GREEN-ROUTED NEURAL OPERATORS: PHYSICS DETERMINES WHERE THE NETWORK READS

Anonymous authors
Paper under double-blind review

ABSTRACT

We identify a mismatch between the physical role of transport fields in many PDEs and their usual role in neural operators: PDEs use them to select read coordinates, whereas neural operators typically treat them only as input values. We address this mismatch with the *Green-Routed Neural Operator* (GRNO), which uses the governing equation to determine where latent features are sampled. A parameter-free equation adapter evaluates the diagnostic relation and constructs a departure map whose values are the read coordinates. A multiscale encoder–decoder combines centered and routed reads of latent features to learn the complete finite-time update. Across five two- and three-dimensional PDE systems, GRNO achieves the lowest mean final relative L^2 error on four under 40-step autoregressive evaluation and remains competitive on Keller–Segel. Fixed-weight route interventions reveal strong dependence on direction and spatial alignment in four systems, with weak dependence in Keller–Segel. In independently trained ablations, GRNO achieves lower mean errors than variants that supply the transport field only as an input feature, substitute a learned displacement for the equation-specified route, or apply the route with a spatial misalignment, across all five systems. It also substantially outperforms directly advecting the physical state and learning the remaining update, indicating that equation-specified read coordinates provide an effective structural prior for long-horizon PDE forecasting.

1 INTRODUCTION

Neural operators learn maps between functions and provide fast surrogates for partial differential equations (PDEs) (Lu et al., 2021; Li et al., 2021; Kovachki et al., 2023). During autoregressive forecasting, predictions become inputs, allowing errors to accumulate across the rollout (Brandstetter et al., 2022; Lippe et al., 2023; List et al., 2025). Existing architectures model spatial interactions through kernels, multiscale convolutions, attention, and coordinate-dependent queries; we examine how known transport fields should determine their read coordinates.

In many evolution PDEs, the equation already specifies more than which field values are relevant: it specifies the coordinates from which the evolving state should be read. Let q denote the state to be advanced, with μ specifying the physical parameters, geometry, and boundary conditions. The current state may first determine an auxiliary field $\psi_\mu[q]$ through an instantaneous diagnostic relation, after which the equation converts $\psi_\mu[q]$ into a velocity, drift, or phase-space vector $\boldsymbol{v}_\mu[q]$. We call $\psi_\mu[q]$ the *diagnostic field* and $\boldsymbol{v}_\mu[q]$ the *diagnostic carrier*. The carrier determines a departure map $\boldsymbol{\Xi}_{\Delta t,\mu}[q]$, whose values are the read coordinates. Sampling the previous state at these coordinates gives the transport contribution $q \circ \boldsymbol{\Xi}_{\Delta t,\mu}[q]$. Using $\boldsymbol{v}_\mu[q]$ as an input channel exposes its values at fixed grid indices, whereas using it to construct the departure map changes the coordinates from which features are read.

Endpoint supervision constrains the map $q^n \mapsto q^{n+1}$ while leaving the carrier-to-coordinate computation unspecified. Models such as Flowers make sampling locations explicit but learn their displacements from forecast data (Muser et al., 2026). Our mismatch lies in the computational role of the known carrier: the PDE uses it to determine read coordinates, while a generic endpoint learner leaves this role implicit.

We address this mismatch with the *Green-Routed Neural Operator* (GRNO). A parameter-free equation adapter computes the carrier and departure map, whose values are the read coordinates

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.

used to sample latent features in a multiscale encoder–decoder. Centered and routed reads of latent features jointly determine the complete finite-time update, and the diagnostic field, diagnostic carrier, and departure map are recomputed from every predicted state. Thus, the equation determines where the network reads, while the network learns how the state evolves (Figure 1).

Across five systems, GRNO achieves the lowest mean final error on four under 40-step autoregressive evaluation and remains competitive on Keller–Segel. Holding trained weights fixed and varying only the departure displacement reveals strong route dependence in four systems. Independently trained controls compare GRNO with variants that supply the transport field only as an input feature, substitute a learned displacement for the equation-specified route, apply the route with a spatial misalignment, or directly advect the physical state and learn the remaining update.

This work identifies a simple mismatch—a PDE can use a known field to choose coordinates while a neural operator receives it only as values—and provides an architecture and matched experiments for correcting and testing it.

2 Related Work

Neural-operator interactions. Graph and multipole kernels model nonlocal interactions (Li et al., 2020a;b); Fourier and multiwavelet operators learn spectral interactions (Li et al., 2021; Guibas et al., 2022; Gupta et al., 2021). CNO and U-NO use multiscale architectures (Raonić et al., 2023; Rahman et al., 2023), while transformer operators learn interactions through attention and coordinate queries (Cao, 2021; Wu et al., 2024; Wang et al., 2025; Liu & Yu, 2025).

Representations and coordinates. G-FNO, SFNO, and ReViT encode spatial symmetries (Helwig et al., 2023; Bonev et al., 2023; Wei et al., 2026), RieszNO represents directional structure (Liu et al., 2026), and RD-JEPA uses predictive latent pretraining for few-trajectory transfer across related PDEs (Si & Yan, 2026). Spatial transformers and deformable convolutions learn sampling transformations (Jaderberg et al., 2015; Dai et al., 2017); Geo-FNO and GINO connect physical domains to latent grids (Li et al., 2023a;b). Flowers and PARADIS learn displacements or latent advection (Muser et al., 2026; Pereira et al., 2026). GRNO constructs read coordinates from the diagnostic carrier.

Hybrid numerical models. Hybrid solvers learn discretizations (Bar-Sinai et al., 2019; Kochkov et al., 2021), corrections within numerical updates (Um et al., 2020), or inference-time corrections with numerical simulation (Fan et al., 2026). GRNO uses the physical transport computation to select latent features, while learning the complete finite-time update.

Green structure. Prior work learns Green kernels, constructs Green-inspired elliptic operators, or embeds discrete Green superposition (Boullé et al., 2022a;b; Melchers & Prins, 2026; Wang et al., 2024; Tang et al., 2026; Tan et al., 2026). GRNO uses the known Green inverse of a diagnostic relation to construct read coordinates for latent features.

3 Motivation: The PDE determines where information is read

Problem setup. Given fully observed training trajectories of a PDE with known coefficients and boundary conditions, we learn a map $\mathcal{M}_\theta$ from the current field $q^n = q(t_0 + n\Delta t)$ to the next field. Forecasting starts from one exact frame and proceeds recursively: $\widehat{q}^0 = q^0$ and $\widehat{q}^{n+1} = \mathcal{M}_\theta(\widehat{q}^n)$, with prescribed static inputs omitted from the notation. Models are trained separately for each system. We use Navier–Stokes to examine how known diagnostic carriers should enter this prediction map.

3.1 A concrete example: diagnostic routing in Navier–Stokes

Let $\omega^n = \omega(t_n)$ denote the vorticity at time t_n, with $\Delta t = t_{n+1} - t_n$. On the periodic domain $\Omega = \mathbb{T}^2$, forced two-dimensional Navier–Stokes can be written as

$$\partial_t \omega + \boldsymbol{u}[\omega]\cdot\nabla\omega = \nu\Delta\omega - \alpha\omega + f, \tag{1}$$

$$\Delta\psi = \omega - \overline{\omega}, \quad \boldsymbol{u}[\omega] = \nabla^\perp\psi. \tag{2}$$

Here ν is viscosity, α is linear drag, f is a prescribed forcing, and $\overline{\omega}$ is the spatial mean. The vorticity ω measures local fluid rotation and is the state being advanced. At each time, solving the Poisson

equation in Eq. (2) gives the diagnostic field ψ from the current ω. Its rotated gradient $\nabla^\perp\psi$ gives the transporting velocity $\boldsymbol{u}[\omega]$, which is the diagnostic carrier.

Solving the Poisson equation and applying $\nabla^\perp$ gives

$$\boldsymbol{u}[\omega](\boldsymbol{x}) = \int_\Omega \boldsymbol{K}(\boldsymbol{x},\boldsymbol{y})\big(\omega(\boldsymbol{y}) - \overline{\omega}\big)\,\mathrm{d}\boldsymbol{y}, \qquad \boldsymbol{K}(\boldsymbol{x},\boldsymbol{y}) = \nabla^\perp_{\boldsymbol{x}} G(\boldsymbol{x},\boldsymbol{y}), \tag{3}$$

where G is the mean-zero periodic Green function and $\boldsymbol{K}$ is the associated Biot–Savart kernel. This kernel weights vorticity contributions across the domain to obtain the velocity at each target point (Majda & Bertozzi, 2001).

Now fix a target coordinate $\boldsymbol{x}$ at time t_{n+1}. Backtracking the velocity field defines a departure map $\boldsymbol{\Xi}^n_{\Delta t}$, where $\boldsymbol{\Xi}^n_{\Delta t}(\boldsymbol{x})$ is the earlier coordinate of the material arriving at $\boldsymbol{x}$. Along this characteristic,

$$\begin{aligned} \omega^{n+1}(\boldsymbol{x}) &= \omega^n(\boldsymbol{\Xi}^n_{\Delta t}(\boldsymbol{x})) + \mathcal{C}^n_{\Delta t}(\boldsymbol{x}), \\ \boldsymbol{\Xi}^n_{\Delta t}(\boldsymbol{x}) &= \boldsymbol{x} - \Delta t\,\boldsymbol{u}^n(\boldsymbol{x}) + \mathcal{O}(\Delta t^2), \qquad \boldsymbol{u}^n := \boldsymbol{u}[\omega^n]. \end{aligned} \tag{4}$$

The pullback $\omega^n \circ \boldsymbol{\Xi}^n_{\Delta t}$ reads the old vorticity at the departure coordinate, while $\mathcal{C}^n_{\Delta t}$ collects viscosity, forcing, drag, and the remaining finite-time effects (Staniforth & Côté, 1991). For intuition, pure uniform rightward transport by one grid spacing makes each target read its left neighbor. The departure map extends this lookup to spatially varying velocities. The update therefore contains the explicit sequence

$$\omega^n \xrightarrow{\text{Biot–Savart}} \boldsymbol{u}^n \xrightarrow{\text{backtrace}} \boldsymbol{\Xi}^n_{\Delta t} \xrightarrow{\text{sample}} \omega^n \circ \boldsymbol{\Xi}^n_{\Delta t} \xrightarrow{+\,\mathcal{C}^n_{\Delta t}} \omega^{n+1}. \tag{5}$$

Appending $\boldsymbol{u}^n$ as an input channel exposes its values at fixed indices. Eq. (4) uses the same velocity differently: it moves the read coordinates before vorticity is read. Figure 1A illustrates this value-versus-coordinate distinction.

3.2 Where the mismatch arises: values versus read coordinates

Changing the current state affects both the sampled values and the coordinates used to read them. For $\mathcal{T}(q) := q \circ \boldsymbol{\Xi}[q]$, let $D\mathcal{T}[q]e$ denote the first-order response to a state perturbation e. The chain rule gives

$$D\mathcal{T}[q]e = \underbrace{e \circ \boldsymbol{\Xi}[q]}_{\text{state changes at a fixed map}} + \underbrace{(\nabla q \circ \boldsymbol{\Xi}[q])\,D\boldsymbol{\Xi}[q]e}_{\text{the departure map also changes}}. \tag{6}$$

The second term captures the response created when the carrier moves the read coordinates. In GRNO, this response enters through the departure map $\boldsymbol{\Xi}[q]$. When $D\boldsymbol{\Xi}[q]$ contains a Green inverse, a remote state perturbation can move a departure point and thereby change a local sample.

Endpoint supervision leaves the route underdetermined. For an endpoint map $\mathcal{P}$ and any alternative $\widetilde{\boldsymbol{\Xi}}[q]$, define

$$\widetilde{\mathcal{C}}[q] := \mathcal{P}(q) - q \circ \widetilde{\boldsymbol{\Xi}}[q] \quad \Longrightarrow \quad \mathcal{P}(q) = q \circ \widetilde{\boldsymbol{\Xi}}[q] + \widetilde{\mathcal{C}}[q]. \tag{7}$$

With an unrestricted remainder, a change in the departure map can be offset by a change in the remainder, producing the same next state. Eq. (7) therefore admits multiple internal decompositions for identical training pairs. The governing equation specifies which departure map GRNO should use.

