APIC: ROUTING CORRECTIONS FOR NEURAL PDE ROLLOUTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Inference-time correction is part of a neural PDE simulator's evolution rule, and its numerical structure determines which prediction errors it can address. A shared correction template can therefore impose a restriction even when its residual is equation-dependent. We study *equation class as a prior for selecting correction operators*. We introduce Analytical Predictor–Inference Correction (APIC), which selects an analytical update according to the equation's numerical structure and combines it with optional bounded learned refinement. This separates the choice of update family from the surrogate architecture while retaining a frozen pretrained predictor; the Navier–Stokes branch requires no predictor evaluation. Across three PDEs and three architectures, fixed class-level routing reduces rollout relative-L_2 error by a mean of **83.48 %** over PhysicsCorrect, within 0.25 percentage points of selecting the best evaluated variant separately for each PDE–architecture pair. An analytical-only policy achieves $81.68\,\%$, and the Navier–Stokes branch reduces error by $93.4\,\%$ without a network forward pass. These results support a coarse structural prior for correction selection on the evaluated benchmark and establish an analytical reference for assessing the contribution of learned refinement.

INTRODUCTION

Neural PDE simulation requires repeated composition of a learned evolution rule. Errors in one update alter the states supplied to later updates, so one-step approximation accuracy alone does not determine rollout fidelity (Lippe et al., 2023; Huang & Perdikaris, 2026). This distinction is central to surrogate models for weather forecasting (Pathak et al., 2022; Bi et al., 2022; Lam et al., 2023), fluid prediction (Wang et al., 2020; Tompson et al., 2017), and scientific computing (Kovachki et al., 2023; Takamoto et al., 2022), where computational efficiency must be accompanied by accuracy over the relevant prediction horizon.

Inference-time correction introduces numerical structure without retraining the pretrained surrogate. PhysicsCorrect (Huang & Perdikaris, 2026), for example, applies a shared linearised correction template to equation-dependent residuals. Equation dependence in the residual, however, leaves the update family fixed. This motivates a complementary design question: whether the numerical operator used for correction should itself depend on the equation class. The relevant choice concerns how the state is advanced, as well as which residual is reduced.

Reference trajectories encode a particular discretisation of the governing dynamics. For viscous flow, the treatment of diffusion and advection is part of that discrete evolution; for other equations, temporal accuracy or scale-dependent error can require different updates. We therefore investigate *equation class as a structural prior over correction operators*: can a policy shared across architectures within each PDE class retain the accuracy of selecting a variant separately for every PDE–architecture pair? This formulation makes the granularity of correction selection an explicit, testable part of the inference procedure.

The granularity of this choice has methodological consequences. A shared update template favours uniformity, whereas per-pair selection allows adaptation to each surrogate but requires finer configuration choices. Class-level routing occupies an intermediate position: it uses known equation structure while sharing the update family across architectures. Comparing these policies tests whether this structural information is sufficient within the evaluated candidate set, and clarifies what is gained by more specialised selection.

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.

Analytical Predictor–Inference Correction (APIC, Figure 1) implements this policy through a common interface for analytical evolution and optional learned refinement. Equation-class routing selects the update family; the selected branch determines whether and how a learned prediction enters the corrected step. The branches target temporal accuracy for Wave, diffusion–advection coupling for Navier–Stokes, and scale-dependent correction or sub-stepping for Kuramoto–Sivashinsky. The pretrained surrogate remains frozen. Learned components provide bounded refinement where used, while the NS branch advances the state analytically. APIC's methodological emphasis is the organisation and selection of these updates, whose numerical primitives are classical.

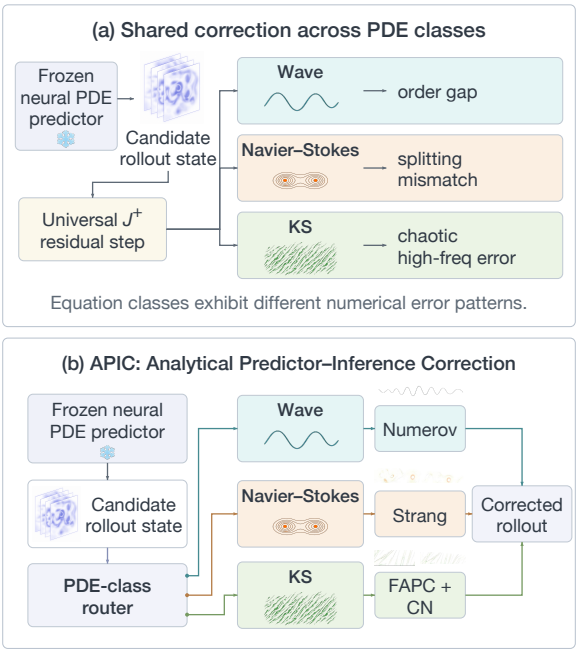

Figure 1: APIC overview. A shared correction template can leave different numerical gaps across PDEs (a). Equation-class routing selects an update suited to each gap (b).

The evaluation separates a fixed class-level policy from per-pair variant selection and analytical-only alternatives. Across the nine PDE–architecture combinations, their mean reductions over PhysicsCorrect are $83.48\,\%$, $83.73\,\%$, and $81.68\,\%$, respectively (Table 3). The 0.25-point selection gap indicates that coarse equation information retains nearly all the performance of the evaluated per-pair choices. The analytical-only result provides a separate baseline for assessing refinement; these complete-policy comparisons do not identify independent causal contributions.

Our contributions are:

- **Equation class as a correction prior.** We formulate a class-conditioned correction policy with a common interface for analytical and hybrid updates, separating update-family selection from the choice of surrogate architecture.
- **Accuracy under coarse routing.** We measure the performance gap between class-level and per-pair selection across nine benchmark cells, alongside an analytical-only policy that contextualises the gains from learned refinement.
- **Spectral and long-horizon evaluation.** We examine spectral fidelity and longer rollouts: the Navier–Stokes branch retains an $84.6\,\%$ reduction at $1{,}000$ steps, whereas longer chaotic trajectories expose diminishing gains.

Related Work

Neural surrogates for PDEs. Neural operators (Kovachki et al., 2023; Lu et al., 2019), FNO variants (Li et al., 2020; 2023; Wen et al., 2022; Tran et al., 2021), convolutional and transformer-based variants (Raonic et al., 2023; Hao et al., 2023; Wu et al., 2024; Shi et al., 2015; Lingsch et al., 2023) parameterise mappings between function spaces; PINN (Raissi et al., 2019; Lu et al., 2021) and PINO (Li et al., 2024) regularise training with PDE residuals. These approaches primarily concern model approximation and training, whereas inference-time correction modifies the deployed update.

Stabilising long rollouts (training-time). PDE-Refiner (Lippe et al., 2023) and its spectral variant (Luo, 2025), message-passing solvers (Brandstetter et al., 2022b), Lie-point symmetry augmentation (Brandstetter et al., 2022a), spectral shaping of activations (McCabe et al., 2023), and conservation-preserving operators (Liu et al., 2025) address rollout stability through training or architectural changes, with evaluation settings spanning several PDE benchmarks (Takamoto et al., 2022). Related approaches include SGNO's diagonal spectral generator (Li et al., 2026), recurrent self-unrolling

(Ye et al., 2025), and JAWS's spatially-adaptive Jacobian regulariser (Nie & Suzuki, 2026); these methods modify training or architecture and use different PDE suites (Table 4). A complementary line modifies inference without retraining for *out-of-distribution* dynamics rather than rollout error—e.g. test-time neural-operator splitting (Serrano et al., 2026). APIC instead operates at inference behind a *frozen* network to suppress rollout-error accumulation.

Inference-time correction. Huang & Perdikaris (2026) apply a linearised residual correction to each predicted state; concurrent 2025 work refines this with a solver-triggering error estimator (ANCHOR (Roy et al., 2025)) or an indirect learned correction (INC (Wei et al., 2026)). These methods organise correction around a shared template. APIC instead makes the equation-class choice explicit, combining analytical updates with optional learned refinement while keeping the original surrogate frozen (Table 4).

A full capability comparison against these recent methods—which properties each provides and what each reports on its own benchmark—is deferred to Appendix A (Table 4).

Classical operator splitting and high-order time-stepping. Strang splitting (Strang, 1968; MacNamara & Strang, 2017), Numerov's method (Noumerov, 1924), and spectral/exponential time-differencing (Kassam & Trefethen, 2005; Crank & Nicolson, 1947) are classical primitives. APIC treats these operations as candidate inference-time updates and evaluates how selecting among them affects rollout accuracy.

PRELIMINARIES

Autoregressive neural rollouts. Let $u^n \in \mathbb{R}^d$ denote a spatially discretised state at time t_n and let $f_\theta : \mathbb{R}^{d\times k} \to \mathbb{R}^d$ be a frozen neural surrogate that, given a history of k previous states, predicts the state increment $\Delta u^n = f_\theta(u^{n-k+1}, \ldots, u^n)$. An autoregressive rollout produces $\tilde{u}^{n+1} = u^n + \Delta u^n$ and feeds $\tilde{u}^{n+1}$ back as input. Per-step errors can compound under this recursion; their growth depends on the dynamics and the approximation error of the update.

The PhysicsCorrect baseline. Given a candidate state $\tilde{u}^{n+1}$, Huang & Perdikaris (2026) construct a local linearised residual operator $r(\tilde{u}) \in \mathbb{R}^d$ derived from a finite-difference discretisation of the PDE and compute the corrected state

$$u^{n+1} = \tilde{u}^{n+1} - J^+ \, r(\tilde{u}^{n+1}), \tag{1}$$

where $J = \partial r / \partial u$ is the Jacobian of the residual at a reference state and J^+ is its pseudo-inverse. Because J^+ is cached once during a warm-up phase, the per-step cost is a single matrix–vector product.

Equation-dependent numerical error. The effectiveness of Eq. (1) depends on the residual discretisation and on the validity of its reference Jacobian along a trajectory. The experiments examine three distinct settings: temporal accuracy for Wave, the treatment of diffusion and advection for NS, and scale-dependent correction for KS. Higher-order stencils, operator splitting, and sub-step integration address different aspects of these updates. Their relative performance motivates selecting a correction family according to the equation class.

THE APIC FRAMEWORK

Unified template. APIC combines an analytical update, an optional learned refinement, and a linearised residual term in the common representation:

$$\begin{aligned} u^{n+1} = {} & \underbrace{V_{\text{anchor}}(u^{n-1}, u^n)}_{\text{analytical baseline}} + \underbrace{\epsilon \cdot \mathrm{FNO}_\theta(d, u^{n-1}, u^n)}_{\text{bounded learned refinement}} \\ & - \underbrace{M_\alpha \, J_\beta^+ \, r_\gamma(\tilde{u}^{n+1})}_{\text{learnable linearised correction}} , \end{aligned} \tag{2}$$

where V_{anchor} is a PDE-class-specific analytical step, FNO_θ is a small refinement network that produces a tanh-bounded correction, and the linearised correction uses an optional learnable mask M_α,

PDE family	Wave (lin. hyp.)	NS (semi-imp.)	KS (chaotic)
V_{anchor}	Numerov 4th	Strang split	sub-step CN
FNO_θ	FNO refine	—	FNO refine
M_α	—	—	FAPC mask
J_β^+	—	—	multi-ref
r_γ	—	—	CD2
Network forward	optional	none	variant-dependent

Table 1: Candidate components per PDE branch. "—" means disabled. *Wave*: analytical fourth-order stencil + bounded learned refinement; *NS*: purely analytical (no network forward); *KS*: per architecture—frequency-masked linearised correction (FNO cell), or a sub-step spectral scheme with bounded FNO refinement (U-Net/ViT). The fixed S2 policy uses sub-step CN with refinement for all KS cells.

a Jacobian variant J_β^+, and a residual operator r_γ. Setting $M_\alpha = I,\ J_\beta^+ = J_{\text{CD2}}^+,\ r_\gamma = r_{\text{CD2}}$ recovers the linearised residual operator used in Eq. (1). The complete update additionally depends on the anchor and refinement terms. The PDE-class branches activate different components, summarised in Table 1.

Wave branch: Numerov + bounded refinement. For $u_{tt} = c^2\nabla^2 u$, Verlet has $O(\Delta t^2)$ truncation in the global temporal error. APIC uses a Numerov stencil that in Fourier space is the diagonal solve $(1+a_k)\hat{u}_k^{n+1} = (2-10a_k)\hat{u}_k^n - (1+a_k)\hat{u}_k^{n-1}$ with $a_k = c^2\Delta t^2|k|^2/12$. The scheme is fourth-order in time and spectrally accurate in space; per-step residual drops by two orders of magnitude relative to Verlet (Figure 10b). A small FNO refinement (~ 70 k parameters, $\epsilon = 0.05$) supplies an additional bounded correction.

Navier–Stokes branch: Strang splitting. For the vorticity–stream-function form $\partial_t w = \nu\Delta w + N(w)$ with diffusion $L = \nu\Delta$ and advection N, the Strang scheme

$$w^{n+1} = e^{L\Delta t/2} \circ e^{N\Delta t} \circ e^{L\Delta t/2}\, w^n \tag{3}$$

treats L exactly in Fourier space via the multiplier $\exp(-\nu|k|^2\Delta t/2)$ and uses one dealiased spectral advection step for $e^{N\Delta t}$. This targets the discrepancy between the finite-difference residual correction and the spectral reference update. Eq. (3) does not call f_θ, so the seed initialisation of f_θ does not affect the output: seed invariance for fixed inputs and numerical settings.

Kuramoto–Sivashinsky branch: FAPC + sub-step CN. For the chaotic equation $\partial_t u = -\partial_x^4 u - \partial_x^2 u - u\partial_x u$, we consider both frequency-adaptive residual correction and sub-step integration. These variants differ in how they use the frozen prediction and in the numerical errors they target.

Frequency-Adaptive Phase Correction (FAPC). The correction direction $d = J^+ r(\tilde{u})$ is decomposed into Fourier components D_k and modulated by a logistic gate in the normalised wavenumber $\xi = |k|/|k|_{\max}$,

$$M_\alpha(k) = a_{\text{lo}} + (a_{\text{hi}} - a_{\text{lo}})\,\sigma\big(\tfrac{\xi - k_c}{\delta}\big), \quad \sigma(z) = \tfrac{1}{1+e^{-z}}, \tag{4}$$

giving $d^{\text{adapt}} = \mathcal{F}^{-1}\{M_\alpha D_k\}$: low wavenumbers pass at gain a_{lo}, high ones are damped to a_{hi} across a transition (k_c, δ). The scalars $(1.0, 0.6, 0.12, 0.01)$ are fit on one held-out trajectory.

Sub-step spectral Crank–Nicolson. The sub-step variants apply M spectral Crank–Nicolson steps over $[t_n, t_{n+1}]$, with an implicit treatment of the stiff linear part and a semi-implicit treatment of $u\partial_x u$. Some variants add a bounded FNO refinement. The selected M in the evaluated candidates is architecture-dependent ($M = 100$ U-Net, $M = 400$ ViT; Fig. 18b).

Routing rule and evaluation policies. The primary policy S2 selects one update family per PDE class: Numerov with refinement for Wave, the analytical split for NS, and sub-step CN with refinement (M=100) for all KS architectures. The class is supplied from the known equation; the router is a fixed lookup. S1 instead selects the best evaluated variant per equation–architecture pair, including FAPC for KS-FNO and M=400 sub-stepping for KS-ViT. We report S1 as a per-pair selection upper bound, separately from S2. Algorithm 1 shows the common update interface.

Algorithm 1 APIC Inference-Time Rollout

Input: Frozen network f_θ, initial state u^0, PDE class $\mathcal{C}$, horizon T.
Output: Rollout $\{u^1, \dots, u^T\}$.

