FROST: STOCHASTIC PREDICTION-SPACE ADAPTATION FOR SIM-TO-REAL PDE FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Simulation pretraining followed by real-data fine-tuning is a natural strategy for adapting learned PDE forecasters, such as neural operators, to physical experiments. Yet deterministic adaptation can leave a structured mismatch between forecasts and experimental futures, whose variability reflects unresolved dynamics and measurement noise. To this end, we introduce FROST (Frozen-Operator Stochastic Transport), which freezes an upstream forecaster during correction and learns the conditional distribution of its residual from real measurements. Conditioned on observed history and the base forecast, FROST corrects systematic bias while modeling structured variability around the forecast. Across five RealPDEBench systems, FROST improves long-horizon ensemble-mean accuracy and fair CRPS over the corresponding fine-tuned FNO on every system, with reductions of up to 61% and 75%, respectively, and matches or outperforms every deterministic and generative baseline in mean rollout accuracy on all five systems. Additionally, we find that when applying FROST, the base FNO need not be fine-tuned at all. Together, these results establish stochastic prediction-space correction as a powerful complementary axis of sim-to-real adaptation for PDE forecasting.

1 INTRODUCTION

Learned PDE surrogates such as neural operators are trained and evaluated almost entirely on numerical simulation (Li et al., 2021; Takamoto et al., 2022; Ohana et al., 2024; Koehler et al., 2024; Herde et al., 2024; McCabe et al., 2024), while their value lies in predicting the physical world. Simulation data is abundant, but it differs from experiment through modeling assumptions, unresolved dynamics and the measurement error. When applying a simulation-trained surrogate forecaster to real-world settings, performance can degrade even under matched physical conditions (Hu et al., 2026). The standard recipe is to pretrain on simulation and fine-tune on the real data, with careful ways of regularizing or restricting the update (Kumar et al., 2022; Zhou et al., 2026).

This raises a basic question: *must the entire sim-to-real discrepancy be absorbed through parameter updates to the forecaster?* Most forecasters are deterministic, whereas prediction from experimental measurements is generally not: the observed history is finite and noisy, and may omit boundary conditions, forcing variables, or unresolved physical state. Empirically, we find that real-data fine-tuning can still yield systematic deficiencies in the forecasts. Such fine-tuned forecasts exhibit distorted fluctuation spectra, including substantial loss of high-wavenumber energy, and their residual errors retain pronounced spatial and temporal dependence.

We therefore introduce FROST (Frozen-Operator Stochastic Transport), a prediction-space adaptation method that models the residual distribution around a frozen forecaster. FROST treats the deterministic forecast as a predictive reference and learns, from real measurements, a conditional distribution over the remaining discrepancy given both the observed history and the forecast. We parameterize this distribution using a stochastic interpolant, yielding an ensemble of corrected trajectories rather than a single deterministic adjustment.

Across five RealPDEBench systems, FROST reduces both ten-block rollout error and fair CRPS relative to the corresponding fine-tuned FNO on every system, with improvements of up to 60% and 75%, respectively. The corrected trajectories also improve structural statistics, including measured fluctuation energy. These gains are not specific to FNOs: the same correction framework improves forecasts from FNO, CNO, U-Net, DeepONet, Transolver, and DPOT-S backbones.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

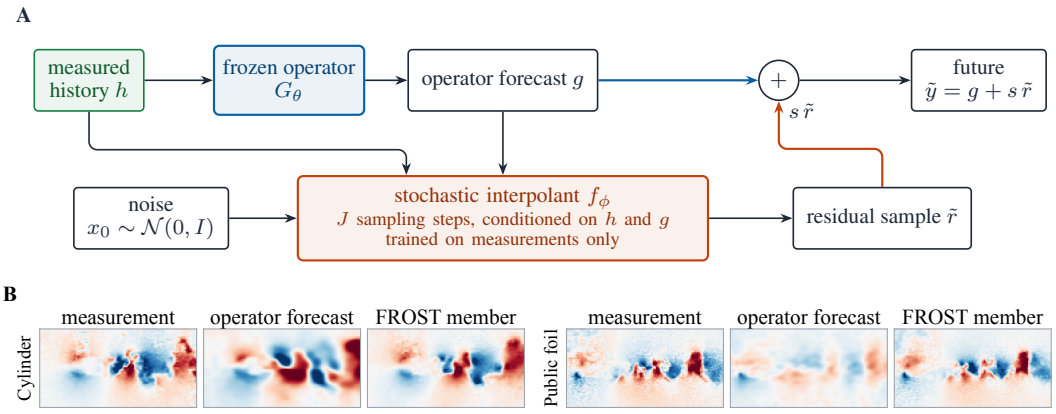

Figure 1: **Predict the predictable, generate the rest.** **(A)** FROST keeps the fine-tuned forecaster G_θ frozen; its forecast g conditions the corrector and anchors the sample, a residual $\tilde{r}$ that a stochastic interpolant transports from noise in J steps, trained on measurements alone. **(B)** The operator forecast smooths the measured structure; a FROST member restores it.

The main contributions of this paper are:

- We characterize the residual errors remaining after real-data fine-tuning, showing that they exhibit structured spatiotemporal dependence and vary systematically with the base forecast.
- We introduce FROST, which treats the deterministic forecast as a predictive anchor and learns a stochastic correction for the remaining discrepancy from real measurements.
- Across five RealPDEBench systems, FROST consistently improves both forecast accuracy and probabilistic skill over standard fine-tuning. The correction generalizes across physical regimes and forecaster architectures, including transfer to an unseen architecture, and can even adapt a simulation-only operator without fine-tuning its parameters on real data.

2 Problem Setup

2.1 Sim-to-Real Forecasting

Let $u_t \in \mathbb{R}^{H\times W\times C}$ denote the measured state of a physical system at frame t. A forecaster receives a history $h = (u_{t-T_{\text{in}}+1}, \ldots, u_t)$ and predicts a future $y = (u_{t+1}, \ldots, u_{t+T_{\text{out}}})$. Long-horizon predictions are done autoregressively: $\hat{y}_b = G(h_b)$ and $h_{b+1} = \text{tail}_{T_{\text{in}}}[h_b, \hat{y}_b]$ for B blocks. The measurement is a partial and noisy observation of the underlying physical state, $u_t = \mathcal{H}(\mathbf{u}_t) + \eta_t$, where $\mathcal{H}$ restricts the three-dimensional state $\mathbf{u}_t$ to a finite-resolution observation plane and η_t denotes measurement noise (Raffel et al., 2018; Sciacchitano, 2019). Moreover, the measured history h does not include all boundary and forcing information required to determine the future uniquely. Thus, "stochastic" in this paper refers to uncertainty in the conditional forecasting problem induced by partial, finite, and noisy observations, rather than stochasticity in the governing equations.

Two sources of data are available. Simulation provides abundant pairs $(h, y) \sim \mathcal{D}_{\text{sim}}$ that are exact under the assumptions of the numerical model, whereas experiments provide comparatively scarce pairs $(h, y) \sim \mathcal{D}_{\text{real}}$. The standard sim-to-real approach trains a deterministic operator on $\mathcal{D}_{\text{sim}}$ and subsequently fine-tunes its parameters on $\mathcal{D}_{\text{real}}$ using the same pointwise objective,

$$\theta_0 = \arg\min_\theta \; \mathbb{E}_{\mathcal{D}_{\text{sim}}} \|G_\theta(h) - y\|^2, \qquad \theta = \arg\min_{\theta \leftarrow \theta_0} \; \mathbb{E}_{\mathcal{D}_{\text{real}}} \|G_\theta(h) - y\|^2, \tag{1}$$

and then deploys G_θ (Figure 1A). We write $g = G_\theta(h)$ for the fine-tuned forecast, $\delta = y - g$ for its residual relative to the measured future, and $r = \delta \oslash s$ for the residual normalized by the per-channel standard deviation s estimated from the training residuals. The corresponding conditional discrepancy is $p(\delta \mid h, g)$. Standard deterministic adaptation produces a single forecast and implicitly seeks to drive δ toward zero. We instead study the structure of $p(\delta \mid h, g)$ after fine-tuning and ask whether modeling this residual distribution provides a complementary form of sim-to-real adaptation.

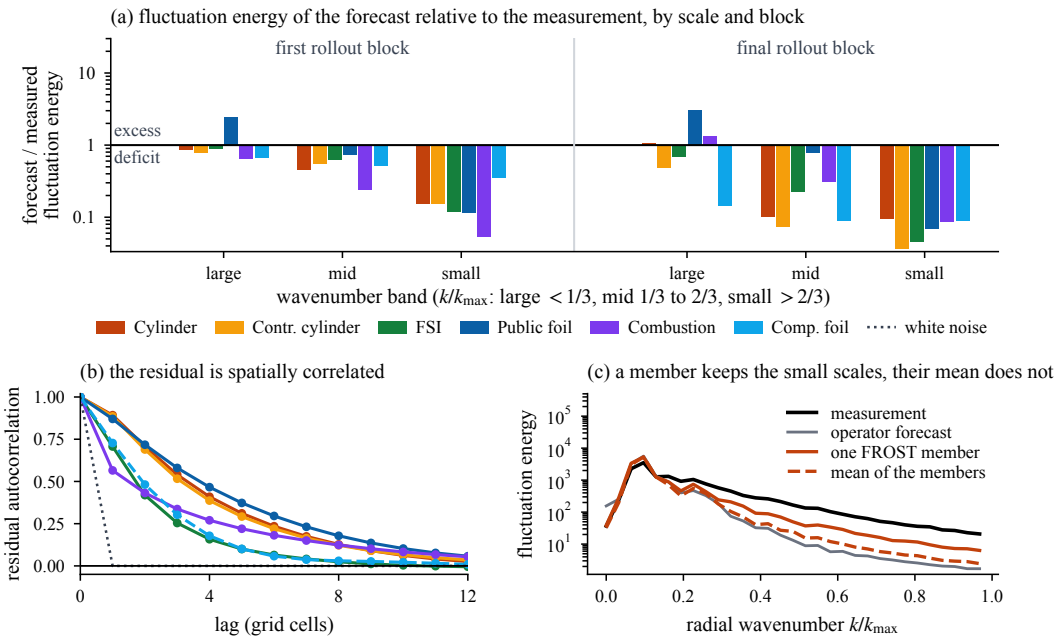

Figure 2: **Fine-tuned operators lose fluctuation energy at small scales and leave a spatially correlated residual.** Every forecaster is a fine-tuned FNO. **(a)** Forecast fluctuation energy as a fraction of the measurement's, summed within three wavenumber bands, at the first and the final rollout block; below the line is a deficit. **(b)** Spatial autocorrelation of the residual $y - g$. **(c)** Fluctuation energy by scale for one cylinder block.

2.2 Residual Structure After Fine-Tuning

Fine-tuning can achieve low pointwise error while leaving systematic errors in fluctuation statistics and structured, forecast-dependent residuals. Figure 2 demonstrates these effects across all five systems; simulation-only operators are considered in Section 4.2.

Fine-tuned operators have systematic fluctuation errors. We decompose each rollout block into its temporal mean and fluctuation, $y = \bar{y} + y'$, and compare fluctuation energy across spatial scales. Fine-tuned FNOs exhibit a pronounced high-wavenumber energy deficit across all five systems, which emerges and compounds over autoregressive rollout (Figure 2a). This deficit is not explained by measurement noise, which accounts for only 0.2–4% of measured fluctuation variance, nor is it specific to FNOs: several other forecasters exhibit similar deficits, while Transolver exhibits excess energy instead (Appendix B.3; Appendix B.1).

Residual errors are structured and forecast-dependent. Residuals exhibit substantial spatial and temporal correlation, ruling out independent noise as an adequate model of the remaining discrepancy (Figure 2b). Forecast interventions further show that correction depends on the time-resolved prediction, as removing the forecast, compressing it to summary statistics, or disrupting its temporal alignment degrades performance, while a better-aligned oracle forecast improves it (Appendix B.2).

A deterministic residual correction cannot represent phase uncertainty. When plausible futures differ in the phase or location of coherent structures, their conditional mean averages those structures away. Figure 2c illustrates this distinction: individual samples can retain sharp fluctuation structure even when their mean is comparably smooth to the deterministic forecast.

This effect compounds under autoregressive rollout, where errors in fluctuation structure become part of the subsequent conditioning history. Accordingly, rollout error increases as fluctuation energy decays across blocks on competition foil, controlled cylinder, and FSI (Figure 2a), while Appendix G.1 attributes much of the single-block to ten-block gap to this generated-history shift. Together, these results motivate modeling the remaining discrepancy as a forecast-conditioned residual distribution.

3 FROST METHOD

FROST freezes the base forecaster and learns a conditional distribution over its residual using a stochastic interpolant. At inference, samples from this distribution are propagated autoregressively, with each ensemble member conditioned on its own generated history. We describe the residual model, the stochastic transport used to sample from it, and the resulting member-wise rollout.

3.1 CONDITIONAL RESIDUAL MODEL

Given the forecast $g = G_\theta(h)$ of the measured history h and the normalized residual $r = (y - g) \oslash s$ from Section 2.1, FROST learns the conditional residual model:

$$\tilde{r} \sim q_\phi(r \mid h, g), \qquad \tilde{y} = g + s \odot \tilde{r}, \tag{2}$$

where q_ϕ denotes the learned residual distribution, and s is the per-channel residual scale defined in Section 2.1. We refer to the neural network f_ϕ that parameterizes this distribution as the *corrector*. Only the corrector parameters ϕ are trained; the forecaster G_θ remains frozen.

In the main experiments, G_θ is pretrained on simulation and fine-tuned on real measurements before being frozen. Section 4.2 applies the same construction to a simulation-only forecaster, and Appendix C.7 considers alternative prediction targets, including direct field generation. The corrector conditions jointly on the measured history and the base forecast. We concatenate them along time, $c = [\, h \,\|\, g \,]$, and use a binary mask m to distinguish history from forecast frames (Figure 1A; Appendices C.3 and C.4). The forecast thus plays two roles: it conditions the residual model and provides the anchor to which the sampled residual is added in Eq. (2).

3.2 STOCHASTIC RESIDUAL TRANSPORT

We use a stochastic interpolant (Albergo & Vanden-Eijnden, 2023; Albergo et al., 2025) to model $p_\phi(r \mid h, g)$. For each training example, we draw $x_0, z \sim \mathcal{N}(0, I)$ and $t \sim \mathcal{U}(0, 1)$, and define:

$$x_t = (1 - t)x_0 + t\, r + \gamma(t)z, \qquad \gamma(t) = \sigma\sqrt{t(1 - t)}. \tag{3}$$

The corrector $f_\phi([x_t, c, m], t)$ predicts two quantities, (v_ϕ, η_ϕ), corresponding to the endpoint displacement and bridge noise on the future frames. With $M = 1 - m$, we optimize:

$$\mathcal{L}(\phi) = \mathbb{E}_{t,x_0,z,h,y}\left[\|M \odot (v_\phi - (r - x_0))\|_2^2 + \|M \odot (\eta_\phi - z)\|_2^2\right]. \tag{4}$$

We sample t uniformly and apply no time-dependent loss weighting. Optimizing the population loss recovers the conditional expectations that determine the probability-flow drift and score,

$$a_\phi = v_\phi + \gamma'(t)\eta_\phi, \qquad \nabla_x \log p_t(x \mid c) = -\eta_\phi/\gamma(t), \tag{5}$$

which yield the following family of samplers:

$$\mathrm{d}x_t = \left[a_\phi(x_t, t, c) - \frac{\varepsilon}{\gamma(t)}\,\eta_\phi(x_t, t, c)\right]\mathrm{d}t + \sqrt{2\varepsilon}\,\mathrm{d}W_t. \tag{6}$$

When the learned fields are exact, these dynamics preserve the interpolant marginals (Albergo et al., 2025; Song et al., 2021; Ma et al., 2024). Setting $\varepsilon = 0$ gives the probability-flow ODE, with randomness entering through x_0; setting $\varepsilon = \sigma^2/2$ gives the corresponding stochastic sampler. The source variables are independent of the experimental data, so all conditional structure is learned through f_ϕ from (h, g). Thus, the stochastic correction learns where and how the realized future may deviate from the deterministic forecast, rather than injecting unconditional variability.

3.3 MEMBER-WISE ROLLOUT

All ensemble members begin from the same measured history and thereafter evolve independently using their own sampled predictions. For block b and member k,

$$g_b^{(k)} = G_\theta\left(h_b^{(k)}\right), \qquad \tilde{y}_b^{(k)} = g_b^{(k)} + s \odot \tilde{r}_b^{(k)}, \qquad h_{b+1}^{(k)} = \mathrm{tail}_{T_{\mathrm{in}}}\left[h_b^{(k)}, \tilde{y}_b^{(k)}\right]. \tag{7}$$

Table 1: **FROST improves accuracy and probabilistic skill over the fine-tuned FNO on all five systems.** Ten-block rollout relative L_2 of the ensemble mean and fair CRPS on the benchmark's 320 test windows, $K = 4$ members, calls counted per member and block. PhysGuard and FROST are three-seed means over our fine-tuned FNO; the U-Net, DPOT-S and Transolver rows are the benchmark's released fine-tunes read as published (–: no usable release). Bold: best full-window entry per column. †128 windows; ‡160 windows.

Method	Cylinder		Contr. cyl.		FSI		Public foil		Combustion	
	L_2	CRPS	L_2	CRPS	L_2	CRPS	L_2	CRPS	L_2	CRPS
Fine-tuned FNO	0.163	0.0220	0.173	0.0128	0.227	0.0177	0.075	0.0136	0.792	0.0092
Fine-tuned U-Net	0.182	0.0243	0.166	0.0122	–	–	0.037	0.0053	–	–
Fine-tuned DPOT-S	0.174	0.0234	0.168	0.0123	–	–	0.036	0.0066	–	–
Fine-tuned Transolver	0.324	0.0404	0.265	0.0218	–	–	0.055‡	0.0115‡	–	–
PhysGuard	0.169	0.0252	0.174	0.0129	0.226	0.0176	0.083	0.0145	0.826	0.0096
FROST (ODE-6, 6 calls)	**0.157**	**0.0177**	**0.149**	**0.0083**	**0.213**	**0.0126**	**0.030**	**0.0034**	**0.775**	**0.0069**
FROST (SDE-30, 30 calls)†	0.154	0.0152	0.149	0.0072	0.219	0.0117	0.029	0.0030	0.767	0.0064

The ensemble mean is used only for evaluation and is never fed back into the rollout, since averaging phase-uncertain trajectories suppresses their fluctuation structure. With K members and J transport steps, each block requires K operator evaluations and KJ corrector evaluations (Appendix C.6). Algorithm 1 gives the complete training and sampling procedure.

Alternative instantiations. The prediction-space correction framework is fairly general. Appendix C.7 considers one-step MeanFlow for the same residual target, as well as direct field generation and forecast-anchored bridges as alternative parameterizations (Geng et al., 2025a; Albergo et al., 2024; Chen et al., 2024). The deterministic residual regressor of Section 4.4 uses the same conditioning and network architecture, but replaces the generative objective with regression.

4 Experiments

4.1 Setup

Task and systems. We evaluate on RealPDEBench (Hu et al., 2026), which provides paired measurements and matched simulations for five physical systems (Table 7 in Appendix A). The fluid systems are observed through planar PIV velocity fields, while combustion is represented by a single chemiluminescence channel. Each task forecasts 20 future frames from 20 historical frames (10 and 10 for controlled cylinder). The systems are:

- **Cylinder**: flow past a cylinder over a range of Reynolds numbers.
- **Controlled cylinder**: cylinder wake under active control.
- **FSI**: fluid–structure interaction with a moving body.
- **Public foil**: flow past a foil over varying angles of attack.
- **Combustion**: turbulent combustion observed through chemiluminescence.

Unless otherwise noted, all results use the benchmark evaluator on the measured channels, with its 320 selected test windows and ten-block rollout protocol (Appendix A.3). The competition-foil dataset, which uses a different grid and trajectory-held-out evaluation protocol, is used only for the diagnostic analysis in Figure 2 and Appendix G.7.

Baselines and controls. We compare FROST against four classes of alternatives:

- **Fine-tuned FNO.** Our primary reference is the RealPDEBench FNO pretrained on simulation and fine-tuned on real measurements. This is the forecaster corrected by FROST in the main experiments. We also evaluate a **simulation-only FNO** without real-data fine-tuning.
- **Alternative forecasters.** We evaluate the released RealPDEBench fine-tuned checkpoints for U-Net (Ronneberger et al., 2015), DPOT-S (Hao et al., 2024), and Transolver (Wu et al., 2024) where available, and on cylinder also DeepONet (Lu et al., 2021) and a CNO (Raonić et al., 2023) trained from scratch on the real split.

Table 2: **FROST improves every frozen cylinder forecaster, and one corrector trained on a mixture of five of them also improves the CNO it never saw.** Each row is a frozen forecaster alone, then with a corrector trained against it (three seeds, bold), the relative reduction in rollout error, the corrected fair CRPS, and the error when the single mixture-trained corrector is read with that row's forecaster instead. 320 test windows; the CNO is trained from scratch on the real split and the other backbones are released fine-tuned checkpoints (Appendix E.3).

Cylinder: frozen forecaster	alone	+FROST	gain	CRPS	mixture corrector
FNO (ours)	0.1632	**0.1571 ± 0.0003**	4%	0.01772	0.1601 ± 0.0011
DPOT-S	0.1744	**0.1588 ± 0.0014**	9%	0.01784	0.1584 ± 0.0007
U-Net	0.1817	**0.1614 ± 0.0015**	11%	0.01901	0.1690 ± 0.0020
DeepONet	0.2341	**0.1597 ± 0.0029**	32%	0.01842	0.1700 ± 0.0005
Transolver	0.3238	**0.1651 ± 0.0024**	49%	0.01940	0.1705 ± 0.0036
CNO *(held out from mixture)*	0.2239	**0.1680 ± 0.0030**	25%	0.01848	0.1709 ± 0.0027

- **Parameter-space adaptation.** We compare against PhysGuard (Zhou et al., 2026), which constrains fine-tuning using a simulation-derived Fisher subspace. We reimplement PhysGuard with rank 60 and three seeds per system (Appendix D.1). Additional fine-tuning and parameter-efficient adaptation baselines, including LP-FT, pushforward, EWC, L2-SP, LoRA, BitFit, and CORAL, are reported in Appendices D.2–D.5.
- **Matched correction controls.** To isolate the value of structured stochastic correction, we compare against three controls using the same network, conditioning inputs, and real training data as FROST: a variance-matched Gaussian perturbation, a deterministic residual regressor, and a PDE-Refiner-style denoiser (Lippe et al., 2023) (Table 6; Appendix F.2).

Training and evaluation. We use a single FROST configuration across all five systems: $\sigma = 0.15$, $K = 4$ ensemble members, and 20,000 AdamW updates with exponential moving-average weights. Unless otherwise noted, inference uses six Euler steps of the probability-flow ODE; a 30-step Euler–Maruyama sampler is used for calibration experiments (Appendix C.6). We train three corrector seeds for each forecaster. Repeated cylinder trainings give a run-to-run variation of 0.0018 in rollout error, which we use as the threshold for statistical ties in tables (Appendix A.3).

Metrics. We evaluate three complementary aspects of forecasting performance (Appendix A.4):

- **Accuracy**: relative L_2 error of the ensemble mean over the ten-block rollouts.
- **Probabilistic skill**: fair CRPS (Ferro, 2014).
- **Trajectory fidelity**: fluctuation amplitude and fluctuation-energy error of individual ensemble members relative to the measurements.

