajectories of ML and DNS overlaid on the training joint PDF, for $\alpha \in \{0.8250, 0.8500, 0.8750\}$, where laminar solutions are marked by square bounds.

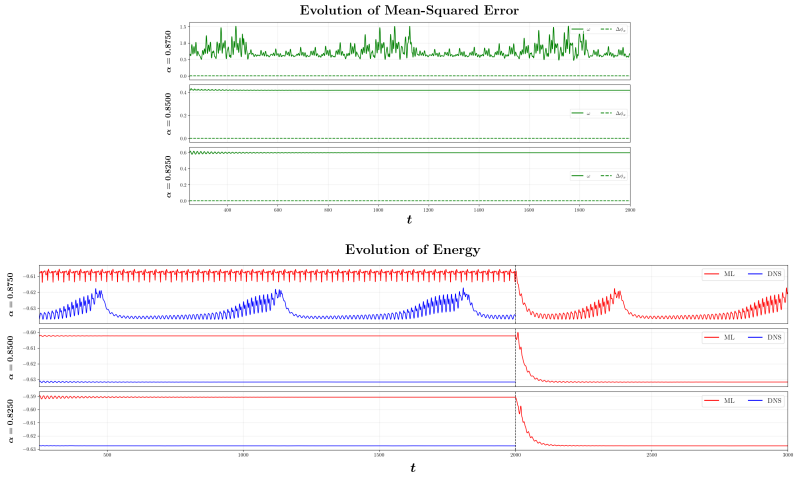

Figure 2: Kolmogorov attractor diagnostics at $\alpha \in \{0.8250, 0.8500, 0.8750\}$. Top: post-transient error of ML relative to DNS. Bottom: post-transient dissipation, where the vertical line marks the switch from ML rollout to DNS continuation.

flow ($\alpha \leq 0.8500$). The emulator timestep is set a priori from the training spectra: candidate values $\Delta t \in \{1.0, 0.5, 0.2\}$ follow from the 90%, 95% and 99% spectral cut-offs of the volume-averaged input energy, the most stringent of the three diagnostics, and we report $\Delta t = 0.5$, which resolves 95% of its variability. The other two candidates behave as an under- and over-resolved discretisation respectively: $\Delta t = 1$ is overly diffusive and laminarises at every α (an effect discussed by Chuang & Barba (2023)), while $\Delta t = 0.2$ does not hold the intended laminar solutions. That the outcome depends on Δt at all is a limitation of the learned map rather than a property of the flow, and we return to it in section 5; the sensitivity analysis is in section G. Figure 1 shows that the emulator predicts laminar states at $\alpha = 0.8500$ and 0.8250, while at $\alpha = 0.8750$ it produces a quasi-periodic orbit rather than the chaotic DNS attractor. Energy also appears more out-of-distribution than dissipation or energy input.

Figure 2 quantifies the error between ML and DNS. Phase MSE remains moderate across all three aspect ratios, while vorticity errors are large precisely when the predicted and DNS attractors differ qualitatively (periodic ML vs chaotic DNS at $\alpha = 0.8750$). The laminar predictions at $\alpha = 0.8500$ and 0.8250 nevertheless retain reasonable field accuracy, which is notable because inference must

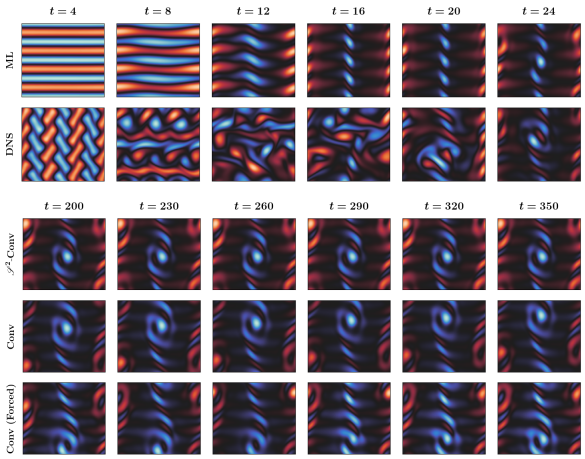

Figure 3: Top: ML vs DNS vorticity in the transient at $\alpha = 0.8500$. Bottom: Vorticity evolution at $\alpha = 0.8500$ for $\mathscr{S}^2$-aware convolution, standard convolution, and standard convolution with forcing injected into the input.

traverse a transient that is absent from training. Pointwise error against a chaotic reference saturates at the level of two unrelated draws from the attractor once trajectories decorrelate, whatever the quality of the model, so the $\alpha = 0.8750$ curve should be read as confirming that the two attractors differ rather than as measuring by how much; the qualitative claims are carried by the phase-space and dissipation diagnostics above. Because 2D Kolmogorov flow is multistable, the quasi-periodic orbit at $\alpha = 0.8750$ could in principle be a true attractor that the DNS simply did not reach from the same initial condition. To test this, we continue the final ML snapshot under the DNS solver (fig. 2). The continued trajectory leaves the ML attractor and relaxes toward the DNS one; even the predicted laminar states adjust toward the exact laminar solutions of the PDE. The emulator therefore learns a nearby but spurious manifold with a different set of stable attractors.

That transient itself is not recovered. Unlike the KS experiments below, where warm-up from noise succeeds after training only on long-term chaos, the Kolmogorov emulator produces a smooth approach from the perturbed base state to the warmed-up regime (fig. 3), missing the chaotic DNS transient; we return to why in section 5.

Finally, fig. 3 also compares the $\mathscr{S}^2$-aware convolution against two alternatives on the same out-of-training laminar task and the same training data. It sustains a laminar solution without continuous y-drift; a standard convolution drifts unphysically in y; and force injection (Parker & Schneider, 2026) prevents drift but yields an unsteady field despite a laminar DNS reference. The pathology that the inductive bias removes is one that the training objective is poorly placed to detect: chaotic training states have no persistent y-phase to preserve, so a spurious continuous y-symmetry costs little on the training measure and surfaces only once the target state is a steady one, outside that measure. The comparison is qualitative and is not parameter-matched—the $\mathscr{S}^2$ layer holds N_y/κ kernels where a standard convolution holds one—so it supports the claim that the spurious continuous y-symmetry is the mechanism of the failure, and that removing it suffices here, rather than that this particular layer is the only remedy.

4.2 Kuramoto–Sivashinsky

The KS experiments probe a complementary question at (mostly) fixed L: can an emulator recover regimes and transitions that were deliberately stripped from training, and can the same conditioned architecture extrapolate in domain size?