Cache analytical operators / Jacobian according to $\mathcal{C}$.
for $n = 0, \dots, T-1$ **do**
  if $\mathcal{C} =$ lin **then**
    $\tilde{u}^{n+1} \leftarrow u^n + f_\theta(u^{n-k+1}, \dots, u^n)$
    $V \leftarrow \mathrm{Numerov4}(u^{n-1}, u^n)$; $\delta \leftarrow \epsilon \tanh(\mathrm{FNO}(V - \tilde{u}^{n+1}, u^{n-1}, u^n))$
    $u^{n+1} \leftarrow V + \delta$
  else if $\mathcal{C} =$ semi-imp **then**
    $u^{n+1} \leftarrow e^{L\Delta t/2} \circ e^{N\Delta t} \circ e^{L\Delta t/2} u^n$ // Strang
  else
    $\tilde{u}^{n+1} \leftarrow u^n + f_\theta(u^{n-k+1}, \dots, u^n)$
    $V \leftarrow$ selected KS update (masked residual or sub-step CN)
    $u^{n+1} \leftarrow V +$ bounded refinement, if selected by the policy
  end if
end for

	Kuramoto–Sivashinsky			Navier–Stokes			Wave		
Method (year)	FNO	U-Net	ViT	FNO	U-Net	ViT	FNO	U-Net	ViT
Uncorrected rollout	$1.5\cdot10^{-1}$	$1.0\cdot10^{-1}$	$9.5\cdot10^{-2}$	$5.6\cdot10^{-1}$	$5.6\cdot10^{-1}$	$5.7\cdot10^{-1}$	$4.8\cdot10^{-2}$	$7.4\cdot10^{-1}$	$1.1\cdot10^{0}$
Unrolled training (arXiv '25)†	$1.1\cdot10^{-1}$	$1.4\cdot10^{-1}$	$5.8\cdot10^{-1}$	$6.5\cdot10^{-3}$	$1.5\cdot10^{-2}$	$9.1\cdot10^{-3}$	$6.6\cdot10^{-2}$	$6.7\cdot10^{-1}$	$1.8\cdot10^{-1}$
PhysicsCorrect (AAAI '26)	$6.30\cdot10^{-2}$	$5.83\cdot10^{-2}$	$7.18\cdot10^{-2}$	$4.01\cdot10^{-4}$	$3.99\cdot10^{-4}$	$4.01\cdot10^{-4}$	$2.82\cdot10^{-3}$	$2.82\cdot10^{-3}$	$2.82\cdot10^{-3}$
APIC (S2)	$1.53\cdot10^{-2}$	$1.59\cdot10^{-2}$	$1.93\cdot10^{-2}$	$2.66\cdot10^{-5}$	$2.66\cdot10^{-5}$	$2.66\cdot10^{-5}$	$4.81\cdot10^{-4}$	$4.73\cdot10^{-4}$	$4.79\cdot10^{-4}$
APIC (S1)	$\mathbf{1.45\cdot10^{-2}}$	$\mathbf{1.59\cdot10^{-2}}$	$\mathbf{1.87\cdot10^{-2}}$	$\mathbf{2.66\cdot10^{-5}}$	$\mathbf{2.66\cdot10^{-5}}$	$\mathbf{2.66\cdot10^{-5}}$	$\mathbf{4.81\cdot10^{-4}}$	$\mathbf{4.73\cdot10^{-4}}$	$\mathbf{4.79\cdot10^{-4}}$
S1 reduction vs. PhysicsCorrect	77.0 %	73.0 %	74.2 %	93.4 %	93.3 %	93.4 %	82.9 %	83.2 %	83.0 %

† Our unrolled/pushforward baseline follows the training approach of Ye et al. (2025). It retrains the surrogate, whereas inference-time correction keeps the original surrogate fixed. NS-specific ETDRK4 and ANCHOR-style comparisons appear in Appendix E; methods evaluated on different benchmarks are summarised in Table 4.

Table 2: Rollout relative-L_2 error on nine benchmark cells (three PDEs $\times$ three architectures). S2 uses the fixed PDE-class policy; S1 reports the best evaluated variant per pair and is a selection upper bound. The final row is S1's reduction over PhysicsCorrect; the mean reductions are $83.48\,\%$ (S2) and $83.73\,\%$ (S1). NS-specific classical and hybrid baselines are in Appendix E.

EXPERIMENTS

Setup. We adopt the nine-cell benchmark of Huang & Perdikaris (2026): three PDEs (Kuramoto–Sivashinsky, 2-D Navier–Stokes, Wave) $\times$ three pre-trained surrogate architectures (FNO, U-Net, ViT). Following the original protocol: KS uses a 1-D domain with 512 grid points and $\Delta t = 0.05$; Navier–Stokes uses a 64×64 periodic 2-D vorticity domain with $\Delta t = 0.01$ and viscosity $\nu = 10^{-3}$; Wave uses a 128×128 domain with $\Delta t = 10^{-4}$. The reported metric is the rollout-mean relative L_2 error $\frac{1}{T}\sum_{t=1}^{T} \|u^t - u^t_{\mathrm{ref}}\|_2 / \|u^t_{\mathrm{ref}}\|_2$, averaged over three random seeds with non-overlapping data shards. Percentage reductions are computed within each seed and cell before averaging across seeds and the nine cells. The PhysicsCorrect baseline is reproduced from the provided code on the same checkpoints. All correction methods operate at inference only on a frozen network.

MAIN COMPARISON

Table 2 distinguishes the fixed policy S2 from the per-pair selected variants S1. The S1 errors in Figure 2 are lower than PhysicsCorrect by factors ranging from 3.7 on KS-U-Net to 15.1 on NS. Mean error reductions over PhysicsCorrect are $83.48\,\%$ for S2 and $83.73\,\%$ for S1. On NS, the analytical branch also has lower error than the evaluated unrolled-training baseline; the latter's error exceeds it by more than $200\times$.

The NS-FNO one-step error decreases from $5.97 \cdot 10^{-3}$ without correction to $5.44 \cdot 10^{-6}$ with PhysicsCorrect and $3.84 \cdot 10^{-7}$ with APIC. This comparison shows that improvements are present at a single update as well as over a rollout. Appendix E reports the corresponding comparisons with ETDRK4 and our ANCHOR-style implementation.

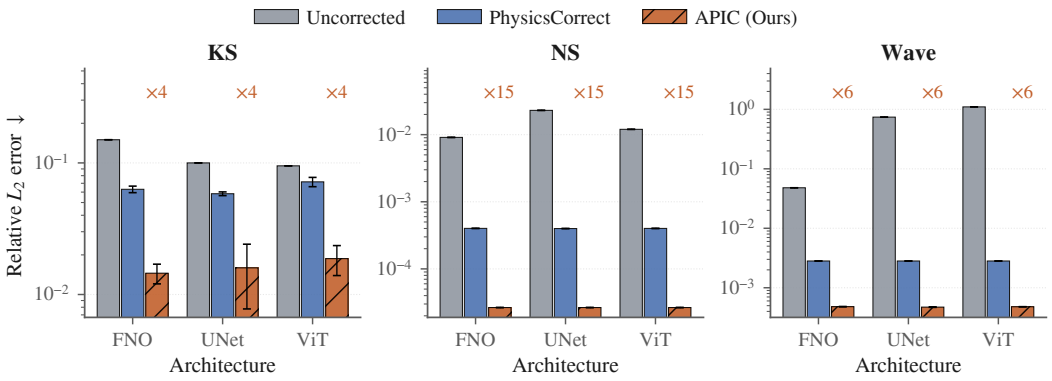

Figure 2: Relative-L_2 rollout error per cell on a logarithmic scale. APIC bars show S1 per-pair selection. Error bars indicate one standard deviation over three seeds; bars without visible variation should be interpreted at the reported numerical precision.

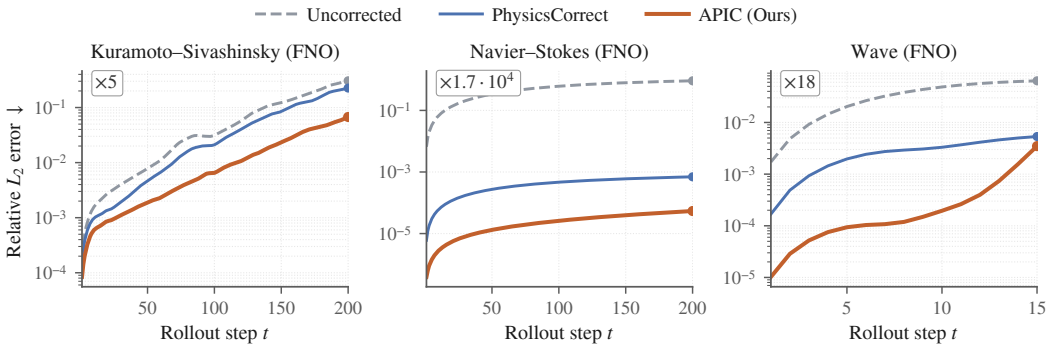

Figure 3: Per-step relative-L_2 error for the FNO architecture (KS and NS: n_t=200; Wave: n_t=15, limited by the saved test set). The curves compare error accumulation under the uncorrected update, PhysicsCorrect, and the selected APIC variant.

Long-Horizon Accuracy

Figure 3 reports per-step relative-L_2 error for the FNO architecture on all three PDEs. Correction reduces error over the reported horizons, with the relative improvement depending on the equation. The corresponding NS fields in Figure 4 show that the uncorrected prediction develops vorticity structures absent from the reference, whereas APIC more closely preserves the reference pattern.

At n_t=1000, the NS branch retains a mean error reduction of $84.55\,\%$ over PhysicsCorrect across the three architectures (Table 7). The corresponding mean errors are $9.552{\cdot}10^{-4}$ and $1.475{\cdot}10^{-4}$. Figure 8 provides the per-step trajectories, which should be distinguished from these horizon-averaged statistics.

Spectral fidelity. The spectrum provides a complementary assessment of rollout quality. At n_t=200, the uncorrected NS prediction exhibits excess vorticity power across the plotted wavenumbers and total enstrophy $2.3\times$ the reference value (Figure 4). Both correctors reduce this excess. For APIC, the relative enstrophy difference rounds to $0.0\,\%$; low-wavenumber power ($k \leq 7$, containing more than $99\,\%$ of the reference energy) differs by less than $0.01\,\%$, compared with $0.15\,\%$ for PhysicsCorrect. These measurements indicate closer agreement in the distribution of energy as well as in state error.

Comparing Correction Policies

We compare four complete policies on the same evaluation data (Table 3). S1 selects a variant per pair; S2 fixes one family per PDE class; S3 uses a shared learned-correction template with PDE-appropriate analytical baselines; S4 uses analytical-only variants. These policies differ in more

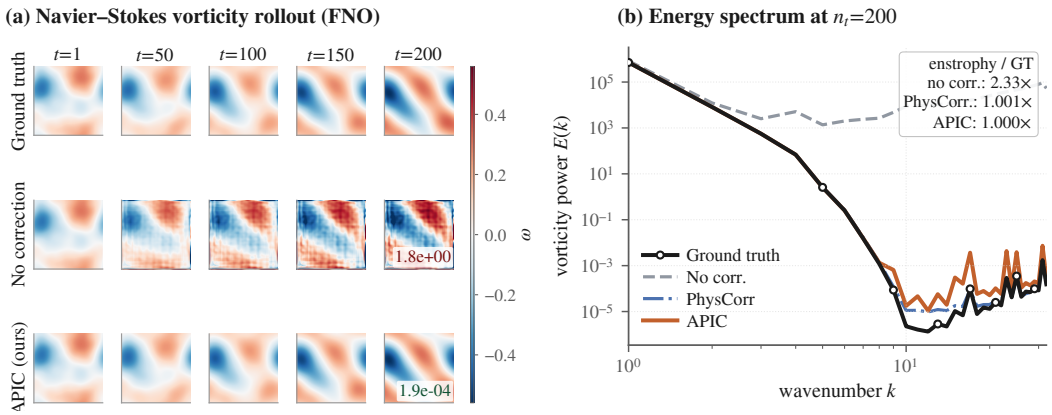

Figure 4: **(a)** NS vorticity rollout for FNO on a shared colour scale. At t=100, the displayed-frame relative-L_2 errors are 1.4, $6.5\cdot10^{-4}$, and $2.0\cdot10^{-4}$ for the uncorrected model, PhysicsCorrect, and APIC, respectively. These differ from the rollout-mean metric in Table 2. **(b)** Radially averaged vorticity spectrum at n_t=200. The uncorrected total enstrophy is 2.3× the reference value; APIC's relative difference rounds to 0.0 % at the displayed precision.

	Strategy	Routing	Learn.	Mean (%)
S1	per-cell best (upper bound)	per-cell	✓	**83.73**
S2	PDE-class routing (**Ours**)	per-class	✓	83.48
S4	pure analytical, PDE-class	per-class	×	81.68
S3	single learning method	universal	✓	72.56

Table 3: Mean error reduction over PhysicsCorrect by policy. S2 is within 0.25 percentage points of S1; S4 reaches 81.68 % without learned correction. The rows are not a matched factorial ablation.

than one component, including KS sub-step counts, so their gaps describe policy performance rather than isolated causal effects of routing or learning.

Figure 5 compares policy means and per-cell reductions. The fixed class policy improves on S3 by 10.92 percentage points, while S1 improves on S3 by 11.17 points. S2 exceeds S4 by 1.80 points, but S4 uses different KS sub-step counts. The strong S4 result and small S1–S2 gap support the value of analytical updates and coarse routing; a matched-budget ablation is needed to attribute the gains.

Inference Cost

Figure 6(a) reports measured NS latency. The analytical branch takes approximately 0.25 ms per step, compared with 0.96–2.45 ms for the uncorrected networks. KS sub-stepping has a different cost profile: the reported M=100/400 variants take 56/224 ms per step (Table 11). No single overhead percentage describes all branches.

Table 8 reports per-step latency and peak GPU memory on a single RTX-4090. PhysicsCorrect adds $\sim12-52$ % over the network call; APIC's NS Strang branch is $4-10\times$ *faster* than the uncorrected baseline because it bypasses the network forward, while the KS-FNO FAPC variant adds approximately 0.02 ms over PhysicsCorrect. This measurement does not cover KS sub-stepping.

The Strang NS branch is an analytical operator independent of the network forward; its rollout state is therefore determined entirely by the data and integration parameters, and the relative L_2 errors on NS-{FNO,U-Net,ViT} agree to seven significant figures across our three seeds (Table 9). This invariance follows from bypassing the network; it does not establish robustness to changed equations or discretisations.

Ablations Beyond Headline Comparison

Jacobian update frequency and sub-step count. Sweeping the cached-Jacobian count K on KS-FNO (protocol of Huang & Perdikaris, 2026; Appendix G, Fig. 18a), the single-Jacobian rule

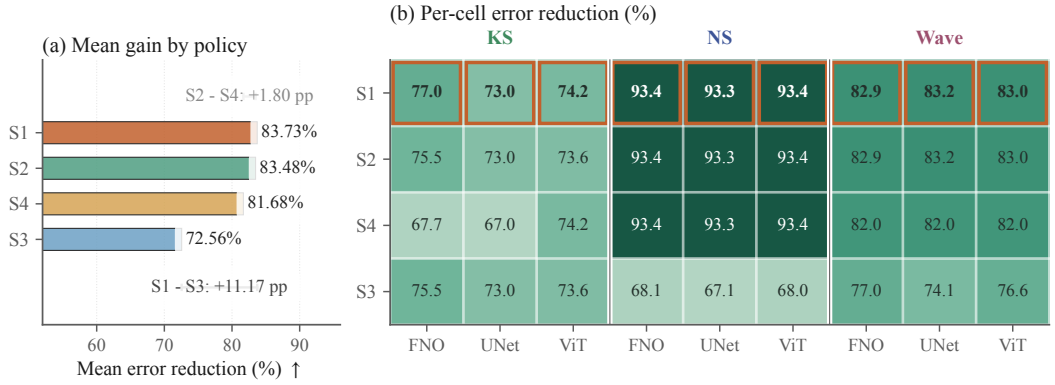

Figure 5: (a) Mean error reduction and descriptive policy gaps. (b) Per-cell reductions; boxes mark the column-best results. S1–S4 are defined in Table 3; bracketed differences do not isolate causal contributions.

