ADAPTIVE DOMINANT OPERATOR SELECTION AND AUXILIARY WEIGHTING FOR NEURAL PDE SOLVERS

Anonymous authors
Paper under double-blind review

ABSTRACT

A simplified PDE can supply auxiliary labels to a neural operator, but a fixed simplification need not follow the full dynamics equally well from every state. We study adaptive dominant operator selection and auxiliary weighting: an offline policy that advances each admissible Basic PDE from the same Full training anchor for three saved intervals and selects the candidate with the smallest channel-normalized mismatch against future Full *training* frames. The selected Basic form supplies a one-step auxiliary target; its mismatch sets a train-calibrated loss weight. A shared Fourier Neural Operator predicts through its Full head alone at test time. In single-seed, all-time, single-step comparisons on held-out initial conditions, adaptive versus fixed step-10 error relative to initial-state RMS is 0.06047 versus 0.05835 for Burgers, 0.09258 versus 0.10537 for FitzHugh–Nagumo, and 0.07505 versus 0.09749 for Navier–Stokes. Adaptive improves two of three systems under this protocol. A separate sparse-anchor protocol with three-step Full-head continuation gives different rankings and is reported in the appendix. Routing uses future Full training frames, and measured costs cover only candidate Basic solves; these results establish neither a state-only selector nor a matched total-cost advantage.

1 INTRODUCTION

Neural PDE solvers offer a promising route to amortizing the cost of repeated numerical simulation. Neural operators, including the Fourier Neural Operator and DeepONet, learn mappings between function spaces and can approximate solution operators across families of PDE instances (Li et al., 2021; Lu et al., 2021; Kovachki et al., 2023). However, reducing inference cost does not eliminate the cost of constructing the training set. Supervised learning still requires simulations that cover relevant parameters, initial conditions, and dynamical behaviors. Benchmark studies highlight the diversity of these learning problems, while work on autoregressive solvers shows that accurate short-horizon predictions need not translate into reliable long rollouts (Takamoto et al., 2022; Lippe et al., 2023). These challenges motivate a question beyond model architecture: what physical information should the training data contain?

Several research directions address this question by exploiting additional structure. Physics-informed neural operators combine simulation data with governing-equation constraints (Li et al., 2024), while Lie point symmetry augmentation generates additional solutions through transformations that preserve the underlying PDE (Brandstetter et al., 2022). Multiphysics pretraining and PDE foundation models reuse representations learned across heterogeneous systems (McCabe et al., 2024; Herde et al., 2024). Complementing these approaches, multifidelity operator learning combines abundant lower-fidelity observations with scarce high-fidelity data, including settings in which the fidelities correspond to different physical models (Howard et al., 2023). Together, these studies motivate exploiting cheaper sources of physical information, but leave open how to choose a simplified auxiliary PDE for a particular target regime.

Most directly, Ma et al. (2026) train neural operators jointly on full PDEs and their decomposed basic forms. Their method uses prespecified basic forms for each PDE family and a fixed auxiliary loss weight. Importantly, their ablations compare alternative forms and support the choices through discrepancies against full simulation trajectories. The remaining challenge is therefore not simply to recognize that basic-form selection matters. It is to make this selection responsive to changing

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.

training states using a short window of existing Full training trajectories, and to determine how strongly the selected auxiliary data should influence learning.

This challenge arises because physical dominance is conditional rather than intrinsic to an equation's name. The relative contributions of advection, diffusion, reaction, and other processes depend on both system parameters and the evolving state. Data-driven dominant-balance methods already identify regime-dependent subsets of active physical terms, including situations in which multiple terms jointly govern the dynamics (Callaham et al., 2021; Kaiser et al., 2022). These results motivate adapting auxiliary-data generation across regimes rather than treating a single basic form as universally representative. They also reveal a limitation of instantaneous operator rankings: a large term at one state need not generate the closest short-time trajectory. We therefore define the selected dominant operator by the smallest measured Basic–Full dynamical mismatch over a short training window, and use that mismatch to modulate simplified supervision. Figure 1 illustrates distinct evolutions from common initial fields.

Recent approaches address complementary aspects of physical reuse and reliability. CompNO and its extension to multidimensional fluid dynamics assemble pretrained fundamental operator blocks into PDE solvers (Hmida et al., 2026b;a), whereas Bayesian multifidelity Laplace neural operators use predictive uncertainty to guide the acquisition of high-fidelity trajectories (Kim et al., 2026). Our focus is different: we couple the choice of an auxiliary physical model with the weight assigned to its supervision. This leads to the central question of this work: which simplified physics should be simulated for a given regime, and how strongly should the resulting data influence learning?

We introduce Adaptive Dominant Operator Selection and Auxiliary Weighting to connect these two decisions. For each eligible anchor in a Full training trajectory, every admissible single-operator Basic form starts from the same current field and advances autonomously for three saved intervals. We compare each candidate with the corresponding future Full training frames using a channel-normalized mismatch $J_k(3)$ and select its minimum. This is offline selection using future *training* data, not a state-only deployment rule. The selected form supplies a one-step auxiliary target, while the Full PDE remains the prediction target.

The auxiliary weighting component adjusts the influence of the selected Basic target using its measured consistency discrepancy. A train-only calibration based on candidate-J bins and an initial-condition-cluster bootstrap upper endpoint sets the auxiliary loss weight. This calibration describes the observed short-window mismatch in the training data, not predictive uncertainty or a guarantee of downstream accuracy. The main all-time runs train with one-step Full and weighted one-step Basic losses. A separate sparse-anchor experiment adds three-step Full-head continuation while keeping Basic supervision one-step. Inference uses only the Full prediction head.

We study this framework on nonlinear Burgers, two-field FitzHugh–Nagumo reaction–diffusion, and projected incompressible Navier–Stokes systems. The all-time, single-step seed-0 comparisons use held-out initial conditions and ten-step Full-head evaluation. Adaptive routing improves both reported test errors for FHN and Navier–Stokes, while fixed routing performs better for Burgers (Table 1). Table 2 gives additional FHN and Navier–Stokes training and checkpoint details. Separate sparse-anchor experiments with three-step Full-head continuation and route–weight ablations appear in the appendix. Measured costs cover candidate Basic replay only; matched total simulation budgets, Full-only controls, and parameter out-of-distribution performance remain unestablished.

Our contributions are threefold:

1. **Training-anchor auxiliary-physics learning.** We formulate auxiliary training-data construction as a coupled problem of selecting a simplified Basic form at each eligible Full training anchor and determining its training influence.

2. **Adaptive dominant operator selection.** We select the admissible Basic form with the smallest three-step, same-anchor mismatch against future Full training frames.

3. **Auxiliary weighting and measured outcomes.** We calibrate the Basic loss weight from training candidate discrepancies and report both favorable and unfavorable ten-step prediction outcomes.

2 Related Work

Neural PDE solvers. Fourier and other neural operators learn maps between fields from simulated trajectories (Li et al., 2021; Lu et al., 2021; Kovachki et al., 2023). Physics-informed neural operators add governing-equation residuals (Li et al., 2024); symmetry-based augmentation transforms existing examples (Brandstetter et al., 2022). Unsupervised reconstruction pretraining and in-context examples offer another route to data-efficient operator learning (Chen et al., 2024). Multiple-physics pretraining and large-scale PDE pretraining learn reusable representations (McCabe et al., 2024; Herde et al., 2024; Hao et al., 2024); scaling and transfer across PDE parameters have also been studied (Subramanian et al., 2023). For parameterized systems, HyperFNO uses hypernetworks to condition an FNO on physical parameters (Alesiani et al., 2022); Jing et al. (2026) combine parameter modulation with spectral coupling for coupled fields. Multiwavelet operator learning (Gupta et al., 2021) has been extended to coupled systems (Xiao et al., 2023), while codomain-attention pretraining models interactions across variables (Rahman et al., 2024). We retain a per-PDE FNO backbone and study which simplified simulator should generate an auxiliary label for its Full-PDE task.

Simplified physics and multi-fidelity learning. Multi-fidelity operator learning combines distinct physical models or resolutions (Howard et al., 2023). Most directly, Ma et al. (2026) jointly train on Full equations and decomposed Basic forms. They compare candidate Basic simulations with the Full simulation from identical initial conditions using RMSE over the entire trajectory, retain a fixed Basic form for each PDE, and use a fixed auxiliary weight of 0.7 (also ablated in their appendix). Our form choice is recomputed at each training anchor from its three-step Basic–Full mismatch, and its auxiliary weight is train-calibrated from that mismatch. This requires advancing all admissible candidates and reading future Full *training* frames; that reference access and candidate-solve cost are part of the method. Uncertainty-guided acquisition of additional high-fidelity trajectories (Kim et al., 2026) addresses a different decision.

