Exactness Matters for Physical Rule Enforcement

Bum Jun Kim*
The University of Tokyo, Japan
bumjun.kim@weblab.t.u-tokyo.ac.jp

Abstract

Autoregressive scientific forecasters often enforce physical or structural constraints by repairing each predicted state before feeding it back into the model. We study when this improves forecasting through two operator properties: exactness on valid targets and geometry controlling error amplification. We compare raw forecasting, post hoc repair, and in-loop repair across periodic incompressible Navier–Stokes, non-periodic CFDBench flows, and supporting conservation-law and hierarchical-forecasting tasks. In the exact periodic regime, Fourier projection substantially improves rollout accuracy. On NS-128, a strong Raw-FNO has final-step rollout MSE at horizon 100 of $(9.390 \pm 6.290) \times 10^{-5}$; post hoc and in-loop projection reduce it to $(1.130 \pm 0.165) \times 10^{-6}$ and $(5.370 \pm 0.113) \times 10^{-7}$. On non-periodic CFDBench, stronger Poisson-based cleanup can instead reduce divergence while worsening rollout error; target distortion explains this harm far better than linear-system residual. Controlled mismatch, from-scratch U-Net checks, and periodic Burgers mass repair corroborate the mechanism. A signed correction-alignment criterion selects operators and analytic blend strengths from shared one-step validation forecasts, with prospective tests on disjoint Cavity and Burgers trajectories. Hierarchical forecasting shows the same broader pattern: exact reconciliation is a stable baseline, whereas blended top-down repair is dataset-dependent. Constraint enforcement should therefore be benchmarked by operator–data alignment before enforcement strength: use exact orthogonal projection as a primary baseline, screen approximate repairs by correction alignment, and use rollout validation for ambiguous or amplification-uncertain choices.

1 Introduction

Autoregressive scientific forecasting is often stabilized by a simple repair loop using a physical rule. A model first predicts the next state; a projection or cleanup operator then repairs that prediction so that it satisfies a known physical or structural rule; the repaired state is finally fed back into the next prediction step. In incompressible flow, as an example, the rule is the divergence-free constraint $\nabla \cdot \boldsymbol{u} = 0$, and the repair loop is especially tempting because small compressible components can accumulate during rollout, corrupt transport, and distort long-horizon energy and vorticity statistics. This stabilization benefit is why learned-simulator papers increasingly report long-horizon stability and physical consistency in addition to one-step error [22, 38, 6, 40, 27, 24]. Nevertheless, as we show in Section 4, overly aggressive enforcement can become a recurring source of distribution shift when the repair operator is only approximate.

We study when this repair loop is a reliable improvement. Two properties matter: whether the repair leaves valid benchmark targets unchanged, and whether its geometry amplifies prediction errors. Exactness removes target distortion; orthogonal projection additionally cannot increase one-step squared error in the evaluation geometry. For periodic Navier–Stokes, Fourier projection has both properties, making it a strong and mandatory baseline.

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

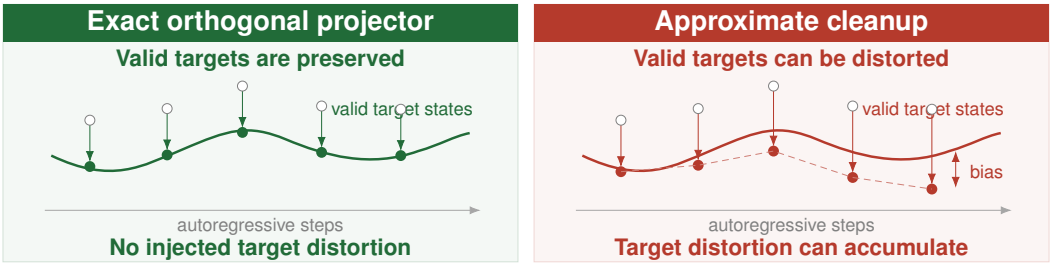


Figure 1: Operator alignment in autoregressive feedback. An exact orthogonal projector preserves valid targets and removes the orthogonal error component. Approximate cleanup can perturb valid targets even while reducing measured constraint violation. Target preservation removes injected operator bias; long-horizon behavior also depends on amplification by the composed predictor and repair.

However, this practice can fail when the available operator is only approximate. Soft residual penalties, learned pressure modules, and boundary-dependent projection schemes are widely used to impose physical structure [33, 43, 26, 18, 7, 15]. On non-periodic benchmarks, a copied-boundary Poisson cleanup can lower measured divergence while perturbing valid target states. Solving that system more accurately makes the chosen operator more faithful, but need not improve its alignment with the data. Repeated repair can then inject bias into autoregressive feedback. Figure 1 illustrates this mechanism; Section 4 and Appendix A separate target distortion from error amplification.

On strong periodic velocity benchmarks, most of the gain comes not from a bespoke decoder but from applying this exact projection to each predicted state before feeding it into the next step. This exact-projection result changes the benchmarking standard. Any new constrained architecture must be compared not only to raw forecasting but also to post hoc and in-loop projection. Our main claim therefore concerns baseline quality under exact and approximate constraint operators, with architecture comparisons used to test whether the regime split persists across different backbones.

We then test what survives outside the periodic geometry using the official cavity, tube, dam, and cylinder splits of CFDBench [29]. Stronger-solver controls, screened cleanup, controlled mismatch, and independently trained U-Nets distinguish operator alignment from solver convergence. Periodic Burgers mass conservation provides a second PDE family; Appendix E adds five public forecasting hierarchies. These studies support a practical operator-level rule without requiring a new forecasting architecture.

Contributions. We identify an exact-versus-approximate regime split under strong autoregressive baselines. On periodic NS-128, exact in-loop projection is the dominant intervention and post hoc projection a mandatory comparator. On non-periodic CFDBench, target distortion and operator amplification explain why stronger cleanup can worsen rollout despite lower residuals. Controlled operator perturbations and Burgers mass repair isolate these mechanisms. From the same alignment analysis, we derive a signed one-step criterion and analytic blend strength, and test their selections prospectively on disjoint trajectories. The resulting rule combines exact orthogonal baselines, inexpensive screening of approximate repairs, and rollout validation when the screen is uncertain. Hierarchical forecasting supports the broader structured-prediction interpretation.

2 Related Work

Operator learning and neural simulators. The Fourier neural operator (FNO) of Li et al. [25] remains a standard backbone for periodic partial differential equation (PDE) forecasting, while the DeepONet line of Lu et al. [28] provides a second canonical operator-learning family. U-Net-style encoder-decoder models remain competitive in scientific surrogate modeling because of their strong locality bias, and convolutional-network fluid surrogates remain important baselines in learned fluid simulation [36, 20]. Learned simulators based on message passing and graph networks emphasize geometry and conservation structure [38, 32, 6], while recent rollout-focused PDE work makes long-horizon stability itself a first-class evaluation target [27]. Our goal here is not to propose yet another backbone family but to identify the strongest constraint-centric intervention that all of them should

be benchmarked against. Hybrid neural and solver surrogates and physics-constrained autoregressive PDE models also increasingly couple learned predictors to explicit numerical structure rather than relying on fitting raw data alone [43, 39, 11, 13].

Physics constraints for incompressible dynamics. Soft penalties, residual losses, stream-function parameterizations, projection layers, and explicit test-time constraint solvers are all common ways to encode incompressibility or related physical structure [33, 43, 26, 37, 18]. The problem is less the existence of these techniques than the standard for evaluating them. If post hoc projection is omitted or if a new constrained decoder is only compared to raw forecasting, then empirical conclusions are difficult to interpret. Our adaptive-cleanup and risk–coverage analyses also connect this literature to a broader structured-prediction issue: constraint enforcement must be judged by both feasibility and test-time decision making [12].

Native constraints and hybrid dynamics. A differentiable spectral Leray layer inside a predictor is an architecture-native exact orthogonal repair on periodic geometry; our Proj-FNO implements this design. Placement inside a network does not itself ensure target preservation on complex boundaries. NeuralGCM couples a differentiable dynamical core to learned physics [23]; such a hybrid core is not, as a whole, an orthogonal projector. Iterative probabilistic forecasters such as DYffusion [8] motivate a separate distributional evaluation beyond the deterministic rollouts studied here.

Constraint repair beyond CFD. Forecast reconciliation in hierarchical time series provides a clean non-CFD instance of exact and approximate repair. Coherent forecasts lie in the column space of a known summing matrix, yielding exact reconciliation operators such as ordinary least squares (OLS) and minimum-trace reconciliation together with exact-but-nonorthogonal bottom-up aggregation, while top-down operators are generally mismatched to the target manifold [17, 45, 31, 3]. Classical top-down and bottom-up heuristics remain important comparators in that literature [14, 2]. Recent probabilistic, quantile, temporal, and softly coherent variants extend the same coherence issue into modern machine-learning forecasting pipelines [41, 34, 16, 35, 19, 44]. This literature is useful here because it isolates constraint-operator exactness outside fluid dynamics.

Benchmarks beyond periodic NS. Most projection-centric empirical claims are easiest to make on periodic 2D velocity benchmarks. CFDBench, introduced by Luo et al. [29], provides a stronger complementary stress test with varied boundary conditions, physical properties, and domain geometries across classic CFD problems. Recent benchmark work on autoregressive PDE emulators likewise emphasizes rollout-aware evaluation rather than one-step accuracy alone [24]. We use CFDBench's cavity-flow, tube-flow, dam-break, and cylinder-flow subsets to test what remains true once exact Fourier-space projection is no longer available.

3 Method

3.1 Forecasting Setup

Given an input history

$$\boldsymbol{u}_{t-T_{\text{in}}+1:t} \in \mathbb{R}^{T_{\text{in}}\times 2\times H\times W}, \tag{1}$$

a backbone b_θ predicts residual updates

$$\Delta\tilde{\boldsymbol{u}}_{t+1:t+K} = b_\theta(\boldsymbol{u}_{t-T_{\text{in}}+1:t}), \tag{2}$$

and the state evolves through

$$\tilde{\boldsymbol{u}}_{t+s} = \hat{\boldsymbol{u}}_{t+s-1} + \alpha\Delta\tilde{\boldsymbol{u}}_{t+s}, \tag{3}$$

$$\hat{\boldsymbol{u}}_{t+s} = \mathcal{T}(\tilde{\boldsymbol{u}}_{t+s}), \text{ for } s = 1, \ldots, K, \tag{4}$$

where $\alpha \in (0, 1)$ is a learned residual scale and $\mathcal{T}$ is either the identity, an exact divergence-free projector, or an approximate boundary-aware cleanup operator.

Terminology. Throughout the paper, exactness is an operator–data property, not a statement about how accurately an internal numerical system is solved. A repair operator T is exact for a benchmark target set $\mathcal{M}$ when $T(y) = y$ for every valid target $y \in \mathcal{M}$; approximate cleanup means that this

Table 1: Operator taxonomy. Exactness concerns target preservation; geometry controls amplification. Mild/strong denotes the cleanup settings studied here, not a universal norm guarantee.

Regime	Fixes targets?	Examples	Operating rule
Exact orthogonal	Yes	FFT, OLS, mass projection	Primary projection baseline
Exact nonorthogonal	Yes	Bottom-up, oblique T_η	Audit amplification
Approximate mild	Not assured	Screened, GeoScreened, blended cleanup	Screen alignment and strength
Approximate strong	Not assured	Jacobi, MG, SOR, CG, Direct	Strong-solve comparison anchor

identity can fail on benchmark-valid targets, even if the cleanup maps many inputs closer to the constraint set. Under this terminology, the Direct CFDBench variant is an exact solve of its boundary-aware Poisson system, but it remains an approximate cleanup operator when that boundary-aware system is not exact-on-target for the benchmark velocity fields.

3.2 Exact Periodic Projection

For periodic 2D velocity fields $\boldsymbol{u} = (u, v)$, Π_{div} denotes the orthogonal Helmholtz–Leray projector: it removes the velocity component parallel to each nonzero Fourier wavevector and preserves the mean. We implement it in Fourier space. The Fourier-space implementation is the periodic spectral version of the classical projection viewpoint for incompressible flow [10, 42]. Let $\bar{\boldsymbol{u}}$ denote the spatial mean of $\boldsymbol{u}$, and write $\boldsymbol{k} = (k_x, k_y)$ for the two-dimensional Fourier wavevector. Define the Fourier-space vorticity by

$$\hat{\omega}(\boldsymbol{k}) = ik_x\hat{v}(\boldsymbol{k}) - ik_y\hat{u}(\boldsymbol{k}), \tag{5}$$

and define the stream function by

$$\hat{\psi}(\boldsymbol{k}) = \frac{\hat{\omega}(\boldsymbol{k})}{\|\boldsymbol{k}\|_2^2} \quad \text{for } \boldsymbol{k} \neq 0, \qquad \hat{\psi}(0) = 0. \tag{6}$$

The projected velocity is then

$$\Pi_{\mathrm{div}}(\boldsymbol{u}) = \bar{\boldsymbol{u}} + (\partial_y\psi, -\partial_x\psi), \tag{7}$$

which preserves the zero Fourier mode and satisfies $\nabla \cdot \Pi_{\mathrm{div}}(\boldsymbol{u}) = 0$ up to numerical precision.

This exact periodic projector is orthogonal with respect to the L_2 geometry of the periodic benchmark. If $\boldsymbol{y} = \Pi_{\mathrm{div}}(\boldsymbol{y})$, then, for any raw prediction $\tilde{\boldsymbol{u}}$,

$$\|\Pi_{\mathrm{div}}(\tilde{\boldsymbol{u}}) - \boldsymbol{y}\|_2^2 = \|\tilde{\boldsymbol{u}} - \boldsymbol{y}\|_2^2 - \|(I - \Pi_{\mathrm{div}})\tilde{\boldsymbol{u}}\|_2^2 \leq \|\tilde{\boldsymbol{u}} - \boldsymbol{y}\|_2^2. \tag{8}$$

Equation 8 explains why post hoc projection is already a strong baseline under the exact periodic projector. Any difference between projecting both the target and the autoregressive state and projecting only the autoregressive state is therefore a separate training-dynamics issue rather than a consequence of the orthogonal-projection identity itself, and in our exact periodic experiments, it is empirically only a second-order effect. In Appendix A, Proposition 1 proves the general version of Equation 8. Projection onto a subspace containing the target cannot increase the squared error.

3.3 Boundary-Preserving Approximate Cleanup

However, for non-periodic cavity, tube, dam, and cylinder flow, exact spectral projection is unavailable. We therefore use training-matched boundary-preserving approximate cleanup operators, denoted $\widetilde{\Pi}_{\mathrm{bc}}$, which are boundary-constrained Chorin–Temam-style cleanup maps rather than exact Leray projectors. These maps copy boundary velocities and update only interior velocities using a Poisson or screened Poisson pressure solve, so copying raw boundary values can break both exact divergence annihilation and idempotence on the full velocity field. We study Jacobi, multigrid (MG), successive over-relaxation (SOR), conjugate-gradient (CG), and Direct pressure solves, alongside a screened family that weakens the pressure response. Appendix D.1 gives the full discrete stencil, solver definitions, and screened-family protocol.

Approximate-regime bias decomposition. Unlike Π_{div}, the boundary-preserving approximate cleanup operator $\widetilde{\Pi}_{\mathrm{bc}}$ is not orthogonal. Equation 8 does not apply, and neither post hoc nor in-loop cleanup is guaranteed to improve one-step MSE. A useful decomposition for any approximate cleanup operator $\widetilde{\Pi}$ and target y is

$$\|\widetilde{\Pi}(x) - y\|_2^2 = \|\widetilde{\Pi}(x) - \widetilde{\Pi}(y)\|_2^2 + \|\widetilde{\Pi}(y) - y\|_2^2 + 2\langle \widetilde{\Pi}(x) - \widetilde{\Pi}(y), \widetilde{\Pi}(y) - y\rangle. \tag{9}$$

The second term is target distortion; the cross-term can have either sign. For $x_{t+1} = T(F(x_t))$, target trajectory (y_t), and $e_t = x_t - y_t$, Appendix A gives

$$\|e_{t+1}\| \le L_T L_F \|e_t\| + L_T \delta_t + \beta_{t+1}, \tag{10}$$

where $\delta_t = \|F(y_t) - y_{t+1}\|$, $\beta_t = \|T(y_t) - y_t\|$, and L_T, L_F are Lipschitz factors. Exactness eliminates β_t; operator geometry controls amplification. This finite-horizon recursion needs no contraction. A uniform steady-state bound additionally requires $L_T L_F < 1$, which we do not assume for CFDBench. Numerical norm audits and controlled oblique/biased operators test these distinct mechanisms (Appendices A.1 and F.3). Lower solver residual controls only implementation error around the chosen operator, not its intrinsic target distortion. All non-periodic cleanup is applied in physical velocity units before renormalization; the frame-choice ablation changes final-horizon MSE by at most 0.14%.

3.4 Forecasting Variants

Raw uses no repair. PostHoc reuses the identical raw-trained checkpoint and repairs each prediction at evaluation before feedback. Proj applies the differentiable repair to every predicted state during both training and evaluation; on NS-128 it is an architecture-native spectral output layer. We also compare constraint-aware projection (CAP), soft-divergence training (SoftDiv), and adaptive blended cleanup. Appendix C gives their objectives and matched-budget protocols.

3.5 Divergence-Gated Adaptive Cleanup

Approximate cleanup can reduce measured divergence while perturbing benchmark-valid states. The nominal discrete constraint set and the target distribution need not coincide. This trade-off suggests that the right action outside the exact regime is often neither to skip cleanup entirely nor to apply it at full strength. We therefore evaluate an adaptive blended cleanup that uses predicted root-mean-square (RMS) divergence as a risk score:

$$d(\tilde{\boldsymbol{u}}) = \|\nabla \cdot_h \tilde{\boldsymbol{u}}\|_{\mathrm{rms}}, \quad \gamma(\tilde{\boldsymbol{u}}) = \min\{(d(\tilde{\boldsymbol{u}})/\tau)^q, 1\}, \quad \hat{\boldsymbol{u}}_{\mathrm{ada}} = \tilde{\boldsymbol{u}} + \gamma(\tilde{\boldsymbol{u}})(\mathcal{P}(\tilde{\boldsymbol{u}}) - \tilde{\boldsymbol{u}}), \tag{11}$$

where $\tau > 0$ and $q > 0$. When $d(\tilde{\boldsymbol{u}})$ is small, the model stays close to the raw forecast; when it is large, the update approaches full post hoc cleanup. We instantiate $\mathcal{P}$ with the Jacobi family and with the screened family for the main deployment study, use Direct only as a stronger-operator stress test, and report additional GeoScreened deployment variants in Appendix D. We tune the threshold τ and exponent q once per dataset on the validation split, sweep those parameters to assess sensitivity, and then evaluate the selected rule on the held-out test trajectories. The method requires no retraining; divergence is an observable gating score, while rollout accuracy determines whether the gate is useful (Appendix F.7).

3.6 One-Step Operator and Strength Selection

The controlled-mismatch quadratic yields a signed criterion before repaired feedback. On labeled, teacher-forced validation pairs, let $x = F(s)$, target y, and correction $c_T(x) = T(x) - x$. Define

$$A_T = \mathbb{E}\langle x - y, c_T(x)\rangle, \qquad B_T = \mathbb{E}\|c_T(x)\|^2. \tag{12}$$

For $T_\alpha(x) = x + \alpha c_T(x)$, the change in one-step squared-error risk and its minimizing strength are

$$\Delta_1(T_\alpha) = 2\alpha A_T + \alpha^2 B_T, \qquad \alpha_T^\star = \mathrm{clip}_{[0,1]}(-A_T/B_T), \quad B_T > 0. \tag{13}$$

Negative Δ_1 means improvement: the correction points toward the target strongly enough to offset its energy. Estimate A_T, B_T on shared one-step forecasts and select the candidate minimizing $S(T) = \Delta_1(T_{\alpha_T^\star})$, including $S(\mathrm{Raw}) = 0$ and choosing Raw on a zero-score tie. If $B_T = 0$,

use Raw. No candidate-specific validation rollout or strength sweep is needed. Exact orthogonal P recovers $\Delta_1(P) = -\mathbb{E}\|(I-P)x\|^2 \leq 0$ and optimal full strength. For approximate repairs, combine the score with offline target-distortion and amplification audits; statistically ambiguous or amplification-uncertain choices use rollout validation. The identity guarantees one-step risk ordering; its long-horizon transfer is tested empirically below. Appendix A.2 gives the derivation and finite-horizon link.