The mismatch is the absence of an explicit link from the known carrier to the read coordinates:

$$q \xrightarrow{\text{diagnostic relation}} \boldsymbol{v}[q] \xrightarrow{\text{backtrace}} \boldsymbol{\Xi}[q] \xrightarrow{\text{sampling}} q \circ \boldsymbol{\Xi}[q]. \tag{8}$$

Routing is most useful when the equation-specified read coordinates materially affect the finite-time read. When centered and routed reads are similar, their predictive contributions are correspondingly similar. Section 5 measures this dependence across the five systems.

3.3 General form: diagnostically routed PDEs

The Navier–Stokes computation extends to systems with different states and diagnostic relations. Let $q : \Omega \to \mathbb{R}^c$ denote the evolving state, and let μ collect physical parameters, geometry, and boundary

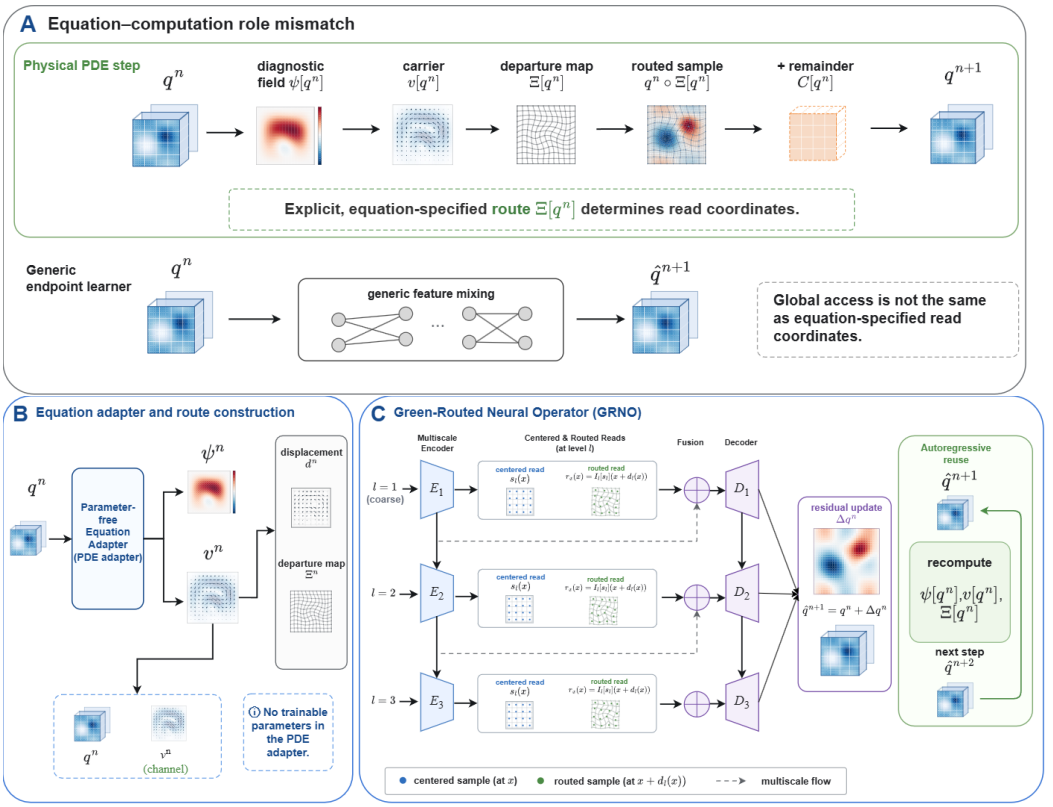


Figure 1: **Equation-specified routing and GRNO. (A)** The current state q^n determines a diagnostic field $\psi[q^n]$ and a diagnostic carrier $\boldsymbol{v}[q^n]$. Backtracing the carrier gives the departure map $\boldsymbol{\Xi}[q^n]$, which selects the routed physical sample $q^n \circ \boldsymbol{\Xi}[q^n]$; $\mathcal{C}[q^n]$ is the remainder containing all other finite-time effects. GRNO explicitly realizes the carrier-to-coordinate-to-sample chain by using the carrier to construct read coordinates. **(B)** The parameter-free equation adapter evaluates the known diagnostic relation, supplies the carrier as an encoder feature, and uses a backward trace to construct the displacement $\boldsymbol{d}^n$ and departure map $\boldsymbol{\Xi}^n$. **(C)** A local multiscale encoder produces features s_ℓ. At each scale, GRNO combines the centered read $s_\ell(\boldsymbol{x})$ with the routed read $r_\ell(\boldsymbol{x}) = \mathcal{I}_\ell[s_\ell](\boldsymbol{x} + \boldsymbol{d}_\ell(\boldsymbol{x}))$. The decoder fuses the two reads, predicts the complete state increment, and recomputes the diagnostic field, diagnostic carrier, and departure map from every predicted state during autoregressive rollout.

conditions. The current state first determines a diagnostic field through an instantaneous diagnostic relation,

$$\mathcal{A}_\mu \psi_\mu[q] = \mathcal{S}_\mu(q), \tag{9}$$

where $\mathcal{S}_\mu(q)$ forms the source from the current state and $\mathcal{A}_\mu^{-1}$ solves the diagnostic relation with the prescribed boundary and uniqueness conditions. The equation then converts the diagnostic field into a carrier,

$$\boldsymbol{v}_\mu[q] = \mathcal{V}_\mu(q) := \boldsymbol{v}_\mu^{\mathrm{loc}}(q) + \mathcal{B}_\mu \psi_\mu[q], \tag{10}$$

where $\boldsymbol{v}_\mu^{\mathrm{loc}}$ is an optional known local contribution and $\mathcal{B}_\mu$ converts the diagnostic field into a carrier. For Navier–Stokes, $\mathcal{A}_\mu = \Delta$, $\mathcal{S}_\mu(\omega) = \omega - \overline{\omega}$, $\mathcal{B}_\mu = \nabla^\perp$, and $\boldsymbol{v}_\mu^{\mathrm{loc}} = 0$. The carrier enters the evolution equation as

$$\partial_t q + \boldsymbol{v}_\mu[q] \cdot \nabla q = \mathcal{R}_\mu(q, \psi_\mu[q]), \tag{11}$$

with $\mathcal{R}_\mu$ collecting diffusion, reaction, forcing, and other dynamics not represented by the transport term. Conservative and phase-space forms are given in Appendix C.

When the diagnostic inverse has a Green representation, the state-to-carrier map can be written as

$$\begin{aligned} \boldsymbol{v}_\mu[q] &= \boldsymbol{v}_\mu^{\mathrm{loc}}(q) + \mathcal{B}_\mu \mathcal{A}_\mu^{-1} \mathcal{S}_\mu(q), \\ \boldsymbol{v}_\mu[q](\boldsymbol{x}) &= \boldsymbol{v}_\mu^{\mathrm{loc}}(q)(\boldsymbol{x}) + \int_\Omega \boldsymbol{K}_\mu(\boldsymbol{x}, \boldsymbol{y}) \mathcal{S}_\mu(q)(\boldsymbol{y}) \, \mathrm{d}\boldsymbol{y}, \qquad \boldsymbol{K}_\mu = \mathcal{B}_{\mu,\boldsymbol{x}} G_\mu. \end{aligned} \tag{12}$$

Here G_μ is the Green function of the diagnostic relation. The carrier then determines the short-time departure map

$$\boldsymbol{\Xi}_{\Delta t,\mu}[q](\boldsymbol{x}) = \boldsymbol{x} - \Delta t\, \mathcal{V}_\mu(q)(\boldsymbol{x}) + \mathcal{O}(\Delta t^2). \tag{13}$$

Let $\mathcal{P}_{\Delta t,\mu}$ denote the exact physical map over one step. Relative to the equation-specified departure map, it can be decomposed as

$$\boxed{\mathcal{P}_{\Delta t,\mu}(q) = q \circ \boldsymbol{\Xi}_{\Delta t,\mu}[q] + \mathcal{C}_{\Delta t,\mu}[q]}, \qquad \mathcal{C}_{\Delta t,\mu}[q] := \mathcal{P}_{\Delta t,\mu}(q) - q \circ \boldsymbol{\Xi}_{\Delta t,\mu}[q]. \tag{14}$$

The first term reads the state at the equation-specified read coordinates; the remainder contains all other finite-time effects. Eq. (14) defines the remainder relative to the equation-specified route; choosing a different route induces a different remainder.

We call a PDE *diagnostically routed* when an instantaneous diagnostic relation determines a carrier that, in turn, determines the read coordinates for the evolving state. Purely local reaction–diffusion equations lie outside this class, as do nonlocal fields that enter only as additive sources.

3.4 How route perturbations accumulate over long rollouts

We next examine how a coordinate error changes the sampled state and affects subsequent predictions. Consider a controlled perturbation of Eq. (14):

$$\begin{aligned} q^{n+1} &= q^n \circ \boldsymbol{\Xi}_n[q^n] + \mathcal{C}_n[q^n], \\ \widetilde{q}^{n+1} &= \widetilde{q}^n \circ \big(\boldsymbol{\Xi}_n[\widetilde{q}^n] + \boldsymbol{\eta}_n\big) + \mathcal{C}_n[\widetilde{q}^n] + r_n. \end{aligned} \tag{15}$$

Here $\boldsymbol{\eta}_n$ perturbs the read coordinates, while r_n perturbs the remainder. Assume locally that composition is bounded by B_n, the route is Lipschitz with constant $L_{\Xi,n}$, and the remainder is Lipschitz with constant $L_{\mathcal{C},n}$. With $e_n = \widetilde{q}^n - q^n$ and $M_n = \|\nabla q^n\|_{L^\infty}$,

$$\|e_{n+1}\|_2 \le A_n \|e_n\|_2 + M_n \|\boldsymbol{\eta}_n\|_2 + \|r_n\|_2, \qquad A_n := B_n + M_n L_{\Xi,n} + L_{\mathcal{C},n}. \tag{16}$$

The term $M_n\|\boldsymbol{\eta}_n\|_2$ bounds the sampling error caused by a coordinate displacement; M_n measures how sharply the state varies in space. The term $M_n L_{\Xi,n}$ captures feedback from an existing state error to the next departure map. Iterating the recursion gives

$$\|e_N\|_2 \le \Big(\prod_{k=0}^{N-1} A_k\Big)\|e_0\|_2 + \sum_{n=0}^{N-1}\Big(\prod_{k=n+1}^{N-1} A_k\Big)\big(M_n\|\boldsymbol{\eta}_n\|_2 + \|r_n\|_2\big). \tag{17}$$

A route defect can propagate through later rollout steps because each predicted state determines the next diagnostic solve and departure map. Eq. (17) separates the propagation of existing state error from newly introduced route and update errors. The assumptions, proof, and volume-preserving special case are provided in Appendix A. These observations motivate an architecture that preserves the centered computation while exposing the equation-specified route as a separate latent read.

4 Green-Routed Neural Operator

Figure 1B–C illustrates the resulting design: the equation determines the read coordinates, while the neural core learns the complete finite-time update.

4.1 Equation-specified routes

For state q^n, the parameter-free equation adapter evaluates

$$\boldsymbol{v}^n = \mathcal{V}_\mu(q^n) = \boldsymbol{v}_\mu^{\mathrm{loc}}(q^n) + \mathcal{B}_\mu \mathcal{A}_\mu^{-1} \mathcal{S}_\mu(q^n), \tag{18}$$

and, for spatial transport, constructs a midpoint departure displacement,

$$\begin{aligned} \boldsymbol{y}^{n+\frac{1}{2}} &= \mathfrak{b}_\mu\big(\boldsymbol{x} - \tfrac{\Delta t}{2}\boldsymbol{v}^n(\boldsymbol{x})\big), \quad \boldsymbol{v}^{n+\frac{1}{2}} = \mathcal{I}[\boldsymbol{v}^n](\boldsymbol{y}^{n+\frac{1}{2}}), \\ \boldsymbol{d}^n &= -\Delta t\, \boldsymbol{v}^{n+\frac{1}{2}}, \qquad\qquad \boldsymbol{\Xi}^n = \mathfrak{b}_\mu(\boldsymbol{x} + \boldsymbol{d}^n). \end{aligned} \tag{19}$$

Here $\mathcal{I}$ interpolates grid values at off-grid coordinates, and $\mathfrak{b}_\mu$ applies the coordinate boundary rule. The field $\boldsymbol{v}^{n+\frac{1}{2}}$ is the current $\boldsymbol{v}^n$ sampled at $\boldsymbol{y}^{n+\frac{1}{2}}$. Vlasov–Poisson instead uses a symmetric phase-space backtrace consisting of a half-step trace in position, a full-step trace in velocity, and a second half-step trace in position; the exact construction is given in Appendix D. Periodic indices are wrapped, bounded indices are clamped, and outflow interpolation contributions are masked as specified there. The normalized neural input concatenates the state, carrier, static context, and normalized time step,

$$x_\mu^n = \mathrm{cat}[\mathcal{N}_{q,\mu}(q^n), \mathcal{N}_{v,\mu}(\boldsymbol{v}^n), c_\mu, \Delta t/\Delta t_\mu]. \tag{20}$$

Here Δt_μ is the reference frame spacing for system μ, and c_μ denotes the prescribed static context.