We first test relaminarisation and warm-up at $L = 56$ after training only on chaotic dynamics for $L \neq 56$. Relaminarisation is the collapse of chaos onto a laminar travelling wave; warm-up is the opposite transient, in which small-amplitude noise organises into localised steep gradients ("kinks") and then fully developed chaos. Neither class of trajectory is present in this training set. Figure 4 shows that the emulator recovers both: the timing of the laminar collapse and the final wave pattern, and the spatial structure of kink nucleation from noise whose amplitude is more than two orders of

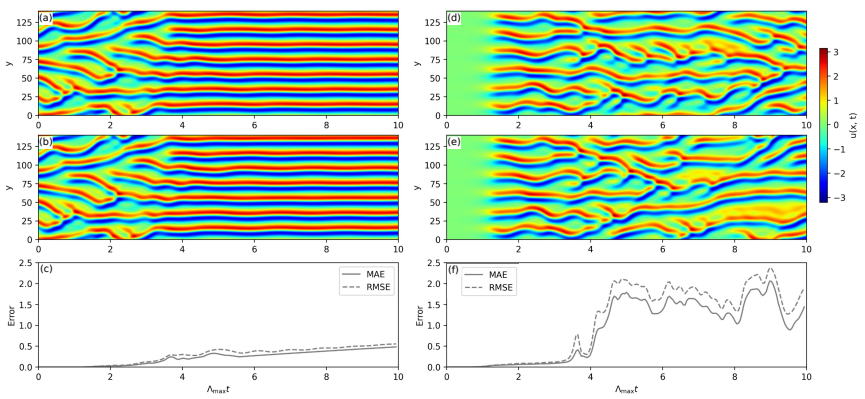

Figure 4: Out-of-training-distribution KS dynamics at $L = 56$. (a,d) DNS relaminarisation and warm-up; (b,e) NN from the same initial conditions; (c,f) MAE and RMSE.

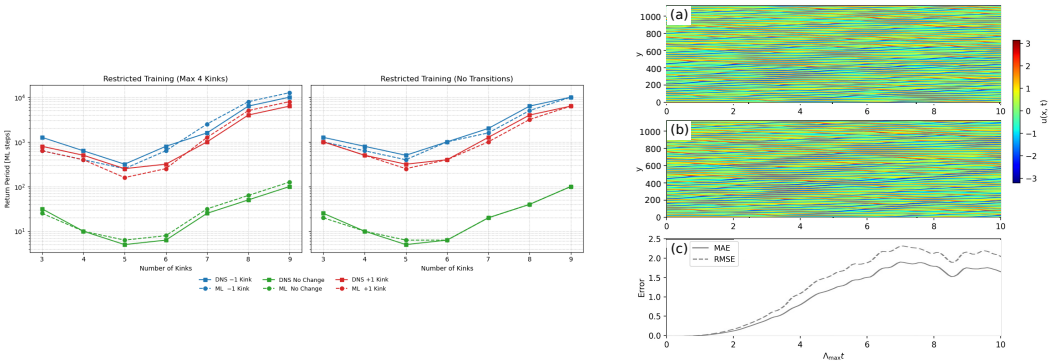

Figure 5: KS restricted-training and extrapolation results; solid: DNS, dashed: NN. Left: kink-count return periods for emulators trained only on ≤ 4 kink states (a) and with all kink-count transitions excluded (b). Right: zero-shot extrapolation to $L = 400$ after pretraining at $L = 22$ and fine-tuning on $L \in \{48, 64, 96, 128, 164\}$, showing (a) DNS, (b) NN, and (c) MAE and RMSE.

magnitude below the training fluctuations. Short-term accuracy persists for roughly four Lyapunov times $1/\Lambda_{\text{max}}$ (section A) before the chaotic rollout diverges onto an equally plausible trajectory. Errors grow slowly after relaminarisation due to a translational phase drift that remains a valid solution of the PDE.

To test whether success merely reflects leakage of precursors into training, we retrain at fixed $L = 56$ on artificially restricted datasets. Laminar travelling waves at this L carry seven equally spaced kinks, so restriction (i) retains only states with at most four kinks—excluding both high-kink chaos and any near-laminar precursors. Restriction (ii) exposes the full range of kink counts but removes every kink-count transition together with the snapshots immediately preceding each, and uses only 100 training frames. In both settings the emulator recovers unseen $5+$ kink states, transition probabilities that match DNS (left panel of fig. 5), and the relaminarisation/warm-up cases of fig. 4. Excluded-regime forecasting therefore does not require examples of the target events or of the states that immediately precede them.

Finally, after pretraining at $L = 22$ and fine-tuning on $L \in \{48, 64, 96, 128, 164\}$, the same architecture extrapolates zero-shot to $L = 400$ (right panel of fig. 5), $2.4\times$ the largest fine-tuning domain, with RMSE reaching $O(1)$ at $5/\Lambda_{\text{max}}$. Appendix controls isolate what this requires (section B): neither an MLP nor an FNO transfers across domain size, the FNO because its trainable mode budget is set by the smallest training domain.

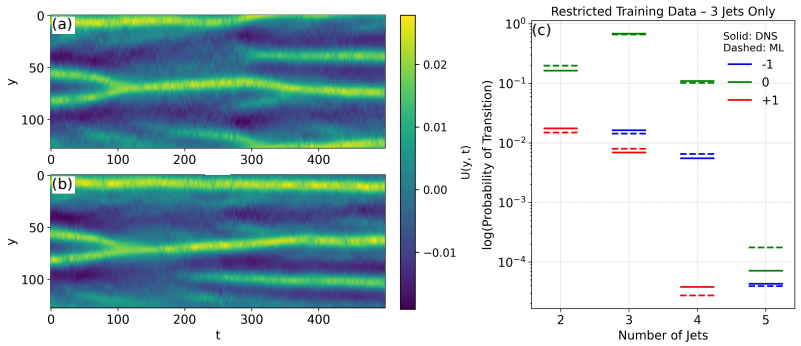

Figure 6: Out-of-training-distribution beta-plane dynamics with training restricted to three-jet states and no transitions. (a) DNS upper-layer zonal-mean zonal velocity $\bar{u}_1(y,t)$; (b) NN from the same initial condition; (c) jet-count transition probabilities over the rollout (solid: DNS; dashed: NN). Panel (c) carries the claim; (b) is illustrative.

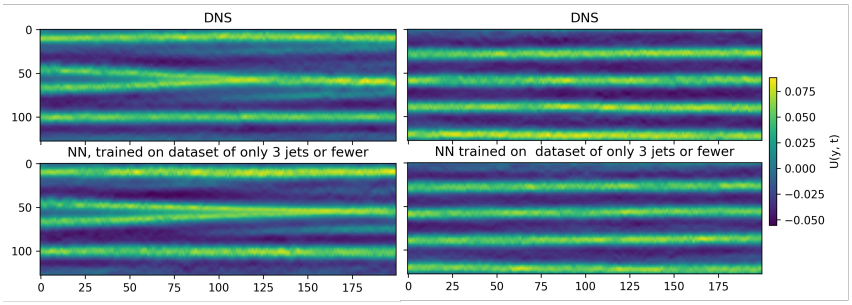

Figure 7: Latitude–time $\bar{u}_1(y,t)$ for an emulator trained only on regimes with three jets or fewer. Despite this restriction, it recovers both (a,c) translating and (b,d) stationary four-jet states, with coalescence and nucleation events absent from training.

4.3 Beta-plane turbulence

As a second supporting system we move from a deterministic 1D PDE to partially observed 2D geophysical turbulence. At fixed (β, R_d) the two-layer flow is bistable between translating jets that undergo coalescence and nucleation and a stationary four-jet regime (fig. 11). Building on the single-layer conditioned emulator of Shokar et al. (2025b), here we treat the two-layer problem under partial observation and impose an excluded-regime restriction: training contains only translating states with at most three jets and no transition events. The question is whether unseen jet counts and transition events can still appear when the network observes only the upper layer ψ_1 and must close the baroclinic coupling implicitly.