4 Experiments

Datasets. We use exact periodic and approximate non-periodic tracks, with cross-PDE and prospective validation studies. The exact periodic main track uses periodic NS-128 with a matched FNO family trained for 18 epochs and evaluated out to 200 rollout steps. The approximate official CFDBench tracks cover cavity, tube, dam, and cylinder under the official splits with matched raw and projected FNO recipes and 20-step rollout evaluation, and dam together with cylinder additionally include 60-step long-horizon reevaluations. Additional approximate-regime experiments add stronger-solver negative controls, adaptive cleanup, controlled mismatch sweeps, the validation-to-deployment protocol below, and the geometry-aware, projection-frame, and from-scratch U-Net studies collected in Appendix D. Appendix E adds a real-data hierarchical forecasting support track using public Traffic, Labour, TourismLarge, OldTourismLarge, and Wiki2 hierarchies from the HierarchicalForecast benchmark collection introduced by Olivares et al. [30]. That track uses long-horizon autoregressive gated recurrent unit (GRU) models following Cho et al. [9], exact OLS and bottom-up reconciliation, and a validation-tuned blended top-down family.

Metrics. We report one-step test mean squared error (MSE), rollout MSE averaged over the evaluation horizon (MSE-AUC), final-horizon MSE and divergence root mean square (Div), and latency. MSE-AUC is the discrete average $T_{\text{eval}}^{-1}\sum_{t=1}^{T_{\text{eval}}}\text{MSE}_t$. Final-horizon metrics are denoted MSE@T_{eval} and Div@T_{eval}; H, W denote grid dimensions. Entries are mean $\pm$ sample standard deviation across training seeds unless specified otherwise. Paired confidence intervals (CIs) quantify intervention differences. Appendix F.8 consolidates definitions, units, and aggregation; Appendix F.4 specifies dependence-aware inference. Hierarchical forecasting additionally reports coherence RMS and target distortion.

Recipe. Appendix H gives the full reproducibility recipe, covering optimizer, hardware, launch, artifact management, compute, latency, validation model selection, and random-initialization details, including the initialization counts and subset conventions. It also includes the cost comparison used to rule out hidden capacity or training-cost explanations for the main exact-regime ordering and the tuned cylinder cleanup comparison so that the body can focus on dataset scope, the metric definitions above, and fairness-critical comparisons.

4.1 Main Evidence Across Regimes

The clearest evidence comes from the exact periodic regime. Table 2 shows that Raw FNO is already strong on NS-128, with test MSE $(5.810 \pm 3.600) \times 10^{-8}$, so projection is not beating a weak baseline. Yet exact cleanup is still decisive, consistent with the one-step identity and finite-horizon analysis in Theorem 2. Post hoc projection reduces rollout MSE@100 from $(9.390 \pm 6.290) \times 10^{-5}$ to $(1.130 \pm 0.165) \times 10^{-6}$, and in-loop projection reduces it again to $(5.370 \pm 0.113) \times 10^{-7}$, with CAP statistically tied to Proj-FNO as expected under the exact orthogonal-projector setting. SoftDiv remains worse than Raw despite matched compute. A separate five-seed ON/OFF replication isolates inference-time projection from training alignment: paired Raw-minus-PostHoc AUC improves by 6.590×10^{-5} (95% CI $[1.999 \times 10^{-5}, 1.118 \times 10^{-4}]$), with a further gain from projected training (Appendix F.1).

In the approximate regime, operator family matters more than solving the nominal constraint harder. Independently trained U-Nets reproduce the distortion sensitivity: on all four official CFDBench subsets, Jacobi and Direct reduce divergence but increase rollout MSE in every seed. Mild cleanup helps Tube but hurts Dam and Cylinder (Appendix D.6).

Figure 2 summarizes the split: exact projection lowers both error and divergence, whereas approximate cleanup traces a dataset-dependent accuracy–consistency frontier.

Table 2: Main capacity- and compute-matched NS-128 FNO study. Exact in-loop projection is the dominant intervention on the strongest periodic benchmark. Smaller values are better in every numeric column.

Variant	Test MSE	Rollout MSE-AUC	Rollout MSE@100	Rollout Div@100	Latency (ms)
Raw-FNO	$(5.810 \pm 3.600) \times 10^{-8}$	$(4.210 \pm 2.750) \times 10^{-5}$	$(9.390 \pm 6.290) \times 10^{-5}$	$(2.900 \pm 1.030) \times 10^{-2}$	8.75 ± 0.25
PostHoc-FNO	$(1.570 \pm 0.305) \times 10^{-9}$	$(6.200 \pm 0.749) \times 10^{-7}$	$(1.130 \pm 0.165) \times 10^{-6}$	$(2.670 \pm 0.134) \times 10^{-5}$	11.39 ± 0.26
Proj-FNO	$(8.270 \pm 0.577) \times 10^{-10}$	$(3.130 \pm 0.029) \times 10^{-7}$	$(5.370 \pm 0.113) \times 10^{-7}$	$(2.100 \pm 0.083) \times 10^{-5}$	11.49 ± 0.13
CAP-FNO	$(8.150 \pm 0.244) \times 10^{-10}$	$(3.180 \pm 0.126) \times 10^{-7}$	$(5.410 \pm 0.243) \times 10^{-7}$	$(2.150 \pm 0.252) \times 10^{-5}$	11.42 ± 0.08
SoftDiv-FNO	$(1.550 \pm 1.110) \times 10^{-7}$	$(1.000 \pm 0.542) \times 10^{-4}$	$(2.360 \pm 1.370) \times 10^{-4}$	$(4.550 \pm 1.100) \times 10^{-2}$	8.69 ± 0.24

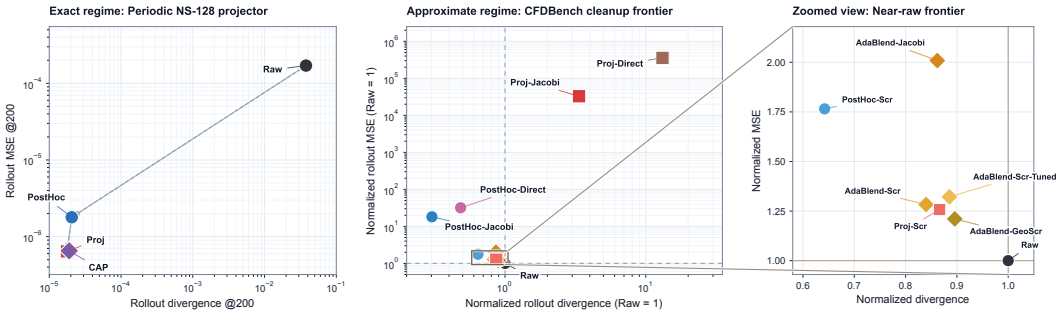


Figure 2: Regime split between exact and approximate constraint enforcement. Exact projection behaves like a benign inductive bias on the periodic manifold, while approximate cleanup produces an accuracy–consistency frontier whose shape depends on operator family and dataset. MSE and divergence are divided by the matched Raw values within each dataset; Raw is therefore at $(1, 1)$. Lower values are better.

4.2 Mechanism Evidence: Distortion, Not Residual, Explains the Reversal

The core mechanism is visible directly on held-out targets in Table 3. Exact fast Fourier transform (FFT) projection is essentially the identity on periodic ground truth, while approximate cleanup operators are qualitatively different. They often reduce divergence, but, under copied-boundary stencils, the measured full-domain divergence need not decrease for every operator, and they can perturb target states non-negligibly on cavity, tube, and dam. Direct is the cleanest stress test for Proposition 5; it drives the linear-system residual near zero while producing larger target distortion than weaker approximate operators.

Table 4 connects target distortion to rollout harm. Its 12 points are operator–dataset conditions sharing checkpoints and trajectories within each dataset, not independent runs. Distortion remains strongly associated with rollout penalty under an exact within-dataset permutation test, whereas solver residual has no significant association. GT distortion is an offline mechanism audit on labeled targets; deployment choices use labeled validation forecasts before deployment, never test labels or online target access. Appendix F.4 reports run-wise and dataset-adjusted checks.

Appendix D.2 reports the dense screened-family sweep behind this near-identity reading. We next test the same mechanism at the rollout level through an intervention motivated by Proposition 9, blending raw and cleaned states without retraining as $\tilde{\boldsymbol{u}}_\alpha = (1-\alpha)\boldsymbol{u}_{\text{raw}} + \alpha \boldsymbol{u}_{\text{clean}}$. The result is decisive in Table 5. In the matched comparison using the screened operator with $\lambda = 16$ (Screened16) and a two-cell post hoc taper (Taper2), Dam@60 is minimized at $\alpha = 0$ for both Screened16 and GeoScreened+Taper2, while Cylinder@60 prefers $\alpha = 0$ for Screened16 and a barely blended $\alpha = 0.1$ for GeoScreened+Taper2. Full cleanup is consistently worse. The approximate-regime sweet spot therefore stays at the raw forecast or very near the identity map rather than at strong cleanup.

Appendix D.3 reports the corresponding controlled-mismatch curves, and Appendix D.5 reports case-level response-atlas and cylinder alignment-flip diagnostics for the same approximate-regime mechanism.

Rollout-validation fallback. Table 6 provides the complementary full-rollout validation protocol for cases where the one-step screen is ambiguous or amplification is uncertain. Starting from a strong raw checkpoint, it compares Raw, adaptive and post hoc screened/GeoScreened candidates, and

Table 3: Constraint-operator diagnostics on held-out targets. Exact FFT projection is nearly identity, whereas approximate cleanup trades constraint-solve strength against target distortion; measured full-domain divergence need not decrease for every copied-boundary operator. GT divergence before is shared within each dataset. Columns report $\mathrm{Div}(T(y))$, $\mathrm{MSE}(T(y), y)$, $\|A_\lambda p - b\|_2/(\|b\|_2 + \epsilon)$ with $b = -D_h y$, and $100\,\mathbb{E}[\mathrm{MSE}(T(y), y)/\mathrm{mean}(y^2)]$. These are offline labeled-target audits in physical units, not online monitoring signals.

Operator	GT div after	GT distortion MSE	Relative system residual	Distortion relative to energy (%)
NS-128 exact periodic, GT div before $= (2.263 \pm 0.450) \times 10^{-6}$				
FFT projection	$(2.787 \pm 0.521) \times 10^{-6}$	$(8.126 \pm 4.079) \times 10^{-12}$	not applicable	0.00
Cavity approximate boundary-aware, GT div before $= (1.840 \pm 1.190) \times 10^{-1}$				
Jacobi-10	$(6.400 \pm 5.800) \times 10^{-2}$	$(1.200 \pm 1.500) \times 10^{-2}$	$(1.954 \pm 0.822) \times 10^{-1}$	0.19
MG-2	$(6.000 \pm 5.600) \times 10^{-2}$	$(1.500 \pm 1.700) \times 10^{-2}$	$(1.666 \pm 0.694) \times 10^{-1}$	0.25
SOR-10	$(8.100 \pm 8.400) \times 10^{-2}$	$(3.000 \pm 3.900) \times 10^{-2}$	$(2.081 \pm 1.369) \times 10^{-1}$	0.66
CG-10	$(8.200 \pm 8.300) \times 10^{-2}$	$(3.800 \pm 5.300) \times 10^{-2}$	$(1.984 \pm 1.011) \times 10^{-1}$	0.94
Direct	$(6.200 \pm 6.800) \times 10^{-2}$	$(4.900 \pm 7.100) \times 10^{-2}$	$(4.440 \pm 2.765) \times 10^{-6}$	1.25
Tube approximate boundary-aware, GT div before $= (5.300 \pm 2.200) \times 10^{-2}$				
Jacobi-20	$(3.800 \pm 2.100) \times 10^{-2}$	$(5.928 \pm 2.283) \times 10^{-4}$	$(2.968 \pm 2.024) \times 10^{-1}$	0.25
MG-2	$(3.700 \pm 1.900) \times 10^{-2}$	$(5.235 \pm 2.066) \times 10^{-4}$	$(3.237 \pm 1.912) \times 10^{-1}$	0.22
SOR-20	$(5.300 \pm 1.500) \times 10^{-2}$	$(2.602 \pm 2.677) \times 10^{-3}$	$(5.563 \pm 4.560) \times 10^{-1}$	0.85
CG-20	$(5.900 \pm 1.900) \times 10^{-2}$	$(9.616 \pm 1.087) \times 10^{-3}$	$(5.357 \pm 4.229) \times 10^{-1}$	2.98
Direct	$(5.400 \pm 2.800) \times 10^{-2}$	$(1.300 \pm 1.400) \times 10^{-2}$	$(2.945 \pm 2.568) \times 10^{-5}$	4.18
Dam approximate boundary-aware, GT div before $= (6.600 \pm 1.000) \times 10^{-2}$				
Jacobi-20	$(3.200 \pm 0.500) \times 10^{-2}$	$(2.136 \pm 0.645) \times 10^{-3}$	$(3.244 \pm 0.360) \times 10^{-1}$	0.96
Direct	$(4.400 \pm 0.700) \times 10^{-2}$	$(1.000 \pm 0.700) \times 10^{-2}$	$(1.173 \pm 0.335) \times 10^{-5}$	4.05

Table 4: Association with rollout penalty across 12 paired operator–dataset conditions. Fisher intervals are conventional point-level summaries; blocked p values enumerate all 28,800 within-dataset permutations. Operators share three checkpoints and trajectories within each dataset.

Predictor	Pearson r_{P} [95% CI]	Spearman ρ	Blocked p
GT distortion MSE	$0.905\ [0.688, 0.973]$	0.909	1.39×10^{-4}
Poisson residual	$-0.387\ [-0.786, 0.240]$	-0.427	0.215

Table 5: Controlled mismatch study on the hardest long-horizon rows. The best point is raw or a very mild blend, not full cleanup. Smaller values are better in the three MSE columns.

Dataset	Operator	Best α	Best final-step MSE	Mild $\alpha = 0.10$	Full $\alpha = 1.0$
Dam @60	Screened16	0	$(2.280 \pm 0.122) \times 10^{-4}$	$(2.320 \pm 0.138) \times 10^{-4}$	$(6.130 \pm 0.437) \times 10^{-4}$
Dam @60	GeoScreened+Taper2	0	$(2.280 \pm 0.122) \times 10^{-4}$	$(2.320 \pm 0.136) \times 10^{-4}$	$(5.580 \pm 0.396) \times 10^{-4}$
Cylinder @60	Screened16	0	$(5.300 \pm 3.400) \times 10^{-3}$	$(5.400 \pm 3.600) \times 10^{-3}$	$(1.270 \pm 0.480) \times 10^{-2}$
Cylinder @60	GeoScreened+Taper2	0.1	$(5.300 \pm 3.600) \times 10^{-3}$	$(5.300 \pm 3.600) \times 10^{-3}$	$(8.900 \pm 5.200) \times 10^{-3}$

Direct in physical units. Family and strength minimize validation final-horizon MSE; test results report the selected rule alongside Raw and Direct. The five-run extension leaves the selected rule unchanged (Appendix H).

On the official-split Dam@60 and Cylinder@60 rows, validation keeps the raw forecast rather than any always-clean intervention, exactly matching the controlled-mismatch story that the optimal point is raw or nearly raw. On the geometry holdouts, validation shifts to milder approximate operators instead, with PostHoc-Screened at $\lambda = 16$ on Dam and AdaBlend-GeoScreened at $\tau = 0.60$ and $q = 1.0$ on Cylinder. Across all four validation-selected deployment rows, Direct is never selected; validation chooses either Raw or a milder screened or GeoScreened operator.

The one-step criterion makes rollout comparison a targeted fallback rather than the default candidate-wise search. Both stages retain Raw and evaluate accuracy jointly with constraint violation. Domain-matched validation is essential: a rule selected under one geometry, forcing, or resolution need not transfer unchanged.

Table 6: Validation-to-deployment protocol on the four hardest approximate-regime rows. The official-split Dam and Cylinder rows keep the raw forecast, while the geometry holdouts select milder approximate operators rather than Direct cleanup. Smaller values are better in the three MSE columns.

Setting	Validation-selected rule	Hyperparameters	Selected final-step MSE	Raw final-step MSE	Direct final-step MSE
Dam (official split)	Raw	none	$(2.400 \pm 0.191) \times 10^{-4}$	$(2.400 \pm 0.191) \times 10^{-4}$	$(3.020 \pm 0.240) \times 10^{-2}$
Cylinder (official split)	Raw	none	$(4.400 \pm 2.800) \times 10^{-3}$	$(4.400 \pm 2.800) \times 10^{-3}$	$(2.380 \pm 0.328) \times 10^{-1}$
Dam geometry holdout	PostHoc-Screened	$\lambda = 16$	$(3.540 \pm 0.470) \times 10^{-2}$	$(3.720 \pm 0.520) \times 10^{-2}$	$(5.210 \pm 0.400) \times 10^{-2}$
Cylinder geometry holdout	AdaBlend-GeoScreened	$\tau = 0.60, q = 1.0$	$(1.910 \pm 0.480) \times 10^{-2}$	$(2.210 \pm 0.550) \times 10^{-2}$	$(9.040 \pm 2.660) \times 10^{-2}$

Table 7: Prospective validation-only selections on disjoint test trajectories. Positive gain is lower MSE-AUC than Raw. Cavity scores only steps 2–20. CI methods and complete seed results are in Appendix F.6. The Burgers audit is distinct from the mass-bias mechanism experiment.

Test	Seeds	Validation choice	AUC gain [95% CI]
CFDBench Cavity	3	Screened-4, $\alpha^\star = 0.057$–0.080	0.467% $[0.256, 1.274]$
Burgers	5	ExactMass	14.39% $[9.13, 19.66]$

4.3 Cross-PDE and Prospective Validation

A second conservation law. On periodic 1D viscous Burgers, five from-scratch circular CNNs test mass conservation instead of incompressibility. Exact mass repair lowers MSE-AUC by 18.2% and MSE@100 by 21.6%. Biased repair exactly satisfies an intentionally wrong mass (imposed-rule residual below 9×10^{-8}), yet error increases with target distortion in every seed (Appendix F.5).

Selection before test rollouts. We separately test Section 3.6 on fresh Cavity FNOs and Burgers CNNs. All choices use labeled validation one-step pairs and are recorded before any candidate is rolled out on disjoint test trajectories. Cavity selects mild Screened-4 blends and improves future-step AUC in all three seeds; full-strength Jacobi/Direct are correctly rejected in all six seed–operator cases. The independent Burgers audit selects the test oracle in all five seeds, with 46/46 non-abstained help/harm predictions correct and nine ambiguous margins. Table 7 summarizes the gains; Appendix F.6 gives candidate menus, CIs, and oracle regrets. These tests establish the predictive utility of correction alignment on unseen future rollouts.

5 Discussion

An exact orthogonal projector aligned with benchmark targets is a mandatory baseline and, on NS-128, the dominant intervention. Matched-checkpoint toggles separate its inference-time benefit from the additional effect of training with projected feedback. Exactness and geometry play complementary roles: oblique target-preserving maps can amplify errors, while nonexpansive biased maps can repeatedly perturb valid states.

For approximate operators, stronger solves need not improve forecasting. The controlled mismatch and Burgers experiments isolate target distortion even when constraint residuals decrease or vanish. Correction alignment turns this mechanism into a signed, testable operator-and-strength decision using shared one-step forecasts. Prospective Cavity and Burgers tests support its predictive value; uncertain margins or amplification retain a clear rollout-validation fallback.

The hierarchical support track (Appendix E) broadens the conclusion beyond PDEs. Exact reconciliation is a stable baseline across five hierarchies, whereas blended top-down repair ranges from harmful to highly beneficial. What transfers is the need to audit target preservation, geometry, and correction direction, rather than a universal ordering of approximate repairs. Representative validation and strong raw baselines remain central to identifying the useful operating point.

6 Conclusion

We establish a regime-level evaluation rule for constraint enforcement in autoregressive forecasting. When the operator is exact on valid targets and orthogonal in the benchmark geometry, as in periodic NS-128, projection substantially improves rollout accuracy. Post hoc projection is a mandatory comparator and in-loop projection the primary baseline. For approximate cleanup, target distortion and amplification matter more than the accuracy of the internal Poisson solve. Stronger-solver controls, controlled perturbations, from-scratch U-Nets, Burgers mass repair, and hierarchical forecasting support this mechanism. A signed one-step criterion predicts correction benefit and selects analytic blend strength; prospective held-out tests demonstrate its utility. Keep Raw in the candidate menu, audit operators in physical units, and use rollout validation when alignment or amplification is uncertain. Constraint enforcement should be benchmarked by operator–data alignment first and enforcement strength second.

Acknowledgements

B.J.K. was supported by JSPS KAKENHI Grant Number JP26K21295. This research used resources of the Argonne Leadership Computing Facility under the ALCF project RobustViT and resources of the Oak Ridge Leadership Computing Facility under OLCF Project ID CSC704 through Director's Discretionary allocation awards. The authors declare no competing interests.