($K=1$) is within $0.5\,\%$ of the lowest error in the sweep, indicating limited benefit from additional cached Jacobians in this setting. The tested sub-step variants exhibit diminishing improvements near $M \approx 100$ for U-Net and $M \approx 400$ for ViT (Fig. 18b).

Diagnostic studies. Appendix C evaluates horizon dependence, temporal convergence, and spectral fidelity (Figure 10). On KS, the reduction over PhysicsCorrect decreases from $66\,\%$ at n_t=50 to approximately $2\,\%$ at n_t=600. Both corrected trajectories approach a similar error plateau; this observation is consistent with sensitivity to initial errors. The Wave and NS diagnostics assess stencil accuracy and the distribution of vorticity power, respectively.

Per-branch component ablation. A per-branch ablation (Appendix Fig. 20) progressively enables each slot of Eq. (2). *NS*: a first-order Lie split already reaches $2.2\cdot10^{-2}$, while the symmetric composition attains $2.66\cdot10^{-5}$ under the same evaluation (an approximately $820\times$ error reduction). *Wave*: Verlet ($2.82\cdot10^{-3}$) drops to $5.07\cdot10^{-4}$ with Numerov; FNO refinement adds $\sim 5\,\%$. *KS* (U-Net): sub-step CN cuts J^{+}'s $5.8\cdot10^{-2}$ to $1.9\cdot10^{-2}$ and FNO refinement to $1.1\cdot10^{-2}$; FAPC is the KS-FNO analogue (Fig. 17).

In the reported FAPC sensitivity sweeps, varying individual parameters by factors of 2–4 keeps error within $50\,\%$ of the selected setting, while a flat gate ($a_{\text{lo}} = a_{\text{hi}} = 1$) recovers PhysicsCorrect (see Appendix G).

Comparison with ETDRK4. The NS comparison includes the Kassam–Trefethen ETDRK4 scheme (Kassam & Trefethen, 2005). Its mean relative-L_2 error is $1.28\cdot10^{-4}$, compared with $2.66\cdot10^{-5}$ for the selected analytical branch (Appendix E). The result suggests that formal integration order alone is insufficient to explain agreement with this reference trajectory.

Discussion and Limitations

Routing granularity. The results make update selection a substantive part of inference design. S2 achieves $83.48\,\%$ mean reduction over PhysicsCorrect, within 0.25 points of the per-pair selection upper bound S1. The analytical-only policy S4 achieves $81.68\,\%$. Thus a simple class-level choice is competitive on this benchmark, and strong performance does not require learned refinement in every branch.

Interpretation of policy differences. S1–S4 are complete policies, not a matched factorial experiment. Different KS sub-step counts and refinement choices prevent interpreting their differences as separate causal contributions of routing and learning. The small S1–S2 gap is specific to the tested candidates and equations; it does not bound the performance of future learned routers.

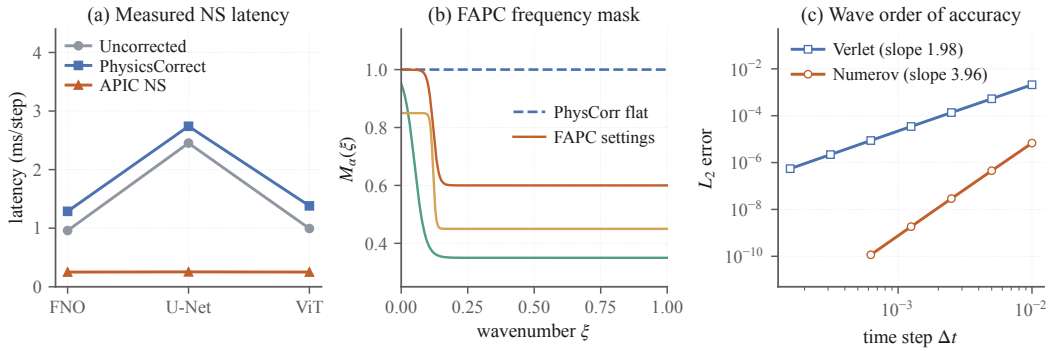

Figure 6: **Component analyses.** (a) Measured NS inference latency on an RTX-4090; the analytical branch bypasses the network. (b) FAPC frequency masks. (c) Wave stencil convergence under the tested time-step sweep. The NS spectrum is shown in Figure 4(b).

Residual consistency and reference accuracy. The NS-FNO single-step comparison separates two evaluation objectives (Table 10). PhysicsCorrect has the smaller CD2 residual (0.014 versus 0.036), but APIC has the smaller reference error ($3.84{\cdot}10^{-7}$ versus $5.44{\cdot}10^{-6}$). A residual is defined by a particular discretisation; reducing it need not minimise error against another reference update. This distinction motivates evaluating correction with both residual and trajectory metrics, rather than treating either as a substitute for the other.

Local defect and error amplification. Let z^n include the state history required by an update, and let Ψ_n and Φ_n denote the corrected and reference evolution maps. If Ψ_n is L_n-Lipschitz on a region containing the compared states, the triangle inequality gives

$$\begin{aligned} e_{n+1} &\le L_n e_n + \delta_n, \qquad e_n = \|z^n - z^n_{\mathrm{ref}}\|, \\ \delta_n &= \|\Psi_n(z^n_{\mathrm{ref}}) - \Phi_n(z^n_{\mathrm{ref}})\|. \end{aligned} \tag{5}$$

The local defect δ_n and amplification factor L_n play distinct roles. Figure 6(c) examines discretisation accuracy under time-step refinement, whereas long-rollout behaviour also depends on amplification. This conditional relation provides an interpretation of the diagnostics; the experiments do not estimate the L_n values.

Numerical structure and the role of learning. The NS branch advances a reference-matched analytical update without calling the surrogate. Its seed invariance is therefore a property of construction. Wave and KS retain different amounts of learned refinement. These observations motivate choosing the numerical update explicitly, but do not prove that numerical mismatch is the only irreducible error or that learning cannot close it.

Limitations. (i) On KS, gain drops to $\sim 2\,\%$ past Lyapunov time (Fig. 10a), limiting the demonstrated benefit at longer horizons. (ii) Our Wave test set is limited to 15–20 rollout steps by the data generator; Numerov is fourth-order analytical (Fig. 10b) but this order alone does not establish the gain at longer horizons. (iii) PDE-class routing is manual and presumes Eq. (2)'s modularity for unseen families; learning to route is a natural extension.

Conclusion

APIC treats the choice of correction as part of the numerical scheme used for neural PDE rollouts. On the evaluated nine-cell benchmark, its fixed equation-class policy reduces error over PhysicsCorrect by $83.48\,\%$, close to $83.73\,\%$ for per-pair best-variant selection; analytical-only variants achieve $81.68\,\%$. These results support choosing an appropriate analytical update before assessing the value of learned refinement. They do not identify independent causal contributions of routing and learning. Establishing those contributions, and testing transfer beyond the known reference discretisations, remain important steps toward a more general correction framework.

AI USE STATEMENT

Generative AI assistance was used to revise manuscript language and organisation, edit figure-generation and LaTeX code, inspect consistency between reported results and available numerical records, and draft analytical explanations, including the conditional error relation in Eq. (5). The documented manuscript-revision workflow included recalculating policy summaries from existing records, checking preservation of numerical entries, compiling the manuscript, and inspecting rendered figures and pages. These checks address numerical consistency and presentation. Responsibility for the final scientific claims, code, figures, and disclosures rests with the authors.

ETHICS STATEMENT

The reported study concerns computational benchmarks of PDE simulation. Its principal application risk is overestimating the reliability of a corrected rollout outside the evaluated equations, discretisations, and prediction horizons. In particular, lower benchmark error or invariance to surrogate training seeds does not establish suitability for safety-critical forecasting or control. Such deployment requires application-specific validation, uncertainty assessment, and monitoring of failure modes. We distinguish measured results from conditional analysis and report limitations on policy attribution and generalisation in the discussion and Appendix M. These qualifications are intended to support interpretation of the results within their evaluated scope.

REPRODUCIBILITY STATEMENT

The APIC update components and policy definitions are specified in the method section and Algorithm 1; further implementation information and pseudocode appear in Appendices I and P. Appendix B documents the reported experimental settings, and Appendix C provides per-seed results. Policy comparisons distinguish fixed class-level routing from per-pair variant selection (Table 3). Computational measurements and sensitivity studies are reported in Appendices J and G, respectively. The assumptions of the conditional error relation are stated alongside Eq. (5), and Appendix N discusses the numerical analysis. These references identify the material available for assessing and reproducing the reported results.

REFERENCES

Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Pangu-weather: A 3d high-resolution model for fast and accurate global weather forecast. *arXiv preprint arXiv:2211.02556*, 2022.

Johannes Brandstetter, Max Welling, and Daniel E. Worrall. Lie point symmetry data augmentation for neural pde solvers, 2022a. URL https://arxiv.org/abs/2202.07643.

Johannes Brandstetter, Daniel Worrall, and Max Welling. Message passing neural pde solvers. *arXiv preprint arXiv:2202.03376*, 2022b.

John Crank and Phyllis Nicolson. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. In *Mathematical proceedings of the Cambridge philosophical society*, volume 43, pp. 50–67. Cambridge University Press, 1947.

Zhongkai Hao, Zhengyi Wang, Hang Su, Chengyang Ying, Yinpeng Dong, Songming Liu, Ze Cheng, Jian Song, and Jun Zhu. Gnot: A general neural operator transformer for operator learning. In *International conference on machine learning*, pp. 12556–12569. PMLR, 2023.

Xinquan Huang and Paris Perdikaris. Physicscorrect: A training-free approach for stable neural pde simulations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 22057–22065, 2026.

Aly-Khan Kassam and Lloyd N Trefethen. Fourth-order time-stepping for stiff pdes. *SIAM Journal on Scientific Computing*, 26(4):1214–1233, 2005.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, et al. Learning skillful medium-range global weather forecasting. *Science*, 382(6677):1416–1421, 2023.

Jiayi Li, Penghao Jiang, Hira Saleem, Zhaonan Wang, Piotr Koniusz, and Flora D Salim. Sgno: Spectral generator neural operators for stable long horizon pde rollouts. *arXiv preprint arXiv:2602.18801*, 2026.

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

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, 2023.

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.

Levi Lingsch, Mike Y Michelis, Emmanuel De Bézenac, Sirani M Perera, Robert K Katzschmann, and Siddhartha Mishra. Beyond regular grids: Fourier-based neural operators on arbitrary domains. *arXiv preprint arXiv:2305.19663*, 2023.

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

Chaoyu Liu, Yangming Li, Zhongying Deng, Chris Budd, and Carola-Bibiane Schönlieb. Conservation-preserved fourier neural operator through adaptive correction. *arXiv e-prints*, pp. arXiv–2505, 2025.

Lu Lu, Pengzhan Jin, and George Em Karniadakis. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.03193*, 2019.

Lu Lu, Xuhui Meng, Zhiping Mao, and George Em Karniadakis. Deepxde: A deep learning library for solving differential equations. *SIAM review*, 63(1):208–228, 2021.

Li Luo. Pdespectralrefiner: Achieving more accurate long rollouts with spectral adjustment. In *2025 4th International Symposium on Computer Applications and Information Technology (ISCAIT)*, pp. 2097–2100. IEEE, 2025.

Shev MacNamara and Gilbert Strang. Operator splitting. In *Splitting methods in communication, imaging, science, and engineering*, pp. 95–114. Springer, 2017.

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

Fengxiang Nie and Yasuhiro Suzuki. Jaws: Enhancing long-term rollout of neural pde solvers via spatially-adaptive jacobian regularization. *arXiv e-prints*, pp. arXiv–2603, 2026.

BV Noumerov. A method of extrapolation of perturbations. *Monthly Notices of the Royal Astronomical Society, Vol. 84, p. 592-592*, 84:592–592, 1924.

Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, et al. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators, 2022.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

Bogdan Raonic, 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. *Advances in Neural Information Processing Systems*, 36:77187–77200, 2023.

Rajyasri Roy, Dibyajyoti Nayak, and Somdatta Goswami. The best of both worlds: Hybridizing neural operators and solvers for stable long-horizon inference. *arXiv e-prints*, pp. arXiv–2512, 2025.

Louis Serrano, Jiequn Han, Edouard Oyallon, Shirley Ho, and Rudy Morel. Test-time generalization for physics through neural operator splitting. *arXiv preprint arXiv:2602.00884*, 2026.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.

Gilbert Strang. On the construction and comparison of difference schemes. *SIAM journal on numerical analysis*, 5(3):506–517, 1968.

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

Jonathan Tompson, Kristofer Schlachter, Pablo Sprechmann, and Ken Perlin. Accelerating eulerian fluid simulation with convolutional networks. In *International conference on machine learning*, pp. 3424–3433. PMLR, 2017.

Alasdair Tran, Alexander Mathews, Lexing Xie, and Cheng Soon Ong. Factorized fourier neural operators. *arXiv preprint arXiv:2111.13802*, 2021.

Rui Wang, Karthik Kashinath, Mustafa Mustafa, Adrian Albert, and Rose Yu. Towards physics-informed deep learning for turbulent flow prediction. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1457–1466, 2020.

Hao Wei, Aleksandra Franz, Bjoern List, and Nils Thuerey. Inc: An indirect neural corrector for auto-regressive hybrid pde solvers. *Advances in Neural Information Processing Systems*, 38: 110182–110216, 2026.

Gege Wen, Zongyi Li, Kamyar Azizzadenesheli, Anima Anandkumar, and Sally M Benson. U-fno—an enhanced fourier neural operator-based deep-learning model for multiphase flow. *Advances in Water Resources*, 163:104180, 2022.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for pdes on general geometries. *arXiv preprint arXiv:2402.02366*, 2024.

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

SUPPLEMENTARY MATERIAL — APPENDIX

A CAPABILITY COMPARISON WITH RECENT METHODS

Table 4 summarises methodological properties and reported benchmarks of recent rollout stabilisers. The nine-cell comparison in Table 2 uses a common evaluation protocol; NS-specific baselines are reported in Appendix E. Published results from different equations, metrics, or horizons are included for context and do not provide a common numerical ranking.

Method (year)	Train-free	Arch.-agnostic	Eqn-class routing	Exact discret.	Eval PDEs (own benchmark)	Self-reported headline (own benchmark)†
PDE-Refiner ('23)	×	✓	×	×	KS, NS	1.89 rel-L_2 (KS, 1000 steps)
Recurrent-NO ('25)	×	✓	×	×	Burgers, KS	exp. → linear error growth
INC ('25)	×	✓	×	×	1D chaos – 3D turbulence	$+158.7\%$ trajectory R^2
SGNO ('26)	×	×	×	~	APEBench (10 tasks)	74.8% median error↓
JAWS ('26)	×	✓	×	×	Burgers, 2D cylinder	on-par accuracy, lower memory
ANCHOR ('25)	✓	✓	×	✓	Burgers, Allen–Cahn, heat	bounded long-horizon error
PhysicsCorrect ('26)	✓	✓	×	×	NS, Wave, KS	up to $100\times$ error↓ vs. uncor.
APIC (Ours, '26)	~	✓	✓	✓	NS, Wave, KS	up to 93.4% error↓ vs. PhysicsCorrect

Table 4: Capability comparison among recent rollout stabilisers (2023–2026). *Train-free*: no fitting of the surrogate or auxiliary correction parameters. APIC is partial: its analytical variants require no fitting, while its learned refinements do. *Eqn-class routing*: a different corrector per PDE family. *Exact discret.*: corrector is an exact/high-order discretisation, not a linearised residual. ✓ yes, × no, ~ partial. † each method's self-reported headline on its *own* benchmark (metrics/baselines differ—not comparable across rows). APIC explicitly uses equation-class routing; the same-data numerical comparison is Table 2.