Dominant balance and finite-time model error. Term-magnitude analysis identifies changing physical balances (Callaham et al., 2021; Kaiser et al., 2022). A dominant balance can include compensating terms, so its largest instantaneous member need not minimize the error of a single-term forward model. The primary route minimizes measured three-step Basic–Full discrepancy. Its in-window minimum follows from the definition of an argmin; it does not guarantee a better learned predictor or lower ten-step error.

3 Method

3.1 Problem Setup and Admissible Basic Forms

We study finite-time prediction for PDEs composed of interacting physical operators. Let $q(\mathbf{x}, t)$ denote a state with C_q channels, μ its physical parameters, and $m \in \{0,1\}^3$ an operator mask ordered as advection, diffusion, and reaction. Given a Full training trajectory, we write $q_0 = q^F(t_n)$ and $q_h^F = q^F(t_{n+h})$, with saved offsets $\tau_h = t_{n+h} - t_n$. The Full PDE and its admissible single-operator Basic candidates are

$$\begin{aligned} \partial_t q^F &= \sum_{k\in\{A,D,R\}} m_k \mathcal{O}_k(q^F;\mu), \\ \partial_t q_k^B &= \mathcal{O}_k(q_k^B;\mu), \qquad q_k^B(0) = q_0, \qquad k \in \mathcal{K}_{\mathrm{adm}}, \end{aligned} \tag{1}$$

where $\mathcal{K}_{\mathrm{adm}} = \{k : m_k = 1\}$. Each Basic candidate retains the corresponding Full operator with unchanged coefficients, state variables, grid, and applicable boundary or constraint treatment. Numerical validity is checked separately at each anchor. The prediction task is to learn the finite-time Full evolution map,

$$f_\theta^F(q_0;\mu,m,\Delta t) \approx \Phi_{\Delta t}^F(q_0;\mu), \tag{2}$$

where $\Phi_{\Delta t}^F$ denotes the Full-PDE solution operator over Δt. Basic simulations provide auxiliary training targets; they do not redefine the prediction task or replace Full supervision.

Admissible operators. For periodic Burgers dynamics, $m = (1, 1, 0)$, with $\mathcal{O}_A(q) = -q\,\partial_x q$ and $\mathcal{O}_D(q) = D\partial_{xx} q$. For FitzHugh–Nagumo dynamics, $m = (0, 1, 1)$, with coupled fields $q = (u, v)$ and homogeneous Neumann boundaries:

$$\begin{aligned} \mathcal{O}_D(u, v) &= (D_u \Delta u, D_v \Delta v), \\ \mathcal{O}_R(u, v) &= \alpha(u - u^3 - \kappa - v,\ u - v). \end{aligned} \tag{3}$$

For periodic, unforced incompressible Navier–Stokes dynamics, $m = (1, 1, 0)$, with $\mathcal{O}_A(\mathbf{u}) = -\mathcal{P}[(\mathbf{u} \cdot \nabla)\mathbf{u}]$ and $\mathcal{O}_D(\mathbf{u}) = \nu\Delta\mathbf{u}$, where $\mathcal{P}$ is the discrete divergence-free projection. The projection remains part of the advection-only candidate: pressure is not treated as an independently removable operator.

Roles of Full and Basic data. Full training trajectories supply the supervised prediction targets and the reference dynamics used for offline routing. Basic trajectories start from the same training anchors and supply aligned auxiliary targets. Across candidates, physical parameters, grids, boundary conditions, and saved output times are held fixed. Routing and auxiliary-weight calibration use only the training split; no Full test futures are used to construct training pairs.

3.2 Adaptive Dominant-Operator Routing

For auxiliary supervision, an operator is useful insofar as its Basic PDE preserves the relevant Full dynamics. We therefore operationalize *dominance* through short-window dynamical consistency, rather than the instantaneous magnitude of an operator alone. At each training anchor q_0, every admissible Basic candidate is advanced autonomously to τ_1, τ_2, τ_3, without teacher forcing or resets to Full states. These are three saved output times, not necessarily three internal solver steps.

Short-window consistency. Let $\langle\cdot\rangle_x$ denote the spatial mean and set $\varepsilon = 10^{-7}$. We measure each candidate's discrepancy using channel-wise normalization by the anchor energy:

$$\begin{aligned} e_{k,h}(q_0) &= \frac{1}{C_q} \sum_{c=1}^{C_q} \frac{\left\langle \left(q^B_{k,c}(\tau_h; q_0) - q^F_{h,c}\right)^2 \right\rangle_x}{\left\langle q^2_{0,c} \right\rangle_x + \varepsilon}, \\ J_k(3; q_0) &= \frac{1}{3} \sum_{h=1}^{3} e_{k,h}(q_0). \end{aligned} \tag{4}$$

Each channel's denominator remains fixed at q_0 throughout the window. Thus, the score is neither normalized by future Full-frame energies nor computed as a single globally normalized field error. This distinction matters for the two-channel systems.

Validity-aware selection. Let $\mathcal{K}_{\mathrm{valid}}(q_0) \subseteq \mathcal{K}_{\mathrm{adm}}$ contain candidates that pass the numerical validity checks and yield finite autonomous trajectories and scores. Numerical failures are excluded, rather than treated as valid targets with large finite errors. In particular, the inviscid Burgers advection candidate is rejected beyond the generator's conservative preshock horizon or when its high-mode energy is unresolved. For a nonempty valid set, routing selects

$$k^*(q_0) = \arg \min_{k \in \mathcal{K}_{\mathrm{valid}}(q_0)} J_k(3; q_0), \qquad J_k = +\infty \text{ for invalid candidates.} \tag{5}$$

The route is assigned per training anchor, not per spatial cell or once per trajectory. When no candidate is valid, no Basic target is assigned and the Full training row is retained.

The fixed-route comparators retain diffusion for Burgers, reaction for FitzHugh–Nagumo, and projected advection for Navier–Stokes. Whenever the fixed candidate is valid, $J_{k^*} \leq J_{k_{\mathrm{fixed}}}$ follows directly from equation 5. This property concerns the offline selection criterion; it is not evidence of improved neural prediction. Importantly, routing reads three future Full *training* frames. It is therefore an *offline oracle for auxiliary-data construction*, not a state-only selector or an inference-time PDE-switching rule.

3.3 CONFIDENCE-AWARE AUXILIARY SUPERVISION

Selecting the best available Basic candidate does not establish that it approximates the Full dynamics sufficiently well. We therefore separate *which candidate to use* from *how strongly to use it*. For a valid selected candidate, the aligned targets are $y^B = q^B_{k^*}(\tau_1; q_0)$ and $y^F = q^F_1$. Only the first Basic output is used as an auxiliary target. The second and third Full frames contribute to routing and may also supervise Full-head continuation; they do not define additional Basic targets.

Train-only discrepancy calibration. For each PDE, calibration pools $J_k(3)$ from *all valid training candidates*, rather than only the selected minima. A cluster bootstrap with 1000 resamples uses independent initial-condition groups as the resampling units. Sparse quantile bins are merged, and a monotone envelope is applied to the upper endpoints of the bootstrap intervals for the bin-wise mean discrepancy. Let $U_{\mathrm{mono}}(j)$ interpolate this envelope at discrepancy j, and define the training-derived scale

$$\tau_J = \max\left\{\mathrm{median}(J_{\mathrm{train,candidates}}),\, 10^{-8}\right\}. \tag{6}$$

Here $J_{\mathrm{train,candidates}}$ denotes the pooled valid candidate scores for the corresponding PDE.

Confidence-to-weight mapping. For a finite, nonnegative candidate discrepancy j, define

$$c_{\mathrm{dyn}}(j) = \exp\left[-\frac{\max\{j, U_{\mathrm{mono}}(j)\}}{\tau_J}\right] \in (0, 1]. \tag{7}$$

This score expresses confidence in the dynamical consistency of the Basic target, not uncertainty in the Full prediction. To compare adaptive and fixed routes on the same valid support, let v_{common} equal one only when both their Basic targets are valid. The implemented dynamic and constant auxiliary weights are

$$\begin{aligned} w_{\mathrm{dyn}}(j) &= \begin{cases} \lambda_B c_{\mathrm{dyn}}(j), & v_{\mathrm{common}} = 1, \quad 0 \le j < \infty, \\ 0, & \text{otherwise}, \end{cases} \\ w_{\mathrm{const}} &= \lambda_B\, \mathbf{1}\{v_{\mathrm{common}} = 1\}, \qquad \lambda_B = 0.7. \end{aligned} \tag{8}$$

Adaptive and fixed routes use their own candidate discrepancies and the same train-only calibration; their dynamic weights therefore need not coincide. An adaptive target outside the common-valid support also receives zero auxiliary weight, even if it is individually valid.