References

[1] Rong An. Error Analysis of a New Fractional-Step Method for the Incompressible Navier-Stokes Equations with Variable Density. *J. Sci. Comput.*, 84(1):3, 2020.

[2] George Athanasopoulos, Roman A Ahmed, and Rob J Hyndman. Hierarchical forecasts for Australian domestic tourism. *International Journal of Forecasting*, 25(1):146–166, 2009.

[3] George Athanasopoulos, Rob J Hyndman, Nikolaos Kourentzes, and Anastasios Panagiotelis. Forecast reconciliation: A review. *International Journal of Forecasting*, 40(2):430–456, 2024.

[4] John B Bell, Phillip Colella, and Harland M Glaz. A second-order projection method for the incompressible Navier-Stokes equations. *Journal of computational physics*, 85(2):257–283, 1989.

[5] Harsh Bhatia, Gregory Norgard, Valerio Pascucci, and Peer-Timo Bremer. The Helmholtz-Hodge Decomposition - A Survey. *IEEE Trans. Vis. Comput. Graph.*, 19(8):1386–1404, 2013.

[6] Johannes Brandstetter, Daniel E. Worrall, and Max Welling. Message Passing Neural PDE Solvers. In *ICLR*, 2022.

[7] David L Brown, Ricardo Cortez, and Michael L Minion. Accurate projection methods for the incompressible Navier–Stokes equations. *Journal of computational physics*, 168(2):464–499, 2001.

[8] Salva Rühling Cachay, Bo Zhao, Hailey Joren, and Rose Yu. DYffusion: A Dynamics-informed Diffusion Model for Spatiotemporal Forecasting. In *NeurIPS*, 2023.

[9] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *EMNLP*, pages 1724–1734, 2014.

[10] Alexandre Joel Chorin. Numerical solution of the Navier-Stokes equations. *Mathematics of computation*, 22(104):745–762, 1968.

[11] Filipe de Avila Belbute-Peres, Thomas D. Economon, and J. Zico Kolter. Combining Differentiable PDE Solvers and Graph Neural Networks for Fluid Flow Prediction. In *ICML*, pages 2402–2411, 2020.

[12] Yonatan Geifman and Ran El-Yaniv. Selective Classification for Deep Neural Networks. In *NIPS*, pages 4878–4887, 2017.

[13] Nicholas Geneva and Nicholas Zabaras. Modeling the dynamics of PDE systems with physics-constrained deep auto-regressive networks. *J. Comput. Phys.*, 403, 2020.

[14] Charles W Gross and Jeffrey E Sohl. Disaggregation methods to expedite product line forecasting. *Journal of forecasting*, 9(3):233–254, 1990.

[15] Jean-Luc Guermond, Peter Minev, and Jie Shen. An overview of projection methods for incompressible flows. *Computer methods in applied mechanics and engineering*, 195(44-47): 6011–6045, 2006.

[16] Xing Han, Sambarta Dasgupta, and Joydeep Ghosh. Simultaneously Reconciled Quantile Forecasting of Hierarchically Related Time Series. In *AISTATS*, pages 190–198, 2021.

[17] Rob J. Hyndman, Roman A. Ahmed, George Athanasopoulos, and Han Lin Shang. Optimal combination forecasts for hierarchical time series. *Comput. Stat. Data Anal.*, 55(9):2579–2589, 2011.

[18] Xiaowei Jin, Shengze Cai, Hui Li, and George Em Karniadakis. NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 426:109951, 2021.

[19] Harshavardhan Kamarthi, Lingkai Kong, Alexander Rodríguez, Chao Zhang, and B. Aditya Prakash. When Rigidity Hurts: Soft Consistency Regularization for Probabilistic Hierarchical Time Series Forecasting. In *KDD*, pages 1057–1072, 2023.

[20] Byungsoo Kim, Vinicius C. Azevedo, Nils Thuerey, Theodore Kim, Markus H. Gross, and Barbara Solenthaler. Deep Fluids: A Generative Network for Parameterized Fluid Simulations. *Comput. Graph. Forum*, 38(2):59–70, 2019.

[21] John Kim and Parviz Moin. Application of a fractional-step method to incompressible Navier-Stokes equations. *Journal of computational physics*, 59(2):308–323, 1985.

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

[23] Dmitrii Kochkov, Janni Yuval, Ian Langmore, Peter C. Norgaard, Jamie A. Smith, Griffin Mooers, Milan Klöwer, James Lottes, Stephan Rasp, Peter D. Düben, Sam Hatfield, Peter W. Battaglia, Alvaro Sanchez-Gonzalez, Matthew Willson, Michael P. Brenner, and Stephan Hoyer. Neural general circulation models for weather and climate. *Nat.*, 632(8027):1060–1066, 2024.

[24] Felix Koehler, Simon Niedermayr, Rüdiger Westermann, and Nils Thuerey. APEBench: A Benchmark for Autoregressive Neural Emulators of PDEs. In *NeurIPS*, 2024.

[25] Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew M. Stuart, and Anima Anandkumar. Fourier Neural Operator for Parametric Partial Differential Equations. In *ICLR*, 2021.

[26] Zongyi Li, Hongkai Zheng, Nikola B. Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-Informed Neural Operator for Learning Partial Differential Equations. *CoRR*, abs/2111.03794, 2021.

[27] Phillip Lippe, Bas Veeling, Paris Perdikaris, Richard E. Turner, and Johannes Brandstetter. PDE-Refiner: Achieving Accurate Long Rollouts with Neural PDE Solvers. In *NeurIPS*, 2023.

[28] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nat. Mach. Intell.*, 3(3):218–229, 2021.

[29] Yining Luo, Yingfa Chen, and Zhen Zhang. CFDBench: A large-scale benchmark for machine learning methods in fluid dynamics. *arXiv preprint arXiv:2310.05963*, 2023.

[30] Kin G. Olivares, Federico Garza, David Luo, Cristian Challu, Max Mergenthaler, and Artur Dubrawski. HierarchicalForecast: A Reference Framework for Hierarchical Forecasting in Python. *CoRR*, abs/2207.03517, 2022.

[31] Anastasios Panagiotelis, George Athanasopoulos, Puwasala Gamakumara, and Rob J Hyndman. Forecast reconciliation: A geometric view with new insights on bias correction. *International Journal of Forecasting*, 37(1):343–359, 2021.

[32] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning Mesh-Based Simulation with Graph Networks. In *ICLR*, 2021.

[33] 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. *J. Comput. Phys.*, 378:686–707, 2019.

[34] Syama Sundar Rangapuram, Lucien D. Werner, Konstantinos Benidis, Pedro Mercado, Jan Gasthaus, and Tim Januschowski. End-to-End Learning of Coherent Probabilistic Forecasts for Hierarchical Time Series. In *ICML*, pages 8832–8843, 2021.

[35] Syama Sundar Rangapuram, Shubham Kapoor, Rajbir-Singh Nirwan, Pedro Mercado, Tim Januschowski, Yuyang Wang, and Michael Bohlke-Schneider. Coherent Probabilistic Forecasting of Temporal Hierarchies. In *AISTATS*, pages 9362–9376, 2023.

[36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *MICCAI (3)*, pages 234–241, 2015.

[37] Yulia Rubanova, Alvaro Sanchez-Gonzalez, Tobias Pfaff, and Peter W. Battaglia. Constraint-based graph network simulator. In *ICML*, pages 18844–18870, 2022.

[38] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter W. Battaglia. Learning to Simulate Complex Physics with Graph Networks. In *ICML*, pages 8459–8468, 2020.

[39] Connor Schenck and Dieter Fox. SPNets: Differentiable Fluid Dynamics for Deep Neural Networks. In *CoRL*, pages 317–335, 2018.

[40] Kimberly L. Stachenfeld, Drummond Buschman Fielding, Dmitrii Kochkov, Miles D. Cranmer, Tobias Pfaff, Jonathan Godwin, Can Cui, Shirley Ho, Peter W. Battaglia, and Alvaro Sanchez-Gonzalez. Learned Simulators for Turbulence. In *ICLR*, 2022.

[41] Souhaib Ben Taieb, James W. Taylor, and Rob J. Hyndman. Coherent Probabilistic Forecasts for Hierarchical Time Series. In *ICML*, pages 3348–3357, 2017.

[42] Roger Temam. Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires (I). *Archive for Rational Mechanics and Analysis*, 32(2):135–153, 1969.

[43] Jonathan Tompson, Kristofer Schlachter, Pablo Sprechmann, and Ken Perlin. Accelerating Eulerian Fluid Simulation With Convolutional Networks. In *ICML*, pages 3424–3433, 2017.

[44] Asterios Tsiourvas, Wei Sun, Georgia Perakis, Pin-Yu Chen, and Yada Zhu. Learning Optimal Projection for Forecast Reconciliation of Hierarchical Time Series. In *ICML*, pages 48713–48727, 2024.

[45] Shanika L Wickramasuriya, George Athanasopoulos, and Rob J Hyndman. Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *Journal of the American Statistical Association*, 114(526):804–819, 2019.

Appendix Table of Contents

A Theoretical Analysis **13**

A.1 Operator Perturbations and Rollout MSE 18

A.2 Signed Correction Alignment and Analytic Strength 18

B Exact Periodic Projection Code **19**

C Forecasting Variant Details **20**

D Approximate-Regime Extensions **20**

D.1 Boundary-Preserving Approximate Cleanup Details 20

D.2 Dense Screened-Shift Sweep 22

D.3 Controlled Mismatch Details 22

D.4 Geometry-Aware Evaluator-Side Variants 23

D.5 Qualitative Mechanism Diagnostics 23

D.6 Constructive Rule in the Approximate Regime 24

E Hierarchical Forecasting Support **26**

E.1 Task Setup 26

E.2 Results on Real Data 28

F Additional Experiments and Statistical Analyses **28**

F.1 Matched-Checkpoint Projection Toggles 28

F.2 Measured Operator-Error Propagation 29

F.3 Numerical Norm Audit and Its Coverage 30

F.4 Dependence-Aware Correlation Statistics 30

F.5 Burgers Mass-Conservation Mechanism 31

F.6 Prospective One-Step Selection on Disjoint Test Trajectories 31

F.7 Adaptive-Gating Sensitivity 33

F.8 Metric Definitions and Aggregation 33

G Limitations **33**

H Reproducibility **34**

I List of Notation **35**

A Theoretical Analysis

The following results formalize the mechanism separating exact and approximate operators used throughout the experiments. Theorem 2 shows that exact-on-target repair removes additive target distortion. Proposition 5 shows that solver residual controls only implementation error around a fixed operator. Theorem 6 shows that trajectorywise target distortion enters rollout bounds directly. Proposition 9 shows why raw or near-raw cleanup can be optimal when the cleanup direction is misaligned with the target.

Proposition 1 (Orthogonal projection identity)**.** *Let $\mathcal{H}$ be a finite-dimensional real inner product space, let $\mathcal{M} \subset \mathcal{H}$ be a linear subspace, and let $P : \mathcal{H} \to \mathcal{M}$ be the orthogonal projector onto $\mathcal{M}$. Then, for every $x \in \mathcal{H}$ and $y \in \mathcal{M}$, we have*

$$\|Px - y\|^2 = \|x - y\|^2 - \|(I - P)x\|^2 \leq \|x - y\|^2. \tag{14}$$

Applied to the periodic divergence-free subspace with $P = \Pi_{\mathrm{div}}$, the projection identity is Equation 8.

Proof. Because P is the orthogonal projector, $Px - y \in \mathcal{M}$ and $(I - P)x \in \mathcal{M}^{\perp}$. Since

$$x - y = (Px - y) + (I - P)x, \tag{15}$$

the two summands are orthogonal. The Pythagorean theorem therefore gives

$$\|x - y\|^2 = \|Px - y\|^2 + \|(I - P)x\|^2, \tag{16}$$

which rearranges to Equation 14. The inequality follows because the last term is nonnegative. □

Theorem 2 (Exact-on-target repair removes additive target distortion)**.** *Let $\mathcal{H}$ be a finite-dimensional real inner product space, let $\mathcal{M} \subset \mathcal{H}$ be a target set, and let $T : \mathcal{H} \to \mathcal{H}$ satisfy*

$$T(y) = y \text{ for every } y \in \mathcal{M}. \tag{17}$$

Assume that T is L_T-Lipschitz on $\mathcal{H}$. Then, for every $x \in \mathcal{H}$ and every $y \in \mathcal{M}$,

$$\|T(x) - y\| = \|T(x) - T(y)\| \leq L_T \|x - y\|. \tag{18}$$

In particular, if $L_T \leq 1$, then post hoc repair with T cannot worsen one-step error to any target on $\mathcal{M}$.
Now, let $F : \mathcal{H} \to \mathcal{H}$ be a predictor, let $(y_t)_{t \geq 0} \subset \mathcal{M}$ be a target trajectory, and define the repaired rollout $x_{t+1} = T(F(x_t))$. If F is L_F-Lipschitz and $\delta_t := \|F(y_t) - y_{t+1}\|$, then the rollout error satisfies

$$\|x_{t+1} - y_{t+1}\| \leq L_T L_F \|x_t - y_t\| + L_T \delta_t. \tag{19}$$

Consequently, for every $t \geq 1$,

$$\|x_t - y_t\| \leq (L_T L_F)^t \|x_0 - y_0\| + \sum_{s=0}^{t-1} (L_T L_F)^{t-1-s} L_T \delta_s. \tag{20}$$

Proof. Equation 18 follows immediately from Equation 17 and the Lipschitz property:

$$\|T(x) - y\| = \|T(x) - T(y)\| \leq L_T \|x - y\|. \tag{21}$$

For the rollout recursion, because $y_{t+1} \in \mathcal{M}$, Equation 17 gives $T(y_{t+1}) = y_{t+1}$. Therefore,

$$\begin{aligned}
\|x_{t+1} - y_{t+1}\| &= \|T(F(x_t)) - T(y_{t+1})\| && (22)\\
&\leq L_T \|F(x_t) - y_{t+1}\| && (23)\\
&\leq L_T \|F(x_t) - F(y_t)\| + L_T \|F(y_t) - y_{t+1}\| && (24)\\
&\leq L_T L_F \|x_t - y_t\| + L_T \delta_t, && (25)
\end{aligned}$$

which is Equation 19. Iterating this affine recursion gives Equation 20, without a contraction assumption.

□

Corollary 3 (Conditional contraction and steady-state error)**.** *Under Theorem 2, if $L_T L_F < 1$ and $\delta_t \leq \bar{\delta}$ for all t, then*

$$\limsup_{t \to \infty} \|x_t - y_t\| \leq \frac{L_T \bar{\delta}}{1 - L_T L_F}. \tag{26}$$

Proof. Sum the geometric series in Equation 20 and let $t \to \infty$. □

This corollary is conditional; the empirical projection gains do not require verification of a composite contraction. Exact orthogonal projection guarantees one-step no-harm, not by itself bounded autoregressive error.

Corollary 4 (Automatic exactness for linear idempotent repair)**.** *Let $\mathcal{M}$ be a linear subspace of $\mathcal{H}$. If $T : \mathcal{H} \to \mathcal{H}$ is linear and idempotent with $\mathrm{range}(T) = \mathcal{M}$, then $T(y) = y$ for every $y \in \mathcal{M}$. Thus, Equation 17 holds automatically. The optimal Lipschitz constant is $\|T\|_{\mathrm{op}}$, the operator norm induced by the norm defined by the chosen inner product. Orthogonal projection onto a nonzero target subspace has $\|T\|_{\mathrm{op}} = 1$, with the zero-subspace projector as the degenerate norm-zero case, while exact-but-nonorthogonal repair operators can have $\|T\|_{\mathrm{op}} > 1$ without introducing any additive target-distortion term.*

Proof. For any $y \in \mathcal{M}$, there exists z with $y = Tz$, so $Ty = T^2 z = Tz = y$. For linear maps, the optimal Lipschitz constant with respect to the induced norm is $\|T\|_{\mathrm{op}}$. □

Remark 1. Theorem 2 and Corollary 4 give the abstract version of our exact-regime rule. Exact-but-nonorthogonal operators do not enjoy the Pythagorean identity from the periodic FFT projector, but they still avoid the additive operator-bias term that appears in approximate cleanup bounds. Their risk is conditioning through $\|T\|_{\mathrm{op}}$, not target distortion. This absence of target distortion is why exact bottom-up reconciliation is still a principled baseline even though it is not an orthogonal projector.

Proposition 5 (Poisson residual controls only solver error around the chosen operator)**.** *Let $A_\lambda \in \mathbb{R}^{n\times n}$ be symmetric positive definite, let $\mu_\lambda := \lambda_{\min}(A_\lambda) > 0$, and let $D \in \mathbb{R}^{n\times N}$ together with $G \in \mathbb{R}^{N\times n}$ define the cleanup family $Q^\star(x) := x - Gp^\star(x)$, where $A_\lambda p^\star(x) = -Dx$. Let $p(x)$ be any approximate linear-system solve with residual $r(x) := -Dx - A_\lambda p(x)$, and define the implemented cleanup $Q(x) := x - Gp(x)$. Then, for every $x \in \mathbb{R}^N$,*

$$p(x) - p^\star(x) = -A_\lambda^{-1} r(x), \tag{27}$$

$$Q(x) - Q^\star(x) = GA_\lambda^{-1} r(x), \tag{28}$$

$$\|Q(x) - Q^\star(x)\|_2 \le \frac{\|G\|_2}{\mu_\lambda} \|r(x)\|_2, \tag{29}$$

$$\|DQ(x) - DQ^\star(x)\|_2 \le \frac{\|DG\|_2}{\mu_\lambda} \|r(x)\|_2. \tag{30}$$

Moreover, for every target $y \in \mathbb{R}^N$,

$$\big|\|Q(x) - y\|_2 - \|Q^\star(x) - y\|_2\big| \le \frac{\|G\|_2}{\mu_\lambda} \|r(x)\|_2, \tag{31}$$

and in particular

$$\|Q(y) - y\|_2 \ge \|Q^\star(y) - y\|_2 - \frac{\|G\|_2}{\mu_\lambda} \|r(y)\|_2. \tag{32}$$

Consequently, if a sequence of stronger solves p_k satisfies $\|r_k(x)\|_2 \to 0$ for every x, then $Q_k(x) \to Q^\star(x)$ pointwise. A lower linear-system residual therefore shrinks the residual-controlled upper bound on the implementation gap, and residual convergence makes the implementation pointwise faithful to the same operator family $Q^\star$; it does not remove any intrinsic target distortion already present in $Q^\star$.

Proof. By the definitions of $p^\star(x)$ and $r(x)$,

$$A_\lambda(p^\star(x) - p(x)) = -Dx - A_\lambda p(x) = r(x), \tag{33}$$

which proves Equation 27. Substituting into the velocity update gives

$$Q(x) - Q^\star(x) = -G(p(x) - p^\star(x)) = GA_\lambda^{-1} r(x), \tag{34}$$

which is Equation 28. Since A_λ is symmetric positive definite,

$$\|A_\lambda^{-1}\|_2 = \frac{1}{\mu_\lambda}. \tag{35}$$

Therefore,

$$\|Q(x) - Q^\star(x)\|_2 \le \|G\|_2 \|A_\lambda^{-1}\|_2 \|r(x)\|_2 = \frac{\|G\|_2}{\mu_\lambda} \|r(x)\|_2, \tag{36}$$

$$\|DQ(x) - DQ^\star(x)\|_2 \le \|DG\|_2 \|A_\lambda^{-1}\|_2 \|r(x)\|_2 = \frac{\|DG\|_2}{\mu_\lambda} \|r(x)\|_2, \tag{37}$$

which proves Equations 29 and 30. The reverse triangle inequality then yields

$$|\|Q(x) - y\|_2 - \|Q^\star(x) - y\|_2| \le \|Q(x) - Q^\star(x)\|_2, \tag{38}$$

and Equation 31 follows from the previous bound. Setting $x = y$ gives Equation 32. Finally, if $\|r_k(x)\|_2 \to 0$, then Equation 29 implies $\|Q_k(x) - Q^\star(x)\|_2 \to 0$ for every x. □

Remark 2. Proposition 5 is the formal version of our stronger-solver negative controls. Direct, CG, SOR, MG, and Jacobi can all be compared to the exact application of their own underlying operator family. Lower residual only shrinks the residual-controlled bound in Equation 29; it does not change the target-distortion floor in Equation 32. The residual-distortion separation is exactly why a near-exact solve of a mismatched operator can still lose to a milder screened or geometry-aware family.

