TEST-TIME SCALING FOR PDE SURROGATES VIA SOLUTION-SPACE REFINEMENT FLOWS

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural PDE surrogates have emerged as powerful tools for accelerating scientific and engineering simulations, but are typically deployed as one-step predictors and remain largely decoupled from the test-time scaling paradigm that has driven recent advances in generative models and large language models. Inspired by flow-based learning and on-policy training, we introduce *Flow Corrector* (FOCO), a computationally-efficient general framework that improves frozen PDE surrogates through iterative test-time refinement. FOCO learns a shared corrector from self-generated on-policy states in solution space and stochastic detached rollouts, avoiding backpropagation through long correction trajectories while substantially reducing training computation and activation-memory requirements. Across global weather super-resolution and next-state prediction for 2D and 3D spatiotemporal PDE systems, FOCO consistently improves frozen surrogates and outperforms matched baselines, reducing normalized MSE by 78.6% on 3D magnetohydrodynamic turbulence and by 84.8% over a 20-step rollout on the Wave Layer benchmark. The learned corrector further transfers zero-shot across neural-operator architectures without retraining.

1 INTRODUCTION

Neural PDE surrogates accelerate scientific and engineering simulations by learning solution operators across physical parameters, initial conditions, and forcing terms (Li et al., 2021; Kovachki et al., 2023). However, their usual single-pass deployment provides no built-in mechanism for improving a prediction with additional computation. This limitation is particularly consequential for challenging physical systems where prediction errors can compound during autoregressive simulation.

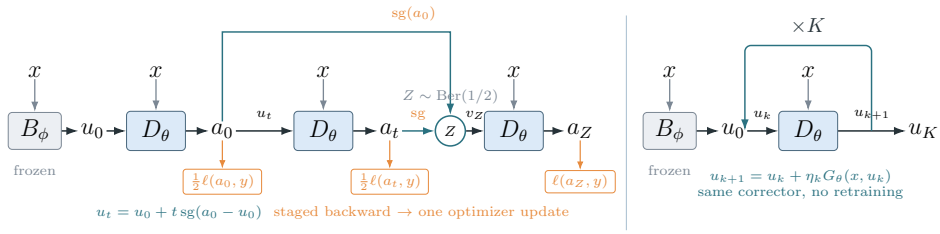
Classical numerical methods offer a natural perspective: *relaxed Picard iteration* repeatedly updates an approximation through a fixed-point map, enabling spending compute for iterative improvement (Kelley, 1995; Combettes & Yamada, 2015). A related principle motivates *test-time scaling* in generative and large language models, where additional sampling, refinement, or reasoning computation can improve outputs (Ma et al., 2025; Snell et al., 2024; Helbling et al., 2026). While recent PDE refiners and recurrent simulators explore this direction (Lippe et al., 2023; Abdul Majid et al., 2026; Liu et al., 2026), a key challenge is to efficiently learn a corrector that remains reliable on states generated by its own updates (e.g., without expensive differentiation through long trajectories).

We introduce *Flow Corrector* (FOCO), a post-hoc framework for test-time scaling via learned relaxed Picard iteration in solution space. Starting from a frozen surrogate prediction, a shared endpoint map repeatedly corrects the current estimate, with relaxation controlling the update strength and iteration count controlling inference computation. Our key idea is to *learn from self-generated refinement states without differentiating through their construction*. FOCO samples partial corrections and detached successors, supervises all outputs toward the same target, and accumulates gradients through staged backward passes. Training requires only three corrector evaluations per example and retains one corrector activation graph at a time, while the learned map supports substantially deeper inference without retraining. Empirically, FOCO outperforms a recent state-of-the-art post-hoc iterative-refinement method (Liu et al., 2026), especially on challenging 3D PDE dynamics, while substantially reducing training time and memory.

Our main contributions are:

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.

- **On-policy refinement-state training.** We introduce self-generated on-policy refinement-state training with detached successor supervision, enabling variable-depth inference while requiring only three corrector evaluations and no recurrent backpropagation through the refinement trajectory.
- **Fixed-point interpretation and guarantees.** We formulate FOCO as a learned relaxed Picard iteration, relate the training objective to target fixed-point consistency, and derive finite-budget error and cross-surrogate stability bounds under local contraction.
- **Substantial training-efficiency gains.** In capacity- and batch-matched comparisons with the strong recurrent baseline across three physical systems, FOCO achieves 2.00–2.14× faster training epochs while reducing peak allocated GPU memory by 66.2–76.4%.
- **Strong and transferable predictive performance.** Across five benchmarks, FOCO consistently improves frozen surrogates and outperforms matched baselines, including substantial gains on 3D MHD and long autoregressive Wave Layer rollouts. The same trained corrector supports refinement budgets up to 32 steps and transfers zero-shot across different PDE surrogate architectures within the same physical task.



(a) FOCO training (b) Test-time refinement

Figure 1: **FOCO training and test-time refinement.** B_ϕ is the frozen surrogate, G_θ predicts the residual correction, and during training $D_\theta(x, u) = u + G_\theta(x, u)$ gives the corrected estimate; sg denotes stop-gradient. In training, FOCO constructs self-generated partial states and detached successors, while staged backpropagation keeps each differentiable path one corrector evaluation deep. At test time, the shared corrector iteratively refines the frozen surrogate prediction for K steps.

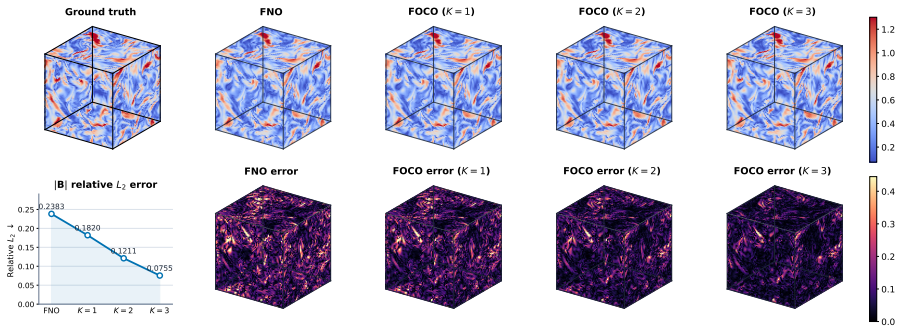


Figure 2: MHD refinement of the magnetic-field magnitude $|\mathbf{B}|$ on the test snapshot with the largest refinement step $K = 3$ gain. Top: ground truth, FNO, and FOCO predictions at $K = 1, 2, 3$. Bottom: physical relative L_2 error and matched absolute-error fields; color limits are shared across methods.

2 Related Work

Neural Operators for PDE Surrogates. Neural operators learn mappings between function spaces and provide efficient approximations to PDE solution operators (Kovachki et al., 2023). Architectures such as FNO and Transolver (Li et al., 2021; Wu et al., 2024) predict physical fields across varying parameters, initial and boundary conditions. These models are typically deployed as fixed, single-pass predictors, without a built-in mechanism to improve an individual prediction through additional

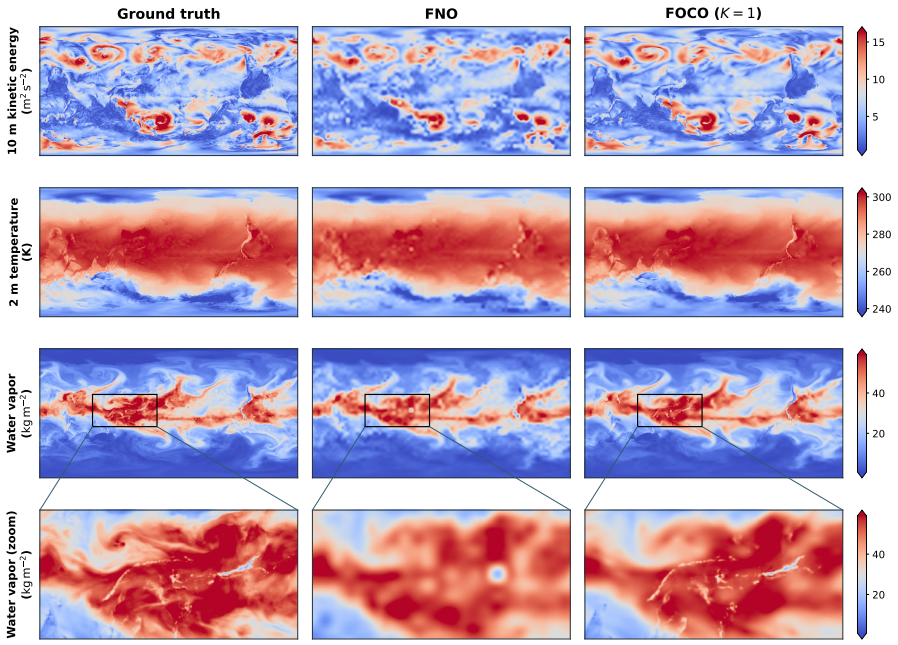


Figure 3: ERA5 16× super-resolution. Ground truth, FNO, and one-step FOCO refinement ($K = 1$) are shown for three physical fields. The bottom row magnifies a ground-truth-selected high-gradient water-vapor region; color limits are shared across methods within each row.

inference computation. FOCO is complementary: it leaves the pretrained surrogate frozen and augments it with a reusable solution-space corrector.

Test-Time Scaling in Generative Models and LLMs. Diffusion and flow-based models use iterative denoising or transport at inference time, exposing a computation–quality trade-off through the sampling procedure (Ho et al., 2020; Lipman et al., 2022). More broadly, generative models and LLMs can benefit from additional inference computation through sampling, search, verification, or iterative refinement (Ma et al., 2025; Snell et al., 2024). FOCO brings this principle to PDE surrogates by repeatedly refining the same solution estimate with fixed model parameters. Unlike conventional generative sampling, refinement starts from the prediction of a pretrained PDE surrogate rather than from a random reference distribution.

Iterative Refinement for PDE Surrogates. PDE-Refiner adapts diffusion-model denoising to PDE forecasting (Lippe et al., 2023), while RecurrSim varies recurrent latent depth to trade inference cost for accuracy (Abdul Majid et al., 2026). Most closely related, IRNO trains a shared corrector on frozen neural operators through recurrent refinement with spatial, spectral, and fixed-point losses (Liu et al., 2026). FOCO instead learns from self-generated partial states and detached successors using a unified endpoint objective, requiring only three corrector evaluations and no recurrent backpropagation. Further comparisons are provided in Appendix A.

3 METHODS

Let $B_\phi : \mathcal{X} \to \mathcal{U}$ be a frozen PDE surrogate, where $\mathcal{X}$ and $\mathcal{U}$ denote Banach spaces of PDE inputs and solution fields, and $u_0 = B_\phi(x)$ its prediction for target solution $y \in \mathcal{U}$. We learn a residual solution-space map:

$$D_\theta(x, u) = u + G_\theta(x, u), \qquad \ell(u, y) = \|u - y\|_{\mathcal{U}}^2. \tag{1}$$

At inference, D_θ acts as a learned fixed-point map in a relaxed Picard iteration,

$$u_{k+1} = (1 - \eta_k)u_k + \eta_k D_\theta(x, u_k) = u_k + \eta_k G_\theta(x, u_k), \qquad k = 0, \ldots, K - 1. \tag{2}$$

where $\eta_k \in [0, 1]$. Thus B_ϕ provides the initialization, while K controls the test-time computation budget. The central challenge is to train D_θ to remain accurate on states generated by its own iterations without differentiating through a long refinement trajectory.

3.1 On-Policy Refinement-State Training

Starting from u_0, we first compute $a_0 = D_\theta(x, u_0)$ and construct a detached partial refinement state:

$$u_t = u_0 + t\ \mathrm{sg}(a_0 - u_0), \qquad a_t = D_\theta(x, u_t), \qquad t \sim p(t). \tag{3}$$

u_t follows the correction direction predicted by the current model and retains its characteristic errors. We use a logit-normal $p(t)$ by default; t is used only to construct states and is not provided to the corrector. Ablation study over $t \sim \mathcal{U}[0, 1]$ is provided in Appendix F. To expose the model to its own successor states, we detach $v_0 = \mathrm{sg}(a_0)$ and $v_t = \mathrm{sg}(a_t)$, sample $Z \sim \mathrm{Bernoulli}(1/2)$, and set $v_Z = (1 - Z)v_0 + Zv_t$ and $a_Z = D_\theta(x, v_Z)$. The complete objective is simply

$$\mathcal{L}_{\mathrm{FOCO}} = \frac{1}{2}\ell(a_0, y) + \frac{1}{2}\ell(a_t, y) + \ell(a_Z, y). \tag{4}$$

All branches regress to the same endpoint y; no explicit fixed-point, spectral, contraction, or PDE-residual loss is required.

Staged optimization. The generated states are detached and treated as new training inputs. We backpropagate each of the three losses sequentially, release its graph, accumulate gradients, and update θ once. Training therefore uses three logical corrector evaluations, while every differentiable path contains only one corrector application. This avoids recurrent backpropagation and keeps live corrector-activation storage independent of the deployment refinement depth.

3.2 Test-Time Refinement

After training, both B_ϕ and G_θ are frozen and the same corrector can be applied for any budget K without retraining. For temporal PDEs, k in Eq. 2 indexes refinement of the same predicted state, not physical time. We use two complementary relaxation schedules:

$$\eta_k = \begin{cases} 1, & k = 0, 1, \\ q^{k-1}, & k \geq 2, \end{cases} \quad \text{or} \quad \eta_k = \frac{1}{K - k}. \tag{5}$$

The first is budget-independent and supports anytime stopping, we choose $q = 0.5$ by default; the second is budget-aware and follows from Euler discretization of the auxiliary relaxation dynamics $\dot{u} = D_\theta(x, u) - u$. The connection to Neural ODE (Chen et al., 2018) and Flow Matching/Rectified Flow (Lipman et al., 2022; Liu et al., 2023) is provided in Appendix A. The derivation and complete inference procedure are provided in Appendix C.4.

3.3 Theoretical Analysis

Eq. 2 is a learned relaxed Picard iteration for the fixed-point map $D_x(u) = D_\theta(x, u)$ (Kelley, 1995; Combettes & Yamada, 2015). We summarize its main properties below; all proofs and the Euler interpretation are given in Appendix C.

Implicit fixed-point consistency. Let $\mathcal{E}_x = \mathbb{E}_t\mathcal{L}_{\mathrm{end}}$, $\mathcal{S}_x = \mathbb{E}_{t,Z}\mathcal{L}_{\mathrm{roll}}$, and $\mathcal{R}_x = \mathcal{E}_x + \mathcal{S}_x$.

Proposition 3.1 (Loss-controlled consistency)**.** *If D_x is L-Lipschitz near y and the supervised successor states, then*

$$\delta_x := \|D_x(y) - y\| \leq L\sqrt{\mathcal{E}_x} + \sqrt{\mathcal{S}_x} \leq \sqrt{(1 + L^2)\mathcal{R}_x}.$$

Thus, endpoint and successor supervision control the target fixed-point residual without an explicit fixed-point loss; this implies approximate stationarity, not contractivity.

Stochastic supervision and training complexity. Let A denote one corrector's live activation storage, P parameter and optimizer storage, and S auxiliary state storage.

Proposition 3.2 (Staged training complexity)**.** *Conditioned on detached successor states, stochastic successor sampling is unbiased for the corresponding two-branch semi-gradient. FOCO uses three logical corrector evaluations with peak live storage $\mathcal{O}(A + P + S)$, versus $K_{\mathrm{tr}} + 1$ evaluations and $\mathcal{O}(K_{\mathrm{tr}}A + P + S)$ for ordinary stored-graph recurrent unrolling with a fixed-point evaluation.*

For matched $K_{\mathrm{tr}} = 6$ IRNO training, this reduces the corrector count from seven to three while keeping live activation storage independent of refinement depth; IRNO additionally supervises the recurrent trajectory with progressive spectral losses which requires extra computation.

Conditional refinement and transfer.

Theorem 3.3 (Finite-budget refinement)**.** *Suppose $D_x : \Omega \to \Omega$ is ρ-contractive on a closed convex $\Omega \subseteq \mathcal{U}$, with $0 < \rho < 1$, and $u_0, y \in \Omega$. Define*

$$Q_K = \prod_{k=0}^{K-1} [1 - \eta_k(1-\rho)] .$$

Then

$$\|u_K - y\| \le Q_K \|u_0 - y\| + (1 - Q_K)\frac{\delta_x}{1-\rho}.$$

Moreover, for two surrogate initializations $u_0, \tilde{u}_0 \in \Omega$,

$$\|u_K - \tilde{u}_K\| \le Q_K \|u_0 - \tilde{u}_0\|.$$

For full updates, $Q_K = \rho^K$; for $\eta_k = 1/(K-k)$, $Q_K = \mathcal{O}(K^{-(1-\rho)})$.

The bound separates initialization error from learned fixed-point bias and controls sensitivity to the base surrogate. Contractivity is a sufficient condition, not a property guaranteed by training; convergence to a biased fixed point need not monotonically improve target accuracy. Appendix C.8 further connects successor supervision to average error reduction and clarifies the roles of state coverage, relaxation, and late-stage over-correction.

4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate FOCO on five benchmarks spanning spatial super-resolution and two- and three-dimensional PDE dynamics: ERA5 $16\times$ global weather super-resolution from SuperBench (Ren et al., 2023; Chen et al., 2024; Hassan et al., 2023), Turbulent Radiative Layer (TR-2D), Active Matter, and three-dimensional magnetohydrodynamics (MHD) from The Well (Ohana et al., 2025), and Wave Layer from Poseidon dataset (Herde et al., 2024; Mousavi et al., 2025; Wen et al., 2026). For benchmarks shared with IRNO, we follow the same data splits, normalization, and primary evaluation protocols (Liu et al., 2026; Ohana et al., 2025). TR-2D, Active Matter, and MHD use four consecutive states to predict the next state, while Wave Layer uses the current wave field and static wave-speed coefficient. Dataset details are provided in Appendix D.1.

PDE Surrogate and Baselines. The main experiments use independently trained FNO, TFNO, and Transolver surrogates (Li et al., 2021; Kossaifi et al., 2023; Wu et al., 2024), which remain frozen during corrector training and evaluation. We reproduce IRNO (Liu et al., 2026) using the same frozen checkpoints and the same corrector architecture as FOCO, providing a controlled comparison under a common training and evaluation pipeline. Accordingly, all reported Base, IRNO, and FOCO results are obtained from our checkpoints; only the ResUNet-HFS (Khodakarami et al., 2025) and HiNOTE (Luo et al., 2024) results in Table 2 are quoted from Liu et al. (2026). Following IRNO, we use a lightweight U-Net corrector, but adopt a modernized design with residual and attention blocks. Exact architectures and training configurations are provided in Appendix D.2.

Metrics and Refinement Protocol. For ERA5, we report the SuperBench RFNE and ACC. For the remaining benchmarks, we report normalized mean squared error (NMSE) and variance-scaled root mean squared error (VRMSE). Unless stated otherwise, two-step FOCO refinement uses budget-independent refinement $\eta_0 = 1$ for $K = 1$ and $(\eta_0, \eta_1) = (1, 1)$ for $K = 2$ to match the constant-step refinement used by IRNO. For test-time scaling, we instead use budge-aware ODE refinement schedue as described in Section 3.2. Additional metric and evaluation details are provided in the Appendix D.3.