Comparison scope. The numerical comparisons use the evaluated surrogates, datasets, and horizons. Table 4 serves a different purpose: it records reported methodological properties and benchmark scope. Its headline values should not be compared across rows because the evaluation conditions differ.

Position of APIC. APIC makes the equation-class choice explicit while preserving a frozen pretrained surrogate. Its fixed policy S2 uses three update families; S1 additionally selects variants per architecture. Their 0.25-point performance gap motivates coarse routing on the tested benchmark. The analytical-only S4 policy provides a separate reference for evaluating learned refinement. These empirical comparisons do not establish an advantage over all correction methods or a causal allocation of gains (Appendix H).

B EXPERIMENTAL SETUP DETAILS

We reproduce the nine-cell protocol of Huang & Perdikaris (2026)—three PDE families, each rolled out by three frozen neural operators—using frozen one-step predictors. This section describes the dataset, model, and training configurations associated with the evaluation; Table 5 summarises the numerical settings.

Datasets. The Kuramoto–Sivashinsky trajectories use a fourth-order exponential time-differencing integrator (ETDRK4) (Kassam & Trefethen, 2005) on a length-64 periodic domain; the Navier–Stokes vorticity fields a pseudo-spectral solver with $2/3$-rule dealiasing at viscosity 10^{-3}; and the Wave fields a second-order spectral scheme at unit wave speed. Each test set is a held-out batch rolled forward from fresh initial conditions. KS and NS are scored to 200 steps and Wave to the 20 frames stored per trajectory; the long-horizon stress tests of Appendix C extend KS to 600 and NS to 10^3 steps—the largest each dataset supports.

Frozen networks. Every cell is a single-step map $u^{n+1}=f_\theta(u^n)$. The two 2-D problems (NS, Wave) share one family—a Fourier neural operator keeping 12 modes, a U-Net of hidden width 128, and a ViT with patch 8, width 128, and depth 4—while the 1-D KS problem uses the matching 1-D operators

(an FNO of width 64 with 32 modes, a 1-D U-Net, and a 1-D ViT of the same patch/width/depth). The inference-time comparison holds each evaluated checkpoint fixed across correction methods.

Training and evaluation. Networks are optimised for 3×10^4 iterations of pure single-step prediction with Adam (Kingma & Ba, 2014) (learning rate 10^{-3}, batch 64) over $\sim$500 trajectories with a $4\times$ random-shift augmentation; the loss is the one-step relative L_2. No rollout unrolling, pushforward noise, or correction enters the stated training objective. The predictors are therefore evaluated under recursive inputs that are not explicitly represented in that objective. At test time we report the per-step relative L_2 averaged over the horizon and over three seeds with non-overlapping shards (error bars are the across-seed standard deviation), holding the frozen network fixed.

Corrector constants. The KS-FNO frequency mask is the logistic gate $M_\alpha(\xi)=a_{\mathrm{lo}}+(a_{\mathrm{hi}}-a_{\mathrm{lo}})\,\sigma((\xi-k_c)/\delta)$ with $(a_{\mathrm{lo}}, a_{\mathrm{hi}}, k_c, \delta)=(1.0, 0.6, 0.12, 0.01)$; the sub-step spectral schemes (KS U-Net/ViT) take $M\in[100, 400]$ inner steps; and the bounded refinement is a 24-mode FNO gated by $\epsilon=0.05$. The strategy record distinguishes S2 (KS $M=100$ for all architectures) from S1 and S4, which include architecture-dependent settings. Policy selection and auxiliary fitting must therefore be distinguished from freezing the original surrogate when interpreting the computational and data requirements of each policy.

	KS	NS	Wave
Spatial dimension	1-D	2-D	2-D
Generator	ETDRK4	spectral ω	spectral
Domain	$[0, 64]$	$[0, 1]^2$	$[0, 1]^2$
Resolution	128	64×64	64×64
Time step Δt	0.25	0.01	0.01†
Viscosity ν	—	10^{-3}	—
Test trajectories	64	32	32
Eval. horizon (steps)	200	200	20
APIC candidate settings (policy-dependent)			
FAPC gate $(a_{\mathrm{lo}}, a_{\mathrm{hi}}, k_c, \delta)$	$(1.0,\ 0.6,\ 0.12,\ 0.01)$		
Sub-step count M	100–400		
Refinement modes / gate ϵ	24 / 0.05		

Table 5: Reported benchmark configuration. †Wave is integrated at $\Delta t=10^{-3}$ and stored every 10 steps. All nine networks are one-step predictors (Adam, 3×10^4 iterations; architectures in the text); corrector constants are fixed once on the seed-0 validation rollout. Reported numbers average over 3 seeds with non-overlapping shards.

C Per-Architecture Long-Rollout Results

Table 6 reports per-seed relative L_2 errors for every cell under S1 per-pair selection. The reported APIC error is below PhysicsCorrect in all 27 evaluated seed–cell combinations.

Figure 7 visualises the per-step trajectories of the NS row of Table 6.

Figure 8 extends the NS-FNO seed-0 trajectory to 1000 steps and reports final-step errors for all three architectures. APIC retains lower error over this evaluation horizon. The final-step statistics in the figure and the horizon averages in Table 7 measure different aspects of the rollout.

KS chaotic-horizon saturation (detailed trajectories)

Figure 9 unfolds the KS-FNO chaos-saturation summary of Fig. 10a by showing the full per-step error trajectory at $n_t \in \{50, 100, 200, 400, 600\}$. Both PhysicsCorrect (blue shades) and APIC (orange shades) trajectories rise toward the same saturation plateau around ~ 1.0 past the Lyapunov time ($n_t \sim 200$). Both methods therefore exhibit a similar limitation over these trajectories; the observation does not identify a method-independent error bound.

	PhysicsCorrect (PC)			APIC (Ours)		
	Seed 0	Seed 1	Seed 2	Seed 0	Seed 1	Seed 2
KS-FNO	$6.01\cdot10^{-2}$	$6.08\cdot10^{-2}$	$6.81\cdot10^{-2}$	$1.53\cdot10^{-2}$	$1.12\cdot10^{-2}$	$1.71\cdot10^{-2}$
KS-U-Net	$5.84\cdot10^{-2}$	$5.59\cdot10^{-2}$	$6.06\cdot10^{-2}$	$1.12\cdot10^{-2}$	$9.17\cdot10^{-3}$	$2.74\cdot10^{-2}$
KS-ViT	$6.87\cdot10^{-2}$	$6.67\cdot10^{-2}$	$7.99\cdot10^{-2}$	$1.76\cdot10^{-2}$	$1.35\cdot10^{-2}$	$2.51\cdot10^{-2}$
NS-FNO	$4.01\cdot10^{-4}$	$4.01\cdot10^{-4}$	$4.01\cdot10^{-4}$	$2.66\cdot10^{-5}$	$2.66\cdot10^{-5}$	$2.66\cdot10^{-5}$
NS-U-Net	$3.99\cdot10^{-4}$	$3.99\cdot10^{-4}$	$3.99\cdot10^{-4}$	$2.66\cdot10^{-5}$	$2.66\cdot10^{-5}$	$2.66\cdot10^{-5}$
NS-ViT	$4.01\cdot10^{-4}$	$4.01\cdot10^{-4}$	$4.01\cdot10^{-4}$	$2.66\cdot10^{-5}$	$2.66\cdot10^{-5}$	$2.66\cdot10^{-5}$
Wave-FNO	$2.82\cdot10^{-3}$	$2.82\cdot10^{-3}$	$2.82\cdot10^{-3}$	$4.80\cdot10^{-4}$	$4.77\cdot10^{-4}$	$4.85\cdot10^{-4}$
Wave-U-Net	$2.82\cdot10^{-3}$	$2.82\cdot10^{-3}$	$2.82\cdot10^{-3}$	$4.79\cdot10^{-4}$	$4.67\cdot10^{-4}$	$4.72\cdot10^{-4}$
Wave-ViT	$2.82\cdot10^{-3}$	$2.82\cdot10^{-3}$	$2.82\cdot10^{-3}$	$4.81\cdot10^{-4}$	$4.77\cdot10^{-4}$	$4.80\cdot10^{-4}$

Table 6: Per-seed relative L_2 errors for PhysicsCorrect (PC) and APIC under S1 selection on all nine cells. NS errors coincide at the reported precision; the analytical update does not use surrogate weights.

Arch. (NS, $n_t=1000$)	PhysCorr.	APIC	Reduction
FNO	$9.435\cdot10^{-4}$	$\mathbf{1.475\cdot10^{-4}}$	$84.36\,\%$
U-Net	$9.785\cdot10^{-4}$	$\mathbf{1.475\cdot10^{-4}}$	$84.92\,\%$
ViT	$9.437\cdot10^{-4}$	$\mathbf{1.475\cdot10^{-4}}$	$84.37\,\%$
Mean	$9.552\cdot10^{-4}$	$\mathbf{1.475\cdot10^{-4}}$	$\mathbf{84.55}\,\%$

Table 7: NS evaluation at n_t=1000. Entries report horizon-averaged relative-L_2 errors and APIC's reduction over PhysicsCorrect for each architecture. Per-step behaviour is shown in Figure 8.

Figure 10 collects the three per-PDE diagnostics referenced from the main body: the KS chaos-horizon attenuation (a, the same saturation phenomenon quantified above), the Wave Numerov-vs-Verlet order-of-accuracy gap (b), and the NS power-spectrum match (c).

Qualitative KS space–time. Figure 11 renders the KS field $u(x,t)$ over the rollout as a space–time diagram: the qualitative counterpart of the per-step trajectories in Fig. 9.

The same comparison in *error* space (Figure 12) localises where each corrector first loses the trajectory.

D One-Step Correction and PDE Residual

Table 10 replicates the one-step analysis of Huang & Perdikaris (2026)'s Fig. 5 with our method added. APIC reduces the one-step relative L_2 error on NS-FNO from $5.97\cdot10^{-3}$ (uncorrected) to $3.84\cdot10^{-7}$, a$\sim1.5\cdot10^{4}\times$ reduction and $14\times$ improvement over PhysicsCorrect at the same step. The CD2-residual is slightly higher for APIC than for PhysicsCorrect (0.036 vs 0.014) because Strang splitting does not

PDE / Arch.	Method	ms/step	Peak MB	vs. uncorr.
NS, FNO	uncorrected	0.96	185	1.00×
	PhysicsCorr.	1.29	185	1.34×
	APIC (Strang)	**0.25**	**173**	**0.26**×
NS, U-Net	uncorrected	2.45	478	1.00×
	PhysicsCorr.	2.74	478	1.12×
	APIC (Strang)	**0.25**	**401**	**0.10**×
NS, ViT	uncorrected	0.99	143	1.00×
	PhysicsCorr.	1.38	143	1.39×
	APIC (Strang)	**0.25**	**142**	**0.25**×
KS, FNO	uncorrected	0.37	13.5	1.00×
	PhysicsCorr.	0.56	13.5	1.52×
	APIC (FAPC)	0.58	13.5	1.57×

Table 8: Per-step inference latency and peak GPU memory on a single RTX-4090, batch 1. The NS Strang branch is $4-10\times$ *faster* than the uncorrected baseline because it bypasses the network forward entirely.

Cell	Mean gain	Std. (pp)	Category
NS-FNO	93.38	0.00	Deterministic
NS-U-Net	93.35	0.00	Deterministic
NS-ViT	93.37	0.00	Deterministic
Wave-FNO	82.93	0.15	Quasi-deterministic
Wave-U-Net	83.22	0.21	Quasi-deterministic
Wave-ViT	82.99	0.07	Quasi-deterministic
KS-FNO	77.03	4.04	Stable
KS-U-Net	73.04	14.20	Variable
KS-ViT	74.25	5.59	Stable

Table 9: Reported seed variance per cell. NS entries have zero recorded variance. Wave entries exhibit smaller reported variation than KS. All evaluated seed–cell combinations exceed a 20-percentage-point error reduction.

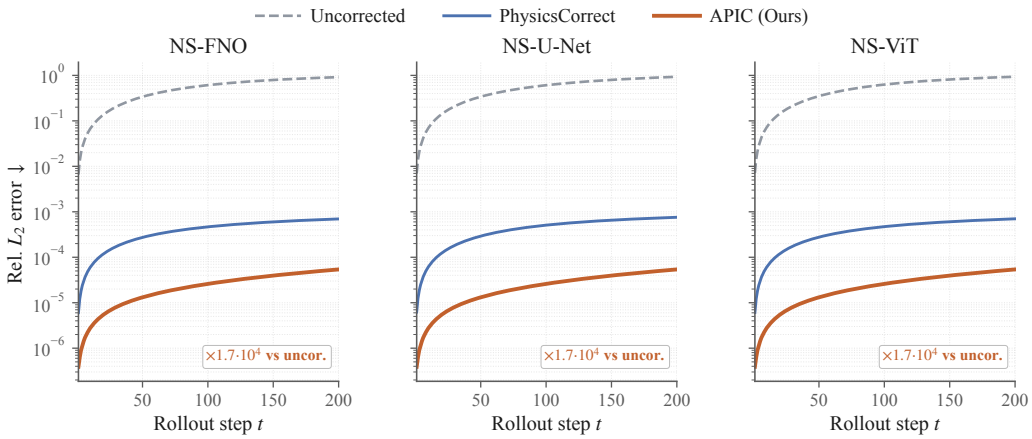

Figure 7: Per-architecture NS per-step error trajectories at n_t=200. Across FNO, U-Net, and ViT the Uncorrected baseline grows to$\sim$1.0, PhysicsCorrect plateaus at$\sim 7\cdot10^{-4}$, and APIC's Strang branch reaches$\sim 5\cdot10^{-5}$ identically at the reported precision across architectures.

explicitly minimise the finite-difference residual used in the linearised correction; the smaller error shows that APIC has lower error relative to the reference despite a larger CD2 residual.

Method (NS-FNO, one step)	Rel. L_2 error	CD2 residual
Uncorrected	$5.97\cdot10^{-3}$	11.07
PhysicsCorrect	$5.44\cdot10^{-6}$	**0.014**
APIC (Ours)	$\mathbf{3.84\cdot10^{-7}}$	0.036

Table 10: One-step relative-L_2 error and CD2 residual on NS-FNO. APIC has the lowest reference error among the listed methods, whereas PhysicsCorrect has the lowest CD2 residual.

E Classical-Baseline Comparison on Navier–Stokes

Baseline implementations and evaluation scope. The additional NS comparisons include ETDRK4 (Kassam & Trefethen, 2005) and an ANCHOR-style residual-triggered hybrid (Roy et al., 2025). ETDRK4 advances the state without using the surrogate. The hybrid uses the same analytical NS update as APIC and varies its activation rate. The unrolled/pushforward baseline is evaluated after retraining the surrogate with rollout lengths K=8, 6, and 2 for NS, Wave, and KS, respectively. Results for methods not evaluated under this protocol are included only in the capability comparison.

The unrolled NS baseline reduces error from 0.56 to 6.5×10^{-3}, while retaining higher error than the inference-time corrections in Table 2. The Wave and KS results indicate that its benefit varies with equation and architecture. These comparisons concern the implemented configurations and do not establish an ordering over all training-based stabilisation methods.