Theorem 6 (Trajectorywise target distortion enters rollout bounds)**.** *Let $\mathcal{H}$ be a finite-dimensional real inner product space, let $\mathcal{M} \subset \mathcal{H}$ be a target set, and let $F : \mathcal{H} \to \mathcal{H}$ be an L_F-Lipschitz predictor. For $i \in \{1, 2\}$, let $T_i : \mathcal{H} \to \mathcal{H}$ be an L_i-Lipschitz cleanup operator. Consider a target trajectory $(y_t)_{t\ge0} \subset \mathcal{M}$ and the repaired rollouts $x_{t+1}^{(i)} = T_i(F(x_t^{(i)}))$, with $e_t^{(i)} := x_t^{(i)} - y_t$. Define the raw model's one-step target error as $\delta_t := \|F(y_t) - y_{t+1}\|$ and the target-distortion sequence along that trajectory as $\beta_{i,t} := \|T_i(y_t) - y_t\|$, $t \ge 0$. Then, for every $t \ge 0$ and each $i \in \{1, 2\}$,*

$$\|e_{t+1}^{(i)}\| \le L_i L_F \|e_t^{(i)}\| + L_i \delta_t + \beta_{i,t+1}. \tag{39}$$

Consequently, if $b_0^{(i)} \ge \|e_0^{(i)}\|$ and

$$b_{t+1}^{(i)} := L_i L_F b_t^{(i)} + L_i \delta_t + \beta_{i,t+1}, \tag{40}$$

then $\|e_t^{(i)}\| \le b_t^{(i)}$ for all $t \ge 0$.

Proof. Fix $i \in \{1, 2\}$. Using the rollout definition, we add and subtract $T_i(F(y_t))$ and $T_i(y_{t+1})$, obtaining

$$\begin{aligned}
\|e_{t+1}^{(i)}\| &= \|T_i(F(x_t^{(i)})) - y_{t+1}\| && (41)\\
&\le \|T_i(F(x_t^{(i)})) - T_i(F(y_t))\| + \|T_i(F(y_t)) - T_i(y_{t+1})\| + \|T_i(y_{t+1}) - y_{t+1}\| && (42)\\
&\le L_i \|F(x_t^{(i)}) - F(y_t)\| + L_i \|F(y_t) - y_{t+1}\| + \beta_{i,t+1} && (43)\\
&\le L_i L_F \|e_t^{(i)}\| + L_i \delta_t + \beta_{i,t+1}, && (44)
\end{aligned}$$

which proves Equation 39. The recursion in Equation 40 therefore upper-bounds the true error by induction on t. □

Corollary 7 (Smaller target distortion gives a tighter autoregressive bound)**.** *Under the setup of Theorem 6, assume*

$$\begin{aligned}
b_0^{(1)} &\le b_0^{(2)},\\
L_1 &\le L_2,\\
\beta_{1,t} &\le \beta_{2,t} \text{ for every } t \ge 1.
\end{aligned} \tag{45}$$

Then we have

$$b_t^{(1)} \le b_t^{(2)} \text{ for all } t \ge 0. \tag{46}$$

Thus, under matched raw dynamics, any operator family with no larger Lipschitz factor and no larger trajectorywise target distortion along the same target path has a uniformly tighter autoregressive bound at every horizon.

Proof. The claim follows by induction. It holds at $t = 0$ by assumption. If it holds at time t, then

$$\begin{aligned}
b_{t+1}^{(1)} &= L_1 L_F b_t^{(1)} + L_1 \delta_t + \beta_{1,t+1} && (47)\\
&\le L_2 L_F b_t^{(1)} + L_2 \delta_t + \beta_{2,t+1} && (48)\\
&\le L_2 L_F b_t^{(2)} + L_2 \delta_t + \beta_{2,t+1} && (49)\\
&= b_{t+1}^{(2)}, && (50)
\end{aligned}$$

which closes the induction. □

Corollary 8 (Uniform distortion and conditional steady-state bounds)**.** *Under the setup of Theorem 6, if the distortion is uniformly bounded on the whole target set,*

$$\beta_i := \sup_{y \in \mathcal{M}} \|T_i(y) - y\| < \infty, \tag{51}$$

then $\beta_{i,t} \le \beta_i$ for every t, and therefore

$$\|e_{t+1}^{(i)}\| \le L_i L_F \|e_t^{(i)}\| + L_i \delta_t + \beta_i. \tag{52}$$

Finally, if $L_i L_F < 1$, $\delta_t \le \bar{\delta}$ for all t, and $\beta_{i,t} \le \bar{\beta}_i$ for all t, then

$$\limsup_{t \to \infty} \|e_t^{(i)}\| \le \frac{L_i \bar{\delta} + \bar{\beta}_i}{1 - L_i L_F}. \tag{53}$$

Under Equation 51, one may take $\bar{\beta}_i = \beta_i$. Exact-on-target operators are the special case $\bar{\beta}_i = 0$.

Proof. If Equation 51 holds, then $\beta_{i,t+1} \le \beta_i$ for every t, and Equation 52 follows immediately from Equation 39.

Finally, if $\delta_t \le \bar{\delta}$, $\beta_{i,t} \le \bar{\beta}_i$, and $L_i L_F < 1$, then Equation 39 yields the scalar affine recursion

$$\|e_{t+1}^{(i)}\| \le L_i L_F \|e_t^{(i)}\| + (L_i \bar{\delta} + \bar{\beta}_i). \tag{54}$$

Iterating this recursion and summing the geometric series gives Equation 53. □

Remark 3. Theorem 6 uses a trajectorywise formulation rather than a bare global supremum. For a nonexact linear operator on an unbounded target subspace, the quantity in Equation 51 is typically infinite. The right object for our rollout diagnostics is therefore the distortion actually seen along the benchmark trajectory, with the uniform bound kept only as a stronger corollary when the target set is bounded or the operator is exact on it. Under this formulation, the message stays the same: once the raw predictor F and rollout regime are fixed, the operator family enters the bound through a Lipschitz factor and a target-distortion term, while Proposition 5 says that linear-system residual only measures how closely we implement a chosen operator family.

Proposition 9 (Controlled-mismatch optimum)**.** *Let $\mathcal{H}$ be a finite-dimensional real inner product space, let $x \in \mathcal{H}$ be a raw forecast, let $y \in \mathcal{H}$ be the corresponding target, and let $Q : \mathcal{H} \to \mathcal{H}$ be any cleanup operator. Define the cleanup increment $c := Q(x) - x$ and the blended controlled-mismatch family $x_\alpha := (1-\alpha)x + \alpha Q(x) = x + \alpha c$, $\alpha \in [0,1]$. Then the one-step squared error $\phi(\alpha) := \|x_\alpha - y\|^2$ is the convex quadratic*

$$\phi(\alpha) = \|x - y\|^2 + 2\alpha\langle x - y, c\rangle + \alpha^2 \|c\|^2. \tag{55}$$

If $c \neq 0$, the minimizing blend strength is therefore

$$\alpha^\star = \mathrm{clip}_{[0,1]}\left(-\frac{\langle x - y, c\rangle}{\|c\|^2}\right), \tag{56}$$

while every $\alpha \in [0,1]$ is optimal when $c = 0$.
Several immediate consequences follow. If $c \neq 0$ and $\langle x - y, c\rangle \ge 0$, then $\alpha^\star = 0$, so raw forecasting is the unique optimizer, and every nonzero amount of cleanup worsens the one-step MSE.
Full cleanup is no worse than raw forecasting if and only if $2\langle x - y, c\rangle + \|c\|^2 \le 0$, and it is strictly better if and only if the inequality is strict.

Proof. Set $e := x - y$. Since $x_\alpha - y = e + \alpha c$, expanding the square gives

$$\phi(\alpha) = \|e + \alpha c\|^2 = \|e\|^2 + 2\alpha\langle e, c\rangle + \alpha^2 \|c\|^2, \tag{57}$$

which is Equation 55. If $c \neq 0$, the blend objective is a strictly convex quadratic, so its unconstrained minimizer is $-\langle e, c\rangle / \|c\|^2$. Clipping to $[0,1]$ yields Equation 56. The raw-optimality claim is then immediate, and the full-cleanup condition follows from

$$\phi(1) - \phi(0) = 2\langle x - y, c\rangle + \|c\|^2. \tag{58}$$

□

Corollary 10 (Controlled mismatch under orthogonal projection)**.** *In the setup of Proposition 9, suppose $Q = P$ is the orthogonal projector onto a closed subspace $\mathcal{M} \subset \mathcal{H}$ and $y \in \mathcal{M}$. Then,*

$$\phi(\alpha) = \|Px - y\|^2 + (1-\alpha)^2\|(I-P)x\|^2, \tag{59}$$

so $\alpha = 1$ is a minimizer, uniquely unless $x \in \mathcal{M}$; when $x \in \mathcal{M}$, every $\alpha \in [0,1]$ is optimal.

Proof. Decompose

$$x - y = (Px - y) + (I-P)x. \tag{60}$$

Because $y \in \mathcal{M}$ and P is the orthogonal projector onto $\mathcal{M}$, the first term lies in $\mathcal{M}$ and the second lies in $\mathcal{M}^\perp$; hence, they are orthogonal. Moreover, $c = Px - x = -(I-P)x$, so

$$x_\alpha - y = (Px - y) + (1-\alpha)(I-P)x. \tag{61}$$

Taking norms and using orthogonality proves Equation 59. The right-hand side is minimized at $\alpha = 1$, and the minimizer is unique unless $(I-P)x = 0$, in which case every $\alpha \in [0,1]$ is optimal. □

Remark 4. Proposition 9 turns the controlled-mismatch experiment into a precise diagnostic. Because the true residual is $y-x$, any improvement from some nonzero cleanup strength requires $\langle y-x, c\rangle > 0$, equivalently $\langle x-y, c\rangle < 0$. Corollary 10 gives the exact orthogonal regime: the cleanup increment is $c = Px - x = -(I-P)x$, so full cleanup cancels the orthogonal-to-subspace component $(I-P)x$. In the approximate regime, this sign is not fixed, which is exactly why the best empirical operating point can move to $\alpha = 0$ or a very mild blend.

A.1 Operator Perturbations and Rollout MSE

Proposition 11 (Finite-horizon operator perturbation)**.** *Let $P, \widetilde{P}$ be linear maps on $\mathbb{R}^m$ with $\|\widetilde{P} - P\|_{\mathrm{op}} \le \epsilon$ and $\|\widetilde{P}\|_{\mathrm{op}} \le K$. Let F be L-Lipschitz on the states used below, and consider $x_{t+1} = PF(x_t)$ and $z_{t+1} = \widetilde{P}F(z_t)$ with $x_0 = z_0$. If $\|F(x_t)\| \le B$ along the reference trajectory, then, for $d_t = \|z_t - x_t\|$,*

$$d_{t+1} \le KLd_t + \epsilon B, \qquad d_t \le \epsilon B \sum_{j=0}^{t-1} (KL)^j. \tag{62}$$

For any target y_t and $a_t = x_t - y_t$,

$$|\mathrm{MSE}(z_t, y_t) - \mathrm{MSE}(x_t, y_t)| \le \frac{2\|a_t\|d_t + d_t^2}{m}. \tag{63}$$

Proof. Adding and subtracting $\widetilde{P}F(x_t)$ gives

$$\|\widetilde{P}F(z_t) - PF(x_t)\| \le \|\widetilde{P}\|_{\mathrm{op}}\|F(z_t) - F(x_t)\| + \|\widetilde{P} - P\|_{\mathrm{op}}\|F(x_t)\| \le KLd_t + \epsilon B.$$

Iterating from $d_0 = 0$ proves the first bound. Writing $h_t = z_t - x_t$, $m\,\Delta\mathrm{MSE}_t = 2\langle a_t, h_t\rangle + \|h_t\|^2$; Cauchy–Schwarz and the triangle inequality give the second. □

The perturbation bound is uniform for $KL < 1$, at most $t\epsilon B$ for $KL = 1$, and may grow geometrically for $KL > 1$. This is a finite-horizon comparison, not an unconditional stability guarantee. Appendix F.2 reports measured perturbations under known operator changes.

A.2 Signed Correction Alignment and Analytic Strength

Proposition 12 (Validation-risk identity)**.** *Let (x, y) be labeled one-step forecast–target pairs, $c_T(x) = T(x) - x$, and assume finite second moments. Define $A_T = \mathbb{E}\langle x - y, c_T(x)\rangle$, $B_T = \mathbb{E}\|c_T(x)\|^2$, and $R_1(T) = \mathbb{E}\|T(x) - y\|^2$. For $T_\alpha = I + \alpha(T - I)$,*

$$R_1(T_\alpha) - R_1(I) = 2\alpha A_T + \alpha^2 B_T. \tag{64}$$

If $B_T > 0$, the minimizer on $[0,1]$ is $\alpha_T^\star = \mathrm{clip}_{[0,1]}(-A_T/B_T)$. With $\Gamma_T = -2A_T/B_T$, full-strength repair is strictly favorable if and only if $\Gamma_T > 1$, and some positive strength is favorable if and only if $\Gamma_T > 0$. If $B_T = 0$, repair equals Raw almost surely on the validation distribution.

Proof. Apply Proposition 9 to each pair and take expectations. The derivative of the resulting quadratic is $2A_T + 2\alpha B_T$. Clipping its zero to $[0, 1]$ gives the minimizer, and comparing the quadratic with zero gives the stated signs. The same algebra holds exactly for empirical averages. □

For MSE units, divide A_T, B_T, and the risk by state dimension m; all signs and optimal strengths are unchanged. The empirical score $S(T) = 2\alpha_T^\star A_T + (\alpha_T^\star)^2 B_T \le 0$ includes $S(\mathrm{Raw}) = 0$. Select the damped map $T_{\alpha_T^\star}$ of the minimum-score family, choosing Raw if no score is strictly negative. A fixed order resolves any remaining family ties. All candidates reuse the same raw predictions.

If P is orthogonal and $y \in \mathrm{range}(P)$, then $A_P = -B_P$ and $\Delta_1(P) = -B_P \le 0$, recovering full-strength optimality. If $B_P = 0$, every strength is equivalent. The same result holds for projection onto an affine subspace by translation, covering the fixed initial-mass hyperplane in Burgers.

Finite-horizon link and scope. For repaired dynamics $T \circ F$ with Lipschitz factor κ_T on the relevant states, let $\epsilon_{T,t} = \|T(F(y_t)) - y_{t+1}\|$ and $e_t^T = x_t^T - y_t$. The same add-and-subtract argument gives

$$\|e_{T_{\mathrm{eval}}}^T\| \le \kappa_T^{T_{\mathrm{eval}}} \|e_0\| + \sum_{t=0}^{T_{\mathrm{eval}}-1} \kappa_T^{T_{\mathrm{eval}}-1-t} \epsilon_{T,t}. \tag{65}$$

One-step alignment controls consistency error on validation pairs, while amplification determines how perturbations accumulate after feedback. A smaller one-step risk alone does not order long-horizon errors or these trajectorywise bounds. Operationally, assess corrections at the proposed strength, retain Raw when none is favorable, and use rollout validation for statistically ambiguous or amplification-uncertain choices. Norm audits identify expansion and guide this fallback; numerical cleanup norms alone do not certify the composite factor κ_T.

B Exact Periodic Projection Code

Listing 1 gives the compact PyTorch implementation of the exact periodic projection used in the NS-128 track.

```python
import math
import torch


def spectral_grids(
    height: int,
    width: int,
    device: torch.device,
    domain_size: float = 2 * math.pi,
):
    dx = domain_size / width
    dy = domain_size / height
    kx = 2 * math.pi * torch.fft.rfftfreq(width, d=dx)
    ky = 2 * math.pi * torch.fft.fftfreq(height, d=dy)
    ky_grid, kx_grid = torch.meshgrid(ky, kx, indexing="ij")
    return kx_grid.to(device), ky_grid.to(device)


def curl_from_streamfunction(
    stream: torch.Tensor,
    domain_size: float = 2 * math.pi,
) -> torch.Tensor:
    flat_stream = stream.reshape(-1, *stream.shape[-2:])
    height, width = flat_stream.shape[-2:]
    kx, ky = spectral_grids(height, width, flat_stream.device, domain_size)
    stream_hat = torch.fft.rfft2(flat_stream.float())
    u_hat = 1j * ky * stream_hat
    v_hat = -1j * kx * stream_hat
    u = torch.fft.irfft2(u_hat, s=(height, width))
    v = torch.fft.irfft2(v_hat, s=(height, width))
    velocity = torch.stack([u, v], dim=1)
    return velocity.reshape(*stream.shape[:-2], 2, height, width).to(stream.dtype)


def periodic_hodge_projection(
    velocity: torch.Tensor,
    domain_size: float = 2 * math.pi,
) -> torch.Tensor:
```

```
    """Project periodic 2D velocity fields onto the divergence-free subspace."""
    if velocity.shape[-3] != 2:
        raise ValueError("velocity must have shape (..., 2, H, W)")

    original_dtype = velocity.dtype
    leading = velocity.shape[:-3]
    flat = velocity.reshape(-1, *velocity.shape[-3:])
    height, width = flat.shape[-2:]
    kx, ky = spectral_grids(height, width, flat.device, domain_size)

    u = flat[:, 0].float()
    v = flat[:, 1].float()
    u_hat = torch.fft.rfft2(u)
    v_hat = torch.fft.rfft2(v)
    k_sq = kx.square() + ky.square()
    k_sq = k_sq.clone()
    k_sq[0, 0] = 1.0

    vort_hat = 1j * kx * v_hat - 1j * ky * u_hat
    stream_hat = vort_hat / k_sq
    stream_hat[..., 0, 0] = 0.0
    # Real-valued spectral derivatives use the standard zero-Nyquist convention.
    if height % 2 == 0:
        stream_hat[..., height // 2, :] = 0.0
    if width % 2 == 0:
        stream_hat[..., :, width // 2] = 0.0
    stream = torch.fft.irfft2(stream_hat, s=(height, width))
    projected = curl_from_streamfunction(stream, domain_size)
    mean_velocity = flat.mean(dim=(-2, -1), keepdim=True)
    return (projected + mean_velocity).reshape(*leading, 2, height, width).to(original_dtype)
```

Listing 1: Exact periodic Hodge projection for velocity tensors with shape `(..., 2, H, W)`.

C Forecasting Variant Details

On the periodic NS-128 main track, the raw baseline (Raw) uses no projection. The post hoc projection baseline (PostHoc) trains raw and projects each prediction only at evaluation time before feeding it back. The soft-divergence training variant (SoftDiv) optimizes

$$\mathcal{L}_{\text{soft}} = \mathcal{L}_{\text{vel}} + \lambda_{\text{div}} \|\nabla \cdot \hat{\boldsymbol{u}}\|_2^2 + \mathcal{L}_{\omega} + \mathcal{L}_{\text{spec}}. \tag{66}$$

The projected-rollout variant (Proj) applies its differentiable output projection to every predicted state during both training and evaluation, and constraint-aware projection (CAP) projects both the autoregressive state and the training target.

On the non-periodic CFDBench tracks, we use adaptive blended cleanup (AdaBlend) to denote the gated variant and include Raw and the Dirichlet-Poisson cleanup baselines AdaBlend-Jacobi, PostHoc-Jacobi, PostHoc-MG, PostHoc-SOR, Proj-Jacobi, Proj-SOR, PostHoc-CG, and Proj-CG. We also add the screened family AdaBlend-Screened, PostHoc-Screened, and Proj-Screened. The projected models are trained with the same operator family and iteration budget used at evaluation time, so the comparison is training-matched rather than an evaluator-only sweep. We use matched budgets of $k = 10$ for cavity and $k = 20$ for tube, choose the MG V-cycle count by validation search under the same rollout protocol, and use $\lambda = 8$ for the main screened-family training runs across the non-periodic experiments. Appendix D reports additional evaluation-only replicated-boundary, tapered, and GeoScreened stress tests together with the corresponding GeoScreened adaptive variants on the hardest rows.