4.2 Multiscale mixing of centered and routed reads

Our four-scale encoder–decoder uses skip connections and local depthwise blocks (Ronneberger et al., 2015; Liu et al., 2022). Its encoder produces features s_ℓ. The displacement is average-pooled to each scale to sample the same feature tensor,

$$r_\ell(\boldsymbol{x}) = \mathcal{I}_\ell[s_\ell](\mathfrak{b}_{\mu,\ell}(\boldsymbol{x} + \boldsymbol{d}_\ell(\boldsymbol{x}))), \qquad \ell = 0, \dots, 3. \tag{21}$$

At each target grid point, $s_\ell(\boldsymbol{x})$ provides the local feature vector and $r_\ell(\boldsymbol{x})$ provides the vector interpolated from the departure location. The output grid stays fixed. At the coarsest scale, the decoder receives the two reads directly; at each finer scale, it combines them with the upsampled decoder feature from the next coarser scale. Both reads share one encoder; layer widths and block details are given in Appendix E.

4.3 State increment, rollout, and training

The finest decoder feature z_0 is mapped by the learned output head H_θ to predict the complete state increment $\widehat{q}^{n+1} - \widehat{q}^n$, including transport and the remaining dynamics:

$$\widehat{q}^{n+1} = \widehat{q}^n + \Pi_\mu(\sigma_{\Delta q,\mu} H_\theta(z_0)), \tag{22}$$

where $\sigma_{\Delta q,\mu}$ is the fixed increment scale computed from training data and Π_μ enforces a known integral constraint when applicable. At the next step, Eqs. (18)–(19) are recomputed from $\widehat{q}^{n+1}$. Training backpropagates through a $K = 4$ rollout using

$$\mathcal{L}(\theta) = Z_K^{-1} \sum_{k=1}^{K} \rho^{k-1} \big(\ell_{\mathrm{state}}^{(k)} + \lambda_\nabla \ell_\nabla^{(k)} + \lambda_{\mathrm{spec}} \ell_{\mathrm{spec}}^{(k)}\big), \tag{23}$$

where $Z_K = \sum_{k=1}^{K} \rho^{k-1}$. Loss definitions and hyperparameters are given in Appendix F. The diagnostic solve costs $\mathcal{O}(N \log N)$ for our spectral adapters; the local neural core and routed reads are linear in the number of grid points, and the equation adapter adds no trainable parameters.

5 Experiments

We evaluate long-horizon predictive accuracy and isolate the contribution of equation-specified routing across five PDE systems.

5.1 Benchmarks and protocol

We evaluate Kolmogorov flow, dissipative SQG, Vlasov–Poisson, 3D Keller–Segel, and 3D porous convection. These systems span periodic and bounded domains, physical space and phase space, and five distinct diagnostic relations for computing the carrier (Appendix C).

The baselines cover four spatial interaction mechanisms. ReViT (Wei et al., 2026) uses local canonical frames for rotationally equivariant attention. RieszNO (Liu et al., 2026) combines directional Riesz features with spectral mixing. NIPS-AR (Liu & Yu, 2025), our autoregressive adaptation of NIPS, combines linear attention with channel-independent Fourier kernels. Flowers (Muser et al., 2026) learns multihead displacements for feature resampling, making it the closest coordinate-warp comparison to GRNO's equation-specified read coordinates.

Table 1: **Long-horizon predictive accuracy.** Final relative L^2 error ($\times 10^{-2}$) at $H = 40$. Entries are mean$\pm$standard deviation over three training seeds. Lower is better; bold marks the lowest mean.

Model	Kolmogorov	SQG	Vlasov–Poisson	Keller–Segel	Porous convection
ReViT (Wei et al., 2026)	27.84 ± 1.18	3.25 ± 0.03	4.86 ± 0.90	1.23 ± 0.02	19.35 ± 1.27
Flowers (Muser et al., 2026)	14.54 ± 5.39	13.12 ± 0.50	0.30 ± 0.01	$\mathbf{0.32 \pm 0.09}$	16.51 ± 2.71
NIPS-AR (Liu & Yu, 2025)	134.25 ± 12.96	51.65 ± 0.06	19.33 ± 6.10	1.46 ± 0.39	69.33 ± 32.39
RieszNO (Liu et al., 2026)	39.01 ± 0.48	8.07 ± 0.63	0.49 ± 0.05	0.34 ± 0.01	34.84 ± 1.77
GRNO (ours)	$\mathbf{7.80 \pm 0.56}$	$\mathbf{2.34 \pm 0.24}$	$\mathbf{0.16 \pm 0.01}$	0.37 ± 0.02	$\mathbf{3.34 \pm 0.16}$

Table 2: **Fixed-weight route interventions.** Final relative L^2 error ($\times 10^{-2}$) at $H = 40$. Entries are mean$\pm$sample standard deviation across the three seed means.

System	Correct	Zero	Reverse	Shift
Kolmogorov	$\mathbf{7.80 \pm 0.56}$	210.18 ± 0.17	196.34 ± 0.21	130.67 ± 0.63
SQG	$\mathbf{2.34 \pm 0.24}$	11.77 ± 0.36	24.72 ± 0.27	27.58 ± 0.31
Vlasov–Poisson	$\mathbf{0.16 \pm 0.01}$	6.71 ± 0.09	19.75 ± 0.21	20.31 ± 0.06
Keller–Segel	$\mathbf{0.37 \pm 0.02}$	0.38 ± 0.02	0.39 ± 0.01	0.38 ± 0.01
Porous convection	$\mathbf{3.34 \pm 0.16}$	41.08 ± 0.91	47.76 ± 1.17	44.42 ± 1.21

Baseline comparisons use matched splits, normalization, training windows, loss, optimizer updates, and checkpoint selection. Models train through $K = 4$ steps and are tested by free $H = 40$ rollouts from prescribed exact frames, with each prediction reused as the next input. We average errors over 12 prescribed test-trajectory windows within each seed and report mean$\pm$standard deviation over seeds $\{0, 1, 2\}$ for the baseline comparison. Route interventions use seeds $\{0, 1, 2\}$, whereas independently trained ablations use seed 0. Full protocols and dimensional or autoregressive adaptations are specified in Appendices F–G.

5.2 Predictive utility

GRNO achieves the lowest mean final error on four of five systems (Table 1), reducing error relative to the strongest baseline by approximately 46% on Kolmogorov flow, 28% on SQG, 47% on Vlasov–Poisson, and 80% on porous convection. The strongest comparator is ReViT on SQG and Flowers on the other three tasks. On Keller–Segel, Flowers, RieszNO, and GRNO obtain errors of 0.32, 0.34, and 0.37 in the table's units.

Figure 2a shows lower errors across substantial portions of these four rollouts, extending beyond the four supervised steps to the 40-step test horizon. On Kolmogorov flow and porous convection, GRNO also has smaller standard deviations across training seeds than Flowers: 0.56 versus 5.39 and 0.16 versus 2.71, respectively, in the units of Table 1.

Figure 2b shows that GRNO recovers the alternating vorticity regions and filaments in Kolmogorov flow and the concentration patterns in porous convection, with localized discrepancies. Appendix J.1 compares both models' spatial errors across ten rollout horizons. The following experiments isolate the contribution of equation-specified routing.

5.3 Does the fitted model use the route?

The accuracy gains above raise a direct mechanistic question: do the fitted GRNOs actually exploit the equation-specified read coordinates, or could comparable results be achieved while ignoring the route? We answer this by keeping each checkpoint and all other operations unchanged, and modifying only the departure displacement in Eq. (19): *zero* sets $\boldsymbol{d}^n = 0$, *reverse* uses $-\boldsymbol{d}^n$, and *shift* spatially misaligns the displacement with the current state.

Each intervention increases mean final error by at least $16.7\times$ on Kolmogorov flow, $5.0\times$ on SQG, $41.0\times$ on Vlasov–Poisson, and $13.0\times$ on porous convection, while Keller–Segel changes little (Table 2). Figure 2c–d shows degradation factors and rollout profiles; Appendix J.2 shows the corresponding fields and pointwise errors. The fitted predictors depend strongly on route direction and alignment in four systems, and weakly in Keller–Segel.

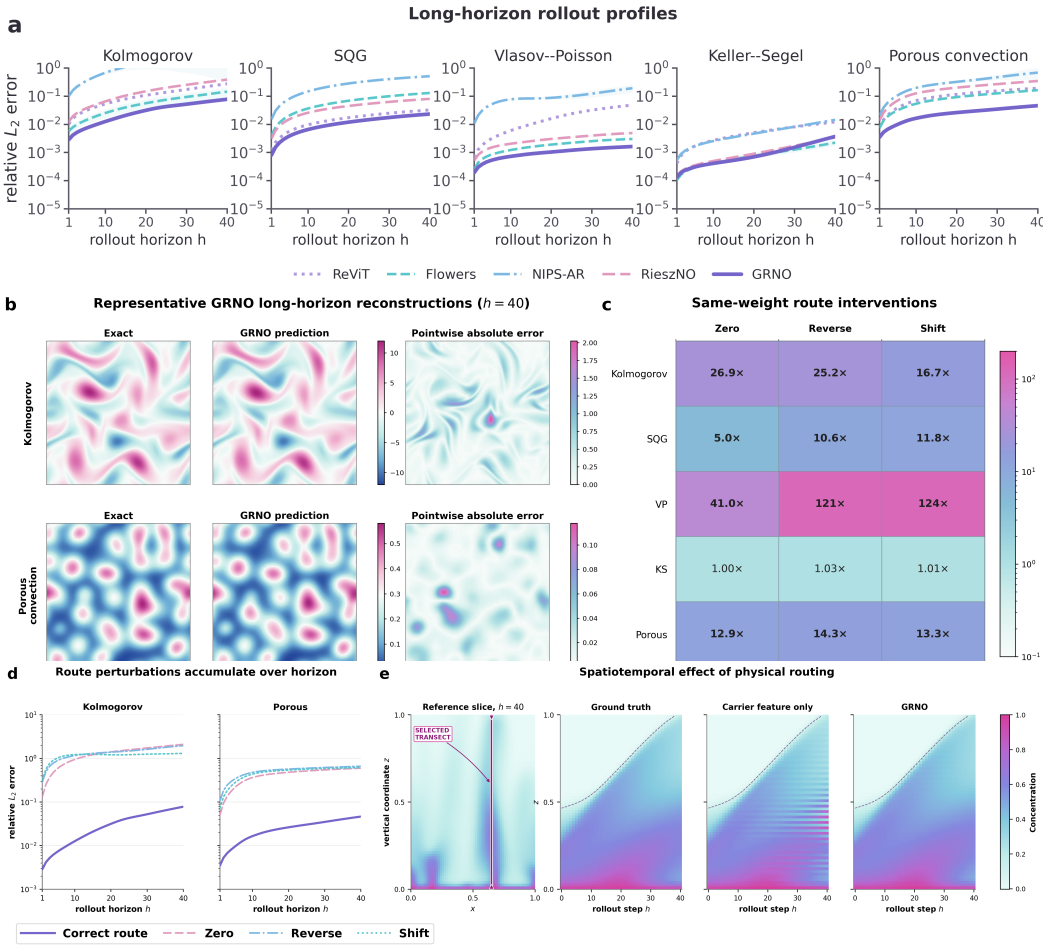


Figure 2: **Long-horizon GRNO accuracy and system-dependent route use. a**, Relative L^2 error over 40-step free rollouts. GRNO gives the lowest error on four systems and remains competitive on Keller–Segel. **b**, Reference fields, GRNO predictions and absolute errors at $h = 40$ for Kolmogorov flow and porous convection; pointwise errors remain localized around the recovered structures. **c**, Ratios of final error under the zero, reverse, and shift interventions relative to the correct route, with network weights fixed. In four systems, intervention errors are 5.0–124 times those obtained with the correct route; in Keller–Segel, the ratios remain near one. **d**, The error differences appear early and persist throughout the rollout. **e**, Space–time concentration along the selected porous-convection transect for the reference, *Carrier feature only*, and GRNO. In this selected window, *Carrier feature only* develops temporal banding, whereas GRNO follows the advancing front.