4.2 Main results

FROST improves upon standard fine-tuning. Applied to the fine-tuned FNO, FROST reduces ten-block rollout error and fair CRPS on all five systems, by 17% and 37% on average and by as much as 60% and 75%, respectively (Table 1). Among the adaptation and correction methods we evaluate, FROST is the only one that improves both metrics on every system. Additional parameter-space and probabilistic baselines are reported in Appendix D; in particular, probabilistic retrofitting improves CRPS but degrades point accuracy on all five systems (Appendix D.5).

The forecaster need not be fine-tuned on real data. We next freeze the simulation-pretrained FNO before it has seen any real measurements and train only the FROST corrector on the same real split. Despite the substantially weaker base forecast, the corrected simulation-only FNO is more accurate than the uncorrected fine-tuned FNO on four of five systems and matches or outperforms the fine-tune-then-correct pipeline on three (Table 3).

The same correction recipe applies across forecaster architectures. FROST improves all six frozen forecasters on cylinder (Table 2). Their uncorrected rollout errors range from 0.163 to 0.324, whereas the corrected forecasts lie in a narrow range of 0.157–0.168. This indicates that prediction-space correction is not specific to the FNO backbone. Section 4.3 asks the stronger question of whether a corrector trained around one forecaster can transfer to another.

Table 3: **FROST adapts a simulation-only FNO without fine-tuning the operator on real data.** The corrector is trained on the same real split in both rows (three seeds; SD at most 0.0073, Table 23); only the frozen operator differs. Corrected, the operator that never saw a measurement is more accurate than the uncorrected fine-tuned FNO on four of five systems. Bold: the better corrected value per system, within the 0.0018 floor.

Frozen operator	Rollout relative L_2: the operator alone → the operator corrected by FROST				
	Cylinder	Contr. cyl.	FSI	Public foil	Combustion
Simulation-only FNO	0.460 → **0.157**	0.552 → 0.152	0.640 → **0.210**	0.132 → **0.030**	0.892 → 0.798
Fine-tuned FNO	0.163 → **0.157**	0.173 → **0.149**	0.227 → 0.213	0.075 → **0.030**	0.792 → **0.775**

Table 4: **FROST improves both accuracy and probabilistic skill across every seen and held-out parameter group.** One row per parameter partition of one system's test windows; FROST is a three-seed mean. "Out of range" is the benchmark's label for a held-out group rather than a strict extrapolation test: eight of cylinder's ten such trajectories stay inside the Reynolds range seen in training. FSI has no held-out in-range group, and combustion has no held-out group. Relative L_2 is normalized within each subset, so the rows do not average to Table 1, whereas CRPS does.

System	Parameter group	Windows	FNO	FROST	Δ	FNO CRPS	FROST CRPS
Cylinder	seen parameters	187	0.1541	**0.1505 ± 0.0002**	-2.4%	0.0121	**0.0089 ± 0.0000**
	held-out, in range	69	0.1636	**0.1545 ± 0.0019**	-5.5%	0.0565	**0.0482 ± 0.0004**
	held-out, out of range	64	0.1557	**0.1477 ± 0.0013**	-5.1%	0.0137	**0.0106 ± 0.0001**
Contr. cyl.	seen parameters	190	0.1678	**0.1504 ± 0.0012**	-10.4%	0.0129	**0.0085 ± 0.0001**
	held-out, in range	65	0.1546	**0.1368 ± 0.0009**	-11.5%	0.0127	**0.0081 ± 0.0001**
	held-out, out of range	65	0.2550	**0.1837 ± 0.0008**	-28.0%	0.0127	**0.0076 ± 0.0001**
FSI	seen parameters	236	0.2199	**0.2050 ± 0.0004**	-6.8%	0.0181	**0.0129 ± 0.0000**
	held-out, out of range	84	0.2498	**0.2404 ± 0.0010**	-3.8%	0.0169	**0.0118 ± 0.0000**
Public foil	seen parameters	192	0.0707	**0.0235 ± 0.0002**	-66.7%	0.0129	**0.0025 ± 0.0000**
	held-out, in range	63	0.0646	**0.0169 ± 0.0001**	-73.8%	0.0096	**0.0012 ± 0.0000**
	held-out, out of range	65	0.0812	**0.0406 ± 0.0018**	-50.0%	0.0197	**0.0081 ± 0.0002**
Combustion	seen parameters (no held-out group)	320	0.7918	**0.7750 ± 0.0030**	-2.1%	0.0092	**0.0069 ± 0.0000**

Further experiments. Additional experiments cover the sampling budget (Appendices E.5 and E.6), the amount of real training data (Appendix E.2), generated-history and longer rollouts (Appendices G.1, G.2 and G.4), and conditioning choices (Appendix C.8).

4.3 Generalization

FROST generalizes across held-out physical parameters. We evaluate FROST separately on the benchmark's seen and held-out parameter groups (Table 4). FROST lowers both rollout error and fair CRPS in every partition of every system, including every held-out group. The gains therefore extend to held-out trajectories that contribute no window to the real training data (Appendix A.2). Appendix E.1 gives trajectory-resampled intervals for the per-system gains; every interval against the fine-tuned FNO excludes zero.

Mixture training enables transfer across forecasters. A corrector trained around a single forecaster is generally specialized to that forecaster: transfer degrades as the test forecast differs from the one used during corrector training (Table 5). Training instead on a mixture of five cylinder forecasters substantially reduces this dependence. The resulting corrector stays within 0.011 of the matched corrector on every forecaster and transfers to the held-out CNO, reducing its rollout error from 0.2239 to 0.1709, compared with 0.1680 for a CNO-specific corrector. This suggests that a single corrector can learn a correction rule shared across a family of forecasters rather than one tied to a single backbone (Appendices E.3–E.4).

4.4 Structure of the residual

Members recover fluctuation structure suppressed by the operator. Figure 3 illustrates how individual members restore coherent structures that the deterministic forecast smooths or misplaces.

Table 5: **A corrector learns its forecaster's error geometry, and a mixture-trained corrector transfers, including to a held-out CNO.** Every cell is a rollout relative L_2 on cylinder: the first row is each forecaster left uncorrected, and every row below it is one corrector read with each forecaster in turn, the condition and the anchor changing together. Three corrector seeds in every corrected cell (SDs of 0.0002–0.014 omitted for legibility). Underlined: matched training; bold: best in the column within the floor. Persistence and climatology columns feed the corrector a trivial forecast.

Corrector trained on \ read with	FNO	DPOT-S	U-Net	CNO	DeepONet	Transolver	Persist.	Climo.
Forecaster alone	0.1632	0.1744	0.1817	0.2239	0.2341	0.3238	0.3656	0.3600
FNO	**0.1571**	0.1641	0.1786	0.1799	0.2076	0.2598	0.3534	0.3380
DPOT-S	0.1670	**0.1588**	0.1821	0.1947	0.2145	0.2489	0.3196	0.3163
U-Net	0.1664	0.1806	**0.1614**	0.1895	0.2093	0.2738	0.3517	0.3430
CNO	0.1609	0.1708	0.1689	**0.1680**	0.2111	0.2807	0.3474	0.3428
DeepONet	0.1807	0.1716	0.1883	0.1887	**0.1597**	0.2049	0.3136	0.2835
Transolver	0.2008	0.1893	0.1997	0.2297	0.1980	**0.1651**	0.3277	0.2945
Five-way mixture	0.1601	**0.1584**	0.1690	0.1709	0.1700	0.1705	**0.2891**	**0.2529**

Table 6: **FROST improves the fluctuation statistics of individual forecast trajectories.** We report the fluctuation-amplitude ratio (ideal value 1) and relative fluctuation-energy error of individual members over the ten-block rollout. The Gaussian null adds white noise, while the deterministic regressor predicts the conditional mean of the residual. $\text{FROST}_{\text{sim}}$ denotes FROST applied to the simulation-only FNO. Statistics are computed per member before averaging; additional member metrics are reported in Appendix F.1. Grey denotes a single sampling seed.

System	Member amplitude ratio (1 is exact)					Member fluctuation-energy error				
	FNO	Null	Regressor	FROST	$\text{FROST}_{\text{sim}}$	FNO	Null	Regressor	FROST	$\text{FROST}_{\text{sim}}$
Cylinder	1.24	1.24	1.07	**1.04**	0.77	0.529	0.540	0.147	**0.090**	0.412
Contr. cyl.	0.91	0.94	0.93	0.93	**0.96**	0.166	0.107	0.136	0.129	**0.074**
FSI	0.95	0.98	0.95	**0.96**	0.96	0.096	0.046	0.106	**0.071**	0.085
Public foil	0.86	1.00	0.75	**0.81**	0.74	0.262	0.003	0.429	**0.348**	0.445
Combustion	1.01	1.02	1.04	**1.02**	0.61	0.030	0.038	0.077	**0.043**	0.628

Across systems where the forecaster has a clear fluctuation-amplitude mismatch, FROST moves the members toward the measured fluctuation statistics (Table 6).

The residual is neither deterministic bias nor unstructured noise. Matched controls in Appendix F.2 show that deterministic residual regression captures substantial bias on public foil but provides little or no benefit on cylinder and combustion, while variance-matched Gaussian noise increases rollout error on four of five systems. FROST improves both accuracy and CRPS over these controls on every system; a PDE-Refiner-style residual denoiser is also consistently worse.

5 Related work

Sim-to-real adaptation of learned forecasters. The standard sim-to-real approach is to adapt the forecaster itself, by pretraining on simulation, and then updating its parameters using real measurements, often with constraints designed to preserve useful simulation structure (Hu et al., 2026; Zhou et al., 2026; Kirkpatrick et al., 2017; Li et al., 2018; Kumar et al., 2022; Setinek et al., 2025). FROST takes a different view. It leaves the forecaster fixed and models the discrepancy between its forecast and the measured future in prediction space. This is closer in spirit to classical model-discrepancy and multi-fidelity methods, where the gap between a lower-fidelity predictor and reality is treated as a separate object to be learned (Kennedy & O'Hagan, 2001; Peherstorfer et al., 2018; Lu et al., 2022). The distinction is what the discrepancy is conditioned on and what its spread represents: Kennedy & O'Hagan (2001) place a Gaussian-process prior on it over the model inputs, so its uncertainty is about a single unknown correction function, whereas FROST learns the conditional law of a high-dimensional residual field given the measured history and the forecast, whose spread is the variability of the measured future that the history leaves undetermined.

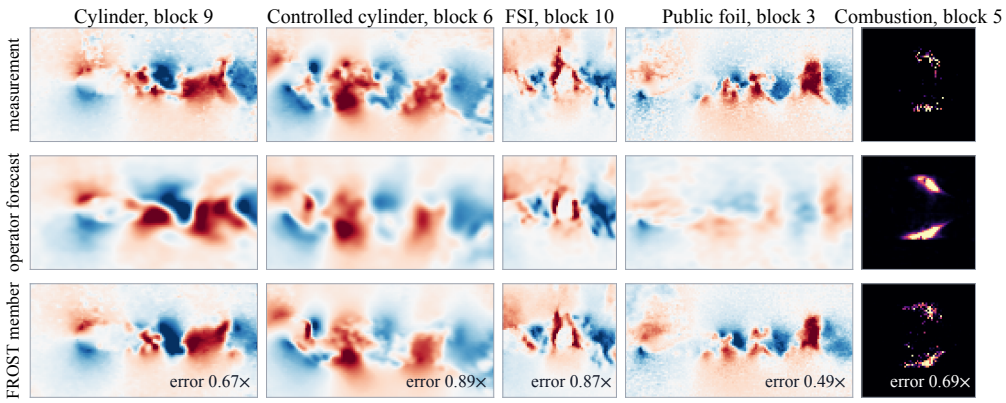

Figure 3: **FROST members recover coherent structure smoothed or misplaced by the operator forecast.** One selected test frame per system shows the measurement, operator forecast, and a FROST member; the annotation reports the member's mean absolute error relative to the forecast's. Frames are selected to illustrate the correction clearly; quantitative member statistics over the full evaluation set are reported in Table 6.

Generative correction of deterministic forecasts. Recent work in weather and PDE forecasting has shown that a stochastic model layered on top of a deterministic predictor can recover variability that point forecasts average away (Mardani et al., 2025; Couairon et al., 2026; Fotiadis et al., 2025; Oommen et al., 2025; Park et al., 2025; Bhola & Duraisamy, 2025; Chen et al., 2026; Li et al., 2024; Cheon, 2026; Lippe et al., 2023). FROST brings this idea to a different regime of scarce, partial, and noisy laboratory measurements, with a base forecaster that may already have been adapted to the same real data. The remaining error is therefore not simply unresolved simulation detail, but a forecast-dependent sim-to-real discrepancy. A complementary approach is to make the forecaster itself probabilistic (Diaconu et al., 2026; Rühling Cachay et al., 2026; Bülte et al., 2025); FROST instead separates the deterministic forecaster from the stochastic correction.

Stochastic interpolants. We parameterize the conditional residual using stochastic interpolants (Albergo & Vanden-Eijnden, 2023; Albergo et al., 2025), which unify flow- and diffusion-based transport and admit both probability-flow ODE and stochastic sampling (Lipman et al., 2023; Liu et al., 2023b; Song et al., 2021; Ma et al., 2024). Related bridge and coupling constructions provide alternative ways to anchor the transport to a forecast (Albergo et al., 2024; Chen et al., 2024; Liu et al., 2023a; Zhou et al., 2024; Shi et al., 2023). Our low-level contribution is use of generative transport as a prediction-space interface for sim-to-real adaptation around a frozen forecaster.

6 Conclusion

Real-data fine-tuning can leave structured, forecast-dependent residuals and mismatched fluctuation statistics. FROST freezes the base forecaster and learns the conditional distribution of its remaining discrepancy from real measurements; across five RealPDEBench systems it improves accuracy and probabilistic skill over standard fine-tuning, applies across six forecaster architectures, and adapts a simulation-only operator without updating its weights. Together, these results establish stochastic prediction-space correction as a complementary axis of sim-to-real adaptation. Predict the predictable, generate the rest.

FROST remains limited by its base forecast: upstream fine-tuning still improves the fluctuation fidelity of individual members (Table 23). It samples measured futures under partial, planar and noisy observation, not the unobserved three-dimensional state. Correctors are trained per system, and transfer across systems is untested. Inference adds six corrector evaluations per member and block, or thirty for the better-calibrated stochastic read (Appendix E.5).

Reproducibility Statement

The method is specified in Section 3 and Algorithm 1, and every setting of the single configuration used on all five systems is listed in Table 13 (Appendix C.6). All experiments use the public RealPDEBench data at a pinned revision, with the benchmark's own splits, evaluator and 320 test windows (Appendix A); the baseline implementations and their settings are in Appendix D. FROST and PhysGuard results are means over three training seeds, and every seed is reported in Appendix D.6. All tables and figures are generated from archived result files, with provenance recorded for each reported value.

AI Use Statement

In this work, we used generative AI tools for code debugging, creating figures and tables, searching and reviewing the literature, and manuscript editing. We have not used generative AI tools to generate synthetic data sets, formulate mathematical claims, or assist with proofs; translation and qualitative data analysis are not applicable to this work. We reviewed all AI-assisted work, and every reported number is regenerated from archived result files by our scripts. We take responsibility for the final content of this work.

References

Michael S. Albergo and Eric Vanden-Eijnden. Building Normalizing Flows with Stochastic Interpolants. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2209.15571.

Michael S. Albergo, Mark Goldstein, Nicholas M. Boffi, Rajesh Ranganath, and Eric Vanden-Eijnden. Stochastic interpolants with data-dependent couplings. In *International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2310.03725.

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic Interpolants: A Unifying Framework for Flows and Diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025. URL https://arxiv.org/abs/2303.08797.

Elad Ben-Zaken, Shauli Ravfogel, and Yoav Goldberg. BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models. In *Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pp. 1–9, 2022. URL https://arxiv.org/abs/2106.10199.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://arxiv.org/abs/1506.03099.

Sahil Bhola and Karthik Duraisamy. Residual-Augmented Flow Matching Operators for Probabilistic Partial Differential Equations, 2025. URL https://arxiv.org/abs/2512.12749.

Johannes Brandstetter, Daniel Worrall, and Max Welling. Message Passing Neural PDE Solvers. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2202.03376.

Christopher Bülte, Philipp Scholl, and Gitta Kutyniok. Probabilistic neural operators for functional uncertainty quantification. *Transactions on Machine Learning Research*, 2025. URL https://arxiv.org/abs/2502.12902.

Sipeng Chen, Junliang Liu, Hewei Tang, and Shibo Li. Multi-Fidelity Flow Matching: Cascaded Refinement of PDE Solutions, 2026. URL https://arxiv.org/abs/2605.16118.

Yifan Chen, Mark Goldstein, Mengjian Hua, Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Probabilistic Forecasting with Stochastic Interpolants and Föllmer Processes. In *International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2403.13724.

Minjong Cheon. Rescene: band-limited stochastic forcing turns a frozen neural weather operator into a climate emulator, 2026. URL https://arxiv.org/abs/2608.09971.

Özgün Çiçek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox, and Olaf Ronneberger. 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 424–432, 2016. doi: 10.1007/978-3-319-46723-8_49.

Guillaume Couairon, Renu Singh, Anastase Charantonis, Christian Lessig, and Claire Monteleoni. ArchesWeatherGen: Skillful and compute-efficient probabilistic weather forecasting with machine learning. *Science Advances*, 12(17):eadx2372, 2026. doi: 10.1126/sciadv.adx2372. URL https://arxiv.org/abs/2412.12971.

Prafulla Dhariwal and Alex Nichol. Diffusion Models Beat GANs on Image Synthesis. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://arxiv.org/abs/2105.05233.

Cristiana Diaconu, Miles Cranmer, Richard E. Turner, Tanya Marwah, and Payel Mukhopadhyay. Probabilistic Retrofitting of Learned Simulators. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2603.01949.

Claudia Drygala, Edmund Ross, Mohammad Sharifi Ghazijahani, Christian Cierpka, Francesca di Mare, and Hanno Gottschalk. Comparison of Generative Learning Methods for Turbulence Surrogates, 2024. URL https://arxiv.org/abs/2411.16417.

C. A. T. Ferro. Fair scores for ensemble forecasts. *Quarterly Journal of the Royal Meteorological Society*, 140(683):1917–1923, 2014. doi: 10.1002/qj.2270.

V. Fortin, M. Abaza, F. Anctil, and R. Turcotte. Why Should Ensemble Spread Match the RMSE of the Ensemble Mean? *Journal of Hydrometeorology*, 15(4):1708–1713, 2014. doi: 10.1175/jhm-d-14-0008.1.

Stathi Fotiadis, Noah Brenowitz, Tomas Geffner, Yair Cohen, Michael Pritchard, Arash Vahdat, and Morteza Mardani. Adaptive Flow Matching for Resolving Small-Scale Physics. In *International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2410.19814.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean Flows for One-step Generative Modeling. In *Advances in Neural Information Processing Systems*, volume 38, 2025a. URL https://arxiv.org/abs/2505.13447.

Zhengyang Geng, Yiyang Lu, Zongze Wu, Eli Shechtman, J. Zico Kolter, and Kaiming He. Improved Mean Flows: On the Challenges of Fastforward Generative Models, 2025b. URL https://arxiv.org/abs/2512.02012.

Tilmann Gneiting and Adrian E. Raftery. Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437.

Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E. Raftery. Probabilistic Forecasts, Calibration and Sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2):243–268, 2007. doi: 10.1111/j.1467-9868.2007.00587.x.

Sam F. Greenbury, Radka Jersakova, Paolo Conti, Marjan Famili, Christopher Iliffe Sprague, Edwin Brown, and Jason D. McEwen. Reliability of Probabilistic Emulation of Physical Systems, 2026. URL https://arxiv.org/abs/2606.12997.

Zhongkai Hao, Chang Su, Songming Liu, Julius Berner, Chengyang Ying, Hang Su, Anima Anandkumar, Jian Song, and Jun Zhu. DPOT: Auto-Regressive Denoising Operator Transformer for Large-Scale PDE Pre-Training. In *International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2403.03542.

Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient Foundation Models for PDEs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 72525–72624, 2024. URL https://arxiv.org/abs/2405.19101.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2106.09685.

Peiyan Hu, Haodong Feng, Hongyuan Liu, Tongtong Yan, Wenhao Deng, Tianrun Gao, Rong Zheng, Haoren Zheng, Chenglei Yu, Chuanrui Wang, Kaiwen Li, Zhi-Ming Ma, Dezhi Zhou, Xingcai Lu, Dixia Fan, and Tailin Wu. RealPDEBench: A Benchmark for Complex Physical Systems with Real-World Data. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2601.01829.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self Forcing: Bridging the Train-Test Gap in Autoregressive Video Diffusion. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2506.08009.

Shreya Jha, Timo Schorlepp, Nicholas Geissler, Jules Berman, and Benjamin Peherstorfer. First-Order Trajectory Matching: Fast Ensemble Predictions of Chaotic, Turbulent, Stochastic Systems, 2026. URL https://arxiv.org/abs/2606.11138.

Marc C. Kennedy and Anthony O'Hagan. Bayesian Calibration of Computer Models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001. doi: 10.1111/1467-9868.00294.

Yujin Kim, Nidhi Soma, and Sarah Dean. Mind the Residual Gap: Probabilistic Downscaling under Real-World Bias, 2026. URL https://arxiv.org/abs/2606.30821.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114.

Felix Koehler, Simon Niedermayr, Rüdiger Westermann, and Nils Thuerey. APEBench: A Benchmark for Autoregressive Neural Emulators of PDEs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 120252–120310, 2024. URL https://arxiv.org/abs/2411.00180.

Georg Kohl, Li-Wei Chen, and Nils Thuerey. Benchmarking Autoregressive Conditional Diffusion Models for Turbulent Flow Simulation. *Neural Networks*, 199:108641, 2026. doi: 10.1016/j.neunet.2026.108641. URL https://arxiv.org/abs/2309.01745.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-Tuning can Distort Pretrained Features and Underperform Out-of-Distribution. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2202.10054.

Lizao Li, Rob Carver, Ignacio Lopez-Gomez, Fei Sha, and John Anderson. Generative emulation of weather forecast ensembles with diffusion models. *Science Advances*, 10(13):eadk4489, 2024. doi: 10.1126/sciadv.adk4489. URL https://arxiv.org/abs/2306.14066.

Xuhong Li, Yves Grandvalet, and Franck Davoine. Explicit Inductive Bias for Transfer Learning with Convolutional Networks. In *International Conference on Machine Learning*, 2018. URL https://arxiv.org/abs/1802.01483.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier Neural Operator for Parametric Partial Differential Equations. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.08895.

Zongyi Li, Miguel Liu-Schiaffini, Nikola Kovachki, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Learning Chaotic Dynamics in Dissipative Systems. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://arxiv.org/abs/2106.06898.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.02747.

Phillip Lippe, Bastiaan S. Veeling, Paris Perdikaris, Richard E. Turner, and Johannes Brandstetter. PDE-Refiner: Achieving Accurate Long Rollouts with Neural PDE Solvers. In *Advances in Neural Information Processing Systems*, volume 36, pp. 67398–67433, 2023. URL https://arxiv.org/abs/2308.05732.

Bjoern List, Li-Wei Chen, Kartik Bali, and Nils Thuerey. Differentiability in Unrolled Training of Neural Physics Simulators on Transient Dynamics. *Computer Methods in Applied Mechanics and Engineering*, 433:117441, 2025. doi: 10.1016/j.cma.2024.117441. URL https://arxiv.org/abs/2402.12971.

Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A. Theodorou, Weili Nie, and Anima Anandkumar. I^2SB: Image-to-Image Schrödinger Bridge. In *International Conference on Machine Learning*, 2023a. URL https://arxiv.org/abs/2302.05872.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In *International Conference on Learning Representations*, 2023b. URL https://arxiv.org/abs/2209.03003.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

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

Lu Lu, Raphael Pestourie, Steven G. Johnson, and Giuseppe Romano. Multifidelity deep neural operators for efficient learning of partial differential equations with application to fast inverse design of nanoscale heat transport. *Physical Review Research*, 4(2):023210, 2022. doi: 10.1103/physrevresearch.4.023210. URL https://arxiv.org/abs/2204.06684.

Nanye Ma, Mark Goldstein, Michael S. Albergo, Nicholas M. Boffi, Eric Vanden-Eijnden, and Saining Xie. SiT: Exploring Flow and Diffusion-based Generative Models with Scalable Interpolant Transformers. In *European Conference on Computer Vision*, pp. 23–40, 2024. URL https://arxiv.org/abs/2401.08740.

Morteza Mardani, Noah Brenowitz, Yair Cohen, Jaideep Pathak, Chieh-Yu Chen, Cheng-Chin Liu, Arash Vahdat, Mohammad Amin Nabian, Tao Ge, Akshay Subramaniam, Karthik Kashinath, Jan Kautz, and Mike Pritchard. Residual Corrective Diffusion Modeling for Km-scale Atmospheric Downscaling. *Communications Earth & Environment*, 6(1):124, 2025. doi: 10.1038/s43247-025-02042-5. URL https://arxiv.org/abs/2309.15214.

Michael McCabe, Peter Harrington, Shashank Subramanian, and Jed Brown. Towards Stability of Autoregressive Neural Operators. *Transactions on Machine Learning Research*, 2023. URL https://arxiv.org/abs/2306.10619.

Michael McCabe, Bruno Régaldo-Saint Blancard, Liam Holden Parker, Ruben Ohana, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Siavash Golkar, Geraud Krawezik, Francois Lanusse, Mariel Pettee, Tiberiu Tesileanu, Kyunghyun Cho, and Shirley Ho. Multiple Physics Pretraining for Spatiotemporal Surrogate Models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2310.02994.

Nikolaj T. Mücke and Benjamin Sanderse. Physics-aware Generative Models for Turbulent Fluid Flows through Energy-Consistent Stochastic Interpolants. *Computers & Fluids*, 314:107098, 2026. doi: 10.1016/j.compfluid.2026.107098. URL https://arxiv.org/abs/2504.05852.

Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina J. Agocs, Miguel Beneitez, Marsha Berger, Blakesley Burkhart, Keaton Burns, Stuart B. Dalziel, Drummond B. Fielding, Daniel Fortunato, Jared A. Goldberg, Keiya Hirashima, Yan-Fei Jiang, Rich R. Kerswell, Suryanarayana Maddu, Jonah Miller, Payel Mukhopadhyay, Stefan S. Nixon, Jeff Shen, Romain Watteaux, Bruno Régaldo-Saint Blancard, François Rozet, Liam H. Parker, Miles Cranmer, and Shirley Ho. The Well: a Large-Scale Collection of Diverse Physics Simulations for Machine Learning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 44989–45037, 2024. URL https://arxiv.org/abs/2412.00568.

Vivek Oommen, Aniruddha Bora, Zhen Zhang, and George Em Karniadakis. Integrating Neural Operators with Diffusion Models Improves Spectral Representation in Turbulence Modeling. *Proceedings of the Royal Society A*, 481(2309):20240819, 2025. doi: 10.1098/rspa.2024.0819. URL https://arxiv.org/abs/2409.08477.

Jaewan Park, Farid Ahmed, Kazuma Kobayashi, Seid Koric, Syed Bahauddin Alam, Iwona Jasiuk, and Diab Abueidda. Bridging Sequential Deep Operator Network and Video Diffusion: Residual Refinement of Spatio-Temporal PDE Solutions, 2025. URL https://arxiv.org/abs/2507.06133.

Benjamin Peherstorfer, Karen Willcox, and Max Gunzburger. Survey of Multifidelity Methods in Uncertainty Propagation, Inference, and Optimization. *SIAM Review*, 60(3):550–591, 2018. doi: 10.1137/16m1082469.

Sebastian Pfister, Benjamin Holzschuh, and Nils Thuerey. StocBench: A Benchmark for Generative Modeling of Stochastic Dynamics, 2026. URL https://arxiv.org/abs/2608.22309.

Markus Raffel, Christian E. Willert, Fulvio Scarano, Christian J. Kähler, Steve T. Wereley, and Jürgen Kompenhans. *Particle Image Velocimetry: A Practical Guide*. Springer, 3 edition, 2018. doi: 10.1007/978-3-319-68852-7.

Bogdan Raonić, Roberto Molinaro, Tim De Ryck, Tobias Rohner, Francesca Bartolucci, Rima Alaifari, Siddhartha Mishra, and Emmanuel de Bézenac. Convolutional Neural Operators for robust and accurate learning of PDEs. In *Advances in Neural Information Processing Systems*, volume 36, pp. 77187–77200, 2023. doi: 10.52202/075280-3376.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241, 2015. doi: 10.1007/978-3-319-24574-4_28.

Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In *International Conference on Artificial Intelligence and Statistics*, 2011. URL https://arxiv.org/abs/1011.0686.

Salva Rühling Cachay, Duncan Watson-Parris, and Rose Yu. U-Cast: A Surprisingly Simple and Efficient Frontier Probabilistic AI Weather Forecaster. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2604.09041.

Anish Sambamurthy and Ashesh Chattopadhyay. Lazy Diffusion: Mitigating spectral collapse in generative diffusion-based stable autoregressive emulation of turbulent flows, 2025. URL https://arxiv.org/abs/2512.09572.

A. Sciacchitano. Uncertainty quantification in particle image velocimetry. *Measurement Science and Technology*, 30(9):092001, 2019. doi: 10.1088/1361-6501/ab1db8.

Paul Setinek, Gianluca Galletti, Thomas Gross, Dominik Schnürer, Johannes Brandstetter, and Werner Zellinger. SIMSHIFT: A Benchmark for Adapting Neural Surrogates to Distribution Shifts, 2025. URL https://arxiv.org/abs/2506.12007.

Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion Schrödinger Bridge Matching. In *Advances in Neural Information Processing Systems*, volume 36, pp. 62183–62223, 2023. URL https://arxiv.org/abs/2303.16852.

Aliaksandra Shysheya, Cristiana Diaconu, Federico Bergamin, Paris Perdikaris, José Miguel Hernández-Lobato, Richard E. Turner, and Emile Mathieu. On conditional diffusion models for PDE simulations. In *Advances in Neural Information Processing Systems*, volume 37, pp. 23246–23300, 2024. doi: 10.52202/079017-0732.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2011.13456.

Baochen Sun and Kate Saenko. Deep CORAL: Correlation Alignment for Deep Domain Adaptation. In *European Conference on Computer Vision Workshops*, pp. 443–450, 2016. URL https://arxiv.org/abs/1607.01719.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Dan MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. PDEBench: An Extensive Benchmark for Scientific Machine Learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 1596–1611, 2022. URL https://arxiv.org/abs/2210.07182.

Ronald J. Williams and David Zipser. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation*, 1(2):270–280, 1989. doi: 10.1162/neco.1989.1.2.270.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A Fast Transformer Solver for PDEs on General Geometries. In *International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2402.02366.

Fan Xu, Yuan Gao, Kun Wang, Rui Su, Fenghua Ling, Hao Wu, and Wanli Ouyang. Tyche: One Step Flow for Efficient Probabilistic Weather Forecasting, 2026. URL https://arxiv.org/abs/2605.06916.

Tianyue Yang and Xiao Xue. Autoregressive One-Step Generative Modeling for Dynamical System Forecasting, 2026. URL https://arxiv.org/abs/2605.05540.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding Conditional Control to Text-to-Image Diffusion Models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3813–3824, 2023. doi: 10.1109/ICCV51070.2023.00355.

Anthony Zhou, Alexander Wikner, Amaury Lancelin, Pedram Hassanzadeh, and Amir Barati Farimani. Reframing Generative Models for Physical Systems using Stochastic Interpolants, 2025. URL https://arxiv.org/abs/2509.26282.

Changjian Zhou, Junfeng Fang, Negin Yousefpour, Peng Wu, Bin Yan, and Guillermo A Narsilio. PhysGuard: Fisher-Guided Gradient Projection for Sim-to-Real Neural PDE Surrogates, 2026. URL https://arxiv.org/abs/2606.16602.

Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising Diffusion Bridge Models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.16948.

APPENDIX CONTENTS

A Task, data and evaluation **18**
A.1 Data and checkpoint revisions . . . 18
A.2 What the parameter labels contain . . . 18
A.3 Evaluation windows and uncertainty in the estimate . . . 19
A.4 Metrics and aggregation . . . 19

B Diagnosis in full (Section 2.2) **21**
B.1 Error decomposition and the diagnostic rollouts . . . 21
B.2 Forecast interventions: the gap is forecast-relative, and its sign is not universal . . . 21
B.3 Residual statistics and the measurement-noise floor . . . 23

C Method details (Section 3) **25**
C.1 Training and sampling algorithm . . . 25
C.2 Architecture and optimization . . . 25
C.3 The known past and forecast condition . . . 25
C.4 What the forecast condition supplies . . . 26
C.5 Residual and bridge objectives, source noise, and samplers . . . 26
C.6 Parameterization, samplers and the main recipe . . . 26
C.7 Other instantiations of prediction-space adaptation . . . 27
C.8 Where the gain comes from: the forecast, not the wiring . . . 28

D Baselines (Section 4.1) **30**
D.1 PhysGuard: implementation and comparison scope . . . 30
D.2 Pushforward fine-tuning . . . 31
D.3 Further adaptations of the same operator . . . 31
D.4 Probabilistic baselines . . . 32
D.5 Probabilistic retrofitting of the operator . . . 32
D.6 PhysGuard, every adaptation seed, and every seed of the main table . . . 33

E Main results in full (Section 4.2) **35**
E.1 Held-out parameter groups and paired intervals . . . 35
E.2 The pretrained operator and real-data scaling . . . 35
E.3 Transfer between forecasters . . . 35
E.4 Transfer: residual-scale controls and CRPS . . . 38
E.5 Skill and calibration against the sampling budget . . . 39
E.6 Members and steps: the sampling ladder . . . 39

F Structure of the residual (Section 4.4) **42**
F.1 Members are flows: full member statistics . . . 42

F.2 The matched correction controls 43

G Additional experiments **45**

G.1 Where the remaining error lives: autoregressive input shift 45

G.2 Generated-prefix training 46

G.3 Cheap sampling and MeanFlow 46

G.4 Twice the horizon 47

G.5 Input-noise augmentation 47

G.6 Backbone port and architectural knobs 47

G.7 Competition foil in full 47

Unless a caption says otherwise, every table in this appendix is scored by the evaluator of Section 4.1 on the benchmark's 320 test windows and ten rollout blocks, reports the mean over three corrector seeds, bolds the best entry within the 0.0018 run-to-run floor, and greys a row or column read from fewer than three seeds or from a window subset, which is never bolded.

A TASK, DATA AND EVALUATION

A.1 DATA AND CHECKPOINT REVISIONS

The principal experiments use RealPDEBench data revision `2c72295abbc99f60333c66062449c9eb36db698e`. The upstream model release is pinned to `4f7319ae6efc72c84a5d1e53750240f489136bcf`; the vendored benchmark implementation is pinned to `62f4c80ab17f78933d046f2b038531dbc6a478a0`. Every result records these identifiers.

Table 7: **The five laboratory systems.** Every system is scored on its measured channels only, over ten autoregressive blocks (200 predicted frames, 100 on controlled cylinder), on the 320 test windows the benchmark's evaluator selects; the parameter groups are the benchmark's seen, in-range and out-of-range held-out labels.

System	Measured channels	Grid	Block (hist. + future)	Test windows	Parameter groups
Cylinder	PIV, u, v	64×128	20 + 20	320	seen / in / out
Controlled cylinder	PIV, u, v	64×128	10 + 10	320	seen / in / out
FSI	PIV, u, v	64×64	20 + 20	320	seen / in / out
Public foil	PIV, u, v	64×128	20 + 20	320	seen / in / out
Combustion	chemiluminescence, 1 ch.	64×64	20 + 20	320	seen / in / out

Table 7 lists the five systems. The initial history is twenty frames for all systems except controlled cylinder, where it is ten; frames are not interchangeable physical time across systems. PIV carries measurement noise whose spectral plateau accounts for 0.2–4% of the fluctuation variance on these systems (Appendix B.3); generative models have been fitted to real PIV before, as generators of flow snapshots rather than forecasters (Drygala et al., 2024).

The public foil experiment uses the benchmark checkpoint family, including DPOT-S rather than the larger DPOT-L reported in some benchmark comparisons. Its uncorrected predictions are evaluated through the same adapter and metric pipeline as the corrected predictions. No score is copied from a published table with a different model size or evaluation convention. The competition foil release is a separate dataset with its own resolution and protocol, and its results are kept apart from public foil.

A.2 WHAT THE PARAMETER LABELS CONTAIN

The cylinder staging contains 72 seen-parameter trajectories and two groups of ten held-out trajectories. The seen Reynolds-number range is 2062–11812. The group named `in_dist` contains Reynolds numbers 3093, 3656, 4125, 4968, 5062, 5156, 5906, 7031, 8437, and 9656. The group named `out_dist` contains 1781, 1875, 2156, 2531, 11062, 11156, 11250, 11343, 11625, and 11718, of which 1781 and 1875 lie below the seen range. The held-out groups are read as held-out parameter values, identified by the Reynolds numbers of their trajectories.

The benchmark's real training split, on which the fine-tuned operators and every corrector are trained, draws its 9,063 windows from the 72 seen trajectories only; the 20 held-out trajectories appear only in the validation and test splits. The same holds on controlled cylinder and public foil. On FSI the benchmark's parameter files list six trajectories, three runs at each of two parameter settings, both among the held-out and among the seen parameters, and its training split uses them; its evaluator scores them in the seen group, so the 84 windows of FSI's held-out row in Table 4 come from trajectories that contribute no training window. The simulation used for pretraining covers every parameter value, so the held-out groups are held out from the measurements, not from the simulator. The data-fraction study of Appendix E.2 draws its subsets over the 92 staged cylinder trajectories, of which the 72 seen ones carry training windows.

Within the seen group the benchmark's split interleaves window starts along each trajectory, as its protocol specifies: a training sample spans the history and one future block, a test sample the history and ten future blocks. The held-out parameter groups separate whole trajectories, and Table 4 reports them separately. Because the benchmark's normalized metrics are recomputed on each subset, the pooled score need not equal a window-count-weighted average of the partition scores.

A.3 Evaluation windows and uncertainty in the estimate

The benchmark's evaluator selects $B = 40 \times 8 = 320$ windows by taking every $\lfloor N/B \rfloor$-th item of an indexed split of size N; this is the window set behind every main-table number. The thirty-step stochastic and sixteen-member reads use 128 of these windows and are marked wherever they appear.

The three corrector seeds are independent training initializations around one fixed operator checkpoint per system, and the sampling seed follows the training seed, so the spread across seeds includes the sampling realization. The run-to-run floor of 0.0018 is the range, largest minus smallest, of the cylinder rollout error over repeated trainings of one configuration; entries that differ by less than it are read as tied. The trajectory-resampled intervals of Appendix E.1 quantify a different uncertainty, that of the test set around a fixed checkpoint, and resample whole trajectories, the independent unit.

A.4 Metrics and aggregation

A.4.1 Point prediction and sample prediction

Let $\hat{y}^{(1)}, \ldots, \hat{y}^{(K)}$ be the predicted members and $\bar{y} = K^{-1} \sum_k \hat{y}^{(k)}$ their mean. The main point-prediction score evaluates $\bar{y}$ against the measured target with the benchmark relative-L_2 implementation. The member score evaluates each $\hat{y}^{(k)}$ and then averages the resulting scores. The benchmark evaluator uses a fixed metric chunk size of 64, distinct from the batch size used to generate predictions. We preserve that convention in the headline scores; per-window alternatives are named separately where they appear.

A.4.2 Fair CRPS

For one scalar observation y and $K > 1$ independently sampled members, the estimator is

$$\widehat{\mathrm{CRPS}}_{\mathrm{fair}} = \frac{1}{K} \sum_{k=1}^{K} |\hat{y}^{(k)} - y| - \frac{1}{2K(K-1)} \sum_{j \neq k} |\hat{y}^{(j)} - \hat{y}^{(k)}|. \tag{8}$$

The implementation sums unordered pairs once and divides their sum by $K(K-1)$, which is the same expression. It averages scalar scores over the scored field coordinates and windows. For a deterministic forecast, we report absolute error, the CRPS of its point-mass distribution. Treating the ensemble as a fixed empirical distribution would divide the pairwise term by K^2 instead; every CRPS in the paper is the fair estimator above (Ferro, 2014).

Fair CRPS concerns marginal distributions at the scored coordinates. It is insensitive to some errors in spatial or temporal dependence. In particular, an ensemble can have useful marginal uncertainty while producing implausible joint fields. The member statistics below score the joint structure that the CRPS does not.

A.4.3 Fluctuation statistics

For each predicted sequence, subtract its own temporal mean: $u' = u - \langle u \rangle_t$. The measured velocity-fluctuation energy is

$$E(u) = \tfrac{1}{2} \langle u_1'^2 + u_2'^2 \rangle_{t,x}, \qquad \epsilon_E = \frac{|E(\hat{u}) - E(u)|}{E(u)}. \tag{9}$$

The amplitude ratio is the ratio of root-mean-square fluctuations, with ideal value one. The spectrum diagnostic compares radially binned spatial Fourier energy magnitudes, rather than complex coefficients; this makes it less sensitive to a spatial phase shift. Each statistic is computed for each member before averaging over members. It is not computed only on the smoother ensemble mean.

For combustion, applying the same variance formula to the single observed intensity channel yields an intensity-fluctuation statistic. It has no interpretation as measured kinetic energy. For fluid systems, two measured components are still not full three-dimensional kinetic energy. These are finite-window diagnostics of the available measurements.

A.4.4 SPREAD AND COVERAGE

The spread/skill statistic uses the square root of the mean unbiased ensemble variance divided by the root-mean-square error of the ensemble mean, multiplied by $\sqrt{(K+1)/K}$. Averaging local standard deviations first would be a different estimator and can produce a substantially smaller value in spatially heterogeneous fields.

Coverage is that of the Gaussian-fitted 90% interval built from the members' mean and standard deviation at each scored coordinate. With four members an empirical quantile interval is held well below the nominal level by the ensemble size alone, so it is not used as a calibration statistic.

B DIAGNOSIS IN FULL (SECTION 2.2)

B.1 ERROR DECOMPOSITION AND THE DIAGNOSTIC ROLLOUTS

The error decomposition. Split a block into its temporal mean and fluctuation, $y = \bar{y} + y'$, and write $\varphi = \|y'\|^2/\|y\|^2$ for the fluctuation share of the measured energy, $\rho = \|g'\|/\|y'\|$ for the amplitude ratio of the forecast's fluctuation to the measurement's, and c for their alignment. Then

$$\frac{\|y-g\|^2}{\|y\|^2} = \frac{\|\bar{y}-\bar{g}\|^2}{\|y\|^2} + \varphi\left(1+\rho^2-2\rho c\right), \tag{10}$$

so a forecast that is uncertain where a fluctuation will occur lowers its amplitude under squared error, the conditional-mean argument behind residual diffusion (Mardani et al., 2025), and adding an independent perturbation raises the expected squared error by that perturbation's own variance.

Derivation. For a block of T frames write $\bar{y} = \frac{1}{T}\sum_t y_t$ and $y'_t = y_t - \bar{y}$, and likewise for the forecast g; norms and inner products run over frames, pixels and channels. Then $y - g = (\bar{y}-\bar{g}) + (y'-g')$, and the cross term vanishes because $\sum_t (y'_t - g'_t) = 0$ at every pixel:

$$\|y-g\|^2 = \|\bar{y}-\bar{g}\|^2 + \|y'-g'\|^2, \qquad \|y'-g'\|^2 = \|y'\|^2\big(1+\rho^2-2\rho c\big),$$

with $\rho = \|g'\|/\|y'\|$ and $c = \langle g', y'\rangle/(\|g'\|\|y'\|)$. Dividing by $\|y\|^2$ and writing $\varphi = \|y'\|^2/\|y\|^2$ gives Eq. (10). Three facts follow. (i) A forecast with the true block mean and $g' = 0$ has relative error $\sqrt{\varphi}$. (ii) For fixed c the bracket is minimized at $\rho^\star = c$, with value $1 - c^2$; a squared-error-optimal amplitude therefore equals the alignment, and tends to zero as alignment is lost. (iii) Let n have zero block mean, be independent of y and g, and have $\mathbb{E}\|n\|^2 = \nu\|y'\|^2$. Replacing g' by $g' + n$ gives $\mathbb{E}\|g'+n\|^2 = (\rho^2+\nu)\|y'\|^2$ and $\mathbb{E}\langle g'+n, y'\rangle = \langle g', y'\rangle$, so the bracket becomes $1+\rho^2+\nu-2\rho c$: the squared fluctuation error rises by exactly ν, for any covariance of n. Matching the measured energy ($\rho^2 + \nu = 1$) costs $\nu = 1 - \rho^2$. With the competition-foil training fit, $\rho^2 = 9.813\times10^{-5}/1.830\times10^{-4} = 0.536$ and the residual-to-fluctuation variance ratio is $1.722/1.830 = 0.941$, so $c \approx (1 + 0.536 - 0.941)/(2\sqrt{0.536}) = 0.41$, whereas perfect alignment would leave $(1-\rho)^2 = 0.072$. The logged residual variance is taken per pixel over windows and frames and therefore includes between-window bias, which makes 0.41 a lower bound on c.

Diagnostic rollouts. Table 8 and Figure 2(a,c) use diagnostic rollouts on the first 48 test windows (six batches of eight), seed zero, in physical units, with each frozen forecaster run free, feeding its own output back as history; Figure 2(b) pools the residual of the operator forecast over 800 frames per system (400 for competition foil), five per block of the first sixteen test windows. Fluctuations are taken about the block's temporal mean. The spectrum sums $|\mathrm{FFT}|^2$ over integer radial wavenumber bins and averages over windows, frames and channels, so bin sums are proportional to energy; the last bin collects every higher wavenumber. Relative error here is a per-window mean without metric chunking. These numbers are comparable with one another and are not the 320-window benchmark scores.

B.2 FORECAST INTERVENTIONS: THE GAP IS FORECAST-RELATIVE, AND ITS SIGN IS NOT UNIVERSAL

The residual is organized around a particular forecast, which we test by intervening on the forecast supplied to the corrector (Figure 4). On competition foil, removing the forecast entirely and training the same generative architecture from the measured history alone produces rollout errors of 0.560 and 0.573 across two seeds, more than six times the uncorrected forecaster's 0.0903; increasing the sampler from six to thirty steps still leaves errors of 0.37–0.40. Providing the forecast reduces the direct generator to 0.0848–0.0855, while residual correction reaches 0.0845 ± 0.0005. The useful information is not summarized by global forecast statistics: replacing the full forecast with its temporal mean and RMS gives rollout error 0.0915 (one seed), worse than leaving the forecaster uncorrected, and degrades CRPS and fluctuation-energy error likewise (Table 9). A temporal intervention makes the dependence sharper. On cylinder, cyclically rotating the forecast by five or ten frames preserves its marginal mean, amplitude and spectrum while changing *when* predicted structures occur; the corrected rollout error rises from 0.1570 (seed zero) to 0.3448 and 0.3245, more than twice the uncorrected forecaster's 0.1632, while CRPS rises from 0.0176 to 0.0295 and 0.0282

Table 8: Free-running deterministic forecasts in the 48-window diagnostic rollouts. φ: fluctuation share of measured field energy at block one; $\sqrt{\varphi}$: relative error of an oracle that predicts the true block mean and no fluctuation. Error, amplitude ratio ρ and total fluctuation-energy ratio (the spectrum summed over bins) are given at the first and last block; the final column splits the last-block energy ratio into thirds of the wavenumber range (low / mid / high). Cylinder rows other than FNO are released fine-tuned checkpoints. Amplitude ratios on these 48-window subsets differ from the full 320-window evaluation that the text quotes (FNO: 1.236 on cylinder, 0.859 on public foil).