D Approximate-Regime Extensions

D.1 Boundary-Preserving Approximate Cleanup Details

For non-periodic cavity, tube, dam, and cylinder flow, exact spectral projection is unavailable. We therefore use training-matched boundary-preserving approximate cleanup operators. The resulting Poisson solves are approximate, boundary-constrained members of the classical Chorin–Temam projection family rather than exact Leray projectors [7, 15, 21, 4, 1]. Let $\tilde{\boldsymbol{u}}$ be the raw predicted velocity. Throughout this discrete cleanup description, we use unit grid spacing; with physical

spacings $\Delta x, \Delta y$, the corresponding difference quotients include the usual $1/\Delta x$ and $1/\Delta y$ factors. The interior discrete divergence $D_h \equiv \nabla \cdot_h$ is

$$(\nabla \cdot_h \tilde{\boldsymbol{u}})_{i,j} = \tilde{u}_{i,j} - \tilde{u}_{i,j-1} + \tilde{v}_{i,j} - \tilde{v}_{i-1,j}. \tag{67}$$

We solve the positive Dirichlet pressure system on interior cells,

$$\begin{aligned} Ap &= -\nabla \cdot_h \tilde{\boldsymbol{u}}, \\ A &:= -\Delta_h = 4I - S_x^+ - S_x^- - S_y^+ - S_y^-, \end{aligned} \tag{68}$$

where the pressure has zero-Dirichlet boundary values and $S_x^{\pm}$ and $S_y^{\pm}$ are interior shift operators under this boundary convention. The interior velocities are then updated by forward differences, with boundary values copied from the raw prediction:

$$\hat{u}_{i,j} = \tilde{u}_{i,j} - (p_{i,j+1} - p_{i,j}), \tag{69}$$

$$\hat{v}_{i,j} = \tilde{v}_{i,j} - (p_{i+1,j} - p_{i,j}), \tag{70}$$

Equivalently, with G_{int} denoting this interior-only forward-gradient update, the velocity map is $\hat{\boldsymbol{u}} = \tilde{\boldsymbol{u}} - G_{\mathrm{int}} p$. Because the boundary velocities are copied, $D_h G_{\mathrm{int}} \neq -A$ on divergence stencils that touch copied boundary velocity values; even an exact solve of $Ap = -D_h \tilde{\boldsymbol{u}}$ need not make the full interior divergence exactly zero. We denote the resulting operator by $\widetilde{\Pi}_{\mathrm{bc}}$. Despite the $\widetilde{\Pi}$ notation, these boundary-aware maps are cleanup operators rather than exact projections in the algebraic sense. Copying raw boundary values generally breaks both exact divergence annihilation and idempotence on the full velocity field.

Boundary-aware Poisson–Hodge family. We study two approximate operator families. The first is the boundary-aware Poisson–Hodge family. It includes a fixed-budget Jacobi iteration, a geometric multigrid V-cycle that we abbreviate as MG and equip with bilinear restriction, prolongation, and Jacobi smoothing, a red-black successive over-relaxation iteration that we abbreviate as SOR and pair with a finite-budget damped relaxation factor, and a matrix-free conjugate-gradient solve that we abbreviate as CG for this same positive Dirichlet system,

$$Ap = -\nabla \cdot_h \tilde{\boldsymbol{u}}. \tag{71}$$

The fifth member of this family is an exact discrete-Dirichlet solve for this same pressure operator, implemented by diagonalizing A in the separable sine basis. Writing the interior pressure as $P \in \mathbb{R}^{(H-2)\times(W-2)}$ and the orthonormal type-I discrete sine transform (DST-I) matrices as $S_{H-2} \in \mathbb{R}^{(H-2)\times(H-2)}$ and $S_{W-2} \in \mathbb{R}^{(W-2)\times(W-2)}$, we solve

$$\widehat{R} = -S_{H-2}(\nabla \cdot_h \tilde{\boldsymbol{u}})_{\mathrm{int}} S_{W-2}^{\top}, \tag{72}$$

$$\widehat{P}_{ij} = \frac{\widehat{R}_{ij}}{\lambda_i^{(H)} + \lambda_j^{(W)}}, \tag{73}$$

$$P = S_{H-2}^{\top} \widehat{P} S_{W-2}, \tag{74}$$

where $\lambda_i^{(H)}$ and $\lambda_j^{(W)}$ are the one-dimensional Dirichlet eigenvalues for $i = 1, \ldots, H-2$ and $j = 1, \ldots, W-2$. We refer to the resulting direct cleanup operator as Direct. Jacobi, MG, SOR, CG, and Direct therefore share the same boundary-aware discrete Poisson operator but differ in how aggressively they solve it, letting us separate solver under-convergence from operator mismatch in the approximate regime.

Screened family. The second family is a screened variant related to the Helmholtz–Hodge cleanup perspective of Bhatia et al. [5]. Instead of solving the pure Poisson system, we solve

$$(A + \lambda I)p = -\nabla \cdot_h \tilde{\boldsymbol{u}}, \tag{75}$$

with the same boundary-aware stencil and interior-only update $\hat{\boldsymbol{u}} = \tilde{\boldsymbol{u}} - G_{\mathrm{int}} p$. Equation 75 shrinks the pressure response as the screened shift λ increases, tending to make the cleanup milder and, in our sweeps, trading divergence removal against lower target distortion. In the main training-matched comparisons, we use a fixed screened shift $\lambda = 8$ and the same training and evaluation iteration budgets as the Jacobi family, so this comparison isolates operator family rather than compute

Table 8: Dense screened-shift sweep. As operator mismatch grows, the best screened cleanup shifts toward larger λ and behavior closer to the identity, while the raw reference can still remain best overall on the hardest rows. Smaller values are better in the metric columns.

Dataset	Best screened shift with $\lambda > 0$	Raw rollout MSE@20	Best screened PostHoc rollout MSE@20	GT distortion MSE
CFDBench cavity	16	$(2.190 \pm 0.170) \times 10^{-1}$	$(2.160 \pm 0.160) \times 10^{-1}$	1.223×10^{-4}
CFDBench tube	1	$(3.200 \pm 0.900) \times 10^{-2}$	$(2.900 \pm 0.091) \times 10^{-2}$	1.280×10^{-4}
CFDBench dam	64	$(2.476 \pm 0.118) \times 10^{-4}$	$(2.499 \pm 0.108) \times 10^{-4}$	7.469×10^{-7}
CFDBench cylinder	64	$(2.000 \pm 0.863) \times 10^{-3}$	$(2.000 \pm 0.894) \times 10^{-3}$	3.430×10^{-6}

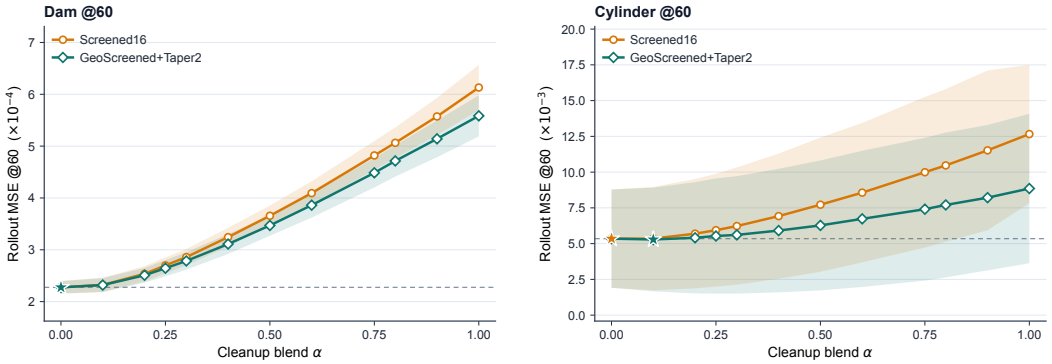


Figure 3: Controlled mismatch curves for CFDBench Dam@60 and Cylinder@60, namely the 60-step dam and cylinder long-horizon rows. Increasing cleanup strength lowers divergence but usually worsens rollout error once the operator is mismatched to the data manifold.

budget. Separately, we run evaluator-side post hoc sweeps over $\lambda \in \{1, 2, 4, 8, 16, 32, 64\}$ on the raw CFDBench checkpoints to test whether the family-level conclusion depends on a single screened setting, and we also run a validation-selected retraining study that chooses λ on held-out rollout mean squared error (MSE) area under the curve (AUC), abbreviated MSE-AUC, before retraining the screened projected baseline on the official CFDBench splits. The following subsections report the dense screened-shift sweep, controlled-mismatch details, boundary-strip audit, and tapered and geometry-aware screened (GeoScreened) evaluator-side variants used in additional approximate-regime stress tests.

D.2 Dense Screened-Shift Sweep

The dense screened-family sweep already suggested that the useful operators are the ones that stay closest to the identity map on the hardest rows. Table 8 shows the pattern cleanly. Tube prefers strong cleanup, cavity sits on a broad intermediate plateau, and the hardest dam and cylinder rows move back toward the raw frontier as screening weakens toward the identity, with Raw still remaining slightly best overall. The table therefore reports the best screened shift among $\lambda > 0$ together with the separate raw reference.

D.3 Controlled Mismatch Details

Figure 3 shows the full controlled-mismatch curves corresponding to Table 5. The curves make the trade-off behind the table explicit. On Dam@60, both cleanup families move almost immediately away from the raw optimum as α increases. Final-step rollout MSE rises while the cleaned state becomes more divergence-suppressed. Cylinder@60 shows the same pattern for Screened16, and the GeoScreened+Taper2 exception is deliberately small, with only a shallow near-identity minimum at $\alpha = 0.1$ before the curve turns upward. Thus, the controlled blend is not selecting a hidden stronger projection; it is exposing that the approximate cleanup direction is mostly misaligned with the rollout-error direction on these long-horizon non-periodic rows. This controlled-mismatch pattern is the visual counterpart of Proposition 9. Once the cleanup increment has the wrong or only weakly favorable alignment with the target residual, increasing enforcement strength can monotonically improve the measured constraint residual while worsening the forecasting objective.

Table 9: Boundary-strip diagnostics on held-out targets under a boundary-focused audit. Stronger cleanup suppresses core divergence more aggressively, but it can also concentrate mismatch near the boundary-adjacent strip, motivating tapered and geometry-aware variants. The distortion column is measured in this strip-audit protocol and is therefore not intended to numerically match the global projector-distortion audit in Table 3. Smaller values are better, except for the strip-to-core ratio, where larger values indicate stronger boundary concentration.

Operator regime	Boundary-audit distortion MSE	Boundary-strip div RMS	Core div RMS	Strip-to-core ratio
Cavity Jacobi-10	$(7.167 \pm 9.759) \times 10^{-3}$	0.122 ± 0.125	$(3.400 \pm 3.400) \times 10^{-2}$	$(3.600 \pm 0.500) \times 10^{0}$
Cavity Direct	$(2.700 \pm 5.500) \times 10^{-2}$	0.170 ± 0.209	$(8.139 \pm 9.159) \times 10^{-7}$	$(2.059 \pm 0.238) \times 10^{5}$
Cavity Screened8-10	$(2.352 \pm 3.041) \times 10^{-4}$	0.179 ± 0.106	$(1.190 \pm 0.660) \times 10^{-1}$	$(1.500 \pm 0.200) \times 10^{0}$
Tube Jacobi-20	$(6.098 \pm 5.333) \times 10^{-4}$	0.123 ± 0.080	$(1.600 \pm 1.100) \times 10^{-2}$	$(1.190 \pm 0.810) \times 10^{1}$
Tube Direct	$(1.400 \pm 2.000) \times 10^{-2}$	0.210 ± 0.105	$(1.633 \pm 1.407) \times 10^{-6}$	$(1.848 \pm 0.867) \times 10^{5}$
Tube Screened8-20	$(1.207 \pm 0.801) \times 10^{-5}$	0.145 ± 0.088	$(2.600 \pm 1.600) \times 10^{-2}$	$(7.700 \pm 5.700) \times 10^{0}$
Dam Jacobi-20	$(2.154 \pm 0.644) \times 10^{-3}$	0.092 ± 0.010	$(2.300 \pm 0.600) \times 10^{-2}$	$(4.100 \pm 0.700) \times 10^{0}$
Dam Direct	$(1.000 \pm 0.700) \times 10^{-2}$	0.178 ± 0.027	$(8.473 \pm 3.723) \times 10^{-7}$	$(2.282 \pm 0.483) \times 10^{5}$
Dam Screened8-20	$(2.768 \pm 0.790) \times 10^{-5}$	0.069 ± 0.013	$(5.900 \pm 0.900) \times 10^{-2}$	$(1.200 \pm 0.200) \times 10^{0}$
Cylinder Jacobi-20	$(1.600 \pm 2.600) \times 10^{-2}$	0.365 ± 0.250	$(7.100 \pm 3.300) \times 10^{-2}$	$(5.000 \pm 1.600) \times 10^{0}$
Cylinder Direct	$(1.000 \pm 1.320) \times 10^{-1}$	0.548 ± 0.309	$(2.086 \pm 0.962) \times 10^{-6}$	$(2.710 \pm 0.860) \times 10^{5}$
Cylinder Screened8-20	$(1.319 \pm 2.676) \times 10^{-4}$	0.335 ± 0.270	$(1.340 \pm 0.750) \times 10^{-1}$	$(2.400 \pm 0.600) \times 10^{0}$

D.4 Geometry-Aware Evaluator-Side Variants

Boundary taper. The boundary-strip diagnostics in Table 9 suggest a third, explicitly geometry-aware evaluator-side variant. Let $\mathcal{P}_\lambda$ denote the validation-selected screened cleanup, and let $d(i,j)$ be the distance from cell (i,j) to the domain boundary in grid cells. We define a boundary-taper mask

$$M_w(i,j) = \mathrm{clip}((d(i,j)-1)/w, 0, 1), \text{ for } w > 0, \tag{76}$$

and apply the tapered cleanup

$$\hat{\boldsymbol{u}}_{\mathrm{taper}} = \tilde{\boldsymbol{u}} + M_w \odot (\mathcal{P}_\lambda(\tilde{\boldsymbol{u}}) - \tilde{\boldsymbol{u}}). \tag{77}$$

In the main tapered evaluation, we use the validation-selected screened shift together with $w = 1$, so the first interior ring is left uncorrected, and deeper interior cells retain the screened update. This intentionally weak, evaluation-only taper tests whether the measured boundary-strip mismatch can be turned into a practical accuracy gain on the hardest non-periodic cases.

GeoScreened family. The same diagnosis suggests a stronger geometry-aware family that weakens the cleanup within the solve instead of only after the solve. Using the same distance mask, we define a spatially varying screened shift

$$\lambda_w(i,j) = \lambda_{\mathrm{bdry}} + (\lambda_{\mathrm{core}} - \lambda_{\mathrm{bdry}}) M_w(i,j), \text{ where } \lambda_{\mathrm{bdry}} \geq \lambda_{\mathrm{core}} \geq 0, \tag{78}$$

and solve

$$\begin{aligned} (A + \Lambda_w)p &= -\nabla \cdot_h \tilde{\boldsymbol{u}}, \\ (\Lambda_w p)_{i,j} &= \lambda_w(i,j) p_{i,j}. \end{aligned} \tag{79}$$

We refer to the resulting cleanup as the GeoScreened family with boundary shift λ_{bdry}, core shift λ_{core}, and taper width w. When $\lambda_{\mathrm{bdry}} = \lambda_{\mathrm{core}}$, Equation 79 reduces to the global screened family. We select these three parameters on validation rollouts and then optionally compose the selected GeoScreened operator with the same two-cell post hoc taper.

D.5 Qualitative Mechanism Diagnostics

Figure 4 shows the same residual and distortion failure mode on an actual held-out cylinder wake snapshot. The milder screened family leaves more divergence but preserves the recirculation structure and stays near the raw error level, whereas Jacobi and Direct reduce divergence more aggressively while overcorrecting the wake and spreading much larger velocity error.

Figures 5 and 6 turn the aggregate pattern into case-level diagnostics. The response atlas shows that the approximate-regime split appears in the direction each real trajectory moves in the error–divergence plane, while the paired cylinder comparison shows that the same problem family can flip from cases where Raw is best to cases where mild cleanup is best when the evaluation manifold changes.

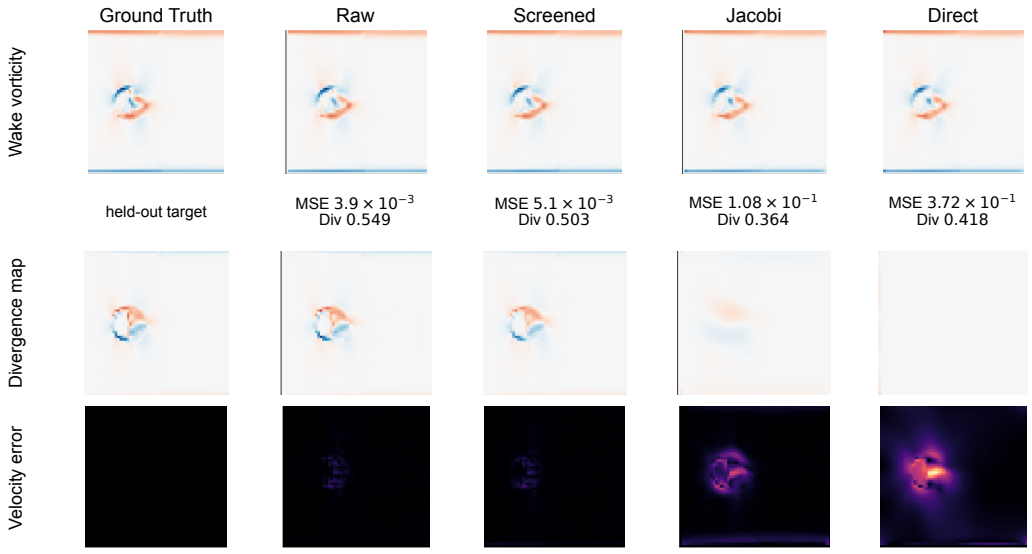


Figure 4: Spatial solver-family mechanism board for a held-out CFDBench cylinder wake snapshot, organized to make the stronger-solver negative control visible. Columns compare Raw, Screened, Jacobi, and Direct against the ground truth; rows show wake vorticity, divergence, and velocity error. The useful qualitative contrast is not only between Raw and Direct. Screened leaves substantially more divergence than Jacobi or Direct, yet it preserves the recirculation structure and stays near the raw error level. Jacobi and Direct both reduce divergence relative to Raw, but both visibly overcorrect the wake and spread much larger error. The solver-family board is the spatial companion to Proposition 5. On the approximate cylinder row, solving the Dirichlet family more faithfully does not recover the right operating point, whereas the milder screened family stays closer to the benchmark manifold.

D.6 Constructive Rule in the Approximate Regime

Validation protocol. Section 3.6 first estimates correction alignment and analytic blend strength on shared teacher-forced validation pairs. Before selection, audit the operator on labeled held-out targets in physical units using target distortion $\mathrm{MSE}(T(y), y)$ and constraint violation, and examine its amplification. These offline audits distinguish a well-solved system from a target-preserving operator. A favorable one-step score supplies a candidate prediction; uncertain confidence margins or uncontrolled amplification trigger the rollout-validation fallback below.

Raw should remain an explicit deployment candidate, and the comparison should include a strong-solver anchor such as Direct. Raw tests whether cleanup is needed at all, while Direct tests whether solving the nominal constraint harder is actually beneficial. When exact-on-target operators are available, they should be benchmarked before approximate alternatives. When only approximate cleanup is available, the validation menu should span a small family of strengths, including mild or screened operators, blended operators, and adaptive gates.

For the fallback, tune cleanup hyperparameters on validation rollouts at the intended deployment horizon and metric, rather than on divergence or Poisson residual. Report the selected rule on disjoint test trajectories together with Raw and strong-cleanup anchors, including both error and violation. The retrospective deployment table uses final-horizon MSE; the prospective one-step audits explicitly report their AUC windows. Neither offline target distortion nor either validation stage requires deployed ground truth, and test labels never enter the prospective selection.