5.4 Does equation-specified routing improve prediction?

Route reliance in a fixed checkpoint does not by itself establish that equation-specified routing improves the learned predictor. We therefore retrain all variants independently to test whether equation-specified read coordinates reduce error. A 2×2 ablation independently enables or disables carrier features and equation-specified latent routing in the shared backbone. *Centered only* uses neither component, *Carrier feature only* and *Route only* use one component each, and GRNO uses both. This separates the value of supplying the carrier from the value of using it to select read coordinates.

With carrier features present, adding equation-specified routing reduces mean final error by 16.2%, 37.5%, 42.6%, 1.8%, and 46.9% on Kolmogorov flow, SQG, Vlasov–Poisson, Keller–Segel, and porous convection, respectively (Table 3). GRNO achieves the lowest mean on four systems, whereas *Centered only* leads on Keller–Segel. On SQG and Vlasov–Poisson, either *Carrier feature only* or *Route only* improves on *Centered only*, and GRNO improves further. On Kolmogorov flow, the

Table 3: **Independently trained ablations and route controls.** Mean final relative L^2 error ($\times 10^{-2}$) at $H = 40$, for seed 0 and 12 prescribed test windows per system. The first four rows form the 2×2 factorial. Bold marks the lowest mean among evaluated variants.

Variant	Kolmogorov	SQG	Vlasov–Poisson	Keller–Segel	Porous convection
Centered only	8.1659	4.2917	0.3158	**0.3563**	6.5352
Carrier feature only	9.3106	3.7392	0.2849	0.3788	6.2905
Route only	9.0460	3.3361	0.2256	0.3710	4.8331
Full (GRNO)	**7.8023**	**2.3354**	**0.1635**	0.3718	**3.3426**
Learned route	8.7553	2.8387	0.1782	0.3892	3.4775
Shifted equation-specified route	9.5599	3.3667	0.1913	0.3791	4.0076
State advection + residual	11.3067	6.2341	0.6775	0.5907	9.3214

improvement over *Centered only* emerges when carrier features and routing are combined. Figure 2e compares *Carrier feature only* with GRNO along the same porous-convection transect.

We next test how the read coordinates are constructed. The *Learned route* control replaces the equation-specified departure displacement with a displacement head trained using the forecasting loss. The *Shifted equation-specified route* control spatially misaligns the equation-specified departure displacement with the current state. Both retain carrier features and are independently trained using their respective read coordinates. GRNO has lower mean errors than both controls on all five systems.

Finally, we compare routing the physical state with routing latent features. The *State advection + residual* control uses the same equation-specified read coordinates to resample the physical state and then applies a centered network to learn the remaining update, without latent routing. GRNO reduces final error relative to this control by 31.0%, 62.5%, 75.9%, 37.1%, and 64.1% on Kolmogorov flow, SQG, Vlasov–Poisson, Keller–Segel, and porous convection, respectively. *State advection + residual* has the highest mean error among the evaluated variants on all five systems. These ablations indicate that the gain depends on using the equation-specified route for latent feature reads while learning the complete finite-time update.

6 Discussion

GRNO uses the diagnostic carrier both as an input feature and to determine where latent features are read. The diagnostic relation determines the diagnostic field, from which the carrier and departure map are computed. The values of the departure map are the read coordinates used by the neural operator. The equation therefore specifies the read coordinates, while the network learns the complete finite-time update. This interface applies across two-dimensional flow, phase-space dynamics, and three-dimensional transport.

Fixed-weight route interventions show that four fitted predictors depend strongly on route direction and alignment. In the independently trained controls, GRNO has lower mean error than *Learned route* and *Shifted equation-specified route* on all five systems. It also has lower mean error than *State advection + residual* on every benchmark. These results indicate that performance depends on both supplying the carrier and using it to determine latent read coordinates.

Dependence on routing varies by system. Keller–Segel shows little sensitivity to fixed-weight route interventions. In the ablations, GRNO improves only slightly on *Carrier feature only*, and *Centered only* has the lowest error. In the other four systems, the trained models are sensitive to route interventions, and GRNO has the lowest mean error in the ablations. These results support equation-specified latent routing when the route materially affects finite-time prediction.

AI use statement

The authors used generative AI tools to help polish the text, check the mathematical arguments, give feedback on the experimental design and controls, implement and debug the model code, and write the plotting code based on the model code from their trained models and the requirements they gave.

The authors conducted all numerical experiments and reviewed the AI-assisted code, text, and figures. All reported numerical results were obtained from these experiments. The authors take responsibility for the final content of this work.

Ethics statement

This work uses synthetic numerical PDE data and does not involve human subjects, personal data, or sensitive datasets.

Reproducibility Statement

All equations, datasets, adapters, boundary rules, model configurations, losses, training budgets, and evaluation metrics are specified in the appendices. The code will be shared after the paper is accepted.

References

Yohai Bar-Sinai, Stephan Hoyer, Jason Hickey, and Michael P. Brenner. Learning data-driven discretizations for partial differential equations. *Proceedings of the National Academy of Sciences*, 116(31):15344–15349, 2019. doi: 10.1073/pnas.1814058116.

Boris Bonev, Thorsten Kurth, Christian Hundt, Jaideep Pathak, Maximilian Baust, Karthik Kashinath, and Anima Anandkumar. Spherical Fourier neural operators: Learning stable dynamics on the sphere. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2806–2823. PMLR, 2023. URL https://proceedings.mlr.press/v202/bonev23a.html.

Nicolas Boullé, Christopher J. Earls, and Alex Townsend. Data-driven discovery of green's functions with human-understandable deep learning. *Scientific Reports*, 12:4824, 2022a. doi: 10.1038/s41598-022-08745-5.

Nicolas Boullé, Seick Kim, Tianyi Shi, and Alex Townsend. Learning green's functions associated with time-dependent partial differential equations. *Journal of Machine Learning Research*, 23 (218):1–34, 2022b. URL https://www.jmlr.org/papers/v23/22-0433.html.

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

Shuhao Cao. Choose a transformer: Fourier or Galerkin. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/d0921d442ee91b896ad95059d13df618-Abstract.html.

C. Z. Cheng and Georg Knorr. The integration of the Vlasov equation in configuration space. *Journal of Computational Physics*, 22(3):330–351, 1976. doi: 10.1016/0021-9991(76)90053-X.

S. M. Cox and P. C. Matthews. Exponential time differencing for stiff systems. *Journal of Computational Physics*, 176(2):430–455, 2002. doi: 10.1006/jcph.2002.6995.

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 764–773, 2017. URL https://openaccess.thecvf.com/content_iccv_2017/html/Dai_Deformable_Convolutional_Networks_ICCV_2017_paper.html.

Zexi Fan, Yan Sun, Shihao Yang, and Yiping Lu. Physics-informed inference time scaling for solving high-dimensional partial differential equations. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/cf8353a0efc3644454a5591a7cb55c17-Abstract-Conference.html.

John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. Efficient token mixing for transformers via adaptive Fourier neural operators. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=EXHG-A3jlM.

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

Jacob Helwig, Xuan Zhang, Cong Fu, Jerry Kurtin, Stephan Wojtowytsch, and Shuiwang Ji. Group equivariant Fourier neural operators for partial differential equations. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 12907–12930. PMLR, 2023. URL https://proceedings.mlr.press/v202/helwig23a.html.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/hash/33ceb07bf4eeb3da587e268d663aba1a-Abstract.html.

Dmitrii Kochkov, Jamie A. Smith, Ayya Alieva, Qing Wang, Michael P. Brenner, and Stephan Hoyer. Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21):e2101784118, 2021. doi: 10.1073/pnas.2101784118.

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. Neural operator: Graph kernel network for partial differential equations, 2020a. URL https://arxiv.org/abs/2003.03485.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Andrew Stuart, Kaushik Bhattacharya, and Anima Anandkumar. Multipole graph neural operator for parametric partial differential equations. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6755–6766, 2020b. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/4b21cf96d4cf612f239a6c322b10c8fe-Abstract.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, Daniel Zhengyu Huang, Burigede Liu, and Anima Anandkumar. Fourier neural operator with learned deformations for PDEs on general geometries. *Journal of Machine Learning Research*, 24(388):1–26, 2023a. URL https://jmlr.org/papers/v24/23-0064.html.

Zongyi Li, Nikola Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, and Animashree Anandkumar. Geometry-informed neural operator for large-scale 3D PDEs. In *Advances in Neural Information Processing Systems*, volume 36, 2023b. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/70518ea42831f02afc3a2828993935ad-Abstract-Conference.html.

Phillip Lippe, Bas Veeling, Paris Perdikaris, Richard 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.

Bjoern List, Li-Wei Chen, Kartik Bali, and Nils Thuerey. Differentiability in unrolled training of neural physics simulators on transient dynamics. *Computer Methods in Applied Mechanics and Engineering*, 433:117441, 2025. doi: 10.1016/j.cma.2024.117441.

Ning Liu and Yue Yu. Neural interpretable PDEs: Harmonizing Fourier insights with attention for scalable and interpretable physics discovery. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 40085–40099. PMLR, 2025. URL https://proceedings.mlr.press/v267/liu25cp.html.

Shouyi Liu, Xiaokang Yang, and Yuntian Chen. Riesz neural operator for solving partial differential equations. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/ed3fea9033a80fea1376299fa7863f4a-Abstract-Conference.html.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11976–11986, 2022. doi: 10.1109/CVPR52688.2022.01167.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

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.

Andrew J. Majda and Andrea L. Bertozzi. *Vorticity and Incompressible Flow*. Number 27 in Cambridge Texts in Applied Mathematics. Cambridge University Press, 2001. doi: 10.1017/CBO9780511613203.

HA Melchers and JHM Prins. Neural green's operators for parametric partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 455:118893, 2026.

Till Muser, Alexandra Spitzer, Matti Lassas, Maarten V. de Hoop, and Ivan Dokmanić. Flowers: A warp drive for neural PDE solvers. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=TPpUKH8fuQ.

Carlos A Pereira, Stéphane Gaudreault, Valentin Dallerit, Christopher Subich, Shoyon Panday, Siqi Wei, Sasa Zhang, Siddharth Rout, Eldad Haber, Raymond J Spiteri, et al. Learning to advect: A neural semi-Lagrangian architecture for weather forecasting. *arXiv preprint arXiv:2601.21151*, 2026.

Md Ashiqur Rahman, Zachary E. Ross, and Kamyar Azizzadenesheli. U-NO: U-shaped neural operators. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=j3oQF9coJd.

Bogdan Raonić, Roberto Molinaro, Tim De Ryck, Tobias Rohner, Francesca Bartolucci, Rima Alaifari, Siddhartha Mishra, and Emmanuel de Bézenac. Convolutional neural operators for robust and accurate learning of PDEs. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/f3c1951b34f7f55ffaecada7fde6bd5a-Abstract-Conference.html.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, volume 9351 of *Lecture Notes in Computer Science*, pp. 234–241. Springer, 2015. doi: 10.1007/978-3-319-24574-4_28.

Chenhao Si and Ming Yan. RD-JEPA: Predictive latent pretraining for few-trajectory transfer across Reaction-Diffusion equations, 2026. URL https://arxiv.org/abs/2609.29403.

Andrew Staniforth and Jean Côté. Semi-lagrangian integration schemes for atmospheric models—a review. *Monthly Weather Review*, 119(9):2206–2223, 1991. doi: 10.1175/1520-0493(1991)119⟨2206:SLISFA⟩2.0.CO;2.

Gilbert Strang. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, 5(3):506–517, 1968. doi: 10.1137/0705041.

Yingjie Tan, Quanming Yao, and Yaqing Wang. DGNet: Discrete green networks for data-efficient learning of spatiotemporal PDEs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2603.01762.