Figure 6 shows a single forecast draw in which coalescence occurs near day 100 and nucleation near day 200 before the trajectory diverges from DNS, as any chaotic rollout must. One draw cannot establish that the emulator has the right transition behaviour, and we do not rest the claim on it: the load-bearing evidence is the jet-count transition probabilities of fig. 6c, which are statistics accumulated over long rollouts and which align with DNS despite no transition appearing in training. Figure 7 adds the second excluded feature: the same training restriction still yields the stationary four-jet state, and four-to-three coalescence followed by nucleation—a jet count and events that never appear in training. These rollouts are initialised from a four-jet condition, so the claim is that the emulator *sustains* a regime absent from its training measure and produces the correct transitions out of it, not that it discovers that regime spontaneously from a three-jet start. Relative to the KS tests, this is a stricter probe of excluded-regime generalisation: the emulator is probabilistic, the observation is incomplete, and the target includes a qualitatively different bistable state rather than only transitions within one chaotic attractor. Full equations, DNS parameters, and the latent-sampling architecture are given in section C.

5 CONCLUSION

Our main findings concern 2D Kolmogorov flow, where an emulator trained solely on chaotic aspect ratios can predict laminar dynamics beyond an α-bifurcation. Whether those laminar rollouts are physically admissible depends on the architecture's implicit symmetry group: a standard convolution drifts continuously in y along an invariance the forced PDE does not possess, while the $\mathscr{S}^2$-aware convolution, restricted to the discrete shifts $\mathscr{S}^2$ permits, does not. Force injection removes the drift but leaves the field unsteady. The same experiments delimit the claim: crossing the chaotic-to-laminar bifurcation does not guarantee exact attractors—predicted states can lie on a nearby spurious manifold, rollouts are sensitive to Δt, and chaotic transients from a perturbed base state are not recovered.

Two further systems indicate that excluded-regime generalisation is not unique to Kolmogorov flow: a Kuramoto–Sivashinsky emulator recovers relaminarisation, warm-up, and kink transitions absent from training and extrapolates $2.4\times$ beyond its largest fine-tuning domain, and a partially observed beta-plane emulator recovers jet coalescence, nucleation, and stationary four-jet states from translating three-jet training data alone.

We conjecture that these successes reflect the target dynamics being represented locally in the training measure—on neighbourhoods the size of the architecture's effective receptive field rather than as whole states—not the memorisation of trajectories Scher & Messori (2019). This is what separates our failures from our successes: KS warm-up is organised by the same local kink-forming instability already sampled on the chaotic attractor and is recovered, whereas the Kolmogorov chaotic transient explores a set weakly correlated with the post-transient training measure and is not; and training KS only on laminar $L = 12$ fails to produce chaos at $L = 22$, whereas chaotic $L = 128$ transfers (fig. 10). The KS numbers make this concrete: the emulator's effective receptive field is fixed at 26 length units, which spans the whole $L = 22$ pretraining domain but only 6.5% of the $L = 400$ domain it extrapolates to (section A), so what transfers is a stencil-scale rule rather than a memorised global state. The beta-plane emulator probes a different axis: its effective receptive field exceeds its domain (section C), so it sees the jet configuration globally, and what it adds is that excluded-regime generalisation survives partial observation.

What these successes are *not* is a converged discrete evolution map. That reading is in tension with our own Δt result: the emulator laminarises at every α when trained at $\Delta t = 1.0$ and fails to hold the laminar solution at $\Delta t = 0.2$ (section G), and a converged approximation to $\mathcal{F}_{\Delta t}$ would not reorganise its attractors with the sampling interval. Together with the spurious nearby manifold, this suggests that what local coverage governs is whether the emulator reaches the right region of state space, not whether it has learned the evolution rule there.

Two limits bound what we have established. First, local representation is read from the outcome rather than measured on the states: only the KS restricted-training experiments manipulate the training measure directly, and making coverage a predictor rather than a description requires a quantity computed on the data, which we have not done. Second, the symmetry comparison is qualitative and not parameter-matched, so it identifies the mechanism of the drift rather than establishing that this layer is the only remedy. Extending these findings to noisier, higher-dimensional flows, and across Reynolds number as well as aspect ratio, is an important next step.

REPRODUCIBILITY STATEMENT

All three systems use open-source solvers with fully specified parameters: KS with `FourierFlows` Constantinou et al. (2023) (section A), two-layer beta-plane turbulence with `pyqg` Abernathey et al. (2022) (section C), and Kolmogorov flow in section D and section H; those sections give domain sizes, resolutions, timesteps, forcing and dissipation parameters, and the training restriction defining each excluded regime. Architectures and optimisation settings are in section A and section C, the Kolmogorov objective in section F, the timestep diagnostics in section G, the MLP and FNO controls in section B, and the proof of Proposition 1 in section E. Code will be released on publication, withheld here only for anonymity.

REFERENCES

Ryan Abernathey, Cesar B Rocha, Andrew Ross, Malte Jansen, Ziwei Li, et al. pyqg/pyqg: v0.7.2. Zenodo, https://doi.org/10.5281/zenodo.6563667, 2022.

Giancarlo Benettin, Luigi Galgani, Antonio Giorgilli, and Jean-Marie Strelcyn. Lyapunov characteristic exponents for smooth dynamical systems and for hamiltonian systems; a method for computing all of them. part 1: Theory. *Meccanica*, 15(1):9–20, 1980.

Steven L Brunton, Bernd R Noack, and Petros Koumoutsakos. Machine learning for fluid mechanics. *Annual review of fluid mechanics*, 52(1):477–508, 2020.

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

Pi-Yueh Chuang and Lorena A Barba. Predictive limitations of physics-informed neural networks in vortex shedding. *arXiv preprint arXiv:2306.00230*, 2023.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pp. 2990–2999. PMLR, 2016.

Navid C Constantinou, Gregory L Wagner, André Palóczy, Keaton W Ho, Josef Bisits, Morten Piibeleht, Tim Besard, Cameron Robertson, and Vladimir Parfenyev. Fourierflows/fourierflows.jl: v0.10.5. Zenodo, https://doi.org/10.5281/zenodo.10350324, 2023.

Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6185–6194, 2023.

Yi He, Yiming Yang, Xiaoyuan Cheng, Hai Wang, Xiao Xue, Boli Chen, and Yukun Hu. Chaos meets attention: Transformers for large-scale dynamical prediction. *arXiv preprint arXiv:2504.20858*, 2025.

Jacob Helwig, Xuan Zhang, Cong Fu, Jerry Kurtin, Stephan Wojtowytsch, and Shuiwang Ji. Group equivariant fourier neural operators for partial differential equations. *arXiv preprint arXiv:2306.05697*, 2023.

Eberhard Hopf. A mathematical example displaying features of turbulence. *Communications on Pure and Applied Mathematics*, 1(4):303–322, 1948.

Herbert Jaeger. The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German national research center for information technology gmd technical report*, 148(34):13, 2001.

Simon Kneer, Taraneh Sayadi, Denis Sipp, Peter Schmid, and Georgios Rigas. Symmetry-aware autoencoders: s-pca and s-nlpca. *arXiv preprint arXiv:2111.02893*, 2021.

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

Zongyi Li, Miguel Liu-Schiaffini, Nikola Kovachki, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Learning dissipative dynamics in chaotic systems. *arXiv preprint arXiv:2106.06898*, 2021.