From-scratch U-Net validation. We trained 12 independent 7.50M-parameter U-Nets from random initialization on the four official subsets (three seeds each; 80/80/100/100 epochs for Cavity/Tube/Dam/Cylinder). No FNO weights or features were used. Checkpoints were frozen only for paired operator evaluation. The screened and GeoScreened families are the best post hoc operating points in our study because they weaken cleanup in a controlled way. This operating-point advantage

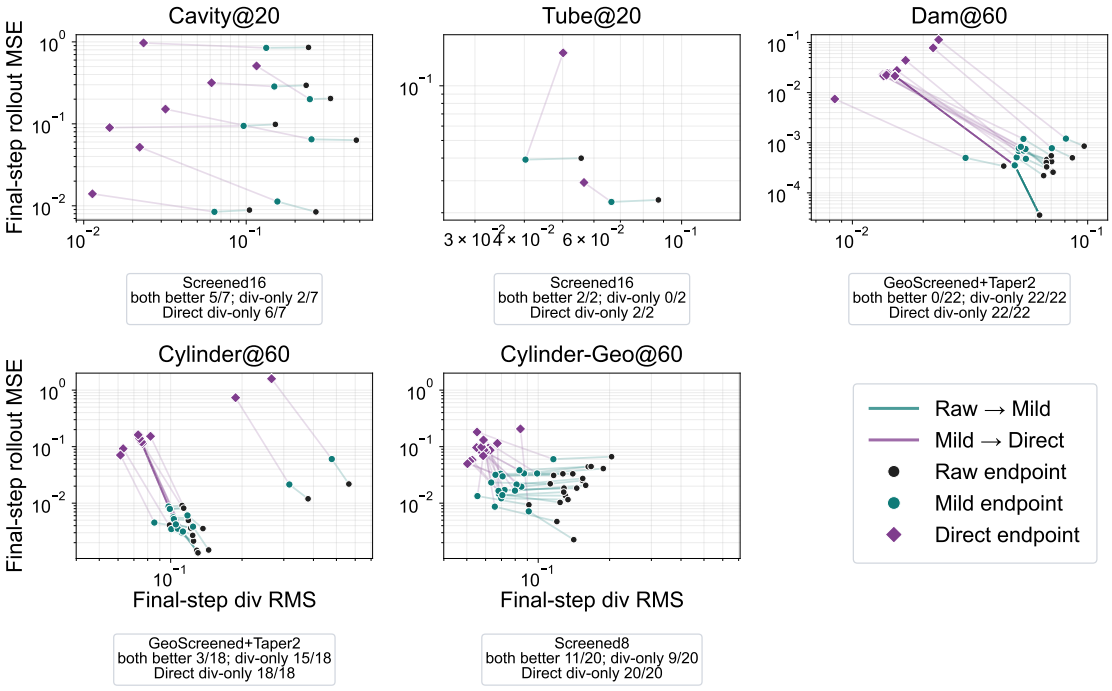


Figure 5: Case-level response atlas in the final-step error–divergence plane, averaged over seeds 42, 123, and 456. Each polyline is one actual trajectory traced from its Raw endpoint to the dataset-tuned mild operator and then to Direct. The callouts keep the count breakdown compact. Cases improving both metrics have lower final-step MSE and lower divergence relative to Raw, while divergence-only improvements reduce divergence but increase MSE. The aggregate pattern is the point. Tube@20 moves to the corner where both metrics improve under mild cleanup, official-split Dam@60 and Cylinder@60 mostly show divergence-only improvements, and Cylinder-Geo@60 flips back toward improvements in both metrics before Direct turns all cases upward in error. This figure therefore makes our main approximate-regime claim visible on a full set of real cases.

is visible in the external U-Net validation results in Table 10. On cavity and tube, mild screened cleanup stays on or near the raw frontier, while Jacobi and Direct overclean. Table 10 shows that, on Dam@60 and Cylinder@60, GeoScreened+Taper2 becomes the strongest cleanup-only point, reaching $(6.570 \pm 0.850) \times 10^{-4}$ and $(3.800 \pm 0.270) \times 10^{-3}$, respectively, far below Jacobi and Direct.

The fairness objection is still important because approximate-regime post hoc gains could reflect a train–test mismatch. The tuned cylinder rows in Table 19 show that this concern is only partially valid. On the short-horizon tuned cylinder row, matched projected screened training can indeed beat Raw-FNO, reaching $(1.409 \pm 0.028) \times 10^{-3}$ compared with $(1.642 \pm 0.863) \times 10^{-3}$. However, this tuned-cylinder win is a bounded exception, not a reversal of the main claim. On the harder long-horizon rows for dam and cylinder, the best operating point still stays with raw or very mild post hoc cleanup. Our approximate-regime recommendation is therefore specific. Compare operator families in velocity space, validate cleanup strength explicitly, and expect the winning operator to be much milder than the strongest physically motivated solve.

A final protocol check concerns coordinate-frame choice. Because the non-periodic cleanup operator is implemented in physical velocity units, one could worry that the approximate-regime ordering is really a coordinate-frame artifact. A dedicated normalized and physical frame ablation on post hoc Jacobi and Screened8 shows otherwise. Across cavity, tube, dam, and cylinder, the largest relative change in final-horizon rollout MSE is only 0.14% on tube under Screened8, and most rows move by less than 0.06%. We therefore use physical-space cleanup for conceptual consistency, but the main regime split between exact and approximate operators is not driven by this implementation choice.

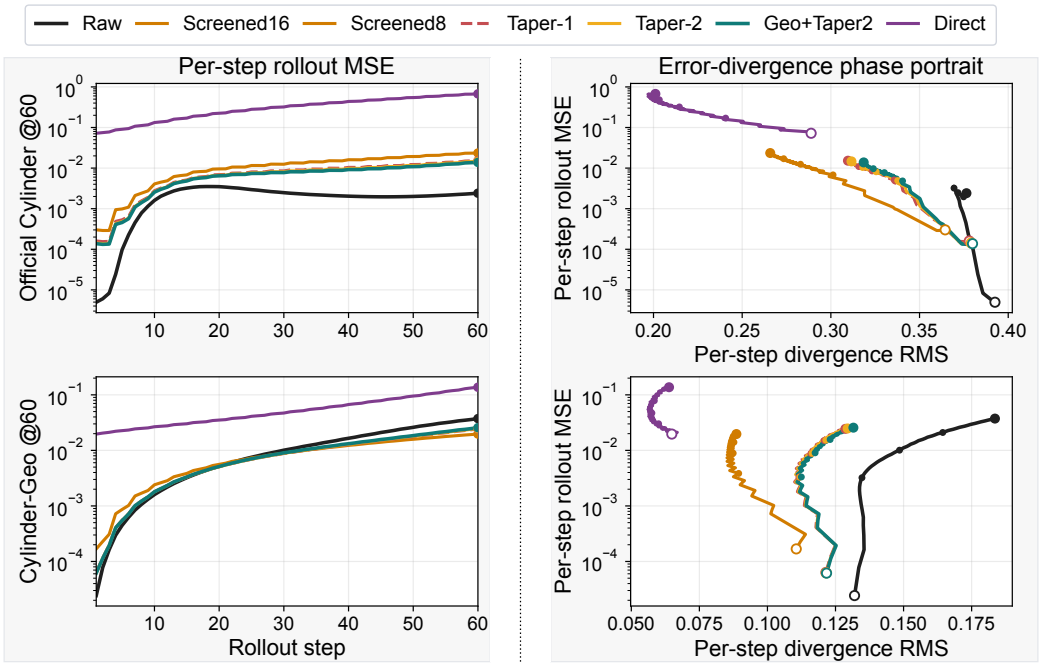


Figure 6: Within-benchmark regime comparison on cylinder flow from the shared seed-42 artifacts. Top row: official-split Cylinder@60 trajectory 1 with the screened curve labeled Screened16. Bottom row: geometry-holdout Cylinder-Geo@60 trajectory 17 with the screened curve labeled Screened8. Left: per-step rollout MSE. Right: the same trajectories traced through the error–divergence plane. In the official-split case, stronger cleanup moves the rollout toward lower divergence but sharply higher error. The Screened16, Taper2, GeoScreened+Taper2, and Direct endpoints all sit above Raw in final-step MSE. On the geometry-holdout case, the ordering flips. Screened8 improves over Raw, Taper2 and GeoScreened+Taper2 remain nearby, and Direct still overcleans. This paired figure turns our main claim into a within-benchmark qualitative comparison. Cleanup strength alone does not determine success; operator alignment with the evaluation manifold does.

E Hierarchical Forecasting Support

E.1 Task Setup

As a compact real-data support study outside CFD, we consider autoregressive hierarchical forecasting and test whether the same exact-versus-approximate operator split appears beyond fluid benchmarks. The hierarchical forecasting setup connects our operator-exactness theme to classical, probabilistic, and end-to-end reconciliation settings that are already well-studied outside CFD [41, 31, 34, 3]. Let $x_t \in \mathbb{R}^m$ denote the vector of all hierarchy nodes at time t, and let $S \in \{0,1\}^{m\times b}$ be the summing matrix that maps bottom-level series $z_t \in \mathbb{R}^b$ to the full hierarchy:

$$\begin{aligned} x_t &= Sz_t, \\ \mathcal{C} &= \{Sz : z \in \mathbb{R}^b\}. \end{aligned} \tag{80}$$

The coherent target manifold is therefore the column space $\mathcal{C}$ of S. We train a strong autoregressive gated recurrent unit model following Cho et al. [9] to forecast all hierarchy levels jointly on the public Traffic, Labour, TourismLarge, OldTourismLarge, and Wiki2 benchmarks as packaged in the HierarchicalForecast collection introduced by Olivares et al. [30], then compare three operator families on the predicted states.

The exact OLS reconciliation is the orthogonal projector

$$\Pi_{\text{ols}} = S(S^\top S)^{-1}S^\top. \tag{81}$$

We also include the exact-but-nonorthogonal bottom-up repair

$$\Pi_{\text{bu}} = SG_{\text{bu}}, \tag{82}$$

Table 10: From-scratch per-dataset U-Net validation on the official CFDBench suite; each trained checkpoint is frozen only for the paired cleanup sweep. Each cell reports final-horizon rollout MSE on the first line and final-horizon rollout Div on the second line. The upper block reports raw, mild screened cleanup, and GeoScreened cleanup, while the lower block reports stronger-cleanup anchors. All non-raw variants are post hoc cleanup operators.

Raw and mild cleanup variants

Dataset	Raw	Screened16	Screened16+Taper2	GeoScreened	GeoScreened+Taper2
Official suite					
	$(2.910 \pm 0.370) \times 10^{-1}$	$(2.910 \pm 0.370) \times 10^{-1}$	$(2.910 \pm 0.370) \times 10^{-1}$	$(2.910 \pm 0.370) \times 10^{-1}$	$(2.910 \pm 0.370) \times 10^{-1}$
Cavity	$(2.010 \pm 0.076) \times 10^{-1}$	$(1.240 \pm 0.032) \times 10^{-1}$	$(1.400 \pm 0.039) \times 10^{-1}$	$(1.270 \pm 0.032) \times 10^{-1}$	$(1.410 \pm 0.040) \times 10^{-1}$
	$(2.100 \pm 0.360) \times 10^{-2}$	$(2.000 \pm 0.370) \times 10^{-2}$	$(2.000 \pm 0.360) \times 10^{-2}$	$(2.000 \pm 0.360) \times 10^{-2}$	$(2.000 \pm 0.360) \times 10^{-2}$
Tube	$(6.200 \pm 0.220) \times 10^{-2}$	$(5.100 \pm 0.230) \times 10^{-2}$	$(5.900 \pm 0.200) \times 10^{-2}$	$(5.300 \pm 0.220) \times 10^{-2}$	$(6.000 \pm 0.190) \times 10^{-2}$
	$(3.330 \pm 0.200) \times 10^{-4}$	$(4.280 \pm 0.270) \times 10^{-4}$	$(4.150 \pm 0.280) \times 10^{-4}$	$(4.190 \pm 0.280) \times 10^{-4}$	$(4.150 \pm 0.280) \times 10^{-4}$
Dam	$(6.700 \pm 0.042) \times 10^{-2}$	$(5.200 \pm 0.067) \times 10^{-2}$	$(5.400 \pm 0.074) \times 10^{-2}$	$(5.300 \pm 0.070) \times 10^{-2}$	$(5.500 \pm 0.074) \times 10^{-2}$
	$(2.260 \pm 0.180) \times 10^{-3}$	$(3.040 \pm 0.370) \times 10^{-3}$	$(2.340 \pm 0.390) \times 10^{-3}$	$(2.410 \pm 0.400) \times 10^{-3}$	$(2.310 \pm 0.400) \times 10^{-3}$
Cylinder	$(1.720 \pm 0.009) \times 10^{-1}$	$(1.320 \pm 0.002) \times 10^{-1}$	$(1.500 \pm 0.006) \times 10^{-1}$	$(1.440 \pm 0.005) \times 10^{-1}$	$(1.520 \pm 0.006) \times 10^{-1}$
Long-horizon stress rows					
	$(2.590 \pm 0.320) \times 10^{-4}$	$(7.180 \pm 0.970) \times 10^{-4}$	$(6.660 \pm 0.880) \times 10^{-4}$	$(6.840 \pm 0.890) \times 10^{-4}$	$(6.570 \pm 0.850) \times 10^{-4}$
Dam@60	$(6.600 \pm 0.075) \times 10^{-2}$	$(4.200 \pm 0.076) \times 10^{-2}$	$(4.500 \pm 0.090) \times 10^{-2}$	$(4.400 \pm 0.081) \times 10^{-2}$	$(4.600 \pm 0.086) \times 10^{-2}$
	$(2.620 \pm 0.081) \times 10^{-3}$	$(6.140 \pm 0.090) \times 10^{-3}$	$(4.060 \pm 0.220) \times 10^{-3}$	$(4.200 \pm 0.250) \times 10^{-3}$	$(3.800 \pm 0.270) \times 10^{-3}$
Cylinder@60	$(1.720 \pm 0.014) \times 10^{-1}$	$(1.170 \pm 0.006) \times 10^{-1}$	$(1.400 \pm 0.008) \times 10^{-1}$	$(1.310 \pm 0.003) \times 10^{-1}$	$(1.440 \pm 0.009) \times 10^{-1}$

Strong-cleanup anchors

Dataset	Raw	Jacobi	Direct
Official suite			
	$(2.910 \pm 0.370) \times 10^{-1}$	$(3.190 \pm 0.380) \times 10^{-1}$	$(3.420 \pm 0.400) \times 10^{-1}$
Cavity	$(2.010 \pm 0.076) \times 10^{-1}$	$(3.800 \pm 0.380) \times 10^{-2}$	$(3.700 \pm 0.710) \times 10^{-2}$
	$(2.100 \pm 0.360) \times 10^{-2}$	$(4.000 \pm 0.730) \times 10^{-2}$	$(1.070 \pm 0.190) \times 10^{-1}$
Tube	$(6.200 \pm 0.220) \times 10^{-2}$	$(2.400 \pm 0.350) \times 10^{-2}$	$(4.900 \pm 0.600) \times 10^{-2}$
	$(3.330 \pm 0.200) \times 10^{-4}$	$(7.460 \pm 0.150) \times 10^{-3}$	$(1.400 \pm 0.025) \times 10^{-2}$
Dam	$(6.700 \pm 0.042) \times 10^{-2}$	$(2.000 \pm 0.016) \times 10^{-2}$	$(2.600 \pm 0.018) \times 10^{-2}$
	$(2.260 \pm 0.180) \times 10^{-3}$	$(5.600 \pm 0.210) \times 10^{-2}$	$(7.500 \pm 0.180) \times 10^{-2}$
Cylinder	$(1.720 \pm 0.009) \times 10^{-1}$	$(7.000 \pm 0.059) \times 10^{-2}$	$(1.130 \pm 0.044) \times 10^{-1}$
Long-horizon stress rows			
	$(2.590 \pm 0.320) \times 10^{-4}$	$(1.200 \pm 0.064) \times 10^{-2}$	$(2.000 \pm 0.100) \times 10^{-2}$
Dam@60	$(6.600 \pm 0.075) \times 10^{-2}$	$(1.100 \pm 0.004) \times 10^{-2}$	$(1.300 \pm 0.098) \times 10^{-2}$
	$(2.620 \pm 0.081) \times 10^{-3}$	$(1.710 \pm 0.210) \times 10^{-1}$	$(2.200 \pm 0.018) \times 10^{-1}$
Cylinder@60	$(1.720 \pm 0.014) \times 10^{-1}$	$(5.100 \pm 0.270) \times 10^{-2}$	$(9.300 \pm 1.100) \times 10^{-2}$

where G_{bu} extracts the model's bottom-level forecasts from the full hierarchy vector.

The approximate comparator starts from the full-strength top-down operator built from historical mean proportions $\bar{p} \in \mathbb{R}^b$ with $\mathbf{1}^\top \bar{p} = 1$. Let $e_{\mathrm{root}} \in \mathbb{R}^m$ denote the selector for the root node, so that $x^{\mathrm{root}} = e_{\mathrm{root}}^\top x$. Then

$$\widetilde{\Pi}_{\mathrm{td}}(x) = S\bar{p}e_{\mathrm{root}}^\top x. \tag{83}$$

We also evaluate a blended variant,

$$\widetilde{\Pi}_{\mathrm{td},\alpha}(x) = x + \alpha(\widetilde{\Pi}_{\mathrm{td}}(x) - x), \tag{84}$$

with α selected on the validation split.

Proposition 13 (Exactness of hierarchical reconciliation operators)**.** *Assume that S has full column rank, that $G_{\mathrm{bu}}S = I_b$, and that the root row satisfies $e_{\mathrm{root}}^\top S = \mathbf{1}^\top$. Then*

$$\Pi_{\mathrm{ols}}x = x \text{ and } \Pi_{\mathrm{bu}}x = x \text{ for every } x \in \mathcal{C}. \tag{85}$$

The full-strength top-down operator $\widetilde{\Pi}_{\mathrm{td}}$ maps every input into $\mathcal{C}$, but, for a coherent point $x = Sz$, it satisfies $\widetilde{\Pi}_{\mathrm{td}}(x) = x$ if and only if $z = (\mathbf{1}^\top z)\bar{p}$. Thus, top-down reconciliation is generally not exact on $\mathcal{C}$ and can distort coherent targets.

Proof. Let $x = Sz \in \mathcal{C}$. For OLS reconciliation,

$$\Pi_{\mathrm{ols}}x = S(S^\top S)^{-1}S^\top Sz = Sz = x, \tag{86}$$

where full column rank makes $S^\top S$ invertible. For bottom-up reconciliation,

$$\Pi_{\mathrm{bu}}x = SG_{\mathrm{bu}}Sz = Sz = x, \tag{87}$$

using $G_{\mathrm{bu}} S = I_b$.

For any input x, $\widetilde{\Pi}_{\mathrm{td}}(x) = S(\bar{p} e_{\mathrm{root}}^{\top} x)$ lies in $\mathrm{range}(S) = \mathcal{C}$, so the full-strength top-down output is coherent. If $x = Sz$ is already coherent, then

$$\widetilde{\Pi}_{\mathrm{td}}(Sz) = S\bar{p} e_{\mathrm{root}}^{\top} Sz = S\bar{p}\mathbf{1}^{\top} z. \tag{88}$$

Because S has full column rank, the top-down expression equals Sz exactly when $z = (\mathbf{1}^{\top} z)\bar{p}$. Coherent targets whose bottom-level proportions differ from $\bar{p}$ are therefore moved by $\widetilde{\Pi}_{\mathrm{td}}$. □

These are operator-level exactness statements with respect to the abstract coherent manifold $\mathcal{C}$; the diagnostics reported below are therefore empirical benchmark measurements rather than literal numerical identity checks in the periodic-FFT sense. For general inputs x and $\alpha < 1$, the blended top-down operator no longer guarantees coherence, so it is a validation-tuned interpolation toward top-down rather than an exact repair. All hierarchical operators are applied in original value units and only then renormalized for the model, mirroring the physical-space protocol used in the non-periodic CFD study.

E.2 Results on Real Data

Hierarchical forecasting provides a real-data testbed outside fluid forecasting. Table 11 evaluates five public hierarchical forecasting benchmarks. A strong autoregressive GRU predicts all hierarchy levels jointly, and we compare operator-level exact coherent reconciliation given by OLS and bottom-up with a validation-tuned blended top-down family. Proposition 13 gives the exactness proof. For coherent targets $x \in \mathcal{C}$, OLS and bottom-up satisfy $\Pi(x) = x$. The small nonzero OLS and Bottom-Up distortion and coherence values reported in Table 11 are coordinate-transform and finite-precision effects of denormalization and renormalization in the evaluation pipeline rather than counterexamples to exactness. The experiment therefore evaluates benchmark-aligned linear repairs, with the exact families serving as the mathematically aligned baseline and the top-down family as the intentionally inexact comparator.

Table 11 shows that the hierarchical evidence does not reduce to a universal ordering. Across Traffic, Labour, TourismLarge, OldTourismLarge, and Wiki2, exact reconciliation is the most stable family. Bottom-Up or Proj-OLS gives the best exact baseline on every dataset and avoids the catastrophic failures seen for approximate top-down repair. The tuned top-down family is highly dataset-dependent. It worsens Traffic and is catastrophic on Labour, where rollout MSE rises to $(3.740 \pm 0.288) \times 10^3$, but it becomes decisively best on Wiki2, reducing rollout MSE from Raw-GRU's $(3.170 \pm 0.251) \times 10^8$ to $(1.610 \pm 0.112) \times 10^8$. TourismLarge and OldTourismLarge lie between these extremes. Bottom-Up remains best, while tuned top-down improves over Raw-GRU but stays worse than the best exact operator. Thus, the support study reinforces the main claim. Exact benchmark-aligned operators are the mandatory low-risk baseline, while approximate operators must be validated dataset by dataset.