Hao Tang, Hao Chen, and Chao Li. Generalized spherical neural operators: Green's function formulation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://iclr.cc/virtual/2026/poster/10008942.

Kiwon Um, Robert Brand, Yun Fei, Philipp Holl, and Nils Thuerey. Solver-in-the-loop: Learning from differentiable physics to interact with iterative PDE-solvers. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/43e4e6a6f341e00671e123714de019a8-Abstract.html.

Haixin Wang, Jiaxin Li, Anubhav Dwivedi, Kentaro Hara, and Tailin Wu. BENO: Boundary-embedded neural operators for elliptic PDEs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ZZTkLDRmkg.

Sifan Wang, Jacob H. Seidman, Shyam Sankaran, Hanwen Wang, George J. Pappas, and Paris Perdikaris. CViT: Continuous vision transformer for operator learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=cRnCcuLvyr.

Hao Wei, Bjoern List, and Nils Thuerey. ReViT: Rotational-equivariant vision transformers for neural PDE solvers. In *Forty-Third International Conference on Machine Learning*, 2026. URL https://tum-pbs.github.io/revit-web/. Oral presentation.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for PDEs on general geometries. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 53681–53705. PMLR, 2024. URL https://proceedings.mlr.press/v235/wu24r.html.

A PROOFS FOR THE ROUTE-SENSITIVITY BOUNDS

Assume that, in a neighborhood of the reference trajectory,

$$\|f \circ (\boldsymbol{\Xi}_n[p] + \boldsymbol{\eta})\|_2 \le B_n \|f\|_2, \tag{24}$$

$$\|\boldsymbol{\Xi}_n[p] - \boldsymbol{\Xi}_n[q]\|_2 \le L_{\Xi,n} \|p - q\|_2, \tag{25}$$

$$\|\mathcal{C}_n[p] - \mathcal{C}_n[q]\|_2 \le L_{\mathcal{C},n} \|p - q\|_2. \tag{26}$$

Let $e_n = \widetilde{q}^{\,n} - q^n$ and M_n bound $\|\nabla q^n\|_{L^\infty}$ along the segment joining the two departure maps. Add and subtract $q^n \circ (\boldsymbol{\Xi}_n[\widetilde{q}^{\,n}] + \boldsymbol{\eta}_n)$ in Eq. (15). The first resulting term is bounded by $B_n \|e_n\|_2$; the mean-value theorem and the second inequality bound the coordinate change by $M_n(L_{\Xi,n}\|e_n\|_2 + \|\boldsymbol{\eta}_n\|_2)$; the final inequality bounds the remainder difference. Summing the three bounds gives the one-step recursion in Eq. (16). Repeated substitution proves Eq. (17). For a measure-preserving continuum characteristic, change of variables gives $B_n = 1$; bounded-domain handling and discrete interpolation are covered by retaining B_n.

B SUPPLEMENTARY GUIDE

The supplement is organized around the claims made in the main paper. Appendix A proves the route-sensitivity bounds. Appendix C instantiates the diagnostically routed PDE class and specifies every dataset. Appendix D gives the parameter-free equation adapter, traces, and boundary rules. Appendix E records the shared trainable architecture, and Appendix F fixes the training and evaluation contract. Appendix G documents baseline provenance and adaptations. Appendices H–I define the reported metrics and describe both the fixed-weight interventions and independently trained controls. Appendix J provides forecasting and route-intervention visualizations.

Table 4: **Claim–evidence map.** This table separates evidence for predictive utility from evidence for the proposed routing mechanism.

Question	Control	Evidence
Does the complete model predict well?	Independently trained models share data, loss, update budget, checkpoint rule, and evaluator.	Main Table 1
Does the trained model use the route?	Keep weights and carrier features fixed; change only the departure displacement.	Figure 2 and Appendix I
Does equation-specified routing improve prediction after training?	Independently train models with different carrier features, coordinate constructions, and routing targets.	Main Table 3 and Appendix I

C PDE INSTANTIATIONS AND DATASETS

C.1 VARIANTS OF THE COMMON TEMPLATE

Eq. (9)–(11) use an advective notation for readability. When the evolved quantity is a density, the corresponding conservative form is

$$\partial_t q + \nabla \cdot \big(\boldsymbol{v}_\mu[q] q\big) = \mathcal{R}_\mu(q, \psi_\mu[q]). \tag{27}$$

For a phase-space state $f(\boldsymbol{x}, \boldsymbol{v}, t)$, the same idea applies on the product domain:

$$\partial_t f + \boldsymbol{a}_x[f] \cdot \nabla_x f + \boldsymbol{a}_v[f] \cdot \nabla_v f = \mathcal{R}(f), \tag{28}$$

where the diagnostic solve may determine either component of the phase-space carrier $(\boldsymbol{a}_x, \boldsymbol{a}_v)$; Table 5 gives the exact instantiation for every task.

C.2 EQUATIONS AND COEFFICIENTS

Kolmogorov flow. On $[0, 2\pi)^2$, the reference trajectories solve

$$\partial_t \omega + \boldsymbol{u} \cdot \nabla \omega = \nu \Delta \omega - \alpha \omega - 4\cos(4x_2), \qquad \Delta \psi = \omega - \bar{\omega}, \quad \boldsymbol{u} = \nabla^\perp \psi, \tag{29}$$

Table 5: **Diagnostic carriers used in the experiments.** Each row lists the evolved state, diagnostic relation, diagnostic carrier that determines the read coordinates, and boundary rule.

System	Evolved state q	Diagnostic relation	Diagnostic carrier	Boundary rule
Kolmogorov	ω	$\Delta\psi = \omega - \bar{\omega}$	$\boldsymbol{u} = \nabla^{\perp}\psi$	periodic x, y
SQG	θ	$\psi = -(-\Delta)^{-1/2}\theta$	$\boldsymbol{u} = \nabla^{\perp}\psi$	periodic x, y
Vlasov–Poisson	$f(x, v)$	$\rho = \int f \, \mathrm{d}v$, $\partial_x E = 1 - \rho$	$(a_x, a_v) = (v, -E)$	periodic x; zero outflow in v
Keller–Segel	ρ	$(-\Delta + \lambda^2)c = \rho - \bar{\rho}$	$\boldsymbol{v} = \chi\nabla c$	periodic x, y, z
Porous convection	C	$\boldsymbol{u} = -\nabla p + C\boldsymbol{e}_z$, $\nabla\cdot\boldsymbol{u} = 0$	Darcy velocity $\boldsymbol{u}$	periodic x, y; bounded z

with $\nu = 10^{-2}$, $\alpha = 0.1$, and frame spacing $\Delta t = 0.1$. The pseudo-spectral reference solver uses $2/3$ de-aliasing, ETDRK2 (Cox & Matthews, 2002), and 20 internal steps per stored frame. Initial vorticity is obtained by Fourier filtering independent standard Gaussian grid values to $\|\boldsymbol{k}\|_\infty \le 8$, removing the zero mode, and normalizing the root-mean-square amplitude to one. The field is advanced for 500 stored-frame intervals before collection.

Dissipative surface quasi-geostrophy. On the same periodic domain,

$$\partial_t\theta + \boldsymbol{u}\cdot\nabla\theta = \kappa\Delta\theta - \gamma(-\Delta)^{-1}\theta + f, \qquad \psi = -(-\Delta)^{-1/2}\theta, \quad \boldsymbol{u} = \nabla^{\perp}\psi, \tag{30}$$

with $\kappa = 0.008$, $\gamma = 0.15$, and $\Delta t = 0.1$. Each trajectory has a frozen random forcing supported on $3 \le |\boldsymbol{k}| \le 4$ with RMS amplitude 0.10. The de-aliased ETDRK2 solver uses 40 internal steps per stored frame. Initial conditions use the same Gaussian filtering procedure as Kolmogorov flow, with $\|\boldsymbol{k}\|_\infty \le 8$, root-mean-square amplitude 0.5, and a 200-frame warm-up.

Vlasov–Poisson. The phase-space domain is $x \in [0, 4\pi)$ and $v \in [-6, 6)$, with

$$\partial_t f + v\,\partial_x f - E\,\partial_v f = 0, \qquad \rho(x) = \int_{-6}^{6} f(x, v)\,\mathrm{d}v, \qquad \partial_x E = 1 - \rho, \tag{31}$$

and $\Delta t = 0.1$. A spectral Strang solver (Strang, 1968; Cheng & Knorr, 1976) applies a half x translation, a full v translation, and a second half x translation, using five internal steps per stored frame. The reference generator implements these translations by FFT on the truncated velocity interval; GRNO instead assigns zero to routed reads that leave $[-6, 6)$. The sampled initial distributions have negligible support at that boundary. Each trajectory independently selects a Landau or two-stream initial condition with probability $1/2$. Writing $M_\sigma(v) = \exp[-v^2/(2\sigma^2)]/(\sqrt{2\pi}\sigma)$, the unnormalized fields are

$$f_{\mathrm{L}}(x, v) = M_\sigma(v)\left[1 + \alpha\cos(x/2 + \phi) + \tfrac{1}{4}\alpha\cos(x - 2\phi)\right], \tag{32}$$

$$f_{\mathrm{T}}(x, v) = \tfrac{1}{2}\left[M_\sigma(v - v_0) + M_\sigma(v + v_0)\right]\left[1 + \alpha\cos(x/2 + \phi) + \tfrac{1}{5}\alpha\cos(3x/2 - \phi)\right]. \tag{33}$$

Parameters are sampled independently and uniformly: $\phi \in [0, 2\pi)$; $\sigma \in [0.8, 1.2]$ and $\alpha \in [0.05, 0.18]$ for Landau; $v_0 \in [1.6, 2.4]$, $\sigma \in [0.35, 0.60]$, and $\alpha \in [0.03, 0.10]$ for two-stream. Each field is divided by its discrete mean density, $(\Delta x\Delta v/L_x)\sum_{i,j} f(x_i, v_j)$, so that the spatially averaged density is one.

Screened Keller–Segel. On $[0, 2\pi)^3$,

$$\partial_t\rho = D\Delta\rho - \nabla\cdot(\rho\boldsymbol{v}), \qquad \boldsymbol{v} = \chi\nabla c, \qquad (-\Delta + \lambda^2)c = \rho - \bar{\rho}, \tag{34}$$

with $D = 0.10$, $\chi = 0.30$, $\lambda = 1$, and $\Delta t = 0.25$. The $2/3$-de-aliased Fourier ETD2 solver uses eight internal steps per stored frame. Initial densities have the form $\rho = \exp(0.2\xi)$, where ξ is a unit-variance random field supported on $1 \le |\boldsymbol{k}| \le 3$; each sample is rescaled to mean one.

Buoyancy-driven porous convection. On the unit cube, with periodic horizontal coordinates and bounded vertical coordinate z,

$$\partial_t C + \nabla\cdot(\boldsymbol{u}C) = Ra_D^{-1}\Delta C, \qquad \boldsymbol{u} = -\nabla p + C\boldsymbol{e}_z, \qquad \nabla\cdot\boldsymbol{u} = 0, \tag{35}$$

with $Ra_D = 1000$ and $\Delta t = 0.1$. We impose $C = 1$ at the top, $\partial_z C = 0$ at the bottom, and impermeability $u_z = 0$ at both vertical boundaries. Pressure is fixed to zero mean. The finite-volume

Table 6: **Dataset contract.** Shapes list trajectories × stored frames; every frame also contains the grid shown in the second column. Splits are fixed before training.

System	Grid	Train	Validation	Test	Stored Δt
Kolmogorov	128^2	64×51	8×101	12×101	0.10
SQG	128^2	64×51	8×101	12×101	0.10
Vlasov–Poisson	$128_x \times 128_v$	64×101	8×121	12×161	0.10
Keller–Segel	64^3	64×51	8×101	12×101	0.25
Porous convection	64^3	64×101	8×101	12×101	0.10

reference solver combines MUSCL advection, SSP/RK2 time stepping, an exact mixed-boundary diffusion semigroup, and Strang splitting; its internal step is at most 0.02 and also satisfies an advective CFL bound of 0.35. Trajectories are generated on 128^3 and volume-averaged to 64^3. The vertical coordinate increases downward from the top boundary, $z = 0$. Before the warm-up, concentration is initialized as

$$C_0(x, y, z) = \mathrm{clip}_{[0,1]}\left[b(z) + 0.08 b(z)(1 - b(z))\xi(x, y)\right], \qquad b(z) = \mathrm{erfc}\left(\frac{z}{2\sqrt{Ra_D^{-1} t_0}}\right). \tag{36}$$

Here t_0 is sampled uniformly from $[0.9, 1.1]$. The horizontal field ξ is a sum of sine and cosine modes with integer wavevectors $1 \le |\boldsymbol{k}| \le 8$, using one representative of each pair $\{\boldsymbol{k}, -\boldsymbol{k}\}$. Its coefficients are independent standard Gaussian variables, and the field is normalized to unit root-mean-square amplitude. The initialized field is advanced for 0.2 before storage.