Low confidence suppresses only the Basic contribution. The Full target and its unit loss coefficient are retained for every training row, including anchors without a usable Basic target. Calibration is frozen before neural training and is neither refitted per epoch nor queried using Full test futures. The bootstrap intervals describe uncertainty in *bin-wise mean discrepancies*, not individual predictive coverage. Accordingly, c_{dyn} is a discrepancy-based confidence score, not a calibrated probability that a Basic target is correct.

3.4 TRAINING AND INFERENCE PROTOCOL

Shared neural operator. For each PDE, we train a four-block Fourier Neural Operator (Li et al., 2021) with shared spectral weights and separate Full and Basic output heads. The Full pass is conditioned on the complete mask m, whereas the Basic pass uses the one-hot mask e_k of its retained operator. Both passes receive the current field, spatial coordinates, transformed physical parameters including the saved time increment, and the current-state mean and standard deviation. We denote the conditioned outputs by f^F_θ and f^B_θ, suppressing these conditioning variables below.

One-step training. The per-example prediction loss is the channel-mean normalized spatial MSE,

$$\ell(\hat{y}, y) = \frac{1}{C_q} \sum_{c=1}^{C_q} \frac{\left\langle (\hat{y}_c - y_c)^2 \right\rangle_x}{\left\langle y_c^2 \right\rangle_x + \varepsilon}. \tag{9}$$

No square root is applied. Unlike the routing score in equation 4, this loss is normalized by each target channel's energy rather than by the anchor energy. Let N be the number of Full training rows

and $\mathcal{I}_B$ the subset with usable Basic targets on the common-valid support. The one-step objective is

$$\mathcal{L}_1(\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell\big(f_\theta^F(q_{0,i}), q_{1,i}^F\big) + \frac{1}{N}\sum_{i\in\mathcal{I}_B} w_i \ell\big(f_\theta^B(q_{0,i}), q_{k_i,1}^B\big)\,, \tag{10}$$

where $q_{k_i,1}^B = q_{k_i}^B(\tau_1; q_{0,i})$ and w_i is given by equation 8. Rows outside $\mathcal{I}_B$ contribute only to the Full loss; no Basic loss is evaluated against a missing or failed target. Both sums retain the Full-row denominator N.

Autoregressive continuation. Starting from $\hat{q}_{i,0} = q_{0,i}$, we unroll only the Full head: $\hat{q}_{i,h} = f_\theta^F(\hat{q}_{i,h-1})$ for $h = 1, 2, 3$, using the corresponding saved time increment at each step. The three-step continuation objective is

$$\mathcal{L}_3(\theta) = \mathcal{L}_1(\theta) + \frac{0.5}{2N}\sum_{i=1}^{N}\sum_{h=2}^{3} \ell(\hat{q}_{i,h}, q_{i,h}^F). \tag{11}$$

Thus, the Basic head retains one-step supervision while later Full predictions are trained autoregressively without teacher forcing. The reported sparse-anchor runs use 20 one-step epochs followed by 60 continuation epochs, with checkpoints selected on validation data. Since auxiliary weights change the total objective, raw objective values are not directly comparable measures of model quality.

Auxiliary-data construction. The reported $J_k(3)$ cache uses saved anchor indices $\{0, 10, 30, 60, 90\}$, and data-budget experiments use nested subsets of these cached training anchors. Some intermediate FitzHugh–Nagumo and Navier–Stokes subsets retain only indices $\{0, 10\}$ from an additional initial-condition group. Although the generator can enumerate every start with three available future frames, the reported sparse-anchor results use the five-index set above. A separate all-time, single-step protocol is documented in Appendix B.2. For implementation traceability, the cache field `selected_operator` stores the minimum-$J_k(3)$ route used for adaptive Basic supervision; `formula_selected_operator` stores a historical diagnostic and does not control this route.

Inference and information boundary. At test time, the model starts from the true initial state and repeatedly applies only f_θ^F under the full mask, feeding each prediction into the next step. The reported ten-step rollout is an *evaluation horizon*, distinct from the three-output routing window. Inference requires no Basic simulation, discrepancy score, operator route, confidence calibration, or future Full reference. The distinction is therefore explicit: adaptive routing and weighting construct auxiliary supervision offline, while the deployed Full predictor operates without either mechanism.

Algorithm 1: Offline routing, confidence weighting, and Full-PDE learning

1. **Input:** Full training trajectories, admissible Basic operators, saved anchors, and physical and numerical parameters.
2. At each training anchor, read three subsequent Full training frames. Advance every admissible Basic candidate autonomously from the same state to the corresponding saved times; record failures.
3. Compute valid discrepancies using equation 4 and select the minimum using equation 5. Store aligned first-step Full and Basic targets. Retain the Full row when no Basic candidate is valid.
4. Pool all valid candidate scores for each PDE and calibrate confidence using the training initial-condition groups. Freeze the mapping and assign weights by equation 8 on common-valid support.
5. Train the shared FNO and its two heads with equation 10. In the separate sparse-anchor continuation protocol only, add equation 11. Select checkpoints on validation.
6. **Output:** A Full predictor for route-free autoregressive inference from the true initial state.

4 EXPERIMENTS

4.1 DATA, TRAINING, AND METRICS

We evaluate nonlinear one-dimensional Burgers (periodic, 256 cells), two-component FitzHugh–Nagumo reaction–diffusion (Neumann, 64^2 cells), and unforced incompressible two-dimensional Navier–Stokes (periodic, 64^2 cells). Burgers uses nonlinear transport $-qq_x$. Train/validation/test trajectory counts are $480/80/80$, $25/5/10$, and $45/9/18$, respectively. The splits hold out initial conditions within the studied parameter grids; they are not parameter-OOD tests. Saved intervals are $\Delta t = 0.01$, 0.05, and 0.005, so ten prediction steps cover physical times 0.1, 0.5, and 0.05.

The primary route minimizes $J_k(3; q_0)$ in equation 5 among valid Basic candidates at each Full *training* anchor. Each candidate evolves autonomously from the same q_0 for three saved steps; the score divides each channel's spatial squared error by that channel's mean $q_0^2 + 10^{-7}$, fixed across all three times. The three future Full training frames are needed to construct the route. The selected Basic target used by the network is only its *first* step. Fixed comparators use Burgers diffusion, FHN reaction, and projected Navier–Stokes advection. Both routes use common-valid Basic support and each selected form's own train-calibrated dynamic weight. For the main all-time runs, the seed-0, width-20, 12-mode FNO trains with a one-step Full loss and dynamically weighted one-step Basic loss: 40 epochs for Burgers and Navier–Stokes, and 50 epochs for FHN. Validation one-step Full error selects checkpoints. The separate sparse-anchor runs use 20 one-step epochs followed by 60 three-step Full-head continuation epochs (Appendix B.3).

At test time only the Full head is rolled out from true q_0, using its own predictions after the initial state. One-step Full-head channel-normalized MSE follows equation 9 and has no square root. Step-10 error is the mean across test trajectories of spatial-and-channel RMSE divided by each trajectory's initial-state RMS. Ten steps are the *neural evaluation horizon*, not the three-step Basic routing window. The two reported error metrics have different scales.

4.2 ALL-TIME SINGLE-STEP TRAINING CURVES

Figure 3 shows Full-head training and validation one-step normalized MSE (NMSE) for the separate all-time FHN and Navier–Stokes runs. Both adaptive and fixed routes use all available one-step training starts and a dynamically weighted Basic auxiliary loss. The adaptive Basic form is selected offline using J_3; the neural training objective itself remains one-step, without a three-step autoregressive continuation. Validation one-step NMSE selects the checkpoints marked in the figure. These curves describe training behavior under this all-time protocol. Test metrics appear in Tables 1 and 2; the sparse-anchor protocol is reported in Appendix B.3.

Table 1: **All-time, single-step $J_k(3)$ routing results on held-out initial conditions.** Seed-0 Full-head test errors after 40 epochs for Burgers and Navier–Stokes or 50 epochs for FHN. Each route uses all available one-step training starts, common-valid Basic support, and its own train-calibrated dynamic auxiliary weight. Checkpoints are selected by validation one-step error. Fixed forms are Burgers diffusion, FHN reaction, and Navier–Stokes projected advection. One-step error is channel-normalized MSE (no square root); step-10 error is trajectory-mean RMSE/initial RMS. The Full head alone is used at test time.