Alec J Linot and Michael D Graham. Deep learning to discover and predict dynamics on an inertial manifold. *Physical Review E*, 101(6):062209, 2020.

Alec J Linot, Joshua W Burby, Qi Tang, Prasanna Balaprakash, Michael D Graham, and Romit Maulik. Stabilized neural ordinary differential equations for long-time forecasting of dynamical systems. *Journal of Computational Physics*, 474:111838, 2023.

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.

Dan Lucas and Rich Kerswell. Spatiotemporal dynamics in two-dimensional kolmogorov flow over large domains. *Journal of fluid mechanics*, 750:518–554, 2014.

Dan Lucas and Rich R Kerswell. Recurrent flow analysis in spatiotemporally chaotic 2-dimensional kolmogorov flow. *Physics of Fluids*, 27(4), 2015.

Dongyu Luo, Jianyu Wu, Jing Wang, Hairun Xie, Xiangyu Yue, and Shixiang Tang. Diffffluid: Plain diffusion models are effective predictors of flow dynamics. *arXiv preprint arXiv:2409.13665*, 2024.

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

Jiquan Ngiam, Zhenghao Chen, Daniel Chia, Pang Koh, Quoc Le, and Andrew Ng. Tiled convolutional neural networks. *Advances in neural information processing systems*, 23, 2010.

Binh Duong Nguyen and Stefan Sandfeld. Out-of-distribution generalization of deep-learning surrogates for 2d pde-generated dynamics in the small-data regime. *arXiv preprint arXiv:2601.08404*, 2026.

Elise Özalp and Luca Magri. Stability analysis of chaotic systems in latent spaces. *Nonlinear Dynamics*, 113(11):13791–13806, 2025.

Jacob Page, Michael P Brenner, and Rich R Kerswell. Revealing the state space of turbulence using machine learning. *Physical Review Fluids*, 6(3):034402, 2021.

Jeremy P Parker and Tobias M Schneider. From synthetic turbulence to true solutions: A deep diffusion model for discovering periodic orbits in the navier-stokes equations. *arXiv preprint arXiv:2602.23181*, 2026.

Jaideep Pathak, Brian Hunt, Michelle Girvan, Zhixin Lu, and Edward Ott. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Physical review letters*, 120(2):024102, 2018.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Carlos E Pérez De Jesús, Alec J Linot, and Michael D Graham. Building symmetries into data-driven manifold dynamics models for complex flows. *Physical Review Fluids*, 10(6):064401, 2025.

Anton Pershin, Cédric Beaume, Kuan Li, and Steven M Tobias. Training a neural network to predict dynamics it has never seen. *Physical Review E*, 107(1):014304, 2023.

Alberto Racca and Luca Magri. Data-driven prediction and control of extreme events in a chaotic flow. *arXiv preprint arXiv:2204.11682*, 2022.

David W Romero and Suhas Lohit. Learning partial equivariances from data. *Advances in Neural Information Processing Systems*, 35:36466–36478, 2022.

Sebastian Scher and Gabriele Messori. Generalization properties of feed-forward neural networks trained on lorenz systems. *Nonlinear processes in geophysics*, 26(4):381–399, 2019.

Ira J. S. Shokar, Peter H. Haynes, and Rich R. Kerswell. Deep learning of the evolution operator enables forecasting of out-of-training dynamics in chaotic systems. *arXiv preprint arXiv:2502.20603*, 2025a.

Ira J. S. Shokar, Rich R. Kerswell, and Peter H. Haynes. Conditioning on PDE parameters to generalise deep learning emulation of stochastic and chaotic dynamics. *arXiv preprint arXiv:2509.09599*, 2025b.

Ira JS Shokar, Rich R Kerswell, and Peter H Haynes. Stochastic latent transformer: Efficient modeling of stochastically forced zonal jets. *Journal of Advances in Modeling Earth Systems*, 16(6): e2023MS004177, 2024.

Y Qiang Sun, Pedram Hassanzadeh, Mohsen Zand, Ashesh Chattopadhyay, Jonathan Weare, and Dorian S Abbot. Can ai weather models predict out-of-distribution gray swan tropical cyclones? *Proceedings of the National Academy of Sciences*, 122(21):e2420914122, 2025.

Dima Tretiak, Anastasia Bizyaeva, J Nathan Kutz, and Steven L Brunton. Physics-enforced reservoir computing for forecasting spatiotemporal systems. *Proceedings of Machine Learning Research vol*, 283:1–15, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Hao Wu, Yuan Gao, Fan Xu, Fan Zhang, Qingsong Wen, Kun Wang, Xiaomeng Huang, and Xian Wu. Differential-integral neural operator for long-term turbulence forecasting. *arXiv preprint arXiv:2509.21196*, 2025.

Chengzuo Zhuge, Jiawei Li, and Wei Chen. Deep learning for predicting the occurrence of tipping points. *Royal Society Open Science*, 12(7), 2025.

A KURAMOTO–SIVASHINSKY SETUP

The KS equation on the periodic interval $[0, L)$ is

$$\partial_t u = -u\partial_x u - \partial_{xx} u - \partial_{xxxx} u.$$

DNS uses `FourierFlows` Constantinou et al. (2023) with a pseudo-spectral method and a fourth-order exponential time-differencing Runge–Kutta scheme ($\delta t = 2.5 \times 10^{-2}$), subsampled to $\Delta t = 1$ for training. Trajectories are initialised with $u \sim \mathcal{N}(0, 0.01)$ and only post-transient snapshots are retained.

The emulator is a neighbourhood-attention transformer conditioned on L via FiLM modulation Hassani et al. (2023); Perez et al. (2018). It is pretrained at $L = 22$ (grid of 56 points, 5,000 snapshots) and fine-tuned on $L \in \{48, 64, 96, 128, 164\}$ with grids of $\{120, 160, 240, 320, 410\}$ points respectively (500 snapshots each). All domains use a fixed grid spacing $\Delta x \approx 0.4$, so the test domains are resolved on 140 points at $L = 56$ and $1{,}000$ points at $L = 400$. Training minimises

$$\mathrm{MSE} = \frac{1}{L} \int_0^L \left| u_{t+\Delta t} - \mathcal{F}_{\Delta t}(u_t) \right|^2 \mathrm{d}x,$$

with autoregressive rollouts $\tilde{u}_{t+\Delta t} = \mathcal{F}_{\Delta t}(\tilde{u}_t)$ and evaluation by MAE/RMSE.

Architecture and optimisation. The network uses $C = 64$ latent channels, temporal history $S = 2$, $N = 8$ transformer blocks, one attention head, and a local-attention window $K = 9$ with circular padding and learned relative positional encodings. Circular padding preserves translation equivariance on the periodic domain. Stacking N blocks of window K gives an effective receptive field of $N(K-1)+1 = 65$ grid points, equivalently 26 length units: the whole of the $L = 22$ pretraining domain, 46% of $L = 56$, 16% of the largest fine-tuning domain $L = 164$, and 6.5% of $L = 400$. Optimisation uses Adam with batch size 128, learning rate 5×10^{-4} for $1{,}000$ pretraining epochs and 10^{-5} for 500 fine-tuning epochs.