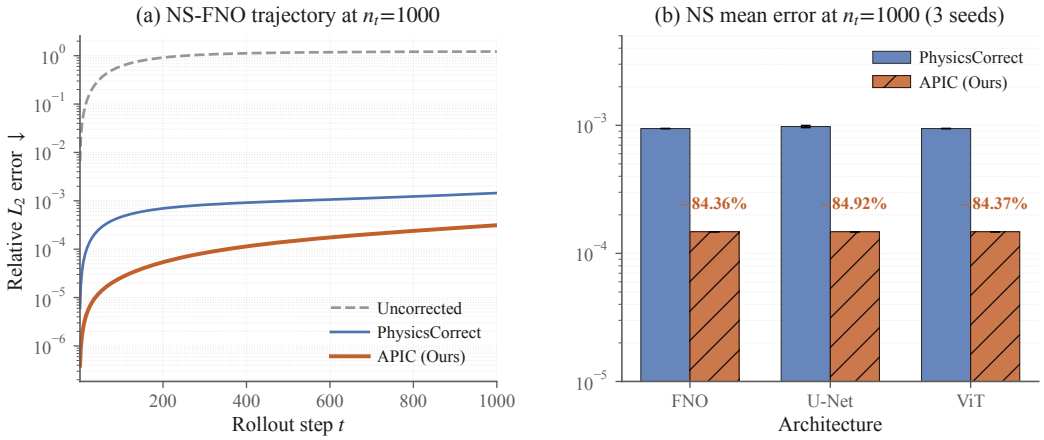

Figure 8: NS rollout at $n_t = 1000$. (a) Per-step relative L_2 on NS-FNO seed 0. (b) Final-step error per architecture (mean±std over 3 seeds); percentage labels are APIC's reduction over PhysicsCorrect.

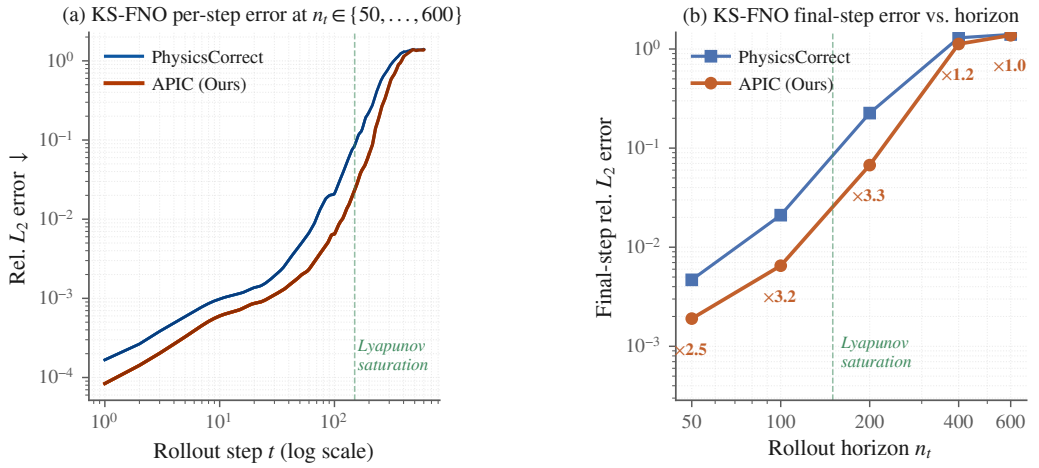

Figure 9: KS-FNO error vs. rollout horizon. (a) Per-step trajectories at $n_t \in \{50, 100, 200, 400, 600\}$; both PhysicsCorrect and APIC plateau near the saturation level past the Lyapunov time. (b) Final-step error vs. horizon, with APIC's improvement factor annotated; gain is $\sim \times 2.5$–3.3 pre-saturation and approaches $\sim \times 1.0$ at the longest tested horizons.

Figure 13 reports the NS error trajectories and measured costs. At the final step, PhysicsCorrect and ETDRK4 errors are approximately 12.9 and 3.8 times the APIC error, respectively. APIC also has lower per-step latency among the configurations shown.

Residual-triggered and unconditional correction. The ANCHOR-style implementation uses the analytical NS step employed by APIC, with a residual-based criterion controlling its activation. At a $100\,\%$ activation rate the two update procedures coincide; this equivalence alone does not establish that full activation is optimal over all possible schedules.

In the reported sweep (Figure 14), error decreases as activation approaches $100\,\%$. At the $50\,\%$ budget, errors are 6.5×10^{-2}, 2.4×10^{-1}, and 1.5×10^{-1} for FNO, U-Net, and ViT, respectively. Thus the evaluated compute-saving configurations retain substantially higher error than unconditional correction on NS.

At the same budget, the KS-FNO and Wave-FNO hybrids yield 7.3×10^{-2} and 6.0×10^{-2}, compared with 1.45×10^{-2} and 4.81×10^{-4} for the selected APIC variants. Their update and refinement choices differ, so these results do not isolate activation frequency as a causal factor.

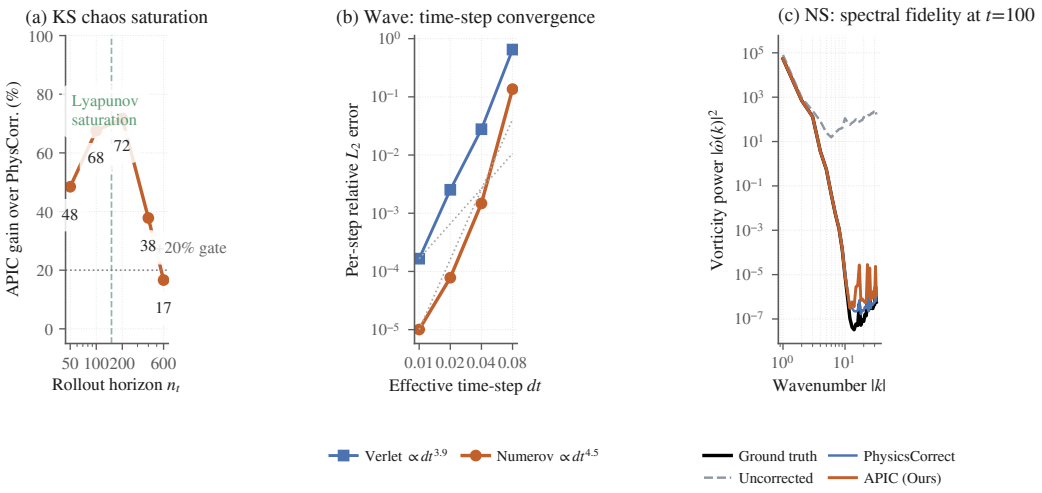

Figure 10: Diagnostics of horizon dependence and numerical fidelity. (a) KS-FNO seed 0: reduction over PhysicsCorrect decreases from 66 % at n_t=50 to approximately 2 % at n_t=600. (b) Wave stencil errors under a time-step sweep. (c) NS vorticity spectra at t=100, compared with the reference.

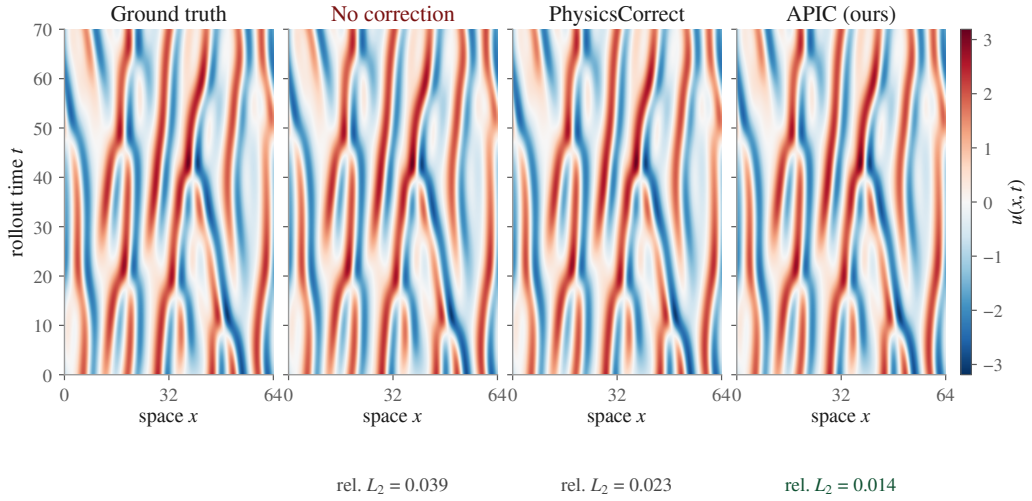

Figure 11: KS space–time gallery $u(x,t)$ on the seed-0 trajectory over the first 280 rollout steps (space horizontal, rollout time vertical; shared symmetric colour scale from the ground truth). The lower (early) rows match across all panels; in the upper (late) rows the No-correction and then the PhysicsCorrect fields visibly decorrelate from the ground-truth flame-front pattern, while APIC retains closer agreement for a longer interval. The displayed trajectories remain bounded; mean relative-L_2 errors are annotated.

Spectral-filtering baseline. We evaluate the filter $\hat{u} \leftarrow \exp[-\alpha(|k|/k_{\max})^{2p}]\hat{u}$ after each neural update, varying α and p across the nine cells. The largest reported reduction is at most 10 % on Wave-FNO. On KS, the selected setting is α=0, corresponding to no filtering; changes on NS are small. Within this sweep, output filtering is less effective than the evaluated dynamical corrections. The comparison is restricted to the filter family and parameter sweep considered here.

F Computational Cost Scaling

The NS analytical step consists of spectral transforms and pointwise operations. An FFT on an $N \times N$ grid has complexity $\mathcal{O}(N^2 \log N)$; total cost also depends on the number of transforms, batching, and implementation. KS sub-stepping repeats its inner operations M times. These operation counts should be considered separately from measured GPU latency.

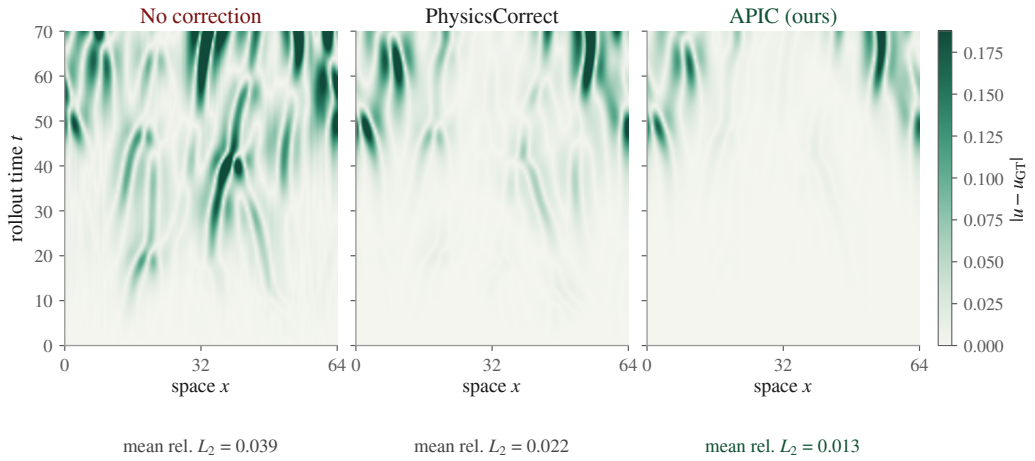

Figure 12: KS pointwise error $|u - u_{\mathrm{GT}}|(x, t)$ on the seed-0 trajectory (shared colour scale). The uncorrected rollout accrues error earliest and most broadly; APIC stays closest to the ground truth for the most rollout time (lowest mean relative L_2). All methods eventually decorrelate at the chaotic horizons considered here; no claim is made about untested correctors.

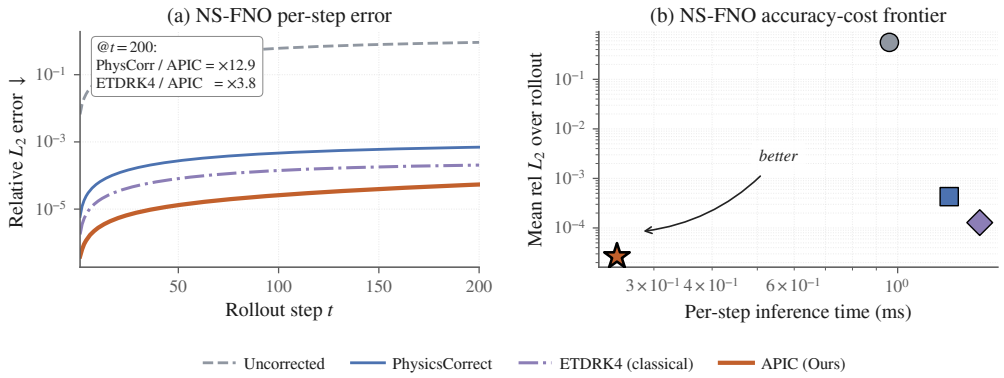

Figure 13: NS-FNO comparison with the evaluated baselines. (a) Per-step relative-L_2 errors over n_t=200 steps. At the final step, PhysicsCorrect and ETDRK4 errors are approximately 12.9 and 3.8 times the APIC error, respectively. (b) Error and measured inference cost for the displayed configurations.

Figure 15 reports the measured batch-1 NS profile. PhysicsCorrect increases latency by approximately 12–39 % over the uncorrected model. The analytical branch takes approximately 0.25 ms per step, corresponding to the reported 3.8–9.8 speedup factors.

Discretisation and resolution. The NS spectral update can be evaluated on a coarser grid by rebuilding its spectral operators. Figure 16 reports one-step error under $1/2/4\times$ spectral downsampling. At $2\times$ downsampling, 25 % of the Fourier modes are retained and error increases from 5.7×10^{-7} to 1.8×10^{-6}. The absolute error remains small in this test, while $4\times$ downsampling produces a larger increase. This finite resolution sweep characterises sensitivity of the analytical update.

G Sensitivity Analyses

The learned FAPC mask. Figure 17 illustrates the frequency-dependent gain for three parameter settings. The curves retain greater gain at low normalised wavenumbers and attenuate correction at higher wavenumbers. They describe the mask's parameterisation, not independent evidence of seed-level fitting behaviour. Setting $a_{\mathrm{lo}} = a_{\mathrm{hi}} = 1$ gives the identity mask; with residual and Jacobian settings unchanged, this recovers the unmasked residual correction.

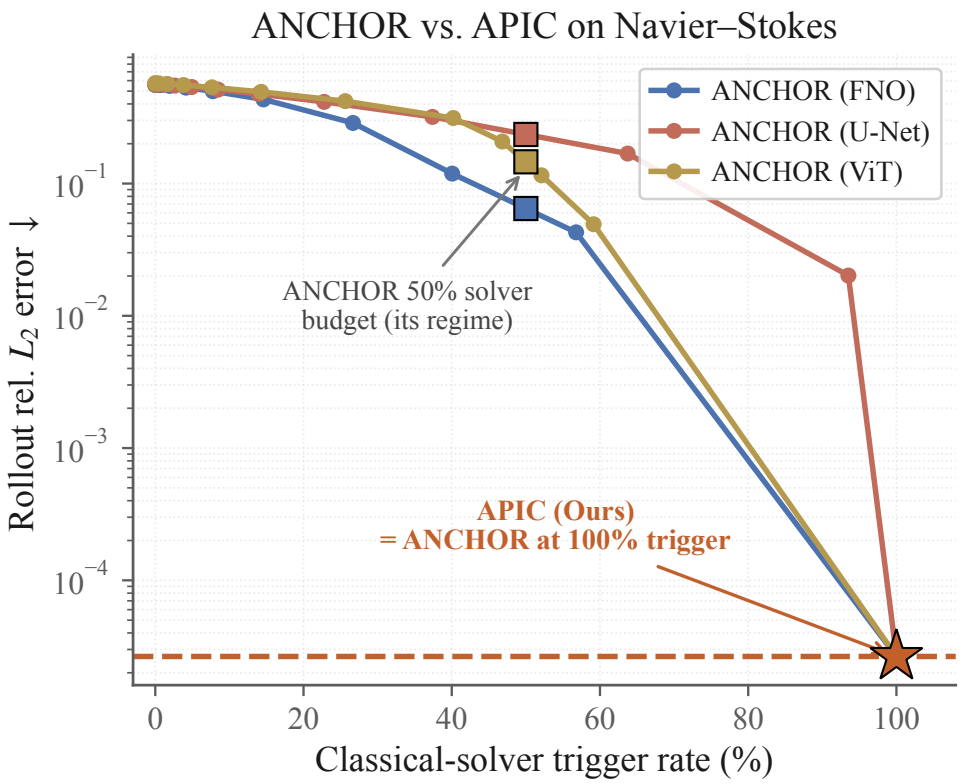

Figure 14: ANCHOR-style NS evaluation using the same analytical step as APIC. The curves show rollout error as the classical-update activation rate varies. Stars mark full activation, which coincides with APIC; squares mark the evaluated 50 % budget.