PDE	Route	Test n	One-step MSE	Step-10 relative RMSE
Burgers	J_3 adaptive	80	0.000665915	0.0604664
	Fixed	80	0.000615593	0.0583517
FitzHugh–Nagumo	J_3 adaptive	10	0.0206921	0.0925771
	Fixed	10	0.0250065	0.105369
Navier–Stokes	J_3 adaptive	18	0.000538791	0.0750537
	Fixed	18	0.000731554	0.0974940

Table 2: **Separate all-time, single-step FHN and Navier–Stokes results.** Seed-0 models train for 50 epochs (FHN) or 40 epochs (Navier–Stokes), using respectively 2,500 or 4,500 training pairs per route. The checkpoint is selected by validation one-step error; best epoch counts from 1. Test one-step error is channel-normalized MSE, while step-10 error is trajectory-mean RMSE/initial RMS. All listed test rollouts are finite. Fixed forms are FHN reaction and projected Navier–Stokes advection. This protocol uses dynamically weighted, one-step Full and Basic losses, without autoregressive training, and is distinct from sparse-anchor Appendix Tables 3 and 4.

PDE	Route	Train ep.	Best ep.	Test n	H1 NMSE	H10 rel. RMSE
FitzHugh–Nagumo	J_3 adaptive	50	48	10	0.0206921	0.0925771
	Fixed	50	49	10	0.0250065	0.105369
Navier–Stokes	J_3 adaptive	40	38	18	0.000538791	0.0750537
	Fixed	40	25	18	0.000731554	0.0974940

4.3 Ten-step Full-head prediction

Table 1 summarizes all six all-time, single-step models. Adaptive routing lowers step-10 error for FHN (0.0925771 versus 0.105369) and Navier–Stokes (0.0750537 versus 0.0974940), while fixed routing is lower for Burgers (0.0583517 versus 0.0604664). The one-step test error has the same route ranking in each system. Each route is evaluated on 80 Burgers, 10 FHN, and 18 Navier–Stokes test trajectories. The three PDEs have distinct physical time horizons.

Figure 4 details the all-time, single-step FHN and Navier–Stokes models summarized in Table 1. Its test curves trace mean Full-head rollout error from steps 1–10, and its parameter panels show step-10 means within the seen Da and Re grids. These models have no autoregressive training loss; ten steps are an evaluation horizon rather than a training or Basic-routing horizon.

Figure 5 illustrates the Navier–Stokes Full-head predictions for one held-out initial condition; the aggregate test errors remain those in Table 1 and Figure 4.

In the all-time, single-step protocol, adaptive routing lowers both FHN and Navier–Stokes test one-step NMSE and step-10 relative RMSE compared with the corresponding fixed form. Its training uses all 100 one-step starts per training trajectory and no autoregressive loss; Figures 3 and 4 show its training curves and test rollouts, while Table 2 reports the test values and checkpoint epochs. The separate sparse-anchor comparison is reported in Appendix Table 3; the all-time data preparation and training protocol is in Appendix B.2.

5 Discussion and Limitations

The primary selector minimizes $J_k(3; q_0)$ (equation 4–equation 5): each valid Basic candidate evolves autonomously from the same Full *training* anchor, and its next three states are compared with that trajectory's three Full training frames. Its in-window advantage over a valid fixed candidate follows from the minimum; it does not establish fidelity over longer Basic horizons or benefit to the learned predictor. The selected Basic contributes only its first-step auxiliary target; the ten-step

rollout in Tables 1 and 3 evaluates the *neural Full head*, not a ten-step Basic route. The library is restricted to admissible one-term equations; Navier–Stokes advection must retain the incompressibility projection. Because $J_k(3)$ reads future Full training frames, selection is offline. Full-head inference needs neither those frames nor a Basic solve, but this does not make the selector available from a new state alone.

The all-time, single-step comparison favors fixed routing for Burgers and adaptive routing for FitzHugh–Nagumo and Navier–Stokes (Table 1). The separate sparse-anchor comparison gives a small Burgers step-10 advantage to J_3 routing, while fixed Basic wins for FitzHugh–Nagumo and Navier–Stokes (Appendix Table 3). The matched route–weight ablation shows mixed effects of the train-calibrated J_3 weight (Appendix Table 4); at both weights, fixed remains better for FHN and Navier–Stokes. Better in-window Basic fidelity does not consistently transfer to better ten-step neural prediction. Channel-scale effects, operator cancellation, optimization, and error accumulation remain possible mechanisms, not established causes.

The all-time FHN and Navier–Stokes comparisons favor adaptive routing on both one-step and ten-step test errors (Table 2). This changes the observed route ranking relative to the sparse-anchor runs, while also changing the number of training windows and removing the three-step Full-head continuation objective. The current experiments do not isolate which protocol difference accounts for the ranking change.

All reported model accuracy uses seed 0. The splits hold out initial conditions within studied parameter grids, not unseen physical parameters; FHN and Navier–Stokes each have only five independent training initial-condition groups. Ten neural steps do not establish longer-horizon stability. Candidate Basic replay timings described in Appendix C exclude Full trajectory generation, I/O, J_3 calculation, bootstrap calibration, and training. Thus total data efficiency is unmeasured. Matched Full-only controls, multiple training seeds, parameter-OOD tests, longer rollouts, and end-to-end cost accounting remain necessary before broader performance claims.

6 Conclusion

We implemented per-training-anchor, minimum-$J_k(3)$ selection of admissible Basic PDEs and a training-set calibration of each selected target's auxiliary weight. The route uses future Full *training* frames to choose a one-step Basic label; test prediction uses only the neural Full head. In the single-seed held-out-initial-condition comparisons, adaptive routing improves both reported test errors for FHN and Navier–Stokes while fixed improves both for Burgers in the all-time, single-step protocol (Table 1). The separate sparse-anchor protocol yields a different route ranking (Appendix Table 3); dynamic weighting has mixed effects in the sparse-anchor ablation. These results define a reproducible offline selection protocol, but do not establish a deployable state-only route, a general learning gain, or a total-cost advantage.

AI use statement

We used OpenAI Codex, a generative AI tool, to assist with method and experiment framing, drafting and editing the manuscript, organizing literature and citations, preparing LaTeX and figures, inspecting code, and interpreting and checking saved experimental metrics. The reported numerical results come from the project's simulation and training records. The authors are responsible for reviewing AI-assisted material and for the accuracy and originality of the final text, citations, figures, and claims.

References

Francesco Alesiani, Makoto Takamoto, and Mathias Niepert. HyperFNO: Improving the generalization behavior of Fourier neural operators. In *NeurIPS 2022 Workshop on Machine Learning and Physical Sciences*, 2022. URL https://neurips.cc/virtual/2022/57001.

Johannes Brandstetter, Max Welling, and Daniel E. Worrall. Lie point symmetry data augmentation for neural PDE solvers. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 2241–2256. PMLR, 2022. URL https://proceedings.mlr.press/v162/brandstetter22a.html.

Jared L. Callaham, James V. Koch, Bingni W. Brunton, J. Nathan Kutz, and Steven L. Brunton. Learning dominant physical processes with data-driven balance models. *Nature Communications*, 12:1016, 2021. doi: 10.1038/s41467-021-21331-z. URL https://www.nature.com/articles/s41467-021-21331-z.

Wuyang Chen, Jialin Song, Pu Ren, Shashank Subramanian, Dmitriy Morozov, and Michael W. Mahoney. Data-efficient operator learning via unsupervised pretraining and in-context learning. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/0bac492172db3311c7e116098cfcf521-Abstract-Conference.html.

Gaurav Gupta, Xiongye Xiao, and Paul Bogdan. Multiwavelet-based operator learning for differential equations. In *Advances in Neural Information Processing Systems*, volume 34, pp. 24048–24062, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/c9e5c2b59d98488fe1070e744041ea0e-Abstract.html.

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 *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 17616–17635, 2024. URL https://proceedings.mlr.press/v235/hao24d.html.

Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for PDEs. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://openreview.net/forum?id=JC1VKK3UXk.

Hamda Hmida, Hsiu-Wen Chang, and Youssef Mesri. Compositional neural operators for multi-dimensional fluid dynamics, 2026a. URL https://arxiv.org/abs/2605.11691. Preprint.

Hamda Hmida, Hsiu-Wen Chang Joly, and Youssef Mesri. CompNO: A novel foundation model approach for solving partial differential equations. *Applied Sciences*, 16(2):972, 2026b. doi: 10.3390/app16020972. URL https://www.mdpi.com/2076-3417/16/2/972.