4.2 Main Results

Across all five benchmarks, FOCO consistently improves frozen PDE surrogates with only one or two correction steps. On ERA5 $16\times$ super-resolution, a single FOCO update already surpasses

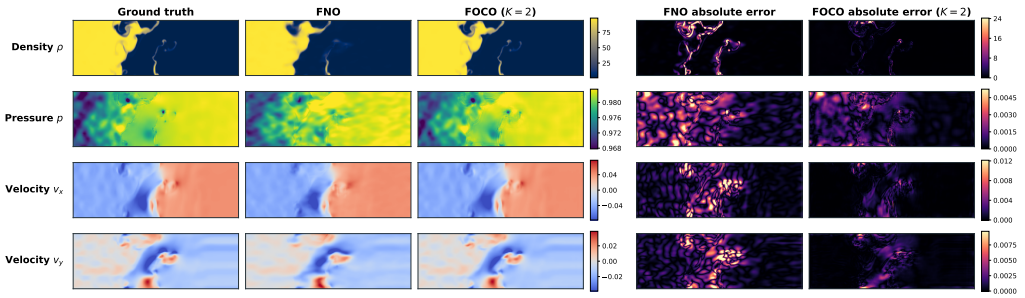


Figure 4: Teacher-forced one-step prediction on TR-2D. Two FOCO refinement steps ($K = 2$) improve the frozen FNO prediction and reduce the absolute error across all physical variables.

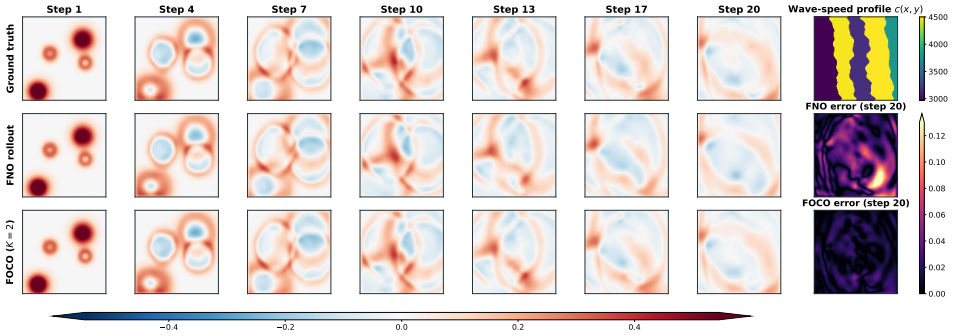


Figure 5: Autoregressive rollout on Wave Layer. Ground-truth, FNO, and FOCO predictions are shown at forecast steps 1, 4, 7, 10, and 20. FOCO applies two refinement updates ($K = 2$) after every surrogate prediction and feeds the corrected state into the next temporal transition. The final column shows the wave-speed profile $c(x, y)$ and the absolute errors at step 20.

the reproduced IRNO after six refinement steps, while recovering substantially sharper small-scale structure in all three weather variables (Figure 3). On the dynamical benchmarks, FOCO outperforms IRNO on all eight dataset–metric comparisons at $K = 1$ and on seven of eight at $K = 2$, with the remaining comparison essentially tied. Figure 4 provides a field-level view on TR-2D, where two refinement steps substantially reduce localized errors across density, pressure, and both velocity components. The advantage is particularly pronounced on the challenging 3D MHD system, where two corrections reduce the base-model NMSE by 78.6%. Figure 2 shows that refinement recovers magnetic structures that are largely absent from the single-pass prediction.

These results highlight an important distinction from recurrent refinement: FOCO does not need a long supervised training trajectory to obtain strong multi-step corrections. Its three depth-one training evaluations are sufficient to produce a reusable correction map that is already highly effective in the first few inference steps. We therefore emphasize $K = 1$ and $K = 2$ in the main comparison; longer-budget behavior is studied in Section 4.5 and Appendix G.

4.3 Autoregressive Rollout

A one-step corrector is useful for dynamical systems only if its outputs remain reliable when fed back into future predictions. We therefore place FOCO inside the autoregressive loop: after every frozen-surrogate prediction, two correction steps are applied before the corrected state is returned to the history.

The benefit persists throughout the rollout on all four dynamical systems, showing that FOCO suppresses rather than merely postpones error accumulation. The effect is especially strong for MHD and Wave Layer. On the full 20-step Wave Layer rollout, FOCO reduces terminal NMSE by 84.8%; visually, Figure 5 shows that the corrected rollout preserves the propagating wave structure long after

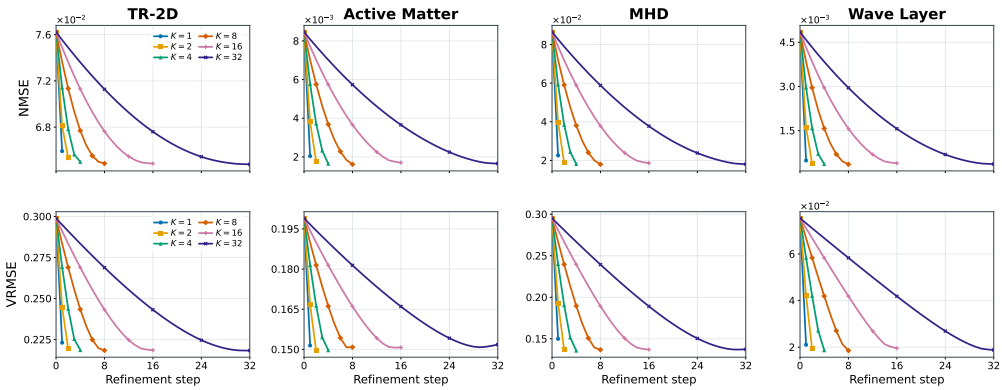


Figure 6: **Test-time scaling of FOCO across four physical systems.** Each curve shows the refinement trajectory for a test-time budget $K \in \{1, 2, 4, 8, 16, 32\}$. K denotes the number of corrector evaluations applied to the same prediction, not the physical forecast horizon. Lower NMSE and VRMSE indicate better performance.

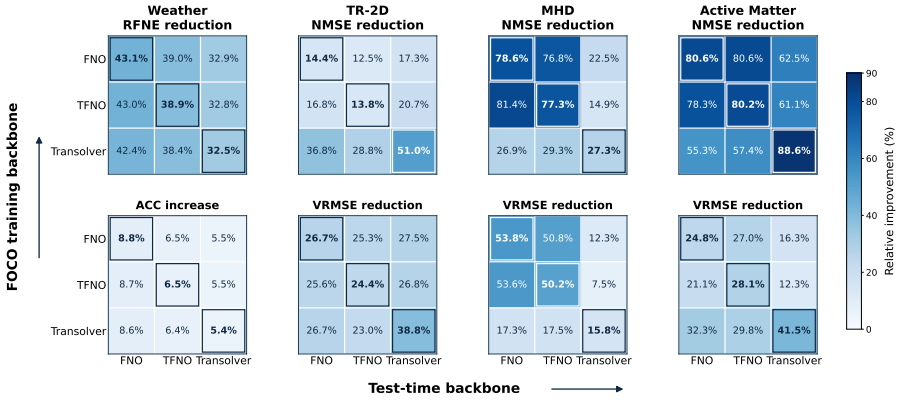


Figure 7: Zero-shot cross-operator transfer of FOCO across four physical systems. Rows indicate the frozen surrogate used to train FOCO, while columns indicate the surrogate corrected at test time. Bold outlined diagonal entries use matched training and test-time surrogates; off-diagonal entries evaluate transfer without retraining. Values report relative improvement over the corresponding uncorrected test-time surrogate, with higher values indicating better performance.

the uncorrected surrogate begins to lose fidelity. Thus, FOCO improves the evolving trajectory itself rather than only post-processing its terminal state. Additional result is reported in Appendix G.1.

4.4 Computational Efficiency

The predictive gains of FOCO do not come from a more expensive training procedure. We compare against a capacity- and batch-matched IRNO reproduction using the same frozen FNO, corrector architecture, data, precision, and activation checkpointing. Across TR-2D, Active Matter, and MHD, FOCO trains 2.00–$2.14\times$ faster per epoch while reducing peak allocated GPU memory by 66.2–76.4%.

The improvement follows directly from the training graph. FOCO uses three corrector evaluations whose differentiable depth is one, whereas the matched IRNO objective uses six recurrent refinement steps plus a fixed-point evaluation, with a maximum differentiable depth of six and additional spectral-loss computation. Consequently, FOCO uses 57.1% fewer logical corrector evaluations and eliminates recurrent backpropagation through the refinement trajectory. This reduction becomes particularly important for high-dimensional 3D PDEs, where storing recurrent solution-field activations is expensive. Full profiling details are given in Appendix E.

Table 1: Capacity- and batch-matched training efficiency. Batch denotes the per-GPU batch size. Epoch time includes data loading and optimization; peak memory is measured with `torch.cuda.max_memory_allocated`. Values are medians over three complete epochs after warm-up. Lower time and memory are better; the better raw measurements are shown in bold.

Dataset	Batch	Peak allocated memory (GiB)			Time per epoch (s)		
		IRNO	FOCO	Reduction	IRNO	FOCO	Speedup
TR-2D	32	20.99	**6.48**	69.15%	323.52	**161.91**	2.00×
Active Matter	32	30.93	**7.31**	76.36%	868.89	**412.66**	2.11×
MHD	1	2.98	**1.01**	66.15%	2024.64	**943.96**	2.14×

Table 2: ERA5 16× super-resolution results based on the FNO surrogate. K denotes the number of refinement steps. Improvements are measured relative to the base model (FNO). Best values are shown in bold.

Metric \ Method	ResUNet-HFS	HiNOTE	Base Model	IRNO ($K=1$)	IRNO ($K=2$)	IRNO ($K=6$)	FOCO (Ours, $K=1$)
ACC ↑	0.8915	0.9055	0.8539	0.9113	0.9171	0.9205	**0.9282**
Relative gain (%)	–	–	–	6.72%	7.39%	7.79%	**8.69%**
RFNE ↓	0.2253	0.2222	0.2584	0.1850	0.1718	0.1626	**0.1476**
Relative gain (%)	–	–	–	28.38%	33.50%	37.05%	**42.88%**

Table 3: Comparison of IRNO and FOCO across physical systems. Both NMSE and VRMSE are lower-is-better metrics. Relative gains at $K = 2$ are computed with respect to the corresponding FNO base-model error. The best one-step results are marked with **brown** color, and the best 2-step results are shown in **bold**.

Dataset	Metric	Base Model	Refinement ($K=1$)		Refinement ($K=2$)		Relative gain at $K=2$ (%)	
			IRNO	FOCO	IRNO	FOCO	IRNO	FOCO
TR-2D	NMSE	0.07623	0.06711	**0.06592**	0.06627	**0.06524**	13.07%	**14.41%**
	VRMSE	0.2989	0.2305	**0.2231**	0.2246	**0.2191**	24.87%	**26.69%**
Active Matter	NMSE	0.008459	0.002554	**0.002045**	0.001908	**0.001643**	77.44%	**80.57%**
	VRMSE	0.1990	0.1549	**0.1515**	**0.1495**	0.1497	**24.85%**	24.76%
MHD	NMSE	0.08645	0.03913	**0.02256**	0.03159	**0.01851**	63.46%	**78.59%**
	VRMSE	0.2951	0.2172	**0.1499**	0.1892	**0.1363**	35.90%	**53.81%**
Wave Layer	NMSE	0.004849	0.0005656	**0.0004845**	0.0003726	**0.0003650**	92.32%	**92.47%**
	VRMSE	0.07530	0.02261	**0.02103**	0.01877	**0.01869**	75.08%	**75.17%**

4.5 Test-Time Scaling

Here, test-time scaling means increasing the refinement budget K, i.e., applying more corrector evaluations to the *same* surrogate prediction, rather than extending the physical forecast horizon.

Although training uses only three non-recurrent corrector evaluations, the same frozen corrector can be deployed with much larger budgets. Figure 6 evaluates $K \in \{1, 2, 4, 8, 16, 32\}$ under the budget-aware schedule. Error generally decreases along each refinement trajectory, while larger budgets expose a clear accuracy–computation trade-off with diminishing returns. The corrector remains useful well beyond the training horizon. Additional large-budget and spectral analysis is provided in Appendix G.9.

4.6 Zero-Shot Cross-Operator Transfer

We further test whether FOCO learns a backbone-specific error corrector or a more general solution-space refinement rule. A corrector trained around one frozen surrogate, FNO, TFNO, or Transolver, is attached directly to another architecture at test time, with no retraining or fine-tuning.

Table 4: Autoregressive long-rollout performance of FOCO across physical systems. At every temporal transition, FOCO applies two correction steps to the prediction of the frozen surrogate. We report the terminal normalized MSE (NMSE) at rollout horizon t. Relative gain is calculated with respect to the corresponding uncorrected base rollout. Lower NMSE is better.

Dataset	Method	Rollout horizon							
		$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$	$t = 6$	$t = 7$	$t = 20$
TR-2D	Base	0.07623	0.1147	0.1507	0.1866	0.2224	0.2540	0.2851	–
	FOCO	0.06525	0.09769	0.1265	0.1549	0.1827	0.2109	0.2361	–
	Gain (%)	14.41%	14.86%	16.01%	17.00%	17.85%	16.96%	17.18%	–
Active Matter	Base	0.008459	0.02296	0.04633	0.08213	0.1335	0.2026	0.2888	–
	FOCO	0.001643	0.006242	0.01611	0.03369	0.06117	0.1006	0.1535	–
	Gain (%)	80.57%	72.81%	65.23%	58.98%	54.19%	50.36%	46.85%	–
MHD	Base	0.08646	0.2101	0.3706	0.6345	1.095	1.847	2.946	–
	FOCO	0.01851	0.05856	0.1252	0.2260	0.3622	0.5323	0.7360	–
	Gain (%)	78.59%	72.13%	66.21%	64.38%	66.93%	71.18%	75.02%	–
Wave Layer	Base	0.004849	0.01056	0.01598	0.02135	0.02668	0.03200	0.03724	0.06777
	FOCO	0.0003650	0.0009366	0.001574	0.002232	0.002891	0.003542	0.004162	0.01032
	Gain (%)	92.47%	91.13%	90.15%	89.55%	89.16%	88.93%	88.82%	84.77%

As shown in Fig 7, across all 48 off-diagonal dataset–metric comparisons, the transferred corrector improves the corresponding uncorrected surrogate. This indicates that FOCO's refinement behavior is not tied to the architecture used during training, but transfers across substantially different operator families, including Fourier-based models and Transolver. The result supports the view that FOCO learns a task-level correction rule in solution space rather than a backbone-specific transformation. We restrict this study to operators sharing the same physical task, discretization, and normalization; cross-physics transfer remains future work. Complete results and details are provided in Appendix G.8.

4.7 Ablation Study

We further examine the main design choices of FOCO. Logit-Normal sampling provides the strongest overall two-step performance, while self-generated states outperform oracle linear interpolation and remain more stable under deeper refinement, supporting the importance of matching the states encountered during iterative inference. Residual prediction is also substantially more effective than asking the lightweight corrector to directly reconstruct the full PDE solution. Finally, we relate FOCO to the single-shot residual-regression baselines studied in IRNO, further motivating iterative rather than one-shot correction. Complete ablations are provided in Appendix F.

5 Conclusion

We introduced *Flow Corrector* (FOCO), a post-hoc framework that turns frozen PDE surrogates into iteratively refinable predictors with controllable test-time computation. By learning from self-generated refinement states without recurrent backpropagation, FOCO substantially reduces training cost while supporting refinement budgets far beyond the training horizon. Across five benchmarks, FOCO consistently improves frozen surrogates, remains effective in long autoregressive rollouts, and transfers zero-shot across different surrogate architectures within the same physical task. Two limitations remain. First, the current method uses prescribed refinement schedules and budgets rather than adapting computation to individual predictions. Second, our transfer study is restricted to compatible surrogate architectures within the same physical task. Future work could explore adaptive stopping and step-size selection, as well as transfer across discretizations, geometries, and physical systems. More broadly, FOCO makes inference-time computation a controllable resource for scientific surrogate modeling.

AI USE STATEMENT

We used generative AI tools, including ChatGPT and Codex, during this work. They were used to provide feedback on experimental design, assist with intermediate mathematical derivations and proof development, debug and edit research code, search for and summarize relevant literature, and edit portions of the manuscript for clarity. All mathematical claims and proofs suggested with AI assistance were independently checked by the authors, AI-assisted code was verified through testing and inspection, and cited literature was manually verified against the original sources. The authors take full responsibility for the methods, experiments, claims, proofs, code, citations, and final manuscript.

REPRODUCIBILITY STATEMENT

We provide the complete training and inference algorithm, theoretical proofs, dataset and preprocessing details, evaluation protocols, model architectures, and training configurations in the appendix. To facilitate full reproducibility, we will publicly release the implementation, experiment configurations, and trained model checkpoints upon publication. All reported results use publicly available datasets and the evaluation procedures described in the paper.

REFERENCES

Harris Abdul Majid, Pietro Sittoni, and Francesco Tudisco. Test-Time Accuracy-Cost Control in Neural Simulators via Recurrent-Depth. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/a3b73551d62d7758bdd99fc018b167a9-Abstract-Conference.html.

B. Burkhart, S. Appel, S. Bialy, J. Cho, A. J. Christensen, D. Collins, C. Federrath, D. Fielding, D. Finkbeiner, A. S. Hill, J. C. Ibanez-Mejia, M. R. Krumholz, A. Lazarian, M. Li, P. Mocz, M.-M. Mac Low, J. Naiman, S. K. N. Portillo, B. Shane, Z. Slepian, and Y. Yuan. The Catalogue for Astrophysical Turbulence Simulations (CATS). *The Astrophysical Journal*, 905(1):14, December 2020. ISSN 0004-637X, 1538-4357. doi: 10.3847/1538-4357/abc484. URL http://arxiv.org/abs/2010.11227. arXiv:2010.11227 [astro-ph.GA].

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural Ordinary Differential Equations, June 2018. URL https://arxiv.org/abs/1806.07366v5.

Wuyang Chen, Jialin Song, Pu Ren, Shashank Subramanian, Dmitriy Morozov, and Michael W Mahoney. Data-efficient operator learning via unsupervised pretraining and in-context learning. *Advances in Neural Information Processing Systems*, 37:6213–6245, 2024.

Patrick L Combettes and Isao Yamada. Compositions and convex combinations of averaged nonexpansive operators. *Journal of Mathematical Analysis and Applications*, 425(1):55–70, 2015. ISSN 0022-247X.

Sheikh Md Shakeel Hassan, Arthur Feeney, Akash Dhruv, Jihoon Kim, Youngjoon Suh, Jaiyoung Ryu, Yoonjin Won, and Aparna Chandramowlishwaran. Bubbleml: A multiphase multiphysics dataset and benchmarks for machine learning. *Advances in Neural Information Processing Systems*, 36:418–449, 2023.

Alec Helbling, Andrey Bryutkin, Mauro Martino, Duen Horng Chau, Nima Dehmamy, and Hendrik Strobelt. Flow Reasoning Models: Turning Flows Into Efficient Recurrent Reasoners, September 2026. URL http://arxiv.org/abs/2606.29150. arXiv:2606.29150 [cs.AI].

Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient Foundation Models for PDEs. *arXiv preprint arXiv:2405.19101*, 2024. URL https://arxiv.org/abs/2405.19101.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://arxiv.org/abs/2006.11239.

Carl T Kelley. *Iterative methods for linear and nonlinear equations*. SIAM, 1995. ISBN 0-89871-352-8.

Siavash Khodakarami, Vivek Oommen, Aniruddha Bora, and George Em Karniadakis. Mitigating spectral bias in neural operators via high-frequency scaling for physical systems. *Neural Networks*, pp. 108027, 2025. ISSN 0893-6080.

Jean Kossaifi, Nikola Kovachki, Kamyar Azizzadenesheli, and Anima Anandkumar. Multi-Grid Tensorized Fourier Neural Operator for High-Resolution PDEs, September 2023. URL http://arxiv.org/abs/2310.00120. arXiv:2310.00120 [cs.LG].

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural Operator: Learning Maps Between Function Spaces With Applications to PDEs. *Journal of Machine Learning Research*, 24(89):1–97, 2023. URL https://jmlr.org/papers/v24/21-1524.html.

Tianhong Li and Kaiming He. Back to Basics: Let Denoising Generative Models Denoise. *arXiv preprint arXiv:2511.13720*, 2025. URL https://arxiv.org/abs/2511.13720.