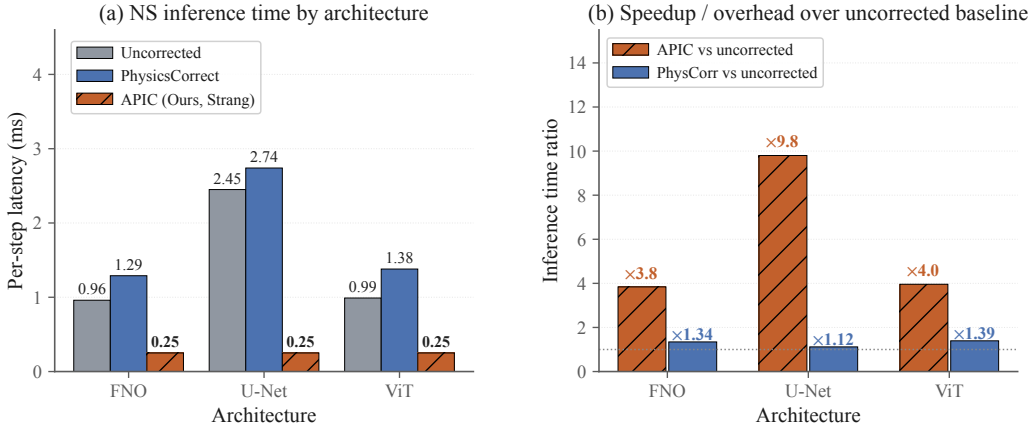

Figure 15: NS inference-time profile on a single RTX-4090 (batch 1, 200 warm-up steps). (a) Per-step latency, grouped by architecture. (b) Ratio vs. uncorrected: APIC is $\times 3.8 - 9.8$ *faster* (because Strang bypasses the network), PhysicsCorrect is $\times 1.12 - 1.39$ *slower* (because of its Jacobian-vector product on top of the network forward).

Figure 18 reports the two design-rule sweeps referenced from the main body. (a) On KS-FNO, sweeping the cached-Jacobian count K shows the original PhysicsCorrect single-Jacobian rule ($K = 1$) is within 0.5 % of the best—additional cached Jacobians provide limited improvement in this sweep. (b) The sub-step count M saturates at $M \approx 100$ for U-Net and $M \approx 400$ for ViT.

Figure 19 varies individual FAPC parameters around the selected configuration. The reported error changes remain within 50 % over the indicated 2–4× ranges. The sub-step sweep in Figure 18(b)

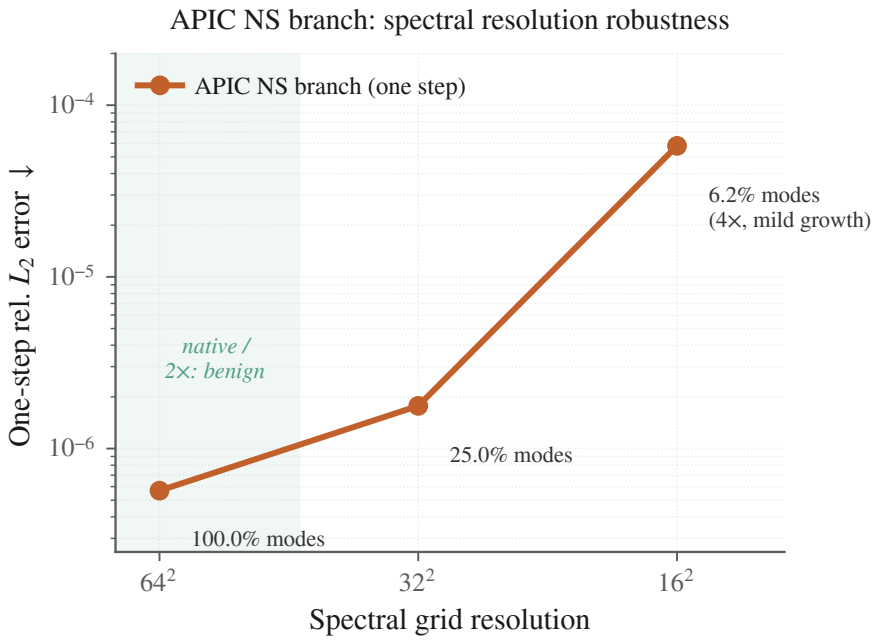

Figure 16: One-step relative-L_2 error for the analytical NS update under spectral downsampling. Labels indicate the fraction of retained Fourier modes. The plot reports sensitivity at the tested resolutions.

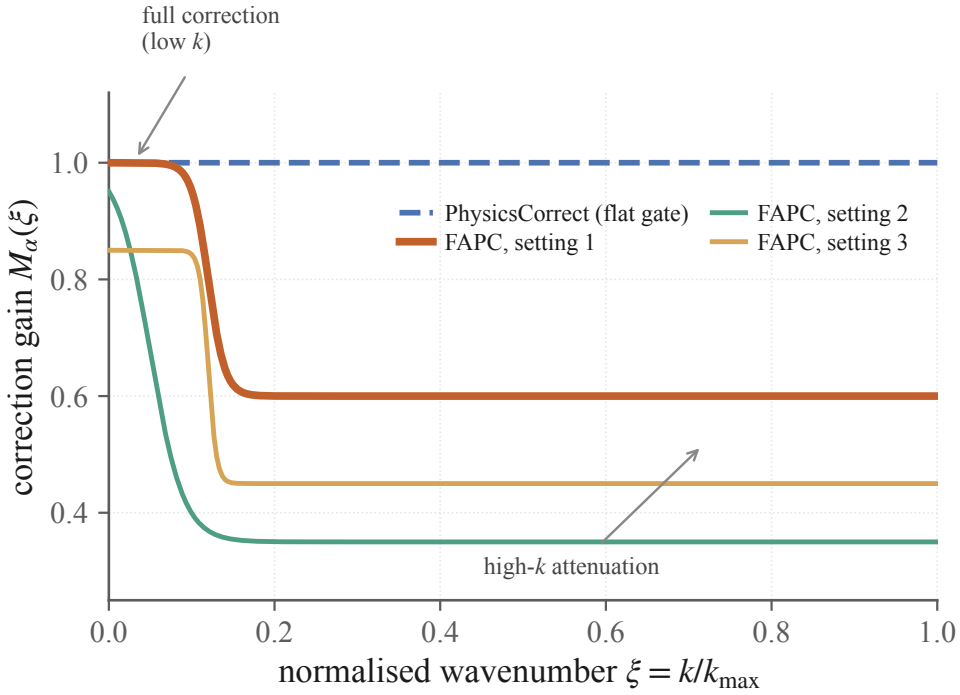

Figure 17: Frequency-adaptive gain for three illustrative parameter settings, compared with the identity mask. The curves show how the low- and high-frequency gains and transition parameters affect the correction; they are not presented as per-seed estimates.

separately evaluates the dependence on M; additional steps give diminishing improvements over the tested values, with architecture-dependent selected settings.

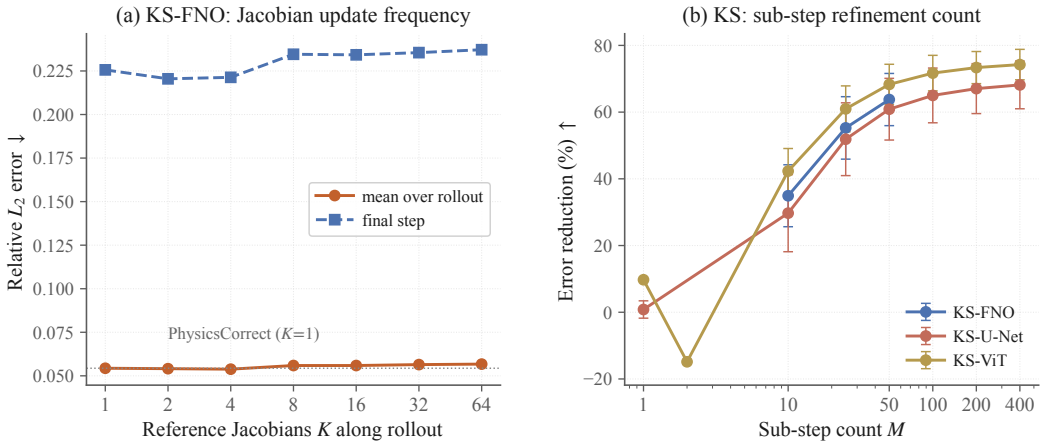

Figure 18: (a) KS-FNO: the original PhysicsCorrect $K=1$ rule is within $0.5\,\%$ of the lowest error in the sweep. (b) KS: selected sub-step counts differ across architectures in the tested candidates.

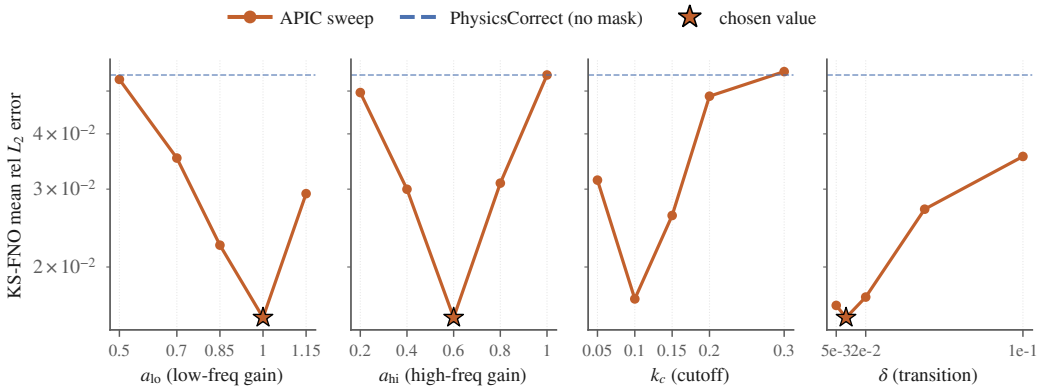

Figure 19: FAPC mask sensitivity on KS-FNO around the chosen parameter values (stars). The dashed line denotes PhysicsCorrect. An identity gate (a_{lo}=a_{hi}=1), with the remaining settings unchanged, recovers the unmasked correction.

Figure 20 compares component configurations within each PDE branch. The relative benefit of an analytical update and its refinement depends on the equation and the evaluated variant. These comparisons complement the complete-policy results in Table 3.

Sensitivity to input perturbations. The NS-FNO evaluation perturbs a reference state with uncorrelated Gaussian or spatially correlated Gaussian-random-field (GRF) noise, with relative-L_2 magnitudes from 10^{-3} to 1.0, before applying one correction step (Figure 21).

The two methods have similar displayed errors under uncorrelated noise. For GRF noise at ϵ=1, PhysicsCorrect gives an error of 0.998 and APIC gives 0.830, compared with an injected error of 1.001. The analytical update is therefore more effective in this particular perturbation setting. The measured benefit therefore depends on the perturbation structure and magnitude in this experiment.

H Routing Rule and Generalisation Discussion

The evaluated router is a fixed mapping from the known equation class to a correction family (Table 1). Extending the mapping requires considering the temporal formulation, the linear and nonlinear operators, and the reference discretisation. These properties motivate candidate updates but do not certify their stability or accuracy on a new equation. Any extension also requires appropriate treatment of boundary conditions, nonlinear solves, and parameter regimes.

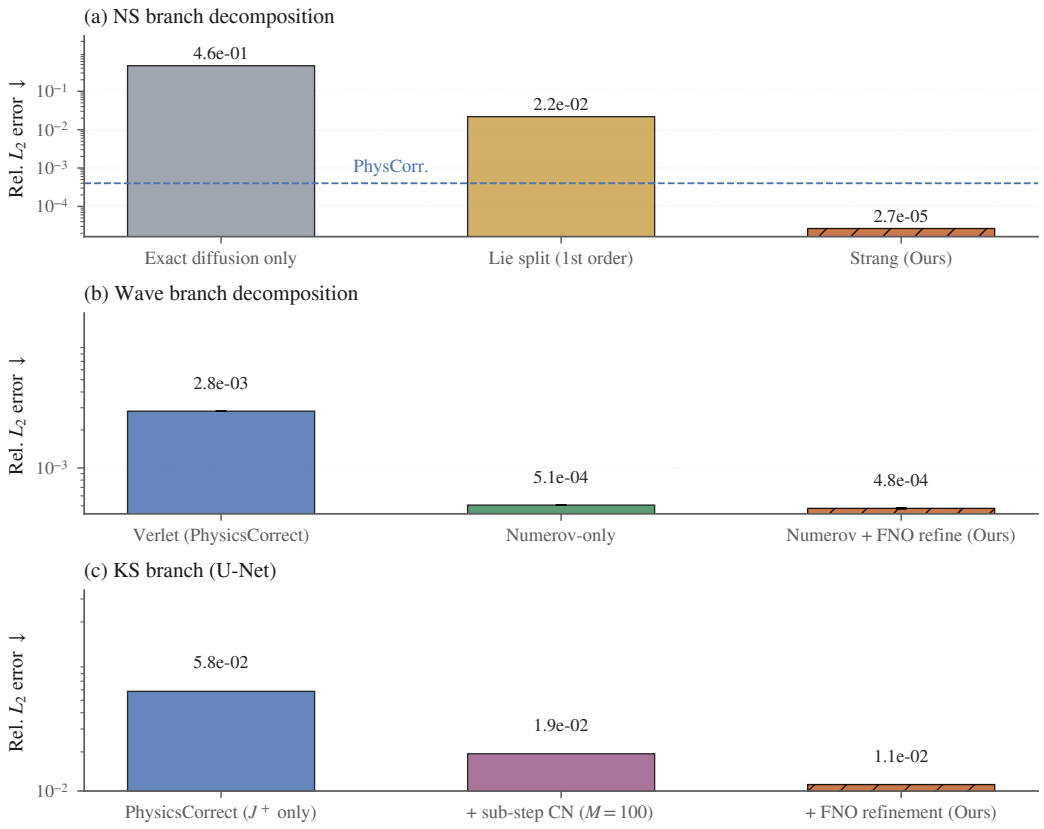

Figure 20: Component configurations by PDE branch. (a) NS: the symmetric split has lower error than the evaluated Lie split. (b) Wave: Numerov reduces error relative to Verlet; refinement provides an additional approximately 5 % reduction. (c) KS: errors for the displayed combinations of correction components.