C.3 Dataset contract

Each training run records a trajectory/start-time manifest; a result is included in the matched table only after its manifest hash agrees with the task reference. Manifests log array shapes, coefficients, split membership, solver version, and file hashes; any run with a mismatched contract is excluded.

D Equation adapter, traces, and boundaries

The adapter is deterministic, has no trainable parameters, and is recomputed from the current predicted state at every rollout step.

D.1 Diagnostic solves

Periodic active scalars. For Kolmogorov flow and SQG, the zero Fourier mode is fixed to zero and

$$\widehat{\psi}(\boldsymbol{k}) = -|\boldsymbol{k}|^{-r}\widehat{q}(\boldsymbol{k}), \qquad \widehat{\boldsymbol{v}}(\boldsymbol{k}) = (ik_2, -ik_1)\widehat{\psi}(\boldsymbol{k}), \tag{37}$$

with $r = 2$ for vorticity and $r = 1$ for SQG. This is the discrete Green multiplier for the respective periodic diagnostic relation.

Vlasov–Poisson. The adapter integrates f over velocity, removes the neutral Fourier mode, and solves Eq. (31) spectrally for $E(x)$. It returns the phase-space carrier $(v, -E(x))$ rather than treating E as a second evolved state.

Keller–Segel. The screened inverse is diagonal in Fourier space:

$$\widehat{c}(\boldsymbol{k}) = \frac{\widehat{\rho}(\boldsymbol{k})}{|\boldsymbol{k}|^2 + \lambda^2} \quad (\boldsymbol{k} \neq 0), \qquad \boldsymbol{v} = \chi \nabla c. \tag{38}$$

The mean source is removed as in Eq. (34).

Porous convection. The adapter first forms the buoyancy on vertical cell faces: interior faces use the average of their two adjacent concentrations, and the top and bottom face values are zero. The discrete divergence of this face buoyancy is the pressure right-hand side. The pressure equation uses the finite-volume Laplacian, diagonalized by Fourier transforms in the periodic horizontal directions and a cosine transform vertically; the zero mode is set to zero. Face velocities are computed from this buoyancy and the corresponding discrete pressure gradient, with zero normal velocity at the top and bottom. Averaging adjacent face velocities gives the cell-centered carrier supplied to the network.

D.2 Backward traces

For spatial transport, GRNO uses the midpoint rule from Eq. (19): the adapter first samples the carrier at $\boldsymbol{x} - \frac{1}{2}\Delta t\, \boldsymbol{v}(\boldsymbol{x})$ and then places the final read at $\boldsymbol{x} - \Delta t\, \boldsymbol{v}_{1/2}(\boldsymbol{x})$.

Vlasov–Poisson uses the symmetric phase-space trace matching the order of the characteristics:

$$\begin{aligned} x_{1/2} &= x - \tfrac{1}{2}\Delta t\, v, \\ E_{1/2} &= \mathcal{I}[E](x_{1/2}), \\ v_d &= v + \Delta t\, E_{1/2}, \\ x_d &= x_{1/2} - \tfrac{1}{2}\Delta t\, v_d. \end{aligned} \tag{39}$$

The routed read is taken at (x_d, v_d). The sign follows $\dot{x} = v$ and $\dot{v} = -E$ in Eq. (31) while integrating backward from the target point.

D.3 Interpolation and boundary rules

Table 7: **Boundary behavior of routed interpolation.** The rule is applied identically at every latent scale.

Coordinate type	Coordinate handling	Sample outside domain
Periodic	wrap modulo domain length	wrapped value
Bounded wall	clamp to the nearest grid index	value at the nearest boundary grid point
Phase-space outflow	mask each out-of-range index	zero contribution

Bilinear interpolation is used in two dimensions and trilinear interpolation in three dimensions. At scale ℓ, a departure displacement $d_{\ell,j}$ shifts the fractional grid index by $d_{\ell,j}/\Delta x_{\ell,j}$, where $\Delta x_{\ell,j} = L_j/N_{\ell,j}$. The full-resolution displacement is average-pooled to each scale and remains in physical units. Each neighboring index in the interpolation stencil is wrapped, clamped, or assigned zero interpolation weight according to Table 7.

E Shared GRNO architecture

The trainable core uses a common 2D architecture and its direct 3D counterpart. Within a dimension, the parameter names, tensor shapes, stage depths, and channel widths are identical across PDEs. Only the equation adapter, boundary metadata, context channel, and deterministic output constraint vary by task.

Inputs and normalization. Each task advances one scalar channel. The neural input has shape $B \times 5 \times N_y \times N_x$ in two dimensions and $B \times 6 \times N_z \times N_y \times N_x$ in three dimensions. For Vlasov–Poisson the two grid axes are (v, x). The channel order is state, carrier components in (x, y) or (x, y, z) order, one context channel, and $\Delta t/\Delta t_\mu$. Using fixed scales computed from training data,

$$\mathcal{N}_{q,\mu}(q) = \begin{cases} q/s_{q,\mu}, & \text{2D and phase space,} \\ (q - m_{q,\mu})/s_{q,\mu}, & \text{3D,} \end{cases} \qquad [\mathcal{N}_{v,\mu}(\boldsymbol{v})]_j = v_j/s_{v,j,\mu}. \tag{40}$$

Input scale denominators are floored at 10^{-7}. For Kolmogorov flow and SQG, state, velocity, forcing and increment scales are read from the training normalization file; forcing is divided by its scale, and the context channel is the normalized forcing pattern. For Vlasov–Poisson, state and increment scales

Table 8: **Dimension-specific GRNO configuration.** "Axial" in 3D denotes the average of three depthwise kernels, one per spatial axis.

Quantity	2D / phase space	3D
Input channels	5	6
Base width	48	24
Stage widths	$(48, 96, 144, 192)$	$(24, 48, 72, 96)$
Encoder depths	$(2, 2, 3, 4)$	$(2, 2, 3, 4)$
Decoder blocks per scale	1	1
Local depthwise kernel	5×5	axial 5-tap
Number of scales	4	4
Trainable parameters	1,534,945	391,561
Trainable adapter parameters	0	0

are root-mean-square values estimated from up to 512 training frame pairs using sampling seed 17. Its carrier scales are $v_{\max}$ and the root-mean-square electric field, and its context is zero. For the 3D tasks, state mean, state scale and increment scale are read from the saved training statistics. Each carrier component is scaled by its root-mean-square value over 24 evenly selected training states. Porous convection uses $c_\mu(z) = 2z/L_z - 1$ as context; Keller–Segel uses zero.

Multiscale fusion. The grids have side lengths $(128, 64, 32, 16)$ in 2D and $(64, 32, 16, 8)$ in 3D. Every encoder scale produces one feature tensor s_ℓ and its routed copy r_ℓ. The coarsest decoder input is $[s_3, r_3]$; at the remaining scales,

$$z_\ell = \mathrm{Dec}_\ell(W_\ell[\mathrm{Up}_\ell(z_{\ell+1}), s_\ell, r_\ell]), \qquad \ell = 2, 1, 0. \tag{41}$$

Brackets denote channel concatenation, and W_ℓ is a learned pointwise projection. Downsampling uses average pooling and a pointwise projection. Upsampling uses a pointwise projection followed by nearest-neighbor interpolation. A local residual block applies

$$u \longmapsto u + W_2\,\mathrm{GELU}\big(W_1\,\mathrm{LN}(\mathrm{DW}(u))\big). \tag{42}$$

Here DW is a depthwise spatial convolution, with the three axial outputs averaged in 3D. Layer normalization acts across channels independently at each grid point, with learned scale and shift and numerical floor 10^{-6}. The pointwise maps expand channels by a factor of two and project them back.

The increment head H_θ is a learned pointwise projection from the finest decoder feature z_0 to the normalized state increment and is initialized to zero. For tasks with an exactly conserved spatial integral, the predicted increment is projected to zero mean. Porous convection uses the identity projection because mass enters through its Dirichlet top boundary. These projections are fixed and contain no learned coefficients.

F Training and evaluation contract

F.1 Optimization and exposure

For a fixed task, all models receive the same window manifest. An epoch is therefore defined by data exposure and optimizer updates, not by how many micro-batches fit in memory. Three-dimensional models use gradient accumulation only to meet the A100 memory budget. Training uses AdamW (Loshchilov & Hutter, 2019); the learning-rate scheduler advances once per epoch. For validation, all admissible (trajectory, start-frame) pairs are ordered by trajectory and then start frame. The evaluator selects 64 evenly spaced pairs in 2D and 32 in 3D, or all pairs if fewer are available. Checkpoint selection minimizes relative L^2 averaged over these windows and all ten validation horizons.

F.2 Loss

Let $e_b^{(k)} = \widehat{q}_b^{n+k} - q_b^{n+k}$ for micro-batch sample b. Let B be the micro-batch size, N the number of grid points, and $s_{q,\mu}$ the fixed state scale defined in Appendix E. All reported tasks have one physical

Table 9: **Matched training and evaluation protocol.** Four 3D micro-batches are accumulated to obtain an effective batch size of four.

Setting	2D / phase space	3D
Epochs	100	100
Windows per epoch	2,048	512
Micro-batch / accumulation	$4/1$	$1/4$
Effective batch	4	4
Optimizer updates per epoch	512	128
Training rollout K	4	4
Validation / test horizon	$10/40$	$10/40$
Optimizer	AdamW	AdamW
Initial learning rate	3×10^{-4}	3×10^{-4}
Weight decay	10^{-4}	10^{-4}
Schedule	cosine to 1.5×10^{-5}	cosine to 1.5×10^{-5}
Gradient clipping	1.0	1.0
Precision	bfloat16 AMP	bfloat16 AMP
Seeds: baseline comparison	$\{0, 1, 2\}$	$\{0, 1, 2\}$
Seeds: fixed-weight route interventions	$\{0, 1, 2\}$	$\{0, 1, 2\}$
Seed: independently trained ablations	0	0

output channel. The state loss is

$$\ell_{\text{state}}^{(k)} = \frac{1}{BN} \sum_{b=1}^{B} \sum_{\boldsymbol{i}} \left| e_b^{(k)}(\boldsymbol{i}) / s_{q,\mu} \right|^2 . \tag{43}$$

Define $\langle a \rangle = (BN)^{-1} \sum_{b,\boldsymbol{i}} a_b(\boldsymbol{i})$. The gradient loss is

$$\ell_{\nabla}^{(k)} = \frac{\sum_j \langle |\delta_j e^{(k)}|^2 \rangle}{\max\{\sum_j \langle |\delta_j q^{n+k}|^2 \rangle, \varepsilon\}}. \tag{44}$$

The difference δ_j uses grid spacing L_j/N_j: centered differences with wraparound on periodic axes, centered interior differences and one-sided endpoint differences otherwise. Let $\mathcal{F}_{\mathcal{P}}$ transform only the periodic axes. The spectral loss averages per-sample ratios:

$$\ell_{\text{spec}}^{(k)} = \frac{1}{B} \sum_{b=1}^{B} \frac{\left\| |\mathcal{F}_{\mathcal{P}} \widehat{q}_b^{n+k}| - |\mathcal{F}_{\mathcal{P}} q_b^{n+k}| \right\|_2^2}{\max\{\|\mathcal{F}_{\mathcal{P}} q_b^{n+k}\|_2^2, \varepsilon\}}. \tag{45}$$

The spectral norm sums over every transformed coefficient and any remaining grid axis. The gradient and spectral terms use physical field values. The implementation uses $\varepsilon = 10^{-8}$, $\lambda_{\nabla} = 0.05$, $\lambda_{\text{spec}} = 0.02$, $\rho = 0.8$, and $Z_K = \sum_{k=1}^{K} \rho^{k-1}$ in Eq. (23). In 3D, gradients from four micro-batches are accumulated for each optimizer update. Gradients pass through all four rollout steps, including the recomputed diagnostic solves, traces, and interpolations.