Amanda A. Howard, Mauro Perego, George E. Karniadakis, and Panos Stinis. Multifidelity deep operator networks for data-driven and physics-informed problems. *Journal of Computational Physics*, 493:112462, 2023. doi: 10.1016/j.jcp.2023.112462. URL https://doi.org/10.1016/j.jcp.2023.112462.

Cheng Jing, Uvini Balasuriya Mudiyanselage, Abhishek Verma, Kallol Bera, Shahid Rauf, and Kookjin Lee. Extending Fourier neural operators for modeling parameterized and coupled PDEs. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/c253f6fe5cc2bd1905427da63ac37917-Abstract-Conference.html.

Bryan E. Kaiser, Juan A. Saenz, Maike Sonnewald, and Daniel Livescu. Automated identification of dominant physical processes. *Engineering Applications of Artificial Intelligence*, 116: 105496, 2022. doi: 10.1016/j.engappai.2022.105496. URL https://doi.org/10.1016/j.engappai.2022.105496.

Bongseok Kim, Haoyang Zheng, Michael Penwarden, and Guang Lin. Active learning with Bayesian multi-fidelity Laplace neural operators for oscillatory parametric PDEs, 2026. URL https://arxiv.org/abs/2502.00550v2. Preprint, version 2, revised September 14, 2026.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to PDEs. *Journal of Machine Learning Research*, 24(89):1–97, 2023. URL https://jmlr.org/papers/v24/21-1524.html.

Zongyi Li, Nikola 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.

Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *ACM/IMS Journal of Data Science*, 1(3):1–27, 2024. doi: 10.1145/3648506. URL https://doi.org/10.1145/3648506.

Phillip Lippe, Bastiaan S. Veeling, Paris Perdikaris, Richard E. Turner, and Johannes Brandstetter. PDE-Refiner: Achieving accurate long rollouts with neural PDE solvers. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/d529b943af3dba734f8a7d49efcb6d09-Abstract-Conference.html.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021. doi: 10.1038/s42256-021-00302-5. URL https://www.nature.com/articles/s42256-021-00302-5.

Siying Ma, Mehrdad Momeni Zadeh, Mauricio Soroco, Wuyang Chen, Jiguo Cao, and Vijay Ganesh. Learning data-efficient and generalizable neural operators via fundamental physics knowledge. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=mJiPqOzc3O.

Michael McCabe, Bruno Régaldo-Saint Blancard, Liam Holden Parker, Ruben Ohana, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Siavash Golkar, Geraud Krawezik, Francois Lanusse, Mariel Pettee, Tiberiu Tesileanu, Kyunghyun Cho, and Shirley Ho. Multiple physics pretraining for spatiotemporal surrogate models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://openreview.net/forum?id=DKSI3bULiZ.

Md Ashiqur Rahman, Robert Joseph George, Mogab Elleithy, Daniel Leibovici, Zongyi Li, Boris Bonev, Colin White, Julius Berner, Raymond A. Yeh, Jean Kossaifi, Kamyar Azizzadenesheli, and Anima Anandkumar. Pretraining codomain attention neural operators for solving multiphysics PDEs. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/bc75fa9843a7905bbed9d83895a88f7f-Abstract-Conference.html.

Shashank Subramanian, Peter Harrington, Kurt Keutzer, Wahid Bhimji, Dmitriy Morozov, Michael W. Mahoney, and Amir Gholami. Towards foundation models for scientific machine learning: Characterizing scaling and transfer behavior. In *Advances in Neural Information Processing Systems*, volume 36, pp. 71242–71262, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/e15790966a4a9d85d688635c88ee6d8a-Abstract.html.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Dan MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. PDEBench: An extensive benchmark for scientific machine learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://arxiv.org/abs/2210.07182.

Xiongye Xiao, Defu Cao, Ruochen Yang, Gaurav Gupta, Gengshuo Liu, Chenzhong Yin, Radu Balan, and Paul Bogdan. Coupled multiwavelet operator learning for coupled differential equations. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=kIo_C6QmMOM.

A PDE DATA, NUMERICAL METHODS, AND SPLITS

The active export contains 752 Full trajectories. The export index, summary, and catalog record each trajectory's source and split. Every trajectory has 101 saved frames. Initial-condition groups, including all studied parameter cases, are kept within one split; the test sets therefore probe new initial conditions at seen parameter values.

Periodic Burgers. On the unit interval, $q_t = -qq_x + Dq_{xx}$ with $D = 1/\mathrm{Re}$ and $\mathrm{Re} \in \{0.1, 0.3, 0.6, 1, 1.7, 3, 10, 30, 100, 300\}$. There are 64 paired initial conditions per value, giving 640 trajectories and train/validation/test counts 480/80/80 (48/8/8 initial-condition groups). The 256-point, $3/2$-padded Fourier solver uses float64 ETDRK4 with internal step 0.001 and saves float32 fields every 0.01 over $t \in [0, 1]$. Its advective term is nonlinear rather than constant-velocity transport.

FitzHugh–Nagumo. The two-field reaction–diffusion system uses a 64^2 finite-volume grid on a length-two square with zero-flux boundaries. The current 40 trajectories come from the direct diffusion scan: the reaction multiplier is fixed at $\alpha = 1$ and the reaction constant is $\kappa = 0.005$; $D_u = 4/\mathrm{Da}$ and $D_v = 5D_u$ for $\mathrm{Da} \in \{100, 300, 1000, 4000, 10000\}$, with eight paired initial conditions per value. The trajectory split is 25/5/10 (5/1/2 groups). The source generator uses RK45 with relative/absolute tolerances $10^{-3}/10^{-6}$ and saves 101 frames over $t \in [0, 5]$ at intervals of 0.05. The export retains unrounded float64 states for these analyses. An older 576-trajectory reaction-multiplier sweep is excluded from this export.

Unforced incompressible Navier–Stokes. On the periodic unit square, $\partial_t \mathbf{u} = \mathcal{P}[-\mathbf{u} \cdot \nabla \mathbf{u}] + \nu \Delta \mathbf{u}$, where $\mathcal{P}$ is the Fourier Leray projector and $\nu = 1/\mathrm{Re}$. The seen Reynolds numbers are $\{1, 3, 10, 30, 50, 70, 100, 300, 1000\}$. Eight paired initial conditions per value give 72 trajectories, split 45/9/18 (5/1/2 groups). The 64^2 solver uses $2/3$ de-aliasing and float64 projected ETDRK4 with internal step 0.001; its 101 float32 saved fields span $t \in [0, 0.5]$ at intervals of 0.005. The advection-only Basic candidate retains the pressure projection. Numerical checks at finite resolution do not certify convergence for every case.

B FNO TRAINING AND CHECKPOINTS

The active FNO implementation uses four shared Fourier blocks of width 20 with 12 modes per spatial axis and separate Full and Basic heads. One observed state, spatial coordinates, train-normalized physical parameters, the operator mask or Basic one-hot, and current-state statistics condition the network. Inference uses only the Full head from a true initial state; it neither runs Basic solvers nor reads future truth.

The common-valid one-step control uses seed 0, Adam, batch size 16, 20 epochs, shared/head learning rates 10^{-3}, weight decay 10^{-4}, and a per-batch cosine schedule. Both routes use constant Basic weight 0.7 on their shared valid support and keep every Full pair. Validation Full one-step error selects the checkpoint; selected adaptive/fixed epochs are 20/19 for Burgers, 20/20 for FHN, and 9/9 for Navier–Stokes. The run records retain executed commands, effective configurations, and test metrics.

The paired Burgers and FHN continuation starts each route from its own validation-selected common-valid checkpoint. It adds 60 epochs of three-step autonomous Full-head supervision while keeping a one-step Basic target. Its future-Full loss multiplier is 0.5; Adam shared/head learning rates are 2×10^{-4}, batch sizes are 16 (Burgers) and 4 (FHN), weight decay is 10^{-4}, the cosine schedule runs per batch, and gradients are clipped at norm 1. Validation one-step and ten-step errors relative to each route's warm start select the checkpoint, without test-based selection. Selected adaptive/fixed continuation epochs are 52/52 (Burgers) and 51/51 (FHN). No Navier–Stokes continuation was run. The execution and metric records retain these choices. This continuation adds both optimization and future-frame supervision; before/after change does not isolate a three-step loss effect.