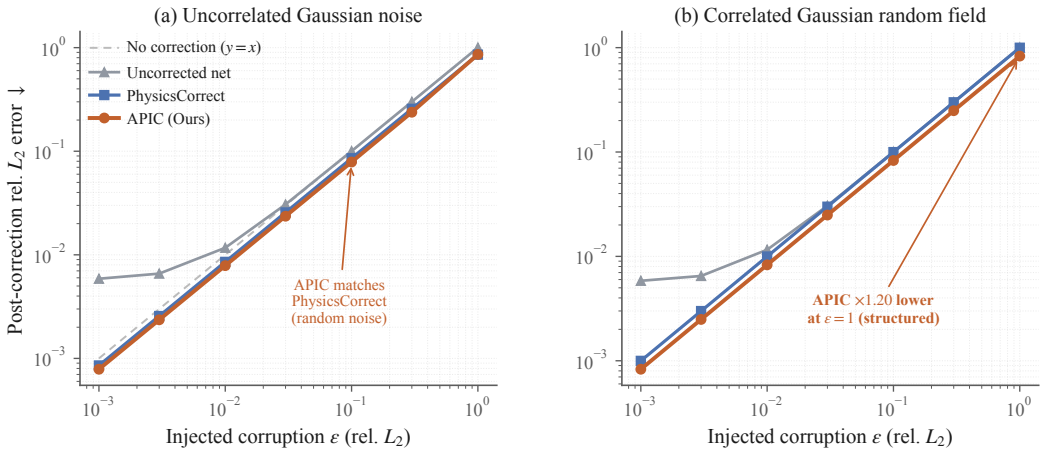

Figure 21: Error-magnitude robustness on NS-FNO (replicating PhysicsCorrect B.6/Fig. 14). Post-correction relative-L_2 error vs. injected corruption ϵ for (a) uncorrelated Gaussian and (b) correlated GRF noise. Lower is better; the dashed grey line is the no-correction reference $y = x$. The methods have similar displayed errors under uncorrelated noise; APIC has lower error for the largest correlated perturbation shown.

Learning a routing policy from rollout statistics is a possible extension. Its performance would need to be evaluated on equations or configurations excluded from policy selection. This is distinct from the present comparison of three predefined equation classes.

I Implementation Details

This appendix expands on the corrector implementations used in APIC's three branches, including the spectral operators and update sequence. The reported GPU measurements use an NVIDIA RTX-4090 (24 GB) with PyTorch 2.

Spectral operators. For an $N \times N$ periodic 2-D grid we precompute the wave-vector arrays $K_x, K_y \in \mathbb{R}^{N\times N}$ with components $k = 2\pi\,\mathrm{shift}(\mathrm{arange}(N))$ and the squared magnitude $|k|^2 = K_x^2 + K_y^2$. A $\frac{2}{3}$-rule dealiasing mask $D(k) = \mathbb{1}\{|K_x|, |K_y| \le \frac{2}{3}\cdot N/2\}$ is applied to all nonlinear advection terms. These tensors are stored once in GPU memory. Both their storage and the transform cost depend on grid resolution.

NS Strang branch. We work in the vorticity–stream-function formulation $w_t = \nu\Delta w + N(w)$, $w = -\Delta\psi$. The implementation in `strang_split_ns.py` performs, per step:

1. Half-step exact diffusion in Fourier space: $\hat{w} \leftarrow \exp(-\nu|k|^2\Delta t/2)\,\hat{w}$.
2. Recover $\hat{\psi} = \hat{w}/|k|^2$ (with $\hat{\psi}(0) = 0$), build $\hat{u} = i\,2\pi K_y\,\hat{\psi}$, $\hat{v} = -i\,2\pi K_x\,\hat{\psi}$, $\hat{\nabla w}$ in the same way, and form the advection $F = u\,\partial_x w + v\,\partial_y w$ in physical space.
3. Apply dealiasing: $\hat{F} \leftarrow D(k)\,\hat{F}$. Add the forcing: $w \leftarrow w + \Delta t(-F + f)$.
4. Half-step exact diffusion again. Recover ψ from w for the next step.

The branch is fully analytical—the network forward f_θ is *not* called inside the loop. For fixed inputs, numerical parameters, and execution conditions, the update is independent of the surrogate training seed.

Wave Numerov branch. For $u_{tt} = c^2\nabla^2 u$ the Numerov stencil in Fourier space is the diagonal recurrence $\hat{u}_k^{n+1}(1 + a_k) = (2 - 10a_k)\hat{u}_k^n - (1 + a_k)\hat{u}_k^{n-1}$, $a_k = c^2\Delta t^2|k|^2/12$, implemented in `numerov_wave.py`. The bounded learned refinement is a small Fourier Neural Operator (FNO) with width 24, 10 modes per direction, depth 3 (about 70 k parameters) that emits $\epsilon\,\tanh(\cdot)$ with $\epsilon = 0.05$; this guarantees the correction has bounded amplitude regardless of network outputs.

KS branch. On the 1-D periodic domain $[0, L]$ ($L = 64$, $N = 512$ grid points, $\Delta t = 0.05$), the chosen KS corrector per architecture is: KS-FNO uses the multi-reference FAPC corrector; KS-U-Net uses sub-step CN with M=100 and an FNO refinement head; KS-ViT uses sub-step CN with M=400. The FAPC logistic gate uses $(a_{\mathrm{lo}}, a_{\mathrm{hi}}, k_c, \delta) = (1.0, 0.6, 0.12, 0.01)$ (low-/high-frequency gain, transition centre and width in normalised wavenumber), all fit on a single held-out trajectory.

J Compute Resources and Wall-Clock Time

Table 11 reports wall-clock times for both the one-time warm-up (Jacobian precomputation for PhysicsCorrect / spectral operator caching for APIC) and the per-step inference, all measured on one RTX-4090 at batch size 1 following a 200-step warm-up phase.

K Summary Diagnostic Table

Table 12 consolidates the headline statistics of APIC under S1 per-pair selection into a single per-cell row: mean / std relative-L_2 error, reduction over PhysicsCorrect, and the reported seed-variability category.

L Reproducibility

Data and checkpoints. The correction evaluation uses frozen checkpoints and held-out trajectories for the three PDEs: KS (512 points, Δt=0.05, generated with the ETDRK4 scheme of Kassam & Trefethen, 2005), NS (64×64 vorticity, Δt=0.01, viscosity ν=10^{-3}), Wave (128×128, Δt=10^{-4}, saved every 100 steps). No retraining of the surrogate is performed during inference-time correction.

	Warm-up	ms/step	GPU MB
Uncorrected baselines			
NS-FNO	—	0.96	185
NS-U-Net	—	2.45	478
NS-ViT	—	0.99	143
KS-FNO	—	0.37	13.5
PhysicsCorrect (cached J^+)			
NS-FNO	8.7 s	1.29	185
NS-U-Net	8.7 s	2.74	478
NS-ViT	8.7 s	1.38	143
KS-FNO	0.4 s	0.56	13.5
APIC (Ours)			
NS-* (Strang)	0.05 s	0.25	142–401
KS-FNO (FAPC)	0.4 s	0.58	13.5
KS-U-Net (subCN+FNO, M=100)	0.4 s	56	95
KS-ViT (subCN, M=400)	—	224	95
Wave-* (Numerov+FNO)	—	0.45	64

Table 11: Wall-clock cost on a single RTX-4090 (24 GB). The NS Strang branch warms up in 50 ms because it only needs to cache spectral operators (no Jacobian inversion), and is $4-10\times$ faster than the uncorrected baseline at inference because it bypasses the network forward entirely. KS-U-Net and KS-ViT pay an M-fold cost for the sub-step spectral CN integration (M=100 and 400 respectively). This table reports cache setup and inference, not auxiliary training.

Cell	APIC mean rel L_2	std (pp)	gain (%)	seed-invariant
NS-FNO	$2.66\cdot10^{-5}$	0.00	93.4	yes
NS-U-Net	$2.66\cdot10^{-5}$	0.00	93.3	yes
NS-ViT	$2.66\cdot10^{-5}$	0.00	93.4	yes
Wave-FNO	$4.81\cdot10^{-4}$	0.15	82.9	near
Wave-U-Net	$4.73\cdot10^{-4}$	0.21	83.2	near
Wave-ViT	$4.79\cdot10^{-4}$	0.07	83.0	near
KS-FNO	$1.45\cdot10^{-2}$	4.04	77.0	no
KS-U-Net	$1.59\cdot10^{-2}$	14.20	73.0	no
KS-ViT	$1.87\cdot10^{-2}$	5.59	74.2	no
Mean	—	—	**83.7**	—

Table 12: S1 per-pair summary. "yes" denotes identical recorded errors across seeds; "near" denotes $\sigma\leq0.25$ percentage points; "no" denotes greater observed variability. These categories describe the reported error statistics, not bitwise equality of state trajectories.

Software stack. PyTorch 2.0, CUDA 11.8, Python 3.10. All FFTs use the standard `torch.fft.fft2` on the GPU. Random seeds 0, 1, 2 control only the shard of the test trajectories used by each run; the Strang branch is seed-independent by construction.

End-to-end reproduction. The full headline table is reproducible by running one aggregator script per PDE branch. Below we show the NS row as an example (remaining commands follow the same pattern; see code release for full details):

```
# NS branch: analytical Strang split, no training
for arch in fno unet vit; do
 for s in 0 1 2; do
  python strang_split_ns.py --arch $arch \
   --data data/ns_mini_Lt10.pt \
   --init_net ckpt/${arch}_ns_v40_seed${s}.pth \
   --seed $s --test_steps 200 \
   --out metrics/ns_${arch}_seed${s}.json
 done
done
```

The stored NS error values agree to the reported precision across seeds (Table 14). Equality of these aggregate values should be distinguished from a bitwise comparison of rollout tensors.

M Limitations and Threats to Validity

Chaotic-horizon limit. On KS, APIC's gain attenuates as the rollout horizon approaches the Lyapunov time (Fig. 10a): from $\sim 66\%$ at n_t=50 to $\sim 2\%$ at n_t=600. PhysicsCorrect degrades

with a similar trend in this evaluation. The n_t=200 headline results therefore do not characterise performance at arbitrary horizons.

KS-U-Net seed variance. Across our three seeds the KS-U-Net cell shows seed-0/1 gains of $\sim 82\%$ and a seed-2 gain of $\sim 55\%$ (mean 73.04%, $\sigma = 14.20$ pp). This spread limits conclusions based on the mean alone; three seeds are insufficient to characterise the full distribution of outcomes.

Wave test horizon. Our Wave test set is limited to $n_t = 15$–20 saved time steps by the released data; the observed stencil convergence does not establish the error reduction at longer horizons. Longer reference trajectories are required to evaluate that regime.

PDE scope. We validate on the three PDE families of the standard PhysicsCorrect benchmark. Generalisation to advection-diffusion-reaction systems, shock-forming hyperbolic equations, and stochastic PDEs is an open question; the modularity of Eq. (2) suggests adding new branches as new PDE classes are encountered.

N Theoretical Notes

Order-of-accuracy analysis for the three branches. Each APIC branch is a routed numerical-analysis primitive whose truncation order is classical and can be inspected directly:

NS Strang split. The composition $e^{L\Delta t/2} \circ e^{N\Delta t} \circ e^{L\Delta t/2}$ has truncation error $\frac{\Delta t^3}{24}\big([L,[L,N]] - 2[N,[L,N]]\big) + O(\Delta t^5)$ for non-commuting operators L and N (Strang, 1968). Because we evaluate $e^{L\Delta t/2}$ exactly in Fourier space (the diagonal multiplier $\exp(-\nu|k|^2\Delta t/2)$), the diffusion substep introduces no approximation to that linear flow. The formal splitting expansion assumes exact subflows; the order of the implemented method also depends on the advection update. An explicit Euler substep does not by itself justify the ideal composition order. Reference agreement is assessed empirically here, separately from a convergence claim about the complete implementation.

Wave Numerov. The Numerov recurrence $(1+a_k)\hat{u}^{n+1} = (2-10a_k)\hat{u}^n - (1+a_k)\hat{u}^{n-1}$ has local truncation error $\frac{1}{240}\Delta t^6 \partial_t^6 u + O(\Delta t^8)$, hence global $O(\Delta t^4)$ for the wave equation, two orders above Verlet's $O(\Delta t^2)$. The time-step sweep in Figure 10(b) reports errors consistent with the expected distinction between the stencils.

KS sub-step CN. The single-step Crank–Nicolson scheme applied to the linear part of KS has $O(\Delta t^2)$ truncation; running M sub-steps reduces it to $O((\Delta t/M)^2)$ per outer step. The FAPC mask is a diagonal logistic gate $M_\alpha(k) = a_{\text{lo}} + (a_{\text{hi}} - a_{\text{lo}})\,\sigma((\xi - k_c)/\delta)$ in the normalised wavenumber $\xi = |k|/|k|_{\max}$ (function `make_sig` in our code) that does reweights the residual from gain a_{lo} to a_{hi}. This reweighting does not, by itself, establish the convergence order of the complete corrected rollout.

Interpreting the policy gaps. The numerical mismatch between a corrector and a reference generator motivates equation-dependent updates, but it is not a lower bound on what a learned model can achieve. The measured S2–S3 difference is 10.92 percentage points; 11.17 points is the S1–S3 difference. Because these policies change multiple components, neither difference identifies an isolated routing effect. Likewise, S2–S4 cannot isolate learning while their KS sub-step counts differ. The defensible empirical finding is that coarse routing approaches per-pair selection and that analytical-only variants provide a strong baseline.

O Detailed Routing Diagnostics

The following structural properties can guide candidate selection for equations beyond the evaluated set. They are design heuristics, not sufficient conditions for transferring a branch without validation.

- (D1) *Temporal formulation.* A linear second-order evolution suggests considering a wave-type recurrence, subject to the spatial operator and boundary conditions.
- (D2) *Operator structure.* A decomposition $u_t = Lu + N(u)$ with a stiff linear component suggests a split or semi-implicit update. Its accuracy depends on how both subproblems are advanced.

(D3) *Sensitivity and scale separation.* Chaotic or hyperdiffusive dynamics motivate examining time-step sensitivity and frequency-dependent corrections. Their usefulness must be tested at the relevant horizon and resolution.

These properties define a candidate set for evaluation. Automated selection would additionally require a validation objective and a protocol separating selection from final testing.

Configurations without improvement. Two comparisons delimit the benefits of additional numerical complexity. On NS, ETDRK4 yields $1.28 \cdot 10^{-4}$ error, compared with $2.66 \cdot 10^{-5}$ for the selected split update (Figure 13); higher nominal order does not imply lower error against this reference. On KS-FNO, additional cached Jacobians provide little improvement beyond $K{=}1$ (Figure 18). Neither observation identifies a unique cause of the remaining error.

P Algorithmic Pseudocode

Algorithm 2 summarises branch operations. The fixed router is an $O(1)$ lookup; the correction also requires the state history, numerical operators, and parameters of the selected variant. The NS branch does not use a surrogate evaluation.