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.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

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

Xiaotian Liu, Shuyuan Shang, Xiaopeng Wang, Pu Ren, and Yaoqing Yang. Iterative Refinement Neural Operators are Learned Fixed-Point Solvers: A Principled Approach to Spectral Bias Mitigation. June 2026. URL https://openreview.net/forum?id=4lAS8tI23u&referrer=%5Bthe%20profile%20of%20Xiaotian%20Liu%5D(%2Fprofile%3Fid%3D~Xiaotian_Liu2).

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*, 2023. URL https://arxiv.org/abs/2209.03003.

Xihaier Luo, Xiaoning Qian, and Byung-Jun Yoon. Hierarchical Neural Operator Transformer with Learnable Frequency-aware Loss Prior for Arbitrary-scale Super-resolution, May 2024. URL http://arxiv.org/abs/2405.12202. arXiv:2405.12202 [cs.CV].

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, and Saining Xie. Inference-Time Scaling for Diffusion Models beyond Scaling Denoising Steps, January 2025. URL http://arxiv.org/abs/2501.09732. arXiv:2501.09732 [cs.CV].

Sepehr Mousavi, Shizheng Wen, Levi Lingsch, Maximilian Herde, Bogdan Raonić, and Siddhartha Mishra. RIGNO: A Graph-based Framework for Robust and Accurate Operator Learning for PDEs on Arbitrary Domains. *arXiv preprint arXiv:2501.19205*, 2025. URL https://arxiv.org/abs/2501.19205.

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, February 2025. URL http://arxiv.org/abs/2412.00568. arXiv:2412.00568 [cs.LG].

Pu Ren, N Benjamin Erichson, Junyi Guo, Shashank Subramanian, Omer San, Zarija Lukic, and Michael W Mahoney. Superbench: A super-resolution benchmark dataset for scientific machine learning. *arXiv preprint arXiv:2306.14070*, 2023.

Ivan Dario Jimenez Rodriguez, Aaron D. Ames, and Yisong Yue. LyaNet: A Lyapunov Framework for Training Neural ODEs, February 2022. URL http://arxiv.org/abs/2202.02526. arXiv:2202.02526 [cs.LG].

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters, August 2024. URL http://arxiv.org/abs/2408.03314. arXiv:2408.03314 [cs.LG].

Shizheng Wen, Arsh Kumbhat, Levi Lingsch, Sepehr Mousavi, Yizhou Zhao, Praveen Chandrashekar, and Siddhartha Mishra. Geometry aware operator transformer as an efficient and accurate neural surrogate for pdes on arbitrary domains. *Advances in Neural Information Processing Systems*, 38: 155423–155501, 2026.

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

APPENDIX CONTENTS

A Positioning of FOCO Relative to Prior Work **14**

A.1 Relation to IRNO and Iterative PDE Refinement 14

A.2 Relation to Recurrent-Depth and Adaptive-Compute PDE Surrogates 14

A.3 Relation to Rectified Flow and Flow Matching 15

A.4 Relation to Lyapunov-trained dynamics and Neural ODEs. 15

A.5 Relation to Test-Time Scaling in Generative Models 15

B Training and Inference Algorithm **17**

C Proofs and Supplementary Analysis **18**

C.1 Notation and Implicit Fixed-Point Consistency 18

C.2 Proof of the Finite-Budget Refinement Bound 18

C.3 Proof of Cross-Surrogate Stability 19

C.4 Connection to Relaxed Picard Iteration and Euler Sampling 19

C.5 Unbiased Stochastic Successor Supervision 20

C.6 Staged Optimization and Training-Memory Complexity 20

C.7 Additional State-Coverage Bound 21

C.8 Scope of the Guarantees and Interpretation of Refinement 21

D Dataset and Base-Operator Specifications **22**

D.1 Dataset Specifications 22

D.2 Base Neural Operators 25

D.3 Evaluation Protocol 28

D.4 FOCO Corrector Architecture and Training Details 30

D.5 IRNO Training Details 31

E Training-Efficiency Profiling **31**

F Ablation Study **33**

F.1 Distribution of the refinement fraction. 33

F.2 Self-generated states versus linear interpolation. 33

F.3 Residual correction versus direct solution prediction. 34

F.4 Relation to naive single-shot residual regression. 35

G Additional Results **36**

G.1 Autoregressive results 36

G.2 FNO-Based ERA5 Refinement Trajectories 36

G.3 FNO-Based TR-2D Refinement Trajectories 37

G.4 FNO-Based MHD Refinement Trajectories 38

G.5 TFNO-Based ERA5 Refinement Trajectories 38
G.6 TFNO-Based MHD Refinement Trajectories 39
G.7 Transolver-Based FOCO Refinement 40
G.8 Complete Cross-Operator Transfer Results 41
G.9 Spectral Convergence under Test-Time Scaling 45
G.10 ERA5 Interpolation and Temporal Extrapolation Cases 47
G.11 Active Matter One-Step Refinement 47
G.12 TR-2D Refinement and Autoregressive Rollouts 47

A Positioning of FOCO Relative to Prior Work

FOCO combines post-hoc PDE refinement with a training procedure that learns from its own intermediate predictions without recurrent differentiation. Its central contribution is the joint design of self-generated partial states, stochastic detached successor supervision, and staged optimization: a shared corrector supports variable-depth inference using only three corrector evaluations per training example. We distinguish this contribution from related formulations below; Table 5 summarizes the main differences.

A.1 Relation to IRNO and Iterative PDE Refinement

IRNO (Liu et al., 2026) is the closest methodological comparison. Both methods refine frozen surrogate predictions through a shared residual corrector and support cross-operator transfer and inference beyond the training depth. The distinction is therefore primarily in how the corrector is trained. IRNO explicitly unrolls and supervises a recurrent refinement trajectory, together with progressive spectral losses and an explicit fixed-point penalty. Its training states are also model-generated. FOCO instead constructs self-generated partial states and detached successors, supervising each toward the same endpoint without differentiating through earlier refinement steps.

Recurrent training becomes increasingly expensive as the unroll depth grows, since each refinement depends sequentially on the previous state and gradients propagate through the trajectory. Moreover, optimization over a finite training horizon does not by itself guarantee stable behavior at substantially larger inference depths; repeated updates may over-correct or amplify errors when the learned map is not sufficiently contractive, the limitation explicitly acknowledged by the IRNO paper. FOCO instead exposes the corrector to its own generated states while keeping the maximum differentiable depth to one corrector application, thereby decoupling training depth from inference depth.

With $K_{\mathrm{tr}} = 6$, our matched IRNO configuration requires seven logical corrector evaluations, including the fixed-point branch, whereas FOCO requires three and retains only one corrector activation graph at a time. Spectral losses introduce additional computation but no additional corrector calls. For the same inference budget K, however, both methods require K sequential corrector evaluations;Detailed accounting is provided in Appendix E.

PDE-Refiner (Lippe et al., 2023) takes a different approach: it trains a step-conditioned predictor with denoising supervision on Gaussian-corrupted targets. FOCO instead preserves an existing pretrained surrogate and learns deterministic corrections from its own generated states, without additive-noise injection or step conditioning.

A.2 Relation to Recurrent-Depth and Adaptive-Compute PDE Surrogates

RecurrSim (Abdul Majid et al., 2026) controls the accuracy–cost trade-off by repeatedly updating a latent representation inside an encoder–recurrent-block–decoder model. The recurrent computation must be incorporated into the simulator's architecture and training. FOCO instead operates on the output field of a frozen surrogate without accessing or modifying its hidden representations or parameters.

RecurrSim already limits activation storage through truncated backpropagation over the final B_{bp} recurrent iterations; its forward computation still traverses the sampled recurrent depth. Thus, the distinction is not simply constant versus unbounded training memory. FOCO uses a fixed three-evaluation objective with one-application differentiable paths, rather than training an internal recurrent simulator through a truncated trajectory. Both approaches offer adjustable inference computation, but FOCO targets reuse of existing pretrained surrogates.

A.3 Relation to Rectified Flow and Flow Matching

Flow matching and rectified flow learn time-dependent vector fields that transport probability distributions along prescribed paths (Lipman et al., 2022; Liu et al., 2023). For a linear conditional path $z_t = (1-t)z_0 + tz_1$, for example, the corresponding velocity target is $z_1 - z_0$. FOCO borrows the interpolation perspective but constructs its training states along the model's own correction direction:

$$u_t = (1-t)u_0 + t\,\mathrm{sg}(a_0) = u_0 + t\ \mathrm{sg}(D_\theta(x, u_0) - u_0)\,. \tag{6}$$

Thus, the endpoint of the interpolation is the model-generated correction $D_\theta(x, u_0)$ rather than the ground-truth solution y, while every supervised prediction is still regressed directly toward y. This exposes the corrector to its own characteristic errors rather than to a prescribed path.

A second connection is parameterization. FOCO does not take t as input or regress a time-dependent velocity. Instead, from any intermediate state u, it directly predicts a residual toward the endpoint through $D_\theta(x, u) = u + G_\theta(x, u)$. This is analogous to the endpoint or "x-prediction" parameterization used in recent flow-matching methods (Li & He, 2025), while remaining iteration-independent.

Finally, the term *refinement flow* refers to the dynamics induced by repeated application of this endpoint map. Inference follows a relaxed Picard iteration and admits the auxiliary ODE interpretation $\dot{u} = D_\theta(x, u) - u$, as developed in Appendix C.4. This connection does not make FOCO a conventional flow-matching model: it does not learn a probability-transport velocity, generative likelihood, or posterior sampler. Moreover, standard diffusion and flow-matching training already avoids differentiation through complete sampling trajectories (Ho et al., 2020; Lipman et al., 2022); FOCO's training-efficiency advantage is specifically relative to recurrent PDE-refinement training.

A.4 Relation to Lyapunov-trained dynamics and Neural ODEs.

Neural ODEs model inference as continuous-time dynamics and typically obtain predictions by numerically integrating a learned vector field (Chen et al., 2018). LyaNet (Rodriguez et al., 2022) further imposes a Lyapunov decrease condition so that local supervision of sampled states yields globally stable inference dynamics, and proposes training schemes that avoid differentiating through the full ODE solver. FOCO shares this local-to-global perspective, but differs in both objective and parameterization: it learns a discrete post-hoc correction map for a frozen PDE surrogate, trains on self-generated on-policy solution states using ordinary endpoint supervision, and requires neither an explicit Lyapunov loss nor a time-conditioned vector field. Its iterative behavior is therefore analyzed through relaxed fixed-point dynamics rather than guaranteed by an imposed Lyapunov condition.

A.5 Relation to Test-Time Scaling in Generative Models

Generative models can allocate additional inference computation through more sampling or solver evaluations, as well as through candidate search and verification (Ho et al., 2020; Lipman et al., 2022; Ma et al., 2025). Related strategies improve language-model reasoning by allocating computation to response refinement or verifier-guided search (Snell et al., 2024). FOCO instantiates the sequential-refinement form of this principle: it spends additional corrector evaluations on the same PDE prediction while keeping the input, model parameters, and spatial discretization fixed. It requires neither candidate selection nor a verifier at inference.

Our two schedules support distinct usage modes. Budget-independent refinement produces a shared trajectory that can be stopped or continued, whereas the budget-aware schedule $\eta_k = 1/(K-k)$ generates a different trajectory for each total budget K. Accordingly, improvement along a single trajectory should be distinguished from improvement in terminal accuracy across budgets. Neither requires every additional evaluation to improve every metric; the relevant evidence is the resulting accuracy–computation trade-off. The contribution of FOCO is to support this trade-off for frozen

PDE surrogates without training through long refinement trajectories. Its three training evaluations are separate supervised branches, not a three-step recurrent training unroll.

Table 5: Positioning of FOCO relative to representative PDE refinement methods. FOCO uniquely combines post-hoc frozen-surrogate refinement, self-generated solution states, and depth-one training without recurrent backpropagation or auxiliary fixed-point/spectral losses.

Property	**PDE-Refiner**	**IRNO**	**RecurrSim**	**FOCO**
Frozen-base post-hoc	No	Yes	No	**Yes**
Training states	Gaussian-corrupted targets	Recurrent predictions	Recurrent latents	**Self-generated**
Gradient depth	1	K_{tr}	$\min(K_{\mathrm{tr}}, B_{\mathrm{bp}})$	1
Trajectory backprop.	No	Yes	Truncated	**No**
Step-conditioned	Yes	No	No	**No**
FP / spectral losses	Neither	Both	Neither	**Neither**
Test-time compute	Denoising steps	Refinement steps	Recurrent depth	**Refinement steps**

B Training and Inference Algorithm

Algorithm 1 FOCO training and test-time iterative refinement.

Require: Frozen surrogate B_ϕ, trainable corrector G_θ, endpoint loss ℓ, interpolation distribution $p(t)$

procedure TRAIN($\mathcal{B} = \{(x_i, y_i)\}_{i=1}^m$)
  $x \leftarrow (x_1, \dots, x_m)$, $y \leftarrow (y_1, \dots, y_m)$
  $u_0 \leftarrow \mathrm{sg}(B_\phi(x))$ ▷ Frozen base prediction
  Sample $t_i \sim p(t)$ and $Z_i \sim \mathrm{Bernoulli}(1/2)$ independently
  Zero accumulated parameter gradients

  $a_0 \leftarrow u_0 + G_\theta(x, u_0)$ ▷ Corrector evaluation 1
  $\Delta_0 \leftarrow \mathrm{sg}(a_0 - u_0)$
  $v_0 \leftarrow \mathrm{sg}(a_0)$
  Backpropagate $\frac{1}{2}\ell(a_0, y)$; release graph

  $u_t^{(i)} \leftarrow u_0^{(i)} + t_i \Delta_0^{(i)}, \quad i = 1, \dots, m$
  $a_t \leftarrow u_t + G_\theta(x, u_t)$ ▷ Corrector evaluation 2
  $v_t \leftarrow \mathrm{sg}(a_t)$
  Backpropagate $\frac{1}{2}\ell(a_t, y)$; release graph

  for $i = 1, \dots, m$ **do**
    $v_Z^{(i)} \leftarrow \begin{cases} v_0^{(i)}, & Z_i = 0, \\ v_t^{(i)}, & Z_i = 1 \end{cases}$
  end for
  $a_Z \leftarrow v_Z + G_\theta(x, v_Z)$ ▷ Corrector evaluation 3
  Backpropagate $\ell(a_Z, y)$; release graph
  Update θ once using the accumulated gradients
end procedure

procedure REFINE($x, K, \{\eta_k\}_{k=0}^{K-1}$)
  $u_0 \leftarrow B_\phi(x)$ ▷ Frozen surrogate evaluation
  for $k = 0, \dots, K-1$ **do**
    $u_{k+1} \leftarrow u_k + \eta_k G_\theta(x, u_k)$
  end for
  return u_K
end procedure

C Proofs and Supplementary Analysis

C.1 Notation and Implicit Fixed-Point Consistency

Fix a conditioning–target pair (x, y) and parameters θ. Write $D_x(u) = D_\theta(x, u)$ and $\|\cdot\| = \|\cdot\|_{\mathcal{U}}$, where $\mathcal{U}$ is a real Banach space. Expectations below are over the training-state draws, conditional on (x, y, θ); all sampled quantities are assumed measurable and the stated risks finite. For the forward values of the training states, define

$$u_0 = B_\phi(x), \qquad a_0 = D_x(u_0), \qquad u_t = (1-t)u_0 + ta_0, \qquad a_t = D_x(u_t). \tag{7}$$

The stop-gradient operations do not change these forward values. Let $V = a_0$ when $Z = 0$ and $V = a_t$ when $Z = 1$, where $Z \sim \mathrm{Bernoulli}(1/2)$ independently of t. The expected endpoint and successor losses are

$$\mathcal{E}_x = \frac{1}{2}\|a_0 - y\|^2 + \frac{1}{2}\mathbb{E}_t\|a_t - y\|^2, \tag{8}$$

$$\mathcal{S}_x = \mathbb{E}_{t,Z}\|D_x(V) - y\|^2, \qquad \mathcal{R}_x = \mathcal{E}_x + \mathcal{S}_x. \tag{9}$$

In particular, the successor distribution satisfies

$$\mathbb{E}\|V - y\|^2 = \mathcal{E}_x. \tag{10}$$

Proof of Proposition 3.1. For every sampled successor v, the assumed Lipschitz condition gives

$$\delta_x = \|D_x(y) - y\| \tag{11}$$

$$\leq \|D_x(y) - D_x(v)\| + \|D_x(v) - y\| \tag{12}$$

$$\leq L\|y - v\| + \|D_x(v) - y\|. \tag{13}$$

Taking expectations and applying Cauchy–Schwarz yields

$$\delta_x \leq L\sqrt{\mathcal{E}_x} + \sqrt{\mathcal{S}_x} \leq \sqrt{(1+L^2)(\mathcal{E}_x + \mathcal{S}_x)} = \sqrt{(1+L^2)\mathcal{R}_x}. \tag{14}$$

□

Endpoint supervision controls the accuracy of the successor inputs, while successor supervision controls their corrected outputs. If $\mathcal{R}_x = 0$, then $D_x(u_0) = y$ and $D_x(D_x(u_0)) = y$, so $D_x(y) = y$ even without a Lipschitz assumption. Neither conclusion alone establishes attraction to the fixed point.

C.2 Proof of the Finite-Budget Refinement Bound

Proof of Theorem 3.3. Because Ω is a nonempty closed subset of a Banach space, it is complete. The assumed contraction $D_x : \Omega \to \Omega$ therefore has a unique fixed point $u_x^\star$ by Banach's fixed-point theorem.

For each step define

$$T_k = (1-\eta_k)I + \eta_k D_x, \qquad c_k = 1 - \eta_k(1-\rho), \qquad Q_k = \prod_{j=0}^{k-1} c_j, \quad Q_0 = 1. \tag{15}$$

Convexity of Ω implies $T_k(\Omega) \subseteq \Omega$, and hence all iterates remain in Ω. Moreover, for $u, v \in \Omega$,

$$\|T_k(u) - T_k(v)\| \leq (1-\eta_k)\|u - v\| + \eta_k\|D_x(u) - D_x(v)\| \leq c_k\|u - v\|. \tag{16}$$

Since $T_k(u_x^\star) = u_x^\star$, iteration gives

$$\|u_K - u_x^\star\| \leq Q_K\|u_0 - u_x^\star\|. \tag{17}$$

To bound error relative to the target, let $e_k = \|u_k - y\|$ and $b_x = \delta_x/(1-\rho)$. The triangle inequality gives

$$e_{k+1} \leq (1-\eta_k)e_k + \eta_k\|D_x(u_k) - y\| \tag{18}$$

$$\leq (1-\eta_k)e_k + \eta_k(\rho e_k + \delta_x) \tag{19}$$

$$= c_k e_k + (1-c_k)b_x. \tag{20}$$

Induction therefore yields

$$\|u_K - y\| \le Q_K \|u_0 - y\| + (1 - Q_K)\frac{\delta_x}{1-\rho}. \tag{21}$$

At the fixed point,

$$\|u_x^\star - y\| \le \|D_x(u_x^\star) - D_x(y)\| + \delta_x \le \rho\|u_x^\star - y\| + \delta_x, \tag{22}$$

so $\|u_x^\star - y\| \le b_x$.

For full-strength refinement, $\eta_k = 1$ and $Q_K = \rho^K$. For the uncapped budget-dependent schedule $\eta_k = 1/(K-k)$, reindexing with $j = K - k$ gives

$$Q_K = \prod_{j=1}^{K}\left(1 - \frac{1-\rho}{j}\right) = \frac{\Gamma(K+\rho)}{\Gamma(\rho)\Gamma(K+1)}. \tag{23}$$

Using $1 - z \le e^{-z}$ and $\sum_{j=1}^{K} j^{-1} \ge \log(K+1)$,

$$Q_K \le \exp\left(-(1-\rho)\sum_{j=1}^{K}\frac{1}{j}\right) \le (K+1)^{-(1-\rho)}. \tag{24}$$

Thus $Q_K = \mathcal{O}(K^{-(1-\rho)})$ and $Q_K \to 0$ under either schedule. Equation equation 17 proves convergence to $u_x^\star$. For the budget-dependent schedule, this statement concerns the terminal prediction of each run initialized at the same u_0; different budgets need not share their intermediate iterates. □

Connection to the training objective. Under these assumptions, u_t and both successors lie in Ω. Proposition 3.1 therefore applies with $L = \rho$, and gives the combined bound

$$\|u_K - y\| \le Q_K\|u_0 - y\| + \frac{1-Q_K}{1-\rho}\sqrt{(1+\rho^2)\mathcal{R}_x}. \tag{25}$$

This separates initialization error from a loss-controlled bias bound. It remains conditional on contraction and does not imply monotonic improvement in every physical evaluation metric.