					Rel. error		ρ		Energy ratio		Last block by band
System	Forecaster	Blocks	φ (%)	$\sqrt{\varphi}$	first	last	first	last	first	last	low / mid / high
Comp. foil	FNO (organizer)	5	0.74	0.086	0.062	0.100	0.78	0.36	0.61	0.12	0.14 / 0.09 / 0.09
Cylinder	FNO	10	1.26	0.112	0.063	0.173	0.89	0.94	0.80	0.90	1.08 / 0.12 / 0.10
Cylinder	DPOT-S	10	1.14	0.107	0.075	0.222	1.01	0.88	1.69	0.60	0.75 / 0.12 / 0.10
Cylinder	U-Net	10	1.14	0.107	0.072	0.231	0.83	0.81	0.60	0.51	0.65 / 0.06 / 0.04
Cylinder	DeepONet	10	1.14	0.107	0.151	0.288	0.67	0.37	0.57	0.13	0.17 / 0.01 / 0.01
Cylinder	Transolver	10	1.14	0.107	0.171	0.422	0.75	1.63	0.44	2.19	0.04 / 0.86 / 39
Contr. cyl.	FNO	10	0.40	0.064	0.070	0.221	0.84	0.62	0.70	0.39	0.51 / 0.08 / 0.04
FSI	FNO	10	0.47	0.068	0.086	0.298	0.85	0.69	0.72	0.47	0.71 / 0.25 / 0.05
Public foil	FNO	10	0.06	0.024	0.022	0.108	1.03	1.11	1.32	1.57	3.04 / 0.91 / 0.07
Combustion	FNO	10	60.29	0.776	0.527	0.903	0.78	1.03	0.59	1.19	1.32 / 0.35 / 0.09

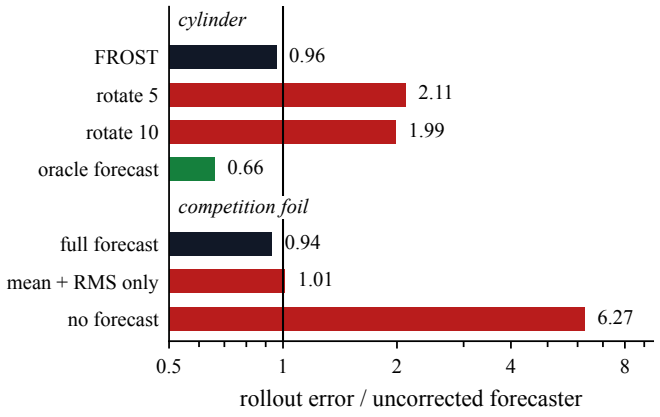

Figure 4: **The correction depends on the organized, time-resolved forecast.** Rollout error of the corrected model relative to the uncorrected forecaster when the forecast supplied to the corrector is intervened on. On cylinder, rotating the forecast in time, which keeps its mean, amplitude and spectrum, doubles the error, and a forecast computed from a history that includes true target frames (an oracle) lowers it; on competition foil the forecast's window statistics give no gain and no forecast is six times worse. Cylinder: seed zero, 320 windows; competition foil: 304 windows (Tables 9 and 10).

(Table 10). Conversely, conditioning on the prediction generated from a history advanced five true frames into the target block, an oracle that supplies future information, lowers the rollout error to 0.1079. The corrector depends on the organized, time-resolved prediction, and since different forecasters produce different organized predictions, the discrepancy is predictor-dependent rather than a noise law that can be detached from the forecaster.

The amplitude finding of the diagnosis also holds in the other direction. The fine-tuned cylinder FNO is over-energetic over the full test rollout, with amplitude ratio 1.236 and fluctuation-energy error 0.529, and its spectrum is wrong in both directions at once: excess energy at the largest scales beside the deficit at high wavenumbers (Figure 2a); PhysGuard adaptation amplifies the excess, reaching amplitude ratios of 1.51, 1.92 and 2.29 across its three adaptation seeds. Damping with a predetermined sign therefore does not describe the mismatch: a universal energy injection would help an under-energetic forecast and harm an over-energetic one, and the discrepancy itself must be modeled. On competition foil the forecast contains 0.54 of the measured fluctuation energy while

the residual variance is 0.94 of it; an attenuated but perfectly aligned copy of the future would leave far less, so much of the discrepancy lies in the organization of the future rather than its total energy. Some high-wavenumber discrepancy is expected from PIV measurement noise (Raffel et al., 2018; Sciacchitano, 2019); the estimated spectral noise plateau accounts for 0.2–4% of fluctuation variance (Table 12), so the diagnosis does not rest on reproducing that regime. "Stochastic" throughout refers to the forecasting problem conditional on the information available to the model, not to the governing equations: the history is finite and partially observed, boundary and forcing variables may be incomplete, the simulator omits unresolved dynamics, and distinct futures can be poorly phase-aligned while sharing their large-scale evolution.

Table 9: Competition foil: conditioning on the full forecast against conditioning on its window mean and RMS only. The statistics-only arm has one seed; both use $K = 4$ and ODE-6.

Forecast representation	n	Single	Rollout	CRPS	Fluct.
Full forecast (FROST)	3	0.0477 ± 0.0000	0.0845 ± 0.0005	0.00362 ± 0.00002	0.408 ± 0.016
Window mean and RMS only	1	0.0570	0.0915	0.00417	0.555

Table 10: Cylinder, seed zero: interventions on the forecast the corrector is conditioned on. Rotating the forecast in time keeps its mean, amplitude and spectrum and changes when its structures occur; the advanced-history forecast is computed from a history that includes true target frames, an oracle rather than a forecasting method.

Forecast supplied to the corrector	Rollout	CRPS	Energy err.
Operator forecast (FROST)	0.1570	0.01761	0.018
Rotated by five frames	0.3448	0.02953	0.272
Rotated by ten frames	0.3245	0.02824	0.557
History advanced five true frames (oracle)	0.1079	0.01124	0.285

A temporal rotation changes when the forecast's structures occur while preserving its marginal statistics, and the error increase establishes the corrector's sensitivity to the forecast's temporal organization. Future-frame insertion supplies privileged information, so it is an oracle probe of what additional alignment could help, not a forecasting method. Each intervention changes the condition and, in the residual model, the anchor together.

B.3 Residual statistics and the measurement-noise floor

The residual the corrector learns is the measurement minus the operator's forecast from the measured history. Table 11 gives its statistics on competition foil, the release whose training-residual fit supplies the alignment estimate of Appendix B.1, and Table 12 gives the part of the measured spectrum attributable to PIV noise. The organizer's FNO, run free over 304 competition-foil test windows and five blocks, has relative error 0.0903, fluctuation amplitude ratio 0.543 and fluctuation-energy error 0.706: accurate pointwise and far from the measured fluctuation statistics. Its training residual is correlated at lag one with coefficient 0.87 in time and 0.73 and 0.83 along the two spatial axes, where an independent perturbation's correlation would vanish.

Table 11: Competition-foil residual statistics: the organizer FNO's test rollout and the fit of its training residual, the quantities behind the alignment estimate of Appendix B.1.

Source	Quantity	Value
Test rollout (304 windows, 5 blocks)	Relative L_2	0.0903
Test rollout	RMS fluctuation ratio	0.543
Test rollout	Fluctuation-energy error	0.706
Training-residual fit	Lag-one correlation, time	0.872
Training-residual fit	Lag-one correlation, height	0.731
Training-residual fit	Lag-one correlation, width	0.826
Training-residual fit	Mean measured fluctuation energy	1.830×10^{-4}
Training-residual fit	Mean forecast fluctuation energy	9.813×10^{-5}
Training-residual fit	Residual variance	1.722×10^{-4}

The measured fluctuation spectrum flattens into a high-wavenumber plateau, the component attributable to PIV noise; Table 12 places it at 0.2–4% of the fluctuation variance on every system, so the deficits of Figure 2(a) are measured structure rather than noise.

Table 12: Measurement-noise floor of the PIV fields: the high-wavenumber plateau of the measured fluctuation spectrum, per system and measured channel, as a fraction of the fluctuation variance (0.2–4%).

System	Channel	Noise fraction	Noise RMS/fluct. RMS	Fluct. var.
Combustion	intensity	0.00960	0.0980	0.001693
Controlled cylinder	1	0.00405	0.0636	0.002071
Controlled cylinder	2	0.00279	0.0529	0.001898
Cylinder	1	0.00729	0.0854	0.000363
Cylinder	2	0.00299	0.0547	0.004747
Public foil	1	0.04038	0.2009	0.000086
Public foil	2	0.02516	0.1586	0.001377
Competition foil	1	0.03149	0.1775	0.000966
Competition foil	2	0.03157	0.1777	0.000128
FSI	1	0.00455	0.0675	0.002173
FSI	2	0.00227	0.0477	0.008017

Algorithm 1 FROST training and one-block sampling. m marks observed history frames, $M = 1 - m$, $\gamma(t) = \sigma\sqrt{t(1-t)}$, and f_ϕ returns (v, η). Both samplers integrate over $t \in [t_{\min}, 1 - t_{\min}]$ with $t_{\min} = 10^{-3}$, since $\gamma'(t)$ and the score parameterization are singular at the endpoints.

Training step input: measured block (h, y), frozen forecaster G_θ, residual scale s

$g \leftarrow \mathrm{stopgrad}(G_\theta(h))$
$c \leftarrow [\,h \| g\,]$
$r \leftarrow (y - g) \oslash s$ on future frames and 0 on history frames
$t \sim \mathcal{U}(0,1)$; $\quad x_0, z \sim \mathcal{N}(0, I)$
$x_t \leftarrow (1-t)x_0 + tr + \gamma(t)z$
$(v, \eta) \leftarrow f_\phi([x_t, c, m], t)$
$\mathcal{L} \leftarrow \mathrm{mean}_M\big(\|v - (r - x_0)\|_2^2 + \|\eta - z\|_2^2\big)$
AdamW update of ϕ; update exponential moving average $\bar{\phi}$

Sampling one block input: member history $h^{(k)}$, steps J, $\varepsilon \in \{0, \sigma^2/2\}$

$g \leftarrow G_\theta(h^{(k)})$; $\quad c \leftarrow [\,h^{(k)} \| g\,]$
$x_0, z \sim \mathcal{N}(0, I)$; $\quad x \leftarrow x_0$; $\quad \Delta \leftarrow (1 - 2t_{\min})/J$
for $j = 0, \ldots, J-1$ **do**
  $t_j \leftarrow t_{\min} + j\Delta$
  $x \leftarrow m \odot \big((1-t_j)x_0 + \gamma(t_j)z\big) + M \odot x$ ▷ hold history on its path
  $(v, \eta) \leftarrow f_{\bar{\phi}}([x, c, m], t_j)$
  $a \leftarrow v + \gamma'(t_j)\eta$
  $\xi \sim \mathcal{N}(0, I)$
  $x \leftarrow x + \Delta\,(a - \varepsilon\eta/\gamma(t_j)) + \sqrt{2\varepsilon\Delta}\,\xi$
end for
$\tilde{r} \leftarrow M \odot x$
$\tilde{y}^{(k)} \leftarrow g + s \odot \tilde{r}$ on future frames
$h^{(k)} \leftarrow \mathrm{tail}_{T_{\mathrm{in}}}[h^{(k)}, \tilde{y}^{(k)}]$

C Method details (Section 3)

C.1 Training and sampling algorithm

Algorithm 1 gives the training step and one-block sampler of Section 3. The main configuration uses $\sigma = 0.15$, $K = 4$ members, 20,000 AdamW updates (Loshchilov & Hutter, 2019) at batch size eight, and exponentially averaged weights for evaluation, with no condition dropout or classifier-free guidance (Table 13).

C.2 Architecture and optimization

The reference corrector is a three-dimensional U-Net (Ronneberger et al., 2015; Çiçek et al., 2016) over time and the two spatial axes, with base width 64, channel multipliers $(1, 2, 4)$, two residual blocks per level, and bottleneck attention. The model uses separate velocity and noise heads, with time conditioning and the history and forecast supplied as input channels. Its parameter count is approximately 40.7 million and depends slightly on the input channel count. Architecture ablations are not pooled into the seed estimate of the reference model.

Training uses 20,000 gradient updates at batch size eight. AdamW uses learning rate 2×10^{-4}, coefficients $(0.9, 0.95)$, and weight decay 0.01. The schedule warms up for 1,000 updates and follows cosine decay. Gradients are clipped at norm one. An exponential moving average with decay 0.999 supplies the evaluation weights.

C.3 The known past and forecast condition

The network operates on a block containing both the known history and the future. The field condition contains the observed history followed by the frozen surrogate's prediction. In the residual parameterization, the known endpoint is constructed in residual coordinates. At each sampling time, the known region is overwritten with its value along the same interpolant path, with fixed source and bridge-noise draws for that sampling trajectory. At the end, the known observations are restored exactly. The scored loss and forecast metrics concern the future region.

C.4 What the forecast condition supplies

Because the surrogate is a function of the observed history, conditioning on it changes the representation and training resources available to the corrector, not the information observed at prediction time. This distinguishes the reference task from directly supplying a synchronized numerical future. The latter may contain additional information and may also be phase-misaligned with the measurements. Forecasting from a model output and correcting a paired simulation are therefore separate experiments.

C.5 Residual and bridge objectives, source noise, and samplers

In the residual construction the training endpoints are independent Gaussian noise and $(y - g)/s$, with the stochastic-interpolant perturbation $\gamma(t)z$ between them. The forecast-anchored bridge uses $g + \alpha s\xi$ and y, and direct generation uses Gaussian noise and y; the known-observation mask is applied in the coordinates of the respective endpoint. Source-noise amplitude α, residual scale s and interpolant noise σ are different quantities, and a change to the source amplitude at inference is not a newly trained residual scale.

Euler ODE sampling evaluates the learned marginal drift once per step. The SDE sampler adds score-dependent drift and stochastic increments that preserve the interpolant marginals for an exact model (Appendix C.6); with a finite network and step count the two reads differ, and we compare their measured outputs rather than treating equal step counts as equal algorithms. Deterministic residual regression has one network call and no generated distribution.

C.6 Parameterization, samplers and the main recipe

Residual and direct generation describe the same laws. Fix (h, g) and let $T_g(r) = g + s \odot r$ with $s > 0$. T_g is an affine bijection of the future block, so every conditional density $q(y \mid h, g)$ corresponds to exactly one residual density $p(r \mid h, g) = q(T_g(r) \mid h, g) \prod_c s_c^{n_c}$, where n_c is the number of future entries in channel c. A sufficiently expressive conditional generator can therefore represent the same predictive distribution in either coordinate, and sampling $\tilde{r}$ then decoding $\tilde{y} = T_g(\tilde{r})$ is sampling q. The two differ as learning problems. In field coordinates the target equals g plus a discrepancy whose variance is s_c^2 in units of the field variance, 0.0095 and 0.032 for the two cylinder channels ($s = (0.0975, 0.1783)$), so most of the regression signal asks the network to copy its condition. In residual coordinates the copy is performed by the decoder, the discrepancy has unit scale per channel, and the zero residual lies at the mode of the source. The scale s is the per-channel standard deviation of $(y - g)$ on future frames over 32 shuffled training batches of eight windows, computed once before training with measured histories and stored with the checkpoint; it is a scalar per channel and does not vary with pixel or lead time. The residual mean is not removed.

One model, a family of samplers. Let p_t be the conditional density of x_t in Eq. (3) and let a be its probability-flow drift, so that $\partial_t p_t + \nabla \cdot (a\, p_t) = 0$. For any $\varepsilon \geq 0$, adding and subtracting $\varepsilon \Delta p_t = \varepsilon \nabla \cdot (p_t \nabla \log p_t)$ gives

$$\partial_t p_t + \nabla \cdot \big((a + \varepsilon \nabla \log p_t)\, p_t \big) - \varepsilon \Delta p_t = 0,$$

the Fokker–Planck equation of Eq. (6). The ODE and every SDE in the family therefore share the marginals p_t, and in particular the law of $\tilde{r}$ at $t = 1$, whenever a and the score are exact (Albergo et al., 2025; Song et al., 2021). With a finite network and J steps they differ: the ODE propagates its errors, while the diffusion term and the score contract toward the learned marginal. We use $\varepsilon = \sigma^2/2$. At the population optimum of Eq. (4), $v_\phi = \mathbb{E}[r - x_0 \mid x_t, c]$ and $\eta_\phi = \mathbb{E}[z \mid x_t, c]$, which give the drift and score of Eq. (5); the heads are combined only at sampling time, so neither regression target contains the singular factor $\gamma'(t)$. Integration runs on a uniform grid over $[\delta, 1 - \delta]$, $\delta = 10^{-3}$, started from x_0, and the score uses $\gamma(t) \geq \sigma\delta$. With $\sigma = 0$ the grid is $[0, 1]$, the second head receives no loss, and only the ODE is available. Members use independent draws of (x_0, z).

Known frames. The state spans the history as well as the future. In residual coordinates the known endpoint on history frames is zero, because the condition block equals the measured history there. At each sampling time those frames are overwritten by their exact interpolant value $(1-t)x_0+\gamma(t)z$, with (x_0, z) fixed for the whole trajectory, and at the end they are set to the known endpoint, so the

decoded block reproduces the measured history exactly. The loss is averaged over future frames only; including the history would only teach the network to copy.

Operator adapter. The operator is evaluated without gradients inside both the training step and the sampler. Its input is the history in physical units, padded with zeros in channels that were not measured (pressure; the fifteen simulation-only combustion channels) and, for controlled cylinder, with its two physical parameters as constant channels, then normalized with the operator's own statistics; its output is un-normalized, restricted to the measured channels and re-normalized with the corrector's field statistics. No simulation field enters the corrector.

Cost. With K members, J corrector steps and B blocks, one forecast window costs KJB corrector evaluations and KB operator evaluations: 240 and 40 for the main read ($K = 4$, $J = 6$, $B = 10$), 1,200 and 40 for the stochastic read ($J = 30$). Batched execution changes latency but not these counts.

Table 13: The main FROST recipe. Every headline result uses these settings; departures are named where they occur.

Quantity	Setting
Block	$T_{\text{in}} = T_{\text{out}} = 20$ frames (10 and 10 for controlled cylinder)
Grid and channels	$64 \times 128 \times 2$ (cylinder, controlled cylinder, public foil); $64 \times 64 \times 2$ (FSI); $64 \times 64 \times 1$ (combustion); $32 \times 64 \times 2$ (competition foil)
Target	residual $r = (y - g)/s$, uncentered; s per channel from 32 training batches
Condition	$[\, h \,\|\| \, g \,]$ concatenated with the state and an observed-frame mask ($2C + 1$ input channels)
Interpolant	Gaussian source, $\gamma(t) = \sigma\sqrt{t(1-t)}$, $\sigma = 0.15$
Loss	two heads, equal weights, future frames only, $t \sim \mathcal{U}(0, 1)$, no time weighting
Network	3-D U-Net, widths 64/128/256, two residual blocks per level, stride-2 sampling in space and time, bottleneck self-attention (4 heads), zero-initialized heads, 40.7 M parameters
Optimizer	AdamW, learning rate 2×10^{-4}, $\beta = (0.9, 0.95)$, weight decay 0.01, 1,000 warm-up updates then cosine decay, gradient clipping at norm 1
Budget	20,000 updates, batch size 8, bfloat16 autocast
Evaluation weights	exponential moving average, decay 0.999
Guidance	none (no condition dropout; one conditional evaluation per step)
Main read	$K = 4$ members, Euler, $J = 6$, $t \in [10^{-3}, 1 - 10^{-3}]$
Stochastic read	Euler–Maruyama, $J = 30$, $\varepsilon = \sigma^2/2$
Rollout	10 blocks (5 for competition foil), members keep their own histories
Seeds	three corrector seeds per system on one fixed operator

C.7 Other instantiations of prediction-space adaptation

The target of Section 3.1, a distribution over the residual around a frozen operator's forecast, is independent of the transport that realizes it and of the coordinate it is expressed in. We instantiated it three other ways, each retaining the same network, condition and data.

Direct generation and the forecast-anchored bridge. Direct generation transports noise to the experimental future y rather than to the residual, with g retained as a condition; the forecast-anchored bridge starts the transport near the forecast, $x_0 = g + \alpha s \odot \xi$, and ends at y, placing the forecast in the source distribution and in the condition (Albergo et al., 2024; Chen et al., 2024; Liu et al., 2023a; Zhou et al., 2024). With the network, condition and data fixed, the three targets land within 0.0023 of one another in accuracy on cylinder and differ by a factor of six in member energy error (Table 14; Figure 5A). The direct generator's members are damped in the way Section 2.2 diagnosed for the forecast: asked to reproduce the whole field, it re-learns the mean flow. Residual coordinates are the default because they are never behind on accuracy and best where it matters most, on the over-energetic cylinder; the bridge is an alternative coupling with a different energy trade.

MeanFlow. Replacing iterative transport by a learned average velocity for one-step generation (Geng et al., 2025a;b) keeps the pointwise gain on cylinder at a single corrector call and leaves the members' fluctuation fidelity sensitive to the sampling budget; it is behind the flow on competition foil. One-evaluation probabilistic forecasters (Xu et al., 2026; Jha et al., 2026; Yang & Xue, 2026)

Table 14: **On cylinder the three generative targets land within 0.0023 of one another in rollout error and differ sixfold in member energy error.** The same network and condition, three targets: the field itself, the normalized residual, or a bridge whose source is the forecast plus scaled noise. Each row is one system and one generation target, over three seeds; competition foil uses width 48 and five blocks. Bold marks the best rollout error and CRPS in each system, the 0.0018 floor applying to rollout only; the amplitude column is not bolded.