Restricted-training experiments. The two restricted datasets of the left panel of fig. 5 are trained from scratch at fixed $L = 56$, with no pretraining or fine-tuning, so that the only variable is the training measure. Restriction (i), retaining states with at most four kinks, uses $5{,}000$ snapshots; restriction (ii), excluding every kink-count transition and the 50 snapshots preceding each, uses 100. Kink-count return periods are accumulated over a $10{,}000$-step autoregressive rollout for both DNS and emulator.

Largest Lyapunov exponent. Time axes for KS and for Kolmogorov flow are both reported in Lyapunov units $\Lambda_{\max} t$, so that rollout lengths are comparable across systems. In each case $\Lambda_{\max}$ is estimated following Benettin et al. (1980) from the separation between two initially nearby trajectories in the prognostic variable (u for KS, ω for Kolmogorov flow); see eq. (16). Values for the Kolmogorov training trajectories are reported in section G.

B KS ARCHITECTURE BASELINES AND CONTROLS

The first two controls hold the training data fixed and vary the architecture; the third does the reverse. A single-hidden-layer MLP reproduces the $L = 56$ restricted-data results when trained at fixed L (fig. 8), but its input layer pins the grid size, so it cannot be evaluated at another L at all. An FNO reproduces some of the same fixed-L phenomena, yet fails to extrapolate jointly across small and large L: its trainable mode budget is set by the smallest training domain, so the additional active modes of a larger domain are not representable with any amount of data (fig. 9). The attention emulator instead learns a translation-equivariant local stencil that transfers across domain sizes without this bottleneck.

The complementary control varies the training data and holds the architecture fixed: training the attention emulator only on laminar $L = 12$ and evaluating at chaotic $L = 22$ fails to produce time-dependent dynamics (fig. 10). Exposure to dynamically varying states, not merely to the correct PDE parameter, is required for transfer. The two kinds of failure are worth distinguishing, because they look different: the architectural limits above produce a field the model cannot resolve, whereas restricted training data here produces a steady state where chaos is required. Both are discussed in section 5.

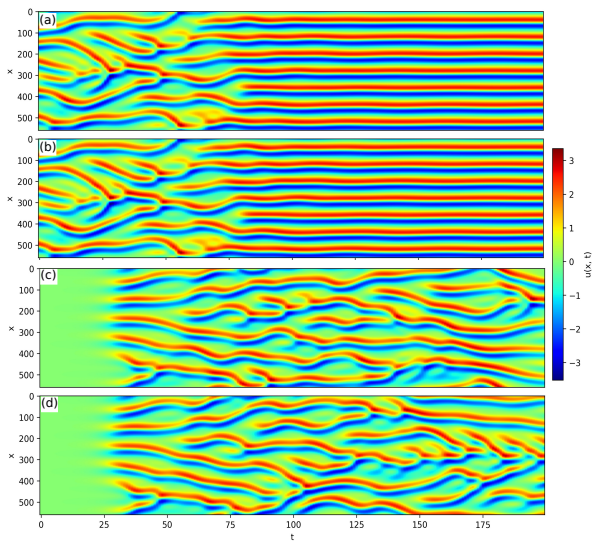

Figure 8: Relaminarisation and warm-up at $L = 56$ reproduced by (a,c) a single-hidden-layer MLP and (b,d) an FNO, each trained only on chaotic $L = 56$ data.

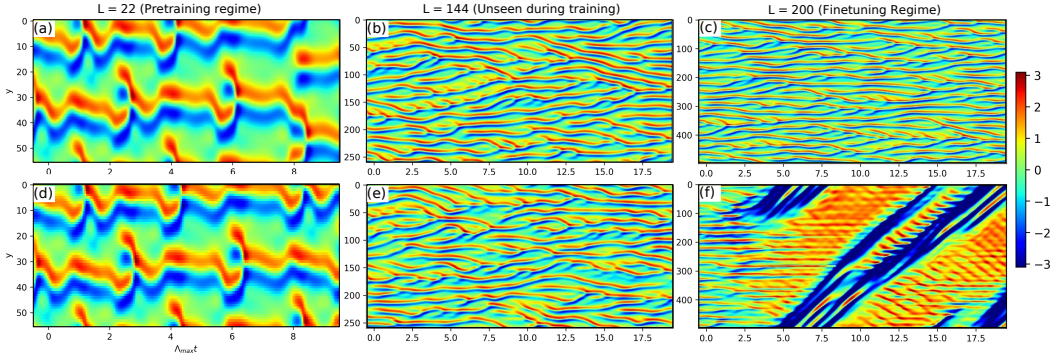

Figure 9: Generalisation across domain length after pretraining at $L = 22$ and fine-tuning on $L \in \{22, 36, 48, 64, 96, 128, 200\}$. (a–c) neighbourhood-attention emulator; (d–f) FNO. The FNO fails to produce a plausible forecast at large L. Note that this fine-tuning list differs from the protocol used elsewhere in the paper ($L \in \{48, 64, 96, 128, 164\}$, section A): the two architectures here are trained on identical data as each other, so the comparison isolates architecture, but it is not a like-for-like extension of the $L = 400$ result in the right panel of fig. 5.

C BETA-PLANE SETUP

The two-layer quasi-geostrophic system is

$$\partial_t \zeta_m + J(\psi_m, \zeta_m) + \beta \partial_x \psi_m + U_m \partial_x \zeta_m = \delta_{m,2}\, r_{ek} \nabla^2 \psi_2 + \mathrm{ssd}, \qquad m \in \{1, 2\}, \tag{2}$$

$$\zeta_m = \nabla^2 \psi_m + (-1)^m k_d^2 (\psi_2 - \psi_1), \qquad k_d = R_d^{-1}, \tag{3}$$

where U_m is the imposed constant background zonal flow in layer m, r_{ek} is the bottom drag coefficient (denoted α in some presentations; we rename it to avoid collision with the Kolmogorov aspect ratio) and ssd is an exponential filter. The diagnosed quantity we forecast is the zonal-mean zonal velocity $\bar{u}_m(y, t) = -\langle \partial_y \psi_m \rangle_x$, which is what the latitude–time plots show and what the loss is computed on; it is distinct from the constant U_m. DNS uses `pyqg` Abernathey et al. (2022) with a pseudo-spectral method and a third-order Adams–Bashforth scheme at $\delta t = 1\,\mathrm{h}$, on a 128×128 grid, downsampled to $\Delta t = 1\,\mathrm{day}$ for the partially observed experiments. Trajectories are initialised with small-amplitude noise in ζ drawn uniformly from $[-10^{-6}, 10^{-6}]$ and only post-transient snapshots are retained.