C.3 Proof of Cross-Surrogate Stability

Proof of the transfer bound in Theorem 3.3. Let $u_0 = B_\phi(x)$ and $\tilde{u}_0 = \widetilde{B}_{\tilde{\phi}}(x)$ belong to the same invariant set Ω. Using the same corrector and schedule, Equation equation 16 gives

$$\|u_{k+1} - \tilde{u}_{k+1}\| \le c_k\|u_k - \tilde{u}_k\|. \tag{26}$$

Multiplying the factors yields

$$\|u_K - \tilde{u}_K\| \le Q_K\|B_\phi(x) - \widetilde{B}_{\tilde{\phi}}(x)\|. \tag{27}$$

□

The result concerns two initializations for the same conditioning and learned corrector. It does not guarantee that arbitrary new surrogate outputs lie in Ω, or establish transfer across physical tasks.

C.4 Connection to Relaxed Picard Iteration and Euler Sampling

The update $u_{k+1} = (1-\eta_k)u_k + \eta_k D_x(u_k)$ is a relaxed Picard iteration for the fixed-point equation $u = D_x(u)$. For the continuous-time interpretation, assume D_x is locally Lipschitz on a neighborhood of the refinement states and consider

$$\frac{dv}{d\tau} = D_x(v) - v, \qquad v(0) = u_0. \tag{28}$$

Explicit Euler with step size η_k recovers the FOCO update. Here τ is auxiliary refinement time, not physical PDE time.

To obtain the budget-dependent schedule, set $\tau = -\log(1-s)$ and $w(s) = v(-\log(1-s))$. Then

$$\frac{dw}{ds} = \frac{D_x(w) - w}{1-s}, \qquad 0 \le s < 1. \tag{29}$$

Euler discretization at $s_k = k/K$, with step $1/K$, gives

$$u_{k+1} = u_k + \frac{1/K}{1-k/K}\big(D_x(u_k) - u_k\big) = u_k + \frac{1}{K-k} G_\theta(x, u_k). \tag{30}$$

This derives the schedule $\eta_k = 1/(K-k)$ from the endpoint map. The vector field is singular at $s = 1$, the final relaxation step is always one, and $\sum_k \eta_k = \sum_{j=1}^{K} j^{-1}$ grows with K. Accordingly, convergence is justified by the discrete contraction bound above, not by a standard nonsingular, fixed-horizon Euler error estimate.

Capping the final updates. For $K \ge 2$, replacing the final two step sizes by a fixed $c \in (0, 1/2]$ changes the product to

$$Q_K^{\mathrm{cap}} = \frac{2[1-c(1-\rho)]^2}{\rho(1+\rho)} Q_K. \tag{31}$$

Thus the capped schedule retains the same polynomial convergence order under the contraction assumptions. This does not establish that capping improves finite-budget prediction accuracy.

C.5 Unbiased Stochastic Successor Supervision

Fix reference parameters $\bar\theta$ and the detached states v_0, v_t generated at those parameters. Define

$$f_0(\theta) = \ell(D_\theta(x, v_0), y), \qquad f_t(\theta) = \ell(D_\theta(x, v_t), y), \tag{32}$$

where the state arguments are held fixed during differentiation. Conditional on these states, Z chooses each branch with probability $1/2$, and hence

$$\mathbb{E}_Z f_Z(\theta) = \frac{1}{2} f_0(\theta) + \frac{1}{2} f_t(\theta). \tag{33}$$

Where the losses are differentiable in θ, differentiating this finite sum gives

$$\mathbb{E}_Z \nabla_\theta f_Z(\theta) = \frac{1}{2} \nabla_\theta f_0(\theta) + \frac{1}{2} \nabla_\theta f_t(\theta). \tag{34}$$

Thus successor sampling is unbiased for the detached objective and its semi-gradient, not for the full gradient through state generation. Writing the expected detached-state objective as $J(\theta; \bar\theta)$, FOCO evaluates $\partial_\theta J(\theta; \bar\theta)|_{\theta=\bar\theta}$ rather than $dJ(\theta;\theta)/d\theta$.

For $g_0 = \nabla_\theta f_0$ and $g_t = \nabla_\theta f_t$, the conditional sampling variance is

$$\mathbb{E}_Z \left\| \nabla_\theta f_Z - \frac{g_0 + g_t}{2} \right\|_2^2 = \frac{1}{4} \|g_0 - g_t\|_2^2. \tag{35}$$

The argument extends to minibatch averages with independent per-example selectors and samplewise corrector evaluations. Batch-coupled operations require separate analysis. Unbiasedness preserves the expected loss weighting; it does not by itself imply better optimization or accuracy.

C.6 Staged Optimization and Training-Memory Complexity

For fixed minibatch size and resolution, let A denote the live activation storage of one corrector evaluation and backward pass, including its saved inputs and outputs. Let P denote resident model, parameter-gradient, and optimizer storage, and let S denote storage for a constant number of additional field tensors. Let C denote the work of one corrector forward/backward evaluation.

FOCO first computes a_0, retains its detached successor and correction, and differentiates $\frac{1}{2}\ell(a_0, y)$. It releases that graph before computing and differentiating $\frac{1}{2}\ell(a_t, y)$, retaining only the required detached states. The third evaluation differentiates $\ell(D_\theta(x, v_Z), y)$. Parameters remain unchanged throughout these passes, and one optimizer update follows gradient accumulation. For the same

random draws, the accumulated gradient therefore equals the gradient of the sum of the three detached losses.

At most one corrector activation graph is live at a time, giving

$$M_{\mathrm{FOCO}} = \mathcal{O}(A + P + S), \qquad N_{\mathrm{FOCO}} = 3, \tag{36}$$

with corrector work proportional to $3C$. An ordinary stored-graph unroll of K_{tr} corrections, together with one fixed-point evaluation, instead uses

$$M_{\mathrm{unroll}} = \mathcal{O}(K_{\mathrm{tr}}A + P + S), \qquad N_{\mathrm{unroll}} = K_{\mathrm{tr}} + 1. \tag{37}$$

For $K_{\mathrm{tr}} = 6$, FOCO reduces logical corrector evaluations from seven to three, or 57.1%.

These estimates count logical corrector evaluations and exclude the frozen surrogate as well as any recomputation introduced by activation checkpointing. They therefore characterize algorithmic workload and live activation storage, rather than exact wall-clock time or total GPU memory. In our matched IRNO setting with $K_{\mathrm{tr}} = 6$, IRNO recurrently unrolls and supervises the six-step refinement trajectory and performs an additional fixed-point evaluation, resulting in seven logical corrector evaluations, compared with three for FOCO. IRNO further applies progressive spectral losses along the trajectory, introducing additional loss computation beyond the recurrent unroll. Checkpointing can reduce the activation-memory cost of such recurrent training at the expense of recomputation, so our comparison is with standard stored-graph unrolling rather than a lower bound for all recurrent implementations. FOCO instead uses detached self-generated states and staged backpropagation: each branch is differentiated and released before the next corrector evaluation, while gradients are accumulated and the parameters are updated only once.

C.7 Additional State-Coverage Bound

For this distributional statement, assume $\mathcal{U}$ is separable. Let λ be the law of u_t and $D_{x\#}\lambda$ the law of $D_x(u_t)$. The supervised-state distribution is

$$\mu_x = \frac{1}{4}\delta_{u_0} + \frac{1}{4}\lambda + \frac{1}{4}\delta_{a_0} + \frac{1}{4}D_{x\#}\lambda, \tag{38}$$

where δ_v denotes the point mass at v. By construction,

$$\mathcal{R}_x = 2\mathbb{E}_{V\sim\mu_x}\|D_x(V) - y\|^2. \tag{39}$$

Let ν_x be a deployment-state distribution for the same (x, y). Assume both laws have finite second moments and D_x is L-Lipschitz on a set containing their supports. Define

$$W_2(\mu_x, \nu_x) = \inf_{\pi\in\Pi(\mu_x,\nu_x)} \left(\int \|v - u\|^2 \, d\pi(v, u)\right)^{1/2}, \tag{40}$$

where $\Pi(\mu_x, \nu_x)$ is the set of couplings. Then

$$\left(\mathbb{E}_{U\sim\nu_x}\|D_x(U) - y\|^2\right)^{1/2} \le \sqrt{\mathcal{R}_x/2} + LW_2(\mu_x, \nu_x). \tag{41}$$

Indeed, for any coupling (V, U), $\|D_x(U) - y\| \le \|D_x(V) - y\| + L\|U - V\|$. Taking scalar L^2 norms and the infimum over couplings proves the bound. It separates supervised endpoint error from state-distribution mismatch, without requiring $L < 1$. It does not establish that on-policy sampling always has smaller mismatch than oracle interpolation, or turn empirical training loss into a population generalization guarantee.

C.8 Scope of the Guarantees and Interpretation of Refinement

Our analysis distinguishes target consistency, convergence of the learned iteration, and prediction accuracy. Proposition 3.1 controls the target fixed-point defect through the training losses, whereas Theorem 3.3 additionally assumes contractivity on an invariant region. The former does not imply the latter: endpoint supervision constrains predictions on sampled states, not the uniform Lipschitz constant of the learned map. The convergence theorem therefore provides a conditional characterization rather than a certificate that training produces contractive dynamics. The stochastic-supervision and training-complexity results do not require this assumption.

Successor supervision and average error reduction. Fix (x, y, θ) and let V be the sampled successor, equal to a_0 or a_t with probability $1/2$. By construction, $\mathbb{E}\ell(V, y) = \mathcal{E}_x$ and $\mathbb{E}\ell(D_x(V), y) = \mathcal{S}_x$. Hence,

$$\mathbb{E}[\ell(D_x(V), y) - \ell(V, y)] = \mathcal{S}_x - \mathcal{E}_x. \tag{42}$$

For the relaxed map $T_\eta = (1-\eta)I + \eta D_x$, with $0 < \eta \leq 1$, convexity of the squared norm also gives

$$\mathbb{E}\ell(T_\eta(V), y) \leq (1-\eta)\mathcal{E}_x + \eta\mathcal{S}_x. \tag{43}$$

Thus, whenever $\mathcal{S}_x < \mathcal{E}_x$, another correction reduces expected error on the supervised successor distribution without requiring pairwise contractivity. This identifies the role of successor supervision: it minimizes the error remaining after correcting the model's own outputs. However, minimizing $\mathcal{E}_x + \mathcal{S}_x$ does not enforce this inequality or guarantee improvement on every sample. Its relevance to later deployment states depends on state coverage, whose effect is quantified in Appendix C.7.

Fixed-point convergence versus prediction accuracy. Even a contractive map can produce nonmonotone target error when its fixed point differs from y. For example, consider $D(u) = u/2 + 1$, $y = 0$, and $u_0 = -4$. Two full updates followed by a half-strength update give

$$(u_1, u_2, u_3) = (-1, 1/2, 7/8).$$

Although D is globally $1/2$-contractive with fixed point $u^\star = 2$, the third update increases target error while moving closer to $u^\star$. Consequently, late-stage accuracy degradation alone does not establish expansivity or divergence. Our target-defect term accounts for the possibility that convergence of the learned iteration and optimal finite-budget prediction occur at different stages.

Relaxation and over-correction. For the grid-based MSE implementation, let $\langle \cdot, \cdot \rangle$ and $\| \cdot \|$ denote the corresponding normalized Euclidean inner product and norm. Writing $r = y - u$ and $g = G_\theta(x, u)$ gives the exact identity

$$\|u + \eta g - y\|^2 - \|u - y\|^2 = -2\eta\langle r, g\rangle + \eta^2\|g\|^2. \tag{44}$$

For $g \neq 0$ with $\langle r, g\rangle > 0$, squared error decreases whenever $0 < \eta < 2\langle r, g\rangle / \|g\|^2$, subject to $\eta \leq 1$. A useful correction direction can therefore worsen accuracy if its magnitude is excessive. Damping can mitigate this overshoot, but it cannot guarantee descent for an arbitrary correction direction or enforce contractivity. The target is used only in this analysis, not during inference.

These observations motivate finite-budget refinement with task-dependent relaxation and stopping, rather than a claim of unlimited accuracy improvement. Establishing conditions under which the training objective itself produces attractive refinement dynamics remains a limitation of the present analysis.

D Dataset and Base-Operator Specifications

This appendix provides detailed descriptions of the physical systems, data splits, preprocessing procedures, prediction tasks, and pretrained base operators used in our experiments. We consider five benchmarks: Turbulent Radiative Layer-2D (TR-2D), Active Matter, ERA5 weather super-resolution, Wave-Layer, and three-dimensional magnetohydrodynamic turbulence (MHD-64). TR-2D, Active Matter, and MHD-64 are obtained from The Well collection (Ohana et al., 2025); ERA5 follows the SuperBench scientific super-resolution protocol (Ren et al., 2023; Hassan et al., 2023); and Wave-Layer follows the dataset introduced by Herde et al. (2024); Mousavi et al. (2025).

Table 6 summarizes the effective data used in our experiments. For temporal benchmarks, every admissible sliding window is used: four input states and one target state for TR-2D, Active Matter, and MHD-64, and one input state and one target state for Wave-Layer.

D.1 Dataset Specifications

D.1.1 Turbulent Radiative Layer-2D

Physical system. The Turbulent Radiative Layer-2D dataset models the nonlinear mixing of hot, dilute gas and cold, dense gas moving at different velocities. The velocity discontinuity generates

Table 6: Summary of the datasets used in our experiments. "Fields" denotes the number of scalar channels after vector and tensor fields are flattened. The split column reports train/validation/test trajectories, except for ERA5, where it reports individual daily fields.

Dataset	Task	Spatial resolution	Fields	Train/validation/test	Train/validation/test windows
ERA5	16× spatial super-resolution	45×90 → 720×1440	3	1460/730/1095	1460/730/1095
Wave-Layer	One-step temporal prediction	128×128	1 + 1 coefficient	9000/1000/512	180000/20000/10240
TR-2D	Four-to-one temporal prediction	128×384	4	72/9/9	6984/873/873
Active Matter	Four-to-one temporal prediction	256×256	11	175/24/26	13475/1848/2002
MHD-64	Four-to-one temporal prediction	64×64×64	7	77/10/10	7392/960/960

Kelvin–Helmholtz instability and subsequent turbulent mixing, while radiative cooling removes energy from the intermediate-temperature gas. A simplified representation of the governing equations is

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \tag{45}$$

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v}\mathbf{v}) = -\nabla P, \tag{46}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot \big((E + P)\mathbf{v}\big) = -\frac{E}{t_{\text{cool}}}, \tag{47}$$

where ρ, $\mathbf{v}$, P, and E denote density, velocity, pressure, and total energy, respectively.

Dataset details. The dataset contains 90 trajectories spanning nine cooling times,

$$t_{\text{cool}} \in \{0.03, 0.06, 0.10, 0.18, 0.32, 0.56, 1.00, 1.78, 3.16\}, \tag{48}$$

with ten random initializations per cooling time. Each trajectory contains 101 snapshots on a 128×384 Cartesian grid. The domain is $x \in [-0.5, 0.5]$ and $y \in [-1, 2]$, and consecutive saved states are separated by approximately $\Delta t = 1.597$ simulation units. The x direction is periodic, whereas the y direction uses open boundaries.

We use the official release split of 72 training, 9 validation, and 9 test trajectories. Each state is represented by four scalar channels: density, pressure, and the two Cartesian velocity components. All fields are standardized using the per-field training-set statistics supplied by The Well.

Task. Given four consecutive normalized states,

$$x_t = (s_{t-3}, s_{t-2}, s_{t-1}, s_t), \qquad s_t = (\rho_t, P_t, v_{x,t}, v_{y,t}), \tag{49}$$

the base operator predicts s_{t+1}. Concatenating time and field dimensions produces a 16-channel input and a 4-channel output. All valid temporal windows are used, giving 6,984/873/873 training/validation/test pairs.

D.1.2 Active Matter

Physical system. The Active Matter benchmark describes the continuum dynamics of self-driven rod-like particles immersed in a Stokes fluid (Ohana et al., 2025). The system exhibits nonlinear coupling between particle concentration, fluid velocity, orientational order, and strain rate, producing multiscale collective dynamics and transitions between isotropic and nematic regimes.

Dataset details. The dataset varies the dipole strength $\alpha \in \{-1, -2, -3, -4, -5\}$ and alignment strength $\zeta \in \{1, 3, 5, 7, 9, 11, 13, 15, 17\}$, while fixing $\beta = 0.8$. Five random trajectories are generated for every parameter combination, resulting in 225 trajectories. We use 175 trajectories for training, 24 for validation, and 26 for testing.

Each trajectory contains 81 snapshots on a periodic 256×256 grid over a square domain with $L_x = L_y = 10$. Snapshots cover $t \in [0, 20]$ with spacing $\Delta t = 0.25$. Each state contains concentration, a two-component velocity field, a 2×2 orientation tensor $\mathbf{D}$, and a 2×2 strain-rate tensor $\mathbf{E}$. Flattening these quantities produces 11 scalar channels:

$$s_t = (\rho, v_x, v_y, D_{xx}, D_{xy}, D_{yx}, D_{yy}, E_{xx}, E_{xy}, E_{yx}, E_{yy})_t \,. \tag{50}$$

Fields are standardized using training-set statistics through The Well's Z-score normalization.

Task. The operator receives four consecutive states, concatenated into 44 input channels, and predicts the next 11-channel state. Using all admissible temporal windows gives 13,475 training, 1,848 validation, and 2,002 test pairs.

D.1.3 ERA5 Weather Super-Resolution

Data source and variables. We use the global ERA5 reanalysis benchmark distributed through the SuperBench scientific super-resolution suite (Ren et al., 2023). Each sample contains three surface or vertically integrated atmospheric variables: 10-meter kinetic energy, 2-meter temperature, and total column water vapor.

Dataset construction. The high-resolution fields are represented on a regular 720×1440 latitude–longitude grid. Following the general computer-vision setting of SuperBench, low-resolution inputs are generated by bicubic downsampling without added noise. Downsampling by a factor of 16 along each spatial dimension produces 45×90 inputs. Thus, the prediction problem is

$$\mathbb{R}^{3\times 45\times 90} \longrightarrow \mathbb{R}^{3\times 720\times 1440}. \tag{51}$$

Training uses the years 2008, 2010, 2011, and 2013, giving 1,460 daily samples. Validation consists of 2012 for temporal interpolation and 2007 for look-back extrapolation, giving 730 samples. Testing uses 2009 for interpolation and 2014–2015 for forward extrapolation, giving 1,095 samples.

Preprocessing and task. Each atmospheric variable is standardized independently using the mean and population standard deviation computed from the training split:

$$z_c = \frac{x_c - \mu_c}{\sigma_c}. \tag{52}$$

The same statistics are applied to validation and test data.

During training, we sample eight aligned patches from every daily field per epoch. Each low-resolution crop has size 8×8 and is paired with the corresponding 128×128 high-resolution target (follow the setting of SuperBench (Ren et al., 2023)). At evaluation time, the model processes complete 45×90 inputs and produces complete 720×1440 predictions without spatial tiling.

D.1.4 Wave-Layer

Physical system. Wave-Layer models acoustic-wave propagation through a spatially heterogeneous layered medium (Herde et al., 2024). The scalar wave field satisfies

$$\frac{\partial^2 u}{\partial t^2} - c(x,y)^2 \Delta u = 0, \qquad (x,y) \in (0,1)^2, \tag{53}$$

with absorbing boundary conditions. Initial conditions are random superpositions of Gaussian pulses. The propagation-speed field $c(x,y)$ consists of three to six vertically arranged layers with nonintersecting, spatially varying interfaces. The speed within each layer is constant and sampled from $\mathcal{U}[2000, 5000]$.