F.3 Free autoregressive evaluation

For test trajectory $i \in \{0, \ldots, 11\}$, let s_i be the prescribed start frame. Evaluation uses

$$\widehat{q}_i^{\,0} = q_i^{s_i}, \qquad \widehat{q}_i^{\,h+1} = \mathcal{M}_\theta(\widehat{q}_i^{\,h}), \qquad h = 0, \ldots, 39, \tag{46}$$

and compares $\widehat{q}_i^{\,h}$ with $q_i^{s_i+h}$. The three 2D systems use $s_i = 0$. Porous convection uses $s_i = 20(i \bmod 4)$. Keller–Segel uses the integer start indices $(0, 5, 10, 16, 21, 27, 32, 38, 43, 49, 54, 60)$, obtained by evenly spacing twelve indices over the admissible interval $[0, 60]$. Errors are averaged over the twelve windows within each seed. Tables 1 and 2 report the mean and sample standard deviation across the three seed means. Table 3 reports seed 0.

Table 10: **Baseline roles and adaptations.** For dimensional or autoregressive adaptations, we retain each published model's interaction mechanism and modify only the input and output interfaces required by the common one-frame protocol. The table summarizes the evaluated implementations.

Model	Retained interaction mechanism	Protocol adaptation
ReViT	canonical local frames and rotationally equivariant transformer blocks	Physical-feature adapter constructs the vector/scalar layout from each task; one-frame residual decoder.
Flowers	learned multihead coordinate warps and latent resampling	The same state, context, boundary sampler, and autoregressive evaluator are used; read coordinates are learned from forecast data.
NIPS-AR	channel-independent Fourier kernel and linear attention	Released spatial block placed in a recurrent one-frame wrapper; reported as an autoregressive adaptation.
RieszNO / RNO-3D	Riesz directional representation plus spectral mixing	Released 2D construction; direct axial 3D extension is explicitly marked as a dimensional adaptation.

G Baseline provenance and matched adaptations

For every baseline reported in the matched comparison, the implementation uses the model-specific adaptation listed in Table 10. Its checkpoint and manifest are verified against the common array files, sampled windows, training-only statistics, $K = 4$ loss, optimizer schedule, effective batch, update count, validation selection, and $H = 40$ evaluator. Micro-batch size is adjusted only for memory.

Each experiment manifest records the source repository revision, architecture configuration, parameter count, peak allocated memory, and forward time for that run.

H Metrics and result records

For each test window, errors are evaluated in physical units. At rollout step h, the relative L^2 error is

$$E_{L^2}(h) = \frac{\|\widehat{q}_h - q_h\|_2}{\max\{\|q_h\|_2, \varepsilon\}}, \qquad \varepsilon = 10^{-8}. \tag{47}$$

The rollout mean is $H^{-1}\sum_{h=1}^{H} E_{L^2}(h)$. The reported spectral error is

$$E_{\mathrm{spec}}(h) = \left[\frac{\left\| |\mathcal{F}_{\mathcal{P}}\widehat{q}_h| - |\mathcal{F}_{\mathcal{P}} q_h| \right\|_2^2}{\max\{\|\mathcal{F}_{\mathcal{P}} q_h\|_2^2, \varepsilon\}}\right]^{1/2}. \tag{48}$$

Thus the per-window spectral metric is the square root of the spectral ratio used in training. Relative H^1 adds the discrete derivative energies to the squared field norms in numerator and denominator before taking the square root. Spatial correlation is computed after subtracting each field's spatial mean. Persistence uses $\widehat{q}_h = q_0$, and its skill score is

$$\mathrm{Skill}(h) = 1 - \frac{E_{L^2}^{\mathrm{model}}(h)}{\max\{E_{L^2}^{\mathrm{persistence}}(h), \varepsilon\}}. \tag{49}$$

Table 11: **Task-specific physical metrics recorded by the common evaluator.** These metrics are reported in addition to relative L^2 error.

Task	Additional records
Kolmogorov / SQG	correlation, spectral error, variance error, and kinetic-energy or scalar-variance error
Vlasov–Poisson	total-mass drift, negative-density fraction, charge-density error, and electric-energy error
Keller–Segel	mass drift, negative-mass fraction, peak and variance error, and screened free-energy error
Porous convection	anomaly error, plume-mask IoU, top dissolution-flux error, plume-depth error, scalar-dissipation error, variance error, and fractions below zero or above one

Errors are computed for each trajectory and rollout step. Within each seed, results are averaged over the 12 prescribed test windows. Tables 1 and 2 report the mean and sample standard deviation across the three seed means, whereas Table 3 reports seed 0.

I EXPANDED ROUTE-MECHANISM STUDY

I.1 WHAT EACH INTERVENTION CHANGES

Table 12: **Fixed-weight route interventions.** Network weights, construction of carrier features, centered branches, and prescribed test windows are held fixed.

Mode	Operation on displacement $\boldsymbol{d}$	Question tested
Correct	d from the equation adapter	reference computation
Zero	$d \leftarrow 0$	Is the carrier used as a coordinate rather than only as an input feature?
Reverse	$d \leftarrow -d$	Does route orientation matter?
Shift	deterministic spatial roll of d	Does alignment between route and current state matter?

At each rollout step, the carrier and departure displacement are recomputed from that run's current predicted state. The chosen intervention is applied to this new displacement before pooling to each feature scale and sampling the features. For the *Shift* intervention, the fixed offsets are $(\lfloor N_y/3 \rfloor, \lfloor N_x/5 \rfloor)$ in 2D and $(\lfloor N_z/5 \rfloor, \lfloor N_y/4 \rfloor, \lfloor N_x/3 \rfloor)$ in 3D. The displacement values continue to evolve with the predicted state at every rollout step.

Final intervention errors are reported in Table 2; Figure 2c–d shows the corresponding error ratios and rollout profiles.

I.2 INDEPENDENTLY TRAINED CONTROLS

All variants share the core architecture within each dimension. Disabling carrier features sets their input channels to zero; disabling routing sets the latent displacement to zero. The centered and routed branches remain in place, giving the four factorial variants identical parameter counts. The *Shifted equation-specified route* control applies the same deterministic shift as above throughout both training and testing.

Learned route. The learned-route displacement head has a pointwise input projection, one local residual block, and a pointwise output projection. Its width is 24 in 2D and 12 in 3D; it outputs two or three displacement components. The final projection is initialized to zero. For each component,

$$d_{\phi,j}(q) = a_j \tanh([H_\phi(x_\mu(q))]_j)\,, \qquad a_j = \max\{3\ \mathrm{RMS}_{\mathrm{train}}(d_j), \Delta x_j\}. \tag{50}$$

Here d_j is the equation-specified departure displacement on training states and Δx_j is the full-grid spacing. The 2D scale estimate uses 64 training states sampled with replacement using seed 23071; the 3D estimate uses 16 evenly spaced entries in the ordered list of training trajectory/frame pairs. The head receives the same normalized input as the forecasting core and is optimized jointly with it through the forecasting loss.

State advection + residual. The *State advection + residual* control first computes $q_{\mathrm{adv}} = \mathcal{I}[q](\boldsymbol{\Xi}[q])$ and then applies

$$\widehat{q}^{+} = q + \Pi_\mu(q_{\mathrm{adv}} - q + \sigma_{\Delta q,\mu} R_\theta(x_\mu(q)))\,. \tag{51}$$

The centered network receives the current state, carrier, and context, and all latent displacements are set to zero. It is trained from scratch with the same loss, training budget, and checkpoint rule. The task projection acts on the complete state increment, including $q_{\mathrm{adv}} - q$. The diagnostic carrier and departure map are recomputed from the current prediction at every step.

I.3 VISUALIZATION SAMPLE AND TRANSECT

For Figure 2e, the selected test window maximizes the reduction in final relative L^2 error from *Carrier feature only* to GRNO among the twelve prescribed windows. Within this window, the vertical line maximizes the reduction in absolute field error averaged over $h = 1, \ldots, 40$ and the vertical grid coordinate. Both independently trained models use the same window and line. The three concentration maps use a common color scale, and the same reference concentration contour is overlaid as a dashed line on each map.

J Additional Visualizations

J.1 Additional qualitative forecasting results

Figures 3–7 compare reference fields and pointwise absolute errors at ten horizons spanning the 40-step free rollout, using GRNO and the strongest baseline in Table 1.

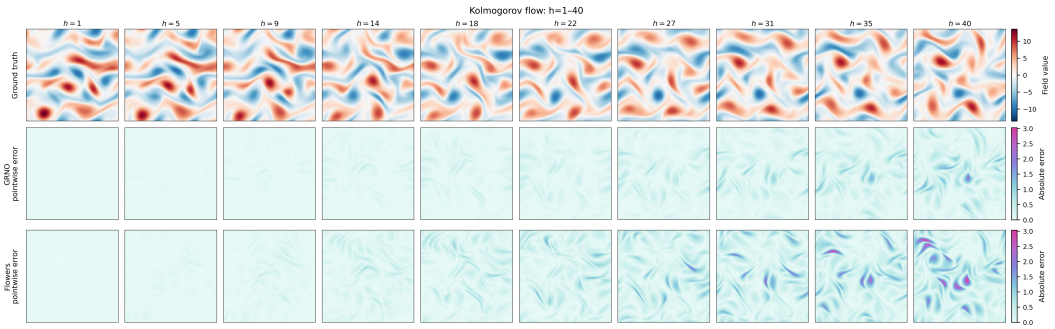


Figure 3: **Long-horizon qualitative comparison on Kolmogorov flow.** The first row shows the reference vorticity field at ten horizons spanning the 40-step free rollout. The second and third rows show the pointwise absolute errors of GRNO and Flowers, respectively. The error scale is shared by both models and all displayed horizons.

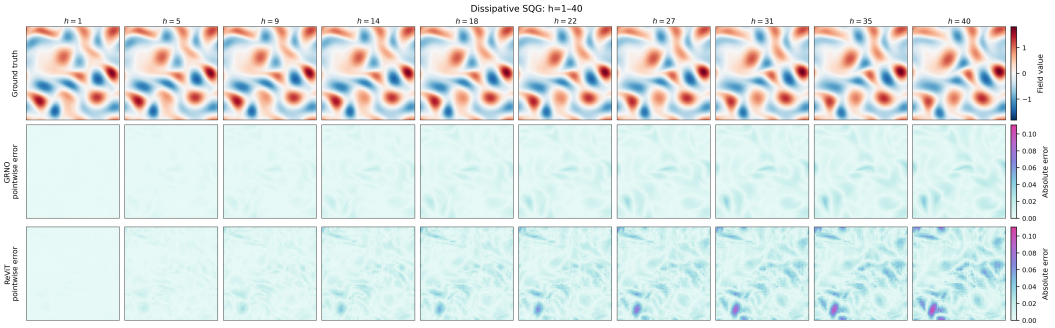


Figure 4: **Long-horizon qualitative comparison on dissipative SQG.** The first row shows the reference scalar field at ten horizons spanning the 40-step free rollout. The second and third rows show the pointwise absolute errors of GRNO and ReViT, respectively. The error scale is shared by both models and all displayed horizons.

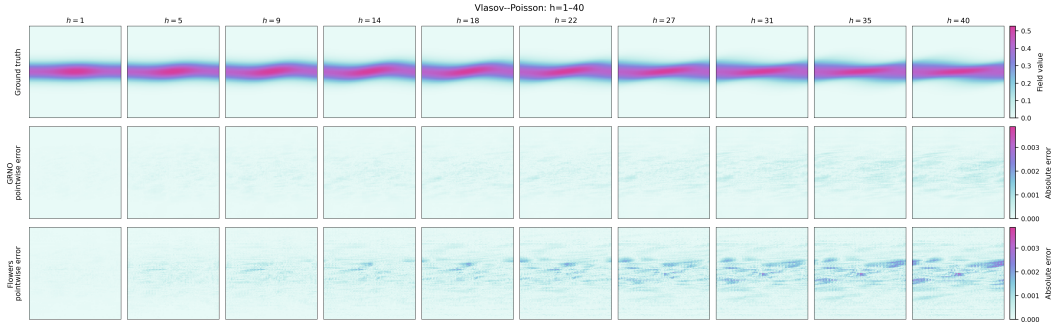


Figure 5: **Long-horizon qualitative comparison on Vlasov–Poisson.** The first row shows the reference phase-space distribution at ten horizons spanning the 40-step free rollout. The second and third rows show the pointwise absolute errors of GRNO and Flowers, respectively. The error scale is shared by both models and all displayed horizons.