Algorithm 2 APIC corrector at inference time

Input: frozen f_θ, current state u^n, optionally u^{n-1}, PDE class $\mathcal{C} \in \{\text{lin}, \text{semi}, \text{chaos}\}$, spectral operators $\{|k|^2, \text{dealias}\}$
Output: corrected next state u^{n+1}

if $\mathcal{C} = \text{semi}$ **then**
  // Navier–Stokes Strang split branch
  $\hat{u} \leftarrow \mathcal{F}[u^n]$; $\hat{\omega} \leftarrow |k|^2 \hat{u}$
  $\hat{\omega} \leftarrow \exp(-\nu |k|^2 \Delta t/2)\, \hat{\omega}$
  $\omega \leftarrow \mathcal{F}^{-1}[\hat{\omega}]$
  $F \leftarrow \text{advection}(\omega, u^n, \text{dealias})$
  $\omega \leftarrow \omega + \Delta t\, (-F + f_{\text{src}})$
  $\hat{\omega} \leftarrow \mathcal{F}[\omega]$; $\hat{\omega} \leftarrow \exp(-\nu |k|^2 \Delta t/2)\, \hat{\omega}$
  $u^{n+1} \leftarrow \mathcal{F}^{-1}[\hat{\omega}/|k|^2]$
else if $\mathcal{C} = \text{lin}$ **then**
  // Wave Numerov + bounded refinement
  $a_k \leftarrow c^2 \Delta t^2 |k|^2/12$
  $\hat{u}^{n+1} \leftarrow [(2 - 10a_k)\hat{u}^n - (1 + a_k)\hat{u}^{n-1}]/(1 + a_k)$
  $u^{n+1} \leftarrow \mathcal{F}^{-1}[\hat{u}^{n+1}]$
  $u^{n+1} \leftarrow u^{n+1} + \epsilon \tanh(\text{FNO}_\phi(u^{n+1} - u^n))$
else
  // Kuramoto–Sivashinsky branch (FAPC + sub-step CN)
  $\tilde{u}^{n+1} \leftarrow u^n + f_\theta(u^n)$
  $r \leftarrow r_{\text{CD2}}(\tilde{u}^{n+1}, u^n)$
  $d \leftarrow J^+ r$
  $\hat{d} \leftarrow \mathcal{F}[d]$; $\hat{d} \leftarrow M_\alpha(k)\, \hat{d}$; $d \leftarrow \mathcal{F}^{-1}[\hat{d}]$
  optionally apply M sub-steps of spectral CN to refine d
  $u^{n+1} \leftarrow \tilde{u}^{n+1} - d$
end if
return u^{n+1}

Q Convergence of the Analytical Correctors

The sub-step KS variants divide an outer interval into M steps of size $\Delta t/M$. Refinement reduces the inner discretisation error, but the complete rollout also depends on the initial prediction, nonlinear evolution, and correction parameters. Figure 22 reports a three-seed KS-FNO sweep: mean error decreases from 4.13×10^{-2} at $M{=}10$ to 2.06×10^{-2} at $M{=}100$. Improvements diminish over this range, while the number of inner updates grows linearly with M. The error and cost measurements jointly characterise this parameter range.

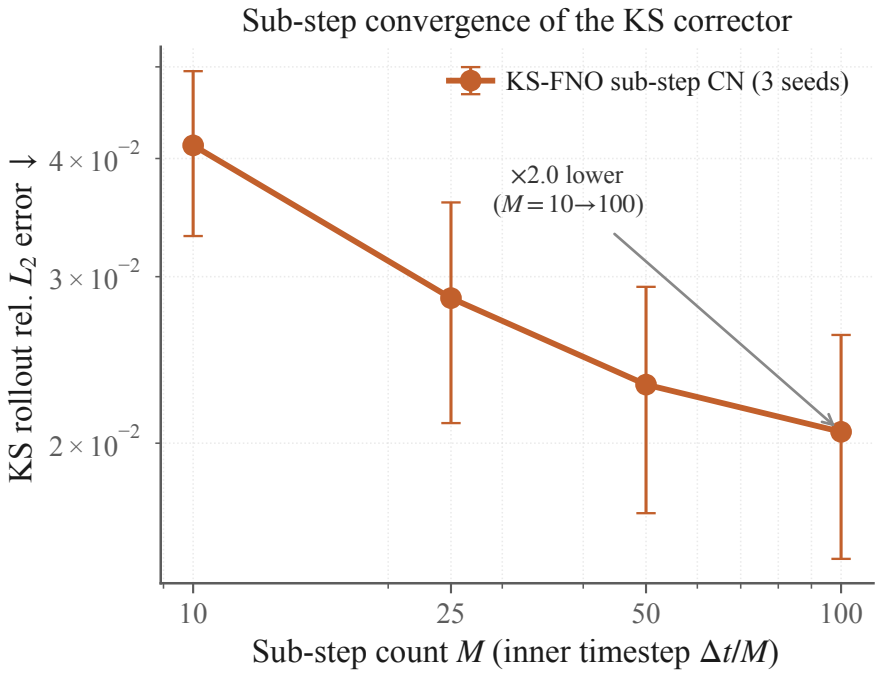

Figure 22: KS-FNO sub-step sweep over three seeds (mean ± one standard deviation). Error decreases across the displayed values of M. Error and across-seed variability are shown separately.

Table 13 provides the corresponding values. The effective error exponent between $M{=}10$ and $M{=}100$ is $\log(4.13/2.06)/\log(10) \approx 0.30$. This is smaller than the nominal second-order time-step dependence of the inner scheme and indicates that inner integration error alone does not describe the measured rollout error. Standard deviation decreases from 8.2×10^{-3} to 5.5×10^{-3}, while the coefficient of variation increases from $19.8\,\%$ to $26.6\,\%$. These statistics should be interpreted jointly when choosing a sub-step count.

Sub-steps M	10	25	50	100
Mean rel. L_2	$4.13\cdot10^{-2}$	$2.85\cdot10^{-2}$	$2.31\cdot10^{-2}$	$2.06\cdot10^{-2}$
Std (3 seeds)	$8.2\cdot10^{-3}$	$7.5\cdot10^{-3}$	$6.2\cdot10^{-3}$	$5.5\cdot10^{-3}$

Table 13: KS-FNO sub-step CN corrector: rollout relative-L_2 error vs. sub-step count M (three seeds; inner timestep $\Delta t/M$), the data behind Fig. 22. Error falls monotonically, by $\times2.0$ over the swept range.

Comparison of KS correction variants. The KS-FNO sub-step configuration at $M{=}100$ has error 2.06×10^{-2}, whereas the S1 FAPC variant in Table 2 has error 1.45×10^{-2}. These values come from different update configurations. Their approximately $30\,\%$ difference therefore cannot be attributed to adding a mask to an otherwise unchanged sub-step method. Sub-stepping and residual reweighting address different aspects of an update; assessing their interaction requires a matched component comparison.

Wave stencil convergence. Figure 23 compares the Wave stencils against an exact spectral reference under time-step refinement. Fitted error slopes are 3.96 for Numerov and 1.97 for Verlet, consistent with fourth- and second-order convergence in this test. The branch-level comparison also reports a smaller incremental reduction from FNO refinement. Stencil convergence and learned-refinement gains are distinct measurements; neither alone establishes performance at longer rollout horizons.

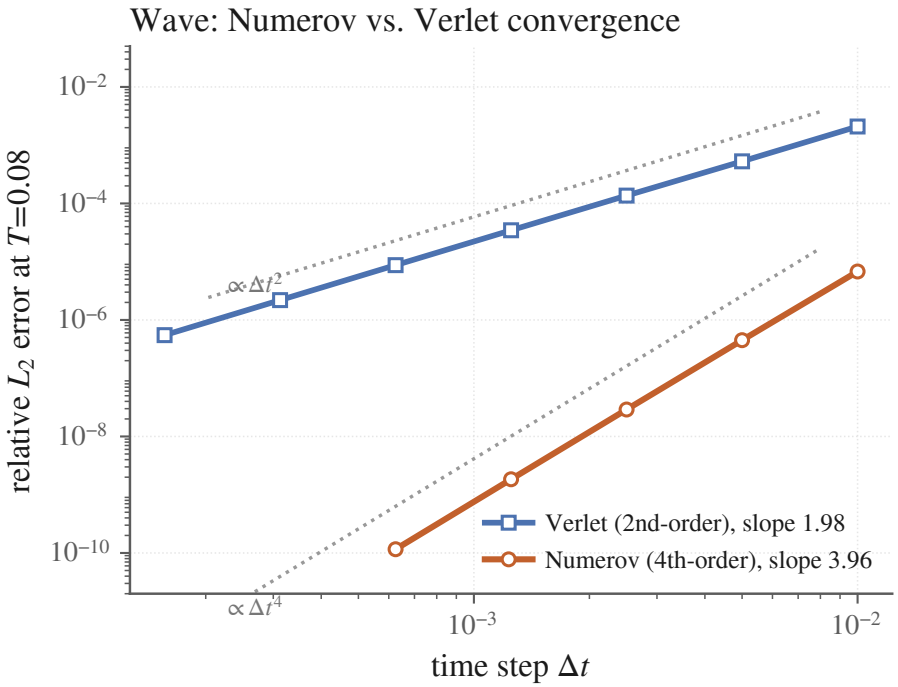

Figure 23: Wave order-of-accuracy: relative L_2 error vs. time-step for the Numerov (APIC) and Verlet (PhysicsCorrect) stencils, integrated from a data initial condition against the exact spectral solution. Fitted slopes 3.96 and 1.97 are consistent with fourth- and second-order convergence in this test; Numerov is $\sim 300\times$ more accurate at the finest step.

R SEED INVARIANCE OF THE NS BRANCH

The NS analytical branch does not use the frozen surrogate inside the rollout loop. With inputs, numerical parameters, and execution conditions fixed, its update is independent of the surrogate weights and their training seed. Table 14 reports identical aggregate errors across the evaluated architectures and seeds, 2.656020×10^{-5} at the recorded precision, with zero reported standard deviation. Figure 24 shows the corresponding per-step errors alongside the uncorrected trajectories.

NS cell	seed 0	seed 1	seed 2	std
FNO	2.656020	2.656020	2.656020	0
U-Net	2.656020	2.656020	2.656020	0
ViT	2.656020	2.656020	2.656020	0

Table 14: Recorded NS rollout errors ($\times 10^{-5}$ relative L_2, n_t=200). All listed values coincide at the displayed precision. The analytical update does not depend on surrogate training weights.

This property reduces one source of variability in the evaluated NS pipeline. It does not remove dependence on input data, numerical parameters, floating-point implementation, or hardware. Changes in those conditions require validation. Wave and KS variants that retain learned components can also depend on their fitted parameters; their reported variability is summarised in Table 6.

Branch-specific computational cost. Table 15 summarises the branch-specific measurements from Table 11, including network evaluations per rollout step and latency on the respective grids.

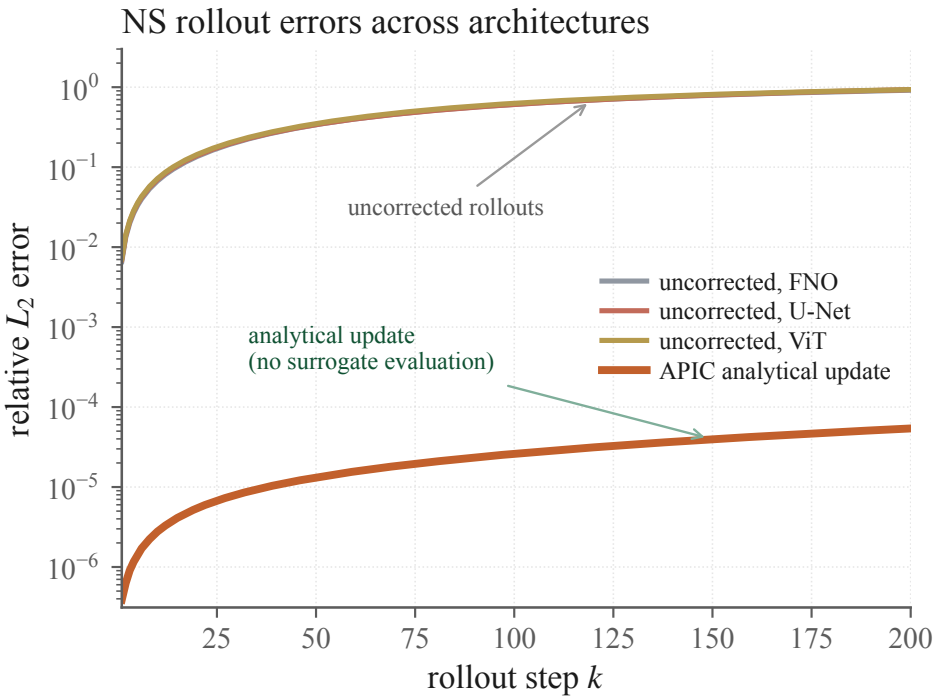

Figure 24: Per-step NS relative-L_2 error averaged over the test set. The reported analytical-branch curves coincide across FNO, U-Net, and ViT, whereas the uncorrected trajectories depend on the surrogate. Coincident plotted curves indicate agreement at the plot's resolution.

Table 15: Branch-specific network evaluations and inference latency. Measurements use the respective PDE grids and are not a common-grid comparison across branches.

Branch	Net calls/step	Dominant op	ms/step
NS Strang	**0**	spectral (FFT pairs)	**0.25**
Wave Numerov+FNO	1 (bounded)	diagonal Fourier solve	0.45
KS FAPC (FNO)	1	J^+r + masked FFT	0.58
KS sub-step CN	1	M CN sub-steps	56–224

Within the reported NS comparison, the analytical update has both lower error and lower latency than the evaluated alternatives. Costs across different PDE grids are not directly comparable. In particular, KS sub-stepping requires substantially more computation than the FAPC variant; Table 11 reports these configurations separately.

Wave refinement magnitude and variability. The analytical Wave configuration has error 5.07×10^{-4}; the refined variants report 4.73–4.81×10^{-4} (Table 6). Their three-seed standard deviations are 1.6–4.9×10^{-6}, corresponding to coefficients of variation of approximately 0.3–$1.0\,\%$. The pointwise refinement magnitude is bounded by ϵ=0.05 through the tanh parameterisation. This amplitude bound does not by itself bound seed variance or accumulated rollout error. Setting ϵ=0 removes the refinement term.

Quantitative analysis of the NS fields. Table 16 reports vorticity-field errors for the seed-0 trajectory displayed in Figure 4. This metric differs from the test-set-averaged state error in Table 2. The uncorrected peak vorticity increases from 0.41 to 2.53 and its final relative-L_2 error is 1.83. PhysicsCorrect's error increases from $9.9\cdot10^{-5}$ to $8.3\cdot10^{-4}$; APIC's error remains near $2\cdot10^{-4}$ after step 50. Thus amplitude agreement and state-error accumulation provide complementary diagnostics over this trajectory.

Synthesis across the three branches. The branches illustrate different allocations of work between analytical and learned updates. NS bypasses the network; Wave combines a higher-order

	GT	No correction		PhysCorr.	APIC
k	$\max\|\omega\|$	$\max\|\omega\|$	rel. L_2	rel. L_2	rel. L_2
1	0.41	0.41	$2.1{\cdot}10^{-2}$	$9.9{\cdot}10^{-5}$	$1.1{\cdot}10^{-4}$
50	0.43	0.88	$8.8{\cdot}10^{-1}$	$4.4{\cdot}10^{-4}$	$2.1{\cdot}10^{-4}$
100	0.47	1.63	1.42	$6.5{\cdot}10^{-4}$	$2.0{\cdot}10^{-4}$
150	0.51	2.16	1.71	$7.7{\cdot}10^{-4}$	$1.9{\cdot}10^{-4}$
200	0.56	2.53	1.83	$8.3{\cdot}10^{-4}$	$\mathbf{1.9{\cdot}10^{-4}}$

Table 16: NS field diagnostics on the displayed seed-0 trajectory: peak vorticity amplitude and vorticity-field relative-L_2 error at five rollout steps. The final uncorrected amplitude is approximately 4.5 times the reference value. APIC's error is approximately constant from steps 50 to 200 at the precision shown.

stencil with refinement; KS uses frequency-dependent correction or sub-stepping. Across complete policies, S2 reaches $83.48\,\%$ mean reduction, S1 reaches $83.73\,\%$, and S4 reaches $81.68\,\%$. These comparisons motivate numerical structure as a design choice without claiming an independent causal contribution for each component.

Scope of extension. The recipe requires an equation and an appropriate numerical update. Transfer to other equations, parameters, and reference discretisations must be tested; it is not guaranteed by the three-way lookup. When the reference generator is unknown, the masked residual correction remains a candidate, but its accuracy must be evaluated. Longer chaotic rollouts already show diminishing advantage in the present experiments.

Practical guidance. Choose a correction policy using accuracy and measured computation jointly. NS bypasses the network in the reported setting; Wave uses a different update, and KS sub-stepping can be substantially more expensive. A learned router would also introduce fitting and selection costs. Neither baseline dominance nor seed invariance on unseen equations follows from the current results.