Dataset details. The dataset contains 10,512 trajectories generated with a finite-difference solver on a 128×128 grid. Each trajectory contains 21 uniformly spaced states over $t \in [0,1]$, corresponding to $\Delta t = 0.05$. We reserve the first 10,000 trajectories as the development set and the final 512 as the held-out test set. Within the development set, the first 9,000 trajectories are used for training and the final 1,000 for validation.

Normalization. The wave solution and wave-speed coefficient are scaled using scalar population standard deviations estimated from the first 1,000 training trajectories:

$$\sigma_u = 0.1050187, \qquad \sigma_c = 649.2779. \tag{54}$$

Here, σ_u is computed jointly over the selected trajectories, time frames, and spatial locations, whereas σ_c is computed over the selected trajectories and spatial locations. The model receives u/σ_u and c/σ_c; no mean subtraction is applied.

Task. For each pair of adjacent snapshots, the model receives the two-channel input

$$x_t = \left[\frac{c(x,y)}{\sigma_c}, \frac{u_t(x,y)}{\sigma_u} \right] \tag{55}$$

and predicts $u_{t+1}(x,y)/\sigma_u$. Each trajectory contains 21 snapshots and therefore provides 20 adjacent-frame prediction windows. The dataset contains 180,000 training, 20,000 validation, and 10,240 test windows, corresponding to 9,000, 1,000, and 512 trajectories, respectively. For autoregressive evaluation, the model begins from the single observed initial state and recursively predicts all 20 future states while retaining the time-independent coefficient $c(x,y)$ at every transition.

D.1.5 MHD-64

Physical system. MHD-64 contains simulations of isothermal, compressible magnetohydrodynamic turbulence relevant to the interstellar medium, galaxy formation, and solar-wind dynamics (Burkhart et al., 2020; Ohana et al., 2025). In idealized form, the dynamics satisfy

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \tag{56}$$

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot [\rho \mathbf{v}\mathbf{v} - \mathbf{B}\mathbf{B}] + \nabla p = 0, \tag{57}$$

$$\frac{\partial \mathbf{B}}{\partial t} - \nabla \times (\mathbf{v} \times \mathbf{B}) = 0, \tag{58}$$

where ρ, $\mathbf{v}$, $\mathbf{B}$, and p denote density, velocity, magnetic field, and isothermal pressure.

Dataset details. The simulations vary the sonic Mach number

$$\mathcal{M}_s \in \{0.5, 0.7, 1.5, 2.0, 7.0\} \tag{59}$$

and the Alfvénic Mach number $\mathcal{M}_A \in \{0.7, 2.0\}$, giving ten physical regimes. The data were originally generated at 256^3 resolution and downsampled to 64^3 after ideal low-pass filtering. All three spatial directions are periodic.

Each trajectory contains 100 snapshots from $t = 0$ to $t = 0.99$ with $\Delta t = 0.01$. The downloaded release used in our experiments contains 77 training, 10 validation, and 10 test trajectories. [1] Each state contains density, three magnetic-field components, and three velocity components:

$$s_t = (\rho, B_x, B_y, B_z, v_x, v_y, v_z)_t \,. \tag{60}$$

All seven channels are standardized using training-set statistics provided by The Well.

Task. Four consecutive states are concatenated into 28 input channels and the model predicts the next seven-channel state. This construction gives 7,392 training windows and 960 windows in each of the validation and test splits. At rollout time, the first four ground-truth states initialize the prediction and subsequent states are generated autoregressively.

D.2 Base Neural Operators

We consider three pretrained surrogate families: Fourier Neural Operator (FNO) (Li et al., 2021), Tensorized Fourier Neural Operator (TFNO) (Kossaifi et al., 2023), and Transolver (Wu et al., 2024). Each surrogate is trained independently and frozen during corrector training and evaluation. FNO is the primary base surrogate on all five benchmarks. TFNO and Transolver are additionally trained on ERA5, TR-2D, Active Matter, and MHD for the cross-operator transfer experiments. Wave-Layer uses only the regular-grid FNO.

All base surrogates expose the same solution-space interface within a given benchmark. For TR-2D, Active Matter, and MHD, four consecutive states are concatenated along the channel dimension:

$$\widehat{u}_{n+1} = B_\phi \left(u_{n-3}, u_{n-2}, u_{n-1}, u_n \right). \tag{61}$$

Physical time is implicit in the fixed temporal sampling interval and is not supplied as an additional input. Wave-Layer uses the static wave-speed field and current wave field to predict the next state. For ERA5, all three architectures first bicubically upsample the low-resolution input by $16\times$ before predicting the high-resolution target.

[1] The public dataset description specifies 100 nominal trajectories. The local release used for the reported experiments contains 97 materialized trajectories across the three splits.

D.2.1 FOURIER NEURAL OPERATOR

For a hidden field v_ℓ, one FNO block combines a pointwise linear transformation with a truncated spectral convolution:

$$v_{\ell+1} = \sigma\left[W_\ell v_\ell + \mathcal{F}^{-1}\left(R_\ell\, \mathcal{P}_\mathcal{K}\mathcal{F}(v_\ell)\right)\right], \tag{62}$$

where $\mathcal{F}$ denotes the Fourier transform, $\mathcal{P}_\mathcal{K}$ retains the prescribed low-frequency modes, W_ℓ is a pointwise linear transformation, and

$$R_\ell \in \mathbb{C}^{C_{\text{out}}\times C_{\text{in}}\times K_1\times\cdots\times K_d} \tag{63}$$

contains the learned dense spectral weights. A lifting layer maps the physical inputs to the hidden width, four spectral blocks process the hidden field, and a projection head maps it to the target variables.

We instantiate the models using `neuralop.models.FNO`. The spectral tensors are directly optimized without tensor factorization. For ERA5, the mapping is

$$\widetilde{x} = \mathcal{I}_{\text{bicubic}}(x_{\text{LR}}), \qquad \widehat{y} = B_\phi^{\text{FNO}}(\widetilde{x}). \tag{64}$$

The high-resolution field is predicted directly; no explicit residual skip from $\widetilde{x}$ to the output is imposed.

D.2.2 TENSORIZED FOURIER NEURAL OPERATOR

TFNO retains the lifting, Fourier-block, and projection structure of FNO but replaces each dense spectral tensor with a Tucker parameterization. For a d-dimensional spectral kernel,

$$(R_\ell)_{o,i,k_1,\ldots,k_d} = \sum_{r_o,r_i,r_1,\ldots,r_d} (\mathcal{G}_\ell)_{r_o,r_i,r_1,\ldots,r_d} (U_\ell^{\text{out}})_{o,r_o} (U_\ell^{\text{in}})_{i,r_i} \times \prod_{j=1}^{d} (U_\ell^{(j)})_{k_j,r_j}, \tag{65}$$

where $\mathcal{G}_\ell$ is the Tucker core and the U_ℓ's are factor matrices. The Fourier contraction is performed directly with these factors without explicitly reconstructing the dense tensor.

We use `neuralop.models.TFNO` with Tucker factorization, `rank=1.0`, `fixed_rank_modes=False`, and `implementation=factorized`. Importantly, `rank=1.0` denotes full multilinear rank in this interface; it does not denote a rank-one tensor. Our TFNO models therefore provide essentially no parameter compression. Their parameter counts are slightly larger than the corresponding dense FNOs because the full-rank Tucker core and factor matrices introduce additional parameters. What's more, `rank=1.0` is also used in IRNO work (Liu et al., 2026), and we follow the design.

Thus, FNO and TFNO share the same operator-level architecture, retained Fourier modes, depth, and hidden width, but they are not identical optimization parameterizations. FNO directly optimizes R_ℓ, whereas TFNO optimizes the Tucker core and factor matrices in Equation equation 65. The Active Matter TFNO has an effective projection width of 192, since its checkpoint uses `projection_channel_ratio=2` with hidden width 96.

D.2.3 STRUCTURED-MESH TRANSOLVER

Transolver replaces spectral convolution with physics-attention over learned slice tokens. Let $p_i \in [0,1]^d$ denote the normalized coordinate of grid point i, and let f_i contain its physical input features. The initial representation is

$$h_i^{(0)} = P_{\text{in}}([p_i, f_i])\,. \tag{66}$$

We set `unified_pos=False`, so normalized Cartesian coordinates are supplied explicitly instead of using distances to an internal reference grid.

For each attention head r, a Transolver block computes a soft assignment from N grid points to G learned physics slices:

$$S_{r,i,g} = \text{softmax}_g\left(\frac{A_r q_{r,i}}{\tau_r}\right), \qquad z_{r,g} = \frac{\sum_{i=1}^N S_{r,i,g} v_{r,i}}{\sum_{i=1}^N S_{r,i,g} + \epsilon}, \tag{67}$$

Table 7: Epoch-100 FNO and TFNO base-surrogate architectures. N_{elem} follows PyTorch's `numel()` convention, under which one complex coefficient counts as one tensor element. $N_{\mathbb{R}}$ counts real scalar degrees of freedom and therefore counts the real and imaginary parts separately. "State" is the raw tensor storage of the model state, excluding optimizer statistics and serialization overhead. TFNO uses full-rank Tucker factorization (`rank=1.0`).

Dataset	Operator	Dimension	Input/output	Modes	Blocks	Hidden/projection	N_{elem}	$N_{\mathbb{R}}$	State
ERA5	FNO	2D	3/3	12^2	4	64/128	1.428M	2.804M	10.70 MiB
	TFNO	2D	3/3	12^2	4	64/128	1.461M	2.871M	10.95 MiB
Wave-Layer	FNO	2D	2/1	12^2	4	64/128	1.427M	2.804M	10.69 MiB
TR-2D	FNO	2D	16/4	16^2	4	128/128	9.642M	19.079M	72.78 MiB
	TFNO	2D	16/4	16^2	4	128/128	9.757M	19.327M	73.73 MiB
Active Matter	FNO	2D	44/11	16^2	4	96/128	5.432M	10.740M	40.97 MiB
	TFNO	2D	44/11	16^2	4	96/192	5.507M	10.890M	41.54 MiB
MHD	FNO	3D	28/7	12^3	4	64/128	16.570M	33.085M	126.21 MiB
	TFNO	3D	28/7	12^3	4	64/128	16.604M	33.154M	126.47 MiB

Table 8: Epoch-100 structured-mesh Transolver base surrogates. "Slices" denotes learned physics slices per attention head, and "checkpoint" indicates training-time activation checkpointing. All parameters are real-valued. ERA5 is trained on 128^2 target-space patches and evaluated on complete 720×1440 fields.

Dataset	Dimension	Grid	Input/output	Width	Blocks	Heads	Slices	Checkpoint	Parameters	State
ERA5	2D	128^2 train; 720×1440 test	3/3	64	4	4	8	Yes	0.359M	1.37 MiB
TR-2D	2D	128×384	16/4	128	4	4	16	No	1.434M	5.47 MiB
Active Matter	2D	256^2	44/11	96	4	4	16	No	0.815M	3.11 MiB
MHD	3D	64^3	28/7	64	4	4	8	Yes	0.953M	3.63 MiB

where $z_{r,g}$ is a physics-aware slice token. On structured meshes, $q_{r,i}$ and $v_{r,i}$ are obtained using 3×3 convolutions in 2D and $3 \times 3 \times 3$ convolutions in 3D.

Self-attention is then applied among the G slice tokens:

$$\overline{Z}_r = \mathrm{softmax}\left(\frac{Q_r(Z_r)K_r(Z_r)^\top}{\sqrt{d_h}}\right)V_r(Z_r), \tag{68}$$

and the processed tokens are mapped back to the original grid:

$$\overline{h}_{r,i} = \sum_{g=1}^{G} S_{r,i,g}\overline{z}_{r,g}. \tag{69}$$

Each block uses pre-layer normalization, residual physics-attention, and a residual GELU MLP. Attention is performed over $G \ll N$ tokens rather than directly over all grid points, giving slice-attention cost $\mathcal{O}(NG + G^2)$ per head. These slices are learned soft aggregates and should not be confused with spatial training patches.

We use the structured 2D architecture for ERA5, TR-2D, and Active Matter, and the structured 3D architecture for MHD. For the dynamical benchmarks, Transolver learns the same fixed-step autoregressive map as Equation equation 61; no explicit physical-time embedding is used. During rollout, each predicted state is inserted into the input history for the following transition.

For ERA5, the low-resolution input is first bicubically lifted by $16\times$. Training uses random 128×128 high-resolution patches with their corresponding 8×8 low-resolution inputs, whereas evaluation processes the complete 720×1440 field using the same learned parameters. All structured Transolvers use four blocks, four heads, MLP ratio one, GELU activations, and zero dropout. ERA5 and MHD additionally use block-level activation checkpointing during training; this changes training memory and computation but not the learned model or inference procedure.

Training. All base surrogates are trained for 100 epochs with AdamW, initial learning rate 5×10^{-4}, minimum learning rate 5×10^{-6}, cosine decay, and weight decay 10^{-6}. The epoch-100 checkpoint

is retained and subsequently frozen. Differences in batch size reflect the memory requirements of the individual architectures and spatial resolutions.

D.3 Evaluation Protocol

Normalization and evaluation units. For each physical variable, we use statistics computed exclusively from the training split:

$$\widetilde{u}_{b,c,p} = \frac{u_{b,c,p} - \mu_c^{\text{train}}}{\sigma_c^{\text{train}}}, \qquad \widetilde{\widehat{u}}^{(k)}_{b,c,p} = \frac{\widehat{u}^{(k)}_{b,c,p} - \mu_c^{\text{train}}}{\sigma_c^{\text{train}}}, \tag{70}$$

where b, c, and $p \in \Omega$ index the sample, physical variable, and spatial location, respectively. For datasets using scale-only normalization, the offset is set to zero. The same statistics are applied to validation and test data.

For TR-2D, Active Matter, and MHD, each one-step test window contains four input frames and the immediately following target. Wave-Layer uses the current wave field and static wave-speed field to predict the next frame. We evaluate all complete test windows: 873 for TR-2D, 2002 for Active Matter, 960 for MHD, and 10,240 for Wave-Layer. ERA5 evaluation uses all full-resolution test fields. Refinement step $k = 0$ denotes the frozen surrogate prediction, whereas $k \geq 1$ denotes k applications of the shared corrector. Refinement steps are distinct from physical-time rollout steps.

Normalized-state MSE. For the dynamical benchmarks, we report the mean squared error in the training-normalized state space:

$$\mathrm{NMSE}^{(k)} = \frac{1}{NC|\Omega|} \sum_{b=1}^{N} \sum_{c=1}^{C} \sum_{p\in\Omega} \left(\widetilde{\widehat{u}}^{(k)}_{b,c,p} - \widetilde{u}_{b,c,p} \right)^2. \tag{71}$$

Here, "normalized" refers to the channel-wise normalization in Equation equation 70; the error is not additionally divided by the target energy.

The Well field-wise VRMSE. Our primary VRMSE follows the public implementation of The Well (Ohana et al., 2025). For each sample and physical variable, let

$$\overline{\widetilde{u}}_{b,c} = \frac{1}{|\Omega|} \sum_{p\in\Omega} \widetilde{u}_{b,c,p}, \qquad s^2_{b,c} = \frac{1}{|\Omega| - 1} \sum_{p\in\Omega} \left(\widetilde{u}_{b,c,p} - \overline{\widetilde{u}}_{b,c} \right)^2. \tag{72}$$

The field-wise variance-scaled root mean squared error is

$$\mathrm{VRMSE}^{(k)}_{b,c} = \left[\frac{|\Omega|^{-1} \sum_{p\in\Omega} \left(\widetilde{\widehat{u}}^{(k)}_{b,c,p} - \widetilde{u}_{b,c,p} \right)^2}{s^2_{b,c} + \epsilon} \right]^{1/2}, \qquad \epsilon = 10^{-7}. \tag{73}$$

We then average all finite sample–field values:

$$\mathrm{VRMSE}^{(k)}_{\text{field}} = \frac{1}{|\mathcal{I}_k|} \sum_{(b,c)\in\mathcal{I}_k} \mathrm{VRMSE}^{(k)}_{b,c}, \tag{74}$$

where $\mathcal{I}_k = \{(b,c) : \mathrm{VRMSE}^{(k)}_{b,c} \text{ is finite}\}$. This aggregation assigns equal weight to every physical variable. Unless otherwise stated, VRMSE in the main text refers to Equation equation 74.

Joint-channel VRMSE. The equation displayed in the IRNO appendix instead pools all output channels and spatial locations inside a single ratio (Liu et al., 2026). We additionally evaluate this interpretation:

$$\mathrm{VRMSE}^{(k)}_{\text{joint},b} = \left[\frac{(C|\Omega|)^{-1} \sum_{c=1}^{C} \sum_{p\in\Omega} \left(\widetilde{\widehat{u}}^{(k)}_{b,c,p} - \widetilde{u}_{b,c,p} \right)^2}{(C|\Omega|)^{-1} \sum_{c=1}^{C} \sum_{p\in\Omega} \left(\widetilde{u}_{b,c,p} - \overline{\widetilde{u}}_{b,c} \right)^2 + \epsilon} \right]^{1/2}, \tag{75}$$

followed by

$$\mathrm{VRMSE}^{(k)}_{\mathrm{joint}} = \frac{1}{N}\sum_{b=1}^{N} \mathrm{VRMSE}^{(k)}_{\mathrm{joint},b}\,. \tag{76}$$

Unlike the field-wise score, this metric pools field errors and target variances before forming the ratio. It therefore does not weight all physical variables equally and is reported only as a protocol diagnostic.

ERA5 RFNE. For ERA5 16× super-resolution, we follow the public SuperBench aggregation (Ren et al., 2023). The relative Frobenius norm error is computed for every sample and channel:

$$\mathrm{RFNE}^{(k)}_{b,c} = \frac{\left\| \widetilde{\widehat{u}}^{(k)}_{b,c} - \widetilde{u}_{b,c} \right\|_F}{\max\left(\left\| \widetilde{u}_{b,c} \right\|_F, \epsilon_{\mathrm{r}} \right)}, \qquad \mathrm{RFNE}^{(k)} = \frac{1}{NC}\sum_{b=1}^{N}\sum_{c=1}^{C} \mathrm{RFNE}^{(k)}_{b,c}, \tag{77}$$

where $\epsilon_{\mathrm{r}} = 10^{-12}$. Thus, the three standardized ERA5 variables contribute equally. We separately retain the joint-channel RFNE displayed in the IRNO appendix as a diagnostic.

ERA5 ACC. Following SuperBench, ACC is reported for the 10-m kinetic-energy channel. Using the first N_{A} selected test fields, we estimate the target and prediction mean fields:

$$m_c(p) = \frac{1}{N_{\mathrm{A}}}\sum_{b=1}^{N_{\mathrm{A}}} \widetilde{u}_{b,c,p}, \qquad \widehat{m}^{(k)}_c(p) = \frac{1}{N_{\mathrm{A}}}\sum_{b=1}^{N_{\mathrm{A}}} \widetilde{\widehat{u}}^{(k)}_{b,c,p}. \tag{78}$$

The corresponding anomalies are

$$u'_{b,c} = \widetilde{u}_{b,c} - m_c, \qquad \widehat{u}'^{(k)}_{b,c} = \widetilde{\widehat{u}}^{(k)}_{b,c} - \widehat{m}^{(k)}_c. \tag{79}$$

Their anomaly correlation is

$$\mathrm{ACC}^{(k)}_{b,c} = \frac{\left\langle \widehat{u}'^{(k)}_{b,c}, u'_{b,c} \right\rangle}{\max\left(\left\| \widehat{u}'^{(k)}_{b,c} \right\|_F \left\| u'_{b,c} \right\|_F, \epsilon_{\mathrm{r}} \right)}. \tag{80}$$

The reported ACC is the average of Equation equation 80 over the selected kinetic-energy fields. For auditing, we additionally evaluate the convention written in the IRNO appendix, which removes each sample's spatial mean and pools all channels within a single correlation coefficient. This differs from the SuperBench ACC that IRNO states it follows; we therefore use the SuperBench definition for our primary evaluation and retain the appendix convention only as a protocol diagnostic.