System	What is generated	n	Single block	Rollout	CRPS	Energy err.	RMS ratio
Cylinder	Direct generation of the field	3	0.0748 ± 0.0002	0.1594 ± 0.0014	0.0192 ± 0.0001	0.558 ± 0.016	0.665 ± 0.012
	Residual (FROST)	3	$\mathbf{0.0667 \pm 0.0001}$	$\mathbf{0.1571 \pm 0.0003}$	$\mathbf{0.0177 \pm 0.0001}$	$\mathbf{0.090 \pm 0.065}$	1.044 ± 0.031
	Forecast-anchored bridge	3	0.0691 ± 0.0000	$\mathbf{0.1583 \pm 0.0020}$	0.0183 ± 0.0003	0.224 ± 0.107	1.105 ± 0.049
Comp. foil	Direct generation of the field	3	0.0483 ± 0.0002	0.0851 ± 0.0004	0.0036 ± 0.0000	0.355 ± 0.008	0.803 ± 0.005
	Residual (FROST)	3	$\mathbf{0.0477 \pm 0.0000}$	$\mathbf{0.0845 \pm 0.0005}$	$\mathbf{0.0036 \pm 0.0000}$	0.408 ± 0.016	0.770 ± 0.010
	Forecast-anchored bridge	3	0.0534 ± 0.0002	0.0905 ± 0.0006	0.0037 ± 0.0000	$\mathbf{0.325 \pm 0.017}$	0.821 ± 0.011

are the cheap end of the same trade, and CRPS-trained ensembles have been found better calibrated than diffusion and flow emulators trained in a latent space, and as well calibrated as ones trained in the ambient space, on simulated two-dimensional systems (Greenbury et al., 2026); interpolants between successive states have been found accurate at two to five sampling steps, with a stochastic sampler and more steps needed to keep the spectrum (Zhou et al., 2025), and on stochastically forced flow to need a relatively large inference budget to compete with noise-sourced generators (Pfister et al., 2026), an energy-consistent interpolant emulator has been trained from scratch on periodic simulated turbulence (Mücke & Sanderse, 2026), and conditioning a diffusion model for PDE forecasting has its own design space (Shysheya et al., 2024), of which channel concatenation is the route we keep (Appendix C.8). The full ladder against its matched control is in Appendix G (Table 37).

C.8 Where the gain comes from: the forecast, not the wiring

Table 15: **No conditioning route beats channel concatenation by more than one floor on cylinder.** Every added pathway is zero-initialized, so each variant equals the main recipe at initialization. Δ is in units of the 0.0018 run-to-run floor; nothing clears two floors at ODE-6. Rows with $n = 1$ are screening.

How the condition enters	n	ODE-2	ODE-6	Δ vs. concat.
Channel concatenation (main recipe)	3	–	0.1571 ± 0.0003	+0.0 f
ControlNet-style branch	3	0.1652 ± 0.0059	0.1569 ± 0.0029	-0.1 f
Bottleneck cross-attention	1	0.1634	0.1580	+0.5 f
Branch + cross-attention	1	0.1633	0.1573	+0.1 f
Surrogate hidden features (all layers)	1	0.1636	0.1571	-0.0 f
Branch + cross-attention + features	1	0.1592	0.1553	-1.0 f
Wider net, parameter-matched (dim 72)	1	0.1702	0.1599	+1.6 f
Width 32	3	0.1620 ± 0.0011	0.1579 ± 0.0011	+0.4 f
ADM port of the same U-Net	3	0.1616 ± 0.0028	0.1567 ± 0.0014	-0.2 f

Section 2.2 and Appendix B.2 measured what the forecast contributes on the same models: removing the forecast is six times worse than not correcting, replacing it with its window statistics gives no gain, rotating it in time doubles the error, and inserting true future frames improves it; Table 34 supplies the remaining piece, the same network trained by regression capturing the bias with FROST better than it in CRPS on every system and in accuracy on all five, on foil by one floor. Nothing we tried in the conditioning pathway moves the result by more than two floors (Table 15; Figure 5C). Every alternative to channel concatenation was built as a zero-initialized addition, so each variant is the main recipe at initialization and differs only in what training can do with the new pathway: a ControlNet-style branch (Zhang et al., 2023) into every skip connection, bottleneck cross-attention, the operator's own hidden features as an extra input, their combinations, a parameter-matched wider network, a half-width network and a port of the U-Net to the guided-diffusion ADM architecture all lie within ± 1.6 floors of the reference. The same holds for every ADM-style knob on cylinder (Table 40) and for σ, the residual scale and the training length on competition foil (Table 42). The

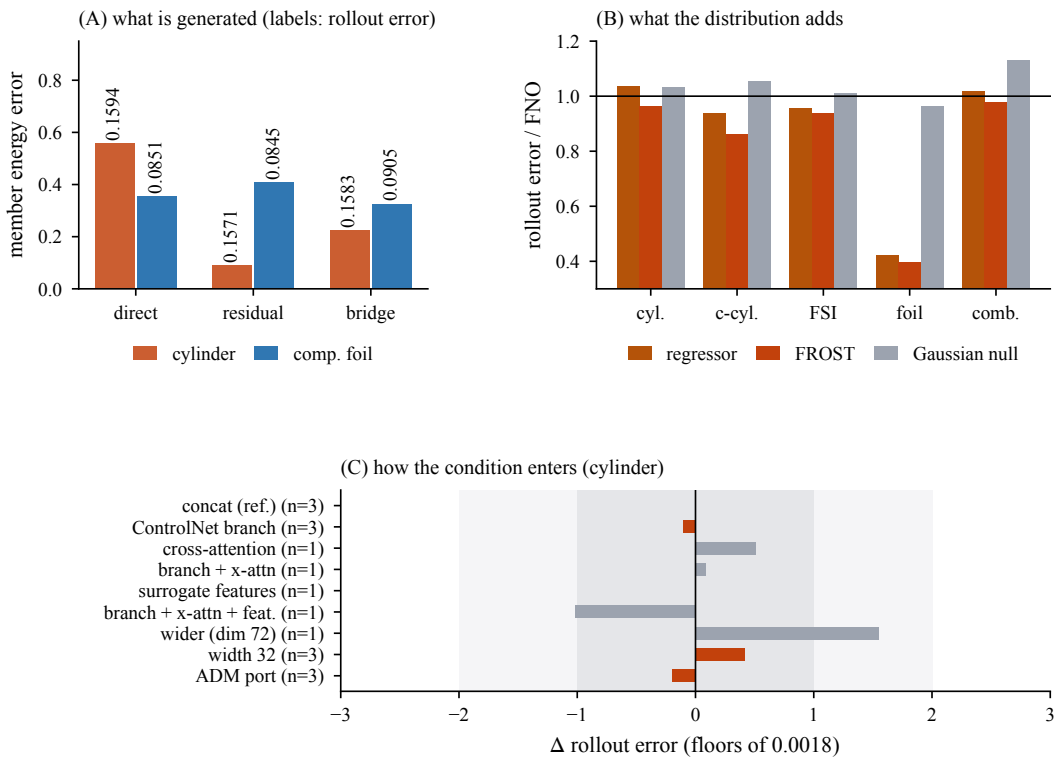

Figure 5: **The gain comes from the forecast, not from the wiring.** **(A)** What is generated: the field directly, the normalized residual, or a forecast-anchored bridge, with the same network and condition; bars are member energy error, labels are rollout error. **(B)** What the distribution adds: rollout error relative to the FNO for the deterministic regressor, FROST and the calibrated Gaussian null. **(C)** How the condition enters the network on cylinder: change in rollout error in floors of 0.0018 (shaded: one and two floors); grey bars are single-seed screening.

inductive bias that matters is giving the generative model a useful, aligned deterministic future to correct.

D BASELINES (SECTION 4.1)

Table 16 states what each comparison implements, so that a reimplementation and a control are read as such.

Table 16: What each comparison implements.

Reference	Implemented experiment	Scope of the comparison
PhysGuard	Fisher-gradient projection, rank cap 60, read through the shared evaluator	Reimplementation at rank cap 60 (published: 500); three adaptation seeds on every system.
PDE-Refiner	Iterative residual denoising with a decreasing noise schedule	Refinement recipe on FROST's network, condition and data; one seed.
Probabilistic Retrofitting	Noise embedding modulating the operator's normalization layers, trained under the fair CRPS (Appendix D.5); a CRPS-trained surrogate with dropout noise is the second variant	Port of the published mechanism onto the benchmark's FNO, three seeds per system.
MC dropout	Dropout injected into a checkpoint at inference	Inference-time perturbation of the fine-tuned operator.
Pushforward	Additional surrogate training under its generated input	Deterministic adaptation control.

Diaconu et al. (2026) train the retrofitted operator under a proper scoring rule with a shared noise embedding modulating its hidden representations (official implementation at `https://github.com/cddcam/lola_crps`, revision `7643376c2949717ee5c2c840584689f529ba77a5`). Appendix D.5 ports that mechanism onto the benchmark's FNO; the CRPS-trained surrogate row keeps the proper-score training with dropout as the noise source.

D.1 PHYSGUARD: IMPLEMENTATION AND COMPARISON SCOPE

The PhysGuard rows use our implementation of Fisher-guided gradient projection (Zhou et al., 2026), not numbers copied from that paper. It uses $\tau = 0.9$, $\alpha = 1$, a maximum of 60 protected directions per parameter tensor, and 500 simulation samples for Fisher estimation; the published setup allows up to 500 directions and 10% of the simulation samples, so we label the baseline a reimplementation at rank cap 60.

Each leaf parameter tensor defines a group, and complex FNO weights are represented by their real and imaginary components. Fisher estimation runs in evaluation mode with simulation augmentation disabled; real-data gradients are projected before an Adam update. Ordinary fine-tuning and PhysGuard use the upstream loss, including the zero-filled unmeasured channels, while reported test scores exclude those channels. PhysGuard's learning rate is 10^{-4} for cylinder, controlled cylinder and foil and 10^{-2} for FSI and combustion.

The PhysGuard checkpoints are read through the same evaluator, data revision, history and forecast lengths and 320-window selection as FROST, and their ordered trajectory and parameter-group identifiers agree with FROST's evaluation.

PhysGuard has three independently adapted FNO checkpoints on every system, whereas FROST's three seeds vary the corrector around one fine-tuned FNO. Each PhysGuard checkpoint supplies a deterministic forecast, evaluated separately with $K = 1$; we do not ensemble the three adapted models, and its CRPS is the absolute error of a point mass. At inference it needs one FNO call per block and no corrector calls. Table 32 gives its member statistics and Table 20 (Appendix D.6) every adaptation seed.

D.2 PUSHFORWARD FINE-TUNING

Pushforward fine-tuning (Brandstetter et al., 2022) continues training the operator on inputs it generated itself, the parameter-space answer to the history shift of Appendix G.1. As a deterministic forecast its fair CRPS stays within 9% of the fine-tuned FNO's on cylinder and controlled cylinder, where FROST lowers it by 20 and 35%, and on FSI and public foil (one seed each) it is less accurate than the fine-tuned FNO. Table 21 gives every seed and Table 22 the paired intervals against FROST.

D.3 FURTHER ADAPTATIONS OF THE SAME OPERATOR

Tables 17 and 18 screen further parameter-space adaptations of the same simulation-pretrained FNO on the same real split, one seed each: regularization toward the pretrained weights by L2-SP (Li et al., 2018) and diagonal EWC (Kirkpatrick et al., 2017); restricted updates by last-layer and surgical fine-tuning, the probe-then-fine-tune schedule LP-FT (Kumar et al., 2022), bias-only BitFit (Ben-Zaken et al., 2022) and low-rank adapters on the linear and convolutional layers (Hu et al., 2022); CORAL feature-covariance alignment (Sun & Saenko, 2016); input-noise training; and a heteroscedastic output head. A rank-matched random subspace tests whether PhysGuard's Fisher directions matter.

Table 17: **No further parameter-space adaptation of the same operator improves its rollout error on all five systems.** Rollout error of further parameter-space adaptations of the same FNO through the shared evaluator, one seed each, and in the lower block two inference-time ensembles of the FNO. BitFit, LoRA and last-layer tuning raise the cylinder and controlled-cylinder error roughly threefold or more.

Method (one seed)	Cylinder	Contr. cyl.	FSI	Public foil	Combustion
Fine-tuned FNO	0.1632	0.1728	0.2274	0.0754	0.7918
BitFit	0.5754	0.6246	1.2413	0.1902	0.9760
LoRA (r=8)	0.6564	0.6135	0.3836	0.0783	0.8270
Surgical: spectral layers	0.1614	0.1723	0.2228	0.0775	0.8223
Surgical: staged spectral, last	0.1626	0.1712	0.2255	0.0763	0.7871
Surgical: last layer	0.5116	0.5434	diverges	0.0766	0.9632
CORAL feature alignment	0.3770	–	0.2401	–	1.4323
Input-noise training	0.1671	0.1807	0.2256	0.1820	0.7838
Heteroscedastic head	0.3676	0.3746	0.2320	0.1879	0.9423
EWC (combustion only)	–	–	–	–	0.8345
L2-SP (combustion only)	–	–	–	–	0.8362
CRPS fine-tune + MC dropout ($K = 4$)	0.5179	0.2646	0.2426	0.0571	0.8189
MC dropout 0.1 ($K = 4$)	0.3561	0.4448	0.2500	0.1020	0.7766

None of these adaptations improves on plain fine-tuning on all five systems (Table 17). The parameter-efficient updates do not survive the rollout: on cylinder, BitFit, LoRA and last-layer tuning reach ten-block errors of 0.29 to 1.06 across the learning rates of Table 18, against 0.1632 for plain fine-tuning, and last-layer tuning diverges on FSI. Surgical fine-tuning of the spectral layers or the convolutional path, input-noise training and LP-FT at its best learning rate stay within 0.004 of plain fine-tuning on cylinder. CORAL feature alignment and the heteroscedastic head are worse than plain fine-tuning on every system where they ran, and an FNO trained on the real data alone, without simulation pretraining, is worse too (0.1847). The rank-matched random subspace scores as plain fine-tuning does (0.1633).

Table 18: **On cylinder, BitFit, LoRA and last-layer tuning raise the rollout error at every learning rate tried.** Learning-rate ladders of the parameter-efficient methods, the remaining surgical variants, the rank-matched random-subspace control for PhysGuard, and a real-data-only FNO. One seed each. Each row is one adaptation of the same fine-tuned FNO, which itself scores 0.1632 / 0.02201 / 0.529 on these three metrics.

Adaptation of the fine-tuned FNO (cylinder, one seed)	Rollout	CRPS	Energy err.
LP-FT, lr $\times 0.3$	0.1829	0.02497	0.472
LP-FT, lr $\times 1$	0.1709	0.02435	0.492
LP-FT, lr $\times 3$	0.1632	0.02379	0.491
LP-FT, lr $\times 10$	0.1612	0.02330	0.496
LoRA $r = 8$, lr $\times 0.3$	0.9961	0.08217	7.716
LoRA $r = 8$, lr $\times 1$	0.6564	0.06435	9.481
LoRA $r = 8$, lr $\times 3$	0.3978	0.04357	0.248
LoRA $r = 8$, lr $\times 10$	0.2872	0.03610	0.318
BitFit, lr $\times 0.3$	1.0568	0.11394	8.335
BitFit, lr $\times 1$	0.5754	0.06303	0.150
BitFit, lr $\times 3$	0.5424	0.06268	0.002
BitFit, lr $\times 10$	0.5514	0.06207	0.048
Surgical: first layer	0.7671	0.12704	113.300
Surgical: convolutional path	0.1627	0.02205	0.543
Surgical: normalization layers	0.6028	0.06931	0.644
Surgical: last layer, lr $\times 0.3$	0.5679	0.06323	0.254
Surgical: last layer, lr $\times 3$	0.4539	0.05325	0.211
Surgical: last layer, lr $\times 10$	0.3680	0.04647	0.107
CORAL, weight 0.1	0.3734	0.04616	0.738
CORAL, weight 1	0.3770	0.04655	0.748
Fisher-random subspace control (PhysGuard rank-matched)	0.1633	0.02209	0.565
Real-data-only FNO (no simulation pretraining)	0.1847	0.02496	0.504

D.4 Probabilistic baselines

Proper scoring rules evaluate predictive distributions in expectation; a finite trained model with improved CRPS need not be calibrated (Gneiting & Raftery, 2007; Gneiting et al., 2007), so the probabilistic baselines are read on CRPS and on the calibration statistics of Appendix E.5.

The PDE-Refiner-style residual denoiser iteratively refines the forecast under its own noise schedule on FROST's network, condition and data, with one training seed. Inference dropout perturbs an existing deterministic checkpoint without having trained it as a dropout model. A heteroscedastic head predicts a mean and a local scale rather than a joint law over the field. The calibrated Gaussian null fixes its scale from training residuals and adds no learned conditional bias.

The CRPS-trained dropout surrogate keeps the proper-score training of probabilistic retrofitting (Diaconu et al., 2026) with dropout as the noise source; on cylinder it diverges on a subset of windows, and its average over all windows is reported. The published noise-modulation mechanism is ported in Appendix D.5. At $K = 1$, CRPS is the absolute error of a point forecast.

D.5 Probabilistic retrofitting of the operator

Probabilistic retrofitting turns the operator itself into a member generator: a shared noise embedding modulates its hidden representations and its weights are trained under a proper scoring rule (Diaconu et al., 2026). We port that mechanism onto the benchmark's FNO, where the embedding drives a conditional affine transform of every normalization layer, and train it on the real split under the fair CRPS with the published two-rate optimizer, at the adaptation budget of the fine-tuned FNO. Three seeds per system, read with $K = 4$ members through the shared evaluator.

Table 19 reads as the trade of Section 2.2. Moving the operator's weights under a proper score lowers its fair CRPS on all five systems, by 26% on average; the same update leaves the corrected mean less accurate than the operator it adapts, on all five. FROST is more accurate than both on every system, and the thirty-step read of its single trained checkpoint is ahead of the retrofit in CRPS on four of five.

Table 19: **Making the operator stochastic buys probabilistic skill and spends accuracy; a distribution around the operator forecast gains both.** The retrofit adapts the same simulation-pretrained FNO on the same real split, under the fair CRPS, for the same number of updates as the fine-tune. Three seeds per row; seed standard deviations are at most 0.0028 in rollout error and 0.00015 in CRPS, and 0.050 and 0.0078 on public foil. The FNO is deterministic, so its CRPS is its absolute error. Bold: best of each column among the full-window three-seed rows.

Method	Cylinder	Contr. cyl.	FSI	Public foil	Combustion
Rollout relative L_2 of the ensemble mean (lower is better)					
Fine-tuned FNO (reference)	0.1632	0.1728	0.2274	0.0754	0.7918
Probabilistic retrofit ($K = 4$)	0.1724	0.1946	0.2317	0.0883	0.7961
FROST (ODE-6)	**0.1571**	**0.1490**	**0.2134**	**0.0298**	**0.7750**
FROST (SDE-30, 128 windows)	0.1538	0.1486	0.2192	0.0292	0.7673
Fair CRPS (lower is better)					
Fine-tuned FNO (reference)	0.02201	0.01279	0.01775	0.01362	0.00917
Probabilistic retrofit ($K = 4$)	**0.01636**	0.00910	**0.01208**	0.01304	**0.00568**
FROST (ODE-6)	0.01772	**0.00826**	0.01264	**0.00339**	0.00694
FROST (SDE-30, 128 windows)	0.01518	0.00717	0.01173	0.00300	0.00639

D.6 PHYSGUARD, EVERY ADAPTATION SEED, AND EVERY SEED OF THE MAIN TABLE

Table 20: **PhysGuard's cylinder members grow more over-energetic with each adaptation seed.** Ten-block rollout through the shared evaluator.

System	Adaptation seed	Rollout	CRPS	Energy err.	RMS ratio
Cylinder	0	0.1648	0.0231	1.288	1.513
	1	0.1695	0.0253	2.682	1.919
	2	0.1720	0.0272	4.229	2.287
Contr. cyl.	0	0.1746	0.0129	0.179	0.906
	1	0.1745	0.0129	0.172	0.910
	2	0.1744	0.0129	0.185	0.903
FSI	0	0.2255	0.0175	0.096	0.951
	1	0.2268	0.0177	0.068	0.965
	2	0.2265	0.0177	0.066	0.967
Public foil	0	0.0815	0.0143	0.296	0.839
	1	0.0937	0.0161	0.203	0.893
	2	0.0737	0.0129	0.296	0.839
Combustion	0	0.8219	0.0095	0.106	1.052
	1	0.8360	0.0097	0.172	1.083
	2	0.8197	0.0095	0.097	1.047

Table 20 prints every PhysGuard adaptation seed through the shared evaluator: on average its ten-block rollout is behind plain fine-tuning on four systems, and its cylinder members grow more over-energetic with each seed (RMS ratio 1.51, 1.92, 2.29). Table 21 prints every seed of PhysGuard, pushforward fine-tuning, the deterministic residual and FROST.

Table 21: **Every training seed of the adaptation and correction arms scored on all five systems**: rollout error / fair CRPS. Each row is one arm and each column group one system. Pushforward has one seed on FSI and public foil; the thirty-step read's seed spread is in Table 29.

System	Method	Seed 0	Seed 1	Seed 2
Cylinder	PhysGuard (rank-60 reimpl.)	0.1648 / 0.02308	0.1695 / 0.02529	0.1720 / 0.02724
	Pushforward fine-tuning	0.1576 / 0.02169	0.1575 / 0.02187	0.1577 / 0.02191
	Deterministic residual	0.1709 / 0.02210	0.1685 / 0.02158	0.1686 / 0.02194
	FROST, ODE-6	0.1570 / 0.01761	0.1568 / 0.01774	0.1575 / 0.01782
Contr. cyl.	PhysGuard (rank-60 reimpl.)	0.1746 / 0.01293	0.1745 / 0.01288	0.1744 / 0.01287
	Pushforward fine-tuning	0.1567 / 0.01180	0.1562 / 0.01168	0.1558 / 0.01165
	Deterministic residual	0.1627 / 0.01177	0.1615 / 0.01178	0.1631 / 0.01199
	FROST, ODE-6	0.1501 / 0.00835	0.1485 / 0.00821	0.1482 / 0.00823
FSI	PhysGuard (rank-60 reimpl.)	0.2255 / 0.01746	0.2268 / 0.01767	0.2265 / 0.01766
	Pushforward fine-tuning	0.2429 / 0.02088	–	–
	Deterministic residual	0.2171 / 0.01604	0.2172 / 0.01607	0.2172 / 0.01607
	FROST, ODE-6	0.2134 / 0.01263	0.2138 / 0.01268	0.2129 / 0.01261
Public foil	PhysGuard (rank-60 reimpl.)	0.0815 / 0.01435	0.0937 / 0.01615	0.0737 / 0.01287
	Pushforward fine-tuning	0.1775 / 0.02705	–	–
	Deterministic residual	0.0319 / 0.00490	0.0322 / 0.00495	0.0310 / 0.00472
	FROST, ODE-6	0.0296 / 0.00337	0.0304 / 0.00341	0.0294 / 0.00338
Combustion	PhysGuard (rank-60 reimpl.)	0.8219 / 0.00953	0.8360 / 0.00972	0.8197 / 0.00952
	Pushforward fine-tuning	0.7472 / 0.00920	1.0449 / 0.01076	0.7474 / 0.00925
	Deterministic residual	0.8041 / 0.00913	0.8035 / 0.00920	0.8107 / 0.00923
	FROST, ODE-6	0.7736 / 0.00694	0.7785 / 0.00696	0.7729 / 0.00691

E MAIN RESULTS IN FULL (SECTION 4.2)

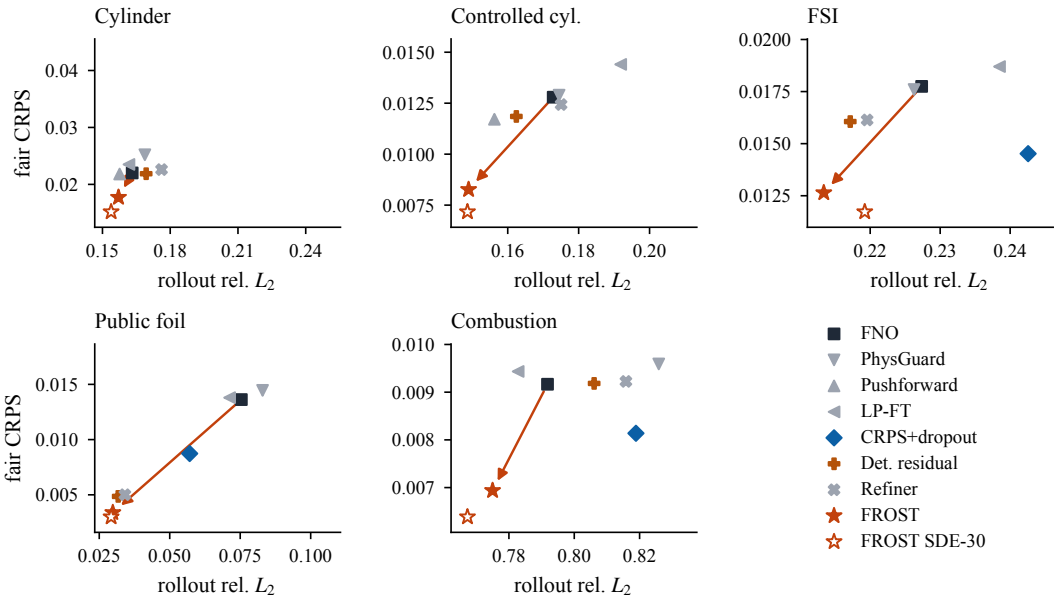

Figure 6: **Accuracy and probabilistic skill move together only for FROST.** The adaptation and correction methods of Appendix D and Table 6 on the rollout-error/CRPS plane, one panel per system; lower left is better. The arrow runs from the fine-tuned FNO to FROST at the default read; the open star is FROST's thirty-step stochastic read on its 128-window subset. The CRPS-trained dropout surrogate on cylinder lies outside the plotted range.