B.1 SPARSE-ANCHOR J_3 REPRODUCIBILITY PROTOCOL

The sparse-anchor J_3 experiment uses seed 0, 20 one-step pretraining epochs at learning rate 10^{-3}, then 60 autonomous three-step epochs at 2×10^{-4}, with batch sizes 16/4/4 for Burgers/FHN/Navier–Stokes. The Full one-step and Basic one-step losses remain in both stages; the later two Full steps enter the continuation objective with multiplier 0.5. The model is initialized separately for each PDE but identically across its fixed/adaptive pair, as verified by the per-run `initial_state_sha256`. The sparse-anchor adaptive and fixed training caches each record `candidate_j`, `candidate_valid`, `candidate_errors`, `selected_j`,

`selected_operator`, and `weight`, with `common_aux_valid` as a group attribute. The cache manifest records source SHA-256 hashes, anchor indices, horizon, and rejected-candidate reasons. The six run records retain training commands, effective hyperparameters, source and cache hashes, initialization hashes, and code hashes.

Checkpoint choice uses only validation trajectories. Pretraining saves the epoch with smallest validation one-step Full NMSE. In continuation, an epoch is feasible when both its validation one-step NMSE and ten-step Full-head RMSE/initial RMS do not exceed that pretraining checkpoint's values. Among feasible epochs, the chosen checkpoint minimizes the sum of the two validation errors divided by their respective pretraining values; if none is feasible, the minimum-score continuation checkpoint is used. The selected adaptive/fixed continuation epochs are 48/55 (Burgers), 57/57 (FHN), and 48/48 (Navier–Stokes). All six selected checkpoints have the recorded `dual_nonworsening` status; test errors were calculated after selection. The training implementation and each run's history and metrics retain the checkpoint decision.

Valid candidate J_3 distribution and calibration. The following quantiles use every finite, admissible three-step Basic candidate in the *training* cache, including both operators where valid; they are not quantiles of the selected minimum J_3. Values are dependent within initial-condition groups.

PDE	N	min	Q_{25}	median	Q_{75}	max
Burgers	4464	0	6.19×10^{-22}	9.30×10^{-5}	7.78×10^{-3}	0.989
FHN	250	7.36×10^{-5}	4.85×10^{-3}	9.46×10^{-3}	2.88×10^{-2}	0.794
Navier–Stokes	450	3.20×10^{-25}	1.77×10^{-5}	2.02×10^{-4}	7.34×10^{-4}	0.238

The quantiles were read from `candidate_j` where `candidate_valid` is true in the adaptive cache; the saved calibration record independently lists candidate counts and medians. Calibration uses six empirical quantile bins per PDE, 1000 seed-0 bootstrap resamples of initial-condition groups, and 95% percentile intervals for each bin-conditional *mean* J_3. Every recorded bin contains 48 Burgers groups or five FHN/Navier–Stokes groups, respectively; the latter group counts limit interval precision. Adjacent bins would be merged if a bin contained fewer than three groups; none was merged in these saved calibrations. Upper endpoints are made monotone before the exponential weight mapping. These intervals do not provide coverage for individual candidate errors. The calibration implementation and record preserve the exact bin edges, bin sizes, group counts, and weight scales. These sparse-anchor runs and their budgets use the same J_3 route and dynamic-weight protocol.

B.2 Separate all-time, single-step protocol

The separate all-time branch retains the same train/validation/test trajectory split but expands each 101-frame *training* trajectory from five sparse anchors to all 100 one-step starts, indices $0, \dots, 99$. At starts $0, \dots, 97$, each admissible Basic form advances autonomously from the same current field for three saved steps. Its trajectory is compared with the corresponding three future Full *training* frames using Eq. 4; adaptive `selected_operator` is the valid candidate with minimum $J_k(3)$. A separate historical diagnostic is preserved in `formula_selected_operator`; it does not route adaptive Basic supervision. Fixed rows use Burgers diffusion, FHN reaction, and projected Navier–Stokes advection without per-anchor selection.

Starts 98 and 99 have a true next Full frame but fewer than three future frames. They retain one-step Full supervision and have no valid Basic candidate or auxiliary loss: `candidate_valid` is false, `common_aux_valid` is false, and the Basic weight is zero. Their adaptive `selected_operator` is -1; fixed rows keep the fixed operator identifier but mark it invalid. No later Full frame is fabricated. Across the three PDEs, this gives 48,000 Burgers, 2,500 FHN, and 4,500 Navier–Stokes training pairs per route, including 1,100 Full-only tail pairs.

The three-step horizon in this branch is solely an *offline Basic-data selection* window. Its formal neural objective is Eq. 10: one-step Full loss plus dynamically weighted one-step Basic loss on common-valid rows. It has no three-step autoregressive training loss or continuation phase; validation one-step error selects checkpoints, and ten-step rollout remains an evaluation diagnostic. The all-time protocol and its larger number of training windows are separate from the sparse-anchor

Table 3: **Sparse-anchor $J_k(3)$ routing results.** Seed-0 Full-head test errors after 20 one-step pretraining plus 60 three-step continuation epochs. Both routes use common-valid support and their own train-calibrated dynamic auxiliary weights. One-step error is channel-normalized MSE; step-10 error is trajectory-mean RMSE/initial RMS. The Full head alone is used at test time.

PDE	Route	Test n	One-step MSE	Step-10 relative RMSE
Burgers	J_3 adaptive	80	0.000316710	0.0400693
	Fixed	80	0.000334482	0.0412066
FitzHugh–Nagumo	J_3 adaptive	10	0.0338639	0.143345
	Fixed	10	0.0344055	0.138301
Navier–Stokes	J_3 adaptive	18	0.00226099	0.206293
	Fixed	18	0.00210255	0.197135

Table 4: **Sparse-anchor J_3 route–weight ablation.** Test step-10 RMSE/initial RMS (lower is better), seed 0, 20+60 epochs. Constant weight is 0.7 on common-valid support. Dynamic weight uses each selected Basic form's own J_3 and a calibration fitted only on training candidates; it is not sample-level predictive coverage. Routing uses future Full *training* frames.

Route	Weight	Burgers	FitzHugh–Nagumo	Navier–Stokes
Fixed	Constant 0.7	0.03791	0.13015	0.20408
J_3 adaptive	Constant 0.7	0.03691	0.14650	0.20525
Fixed	Dynamic	0.04121	0.13830	0.19713
J_3 adaptive	Dynamic	0.04007	0.14335	0.20629

20-plus-60-epoch results in Appendix Tables 3 and 4; those values must not be attributed to this branch.

The FHN and Navier–Stokes values in Tables 1 and 2 come from the four saved run metric and protocol records. The Burgers values in Table 1 come from its paired all-time run records. The test sets hold out initial conditions within seen parameter grids; one training seed and finite ten-step rollouts do not establish statistical significance, parameter extrapolation, or longer-horizon stability.

B.3 Sparse-anchor dynamic-weight results

Table 3 reports the six sparse-anchor models, which use five training starts per trajectory and a 20-plus-60-epoch schedule with three-step Full-head continuation. These outcomes should not be pooled with the all-time, single-step results in main Table 1.

B.4 Sparse-anchor route and weight ablations

Table 4 crosses fixed versus minimum-J_3 routing with constant 0.7 versus the selected candidate's own train-calibrated dynamic weight. Candidate trajectories, common-valid support, data splits, architecture, seed, and 20+60-epoch schedule are matched within this 2×2 protocol. At constant weight, adaptive lowers Burgers step-10 error but fixed wins FHN and Navier–Stokes. Dynamic weighting is mixed: it raises both Burgers errors, lowers adaptive FHN but raises fixed FHN error, and changes the two Navier–Stokes errors in opposite directions. In-window minimum J_3 is true by construction; it neither certifies ten-step neural accuracy nor learning benefit. The bootstrap interval calibrates a training bin's mean J, with only five independent FHN and Navier–Stokes initial-condition groups.

C Extended routing, weighting, and candidate budgets

Saved training indices $\{0, 10, 30, 60, 90\}$ give 2400 Burgers, 125 FHN, and 225 Navier–Stokes Full pairs, totaling 2750. The primary offline J_3 route advances every admissible Basic candidate autonomously for three saved steps and compares it with future *training* Full frames using Eq. 4. The bootstrap upper endpoint describes a training initial-condition cluster bin mean, not coverage of an individual example. The saved cache manifest and routing-count report record candidate validity and route counts.