Relative improvement. For a lower-is-better metric M, we report

$$\mathrm{Gain}^{(k)}_{\downarrow} = 100\frac{M^{(0)} - M^{(k)}}{M^{(0)}}. \tag{81}$$

For ACC, which is higher-is-better, we report the relative increase

$$\mathrm{Gain}^{(k)}_{\uparrow} = 100\frac{\mathrm{ACC}^{(k)} - \mathrm{ACC}^{(0)}}{\mathrm{ACC}^{(0)}}. \tag{82}$$

This is a relative percentage improvement, not an absolute percentage-point difference.

Table 9 demonstrates that VRMSE aggregation materially affects both its scale and measured reduction. For Active Matter, equal field-wise averaging gives a 24.20% reduction, whereas joint aggregation gives 54.53%. This difference arises because the joint score pools field errors and variances before forming the ratio, while the public-code score weights every field equally.

Table 9: VRMSE protocol audit for FNO-based IRNO on Active Matter. Published IRNO values are quoted from Liu et al. (2026); our GroupNorm width-32 reproduction is evaluated at $K = 6$ using both field-wise and joint-channel aggregation. The published and reproduced rows use different trained checkpoints and are therefore not direct checkpoint-level comparisons.

Dataset	Source	VRMSE protocol	Base ($K = 0$)	Refined ($K = 6$)	Reduction (%)
Active Matter	IRNO paper	Reported	0.1017	0.0501	50.73
	Our reproduction	The Well field-wise	0.1990	0.1508	24.20
	Our reproduction	Joint-channel diagnostic	0.07526	0.03422	54.53

Table 10: FOCO corrector and training configurations used for the main FNO-based experiments. Condition/state denotes the numbers of conditioning and current-solution channels. Training units are patches for ERA5 and complete one-step windows otherwise.

Dataset	Dim.	Condition/state	Training domain	Channel multipliers	Norm.	Parameters	Units/epoch	Epochs	Batch
ERA5	2D	3/3	128^2 patches	$(1, 1, 2, 2, 2)$	Channel RMS	1.699M	23,360	100	64
TR-2D	2D	16/4	128×384	$(1, 1, 2, 2, 2)$	GroupNorm	1.705M	6,984	100	32
Active Matter	2D	44/11	256^2	$(1, 1, 2, 2, 2)$	GroupNorm	1.717M	13,475	100	32
MHD	3D	28/7	64^3	$(1, 1, 1, 2)$	GroupNorm	2.422M	7,392	20	1
Wave-Layer	2D	2/1	128^2	$(1, 1, 2, 2, 2)$	GroupNorm	1.700M	180,000	20	64

D.4 FOCO Corrector Architecture and Training Details

Corrector architecture. FOCO parameterizes the shared residual correction using a compact dimension-matched U-Net:

$$G_\theta(x, u) = \mathcal{U}_\theta([x, u]; 0), \qquad D_\theta(x, u) = u + G_\theta(x, u), \tag{83}$$

where $[\cdot, \cdot]$ denotes channel-wise concatenation. The second argument of $\mathcal{U}_\theta$ is the timestep input retained from the diffusion-style U-Net implementation (Ho et al., 2020; Lipman et al., 2022); it is fixed to zero in every call. Therefore, the interpolation fraction t, refinement index k, and test-time step size η_k are not provided to the network. The same iteration-invariant corrector is reused at every refinement step. The choice of a lightweight U-Net follows IRNO (Liu et al., 2026), which also adopts a U-Net corrector and motivates this design by its (i) smoothness for stable iterative refinement, (ii) multi-scale expressiveness for capturing spectral corrections, and (iii) computational efficiency when the same weights are shared across refinement steps.

All correctors use base width 32, SiLU activations, scale–shift residual blocks, zero dropout, convolutional downsampling, nearest-neighbor upsampling followed by convolution, and skip connections between matching resolutions. The encoder configuration uses one residual block per resolution. The bottleneck consists of two residual blocks separated by a single-head self-attention block; no attention is used at the remaining resolutions. Two-dimensional convolutions are used for ERA5, TR-2D, Active Matter, and Wave-Layer, while MHD uses their three-dimensional counterparts.

The final correction convolution is initialized to zero, without the Xavier initialization used in separate ablations. Consequently, $G_\theta(x, u) = 0$ at initialization and the composite endpoint map initially reduces to the identity, $D_\theta(x, u) = u$.

ERA5 uses affine channel-wise RMS normalization,

$$\mathrm{RMSNorm}(h)_{b,c,\mathbf{s}} = \gamma_c \frac{h_{b,c,\mathbf{s}}}{\sqrt{C^{-1} \sum_{j=1}^{C} h_{b,j,\mathbf{s}}^2 + \epsilon}}, \qquad \epsilon = 10^{-5}, \tag{84}$$

where normalization is applied across channels independently at every spatial location $\mathbf{s}$. The remaining benchmarks use affine GroupNorm with 32 groups. These feature-normalization layers are distinct from the per-variable normalization applied to the physical data.

For TR-2D, Active Matter, and MHD, the conditioning input concatenates four preceding solution frames. Wave-Layer uses the static wave-speed field and the current solution, while ERA5 uses the bicubically lifted low-resolution field. The two-dimensional models have feature widths $(32, 32, 64, 64, 64)$, whereas the MHD model uses $(32, 32, 32, 64)$. ERA5 is trained on aligned 128^2 patches but evaluated convolutionally on complete 720×1440 fields. Its frozen base prediction is computed on the complete field before the conditioning, prediction, and target are cropped together. The remaining correctors are trained on complete spatial fields without patching.

Table 11: Capacity- and batch-matched profiling configurations. Both methods process the same complete training split using an identical frozen FNO and corrector architecture within each benchmark. "Condition/state" gives the input channels supplied to the corrector and the channels in the solution estimate, respectively.

Dataset	Spatial grid	Condition/state	Windows	Batch	Batches/epoch	Corrector params.
TR-2D	128×384	$16/4$	6,984	32	219	1.705M
Active Matter	256×256	$44/11$	13,475	32	422	1.717M
MHD	64^3	$28/7$	7,392	1	7,392	2.422M

Optimization details. We train the corrector from scratch in FP32 using AdamW with initial learning rate 5×10^{-4}, minimum learning rate 5×10^{-6}, weight decay 10^{-5}, and global gradient-norm clipping at 1. The learning rate follows a cosine schedule. Activation checkpointing is enabled for all correctors. The reported results use the final checkpoint rather than validation-based checkpoint selection.

The architecture and objective are unchanged in the cross-operator experiments; only the frozen FNO, TFNO, or Transolver used to generate u_0 is changed. The original FNO-based TR-2D checkpoint is the only main model whose cosine scheduler is advanced once per epoch. Its later TFNO- and Transolver-based counterparts, as well as the other benchmarks, advance the scheduler after every optimizer update.

D.5 IRNO Training Details

For the capacity-matched IRNO baseline, we closely follow the configuration of the released IRNO code. We use the same frozen surrogate, width-32 U-Net corrector, data normalization, and core optimizer settings as FOCO. Benchmark-specific training domains and refinement budgets follow the corresponding IRNO configurations. Starting from $u_0 = B_\phi(x)$, IRNO differentiates through a six-step recurrent trajectory,

$$u_{k+1} = u_k + \alpha G_\theta(x, u_k), \qquad k = 0, \dots, K_{\mathrm{tr}} - 1, \qquad K_{\mathrm{tr}} = 6, \tag{85}$$

and minimizes

$$\mathcal{L}_{\mathrm{IRNO}} = \frac{1}{K_{\mathrm{tr}}} \sum_{k=1}^{K_{\mathrm{tr}}} \Big[\mathrm{MSE}(u_k, y) + \beta_{\mathrm{spec}} \mathcal{L}_{\mathrm{spec}}^{(k)}(u_k, y) \Big] + \beta_{\mathrm{fp}} \, \mathrm{MSE}(G_\theta(x, y), 0) \, . \tag{86}$$

Here $\mathcal{L}_{\mathrm{spec}}^{(k)}$ is a progressively weighted MSE between the Fourier magnitudes of u_k and y, with the frequency exponent increasing from 1 to 2 across the unroll. We use $\beta_{\mathrm{spec}} = 1$ and ramp β_{fp} to 0.01 over the first five epochs. The fixed-point term encourages the target to remain unchanged under correction. Following our IRNO reproduction, both training and inference use the constant step size $\alpha = 0.25$:

$$u_{k+1} = u_k + 0.25 \, G_\theta(x, u_k). \tag{87}$$

We report prefixes up to $K = 6$; spectral and fixed-point losses are used only during training.

E Training-Efficiency Profiling

Controlled setup. We compare FOCO with a capacity- and batch-matched IRNO reproduction on TR-2D, Active Matter, and MHD. Within each benchmark, both methods use the same frozen FNO, complete training split, batch size, data pipeline, and width-32 GroupNorm U-Net. Both correctors use activation checkpointing and are optimized with FP32 tensors, AdamW, an initial learning rate of 5×10^{-4}, a minimum learning rate of 5×10^{-6}, weight decay 10^{-5}, and gradient-norm clipping at 1. Table 11 summarizes the matched configurations. The parameter counts refer to the trainable corrector; the frozen FNO is common to each pair, and its forward pass is included in all measurements.

MHD uses batch size one because each sample is a complete 64^3 three-dimensional field. TR-2D and Active Matter use batch size 32, matching their corresponding FOCO and IRNO training configurations.

Table 12: Algorithmic workload per training epoch. "Grad. depth" is the maximum number of recurrent corrector applications on one differentiable path. FFT calls are batched transforms evaluated by the spectral losses at each optimizer step.

Dataset	Method	Calls/sample	Calls/epoch	Grad. depth	Spectral terms	FFTs/step
TR-2D	IRNO	7	48,888	6	6	12
	FOCO	3	20,952	1	0	0
Active Matter	IRNO	7	94,325	6	6	12
	FOCO	3	40,425	1	0	0
MHD	IRNO	7	51,744	6	6	12
	FOCO	3	22,176	1	0	0

Computational workload. FOCO evaluates the corrector at the base prediction, at a detached self-generated intermediate state, and at one stochastically selected detached successor. The three losses are backpropagated sequentially; each computation graph is released after its backward pass, and the accumulated gradients are used for one optimizer update. Every differentiable path therefore contains only one corrector application.

The matched IRNO objective unrolls six recurrent refinement steps and evaluates one additional target fixed-point branch. Its longest differentiable path contains six corrector applications, and each recurrent step contributes a spatial loss and a progressive spectral loss. Table 12 summarizes the resulting workloads. The counts exclude the frozen FNO forward, which is common to both methods.

These are logical workload counts rather than exact FLOPs: activation checkpointing recomputes some forward operations during backpropagation, and IRNO additionally evaluates FFT-based spectral losses. Nevertheless, FOCO reduces the number of logical corrector evaluations by

$$100\left(1-\frac{3}{7}\right)=57.14\%. \tag{88}$$

Measurement protocol. Each method was launched in a fresh process, and each comparison pair was run on the same NVIDIA RTX 6000 Ada Generation GPU with 47.4 GiB of device memory. All runs used PyTorch 2.4.0, CUDA 12.4, and cuDNN 9.1. Matrix-multiplication TF32 was disabled, whereas cuDNN TF32 was enabled. TR-2D and Active Matter were warmed up for one complete epoch, corresponding to 219 and 422 batches, respectively. Because one complete MHD epoch contains 7,392 volumetric batches, MHD was warmed up for 10 batches.

After warm-up, we measured three complete epochs. CUDA was synchronized around each timer, and peak-memory statistics were reset before each measured epoch. Timing includes data loading, frozen-FNO inference, corrector forward and backward computation, and optimization, but excludes model initialization and checkpoint serialization. We report the median epoch time and the sample standard deviation across the three epochs.

Peak allocated memory is measured with `torch.cuda.max_memory_allocated`. It includes tensors managed by PyTorch, such as parameters, activations, gradients, and optimizer states, but excludes the CUDA context and allocations outside the PyTorch allocator. We also report peak reserved memory, which depends on the PyTorch caching allocator.

Let T_{I} and T_{F} denote the median IRNO and FOCO epoch times, respectively, and let M_{I} and M_{F} denote their peak allocated memories. We report

$$\mathrm{Speedup}=\frac{T_{\mathrm{I}}}{T_{\mathrm{F}}}, \tag{89}$$

$$\mathrm{TimeReduction}=100\left(1-\frac{T_{\mathrm{F}}}{T_{\mathrm{I}}}\right), \tag{90}$$

$$\mathrm{MemoryReduction}=100\left(1-\frac{M_{\mathrm{F}}}{M_{\mathrm{I}}}\right). \tag{91}$$

FOCO achieves per-epoch speedups of $2.00\times$, $2.11\times$, and $2.14\times$ on TR-2D, Active Matter, and MHD, respectively, while reducing peak allocated memory by 69.15%, 76.36%, and 66.15%. The

Table 13: Detailed training-efficiency measurements on NVIDIA RTX 6000 Ada Generation GPUs. Epoch time is the median ± sample standard deviation over three complete epochs. Memory entries are median epoch-wise peaks.

Dataset	Batch	Method	Time/epoch (s)	Windows/s	Allocated (GiB)	Reserved (GiB)
TR-2D	32	IRNO	323.52 ± 0.19	21.59	20.99	23.33
	32	FOCO	161.91 ± 0.07	43.13	6.48	10.12
Active Matter	32	IRNO	868.89 ± 1.04	15.51	30.93	32.96
	32	FOCO	412.66 ± 0.40	32.65	7.31	10.38
MHD	1	IRNO	2024.64 ± 7.40	3.65	2.98	3.13
	1	FOCO	943.96 ± 21.56	7.83	1.01	1.15

consistently higher throughput follows from both the smaller number of corrector evaluations and the elimination of recurrent backpropagation through a six-step trajectory.

Scope and limitations. This comparison quantifies per-epoch training efficiency rather than total wall-clock time to reach a prescribed accuracy. It uses a capacity-matched IRNO reproduction and should not be interpreted as a hardware profile of the larger architecture used in the original IRNO paper. Moreover, reductions should be compared within each benchmark because spatial resolution and batch size differ across datasets.

F Ablation Study

We examine several design choices underlying FOCO: the distribution used to sample the partial-refinement fraction t, the construction of intermediate training states, and the residual parameterization. We additionally discuss the naive single-shot residual-correction alternative studied in prior work. Unless stated otherwise, all comparisons use the same frozen FNO surrogate, width-32 corrector, data split, and evaluation protocol.

F.1 Distribution of the refinement fraction.

The Uniform variant samples

$$t \sim \mathcal{U}(0, 1), \tag{92}$$

whereas our default Logit-Normal variant uses

$$t = \mathrm{sigmoid}(z), \qquad z \sim \mathcal{N}(0.8, 0.8^2). \tag{93}$$

The positive location parameter biases training toward more advanced partial corrections. Table 14 compares the two distributions while keeping the remaining training and evaluation settings fixed.

At the trained horizon $K = 2$, Logit-Normal sampling improves seven of the eight dataset–metric comparisons. The largest improvements are obtained on MHD, where NMSE and VRMSE decrease by 0.88% and 0.65%, respectively. The only exception is Active Matter VRMSE, for which Uniform sampling is better by 0.06%. Results at $K = 1$ are more mixed, particularly on MHD, indicating that the preferred state distribution depends partly on the deployment budget. Since Logit-Normal sampling provides the stronger overall two-step performance, we use it as the default in the main experiments.

F.2 Self-generated states versus linear interpolation.

We next isolate the effect of FOCO's self-generated training states by comparing them with an oracle linear-interpolation baseline on TR-2D. The baseline uses the ground-truth target to construct

$$u_t^{\mathrm{linear}} = (1 - t)u_0 + ty, \qquad t \sim \mathcal{U}(0, 1), \tag{94}$$

whereas FOCO constructs

$$u_t^{\mathrm{on}} = (1 - t)u_0 + t\,\mathrm{sg}(D_\theta(x, u_0)) = u_0 + t\ \mathrm{sg}\left(D_\theta(x, u_0) - u_0\right). \tag{95}$$

Table 14: Ablation of the partial-refinement distribution. Uniform denotes $t \sim \mathcal{U}(0,1)$, while Logit-Normal denotes $t = \mathrm{sigmoid}(z)$ with $z \sim \mathcal{N}(0.8, 0.8^2)$. The best result at each refinement budget is shown in bold. LN gain measures the relative improvement of Logit-Normal over Uniform at $K = 2$, with a positive value indicating improvement.

Dataset	Metric	$K = 1$		$K = 2$		LN gain at $K = 2$
		Uniform	Logit-Normal	Uniform	Logit-Normal	
ERA5	RFNE ↓	0.1484	**0.1476**	0.1480	**0.1470**	+0.65%
	ACC ↑	0.9277	**0.9282**	0.9280	**0.9287**	+0.07%
TR-2D	NMSE ↓	**0.06584**	0.06592	0.06530	**0.06524**	+0.09%
	VRMSE ↓	0.2235	**0.2231**	0.2202	**0.2191**	+0.51%
Active Matter	NMSE ↓	0.002047	**0.002045**	0.001648	**0.001643**	+0.28%
	VRMSE ↓	**0.1514**	0.1515	**0.1496**	0.1497	−0.06%
MHD	NMSE ↓	**0.02199**	0.02256	0.01867	**0.01851**	+0.88%
	VRMSE ↓	**0.1469**	0.1499	0.1372	**0.1363**	+0.65%

Table 15: Self-generated on-policy states versus linear interpolation on TR-2D. Gains are relative error reductions from the shared frozen FNO prediction. The best result at each refinement budget is shown in bold.

Training-state construction	K	NMSE ↓	Gain (%) ↑	VRMSE ↓	Gain (%) ↑
Frozen FNO	0	0.07623	–	0.2989	–
Linear interpolation	2	0.06708	12.00	0.2351	21.34
Self-generated on-policy	2	**0.06584**	**13.63**	**0.2235**	**25.23**

For this ablation, we use $t \sim \mathcal{U}(0,1)$ in both cases to remove the effect of the sampling distribution $p(t)$. The linear baseline follows an oracle direction from the base prediction toward the ground truth, whereas FOCO interpolates along the corrector's own predicted direction. Consequently, the self-generated states retain model-induced errors that are also present during iterative inference.

Self-generated states yield lower error at $K = 2$, but the difference becomes more pronounced at larger refinement budgets. Under the budget-aware ODE schedule, the linear-interpolation model quickly deteriorates as refinement proceeds and fails to exhibit stable convergence, whereas the self-generated model remains effective under repeated correction. We attribute this behavior to a train–inference state-distribution mismatch: linear interpolation exposes the corrector only to states lying on the linear segment between u_0 and y, but such states are generally not encountered during inference. In contrast, FOCO trains on states generated by its own correction direction, which better matches the intermediate distributions induced by iterative deployment. These results support self-generated state construction as an important component for stable test-time refinement.

F.3 Residual correction versus direct solution prediction.

We finally ask whether FOCO's improvement arises from solution-space refinement or whether the corrector could simply serve as an alternative (more expressive) PDE surrogate. To isolate the role of residual refinement, we keep the corrector architecture and inputs unchanged but remove the residual parameterization. Instead of

$$D_\theta(x, u) = u + G_\theta(x, u), \tag{96}$$

we train the same network to directly predict the clean solution,

$$D_\theta^{\mathrm{direct}}(x, u) = G_\theta(x, u), \tag{97}$$

using the same endpoint supervision. Thus, the only change is whether the network predicts a correction to the current solution or reconstructs the entire solution field itself.