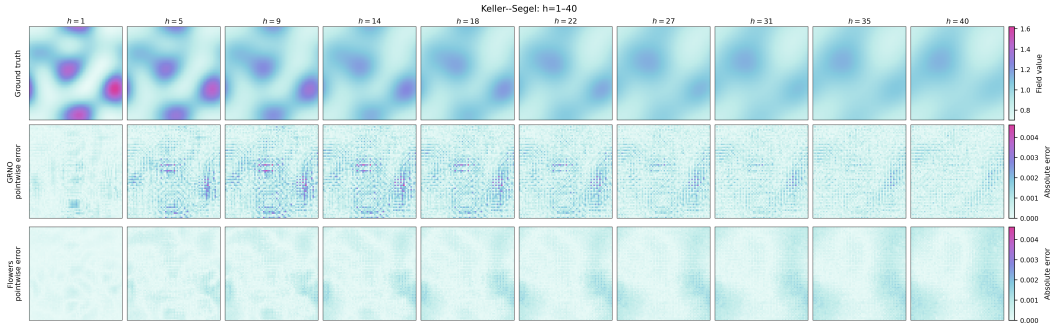


Figure 6: **Long-horizon qualitative comparison on Keller–Segel.** The first row shows the reference density field on the fixed visualization slice at ten horizons spanning the 40-step free rollout. The second and third rows show the pointwise absolute errors of GRNO and Flowers, respectively. The error scale is shared by both models and all displayed horizons.

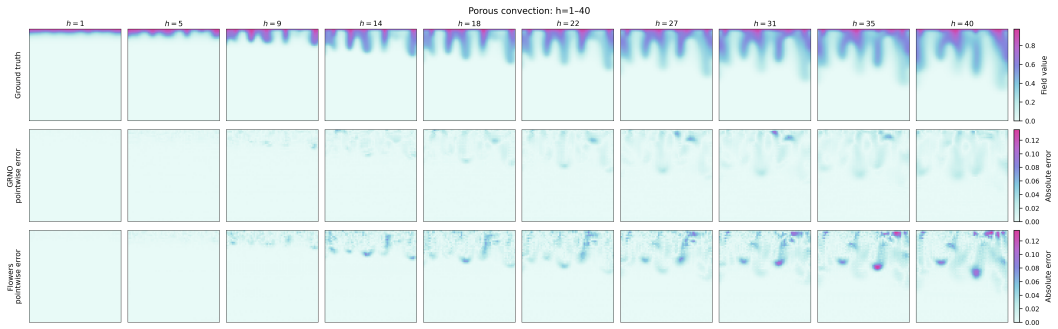


Figure 7: **Long-horizon qualitative comparison on porous convection.** The first row shows the reference concentration field on the fixed visualization slice at ten horizons spanning the 40-step free rollout. The second and third rows show the pointwise absolute errors of GRNO and Flowers, respectively. The error scale is shared by both models and all displayed horizons.

J.2 QUALITATIVE EFFECTS OF ROUTE INTERVENTIONS

Figures 8–12 show the fixed-weight route interventions at $h = 40$. For each system, the fitted GRNO checkpoint is held fixed, and only the departure displacement is changed.

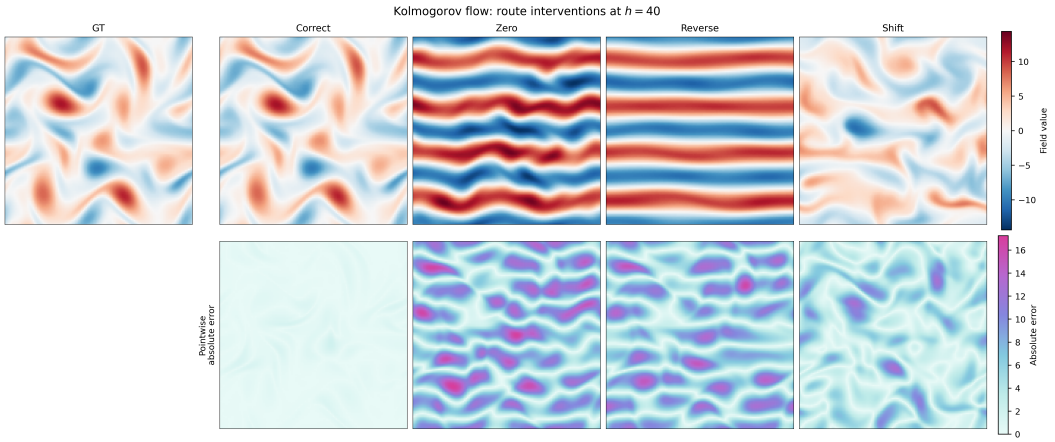


Figure 8: **Route interventions on Kolmogorov flow at $h = 40$.** The top row shows the reference field and predictions using the correct, zero, reversed, and shifted routes. The bottom row shows the corresponding pointwise absolute errors. All predictions use the same fitted GRNO checkpoint; only the departure route is changed.

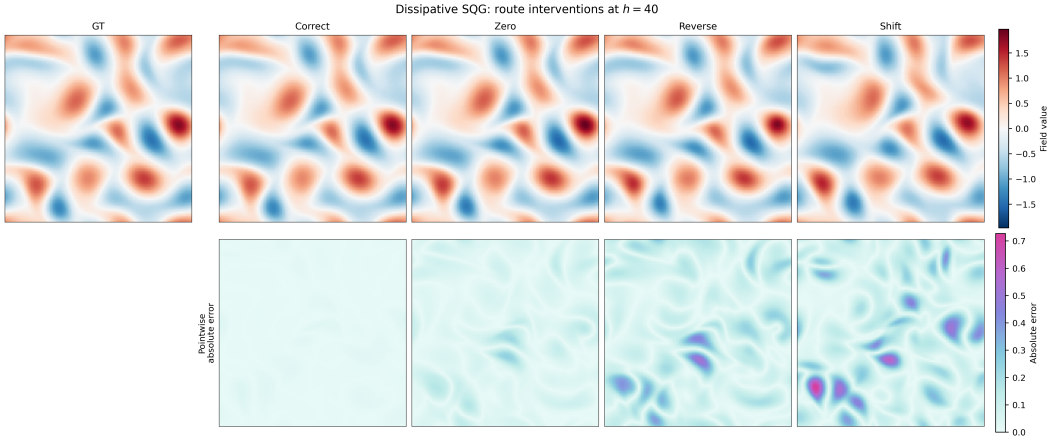


Figure 9: **Route interventions on dissipative SQG at $h = 40$.** The top row shows the reference field and predictions using the correct, zero, reversed, and shifted routes. The bottom row shows the corresponding pointwise absolute errors. All predictions use the same fitted GRNO checkpoint; only the departure route is changed.

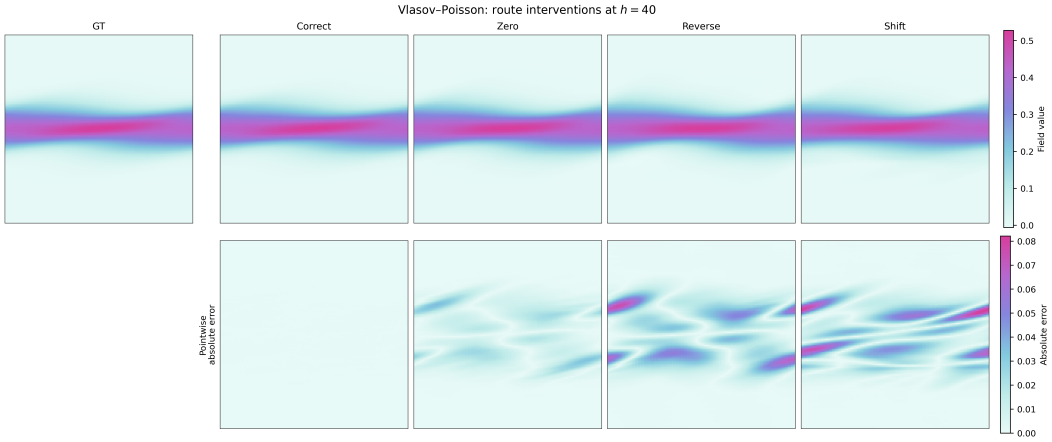


Figure 10: **Route interventions on Vlasov–Poisson at $h = 40$.** The top row shows the reference phase-space distribution and predictions under the *Correct*, *Zero*, *Reverse*, and *Shift* interventions. The bottom row shows the corresponding pointwise absolute errors. All predictions use the same fitted GRNO checkpoint; only the departure route is changed.

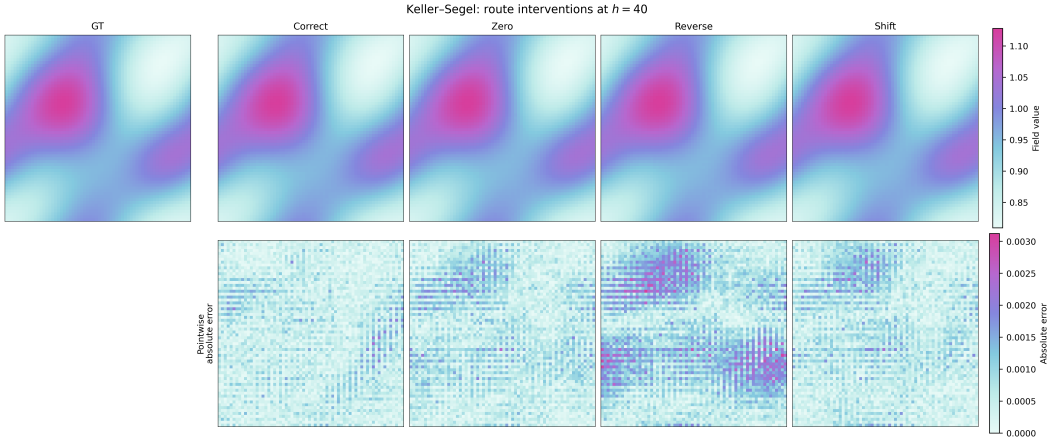


Figure 11: **Route interventions on Keller–Segel at $h = 40$.** The top row shows the reference field and predictions on the fixed visualization slice using the correct, zero, reversed, and shifted routes. The bottom row shows the corresponding pointwise absolute errors. All predictions use the same fitted GRNO checkpoint; only the departure route is changed.

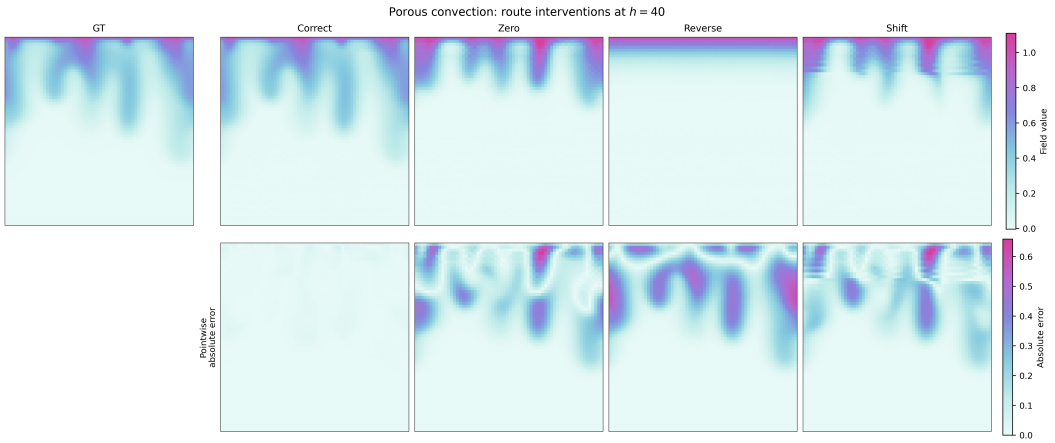


Figure 12: **Route interventions on porous convection at $h = 40$.** The top row shows the reference concentration field and predictions on the fixed visualization slice using the correct, zero, reversed, and shifted routes. The bottom row shows the corresponding pointwise absolute errors. All predictions use the same fitted GRNO checkpoint; only the departure route is changed.