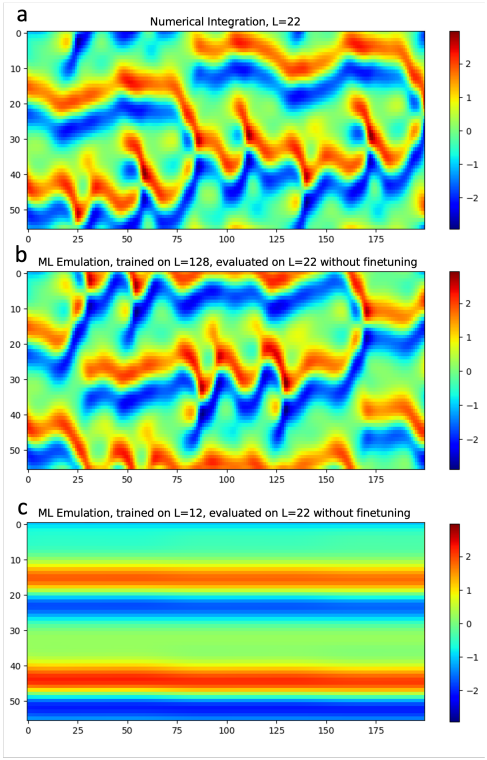

Figure 10: Zero-shot evaluation at $L = 22$ for emulators trained on a single L. (a) DNS; (b) pretrained on chaotic $L = 128$ (qualitatively correct); (c) pretrained on laminar $L = 12$ (fails).

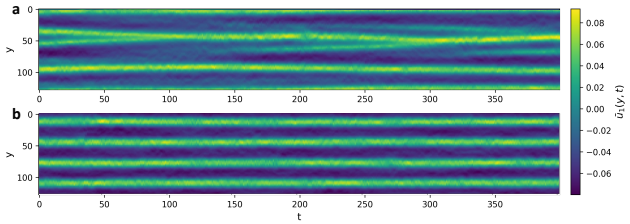

Figure 11: Bistability of two-layer beta-plane turbulence at fixed (β, R_d): latitude–time plots of $\bar{u}_1(y,t)$ for (a) translating dynamics with coalescence/nucleation and (b) a stationary four-jet state.

Parameters are held fixed at $\beta = 5 \times 10^{-11}\,\mathrm{m^{-1}s^{-1}}$, bottom drag $r_{ek} = 4 \times 10^{-8}\,\mathrm{s^{-1}}$, deformation radius $R_d = 12.5\,\mathrm{km}$, background zonal flow $U_1 = 0.025\,\mathrm{m\,s^{-1}}$ and $U_2 = 0$, and depths $H_1 = 500\,\mathrm{m}$, $H_2 = 1500\,\mathrm{m}$. The small-scale dissipation is the exponential filter

$$E_f = \exp\left[C_{\mathrm{ssd}}(\kappa^\star - \kappa_c)^4\right] \text{ for } \kappa^\star \geq \kappa_c, \qquad E_f = 1 \text{ otherwise},$$

where $\kappa^\star = \sqrt{(k\Delta x)^2 + (l\Delta y)^2}$ is the non-dimensional wavenumber, κ_c is 65% of the Nyquist scale $\kappa^\star_{\mathrm{ny}} = \pi$, and $C_{\mathrm{ssd}} = \log(10^{-15})/(0.35\pi)^4 \approx -23.5$ places the energy at $\kappa^\star = \pi$ below machine precision.

At fixed (β, R_d) the flow is bistable between translating and stationary four-jet regimes (fig. 11). Following the parameter-conditioned probabilistic approach of Shokar et al. (2025b) (developed there for a single-layer model), the two-layer emulator receives $(\psi_1(t-1), \psi_1(t))$ and parameters (β, R_d), encodes them with local attention and adaptive layer normalisation, and samples a latent Gaussian before the transformer blocks (fig. 12). Training minimises

$$\mathcal{L} = \mathrm{CRPS}\big(\bar{u}_{1,t+1}, \tilde{\bar{u}}_{1,t+1}\big) + \lambda\,\mathrm{MAE}\big(|\mathcal{F}[\bar{u}_{1,t}]|, |\mathcal{F}[\tilde{\bar{u}}_{1,t}]|\big),$$

with ensemble size $m = 2$ and $\lambda = 1$, where $\mathcal{F}$ denotes the Fourier transform in y and tildes denote predictions. CRPS is evaluated with the fair (unbiased) estimator throughout, both in training and in

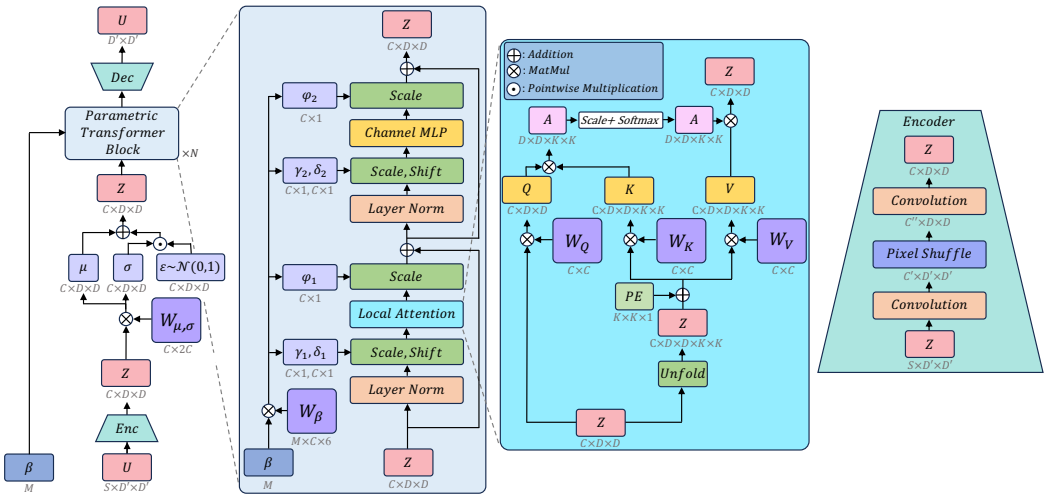

Figure 12: Beta-plane emulator: neighbourhood-attention transformer conditioned on (β, R_d), with latent sampling for probabilistic forecasts from upper-layer streamfunction history.

reporting,

$$\mathrm{CRPS} = \frac{1}{m}\sum_i |\tilde{x}_i - x| - \frac{1}{2m(m-1)}\sum_i\sum_j |\tilde{x}_i - \tilde{x}_j|, \tag{4}$$

in which the spread term carries the $1/2m(m-1)$ normalisation rather than the $1/2m^2$ of the naive estimator. The distinction matters at $m = 2$, where the naive form rewards ensemble spread by half as much as it should and, used as a training objective, would drive the emulator to be systematically under-dispersed. Because ψ_2 is never observed, the network must learn an implicit closure for the baroclinic interaction $G(\psi_1, \psi_2; \beta, R_d)$. The main-text experiments evaluate this two-layer partially observed model under the excluded-regime jet restriction.

Architecture and optimisation. The network uses $C = 64$ latent channels, temporal history $S = 2$, $N = 16$ transformer blocks, one attention head, and a local-attention window $K = 17$. Optimisation uses Adam with batch size 128, learning rate 5×10^{-4} for 500 pretraining epochs and 10^{-5} for 250 fine-tuning epochs, with CRPS training ensemble size $m = 2$. The rollouts plotted in fig. 6 and fig. 7 span 426 days, i.e. 426 autoregressive steps at $\Delta t = 1\,\mathrm{day}$.

Unlike the KS emulator, this network is not local at the scale of its target. The effective receptive field, $N(K-1)+1 = 257$ grid points, exceeds the 128-point domain, so the emulator becomes globally connected in y roughly halfway through its blocks and sees the whole jet configuration at once. The beta-plane experiment is therefore a test of excluded-regime generalisation under partial observation, and should not be read as evidence for the locality argument of section 5, which rests on the KS results.