This heterogeneity makes the broader point sharper. Exact coherent reconciliation is the safest cross-dataset baseline. It never catastrophically fails across the five benchmarks, and exact Bottom-Up or Proj-OLS is the best exact baseline on all five of them. The blended top-down family has both the highest downside and, on Wiki2, the highest upside. What transfers is therefore not a blanket claim that top-down is always harmful but the more operational lesson that benchmark-aligned exact operators should be the mandatory baseline, while approximate operators must be treated as a separate family whose strength and even sign of benefit are dataset-dependent. In-loop OLS training remains mixed rather than uniformly dominant, matching our main framing. The durable lesson is about baseline quality and operator exactness, not about a universal rule that training with the operator in the loop is always best outside the clean periodic setting.

F Additional Experiments and Statistical Analyses

F.1 Matched-Checkpoint Projection Toggles

A separate five-seed NS-128 replication uses the submitted four-layer, width-64, modes-24, 18.9M-parameter FNO recipe, 18 epochs, and 100/20/20 training/validation/test trajectories. Seeds are 42, 123, 456, 789, and 1011; every checkpoint is evaluated on the same ten 100-step test trajectories with

Table 11: Real-data hierarchical forecasting study on five public hierarchies. The OLS and Bottom-Up reconciliation family is the lowest-variance baseline across datasets, while the tuned blended top-down family ranges from harmful to dominant depending on the benchmark. Smaller values are better in every numeric column.

Dataset	Variant	Rollout MSE	Last-step MSE	Constraint RMS	Target distortion
Traffic	Raw-GRU	$(3.663 \pm 0.374) \times 10^{2}$	$(4.588 \pm 0.754) \times 10^{2}$	$(3.695 \pm 0.576) \times 10^{0}$	0
	PostHoc-OLS	$(3.216 \pm 0.373) \times 10^{2}$	$(4.097 \pm 1.099) \times 10^{2}$	$(2.000 \pm 0.013) \times 10^{-2}$	$(4.450 \pm 0.000) \times 10^{-4}$
	PostHoc-Bottom-Up	$(3.040 \pm 0.311) \times 10^{2}$	$(3.696 \pm 0.878) \times 10^{2}$	$(1.800 \pm 0.007) \times 10^{-2}$	$(9.250 \pm 0.000) \times 10^{-6}$
	PostHoc-Top-Down	$(4.037 \pm 0.648) \times 10^{2}$	$(7.321 \pm 1.770) \times 10^{2}$	$(1.990 \pm 0.161) \times 10^{0}$	$(2.240 \pm 0.000) \times 10^{-1}$
	Proj-OLS	$(3.035 \pm 0.481) \times 10^{2}$	$(4.010 \pm 1.524) \times 10^{2}$	$(2.000 \pm 0.011) \times 10^{-2}$	$(4.450 \pm 0.000) \times 10^{-4}$
Labour	Raw-GRU	$(9.343 \pm 2.164) \times 10^{2}$	$(1.830 \pm 0.554) \times 10^{3}$	$(8.772 \pm 1.488) \times 10^{0}$	0
	PostHoc-OLS	$(8.157 \pm 2.223) \times 10^{2}$	$(1.510 \pm 0.558) \times 10^{3}$	$(3.180 \pm 0.022) \times 10^{-4}$	$(4.560 \pm 0.000) \times 10^{-8}$
	PostHoc-Bottom-Up	$(7.657 \pm 1.475) \times 10^{2}$	$(1.390 \pm 0.307) \times 10^{3}$	$(2.050 \pm 0.032) \times 10^{-4}$	$(7.800 \pm 0.000) \times 10^{-9}$
	PostHoc-Top-Down	$(3.740 \pm 0.288) \times 10^{3}$	$(8.380 \pm 0.951) \times 10^{3}$	$(8.360 \pm 1.855) \times 10^{0}$	$(1.921 \pm 0.000) \times 10^{2}$
	Proj-OLS	$(6.007 \pm 0.696) \times 10^{2}$	$(9.824 \pm 1.099) \times 10^{2}$	$(3.200 \pm 0.030) \times 10^{-4}$	$(4.560 \pm 0.000) \times 10^{-8}$
TourismLarge	Raw-GRU	$(1.890 \pm 0.111) \times 10^{5}$	$(2.270 \pm 0.291) \times 10^{5}$	$(1.115 \pm 0.056) \times 10^{2}$	0
	PostHoc-OLS	$(2.230 \pm 0.086) \times 10^{5}$	$(2.830 \pm 0.411) \times 10^{5}$	$(2.360 \pm 0.038) \times 10^{-1}$	$(7.800 \pm 0.000) \times 10^{-2}$
	PostHoc-Bottom-Up	$(1.550 \pm 0.068) \times 10^{5}$	$(1.660 \pm 0.261) \times 10^{5}$	$(2.610 \pm 0.033) \times 10^{-1}$	$(6.600 \pm 0.000) \times 10^{-3}$
	PostHoc-Top-Down	$(1.710 \pm 0.151) \times 10^{5}$	$(1.210 \pm 0.298) \times 10^{5}$	$(8.402 \pm 0.388) \times 10^{1}$	$(1.084 \pm 0.000) \times 10^{2}$
	Proj-OLS	$(1.620 \pm 0.060) \times 10^{5}$	$(1.360 \pm 0.077) \times 10^{5}$	$(2.350 \pm 0.006) \times 10^{-1}$	$(7.800 \pm 0.000) \times 10^{-2}$
OldTourismLarge	Raw-GRU	$(1.800 \pm 0.086) \times 10^{5}$	$(2.220 \pm 0.368) \times 10^{5}$	$(1.115 \pm 0.049) \times 10^{2}$	0
	PostHoc-OLS	$(2.170 \pm 0.115) \times 10^{5}$	$(2.920 \pm 0.488) \times 10^{5}$	$(2.370 \pm 0.042) \times 10^{-1}$	$(7.800 \pm 0.000) \times 10^{-2}$
	PostHoc-Bottom-Up	$(1.470 \pm 0.090) \times 10^{5}$	$(1.570 \pm 0.335) \times 10^{5}$	$(2.640 \pm 0.046) \times 10^{-1}$	$(6.600 \pm 0.000) \times 10^{-3}$
	PostHoc-Top-Down	$(1.640 \pm 0.094) \times 10^{5}$	$(1.080 \pm 0.271) \times 10^{5}$	$(8.352 \pm 0.320) \times 10^{1}$	$(1.084 \pm 0.000) \times 10^{2}$
	Proj-OLS	$(1.590 \pm 0.032) \times 10^{5}$	$(1.270 \pm 0.100) \times 10^{5}$	$(2.310 \pm 0.029) \times 10^{-1}$	$(7.800 \pm 0.000) \times 10^{-2}$
Wiki2	Raw-GRU	$(3.170 \pm 0.251) \times 10^{8}$	$(4.320 \pm 0.599) \times 10^{8}$	$(3.950 \pm 0.479) \times 10^{3}$	0
	PostHoc-OLS	$(3.030 \pm 0.200) \times 10^{8}$	$(4.090 \pm 0.516) \times 10^{8}$	$(2.604 \pm 0.220) \times 10^{0}$	$(1.329 \pm 0.000) \times 10^{1}$
	PostHoc-Bottom-Up	$(3.130 \pm 0.201) \times 10^{8}$	$(4.190 \pm 0.488) \times 10^{8}$	$(2.404 \pm 0.251) \times 10^{0}$	$(1.601 \pm 0.000) \times 10^{1}$
	PostHoc-Top-Down	$(1.610 \pm 0.112) \times 10^{8}$	$(1.670 \pm 0.224) \times 10^{8}$	$(3.199 \pm 5.985) \times 10^{1}$	$(9.330 \pm 1.790) \times 10^{7}$
	Proj-OLS	$(2.770 \pm 0.031) \times 10^{8}$	$(3.270 \pm 0.039) \times 10^{8}$	$(2.885 \pm 0.163) \times 10^{0}$	$(1.329 \pm 0.000) \times 10^{1}$

projection OFF or ON. Table 12 reports run-wise mean $\pm$ sample SD. It is a replication of the recipe, not a replacement for Table 2.

Table 12: Training/evaluation projection factorial. Within each training condition, ON/OFF holds weights and trajectories fixed. Proj-trained/OFF is a deliberate out-of-distribution intervention, not an ordinary deployment condition.

Training / evaluation	MSE-AUC	MSE@100	Div@100
Raw / OFF	$(6.720 \pm 3.702) \times 10^{-5}$	$(1.536 \pm 0.9353) \times 10^{-4}$	$(3.735 \pm 1.126) \times 10^{-2}$
Raw / ON (PostHoc)	$(1.299 \pm 0.3930) \times 10^{-6}$	$(2.765 \pm 1.011) \times 10^{-6}$	$(4.948 \pm 0.1115) \times 10^{-6}$
Proj / OFF (OOD probe)	$(2.908 \pm 0.3630) \times 10^{-3}$	$(7.656 \pm 0.9301) \times 10^{-3}$	$(1.982 \pm 0.1173) \times 10^{-1}$
Proj / ON	$(5.666 \pm 1.692) \times 10^{-7}$	$(8.455 \pm 2.102) \times 10^{-7}$	$(1.894 \pm 0.1140) \times 10^{-5}$

Using one paired difference per training seed, Raw OFF-minus-ON AUC is 6.590×10^{-5}, with Student-t 95% CI $[1.999 \times 10^{-5}, 1.118 \times 10^{-4}]$; the MSE@100 difference is 1.508×10^{-4}, CI $[3.48 \times 10^{-5}, 2.67 \times 10^{-4}]$. For Proj-trained weights, OFF-minus-ON AUC is 2.907×10^{-3}, CI $[2.457, 3.358] \times 10^{-3}$, a 99.98% reduction. Projected training need not identify output components discarded by the projection; removing the layer exposes those components. Comparing the two ON rows quantifies the additional training-alignment gain: 7.325×10^{-7}, CI $[3.194 \times 10^{-7}, 1.146 \times 10^{-6}]$. The within-checkpoint improvements agree in all five seeds.

Native-layer verification. Proj-FNO applies the differentiable Fourier projector to every accumulated predicted state inside the forward map. Across the five retained Proj checkpoints, each internal predicted step has divergence at most 5.157×10^{-6}; reapplying the projector changes its output by at most 1.485×10^{-6} relatively. Removing the layer changes the training gradient by relative ℓ_2 distance at least 13.4, with cosine similarity from -0.64 to -0.11. Native placement therefore affects optimization as well as feedback.

F.2 Measured Operator-Error Propagation

For $Q = I - P$, let R isometrically rotate retained compressible Fourier modes into the solenoidal range. On the retained real Fourier space (excluding self-conjugate Nyquist modes),

$$T_\eta = P + \eta RQ, \qquad T_\eta^2 = T_\eta, \qquad \|T_\eta - P\|_{\mathrm{op}} = \eta, \quad \|T_\eta\|_{\mathrm{op}} = \sqrt{1 + \eta^2}.$$

Every T_η fixes valid targets and has the same divergence-free range, but only $\eta = 0$ is orthogonal. In contrast, $B_\delta = (1-\delta)P$ is nonexpansive and divergence-free but distorts targets. A five-seed matched-checkpoint experiment measures the full perturbation trajectory against the P rollout (Table 13). The RMS distance is $d_t/\sqrt{m}$; the theoretical bounds use unnormalized d_t.

Table 13: Controlled operator changes and measured rollout perturbations. Means $\pm$ sample SD over five training seeds; all repairs have divergence approximately 3×10^{-6}.

Map	$\|T - P\|_{\mathrm{op}}$	$\|T\|_{\mathrm{op}}$	RMS d_{100}	MSE-AUC
P	0	1.000	0	$(1.066 \pm 0.3533) \times 10^{-6}$
$T_{.25}$	.25	1.031	$(1.862 \pm 0.698) \times 10^{-3}$	$(2.983 \pm 1.509) \times 10^{-6}$
$T_{.5}$	.5	1.118	$(3.726 \pm 1.40) \times 10^{-3}$	$(9.270 \pm 5.886) \times 10^{-6}$
T_1	1	1.414	$(7.472 \pm 2.81) \times 10^{-3}$	$(3.506 \pm 2.411) \times 10^{-5}$
T_2	2	2.236	$(1.505 \pm 0.569) \times 10^{-2}$	$(1.404 \pm 0.9893) \times 10^{-4}$
$B_{.0025}$	.0025	.998	$(2.442 \pm 0.00107) \times 10^{-2}$	$(2.542 \pm 0.04624) \times 10^{-4}$

For the oblique family, through-origin fits of d_t against η have $R^2 \geq 0.9999$ at horizons 10, 25, 50, and 100 in every seed. Distances are strictly ordered by the known operator norm at every measured step, with an observed growth factor of approximately 1.016 per step. This factor describes the measured perturbation, not a certified value of L_F or KL. T_2 has 132 times the AUC of P despite exact target preservation. The nonexpansive $B_{.0025}$ has target-distortion MSE 1.733×10^{-6} and 238 times the AUC of P. The MSE-difference inequality in Proposition 11 holds at every measured step. Together, the two interventions separate amplification from injected target bias.

F.3 Numerical Norm Audit and Its Coverage

For each fixed linear cleanup, three-start power iteration on T^*T, with up to 500 iterations, estimates the induced ℓ_2 norm on the actual 64×64 and 66×66 grids. Table 14 reports the range across grids and the indicated settings. A computed ratio $\|Tv\|/\|v\| > 1$ is an expansion witness; these estimates are not certified upper bounds.

Table 14: Numerical operator-norm audit. CG and state-dependent AdaBlend are outside the fixed-linear-map audit.

Operator	Estimated $\|T\|_{\mathrm{op}}$
Exact orthogonal FFT	1.0000–1.0000
GeoScreened (two mild variants)	1.0045–1.0144
Screened16	1.0201–1.0204
Screened8	1.0352–1.0352
Jacobi-10 / Jacobi-20	1.1450–1.1505
SOR-10 / SOR-20 / Direct	1.1543–1.1545
MG-2	1.5977–1.6210

Finite-step CG has input-dependent Krylov coefficients, and AdaBlend has a state-dependent gate; neither is assigned a linear-map norm. In Table 6, Raw has norm one and the fixed screened family has numerical audit support, while AdaBlend-GeoScreened is assessed empirically. Auditing its fixed GeoScreened component does not audit the adaptive composition. None of these estimates establishes $L_T L_F < 1$ for a learned forecaster. In particular, with a cleanup gain near 1.6 and unestimated L_F, the quantitative finite-horizon bound at 20–60 steps is not determined. The empirical approximate-regime conclusions do not invoke the conditional steady-state corollaries.

F.4 Dependence-Aware Correlation Statistics

The 12 conditions comprise five Cavity operators, five Tube operators, and two Dam operators. Within each dataset they share three independently trained checkpoints and trajectories; each point is a mean rollout penalty relative to its paired Raw checkpoint. The exact blocked test permutes outcome labels only within dataset: $5!\,5!\,2! = 28{,}800$ permutations. Table 15 reports both conventional point-level summaries and the blocked test, which is the primary dependence-aware inference.

Table 15: Full statistical report for the two principal mechanism predictors. Fisher intervals and parametric tests use the conventional 12-point calculation; blocked tests preserve dataset membership.

Predictor	Pearson r [Fisher CI]	Parametric p	Spearman ρ	Blocked p
GT distortion	.905 [.688, .973]	5.31×10^{-5}	.909	1.39×10^{-4}
Poisson residual	$-.387$ [$-.786$, .240]	.213	$-.427$	.215

After removing dataset offsets, distortion has $r = .844$ (fixed-effect $df = 8$, $p = .00216$); leave-one-dataset-out r ranges from .855 to .961. Computing the 12 penalties separately for each training seed gives distortion $r = .732$–$.935$ and blocked $p = 1.39 \times 10^{-4}$ in all three seeds; residual has $p = .209$–$.315$. The auxiliary energy-relative distortion predictor has Pearson $r = .162$ and Spearman $\rho = .329$, weaker than absolute distortion. These associations explain the observed mechanism; the prospective selection experiments use correction alignment on labeled validation pairs rather than GT distortion as an online signal.

F.5 Burgers Mass-Conservation Mechanism

We simulate periodic 1D viscous Burgers, $\partial_t u + \partial_x(u^2/2) = \nu \partial_{xx} u$, with $\nu = .02$ on 128 cells using a conservative finite-volume RK2 solver on $[0, 2\pi]$, with time step .001 and a saved frame every five solver steps (data seed 20260724). A 166,978-parameter circular CNN is trained from scratch on 80 trajectories, selected on 16 validation trajectories, and tested on 24 trajectories for 100 steps with seeds 42, 123, 456, 789, and 1011. The CNN uses five input frames, width 64 and four residual blocks; training uses 18 epochs of AdamW (learning rate 3×10^{-4}, weight decay 10^{-5}, batch size 128). For initial spatial mean m_0, the repair is

$$P_{m_0}(u) = u - \mathrm{mean}(u)\mathbf{1} + m_0\mathbf{1}, \qquad P_{m_0+b}(u) = P_{m_0}(u) + b\mathbf{1}. \tag{89}$$

P_{m_0} is the orthogonal projection onto the affine mass hyperplane. Biased repair exactly satisfies an intentionally wrong hyperplane; its near-zero imposed-rule residual does not mean it satisfies the true mass law.

Table 16: Burgers mass-repair mechanism experiment. Residual is the maximum absolute spatial-mean error relative to each operator's imposed mass, not necessarily the true mass. Rollout entries are mean $\pm$ sample SD over five seeds.

Repair	Target-distortion MSE	Max imposed-rule residual	MSE-AUC	MSE@100
Raw CNN	—	—	$(9.838 \pm 2.234) \times 10^{-6}$	$(2.672 \pm 0.6408) \times 10^{-5}$
Exact mass	6.26×10^{-18}	8.94×10^{-8}	$(8.051 \pm 1.325) \times 10^{-6}$	$(2.096 \pm 0.3257) \times 10^{-5}$
Biased $+.0025$	6.25×10^{-6}	5.96×10^{-8}	$(1.511 \pm 0.1643) \times 10^{-5}$	$(3.041 \pm 0.4036) \times 10^{-5}$
Biased $+.01$	1.00×10^{-4}	8.94×10^{-8}	$(1.213 \pm 0.03042) \times 10^{-4}$	$(1.662 \pm 0.08601) \times 10^{-4}$

Exact repair improves AUC by 18.2% and MSE@100 by 21.6%, in all five seeds. The paired Raw-minus-Exact AUC difference is 1.786×10^{-6}, with seed-level 95% CI $[1.91 \times 10^{-8}, 3.55 \times 10^{-6}]$ and crossed seed/trajectory bootstrap CI $[5.69 \times 10^{-7}, 4.42 \times 10^{-6}]$. Bias $+.0025$ raises AUC over Exact by 7.055×10^{-6}, bootstrap CI $[6.52, 7.47] \times 10^{-6}$. Error increases with distortion in every seed while the imposed-rule residual stays below 9×10^{-8}.

F.6 Prospective One-Step Selection on Disjoint Test Trajectories

Fresh CFDBench Cavity. Three fresh 4,733,815-parameter FNOs are trained for 80 epochs on the official Cavity split (seeds 42, 123, 456). All operator statistics use the 16 validation trajectories. The fixed base menu is Raw, Screened-2/4/8/16, Jacobi, and Direct. Each trajectory contributes the average over its teacher-forced windows, and trajectories receive equal weight. The operator and analytic strength minimizing the empirical score are written to CSV before any repair candidate is evaluated autoregressively on test trajectories. Seven official test trajectories contain the 23 frames required for three inputs and 20 future steps. All AUCs in Table 17 score steps 2–20 only, excluding the first forecast step. This tests the predictive operator/strength component; the test rollouts evaluate its long-horizon transfer.

Table 17: Prospective Cavity selection. Tuned-family regret compares against six analytically tuned candidates; full-menu regret compares against Raw, six full-strength repairs, and those six tuned candidates. Both oracles inspect test outcomes only for evaluation and never enter selection.

Seed	Choice	$\alpha^\star$	Raw future AUC	Chosen future AUC	Gain	Tuned regret	Full-menu regret
42	Screened-4	.0804	.162212	.161397	.503%	.0019%	.422%
123	Screened-4	.0692	.169426	.168628	.471%	.0000%	.577%
456	Screened-4	.0573	.182613	.181835	.426%	.0024%	.847%
Mean	—	.0690	.171417	.170620	.467%	.0014%	.615%

All three selections improve future-step AUC; 20 of 21 crossed seed–trajectory pairs improve. A paired two-way bootstrap with 100,000 resamples of both axes gives a 95% interval of $[.256, 1.274]\%$ for the mean .467% gain; the paired-seed t interval is $[.371, .562]\%$. Full-strength Jacobi and Direct have unfavorable validation margins in all six seed–operator cases and all six worsen test rollout. The full-menu oracle is full-strength Screened-16 in all three runs, illustrating that the one-step rule predicts a beneficial operating point without exactly recovering the long-horizon optimum.