Figure 6 places every adaptation and correction method on the error/CRPS plane: FROST is the only one that moves the fine-tuned FNO down and to the left on all five systems. Every seed behind the three-seed rows is in Table 21, the per-group scores are in Table 4, and the member statistics are in Appendix F.1.

E.1 HELD-OUT PARAMETER GROUPS AND PAIRED INTERVALS

Table 4 of Section 4.3 gives the per-group scores: on cylinder the gain on the held-out groups is larger than on the seen group, and on foil the held-out groups gain a factor of two to four. The groups are the benchmark's held-out parameter labels; Appendix A.2 records how they relate to the seen range. Table 22 gives the paired bootstrap intervals, resampling trajectories rather than windows, for every method that shares FROST's windows.

E.2 THE PRETRAINED OPERATOR AND REAL-DATA SCALING

Table 23 gives the simulation-only and fine-tuned pipelines side by side with their member energy, behind Figure 7B and the simulation-only result of Section 4.2. Table 24 gives the cells behind Figure 7A: identical trajectory subsets, three seeds per cell. Fine-tuning alone needs all 72 training trajectories to reach 0.163, with a seed spread of 0.026 at 12.5% of them against 0.0003 for a corrector over the fixed operator; the matched pipeline is ahead of fine-tuning by 0.011–0.012 at every fraction below full data and by 0.006 at full data. The fine-tuning baseline is our own recipe, whose full-data anchor (0.1629) reproduces the released checkpoint (0.1632).

E.3 TRANSFER BETWEEN FORECASTERS

Tables 25 and 26 give the full backbone rows behind Table 2, and Table 5 of Section 4.3 gives the transfer matrix. A corrector can be taught the discrepancy of several forecasters at once, and then it travels: trained on a five-way mixture of forecasters, it stays within 0.005 of the matched corrector

Table 22: **FROST's mean is ahead of every rival on every system, and fifty-eight of the sixty intervals exclude zero.** Each row is one system and one rival; the statistic is FROST (ODE-6) minus that rival, the seed-pair-averaged difference of window means, so negative favors FROST. Trajectories are resampled with replacement (1,000 draws) and every method is scored on the same windows. 'Sign' counts seed pairs whose difference has the sign of the mean. *: interval excludes zero.

System	FROST minus	pairs	sign	Δ rollout rel. L_2 [95% CI]	sign	Δ CRPS [95% CI]
Cylinder	Fine-tuned FNO	3	3/3	-0.0061 [-0.0097, -0.0029]*	3/3	-0.0043 [-0.0070, -0.0030]*
Cylinder	PhysGuard (rank-60)	3	3/3	-0.0117 [-0.0275, -0.0029]*	3/3	-0.0075 [-0.0168, -0.0029]*
Cylinder	Pushforward FT	3	3/3	-0.0005 [-0.0029, +0.0020]	3/3	-0.0041 [-0.0067, -0.0028]*
Cylinder	CRPS-trained dropout ensemble	3	3/3	-0.3608 [-0.8997, -0.1382]*	3/3	-0.0929 [-0.2589, -0.0155]*
Cylinder	Gaussian null	3	3/3	-0.0116 [-0.0166, -0.0076]*	3/3	-0.0024 [-0.0047, -0.0012]*
Cylinder	Deterministic residual	3	3/3	-0.0138 [-0.0162, -0.0117]*	3/3	-0.0044 [-0.0053, -0.0036]*
Contr. cyl.	Fine-tuned FNO	3	3/3	-0.0239 [-0.0295, -0.0183]*	3/3	-0.0045 [-0.0051, -0.0041]*
Contr. cyl.	PhysGuard (rank-60)	3	3/3	-0.0255 [-0.0313, -0.0197]*	3/3	-0.0046 [-0.0052, -0.0042]*
Contr. cyl.	Pushforward FT	3	3/3	-0.0073 [-0.0111, -0.0040]*	3/3	-0.0034 [-0.0038, -0.0031]*
Contr. cyl.	CRPS-trained dropout ensemble	3	3/3	-0.1156 [-0.1316, -0.1009]*	3/3	-0.0078 [-0.0088, -0.0070]*
Contr. cyl.	Gaussian null	3	3/3	-0.0334 [-0.0412, -0.0264]*	3/3	-0.0021 [-0.0025, -0.0018]*
Contr. cyl.	Deterministic residual	3	3/3	-0.0135 [-0.0149, -0.0121]*	3/3	-0.0036 [-0.0040, -0.0032]*
FSI	Fine-tuned FNO	3	3/3	-0.0140 [-0.0169, -0.0112]*	3/3	-0.0051 [-0.0052, -0.0050]*
FSI	PhysGuard (rank-60)	3	3/3	-0.0129 [-0.0155, -0.0105]*	3/3	-0.0050 [-0.0051, -0.0049]*
FSI	Pushforward FT	3	3/3	-0.0295 [-0.0403, -0.0197]*	3/3	-0.0082 [-0.0099, -0.0069]*
FSI	CRPS-trained dropout ensemble	3	3/3	-0.0292 [-0.0320, -0.0262]*	3/3	-0.0019 [-0.0021, -0.0016]*
FSI	Gaussian null	3	3/3	-0.0165 [-0.0200, -0.0132]*	3/3	-0.0015 [-0.0016, -0.0013]*
FSI	Deterministic residual	3	3/3	-0.0038 [-0.0048, -0.0028]*	3/3	-0.0034 [-0.0036, -0.0033]*
Public foil	Fine-tuned FNO	3	3/3	-0.0456 [-0.0476, -0.0436]*	3/3	-0.0102 [-0.0120, -0.0089]*
Public foil	PhysGuard (rank-60)	3	3/3	-0.0531 [-0.0554, -0.0507]*	3/3	-0.0111 [-0.0126, -0.0100]*
Public foil	Pushforward FT	3	3/3	-0.1477 [-0.1567, -0.1376]*	3/3	-0.0237 [-0.0246, -0.0228]*
Public foil	CRPS-trained dropout ensemble	3	3/3	-0.0273 [-0.0304, -0.0238]*	3/3	-0.0053 [-0.0065, -0.0045]*
Public foil	Gaussian null	3	3/3	-0.0428 [-0.0449, -0.0406]*	3/3	-0.0073 [-0.0088, -0.0062]*
Public foil	Deterministic residual	3	3/3	-0.0019 [-0.0025, -0.0014]*	3/3	-0.0015 [-0.0021, -0.0010]*
Combustion	Fine-tuned FNO	3	3/3	-0.0168 [-0.0309, -0.0013]*	3/3	-0.0022 [-0.0026, -0.0018]*
Combustion	PhysGuard (rank-60)	3	3/3	-0.0509 [-0.0742, -0.0274]*	3/3	-0.0027 [-0.0031, -0.0021]*
Combustion	Pushforward FT	3	1/3	-0.0715 [-0.1689, +0.0088]	3/3	-0.0028 [-0.0033, -0.0024]*
Combustion	CRPS-trained dropout ensemble	3	3/3	-0.0439 [-0.0848, -0.0008]*	3/3	-0.0012 [-0.0018, -0.0006]*
Combustion	Gaussian null	3	3/3	-0.1193 [-0.2311, -0.0343]*	3/3	-0.0029 [-0.0031, -0.0027]*
Combustion	Deterministic residual	3	3/3	-0.0311 [-0.0477, -0.0129]*	3/3	-0.0022 [-0.0027, -0.0018]*

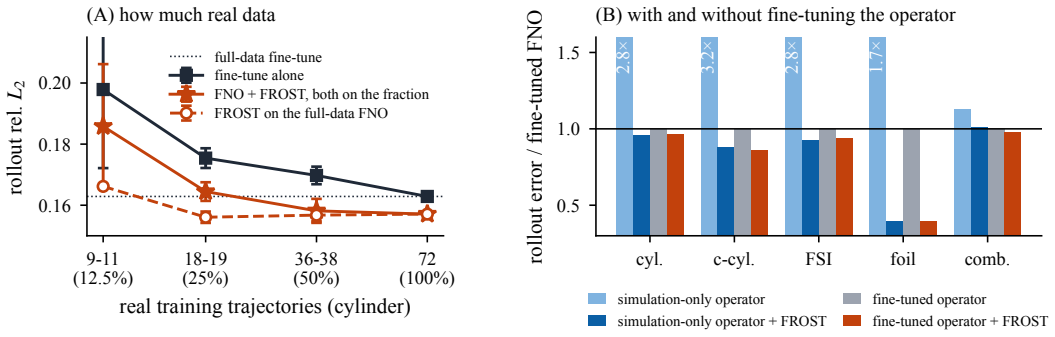

Figure 7: **A quarter of the real trajectories matches full-data fine-tuning on cylinder, and a simulation-only operator with FROST matches or beats the fine-tune-then-correct pipeline on three systems and the fine-tuned operator on four.** **(A)** Cylinder, identical trajectory subsets, three seeds per point: fine-tuning alone, the matched pipeline (operator and corrector both trained on the fraction), and FROST over the full-data operator, whose operator saw all 72 training trajectories, so this curve measures the corrector's own saturation; the axis gives the per-seed range of training trajectories in each subset. **(B)** Rollout error relative to the fine-tuned FNO for the simulation-only operator, the same operator with FROST, the fine-tuned operator and the fine-tuned operator with FROST (Table 23); operator-alone bars above the axis are clipped and carry their value.

Table 23: **Over an operator that never saw a measurement, the corrected mean is more accurate than the fine-tuned operator on four of five systems and matches or beats the fine-tune-then-correct pipeline on three.** The corrector (three seeds) is trained on the same real split in both halves; only the operator differs. Each row is a system; the two column groups are the simulation-only and the fine-tuned operator, each alone and corrected. Bold: the better of the two pipelines, the 0.0018 floor applying to rollout error only.

System	Simulation-only operator (never fine-tuned)				Fine-tuned operator			
	alone	+FROST	CRPS	Energy err.	alone	+FROST	CRPS	Energy err.
Cylinder	0.4600	**0.1565 ± 0.0005**	**0.0174 ± 0.0003**	0.412 ± 0.012	0.1632	**0.1571 ± 0.0003**	0.0177 ± 0.0001	**0.090 ± 0.065**
Contr. cyl.	0.5521	0.1522 ± 0.0022	**0.0080 ± 0.0002**	**0.074 ± 0.003**	0.1728	**0.1490 ± 0.0010**	0.0083 ± 0.0001	0.129 ± 0.010
FSI	0.6396	**0.2103 ± 0.0005**	**0.0112 ± 0.0001**	0.085 ± 0.008	0.2274	0.2134 ± 0.0005	0.0126 ± 0.0000	**0.071 ± 0.002**
Public foil	0.1316	**0.0298 ± 0.0001**	0.0034 ± 0.0000	0.445 ± 0.018	0.0754	**0.0298 ± 0.0005**	**0.0034 ± 0.0000**	**0.348 ± 0.070**
Combustion	0.8919	0.7984 ± 0.0073	0.0073 ± 0.0002	0.628 ± 0.017	0.7918	**0.7750 ± 0.0030**	**0.0069 ± 0.0000**	**0.043 ± 0.017**

Table 24: **Cylinder: a quarter of the real trajectories matches full-data fine-tuning.** Identical trajectory subsets, three seeds per cell; each subset is drawn over the 92 staged cylinder trajectories, of which only the 72 seen-parameter ones carry training windows, so the row label gives the per-seed range of training trajectories. The full-data fine-tune (0.1629) is our own recipe and reproduces the released checkpoint (0.1632). The matched pipeline ties full-data fine-tuning at a quarter of the trajectories and beats it at half. The third column is not a data-efficiency claim: its operator saw all 72 training trajectories. Bold: best per row within the floor.

Training trajectories	Fine-tune alone	FNO + FROST, both on the fraction	FROST on the full-data FNO
12.5% (9–11)	0.1979 ± 0.0257	0.1858 ± 0.0204	**0.1662 ± 0.0003**
25% (18–19)	0.1754 ± 0.0032	0.1645 ± 0.0031	**0.1561 ± 0.0019**
50% (36–38)	0.1698 ± 0.0029	**0.1582 ± 0.0039**	**0.1568 ± 0.0020**
100% (72)	0.1629 ± 0.0004	**0.1571 ± 0.0003**	**0.1571 ± 0.0003**

Table 25: **Cylinder: one recipe brings six forecasters into one band.** Each corrector is trained independently against one released fine-tuned checkpoint (the CNO was trained from scratch on the real split because the released file is unloadable). Accuracy is compressed into a narrow band, 0.1571–0.1680 from an uncorrected 0.1632–0.3238. SDE-30 is the 128-window read.

Frozen forecaster	Uncorrected			FROST, ODE-6 (3 seeds)			SDE-30	Gain
	Rollout	RMS ratio	CRPS	Rollout	RMS ratio	CRPS	Rollout	
FNO (ours)	0.1632	1.236	0.0220	0.1571 ± 0.0003	1.044 ± 0.031	0.0177 ± 0.0001	0.1538 ± 0.0002	4%
DPOT-S	0.1744	0.712	0.0234	0.1588 ± 0.0014	0.725 ± 0.017	0.0178 ± 0.0001	0.1542 ± 0.0008	9%
U-Net	0.1817	0.673	0.0243	0.1614 ± 0.0015	0.698 ± 0.004	0.0190 ± 0.0001	0.1548 ± 0.0007	11%
CNO (trained from scratch)	0.2239	0.757	0.0284	0.1680 ± 0.0030	0.794 ± 0.011	0.0185 ± 0.0003	0.1568 ± 0.0014	25%
DeepONet	0.2341	0.665	0.0323	0.1597 ± 0.0029	0.779 ± 0.006	0.0184 ± 0.0003	0.1551 ± 0.0016	32%
Transolver	0.3238	0.412	0.0404	0.1651 ± 0.0024	0.661 ± 0.012	0.0194 ± 0.0001	0.1574 ± 0.0009	49%

Table 26: **Public foil: the correction improves six forecasters that are already stronger than ours.** The five released checkpoints are stronger than our fine-tuned FNO in rollout error, and the correction still lowers the error and the CRPS of all six, bringing the error into 0.029–0.033. ‡: the uncorrected Transolver read covers 160 of the 320 windows.

Frozen forecaster	Uncorrected		FROST, ODE-6			Gain
	Rollout	CRPS	n	Rollout	CRPS	
U-Net	0.0368	0.0053	3	0.0290 ± 0.0002	0.0033 ± 0.0000	21%
DPOT-S	0.0357	0.0066	3	0.0334 ± 0.0001	0.0049 ± 0.0000	6%
FNO (released fine-tune)	0.0421	0.0079	3	0.0297 ± 0.0001	0.0033 ± 0.0000	30%
Transolver‡	0.0548	0.0115	3	0.0318 ± 0.0005	0.0044 ± 0.0001	42%
CNO	0.0659	0.0129	3	0.0295 ± 0.0002	0.0035 ± 0.0001	55%
FNO (ours)	0.0754	0.0136	3	0.0298 ± 0.0005	0.0034 ± 0.0000	60%

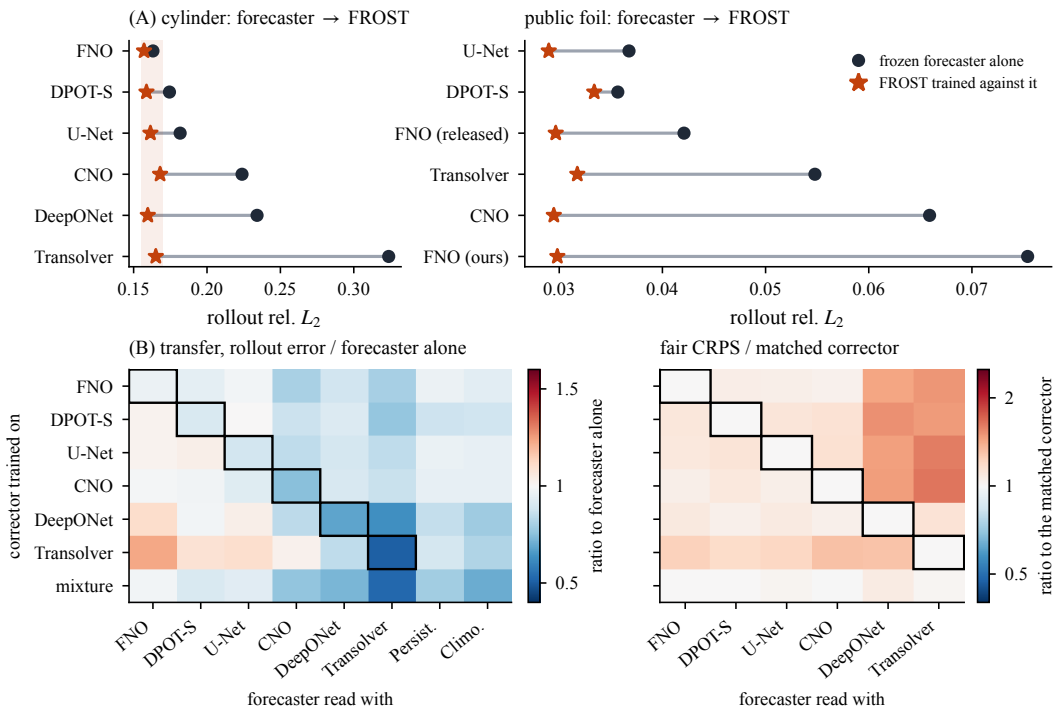

Figure 8: **One recipe corrects every forecaster; a mixture-trained corrector transfers. (A)** Rollout error of each frozen forecaster (dot) and of FROST trained against it (star) on cylinder and public foil; the band is the corrected cylinder range ± one floor. **(B)** Transfer on cylinder: a corrector trained against the row's forecaster is read with the column's forecaster, in rollout error and fair CRPS. Colour is the ratio to the column's forecaster alone for rollout error and to the matched corrector for CRPS (blue better, red worse); boxes mark matched training; the last row is the mixture corrector. Three seeds per cell.

on average and within 0.011 at worst (the DeepONet), improves the CNO it never saw, and degrades gracefully when handed persistence or climatology instead of a forecast (Figure 8B). The CNO corrector improves every column. What a single-forecaster corrector learns is the error geometry of its own forecaster, and the transfer matrix shows this with three regularities. The matched corrector is best or tied in every column. Off the diagonal the effect is triangular along forecast quality: a corrector trained against a good forecaster improves a worse one, though not to the matched level, and a corrector trained against a weak forecaster over-corrects a strong one. The cost tracks the gap in forecast error (Spearman 0.73 over the twenty cross cells) rather than architectural kinship: the FNO–Transolver pair is antisymmetric, one direction helping and the other over-correcting, while DPOT-S and the U-Net, almost identical in forecast error, exchange correctors at the same cost in either direction. Refitting the residual scale on the target forecaster leaves the gap in place (Table 27), so what is forecaster-specific is the shape of the correction, not its amplitude; in fair CRPS the matrix has the same structure (Table 28).

E.4 Transfer: residual-scale controls and CRPS

Table 27 tests whether the residual scale explains the transfer gap. Refitting the scale on the target forecaster changes the FNO-trained corrector's transfer to DPOT-S and the U-Net by less than 0.001 and moves its transfer to DeepONet and the Transolver further from the matched level (0.2077 to 0.2665; 0.2625 to 0.3433); for the weak-forecaster correctors read with the FNO it recovers part of the gap (Transolver corrector 0.2021 to 0.1735) without reaching the FNO's own 0.1632, and a pooled scale sits between the two. The shape of the correction, not its amplitude, is what a corrector learns from its forecaster. The stochastic read lowers the transferred errors by up to 0.011. Table 28 repeats the accuracy matrix of Table 5 in fair CRPS: CRPS follows accuracy, and the best cell of every column is the matched corrector or the mixture.

Table 27: **Refitting the residual scale does not close the transfer gap.** For each off-diagonal pair the residual scale s is either the one stored with the corrector, refitted on the target forecaster's training residuals, or pooled across the two; the last column re-reads the stored-scale transfer with the stochastic sampler. Refitting brings no pair to the matched level, so what is forecaster-specific is the shape of the correction and not its amplitude. Cells marked – were not run.

Pair (seed 0)	Stored scale	Scale refitted on target	Pooled scale	SDE-30 read (128 windows)
FNO corrector → DPOT-S forecast	0.1631	0.1634	0.1944	0.1623
FNO corrector → U-Net forecast	0.1786	0.1795	0.2100	0.1706
FNO corrector → DeepONet forecast	0.2077	0.2665	0.2273	0.2024
FNO corrector → Transolver forecast	0.2625	0.3433	0.2790	0.2716
DPOT-S corrector → FNO forecast	0.1670	0.1670	0.1791	0.1602
U-Net corrector → FNO forecast	0.1677	0.1681	0.1903	0.1639
DeepONet corrector → FNO forecast	0.1823	0.1681	0.1768	0.1742
Transolver corrector → FNO forecast	0.2021	0.1735	0.1896	0.1912
CNO corrector → CNO forecast (matched)	0.1692	0.1710	–	–

Table 28: **Cylinder transfer matrix in fair CRPS.** Rows are the forecaster the corrector was trained against and columns the forecaster it is read with, over the six forecasters of Table 5; that table's persistence and climatology columns are not repeated here. The first row is each forecaster left uncorrected, a fixed checkpoint; every corrected cell is three seeds (SDs omitted for legibility). Underlined: matched training; bold: best in the column. The mixture's CRPS is within 0.0014 of the matched corrector on every column, and within 0.0005 on five of the six.