On TR-2D, direct solution prediction performs poorly and can even be slightly worse than the original frozen PDE surrogate, whereas residual refinement yields substantial error reduction. This result

supports the interpretation that FOCO is not simply replacing the base surrogate with a second predictor. Instead, the frozen surrogate provides the dominant solution structure, while the much smaller corrector learns the remaining solution-space error. For example, the TR-2D FNO contains approximately 19M parameters, compared with only 1.7M parameters in the corrector.

A plausible explanation is that predicting the full PDE solution requires the small corrector to reproduce the entire solution distribution, including its large-scale variability across inputs. Residual learning poses a substantially more localized problem: conditioned on an already accurate surrogate prediction, the corrector only needs to model the remaining error structure. The failure of direct prediction therefore motivates the residual parameterization used throughout FOCO.

F.4 Relation to naive single-shot residual regression.

A natural alternative to iterative refinement is to train a corrector only at the frozen surrogate prediction,

$$G_\theta^{\mathrm{single}}(x, u_0) \approx y - u_0, \qquad \widehat{y} = u_0 + G_\theta^{\mathrm{single}}(x, u_0), \tag{98}$$

and apply it once at inference. This reduces correction to standard single-shot residual regression, without exposing the model to states produced by its own updates.

This alternative is studied directly by IRNO (Liu et al., 2026). Its cost–performance analysis compares iterative refinement with large single-pass monolithic residual-correction models based on U-Net, FNO, and SRCNN architectures. On ERA5, IRNO obtains a consistently stronger accuracy–compute frontier; for example, at approximately matched inference cost, five IRNO refinement steps outperform a $15\times$ U-Net residual corrector in both ACC and RFNE. These results show that the benefit of iterative refinement cannot be explained simply by allocating more capacity or computation to a one-shot residual model.

Our experiments address the next question: how should such an iterative corrector be trained? Under matched frozen surrogates and corrector architectures, FOCO generally outperforms our IRNO reproduction while requiring substantially less training computation. Together, these results support two complementary conclusions: iterative correction is more effective than monolithic single-shot residual regression, while FOCO's self-generated, detached-state training provides a more efficient way to learn the iterative correction map.

G Additional Results

G.1 Autoregressive results

Table 16: Autoregressive long-rollout performance of FOCO across physical systems measured using code VRMSE. At every temporal transition, FOCO applies two correction steps to the prediction of the frozen surrogate. Code VRMSE is computed independently for each sample and physical field and then averaged. Relative gain is calculated with respect to the corresponding uncorrected base rollout. Lower VRMSE is better.

Dataset	Method	Rollout horizon							
		$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$	$t=20$
TR-2D	Base	0.2989	0.3759	0.4420	0.5015	0.5547	0.6035	0.6482	–
	FOCO	0.2191	0.2910	0.3475	0.3981	0.4435	0.4874	0.5265	–
	Gain (%)	26.69%	22.58%	21.37%	20.60%	20.05%	19.25%	18.77%	–
Active Matter	Base	0.1990	0.3551	0.4856	0.6065	0.7298	0.8604	0.9977	–
	FOCO	0.1497	0.2766	0.3839	0.4791	0.5700	0.6610	0.7555	–
	Gain (%)	24.76%	22.10%	20.94%	21.01%	21.90%	23.17%	24.28%	–
MHD	Base	0.2951	0.4609	0.5852	0.7067	0.8391	0.9851	1.144	–
	FOCO	0.1363	0.2367	0.3439	0.4594	0.5749	0.6851	0.7883	–
	Gain (%)	53.81%	48.65%	41.24%	35.00%	31.49%	30.46%	31.09%	–
Wave Layer	Base	0.07530	0.1110	0.1370	0.1595	0.1801	0.1998	0.2187	0.3855
	FOCO	0.01870	0.02942	0.03787	0.04528	0.05211	0.05862	0.06482	0.1320
	Gain (%)	75.17%	73.51%	72.36%	71.60%	71.07%	70.66%	70.36%	65.77%

G.2 FNO-Based ERA5 Refinement Trajectories

Table 17 expands the FNO-based ERA5 comparison in Table 2 to the complete evaluated refinement trajectories. Therelative changes are measured against the shared frozen FNO prediction at $K = 0$, using the definitions in Section D.3. Our IRNO reproduction and FOCO model both use 1.699M-parameter RMS width-32 U-Net correctors trained with the same frozen FNO.

Table 17: Complete FNO-based refinement trajectories on ERA5 $16\times$ super-resolution under the SuperBench protocol. RFNE reduction and ACC increase are relative to the shared frozen FNO at $K = 0$. Best values are shown in bold.

Method	K	RFNE ↓	RFNE reduction (%) ↑	ACC ↑	ACC increase (%) ↑
Frozen FNO	0	0.2584	–	0.8539	–
IRNO	1	0.1850	28.38	0.9113	6.72
IRNO	2	0.1718	33.50	0.9171	7.39
IRNO	3	0.1673	35.26	0.9187	7.58
IRNO	4	0.1650	36.14	0.9195	7.68
IRNO	5	0.1636	36.68	0.9201	7.74
IRNO	6	0.1626	37.05	0.9205	7.79
FOCO	1	0.1476	42.88	0.9282	8.69
FOCO	2	**0.1470**	**43.10**	**0.9287**	**8.76**

IRNO improves monotonically through six applications, reaching a 37.05% RFNE reduction at $K = 6$. A single FOCO update already yields a larger 42.88% reduction and a higher ACC (0.9282 versus 0.9205). Relative to six-step IRNO, one-step FOCO has 9.26% lower RFNE and 0.77 percentage points higher ACC. The smaller change from $K = 1$ to $K = 2$ further shows that most of the ERA5 correction is realized in the first FOCO update.

Table 18: FNO-based refinement trajectories on ERA5 $16\times$ under the diagnostic IRNO-appendix conventions. RFNE reduction is relative to the shared frozen FNO at $K = 0$. Best values are shown in bold.

Method	K	Joint-channel RFNE ↓	RFNE reduction (%) ↑	Joint spatial-anomaly ACC ↑
Frozen FNO	0	0.2958	–	0.9553
IRNO	1	0.2088	29.39	0.9780
IRNO	2	0.1955	33.90	0.9807
IRNO	3	0.1914	35.30	0.9815
IRNO	4	0.1893	36.00	0.9819
IRNO	5	0.1880	36.44	0.9821
IRNO	6	0.1871	36.75	0.9823
FOCO	1	0.1728	41.57	0.9849
FOCO	2	**0.1722**	**41.79**	**0.9850**

Table 19: FNO-based refinement trajectories on TR-2D. Reductions are measured relative to the shared frozen FNO at $K = 0$. IRNO uses a fixed step size of 0.25. FOCO uses two full updates, followed by a damped update of 0.5 at $K = 3$. † denotes a refinement step beyond the two-step FOCO training horizon. Joint VRMSE is a secondary diagnostic; the other two metrics are used for primary comparisons. Best values are shown in bold.

Method	K	NMSE ↓	NMSE reduction (%) ↑	VRMSE ↓	VRMSE reduction (%) ↑	Joint VRMSE ↓
Frozen FNO	0	0.07623	–	0.2989	–	0.2034
IRNO	1	0.06711	11.97	0.2305	22.89	0.1614
IRNO	2	0.06627	13.07	0.2246	24.87	0.1580
IRNO	3	0.06611	13.27	0.2240	25.05	0.1577
IRNO	4	0.06599	13.43	0.2237	25.14	0.1575
IRNO	5	0.06588	13.57	0.2236	25.20	0.1573
IRNO	6	0.06579	13.70	0.2235	25.21	0.1572
FOCO	1	0.06592	13.52	0.2231	25.36	0.1579
FOCO	2	0.06524	14.41	0.2191	26.69	0.1557
FOCO	$3^\dagger$	**0.06517**	**14.51**	**0.2189**	**26.77**	**0.1555**

For completeness, Table 18 evaluates the same IRNO and FOCO predictions using the diagnostic conventions displayed in the IRNO appendix. Joint-channel RFNE pools all channels and spatial locations within each sample, while joint spatial-anomaly ACC removes each sample's spatial mean and pools the channels within one correlation coefficient. These values differ from the SuperBench metrics by construction and are reported for protocol auditing rather than cross-table comparison.

Under the appendix conventions, one FOCO update reduces joint-channel RFNE by 41.57%, compared with 36.75% after six IRNO updates. It also gives a higher joint spatial-anomaly ACC (0.9849 versus 0.9823); the second FOCO update provides only a small additional improvement.

G.3 FNO-Based TR-2D Refinement Trajectories

Tables 19 and 20 expand the FNO-based TR-2D comparison in Table 3. Both methods use the same frozen epoch-100 FNO and the same 1.705M-parameter GroupNorm width-32 corrector architecture. We evaluate all 873 one-step windows from the official test split as complete 128×384 fields. NMSE is the mean squared error in the training-normalized state space, and the primary VRMSE uses the field-wise aggregation implemented by The Well. Joint-channel VRMSE is included only as a protocol diagnostic.

At the matched two-step budget, FOCO achieves larger reductions than IRNO in both NMSE (14.41% versus 13.07%) and VRMSE (26.69% versus 24.87%). Two FOCO updates also outperform six IRNO updates, yielding 0.82% lower NMSE and 1.97% lower VRMSE with one third as many test-time corrector applications. The additional damped update gives only a small further improvement, indicating that performance is already close to saturation at $K = 2$.

Table 20: Per-field standardized VRMSE values underlying the public-code aggregation in Table 19. The four variables are density ρ, pressure p, and velocity components v_x and v_y. Lower is better.

Method	K	Density ρ	Pressure p	Velocity v_x	Velocity v_y
Frozen FNO	0	0.1229	0.6018	0.1898	0.2812
IRNO	1	0.07552	0.4862	0.1525	0.2078
IRNO	2	0.07250	0.4762	0.1489	0.2007
IRNO	3	0.07227	0.4753	0.1485	0.2002
IRNO	4	0.07210	0.4748	0.1482	0.1998
IRNO	5	0.07198	0.4747	0.1480	0.1996
IRNO	6	0.07188	0.4749	0.1479	0.1995
FOCO	1	0.07258	0.4695	0.1492	0.2011
FOCO	2	0.07064	0.4620	0.1471	0.1968
FOCO	$3^\dagger$	**0.07046**	**0.4617**	**0.1468**	**0.1965**

Table 21: FNO-based refinement trajectories on MHD-64. Reductions are measured relative to the shared frozen FNO at $K = 0$. IRNO uses six trained updates with fixed step size 0.25. FOCO uses two full trained updates; † marks one additional inference step with step size 0.5. Best values are shown in bold.

Method	K	NMSE ↓	NMSE reduction (%) ↑	VRMSE ↓	VRMSE reduction (%) ↑
Frozen FNO	0	0.08645	–	0.2951	–
IRNO	1	0.03913	54.74	0.2172	26.40
IRNO	2	0.03159	63.46	0.1892	35.90
IRNO	3	0.02947	65.91	0.1822	38.26
IRNO	4	0.02835	67.21	0.1792	39.29
IRNO	5	0.02760	68.07	0.1777	39.80
IRNO	6	0.02707	68.69	0.1770	40.01
FOCO	1	0.02256	73.91	0.1499	49.19
FOCO	2	0.01851	78.59	0.1363	53.81
FOCO	$3^\dagger$	**0.01800**	**79.19**	**0.1361**	**53.88**

The field-wise breakdown shows that the two-step FOCO advantage over six-step IRNO is shared across density, pressure, and both velocity components, rather than being driven by a single physical variable.

G.4 FNO-Based MHD Refinement Trajectories

Tables 21 and 22 expand the FNO-based MHD comparison in Table 3. At the matched two-step budget, FOCO reduces NMSE and VRMSE by 78.59% and 53.81%, compared with 63.46% and 35.90% for IRNO. Two FOCO updates also outperform six IRNO updates by 31.63% in NMSE and 23.01% in VRMSE while using one third as many test-time corrector applications. The additional damped update improves NMSE from 0.01851 to 0.01800 and VRMSE from 0.1363 to 0.1361. The smaller increment from $K = 2$ to $K = 3$ indicates that most of the improvement is already obtained within the trained two-step horizon.

The per-field results show that two-step FOCO improves on six-step IRNO for all seven physical variables. The slight aggregate VRMSE improvement at $K = 3$ is driven by density; the other six field-wise errors are already lowest at $K = 2$, consistent with the early saturation observed above.

G.5 TFNO-Based ERA5 Refinement Trajectories

Table 23 complements the FNO-based results in Table 17 by using a frozen TFNO as the base surrogate and training each corrector around its predictions.

Table 22: Per-field standardized VRMSE values underlying the public-code aggregation on MHD-64. The variables are density ρ, magnetic-field components B_x, B_y, B_z, and velocity components v_x, v_y, v_z. † marks an inference step beyond the two-step FOCO training horizon. Lower is better.

Method	K	ρ	B_x	B_y	B_z	v_x	v_y	v_z
Frozen FNO	0	0.42133	0.35185	0.30855	0.30840	0.20833	0.23312	0.23426
IRNO	1	0.40298	0.24616	0.19531	0.18829	0.15459	0.16724	0.16591
IRNO	2	0.33075	0.21593	0.16966	0.16363	0.14019	0.15250	0.15156
IRNO	3	0.31869	0.20626	0.16219	0.15667	0.13645	0.14805	0.14719
IRNO	4	0.31629	0.20093	0.15824	0.15307	0.13479	0.14582	0.14498
IRNO	5	0.31789	0.19736	0.15564	0.15072	0.13398	0.14448	0.14365
IRNO	6	0.32135	0.19491	0.15381	0.14909	0.13368	0.14365	0.14284
FOCO	1	0.25808	0.16625	0.13296	0.13241	0.11236	0.12336	0.12422
FOCO	2	0.22184	**0.15264**	**0.12163**	**0.12087**	**0.10552**	**0.11559**	**0.11609**
FOCO	$3^\dagger$	**0.21948**	0.15279	0.12164	0.12089	0.10588	0.11580	0.11635

Table 23: TFNO-based refinement trajectories on ERA5 $16\times$ super-resolution under the SuperBench protocol. RFNE reduction and ACC increase are relative to the shared frozen TFNO. IRNO uses six updates with fixed step size 0.25, whereas FOCO uses two full updates $(\eta_0, \eta_1) = (1, 1)$. Best values are shown in bold.

Method	K	RFNE ↓	RFNE reduction (%) ↑	ACC ↑	ACC increase (%) ↑
Frozen TFNO	0	0.2409	–	0.8718	–
IRNO	1	0.1902	21.04	0.9066	4.00
IRNO	2	0.1819	24.49	0.9115	4.56
IRNO	3	0.1816	24.61	0.9117	4.58
IRNO	4	0.1813	24.75	0.9118	4.59
IRNO	5	0.1809	24.90	0.9119	4.61
IRNO	6	0.1806	25.03	0.9120	4.62
FOCO	1	0.1478	38.64	0.9279	6.45
FOCO	2	**0.1471**	**38.94**	**0.9287**	**6.53**

At the matched two-step budget, FOCO has 19.13% lower RFNE and an ACC that is 1.71 percentage points higher than IRNO. A single FOCO update also outperforms six IRNO updates, with 18.15% lower RFNE and 1.59 percentage points higher ACC. The second FOCO update provides a smaller additional gain, again showing that most of the weather correction is obtained in one pass.

Table 24 gives the channel-wise RFNE values underlying the SuperBench average and separately reports the joint-channel diagnostics used in the IRNO appendix. Joint RFNE pools channels and spatial locations within each sample, while joint spatial-anomaly ACC removes each sample's spatial mean before pooling channels in one correlation coefficient. These diagnostic values use different aggregation and anomaly conventions from the primary SuperBench metrics and are not compared numerically across tables. The channel-wise results show that the FOCO improvement is consistent across all three weather variables rather than being dominated by a single channel.

G.6 TFNO-Based MHD Refinement Trajectories

Tables 25 and 26 report refinement performance with a frozen TFNO surrogate. Both methods use the same 2.422M-parameter width-32 corrector architecture.

At the matched two-step budget, FOCO has 74.83% lower NMSE and 47.47% lower VRMSE than IRNO. A single FOCO update also substantially outperforms six IRNO updates, with 66.27% lower NMSE and 39.95% lower VRMSE. With two updates, these differences increase to 70.83% and 44.65%, respectively, despite using one third as many test-time corrector applications as six-step IRNO.

Table 24: Channel-wise SuperBench RFNE inputs and joint-channel diagnostics for the TFNO-based ERA5 $16\times$ experiment. KE, T2m, and TCV denote 10-m kinetic energy, 2-m temperature, and total-column water vapor, respectively. Lower RFNE and higher ACC are better.

Method	K	KE RFNE ↓	T2m RFNE ↓	TCV RFNE ↓	Joint RFNE ↓	Joint ACC ↑
Frozen TFNO	0	0.42790	0.10825	0.18647	0.2756	0.9623
IRNO	1	0.34114	0.08508	0.14441	0.2186	0.9767
IRNO	2	0.32796	0.08064	0.13708	0.2096	0.9785
IRNO	3	0.32761	0.08038	0.13682	0.2093	0.9785
IRNO	4	0.32722	0.08008	0.13648	0.2090	0.9786
IRNO	5	0.32681	0.07976	0.13608	0.2087	0.9786
IRNO	6	0.32652	0.07950	0.13576	0.2084	0.9787
FOCO	1	0.27495	0.05692	0.11156	0.1737	0.9847
FOCO	2	**0.27353**	**0.05664**	**0.11110**	**0.1728**	**0.9849**

Table 25: TFNO-based refinement trajectories on MHD-64. Reductions are measured relative to the shared frozen TFNO at $K = 0$. IRNO uses six trained updates with fixed step size 0.25, whereas FOCO uses two full trained updates $(\eta_0, \eta_1) = (1, 1)$. Best values are shown in bold.

Method	K	NMSE ↓	NMSE reduction (%) ↑	VRMSE ↓	VRMSE reduction (%) ↑
Frozen TFNO	0	0.06633	–	0.2695	–
IRNO	1	0.06279	5.33	0.2619	2.84
IRNO	2	0.05977	9.89	0.2558	5.11
IRNO	3	0.05719	13.78	0.2510	6.90
IRNO	4	0.05500	17.08	0.2473	8.25
IRNO	5	0.05315	19.87	0.2446	9.25
IRNO	6	0.05158	22.23	0.2427	9.95
FOCO	1	0.01740	73.77	0.1457	45.93
FOCO	2	**0.01505**	**77.32**	**0.1343**	**50.16**

The per-field comparison shows that the two-step FOCO advantage is present for all seven physical variables. In contrast, six-step IRNO slightly worsens the density VRMSE relative to the frozen TFNO while improving the magnetic and velocity components.

G.7 TRANSOLVER-BASED FOCO REFINEMENT

We further evaluate FOCO with frozen structured-mesh Transolver surrogates. For each benchmark, the corrector is trained from scratch around the predictions of the corresponding Transolver; no FNO or TFNO predictions are used during corrector training. ERA5 uses the RMS-normalized width-32 corrector, whereas TR-2D, Active Matter, and MHD use GroupNorm width-32 correctors. Each result starts from one frozen-surrogate prediction and applies iterative correction on the complete test field; the dynamical benchmarks use teacher-forced one-step windows. The first two updates are the full trained-horizon updates $(\eta_0, \eta_1) = (1, 1)$. For MHD, we additionally report one extrapolative third update with $\eta_2 = 0.5$.

ERA5 super-resolution. Table 27 reports the primary SuperBench metrics over full-resolution test fields.

One FOCO update produces most of the improvement, reducing RFNE by 32.22% and increasing ACC by 5.34%. The second update yields a smaller additional gain, reaching an RFNE of 0.1478 and an ACC of 0.9282.

The channel-wise results show consistent improvement in all three weather variables. The joint-channel quantities differ from the primary SuperBench metrics in both channel aggregation and anomaly convention and should not be compared numerically across protocols.

Table 26: Endpoint per-field standardized VRMSE for the TFNO-based MHD-64 experiment. We compare the frozen TFNO, six-step IRNO, and two-step FOCO. The variables are density ρ, magnetic-field components B_x, B_y, B_z, and velocity components v_x, v_y, v_z. Lower is better.