The seven nested retained-anchor budgets are nominally 40, 50, 60, 70, 80, 90, and 100%. Their Burgers pair counts are $950, 1200, 1450, 1700, 1900, 2150, 2400$; FHN counts are $50, 60, 75, 85, 100, 110, 125$; Navier–Stokes counts are $90, 108, 135, 153, 180, 198, 225$. For FHN and Navier–Stokes, nominal 50/70/90% correspond to actual 48/68/88% anchor fractions: these subsets include the first two anchors of an additional initial-condition group. All parameter cases remain at retained groups. Each subset recalibrates weights from its training candidate errors. Both routes require all candidate errors and share a candidate-replay cost at a given budget. Failed numerical solves are timed; candidates rejected before a solver call have no replay cost. Across the seven budgets, adaptive has lower test one-step MSE at four Burgers, three FHN, and two Navier–Stokes budgets. The accuracy curves are nonmonotone and establish neither a general nor a matched total-cost efficiency gain. The 100% checkpoints are reused; 36 additional models use the same 20+60 epoch trainer, so optimizer updates differ across subset sizes. Reported replay time is a median of three warmed, single-thread Basic solver replay totals. It excludes Full simulation, I/O, J computation, calibration, and network training. The saved budget protocol and verification record specify the exact subsets and run receipts.

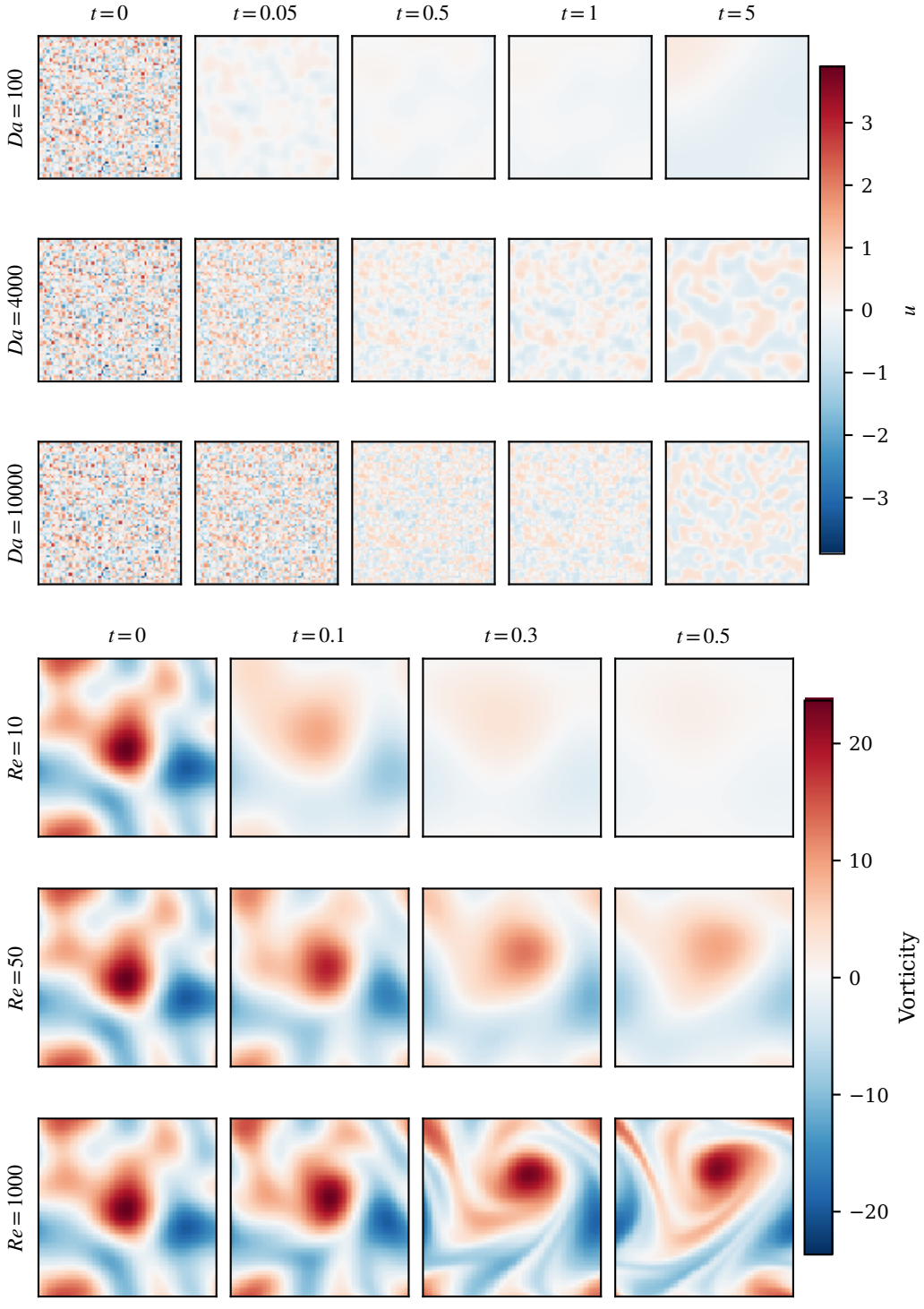


Figure 1: **Different Basic-process evolutions from the same initial field.** Diffusion fields (top) and Navier–Stokes vorticity fields (bottom) illustrate how the choice of process changes the ensuing field evolution. Each set of candidate evolutions begins from a common initial field; the displayed fields motivate comparing candidate dynamics with the corresponding Full trajectory when selecting an auxiliary Basic form.

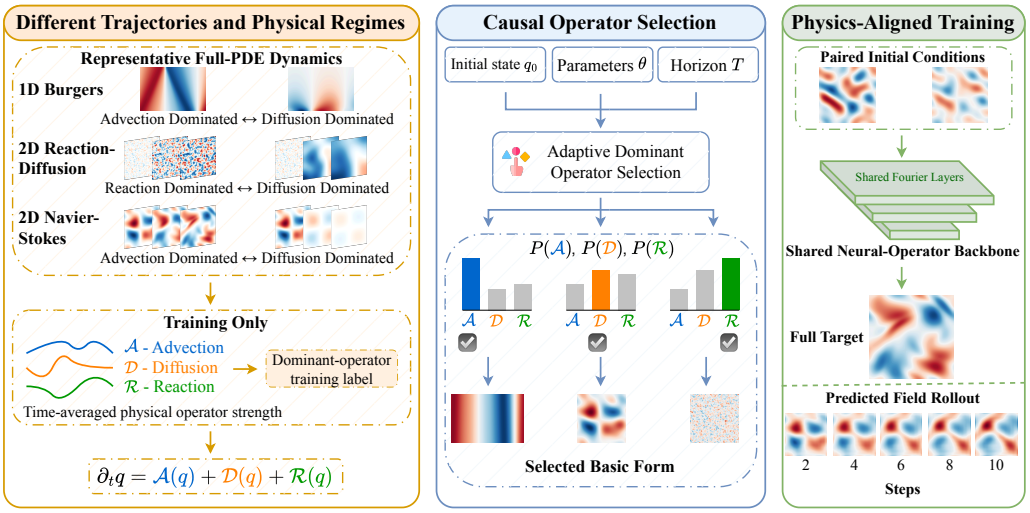


Figure 2: **Overview of the proposed framework.** At each Full training anchor, admissible Basic PDEs are simulated autonomously and compared with a short window of Full training dynamics. Adaptive routing selects the most dynamically consistent candidate, and a train-only confidence score controls its auxiliary supervision. A shared neural operator learns from Full and Basic targets through separate output heads. Inference uses only the Full head, without Basic simulations or routing.