Corrector trained on \ read with	FNO	DPOT-S	U-Net	CNO	DeepONet	Transolver
Forecaster alone	0.02201	0.02337	0.02432	0.02835	0.03229	0.04042
FNO	**0.01772**	0.01898	0.02010	0.01921	0.02647	0.02902
DPOT-S	0.01961	**0.01784**	0.02117	0.02103	0.02806	0.02855
U-Net	0.01949	0.02009	**0.01901**	0.02118	0.02678	0.03104
CNO	0.01867	0.01960	0.02000	0.01848	0.02702	0.03169
DeepONet	0.01941	0.01863	0.02060	0.01940	**0.01842**	0.02200
Transolver	0.02186	0.02117	0.02304	0.02424	0.02392	**0.01940**
Five-way mixture	0.01778	0.01790	0.01950	**0.01842**	0.01982	0.01974

E.5 Skill and calibration against the sampling budget

Our scoring follows the probabilistic-forecast literature: proper scoring rules (Gneiting & Raftery, 2007; Gneiting et al., 2007), the fair CRPS for finite ensembles (Ferro, 2014) and the corrected spread–skill ratio (Fortin et al., 2014). The sampling budget is a calibration dial, and the cheap ways of giving the FNO an ensemble show what the dial replaces (Table 29). A deep ensemble of three fine-tuned FNOs is as accurate as one and puts only 20–29% of the measurements inside its nominal 90% interval on the three fluid systems where it was run: fine-tuning seeds agree with one another far more than with the experiment. MC dropout buys coverage with accuracy along a fixed frontier, and the CRPS-trained dropout surrogate is the same trade with training. The Gaussian null's coverage of 0.79–0.93 is bought the same way: white noise of the right variance covers anything, and its members are less accurate than the forecast they perturb. FROST at six steps is the accuracy setting; the thirty-step stochastic read of the same checkpoint, scored on a 128-window subset, raises Gaussian coverage to 0.78–0.94 and lowers CRPS on every system, and sixteen members at that read reach 0.1487 on cylinder. Six steps for accuracy, thirty for spread, from one trained model; the full ladder of steps and members is in Appendix E.6.

E.6 Members and steps: the sampling ladder

Table 30 and Figure 9 read the main cylinder model at every combination of members and steps we ran, three seeds each. Along the members axis at six steps, $K = 1$, 4 and 16 give 0.1658, 0.1571 and 0.1550 (the sixteen-member read on its 128-window subset): most of the ensemble-mean gain over a single member is in the first four members. Along the steps axis with four members, one, two and six Euler steps give 0.1695, 0.1622 and 0.1571 and CRPS 0.0198, 0.0190 and 0.0177. The

Table 29: **The sampling budget is a calibration dial: six steps for accuracy, thirty for spread.** Calls are corrector evaluations per member per block; 'op.' marks methods whose only cost is the operator call every ensemble makes per member per block. Spread/skill is the small-ensemble-corrected ratio (ideal 1); coverage is that of the Gaussian-fitted 90% interval (ideal 0.90; the cell is empty where the fitted interval was not recorded). Bold: best rollout and CRPS per system (within the floor) among three-seed rows on the full window set. Rows marked 128 windows are the stochastic and sixteen-member reads on their subset and are not bolded; rows absent on a system were not run there.

Method	n	Calls	Rollout	CRPS	Spread/skill	Cov. 90%
Cylinder						
Deep ensemble of 3 fine-tuned FNOs†	1	0 (op.)	0.1625	0.02035	0.144	0.286
MC dropout 0.05 ($K = 4$)†	1	0 (op.)	0.2826	0.02876	0.339	0.772
MC dropout 0.1 ($K = 4$)†	1	0 (op.)	0.3561	0.03402	0.500	0.805
MC dropout 0.2 ($K = 4$)†	1	0 (op.)	0.6779	0.05497	0.802	0.858
CRPS-trained dropout ensemble†	1	0 (op.)	0.5179	0.11064	0.200	0.606
Gaussian null†	1	0 (op.)	0.1687	0.02011	0.129	0.849
FROST, ODE-1	3	1	0.1695 ± 0.0047	0.01977 ± 0.00025	0.067 ± 0.004	0.471 ± 0.015
FROST, ODE-2	3	2	0.1622 ± 0.0003	0.01902 ± 0.00010	0.071 ± 0.001	–
FROST, ODE-6	3	6	$\mathbf{0.1571 \pm 0.0003}$	$\mathbf{0.01772 \pm 0.00010}$	0.116 ± 0.002	0.696 ± 0.011
FROST, SDE-30 (128 windows)	3	30	0.1538 ± 0.0002	0.01518 ± 0.00009	0.205 ± 0.001	0.825 ± 0.002
FROST, $K = 16$, ODE-6 (128 windows)	3	6	0.1550 ± 0.0002	0.01633 ± 0.00008	0.118 ± 0.001	0.626 ± 0.009
FROST, $K = 16$, SDE-30 (128 windows)	3	30	0.1487 ± 0.0005	0.01521 ± 0.00011	0.188 ± 0.003	0.796 ± 0.000
Contr. cyl.						
Deep ensemble of 3 fine-tuned FNOs†	1	0 (op.)	0.1724	0.01203	0.082	0.253
MC dropout 0.1 ($K = 4$)†	1	0 (op.)	0.4448	0.02279	0.860	0.860
CRPS-trained dropout ensemble†	1	0 (op.)	0.2646	0.01610	0.416	0.717
Gaussian null†	1	0 (op.)	0.1824	0.01033	0.498	0.817
FROST, ODE-2	3	2	0.1548 ± 0.0019	0.00980 ± 0.00015	0.236 ± 0.006	–
FROST, ODE-6	3	6	$\mathbf{0.1490 \pm 0.0010}$	$\mathbf{0.00826 \pm 0.00008}$	0.458 ± 0.008	0.702 ± 0.003
FROST, SDE-30 (128 windows)	3	30	0.1486 ± 0.0005	0.00717 ± 0.00002	0.717 ± 0.001	0.834 ± 0.001
FSI						
MC dropout 0.1 ($K = 4$)†	1	0 (op.)	0.2500	0.01412	0.742	0.829
CRPS-trained dropout ensemble†	1	0 (op.)	0.2426	0.01452	0.710	0.764
Gaussian null†	1	0 (op.)	0.2298	0.01415	0.422	0.791
FROST, ODE-2	3	2	0.2136 ± 0.0008	0.01401 ± 0.00004	0.163 ± 0.001	0.459 ± 0.001
FROST, ODE-6	3	6	$\mathbf{0.2134 \pm 0.0005}$	$\mathbf{0.01264 \pm 0.00004}$	0.339 ± 0.001	0.692 ± 0.002
FROST, SDE-30 (128 windows)	3	30	0.2192 ± 0.0002	0.01173 ± 0.00002	0.470 ± 0.001	0.781 ± 0.001
Public foil						
Deep ensemble of 3 fine-tuned FNOs†	1	0 (op.)	0.0761	0.01220	0.190	0.203
MC dropout 0.1 ($K = 4$)†	1	0 (op.)	0.1020	0.01277	0.862	0.928
CRPS-trained dropout ensemble†	1	0 (op.)	0.0571	0.00874	0.506	0.689
Gaussian null†	1	0 (op.)	0.0726	0.01067	0.403	0.790
FROST, ODE-2	3	2	0.0301 ± 0.0011	0.00374 ± 0.00013	0.382 ± 0.010	0.642 ± 0.009
FROST, ODE-6	3	6	$\mathbf{0.0298 \pm 0.0005}$	$\mathbf{0.00339 \pm 0.00002}$	0.508 ± 0.006	0.798 ± 0.006
FROST, SDE-30 (128 windows)	3	30	0.0292 ± 0.0004	0.00300 ± 0.00007	0.640 ± 0.025	0.873 ± 0.002
Combustion						
Deep ensemble of 3 fine-tuned FNOs†	1	0 (op.)	0.7770	0.00747	0.336	0.854
MC dropout 0.1 ($K = 4$)†	1	0 (op.)	0.7766	0.00779	0.301	0.882
CRPS-trained dropout ensemble†	1	0 (op.)	0.8189	0.00814	0.169	0.867
Gaussian null†	1	0 (op.)	0.8943	0.00983	0.429	0.926
FROST, ODE-2	3	2	0.7874 ± 0.0055	0.00789 ± 0.00005	0.164 ± 0.010	0.840 ± 0.043
FROST, ODE-6	3	6	$\mathbf{0.7750 \pm 0.0030}$	$\mathbf{0.00694 \pm 0.00003}$	0.349 ± 0.008	0.907 ± 0.009
FROST, SDE-30 (128 windows)	3	30	0.7673 ± 0.0048	0.00639 ± 0.00003	0.487 ± 0.005	0.939 ± 0.000

Table 30: **Cylinder: the members-and-steps ladder of one trained FROST model.** Three seeds; calls are corrector evaluations per member per block; coverage is that of the Gaussian-fitted 90% interval. Reads on 128 windows are the stochastic and sixteen-member subset and are not directly comparable with the 320-window rows.

Read	Calls	Windows	Rollout	CRPS	Spread/skill	Cov. 90%	Energy err.
$K = 1$, ODE-1	1	320	0.1717 ± 0.0050	0.02159 ± 0.00035	0.000 ± 0.000	–	0.132 ± 0.020
$K = 1$, ODE-6	6	320	0.1658 ± 0.0009	0.02098 ± 0.00014	0.000 ± 0.000	–	0.090 ± 0.067
$K = 4$, ODE-1	1	320	0.1695 ± 0.0047	0.01977 ± 0.00025	0.067 ± 0.004	0.471 ± 0.015	0.131 ± 0.020
$K = 4$, ODE-2	2	320	0.1622 ± 0.0003	0.01902 ± 0.00010	0.071 ± 0.001	–	0.107 ± 0.051
$K = 4$, ODE-6 (main recipe)	6	320	0.1571 ± 0.0003	0.01772 ± 0.00010	0.116 ± 0.002	0.696 ± 0.011	0.090 ± 0.065
$K = 4$, SDE-30	30	128	0.1538 ± 0.0002	0.01518 ± 0.00009	0.205 ± 0.001	0.825 ± 0.002	0.164 ± 0.072
$K = 16$, ODE-6	6	128	0.1550 ± 0.0002	0.01633 ± 0.00008	0.118 ± 0.001	0.626 ± 0.009	0.175 ± 0.067
$K = 16$, SDE-30	30	128	0.1487 ± 0.0005	0.01521 ± 0.00011	0.188 ± 0.003	0.796 ± 0.000	0.162 ± 0.071

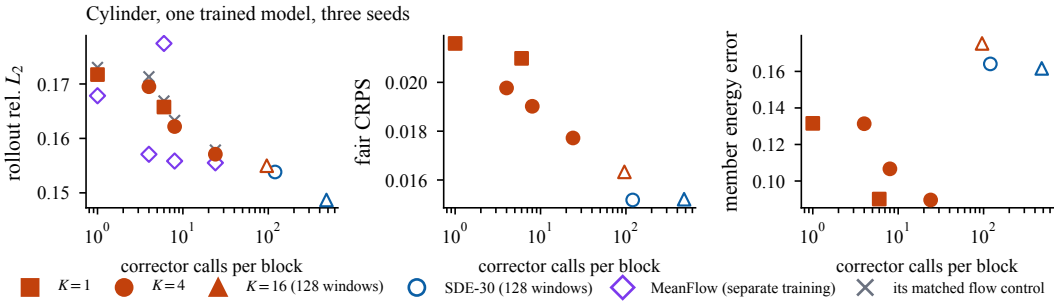

Figure 9: **One trained cylinder corrector read at eight budgets.** Rollout error, fair CRPS and member energy error against corrector calls per block for $K \in \{1, 4, 16\}$ members and 1, 2, 6 ODE steps or 30 SDE steps; open symbols are the 128-window reads. Accuracy and CRPS improve monotonically with budget, and the six-step read keeps the lowest member energy error.

stochastic thirty-step read lowers both further (0.1538, 0.0152) and nearly doubles the spread/skill ratio (0.116 to 0.205); with sixteen members it reaches 0.1487. Member energy error is lowest at the six-step read (0.090), the setting the main results use. Table 31 shows that the thirty-step read calibrates almost identically whichever cylinder forecaster the corrector was trained against (CRPS 0.015–0.016, spread/skill 0.21–0.24, Gaussian coverage 0.83–0.85).

Table 31: **At the thirty-step read every cylinder corrector calibrates alike** (128 windows, three seeds, matched corrector for each forecaster; the mixture is read with the FNO forecast). Calibration is nearly identical across forecasters: CRPS 0.0151–0.0164, spread/skill 0.205–0.241, coverage 0.825–0.854.

Forecaster the corrector was trained on	n	Rollout	CRPS	Spread/skill	Cov. 90%	Energy err.
FNO (ours)	3	0.1538 ± 0.0002	0.01518 ± 0.00009	0.205 ± 0.001	0.825 ± 0.002	0.164 ± 0.072
DPOT-S	3	0.1542 ± 0.0008	0.01506 ± 0.00003	0.236 ± 0.005	0.830 ± 0.001	0.410 ± 0.020
U-Net	3	0.1548 ± 0.0007	0.01596 ± 0.00009	0.214 ± 0.007	0.830 ± 0.005	0.452 ± 0.004
CNO (trained from scratch)	3	0.1568 ± 0.0014	0.01509 ± 0.00018	0.236 ± 0.005	0.830 ± 0.004	0.314 ± 0.015
DeepONet	3	0.1551 ± 0.0016	0.01559 ± 0.00017	0.237 ± 0.003	0.836 ± 0.009	0.311 ± 0.011
Transolver	3	0.1574 ± 0.0009	0.01643 ± 0.00020	0.241 ± 0.009	0.854 ± 0.004	0.489 ± 0.010
Five-way mixture	3	0.1592 ± 0.0008	0.01571 ± 0.00010	0.241 ± 0.002	0.829 ± 0.013	0.347 ± 0.087

F STRUCTURE OF THE RESIDUAL (SECTION 4.4)

F.1 MEMBERS ARE FLOWS: FULL MEMBER STATISTICS

Table 32: **Members recover the measured amplitude on the systems where the operator's error is amplitude.** Physical statistics of individual members over the ten-block rollout. Each statistic is computed per member and averaged over members and seeds (a deterministic method has one member). Energy error is the relative error of fluctuation energy; RMS ratio is predicted over measured fluctuation amplitude (ideal 1); spectrum error is the relative L_1 distance between radially binned energy spectra; profile error is the mean-velocity-profile error (undefined for combustion). Bold: best per column within each system among full-window rows. The SDE-30 read is on a 128-window subset.

Method	n	Member rel. L_2	Energy err.	RMS ratio	Spectrum err.	Profile err.
Cylinder						
Fine-tuned FNO	1	0.1632	0.529	1.236	**0.089**	**0.329**
PhysGuard (rank-60 reimpl.)	3	0.1688 ± 0.0036	2.733 ± 1.471	1.906 ± 0.387	0.492 ± 0.267	0.415 ± 0.079
Pushforward fine-tuning	3	$\mathbf{0.1576 \pm 0.0001}$	0.412 ± 0.019	0.767 ± 0.012	0.314 ± 0.003	0.396 ± 0.002
Deterministic residual	3	0.1693 ± 0.0014	0.147 ± 0.098	1.070 ± 0.046	0.137 ± 0.021	0.336 ± 0.004
Gaussian null†	1	0.1877	0.540	1.241	0.089	0.329
FROST, ODE-6	3	0.1656 ± 0.0004	$\mathbf{0.090 \pm 0.065}$	$\mathbf{1.044 \pm 0.031}$	0.145 ± 0.013	0.331 ± 0.005
FROST, SDE-30 (128 windows)	3	0.1736 ± 0.0001	0.164 ± 0.072	1.079 ± 0.034	0.115 ± 0.011	0.322 ± 0.005
Contr. cyl.						
Fine-tuned FNO	1	0.1728	0.166	0.913	0.036	0.185
PhysGuard (rank-60 reimpl.)	3	0.1745 ± 0.0001	0.179 ± 0.007	0.906 ± 0.004	0.033 ± 0.006	0.186 ± 0.001
Pushforward fine-tuning	3	$\mathbf{0.1562 \pm 0.0004}$	0.175 ± 0.008	0.908 ± 0.004	0.022 ± 0.010	$\mathbf{0.162 \pm 0.001}$
Deterministic residual	3	0.1624 ± 0.0008	0.136 ± 0.021	0.930 ± 0.011	0.021 ± 0.004	0.182 ± 0.002
Gaussian null†	1	0.2100	0.107	0.945	0.039	0.187
FROST, ODE-6	3	0.1602 ± 0.0007	0.129 ± 0.010	0.933 ± 0.005	$\mathbf{0.019 \pm 0.003}$	0.174 ± 0.002
FROST, SDE-30 (128 windows)	3	0.1726 ± 0.0005	0.083 ± 0.008	0.958 ± 0.004	0.011 ± 0.003	0.186 ± 0.003
FSI						
Fine-tuned FNO	1	0.2274	0.096	0.951	0.041	0.053
PhysGuard (rank-60 reimpl.)	3	0.2263 ± 0.0007	0.077 ± 0.017	0.961 ± 0.009	0.043 ± 0.008	0.051 ± 0.001
Pushforward fine-tuning†	1	0.2429	0.081	0.959	0.086	0.109
Deterministic residual	3	$\mathbf{0.2172 \pm 0.0001}$	0.106 ± 0.007	0.945 ± 0.004	0.025 ± 0.002	$\mathbf{0.044 \pm 0.000}$
Gaussian null†	1	0.2471	0.046	0.977	0.035	0.054
FROST, ODE-6	3	0.2242 ± 0.0004	0.071 ± 0.002	0.964 ± 0.001	$\mathbf{0.021 \pm 0.001}$	0.044 ± 0.000
FROST, SDE-30 (128 windows)	3	0.2383 ± 0.0001	0.060 ± 0.003	0.969 ± 0.002	0.018 ± 0.001	0.042 ± 0.001
Public foil						
Fine-tuned FNO	1	0.0754	0.262	0.859	0.021	0.068
PhysGuard (rank-60 reimpl.)	3	0.0829 ± 0.0101	0.265 ± 0.054	0.857 ± 0.031	0.038 ± 0.006	0.067 ± 0.012
Pushforward fine-tuning†	1	0.1775	0.470	1.212	0.169	0.135
Deterministic residual	3	$\mathbf{0.0317 \pm 0.0006}$	0.429 ± 0.042	0.755 ± 0.028	$\mathbf{0.005 \pm 0.002}$	$\mathbf{0.009 \pm 0.002}$
Gaussian null†	1	0.0841	0.003	0.999	0.011	0.065
FROST, ODE-6	3	0.0331 ± 0.0007	0.348 ± 0.070	0.807 ± 0.043	0.007 ± 0.002	0.011 ± 0.000
FROST, SDE-30 (128 windows)	3	0.0339 ± 0.0004	0.272 ± 0.032	0.853 ± 0.018	0.003 ± 0.001	0.011 ± 0.002
Combustion						
Fine-tuned FNO	1	**0.7918**	**0.030**	**1.015**	0.270	–
PhysGuard (rank-60 reimpl.)	3	0.8259 ± 0.0089	0.125 ± 0.041	1.061 ± 0.019	0.361 ± 0.034	–
Pushforward fine-tuning	3	0.8465 ± 0.1718	0.585 ± 0.045	0.644 ± 0.035	0.496 ± 0.058	–
Deterministic residual	3	0.8061 ± 0.0040	0.077 ± 0.012	1.038 ± 0.006	0.298 ± 0.010	–
Gaussian null†	1	1.1371	0.038	1.019	0.133	–
FROST, ODE-6	3	0.8115 ± 0.0055	0.043 ± 0.017	1.021 ± 0.009	0.200 ± 0.013	–
FROST, SDE-30 (128 windows)	3	0.8519 ± 0.0084	0.039 ± 0.002	1.019 ± 0.001	0.142 ± 0.002	–

Table 32 scores individual members on all five statistics. On cylinder the fine-tuned FNO is over-energetic, with 24% too much fluctuation amplitude; FROST's members bring the amplitude ratio to 1.04 and cut the energy error by a factor of six, while the ensemble mean stays the more accurate point forecast. Member energy also improves on controlled cylinder and FSI, and the mean-velocity profile improves on three systems. On foil and combustion the operator's error is dominated by bias rather than amplitude, and there the correction acts on the mean, which is where the 60% and 2% accuracy gains of Table 1 come from. The baselines are instructive by contrast: PhysGuard's

cylinder members are the most over-energetic in the table, and pushforward's are under-energetic on cylinder and far from the measurement on foil. FROST corrects the operator's amplitude error on the three systems where the discrepancy is amplitude-dominated, and its mean is more accurate on every system.

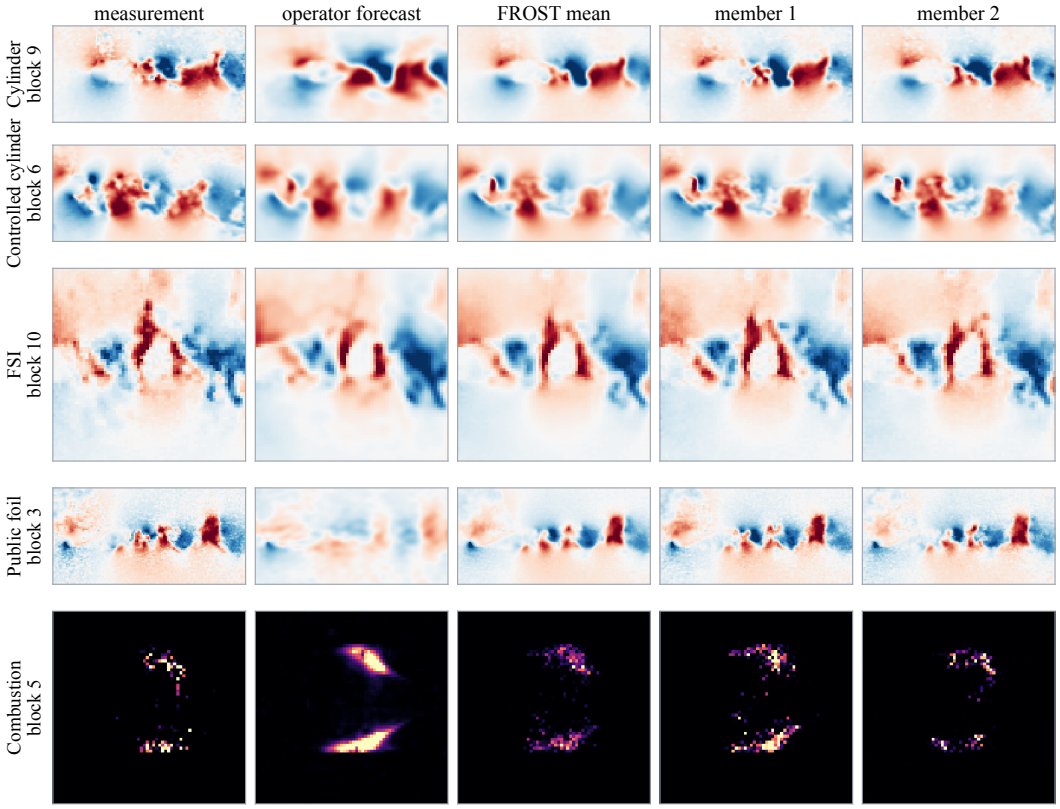

Figure 10: **The mean corrects the forecast and a member sharpens it.** One selected test frame per system: the measurement, the frozen operator's forecast, the FROST ensemble mean and two members (transverse velocity; chemiluminescence for combustion), one colour scale per row. Where the forecast has smoothed or misplaced the measured structure, the mean of the members corrects it and a single member sharpens and relocates the structure the forecast smoothed. Frames are chosen to show the correction at its clearest, from the first sixteen test windows, every block and five frames per block of the default read (seed zero, $K = 4$, six steps); they are illustrations, not scored quantities.

F.2 The matched correction controls

Table 34 gives the rollout error and fair CRPS of the three matched correction controls, the deterministic residual regressor, the PDE-Refiner-style denoiser and the Gaussian null, beside the FNO and FROST; Table 33 adds the Gaussian null's member statistics.