D KOLMOGOROV EQUATIONS AND DIAGNOSTICS

We explore the incompressible 2D Navier–Stokes equations with forcing directed along x and varying sinusoidally in y:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} + \nabla p = \frac{1}{\mathrm{Re}}\nabla^2\mathbf{u} + \sin(\kappa y)\hat{\mathbf{x}}, \tag{5}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{6}$$

where $\mathbf{u} = (u, v)$, κ is the forcing wavenumber, Re is the Reynolds number, and p is the pressure. We take $\mathrm{Re} = 50$ and $\kappa = 4$ throughout, on the doubly-periodic domain $(x, y) \in [0, 2\pi/\alpha) \times [0, 2\pi)$ with $0 < \alpha \leq 1$. Reformulating in vorticity $\omega = \nabla \times \mathbf{u}$ and streamfunction ψ with $\mathbf{u} = \nabla^\perp\psi$ gives

$$\frac{\partial \omega}{\partial t} + \mathbf{u}\cdot\nabla\omega = \frac{1}{\mathrm{Re}}\nabla^2\omega - \kappa\,\cos(\kappa y), \tag{7}$$

$$\omega = -\nabla^2\psi. \tag{8}$$

The system symmetries are Chandler & Kerswell (2013)

$$\mathscr{S} : [u, v, \omega](x, y) \longrightarrow [-u, v, -\omega]\left(-x, y + \frac{\pi}{\kappa}\right), \tag{9}$$

$$\mathscr{R} : [u, v, \omega](x, y) \longrightarrow [-u, -v, \omega]\,(-x, -y)\,, \tag{10}$$

$$\mathscr{T}_l : [u, v, \omega](x, y) \longrightarrow [u, v, \omega](x + l, y), \qquad 0 \le l < \frac{2\pi}{\alpha}. \tag{11}$$

Volume-averaged energy, dissipation, and input energy are Lucas & Kerswell (2014)

$$\bar{E} \coloneqq \frac{E(t) - E_{\text{lam}}}{E_{\text{lam}}}, \qquad \bar{D} \coloneqq \frac{D(t) - D_{\text{lam}}}{D_{\text{lam}}}, \qquad \bar{I} \coloneqq \frac{I(t) - D_{\text{lam}}}{D_{\text{lam}}}, \tag{12}$$

with

$$E(t) \coloneqq \frac{1}{2}\big\langle \mathbf{u}^2 \big\rangle, \qquad D(t) \coloneqq \frac{1}{\text{Re}}\Big\langle |\nabla \mathbf{u}|^2 \Big\rangle, \qquad I(t) \coloneqq \langle u \sin(\kappa y) \rangle, \tag{13}$$

$$\langle\rangle \coloneqq \frac{\alpha}{4\pi^2}\int_0^{2\pi}\int_0^{2\pi/\alpha} \mathrm{d}x\mathrm{d}y, \tag{14}$$

$$\mathbf{u}_{\text{lam}} \coloneqq \frac{\text{Re}}{\kappa^2}\sin(\kappa y)\hat{\mathbf{x}}, \qquad E_{\text{lam}} \coloneqq \frac{\text{Re}^2}{4\kappa^4}, \qquad D_{\text{lam}} \coloneqq \frac{\text{Re}}{2\kappa^2}. \tag{15}$$

The largest Lyapunov exponent is estimated following Benettin et al. (1980):

$$\Lambda_{\text{max}} = \lim_{n\to\infty}\frac{1}{n}\ln\frac{\left\|\delta\omega^{(n)}\right\|}{\left\|\delta\omega^{(0)}\right\|}. \tag{16}$$

E Proof of Proposition 1

Proof. Define $(\tau_m \mathbf{h})(r, c) = \mathbf{h}(r - m, c)$ with indices modulo N_y. Then

$$\begin{aligned}
\text{Conv}(\tau_m \mathbf{h})(r, c) &= \sum_{a,b} \mathbf{W}_{\phi_y(r)}(a, b)\,(\tau_m \mathbf{h})(r + a, c + b) \\
&= \sum_{a,b} \mathbf{W}_{\phi_y(r)}(a, b)\,\mathbf{h}(r + a - m, c + b), \\
\tau_m(\text{Conv}(\mathbf{h}))(r, c) &= (\text{Conv}(\mathbf{h}))(r - m, c) \\
&= \sum_{a,b} \mathbf{W}_{\phi_y(r-m)}(a, b)\,\mathbf{h}(r - m + a, c + b).
\end{aligned}$$

The two expressions agree for all r if and only if $\mathbf{W}_{\phi_y(r)} = \mathbf{W}_{\phi_y(r-m)}$. Because $r \mapsto \mathbf{W}_{\phi_y(r)}$ has period N_y/κ, this holds precisely when $m = qN_y/\kappa$ for some $q \in \mathbb{Z}$, and fails otherwise. Hence equivariance holds exactly on $\mathcal{G}_{\mathscr{S}^2}$. □

F Kolmogorov loss function

The composite loss is

$$\mathcal{L} = \mathcal{L}_\omega + \mathcal{L}_\psi + \mathcal{L}_{\Delta\phi_x}, \tag{17}$$

$$\mathcal{L}_\omega = \frac{1}{N_t N_{xy}}\sum_t \sum_{\kappa_x,\kappa_y} \frac{\left(1 + \kappa_x^2 + \kappa_y^2\right)^2}{\max\limits_{\kappa_x',\kappa_y'}\left(1 + {\kappa_x'}^2 + {\kappa_y'}^2\right)^2}\ \left|\text{FFT}_{x,y}\big(\omega(x, y, t) - \omega^{\text{ref}}(x, y, t)\big)\right|^2, \tag{18}$$

$$\mathcal{L}_\psi = \frac{1}{N_t N_{xy}}\sum_t \sum_{x,y} \left|-\nabla^{-2}\omega(x, y, t) + \nabla^{-2}\omega^{\text{ref}}(x, y, t)\right|^2, \tag{19}$$

$$\mathcal{L}_{\Delta\phi_x} = \frac{1}{N_t}\sum_t \left|\Delta\phi_x(t) - \Delta\phi_x^{\text{ref}}(t)\right|^2. \tag{20}$$