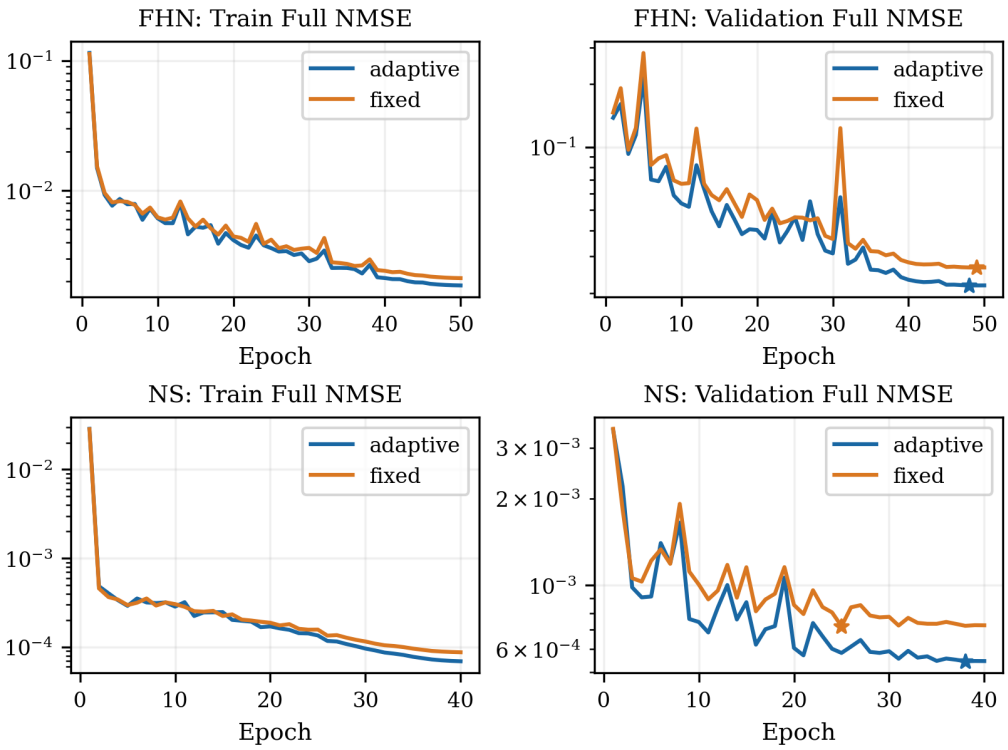


Figure 3: **Full-head one-step training and validation NMSE in the separate all-time runs.** FHN (top, 50 epochs) and Navier–Stokes (bottom, 40 epochs) show training (left) and validation (right) NMSE against epoch on logarithmic vertical axes. Blue denotes J_3 adaptive routing and orange the fixed Basic form. Stars mark the validation-selected one-step checkpoints: adaptive/fixed epochs 48/49 for FHN and 38/25 for Navier–Stokes. Both routes use seed 0 and one-step Full plus dynamically weighted one-step Basic losses; the three-step Basic comparison occurs only during offline auxiliary-data preparation.

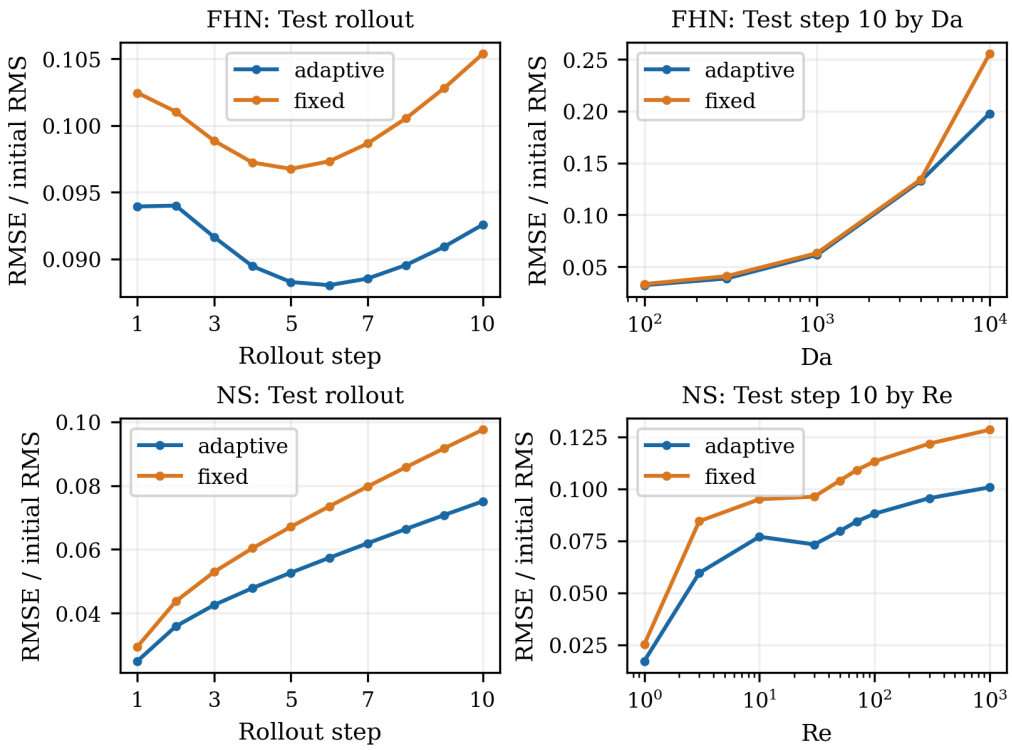


Figure 4: **Ten-step Full-head test rollouts in the separate all-time, single-step runs.** FHN (top) and Navier–Stokes (bottom) compare J_3 adaptive (blue) with fixed Basic routing (orange), seed 0. Left: mean test RMSE relative to each initial state's RMS at steps 1–10, over 10 FHN or 18 Navier–Stokes held-out initial conditions. Right: mean step-10 error by the five FHN Da or nine Navier–Stokes Re values, with two test initial conditions per parameter. The parameter grids are seen during training. J_3 uses future Full training frames only for offline Basic-data selection; test inference uses the Full head alone.

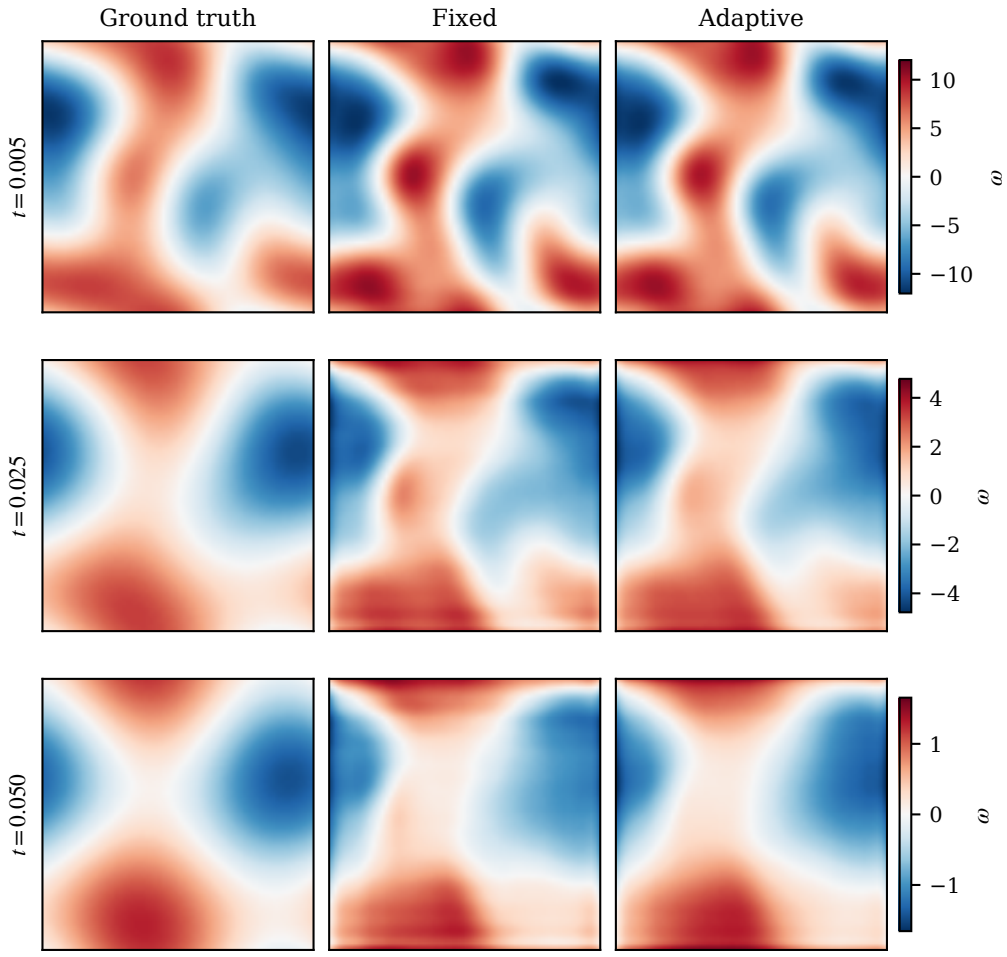


Figure 5: **Navier–Stokes vorticity fields during a ten-step Full-head rollout.** Ground truth, fixed-route prediction, and adaptive-route prediction are shown from left to right at steps 1, 5, and 10 (times 0.005, 0.025, and 0.050, respectively). This illustrative held-out initial condition at $Re = 1$ has the lowest adaptive step-10 error among the 18 test trajectories; it is not an aggregate result. Each row uses a common symmetric color scale. Periodic Gaussian smoothing ($\sigma = 1$ grid cell) and bilinear interpolation are applied only for display; model predictions and reported errors are computed from unsmoothed fields.