Table 33: The variance-matched Gaussian null exposes the distinction between total energy and useful correction. Same evaluation as the main comparison; the noise has one sampling seed, FROST three training seeds.

System	Forecast	Rollout	CRPS	Fluct.	RMS
Cylinder	FNO	0.1632	0.02201	0.529	1.236
Cylinder	Gaussian null	0.1687	0.02011	0.540	1.241
Cylinder	FROST	0.1571 ± 0.0003	0.01772 ± 0.00010	0.090 ± 0.065	1.044 ± 0.031
Controlled cyl.	FNO	0.1728	0.01279	0.166	0.913
Controlled cyl.	Gaussian null	0.1824	0.01033	0.107	0.945
Controlled cyl.	FROST	0.1490 ± 0.0010	0.00826 ± 0.00008	0.129 ± 0.010	0.933 ± 0.005

Continued on next page

System	Forecast	Rollout	CRPS	Fluct.	RMS
FSI	FNO	0.2274	0.01775	0.096	0.951
FSI	Gaussian null	0.2298	0.01415	0.046	0.977
FSI	FROST	0.2134 ± 0.0005	0.01264 ± 0.00004	0.071 ± 0.002	0.964 ± 0.001
Public foil	FNO	0.0754	0.01362	0.262	0.859
Public foil	Gaussian null	0.0726	0.01067	0.003	0.999
Public foil	FROST	0.0298 ± 0.0005	0.00339 ± 0.00002	0.348 ± 0.070	0.807 ± 0.043
Combustion	FNO	0.7918	0.00917	0.030	1.015
Combustion	Gaussian null	0.8943	0.00983	0.038	1.019
Combustion	FROST	0.7750 ± 0.0030	0.00694 ± 0.00003	0.043 ± 0.017	1.021 ± 0.009

Table 34: **FROST is ahead of every matched correction control in accuracy and CRPS on every system.** All three controls share FROST's network, condition and real training data. The deterministic residual regressor is trained by regression ($x_0 = 0, t = 1$), so it predicts the conditional mean of the residual; the PDE-Refiner-style denoiser iteratively refines the forecast under its own noise schedule; the Gaussian null adds white noise with the training-residual standard deviation to the forecast. Regressor and FROST: three seeds; †one seed. A deterministic forecast's CRPS is its absolute error. Bold: best per metric and system, within the 0.0018 floor for accuracy.

System	Rollout relative L_2					Fair CRPS				
	FNO	Regressor	Refiner†	Null†	FROST	FNO	Regressor	Refiner†	Null†	FROST
Cylinder	0.1632	0.1693 ± 0.0014	0.1762	0.1687	$\mathbf{0.1571 \pm 0.0003}$	0.0220	0.0219 ± 0.0003	0.0226	0.0201	$\mathbf{0.0177 \pm 0.0001}$
Contr. cyl.	0.1728	0.1624 ± 0.0008	0.1751	0.1824	$\mathbf{0.1490 \pm 0.0010}$	0.0128	0.0118 ± 0.0001	0.0124	0.0103	$\mathbf{0.0083 \pm 0.0001}$
FSI	0.2274	0.2172 ± 0.0001	0.2196	0.2298	$\mathbf{0.2134 \pm 0.0005}$	0.0177	0.0161 ± 0.0000	0.0161	0.0141	$\mathbf{0.0126 \pm 0.0000}$
Public foil	0.0754	0.0317 ± 0.0006	0.0341	0.0726	$\mathbf{0.0298 \pm 0.0005}$	0.0136	0.0049 ± 0.0001	0.0050	0.0107	$\mathbf{0.0034 \pm 0.0000}$
Combustion	0.7918	0.8061 ± 0.0040	0.8158	0.8943	$\mathbf{0.7750 \pm 0.0030}$	0.0092	0.0092 ± 0.0001	0.0092	0.0098	$\mathbf{0.0069 \pm 0.0000}$

G ADDITIONAL EXPERIMENTS

The experiments below vary one setting of the recipe at a time; the conventions are those stated at the head of the appendix.

G.1 WHERE THE REMAINING ERROR LIVES: AUTOREGRESSIVE INPUT SHIFT

Table 35: **The corrector's single-block skill holds over ten blocks once the history is right.** Resetting every block's history to the measurement (an oracle) returns the ten-block error to the single-block level and halves the CRPS or better. Three seeds; the reset reads use three replicate trainings of the same recipe.

System	Single block	Free rollout (ten blocks)		Measured-history reset (oracle)	
	rel. L_2	rel. L_2	CRPS	rel. L_2	CRPS
Cylinder	0.0667 ± 0.0001	0.1571 ± 0.0003	0.0177 ± 0.0001	0.0665 ± 0.0000	0.0081 ± 0.0000
Public foil	0.0139 ± 0.0001	0.0298 ± 0.0005	0.0034 ± 0.0000	0.0139 ± 0.0001	0.0017 ± 0.0000
FSI	0.0646 ± 0.0001	0.2134 ± 0.0005	0.0126 ± 0.0000	0.0622 ± 0.0001	0.0034 ± 0.0000

Almost all of FROST's ten-block error above its single-block error comes from the histories it generates for itself, not from its skill on any block. Our rollout protocol follows the stability studies of learned PDE surrogates (Li et al., 2022; McCabe et al., 2023; Sambamurthy & Chattopadhyay, 2025) and the benchmarking of autoregressive generative rollouts (Kohl et al., 2026); training on generated histories follows scheduled sampling and the pushforward trick (Bengio et al., 2015; Brandstetter et al., 2022; List et al., 2025), the reduction of imitation to its own state distribution (Ross et al., 2011) and, in a far narrower local form, self forcing (Huang et al., 2025); and the residual gap under distribution shift, with its transport-based alignment (Kim et al., 2026), frames why a fixed corrector is specific to its forecaster. The corrector is trained on measured histories and deployed on its own outputs; to price that shift we hold the trained corrector fixed and reset every block's history to the measurement, teacher forcing (Williams & Zipser, 1989) at test time and an oracle no forecast can use (Table 35; Figure 11). On cylinder, foil and FSI the ten-block error then returns to the single-block level, and CRPS falls in proportion; free rollout on the same models is between two and three and a half times higher. The oracle bounds what better histories could recover, and it locates the corrector's remaining error in the input distribution, not in the correction.

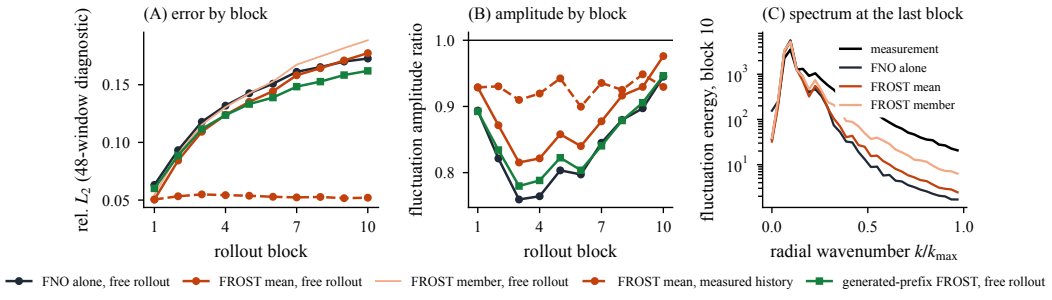

Figure 11: **The remaining error is in the history.** Cylinder, per block, 48-window diagnostic subset, seed zero. **(A)** Relative error of the FNO alone, of the FROST mean and one member under free rollout, of the FROST mean with the history reset to the measurement at every block, and of the generated-prefix corrector. **(B)** Member fluctuation amplitude on the same runs. **(C)** Radial fluctuation spectrum at the last block: the forecast damps the mid-band, a member restores it, the mean of members lies between. Diagnostic curves use an unchunked per-window error on a 48-window subset and are not comparable with the headline tables; on the full 320 windows the ten-block aggregates are 0.157 for FROST and 0.163 for the FNO.

The shift is addressable. Exposing the corrector to its own histories during training (Bengio et al., 2015; Brandstetter et al., 2022; Huang et al., 2025), by replacing the history of a second target block with the model's own detached two-step prediction of the first after a warm-up (Appendix G.2),

Table 36: **Exposure to generated histories lowers the cylinder rollout error at the cheap reads.** Three seeds each; the matched control uses the prefix experiment's 60-frame windows and second-block target with the prefix always taken from the measurement, so the only difference from the prefix rows is exposure to generated history. Δ in units of the 0.0018 floor; SDE-30 is the 128-window read.

Cylinder read	Ordinary training	Matched 60-frame control	Generated-prefix training	Δ vs. ordinary
ODE-2	0.1639 ± 0.0030	0.1665 ± 0.0033	0.1520 ± 0.0005	-0.0119 (-6.6 f)
ODE-6	0.1577 ± 0.0022	0.1580 ± 0.0015	0.1527 ± 0.0005	-0.0049 (-2.7 f)
SDE-30	0.1535 ± 0.0009	0.1536 ± 0.0007	0.1564 ± 0.0010	+0.0029 (+1.6 f)

lowers the cylinder rollout error by 6.6 floors at two steps and 2.7 at six, and stays within two floors of ordinary training at the thirty-step read (Table 36). A matched control that keeps the same window construction but always takes the prefix from the measurement is within 1.5 floors of ordinary training at every read, so the gain is the exposure and not the window construction. Corrupting the history or the forecast with noise does not reproduce the gain (Table 39), consistent with the shift being structured drift rather than noise, and doubling the horizon to twenty blocks keeps FROST ahead of the FNO on the three systems long enough to test it (Table 38; one seed). Changing the conditioning architecture barely moves the result (Appendix C.8); correcting the distribution of histories does. The exposure is reported as evidence about the shift, not as part of the method.

G.2 Generated-prefix training

Generated-prefix training uses windows of $T_{\text{in}} + 2T_{\text{out}}$ frames. It generates the first future block from the initial history and trains the corrector on the second block. The first 2,000 updates use the true intervening history; afterwards the prefix comes from the current corrector and the frozen operator with two Euler steps. Prefix generation is detached, and the loss stays local to the second block. The matched control uses the same windows and second-block target with the prefix always taken from the measurement (Table 36).

G.3 Cheap sampling and MeanFlow

Table 37: **One-step MeanFlow reaches the six-step ensemble mean on cylinder and pays for it in the members' fluctuation energy.** MeanFlow against the ordinary flow trained on every axis MeanFlow is forced onto (fp32, $\sigma = 0$, no compilation), three seeds each, and the main FROST recipe as a practical reference. Bold: best rollout per read within the floor.

Read	Flow, matched control (fp32, $\sigma = 0$)			MeanFlow (finite difference)			FROST ($\sigma = 0.15$, bf16)		
	Rollout	CRPS	Energy err.	Rollout	CRPS	Energy err.	Rollout	CRPS	Energy err.
1 member, 1 step	0.1730 ± 0.0071	0.0216 ± 0.0004	0.161 ± 0.037	$\mathbf{0.1678 \pm 0.0012}$	0.0217 ± 0.0002	0.281 ± 0.046	0.1717 ± 0.0050	0.0216 ± 0.0003	0.132 ± 0.020
4 members, 1 step	0.1713 ± 0.0073	0.0200 ± 0.0004	0.161 ± 0.037	$\mathbf{0.1571 \pm 0.0013}$	0.0181 ± 0.0002	0.281 ± 0.045	0.1695 ± 0.0047	0.0198 ± 0.0002	0.131 ± 0.020
4 members, 2 steps	0.1633 ± 0.0011	0.0191 ± 0.0002	0.130 ± 0.041	$\mathbf{0.1558 \pm 0.0007}$	0.0175 ± 0.0002	0.280 ± 0.059	0.1622 ± 0.0003	0.0190 ± 0.0001	0.107 ± 0.051
4 members, 6 steps	0.1579 ± 0.0006	0.0178 ± 0.0002	0.107 ± 0.044	$\mathbf{0.1555 \pm 0.0002}$	0.0172 ± 0.0001	0.255 ± 0.057	$\mathbf{0.1571 \pm 0.0003}$	0.0177 ± 0.0001	0.090 ± 0.065
1 member, 6 steps	$\mathbf{0.1669 \pm 0.0006}$	0.0211 ± 0.0002	0.106 ± 0.044	0.1774 ± 0.0011	0.0225 ± 0.0000	0.254 ± 0.057	$\mathbf{0.1658 \pm 0.0009}$	0.0210 ± 0.0001	0.090 ± 0.067

Reducing corrector calls is worthwhile only if the samples keep the properties the calls bought. Table 37 reads three cylinder models on one ladder of members and steps: the main FROST recipe; a residual MeanFlow corrector, which learns an interval-averaged velocity for one-step sampling (Geng et al., 2025a), here with the time derivative in its consistency target taken by finite differences rather than a Jacobian-vector product; and MeanFlow's matched control, the ordinary flow trained on every axis MeanFlow is forced onto (fp32, $\sigma = 0$, no compilation), so that the comparison is not confounded by precision, compilation or bridge noise. Three seeds each. Against its matched control, MeanFlow is more accurate at one and two steps for every ensemble size (four members, one step: 0.1571 against 0.1713) and behind only at one member with six steps (0.1774 against 0.1669); its one-step, four-member read equals six-step FROST on the ensemble mean. The members tell the two apart: energy error 0.25–0.28 for MeanFlow against 0.11–0.16 for the control at every read. On competition foil the control is the more accurate at every read (Table 41: 0.0844 against 0.0887 at six steps). One-step transport is therefore a cheap read of the ensemble mean on cylinder, and the six-step interpolant, which keeps the members' fluctuation statistics as well, is the recipe.

G.4 TWICE THE HORIZON

Table 38: **FROST stays ahead of the operator at twenty blocks on every system long enough to test it.** Twenty-block rollouts (400 measured frames) on the 64 test windows long enough to support them, one corrector seed. The ten-block columns repeat the headline for reference. FROST remains ahead of the FNO on all three systems at twenty blocks; the cylinder FNO's member energy error grows to 4 at that horizon while FROST's stays below 1.

System	Ten blocks (320 windows)		FNO, twenty blocks		FROST, twenty blocks (seed 0)		
	FNO	FROST	Rollout	Energy err.	Rollout	CRPS	Energy err.
Cylinder	0.1632	0.1571 ± 0.0003	0.2201	4.086	0.2106	0.02133	0.948
Contr. cyl.	0.1728	0.1490 ± 0.0010	0.2484	0.146	0.2023	0.01086	0.081
FSI	0.2274	0.2134 ± 0.0005	0.3089	0.014	0.2823	0.01628	0.014

Sixty-four test windows are long enough to support a twenty-block rollout (400 measured frames). Table 38 reads the FNO and one FROST seed on them for the three systems that have such windows. FROST stays ahead at twenty blocks on all three (0.2106 against 0.2201 on cylinder, 0.2023 against 0.2484 on controlled cylinder, 0.2823 against 0.3089 on FSI), and the gap widens on controlled cylinder and FSI relative to the ten-block gap. The cylinder FNO's member energy error grows from 0.53 at ten blocks to 4.1 at twenty, the over-energetic drift of Section 2.2 compounding, while FROST's members reach 0.95, under a quarter of the FNO's (one seed).

G.5 INPUT-NOISE AUGMENTATION

Table 39: **Cylinder: noise augmentation of the corrector's inputs during training.** Noise is added to the measured history, to the forecast, or to the prefix before the operator sees it, at the stated fraction of the residual scale. One seed per corruption; the uncorrupted reference is the three-seed main recipe. None reproduces the gain of generated-prefix training (Table 36); prefix noise is worse. Δ in units of the 0.0018 floor.

Training-time input corruption	n	Rollout	CRPS	Energy err.	Δ vs. none
None (main recipe)	3	0.1571 ± 0.0003	0.01772 ± 0.00010	0.090 ± 0.065	+0.0 f
History noise 0.3 during training†	1	0.1561	0.01757	0.060	-0.6 f
History noise 0.1 + forecast noise 0.1†	1	0.1581	0.01767	0.024	+0.6 f
History noise 0.3 + forecast noise 0.3†	1	0.1588	0.01785	0.032	+0.9 f
Forecast noise 0.3†	1	0.1613	0.01792	0.047	+2.4 f
Prefix noise 0.2 (before the operator)†	1	0.1620	0.01809	0.150	+2.7 f

Table 39 tests the natural alternative to generated-prefix training: corrupting the corrector's inputs with noise during training so that it sees imperfect histories. Noise on the measured history at 0.3 of the residual scale is a null (-0.6 floors, one seed), and noise on the forecast or on the prefix does not help either ($+2.7$ floors at most). None reproduces the generated-prefix gain of Table 36, which is consistent with the rollout shift being structured drift rather than noise.

G.6 BACKBONE PORT AND ARCHITECTURAL KNOBS

The corrector's U-Net was ported to guided-diffusion's ADM architecture (Dhariwal & Nichol, 2021) with identical parameter count and a bit-identical read of the main recipe's checkpoint, and each of ADM's knobs was then changed one at a time, three seeds each (Table 40). The port itself is within -0.2 floors of the main recipe, and every knob stays within two floors of it: attention at two resolutions is the largest change (-1.1 floors), and dropout, three residual blocks per level, 32 normalization groups and 1-2-3-4 channel multipliers are within one floor.

G.7 COMPETITION FOIL IN FULL

The competition release of the foil experiment has its own parameter grid, resolution and a trajectory-held-out protocol, and its results are kept apart from public foil. Table 41 scores ev-

Table 40: **The ADM backbone and each of its knobs keep the cylinder rollout error within two floors of the main recipe.** The port has the same 40.7 M parameters and reads the main recipe's checkpoint bit-identically; each knob changes one architectural choice, three seeds each. Δ in units of the 0.0018 floor.

Backbone variant	n	Rollout	CRPS	Energy err.	Δ vs. main
U-Net, main recipe	3	0.1571 ± 0.0003	0.01772 ± 0.00010	0.090 ± 0.065	+0.0 f
ADM port (identical parameters)	3	0.1567 ± 0.0014	0.01775 ± 0.00003	0.075 ± 0.043	-0.2 f
ADM, attention at resolutions 2 and 4	3	0.1552 ± 0.0012	0.01753 ± 0.00010	0.032 ± 0.043	-1.1 f
ADM, attention at resolution 4	3	0.1578 ± 0.0017	0.01767 ± 0.00016	0.071 ± 0.018	+0.4 f
ADM, dropout 0.1	3	0.1587 ± 0.0023	0.01794 ± 0.00018	0.138 ± 0.002	+0.9 f
ADM, 32 normalization groups	3	0.1563 ± 0.0017	0.01776 ± 0.00003	0.128 ± 0.056	-0.4 f
ADM, three residual blocks per level	3	0.1572 ± 0.0024	0.01773 ± 0.00017	0.050 ± 0.012	+0.1 f
ADM, channel multipliers 1-2-3-4	3	0.1579 ± 0.0033	0.01777 ± 0.00021	0.054 ± 0.016	+0.4 f

Table 41: **On competition foil the residual corrector is the most accurate arm, and generation without the forecast is six times worse** (304 held-out-trajectory windows, five blocks of twenty frames, $K = 4$, ODE-6). This dataset has a different parameter grid, resolution (32×64) and protocol from public foil and is never pooled with it. Bold: best rollout and CRPS among three-seed rows.

Model	n	Single block	Rollout	CRPS	Energy err.
Reference forecasters					
Organizer FNO, alone†	1	0.0555	0.0903	0.00569	0.706
Persistence†	1	0.1017	0.1414	0.00823	1.000
Climatology (window mean)†	1	0.0856	0.1214	0.00694	1.000
Correctors on the frozen organizer FNO (width 48 unless stated)					
Deterministic residual†	1	0.0499	0.0884	0.00523	0.420
Dirac transport control ($\sigma = 0$, source = forecast)†	1	0.0533	0.0910	0.00565	0.669
Direct generation of the field	3	0.0483 ± 0.0002	0.0851 ± 0.0004	0.00363 ± 0.00001	0.355 ± 0.008
Direct generation + Re, AoA via adaLN†	1	0.0480	0.0849	0.00358	0.344
Residual (FROST), width 48	3	0.0477 ± 0.0000	$\mathbf{0.0845 \pm 0.0005}$	0.00362 ± 0.00002	0.408 ± 0.016
Residual (FROST), width 32	3	0.0483 ± 0.0002	0.0847 ± 0.0007	0.00362 ± 0.00002	0.413 ± 0.000
Residual, condition = window mean and RMS only†	1	0.0570	0.0917	0.00418	0.555
Generator with no forecast (history only)†	2	0.2293 ± 0.0319	0.5660 ± 0.0092	0.02247 ± 0.00049	0.638 ± 0.085
Forecast-anchored bridge, source noise 0.5	3	0.0571 ± 0.0002	0.0944 ± 0.0007	0.00391 ± 0.00003	0.262 ± 0.032
Forecast-anchored bridge, source noise 1.0	3	0.0534 ± 0.0002	0.0905 ± 0.0006	0.00372 ± 0.00001	0.325 ± 0.017
Forecast-anchored bridge, source noise 2.0†	1	0.0516	0.0882	0.00362	0.318
MeanFlow (finite difference), width 32	3	0.0518 ± 0.0003	0.0887 ± 0.0002	$\mathbf{0.00358 \pm 0.00001}$	0.372 ± 0.004
MeanFlow (finite difference), width 48†	1	0.0514	0.0884	0.00358	0.393

Table 42: **No design choice moves the competition-foil error by more than one floor.** Same evaluator as Table 41. Bridge noise, residual scale, width, training length and backbone move the rollout error by at most 0.0018.

Variant	n	Rollout	CRPS	Energy err.
Bridge noise σ (width 48, fitted residual scale)				
$\sigma = 0$	3	0.0850 ± 0.0002	0.00364 ± 0.00001	0.409 ± 0.015
$\sigma = 0.05$†	1	0.0850	0.00363	0.410
$\sigma = 0.15$ (main recipe)	3	0.0845 ± 0.0005	0.00362 ± 0.00002	0.408 ± 0.016
$\sigma = 0.5$†	1	0.0857	0.00360	0.375
Residual scale				
Fixed at 1 instead of fitted per channel†	1	0.0856	0.00357	0.346
Width and training length				
Width 32	3	0.0847 ± 0.0007	0.00362 ± 0.00002	0.413 ± 0.000
Width 48, 40,000 updates†	1	0.0863	0.00365	0.374
Width 32, fp32, $\sigma = 0$, no compilation (MeanFlow control)	3	0.0844 ± 0.0001	0.00364 ± 0.00001	0.420 ± 0.005
Backbone (width 32)				
ADM port†	1	0.0844	0.00363	0.418
ADM, attention at resolution 4†	1	0.0858	0.00366	0.406
ADM, channel multipliers 1-2-3-4†	1	0.0852	0.00365	0.420

ery arm we trained on it against the organizers' FNO. The deterministic residual regressor barely moves the forecast (0.0884 against 0.0903), the Dirac transport control ($\sigma = 0$, source equal to the forecast) does not move it at all, and generation without the forecast is six times worse. Direct field generation, the residual corrector at widths 32 and 48, and the forecast-anchored bridge at source noise 1 or 2 lie within 0.006 of one another on accuracy (0.0845–0.0905), with the residual corrector best on accuracy and the bridge best on member energy (0.26–0.33 against 0.41). Replacing the forecast by its window statistics is worse than not correcting (0.0917). MeanFlow is 0.004 behind the flow at six steps. Table 42 varies the corrector's design: bridge noise σ from 0 to 0.5, a fixed rather than fitted residual scale, width 32 against 48, 40,000 against 20,000 updates, the ADM port and its knobs, and the fp32 control all lie within 0.002 of the main recipe.