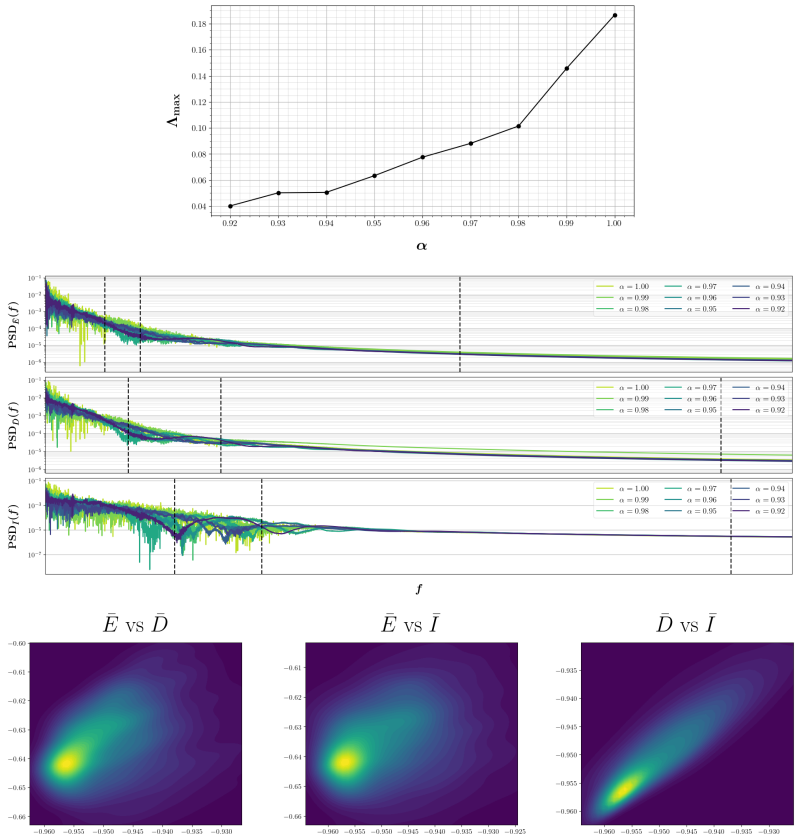

Figure 13: Kolmogorov training diagnostics. Top: largest Lyapunov exponents of the training trajectories. Middle: power spectral density of volume-averaged energy, dissipation, and input energy (transients inclusive); vertical lines mark 90%/95%/99% spectral cut-offs. Bottom: joint probability density of the training trajectories in phase space.

Only $\mathcal{L}_\omega$ is Sobolev-weighted: it is an order-2 Sobolev loss in spectral form, which following Li et al. (2021) emphasises high-frequency structure more than MSE. The streamfunction term $\mathcal{L}_\psi$ is a plain L^2 loss (equivalently order 0) and regularises the Poisson relation eq. (8). The third term $\mathcal{L}_{\Delta\phi_x}$ penalises error in the increment of the streamwise phase ϕ_x, so it constrains drift along x—the direction in which continuous translation $\mathscr{T}_l$ is a genuine symmetry of the PDE, and along which a rollout would otherwise be free to slide relative to the DNS reference without being wrong. It does not act along y. The y-drift suppressed by the $\mathscr{S}^2$-aware convolution is therefore a separate matter from this loss term, and the two are not competing explanations for the same effect.

G KOLMOGOROV TRAINING DIAGNOSTICS

Training comprises input–output snapshots from nine DNS trajectories at $\alpha \in [0.92, 1.00]$, with $\mathrm{Re} = 50$ and $\kappa = 4$, excluding the transient phase (section H). Figure 13 confirms $\Lambda_{\max} > 0$ for all trajectories. Figure 13 shows the spectral cut-offs used to choose the emulator timestep via the Nyquist relation $\Delta t = 1/(2f)$, where f is the frequency below which a given fraction of the spectral power lies. The input-energy spectrum is the most stringent of the three. Because a higher retained fraction implies a higher cut-off frequency, the 90%/95%/99% cut-offs correspond to progressively *smaller* timesteps, roughly $\Delta t \approx 1.0$, 0.5 and 0.2 respectively; $\Delta t = 0.5$ resolves 95% of the input-energy variability. Figure 13 shows the training support in $(\bar{E}, \bar{D}, \bar{I})$ space.

The timestep also has an effect that the spectral criterion does not predict (assuming identical architectures and training budgets). Trained at $\Delta t = 1.0$ the emulator laminarises at every α tested, including those where the DNS remains chaotic, while at $\Delta t = 0.2$ it does not hold the laminar solution at the aspect ratios where DNS is laminar. A converged approximation to a fixed evolu-

tion map, sampled at different intervals, would not reorganise which states it settles on in this way. The sensitivity is thus not only a hyperparameter choice but evidence that the learned map is not a converged discrete evolution operator, a point taken up in section 5.

H KOLMOGOROV DATA GENERATION

We generate DNS vorticity trajectories by pseudospectrally solving the vorticity formulation of the incompressible Navier–Stokes equations:

$$\omega(x,y,t) = \sum_{\kappa_x} \sum_{\kappa_y} \hat{\omega}\left(\kappa_x, \kappa_y, t\right)\, \mathrm{e}^{\mathrm{i}(\kappa_x x + \kappa_y y)}, \tag{21}$$

$$(\kappa_x, \kappa_y) \in \left[-\alpha \left\lfloor \frac{N_x}{3} \right\rfloor, \ldots, \alpha \left\lfloor \frac{N_x}{3} \right\rfloor\right] \times \left[0, \ldots, \left\lfloor \frac{N_y}{3} \right\rfloor\right], \tag{22}$$

with Hermitian symmetry $\hat{\omega}(-\kappa_x, -\kappa_y) = \hat{\omega}(\kappa_x, \kappa_y)^*$, $N_y/N_x = \alpha$, and $(x, y) \in [0, 2\pi/\alpha) \times [0, 2\pi)$. Time integration uses a fourth-order Runge–Kutta scheme, and $N_y = 128$.

For each α, initial conditions are generated by perturbing the laminar base state,

$$\omega^{(0)} = \omega_{\text{lam}} + \omega_{\text{pert}}, \quad \omega_{\text{lam}} = -\frac{\text{Re}}{\kappa}\, \cos(\kappa y), \tag{23}$$

with the perturbation constructed in streamfunction space so that it is incompressible. A white-noise streamfunction ψ_{rand} is drawn from a unit Gaussian, Fourier transformed, and restricted to the de-aliased radial set

$$\mathcal{K} = \left\{ (\kappa_x, \kappa_y) \in \alpha\mathbb{Z} \times \mathbb{Z} : |\kappa_x| \le \alpha \left\lfloor \frac{N_x}{3} \right\rfloor,\ |\kappa_y| \le \left\lfloor \frac{N_y}{3} \right\rfloor,\ 0 < \sqrt{\kappa_x^2 + \kappa_y^2} \le \kappa_{\max} \right\}. \tag{24}$$

The streamfunction perturbation is then

$$\psi_{\text{pert}} = \sum_{(\kappa_x, \kappa_y) \in \mathcal{K}} \hat{\psi}_{\text{rand}} \left(\frac{\mathrm{e}^{\mathrm{i}(\kappa_x x + \kappa_y y)}}{\kappa_x^2 + \kappa_y^2 + 1} \right), \tag{25}$$

where the denominator emphasises large scales. Rescaling to a controlled fraction ϵ of the laminar velocity magnitude gives

$$\psi_{\text{pert}} \longleftarrow \epsilon \left(\frac{\left\langle |\mathbf{u}_{\text{lam}}|^2 \right\rangle^{\frac{1}{2}}}{\left\langle |\nabla^\perp \psi_{\text{pert}}|^2 \right\rangle^{\frac{1}{2}}} \right) \psi_{\text{pert}} = \frac{\epsilon\, \text{Re}}{\kappa^2 \sqrt{2}} \left\langle |\nabla^\perp \psi_{\text{pert}}|^2 \right\rangle^{-\frac{1}{2}} \psi_{\text{pert}}, \tag{26}$$

using eq. (15), and finally $\omega_{\text{pert}} = -\nabla^2 \psi_{\text{pert}}$ via eq. (8). Random search over $\kappa_{\max}$ and ϵ explores multiple basins of attraction at each α.