Independent Burgers audit. A separate deterministic five-seed experiment uses the same Burgers simulator and CNN architecture, 80/16/24 train/validation/test trajectories, and 100-step test rollouts. The fixed 12-candidate menu comprises Raw; ExactMass; mass biases $-.0025, +.0025, +.01$; circular box smoothers of widths 3 and 5; Fourier low-pass maps with (cutoff, gain) $(56, .9), (48, .5), (32, .5)$; and sharpeners $I + \gamma(I - S_3)$ for $\gamma = .5, 1$, where S_3 is the width-3 box smoother. Eight repairs preserve mass-valid inputs' mass, although they need not preserve the full target state. Thus mass feasibility alone cannot rank them.

All candidates share teacher-forced validation predictions. For each operator, average the one-step risk difference within each validation trajectory and construct a Student-t 95% interval across the 16 trajectory means ($df = 15$); time and spatial samples remain clustered. A strictly negative interval predicts help, a strictly positive interval predicts harm, and an interval crossing zero abstains. These are pointwise screening intervals, not simultaneous post-selection coverage guarantees. Choices, scores, and intervals are recorded before any candidate test rollout. Full-strength one-step selection picks ExactMass in all five seeds, also the test oracle in every seed. Table 18 reports this independent audit, whose 14.39% gain is separate from the 18.2% mechanism experiment above.

Table 18: Independent prospective Burgers audit. Gain CI uses paired training-seed differences; sign decisions use trajectory-clustered validation intervals.

Statistic	Result
One-step selection equals test oracle	5/5 seeds
Mean oracle regret	0%
Non-abstained help predictions	5/5 correct
Non-abstained harm predictions	41/41 correct
Ambiguous margins	9/55
Operator-mean Δ_1 versus test AUC penalty	Spearman $\rho = .866$
ExactMass AUC gain versus Raw	14.39% $[9.13, 19.66]$

All 46 issued sign predictions are correct; the nine abstentions contain all three mistakes of the unthresholded sign. These observations support confidence-aware fallback rather than an unconditional long-horizon guarantee.

Existing-artifact temporal and cross-backbone checks. On six existing CFDBench settings, selection from step 1 and evaluation only on future steps $2{:}T_{\text{eval}}$ yields the correct repair-versus-Raw decision in 6/6 settings and the exact future-AUC-optimal operator in 5/6. The one mismatch is between neighboring mild Cavity operators, with .129% regret. This temporal diagnostic reuses existing trajectories and is distinct from the prospective disjoint-trajectory tests above. Across 46 repair comparisons in 12 FNO/U-Net tasks, one-step gain $-\Delta_1$ predicts rollout-AUC gain with Spearman $\rho = .895$ and 95% CI $[.777, .967]$. The interval uses 10,000 dataset-cluster bootstrap resamples; the two backbones for a dataset stay in the same one of six clusters.

F.7 Adaptive-Gating Sensitivity

Across 18 retained solver–dataset (τ, q) searches, each with three training seeds, the selected gate has validation MSE between 0 and 5.4% below its always-on operator. Re-selecting from one training seed's validation evidence, rather than all three seeds, gives mean regret .6% and worst-case regret 9.7% relative to the full-evidence optimum; 30/54 single-seed choices are exactly optimal. Leave-one-seed-out selection has worst-case regret 6.6%. This reduces independent training-run evidence, not the number of labeled validation trajectories. Mild operator families have broad parameter plateaus, whereas strong solvers exhibit sharper degradation. Raw remains in every deployment menu, and geometry-holdout choices use domain-matched validation; these tests do not assert robustness to arbitrary distribution shifts.

F.8 Metric Definitions and Aggregation

For N trajectories, m scalar state coordinates ($m = 2HW$ for velocity), target $u_{n,t}$, prediction $\hat{u}_{n,t}$, and evaluation horizon T_{eval}, define

$$\text{MSE}_t = \frac{1}{Nm} \sum_{n=1}^{N} \|\hat{u}_{n,t} - u_{n,t}\|_2^2, \qquad \text{MSE-AUC} = \frac{1}{T_{\text{eval}}} \sum_{t=1}^{T_{\text{eval}}} \text{MSE}_t. \tag{90}$$

MSE@T_{eval} is the final term; prospective Cavity instead averages $t = 2, \ldots, 20$. Test MSE averages teacher-forced prediction errors. Target-distortion MSE is the same per-coordinate error with $\hat{u} = T(u)$, averaged over labeled target snapshots. Energy-relative distortion averages $100\,\text{MSE}(T(u), u)/\text{mean}(u^2)$ per snapshot; it is not the ratio of two separately averaged quantities.

For each trajectory, divergence is the spatial RMS, and reported Div averages those RMS values across trajectories:

$$\text{Div}_t = \frac{1}{N} \sum_{n=1}^{N} \left[\frac{1}{HW} \sum_{i,j} \left(D_x \hat{u}_{n,t} + D_y \hat{v}_{n,t} \right)_{i,j}^2 \right]^{1/2}. \tag{91}$$

Periodic NS uses spectral derivatives on $[0, 2\pi]^2$ with the zero-Nyquist convention in Listing 1. Non-periodic cleanup uses the backward-divergence/forward-gradient stencil and copied boundaries in Appendix D.1; grid spacing is one in those diagnostics, and physical-spacing factors apply if units are changed. Boundary-strip/core diagnostics restrict the RMS to their stated masks. Physical velocity units are used before component normalization is restored. The pressure relative residual is $\|A_\lambda p - b\|_2/(\|b\|_2 + \epsilon)$, $b = -D_h u$, with a small denominator safeguard. It measures the pressure solve, not the velocity's divergence or forecasting error.

Burgers imposed-rule residual is $|\text{mean}(u) - m_{\text{imposed}}|$; true-mass error instead uses the initial mean m_0. Hierarchical coherence RMS is $\|x - \Pi_{\text{ols}} x\|_2/\sqrt{m}$ for m hierarchy nodes, averaged over the stated windows in original value units. Latency is elapsed inference time per reported prediction call in milliseconds; operator sweeps reuse checkpoints. Metrics are first aggregated within a fitted run; reported means and sample SDs summarize independent training seeds. Confidence intervals identify their paired seed, crossed seed/trajectory, or dataset-block sampling units explicitly. Forecasting MSE, target distortion, and constraint residual remain distinct quantities throughout.

G Limitations

The most extensive evidence is on 2D incompressible forecasting with FNO and independently trained U-Net backbones; 1D Burgers with CNNs and five forecasting hierarchies provide complementary coverage. These experiments do not exhaust 3D, compressible, mesh-based, or structured-prediction settings. Operator selection depends on representative validation: changes in geometry, resolution, or forcing require renewed alignment checks. The one-step identity is exact for squared-error risk, while long-horizon benefit depends on the composed dynamics and is evaluated empirically; numerical cleanup norms are not certified bounds on those dynamics. Our reduced-evidence gate audit varies training-seed evidence, not the number of validation trajectories.

For a deterministic or fixed-noise finite denoising chain, end-only repair treats the chain as F, and per-stage repair admits the same finite-horizon recursion with stage-dependent factors. Stochastic

Table 19: Cost comparison for the exact-regime and tuned cylinder settings. Evaluation-time cleanup baselines reuse the source checkpoint's training cost, while projected retraining pays its extra cost explicitly. Smaller values are better in every numeric column.

Track	Variant	Params	Train hours	Latency (ms)	Final-step rollout MSE
NS-128 exact	Raw-FNO	18.90M	0.84 ± 0.08	8.75 ± 0.25	$(9.389 \pm 6.291) \times 10^{-5}$
NS-128 exact	SoftDiv-FNO	18.90M	0.84 ± 0.07	8.69 ± 0.24	$(2.361 \pm 1.371) \times 10^{-4}$
NS-128 exact	PostHoc-FNO	18.90M	0.84 ± 0.08	11.39 ± 0.26	$(1.133 \pm 0.165) \times 10^{-6}$
NS-128 exact	Proj-FNO	18.90M	0.88 ± 0.08	11.49 ± 0.13	$(5.370 \pm 0.113) \times 10^{-7}$
NS-128 exact	CAP-FNO	18.90M	0.88 ± 0.08	11.42 ± 0.08	$(5.406 \pm 0.243) \times 10^{-7}$
Cylinder-20 tuned	Raw-FNO	6.44M	0.49 ± 0.00	5.50 ± 0.01	$(1.642 \pm 0.863) \times 10^{-3}$
Cylinder-20 tuned	AdaBlend-Screened-Tuned	6.44M	0.49 ± 0.00	11.79 ± 0.05	$(2.540 \pm 1.009) \times 10^{-3}$
Cylinder-20 tuned	PostHoc-Screened-Tuned	6.44M	0.49 ± 0.00	not reported	$(3.164 \pm 0.872) \times 10^{-3}$
Cylinder-20 tuned	Proj-Screened-Tuned	6.44M	0.64 ± 0.00	12.00 ± 0.36	$(1.409 \pm 0.028) \times 10^{-3}$

forecasters such as DYffusion require an additional study of predictive distributions and calibration: sample-wise repair can change both, beyond what deterministic paired rollouts and pointwise MSE establish. Noisy or intentionally nonconservative targets, a different evaluation geometry, or hidden solver state likewise require a fresh target-preservation audit.

H Reproducibility

Table 19 makes the cost structure explicit. On the exact NS-128 FNO track, Raw, SoftDiv, PostHoc, Proj, and CAP all use the same 18.9M-parameter backbone and nearly identical train hours, so the main exact-regime ordering is not a hidden capacity effect. The tuned cylinder cleanups show the complementary fairness point. AdaBlend-Screened-Tuned and PostHoc-Screened-Tuned reuse the raw checkpoint exactly, whereas Proj-Screened-Tuned raises training time from 0.49 to 0.64 hours because it retrains with the projected operator in the loop.

Training and Systems Details. The original CFD studies use AdamW, cosine decay, mixed precision, and validation-based checkpoint selection on four NVIDIA A100 GPU accelerators. The additional NS-128 replication also uses NVIDIA GH200 accelerators; the Burgers and prospective-test recipes are stated in Appendix F. We use PyTorch for training and evaluation. Appendix E uses an autoregressive GRU recipe for the hierarchical forecasting support track but keeps the same long-horizon validation protocol. For the long-trajectory cylinder benchmark, we subsample only the training windows with stride 16 while keeping validation and test rollout evaluation at stride 1, so the stronger cylinder recipe remains compute-matched without weakening the evaluation protocol.

Random Initialization Protocol. The main CFD experiment blocks in the main text, including the exact-regime tables, projector diagnostics, screened-family sweep, and from-scratch U-Net validation, use three random initializations unless noted otherwise in the underlying analysis ledger. The hierarchical forecasting study uses five random initializations on each of the five public hierarchies. The validation-to-deployment protocol on the four hardest approximate-regime rows also uses five random initializations per setting as a robustness extension. Table 5 uses the common three-run subset shared across all controlled-mismatch cleanup strengths so that the blends are compared on identical underlying checkpoints. The larger five-run deployment extension keeps the selected validation rule unchanged on all four hardest rows.

Statistical reporting. A training seed is the unit for reported mean $\pm$ sample SD. Paired seed intervals use $\bar{d} \pm t_{0.975,n-1} s_d/\sqrt{n}$ for within-seed intervention differences. Crossed bootstraps resample both the seed and trajectory axes while retaining paired operator evaluations; they do not treat crossed cells as independent runs. The correlation analysis instead blocks by dataset (Appendix F.4). Additional NS-128 toggles, propagation tests, and Burgers studies use five seeds; the prospective Cavity experiment uses three. These separate replications do not replace or pool the original table estimates.

Recorded compute and validation overhead. The retained original ledger sums 120.9 hours of A100 training wall time over 267 unique CFD checkpoints, including exploratory and negative-control

runs (10.36 hours for the exact NS-128 block and 3.03 for U-Nets). Legacy rows do not retain device multiplicity, so this sum is not an exact GPU-hour total. The five additional projected-training runs use 0.99 GH200 device-hours and the two added Raw runs 0.28, giving approximately 122 recorded training hours across these records, not a complete metered project footprint. Energy and CO_2 were not metered.

For the retained 68-setting validation grid, measured candidate latencies imply 3.9–11.3 minutes per run for sequential evaluation, versus 20.6–59.9 seconds for six deployment anchors. Batching 18–24 trajectories is estimated at 13.0–28.3 seconds for the full grid. These are kernel-equivalent estimates, excluding loading, not end-to-end timing measurements; no candidate requires retraining. The one-step criterion shares backbone forecasts across candidates, adding only operator application and score computation per candidate. Larger-backbone cost therefore depends on the shared forecast pass and on how many uncertain candidates require rollout fallback.

I List of Notation

Table 20: Notation summary. Symbols reused across sections are defined locally.

Symbol	Meaning
$\mathbb{R}, \mathcal{H}, \mathcal{M}$	real field, abstract state space, target manifold
$\boldsymbol{u} = (u, v)$	two-dimensional velocity field
$\tilde{\boldsymbol{u}}, \hat{\boldsymbol{u}}, \boldsymbol{y}$	raw prediction, cleaned prediction, target velocity
x_t, y_t, e_t	rollout state, target state, rollout error
$t, s, K, T_{\mathrm{in}}, T_{\mathrm{eval}}$	time index, rollout step, model rollout length, input length, evaluation horizon
H	spatial grid height
W	spatial grid width
$b_\theta, \Delta\tilde{\boldsymbol{u}}, F$	neural backbone, residual update, abstract predictor
$\alpha, \alpha^\star$	residual or blend weight, optimal blend
$\mathcal{T}, T, Q, Q^\star$	rollout repair operator, abstract cleanup, exact family member
$\Pi_{\mathrm{div}}, \widetilde{\Pi}_{\mathrm{bc}}, \widetilde{\Pi}$	exact periodic projector, boundary-aware cleanup, generic approximate cleanup
$\mathcal{P}, \mathcal{P}_\lambda$	selected post hoc cleanup, screened cleanup
$\nabla\cdot, \nabla\cdot_h, \nabla_h, \Delta_h$	continuous divergence, discrete divergence, gradient, and Laplacian
$\boldsymbol{k} = (k_x, k_y), \widehat{\cdot}, \bar{\boldsymbol{u}}$	Fourier wavevector, Fourier coefficient, spatial mean
ω, ψ	vorticity, stream function
$p, P, A, A_\lambda, \lambda$	pressure, pressure matrix, Poisson operator, screened shift
$S_x^{\pm}, S_y^{\pm}, S_{H-2}, S_{W-2}$	grid shifts, DST-I matrices
$\lambda_i^{(H)}, \lambda_j^{(W)}, \mu_\lambda$	Dirichlet eigenvalues, smallest eigenvalue
$D, G, r(x)$	abstract divergence, gradient, linear-system residual
$\mathcal{L}_{\mathrm{soft}}, \mathcal{L}_{\mathrm{vel}}, \mathcal{L}_\omega, \mathcal{L}_{\mathrm{spec}}$	soft-divergence, velocity, vorticity, spectrum losses
$\lambda_{\mathrm{div}}, k$	divergence penalty weight, cleanup iteration budget
$d(\tilde{\boldsymbol{u}}), \gamma, \tau, q$	divergence score, adaptive gate, threshold, gate exponent
$M_w, d(i,j), w, \odot$	taper mask, boundary distance, taper width, pointwise product
$\lambda_w, \Lambda_w, \lambda_{\mathrm{bdry}}, \lambda_{\mathrm{core}}$	spatial, diagonal, boundary, and core screened shifts
L, L_T, L_F, L_i	Lipschitz constants
$\delta_t, \beta_{i,t}, \beta_i$	raw one-step error, trajectory, and uniform target distortion
$b_t^{(i)}, \bar{\delta}, \bar{\beta}$	error-bound sequence, uniform error and distortion bounds
$c, x_\alpha, \phi(\alpha)$	cleanup increment, blended forecast, blend-error curve
$I, \|\cdot\|_2, \langle\cdot,\cdot\rangle, (\cdot)^\top$	identity, Euclidean norm, inner product, transpose
$S, z_t, m, b, \mathcal{C}$	hierarchy summing matrix, bottom series, node counts, coherent manifold
$\Pi_{\mathrm{ols}}, \Pi_{\mathrm{bu}}, \widetilde{\Pi}_{\mathrm{td}}, \widetilde{\Pi}_{\mathrm{td},\alpha}$	OLS, bottom-up, top-down, blended top-down reconciliation
$G_{\mathrm{bu}}, \bar{p}, e_{\mathrm{root}}$	bottom extractor, historical proportions, root selector
$P, \tilde{P}, \|\cdot\|_{\mathrm{op}}$	reference and perturbed repair maps, induced operator norm
d_t, ϵ, B, K	rollout perturbation, operator difference, forecast bound, repair gain bound
$A_T, B_T, \Delta_1, S(T)$	correction alignment, correction energy, one-step risk change, selection score
$\Gamma_T, \alpha_T^\star, \kappa_T$	alignment ratio, analytic strength, composed-dynamics Lipschitz factor
$\eta, \delta, T_\eta, B_\delta$	obliqueness, target shrinkage, controlled operator families
r_{P}, ρ, n	Pearson correlation, Spearman correlation, sample count
$\mathrm{MSE@}T_{\mathrm{eval}}, \mathrm{Div@}T_{\mathrm{eval}}, \mathrm{AUC}$	final-step error, final-step divergence, rollout area metric

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the operator-exactness and amplification mechanisms, strong projection baselines, and signed one-step selection rule. Section 4 and Appendix F support these claims with paired interventions, cross-PDE evidence, and prospective validation. The one-step guarantee and empirical long-horizon transfer are distinguished explicitly.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The Limitations section states the demonstrated PDE/backbone coverage, the need for representative validation under geometry, forcing, or resolution shifts, the conditional stability assumptions, and the separate evaluation required for stochastic forecast distributions. It also distinguishes reduced training-seed evidence from reduced validation-trajectory count.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Appendix A states assumptions and proofs for orthogonal projection, exact-on-target repair, solver error, target distortion, controlled mismatch, operator perturbation, and signed one-step risk. Contraction-dependent steady-state results are labeled as conditional corollaries. Appendix E proves the hierarchical exactness statements.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 4 and Appendices H and F specify benchmark splits, recipes, horizons, sampling units, paired interventions, and prospective selection before test rollouts. Appendices C and D.1 define the operators; Listing 1 provides the exact projector.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The exact periodic projector is provided as supplemental code and in Listing 1. The study uses public benchmark data and a specified synthetic Burgers simulator; Appendices H and F describe experimental and selection protocols. The released code asset is the compact projector, not a packaged reproduction of every experiment.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 and Appendices C, D, and F give the splits, models, operator settings, horizons, metric definitions, and validation procedures. Appendix H reports optimization, hardware, recorded compute, and initialization counts.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Tables report mean $\pm$ sample SD across training seeds. Appendix H defines paired seed and crossed seed/trajectory intervals. Appendix F.4 reports Fisher intervals and exact within-dataset permutation tests for the 12 dependent conditions. Prospective results report their trajectory-clustered and paired bootstrap sampling units.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Appendix H reports A100/GH200 hardware, parameter counts, representative training times, the retained compute ledger, and latency-based validation-overhead estimates. It distinguishes recorded wall time from device-hours and estimates from measured end-to-end time. Memory/storage requirements and a complete metered project footprint are unavailable; energy and CO_2 were not metered.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

Justification: The research uses public scientific and hierarchical forecasting benchmarks, does not involve human subjects or sensitive personal data collection, and introduces no high-risk deployed model. We are not aware of any deviation from the NeurIPS Code of Ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: The work is foundational methodology and evaluation research for scientific and structured forecasting benchmarks, with no deployed system or direct application to decisions about people. Any societal impacts are indirect and tied mainly to improving reliability standards for scientific forecasting.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release pretrained language models, image generators, scraped datasets, or other assets with a high risk of misuse. The experiments use scientific benchmark data and compact algorithmic code rather than deployable dual-use models.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: The paper credits the main existing datasets, methods, and software ecosystems through citations, including CFDBench, HierarchicalForecast, GRU, FNO, and PyTorch-based implementation context. It does not explicitly enumerate license names, versions, or terms of use for each existing asset.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The only new released asset is the compact exact periodic Hodge projection implementation, which is documented in Appendix B and included as supplemental code. The paper does not introduce a new dataset or model checkpoint.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing, participant studies, surveys, annotation labor, or other research with human subjects. All experiments use public or synthetic forecasting benchmarks.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The research does not involve human subjects, so participant risks, disclosure, compensation, and IRB or equivalent approvals are not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not an important, original, or non-standard component of the core methodology. The paper's methods and experiments are based on CFD projection/cleanup operators, neural forecasting backbones, and hierarchical reconciliation.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.