Method	K	ρ	B_x	B_y	B_z	v_x	v_y	v_z
Frozen TFNO	0	0.35536	0.32929	0.28343	0.28302	0.19599	0.21955	0.22017
IRNO	6	0.36070	0.28801	0.24467	0.24317	0.17575	0.19302	0.19369
FOCO	2	**0.20009**	**0.15576**	**0.12211**	**0.12181**	**0.10689**	**0.11644**	**0.11730**

Table 27: Transolver-based FOCO refinement on ERA5 $16\times$ super-resolution under the SuperBench protocol. RFNE reduction and ACC increase are relative to the frozen Transolver. Best values are shown in bold.

Method	K	RFNE ↓	RFNE reduction (%) ↑	ACC ↑	ACC increase (%) ↑
Frozen Transolver	0	0.2192	–	0.8804	–
FOCO	1	0.1486	32.22	0.9275	5.34
FOCO	2	**0.1478**	**32.56**	**0.9282**	**5.42**

Dynamical systems. Table 29 summarizes the aggregate results on all 873 TR-2D, 2,002 Active Matter, and 960 MHD test windows. NMSE is the mean squared error in the training-normalized state space, while VRMSE uses the public The Well per-sample, per-field aggregation.

Two FOCO updates reduce NMSE by 51.01%, 88.55%, and 27.28% on TR-2D, Active Matter, and MHD, respectively, while reducing VRMSE by 38.82%, 41.46%, and 15.81%. The extrapolative third MHD update slightly degrades both metrics relative to $K = 2$, indicating that additional test-time refinement is not necessarily monotone beyond the trained horizon.

The field-level results confirm that the aggregate gains are not produced by a single variable. Two-step FOCO improves every reported field on TR-2D, Active Matter, and MHD. On MHD, the slight aggregate degradation at $K = 3$ is also visible across all seven fields, making $K = 2$ the best reported Transolver-based refinement budget.

G.8 Complete Cross-Operator Transfer Results

We evaluate whether a corrector trained around one frozen surrogate can refine another surrogate without retraining. Rows identify the surrogate used to generate the corrector's training states, while columns identify the frozen surrogate used at evaluation. Diagonal entries are matched-surrogate controls and off-diagonal entries are zero-shot transfers. All operators within a benchmark share the task, variables, normalization, and spatial discretization. ERA5 uses one full correction; TR-2D, Active Matter, and MHD use two full corrections, $(\eta_0, \eta_1) = (1, 1)$.

Each table combines the frozen baselines, refined metrics, and relative improvements. A refined entry reports the absolute metric followed by its relative improvement in parentheses. For a lower-is-better metric E and evaluation surrogate j,

$$\Delta_E^{i,j} = 100 \frac{E_j^{(0)} - E_{i,j}^{(K)}}{E_j^{(0)}}, \tag{99}$$

whereas the relative ACC increase reverses the numerator. Each percentage therefore uses the frozen surrogate in the same evaluation column as its baseline.

ERA5 super-resolution. We evaluate full-resolution test fields under the SuperBench protocol; kinetic-energy ACC uses the first 120 fields.

Every off-diagonal transfer improves both metrics: RFNE reductions range from 32.55% to 42.73%, and relative ACC increases range from 5.40% to 8.66%. Corrected predictions cluster near RFNE 0.148 and ACC 0.928 despite the different frozen baselines.

Table 28: Channel-wise RFNE and joint-channel diagnostics for the Transolver-based ERA5 experiment. KE, T2m, and TCV denote 10-m kinetic energy, 2-m temperature, and total-column water vapor, respectively. The joint quantities use the IRNO-appendix convention and are reported only as protocol diagnostics.

Method	K	KE RFNE ↓	T2m RFNE ↓	TCV RFNE ↓	Joint RFNE ↓	Joint ACC ↑
Frozen Transolver	0	0.40728	0.09345	0.15691	0.2566	0.9664
FOCO	1	0.27637	0.05742	0.11198	0.1746	0.9845
FOCO	2	**0.27490**	**0.05710**	**0.11153**	**0.1737**	**0.9847**

Table 29: Transolver-based FOCO refinement on TR-2D, Active Matter, and MHD. Reductions are measured relative to the corresponding frozen Transolver. The dagger denotes an additional update beyond the two-step training horizon. Best values within each benchmark are shown in bold.

Dataset	Method	K	NMSE ↓	NMSE reduction (%) ↑	VRMSE ↓	VRMSE reduction (%) ↑
TR-2D	Frozen Transolver	0	0.1015	–	0.3491	–
TR-2D	FOCO	1	0.05134	49.41	0.2159	38.14
TR-2D	FOCO	2	**0.04972**	**51.01**	**0.2136**	**38.82**
Active Matter	Frozen Transolver	0	0.01555	–	0.1737	–
Active Matter	FOCO	1	0.001948	87.47	0.1099	36.71
Active Matter	FOCO	2	**0.001780**	**88.55**	**0.1017**	**41.46**
MHD	Frozen Transolver	0	0.01892	–	0.1503	–
MHD	FOCO	1	0.01469	22.37	0.1307	13.09
MHD	FOCO	2	**0.01376**	**27.28**	**0.1266**	**15.81**
MHD	FOCO †	3	0.01404	25.79	0.1283	14.63

TR-2D. We evaluate all 873 complete 128×384 test windows. NMSE is the global mean squared error in the training-standardized state space, and VRMSE is the public The Well per-sample, per-field score.

All six off-diagonal pairings improve both metrics. Zero-shot reductions span 12.53–36.84% for NMSE and 22.96–27.49% for VRMSE. The different row rankings reflect the distinct aggregation conventions of NMSE and VRMSE.

Active Matter. We use the same two-step protocol on all 2,002 complete 256×256 test windows.

All off-diagonal transfers again improve both metrics. Zero-shot reductions span 55.28–80.58% for NMSE and 12.27–32.32% for VRMSE.

Three-dimensional MHD. We evaluate all 960 teacher-forced one-step test windows as complete 64^3 volumes without spatial patching. Every refined entry, including the Transolver-trained row, uses exactly two full correction steps; no extrapolative third-step result is included.

All six MHD off-diagonal pairings improve both metrics. Zero-shot reductions span 14.91–81.44% for NMSE and 7.51–53.61% for VRMSE. Transfer is strongest between the two Fourier backbones, while the FNO-trained corrector also improves the stronger Transolver baseline.

Across all four benchmarks, every one of the 48 off-diagonal architecture–metric entries improves over its corresponding frozen evaluation surrogate. Thus, the learned correction remains useful under compatible unseen surrogate initializations, although transfer strength depends on the training and evaluation architectures and the metric.

Table 30: Per-field standardized VRMSE for Transolver-based FOCO on TR-2D. Joint VRMSE is a secondary diagnostic that pools fields within each sample and is not used as the primary method-ranking metric.

Method	K	Density	Pressure	v_x	v_y	Joint VRMSE
Frozen Transolver	0	0.13023	0.72411	0.20714	0.33483	0.23213
FOCO	1	0.06843	0.46214	0.13479	0.19840	0.14427
FOCO	2	**0.06697**	**0.45877**	**0.13294**	**0.19560**	**0.14229**

Table 31: Per-field standardized VRMSE for Transolver-based FOCO on Active Matter. Lower is better.

Scalar and velocity fields

Method	K	Concentration	v_x	v_y
Frozen Transolver	0	0.84933	0.11877	0.09473
FOCO	1	0.68165	0.07582	0.05103
FOCO	2	**0.59840**	**0.07549**	**0.05049**

Tensor fields

Method	K	D_{xx}	D_{xy}	D_{yx}	D_{yy}	E_{xx}	E_{xy}	E_{yx}	E_{yy}
Frozen Transolver	0	0.06445	0.06328	0.06327	0.06452	0.15257	0.14363	0.14363	0.15256
FOCO	1	0.03448	0.03319	0.03319	0.03455	0.06990	0.06284	0.06284	0.06988
FOCO	2	**0.03433**	**0.03307**	**0.03307**	**0.03441**	**0.06881**	**0.06088**	**0.06088**	**0.06879**

Table 32: Per-field standardized VRMSE for Transolver-based FOCO on MHD. The variables are density ρ, magnetic-field components B_x, B_y, B_z, and velocity components v_x, v_y, v_z. The dagger denotes an update beyond the training horizon. Lower is better.

Method	K	ρ	B_x	B_y	B_z	v_x	v_y	v_z
Frozen Transolver	0	0.20546	0.17697	0.14049	0.13853	0.12448	0.13157	0.13480
FOCO	1	0.18178	0.15175	0.12110	0.12087	0.10608	0.11593	0.11708
FOCO	2	**0.17917**	**0.14610**	**0.11656**	**0.11655**	**0.10214**	**0.11225**	**0.11319**
FOCO †	3	0.18276	0.14786	0.11802	0.11790	0.10356	0.11359	0.11465

Table 33: One-step ERA5 16× cross-operator transfer. Refined entries report absolute performance followed by relative improvement in parentheses (RFNE reduction or ACC increase). Bold entries are matched-surrogate controls.

		Evaluation surrogate		
Metric	Corrector trained on	FNO	TFNO	Transolver
RFNE ↓	Frozen base	0.25836	0.24087	0.21921
	FNO	**0.14757 (42.88%)**	0.14762 (38.72%)	0.14767 (32.64%)
	TFNO	0.14797 (42.73%)	**0.14781 (38.64%)**	0.14785 (32.55%)
	Transolver	0.15066 (41.69%)	0.14952 (37.93%)	**0.14859 (32.22%)**
ACC ↑	Frozen base	0.85394	0.87175	0.88040
	FNO	**0.92817 (8.69%)**	0.92816 (6.47%)	0.92813 (5.42%)
	TFNO	0.92789 (8.66%)	**0.92794 (6.45%)**	0.92794 (5.40%)
	Transolver	0.92665 (8.52%)	0.92723 (6.36%)	**0.92745 (5.34%)**

Table 34: Two-step TR-2D cross-operator transfer. Refined entries report absolute performance followed by relative error reduction in parentheses. Bold entries are matched-surrogate controls.

		Evaluation surrogate		
Metric	Corrector trained on	FNO	TFNO	Transolver
NMSE ↓	Frozen base	0.07623	0.06396	0.10149
	FNO	**0.06524 (14.41%)**	0.05594 (12.53%)	0.08397 (17.26%)
	TFNO	0.06342 (16.80%)	**0.05515 (13.77%)**	0.08046 (20.72%)
	Transolver	0.04815 (36.84%)	0.04554 (28.79%)	**0.04972 (51.01%)**
VRMSE ↓	Frozen base	0.2989	0.2777	0.3491
	FNO	**0.21913 (26.69%)**	0.20759 (25.26%)	0.25310 (27.49%)
	TFNO	0.22236 (25.61%)	**0.20993 (24.41%)**	0.25550 (26.81%)
	Transolver	0.21920 (26.67%)	0.21396 (22.96%)	**0.21357 (38.82%)**

Table 35: Two-step Active Matter cross-operator transfer. Refined entries report absolute performance followed by relative error reduction in parentheses. Bold entries are matched-surrogate controls.

		Evaluation surrogate		
Metric	Corrector trained on	FNO	TFNO	Transolver
NMSE ↓	Frozen base	0.008459	0.007472	0.01555
	FNO	**0.001643 (80.57%)**	0.001451 (80.58%)	0.005828 (62.53%)
	TFNO	0.001838 (78.27%)	**0.001478 (80.22%)**	0.006052 (61.09%)
	Transolver	0.003783 (55.28%)	0.003182 (57.41%)	**0.001780 (88.55%)**
VRMSE ↓	Frozen base	0.1990	0.2165	0.1737
	FNO	**0.14969 (24.76%)**	0.15800 (27.04%)	0.14531 (16.35%)
	TFNO	0.15694 (21.12%)	**0.15561 (28.14%)**	0.15238 (12.27%)
	Transolver	0.13466 (32.32%)	0.15195 (29.83%)	**0.10169 (41.46%)**

Table 36: Two-step three-dimensional MHD cross-operator transfer. Refined entries report absolute performance followed by relative error reduction in parentheses. Bold entries are matched-surrogate controls. Only $K = 2$ refinement is reported.

		Evaluation surrogate		
Metric	Corrector trained on	FNO	TFNO	Transolver
NMSE ↓	Frozen base	0.08645	0.06633	0.01892
	FNO	**0.01851 (78.59%)**	0.01536 (76.85%)	0.01467 (22.48%)
	TFNO	0.01604 (81.44%)	**0.01505 (77.32%)**	0.01610 (14.91%)
	Transolver	0.06320 (26.90%)	0.04689 (29.31%)	**0.01376 (27.28%)**
VRMSE ↓	Frozen base	0.2951	0.2695	0.1503
	FNO	**0.13631 (53.81%)**	0.13259 (50.81%)	0.13188 (12.27%)
	TFNO	0.13690 (53.61%)	**0.13434 (50.16%)**	0.13903 (7.51%)
	Transolver	0.24399 (17.33%)	0.22238 (17.50%)	**0.12656 (15.81%)**

Table 37: Spectral error after one $K = 16$ FOCO trajectory. $R_{16}^{(\mathcal{B})}$ is the median sample-normalized Fourier-magnitude error in frequency band $\mathcal{B}$; lower is better. "Red." denotes $100(1 - R_{16}^{(\mathcal{B})})$. NMSE reduction is also measured relative to the same frozen FNO prediction. Bold entries mark the most strongly reduced spectral band within each benchmark.

Dataset	NMSE red.	Low frequency		Mid frequency		High frequency	
	(%)	R_{16}	Red. (%)	R_{16}	Red. (%)	R_{16}	Red. (%)
TR-2D	14.94	**0.4496**	**55.04**	0.6943	30.57	0.7912	20.88
Active Matter	79.84	0.2361	76.39	0.0690	93.10	**0.0138**	**98.62**
MHD	78.48	**0.1733**	**82.67**	0.4302	56.98	0.5940	40.60
Wave Layer	91.96	0.0679	93.21	0.0108	98.92	**0.0043**	**99.57**

G.9 Spectral Convergence under Test-Time Scaling

For $K \geq 16$, we cap the final two updates at 0.25 to mitigate late-stage over-correction.

We further examine how additional test-time computation changes the error across spatial frequencies. Following the spectral diagnostic of IRNO (Liu et al., 2026), we record every intermediate state along one teacher-forced, terminal-capped $K = 16$ FOCO trajectory,

$$u_i^{(0)} = B_\phi(x_i), \qquad u_i^{(k+1)} = u_i^{(k)} + \eta_k G_\theta(x_i, u_i^{(k)}), \qquad k = 0, \ldots, 15, \tag{100}$$

with

$$\eta_k = \begin{cases} (16-k)^{-1}, & 0 \leq k \leq 13, \\ 1/4, & k \in \{14, 15\}. \end{cases} \tag{101}$$

Thus, the final two updates are capped at 0.25 (There is tiny over-correction over the last two steps if no terminal-capped when $K \geq 16$, such operation is similar to Flow Matching x-prediction truncation around $t = 1$ (Li & He, 2025), without this,). Importantly, k indexes a prefix of this single $K = 16$ trajectory; it is not the terminal state of an independently scheduled $K = k$ run. The corrector is not retrained as the test-time budget increases.

Spectral metric. For sample i, output field c, refinement state k, and retained Fourier mode ξ, we define the Fourier-magnitude error

$$e_{i,c}^{(k)}(\xi) = \left(\left| \mathcal{F}[u_{i,c}^{(k)}](\xi) \right| - |\mathcal{F}[y_{i,c}](\xi)| \right)^2. \tag{102}$$

We use an orthonormal real FFT over all spatial dimensions in the model-standardized state space, without spatial demeaning. The retained modes are divided into equal-count low-, mid-, and high-frequency tertiles according to their radial frequencies. For band $\mathcal{B}$, the normalized error for sample i is

$$r_{i,k}^{(\mathcal{B})} = \frac{C^{-1} \sum_{c=1}^{C} \sum_{\xi \in \mathcal{B}} e_{i,c}^{(k)}(\xi)}{C^{-1} \sum_{c=1}^{C} \sum_{\xi \in \mathcal{B}} e_{i,c}^{(0)}(\xi) + 10^{-10}}, \qquad R_k^{(\mathcal{B})} = \underset{i \in \mathcal{D}_{\text{test}}}{\text{median}}\, r_{i,k}^{(\mathcal{B})}. \tag{103}$$

Hence, $R_0^{(\mathcal{B})} \approx 1$, a value below one indicates improvement over the frozen FNO, and $100(1 - R_k^{(\mathcal{B})})$ is the relative spectral error reduction. We use all available one-step test windows: 873 for TR-2D, 2,002 for Active Matter, 960 for MHD, and 10,240 for Wave Layer.

Table 37 and Figures 8, 9, 10, and 11 show that the final refinement state reduces error in every frequency band on all four systems, despite FOCO using no explicit spectral loss. Active Matter and Wave Layer exhibit the strongest high-frequency recovery, with reductions of 98.62% and 99.57%, respectively. In contrast, TR-2D and MHD obtain their largest reductions in the low-frequency band, at 55.04% and 82.67%. TR-2D has a small transient increase in mid- and high-frequency error during the first two small updates, after which both fall below the frozen-base level. On MHD, the high-frequency ratio reaches its minimum at $k = 12$ and then rises mildly from 0.5658 to 0.5940 at $k = 16$, while remaining 40.60% below the base. Thus, additional refinement improves the full spectrum, but the ordering and late-stage saturation of spatial scales are system dependent.

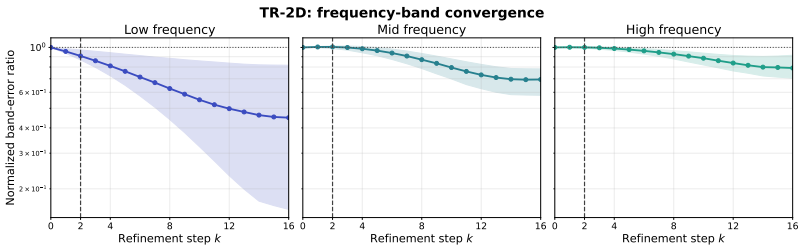


Figure 8: TR-2D frequency-band error along a single terminal-capped $K = 16$ trajectory. Curves and shaded regions show the test-set median and interquartile range, respectively, on a logarithmic scale. The horizontal reference is the frozen FNO error, and the vertical dashed line at $k = 2$ marks the training refinement horizon.

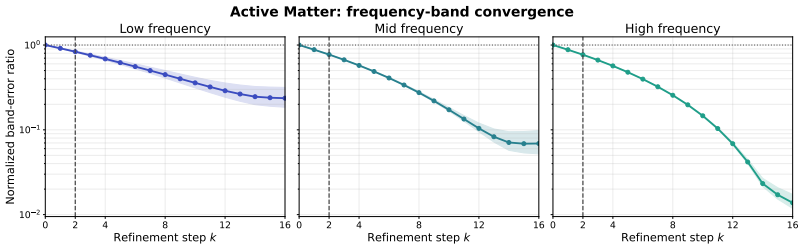


Figure 9: Active Matter frequency-band error along a single terminal-capped $K = 16$ trajectory. Curves and shaded regions show the test-set median and interquartile range, respectively. The horizontal reference is the frozen FNO error, and the vertical dashed line at $k = 2$ marks the training refinement horizon.

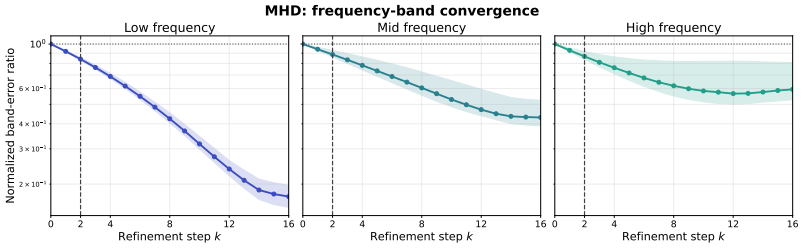


Figure 10: MHD frequency-band error along a single terminal-capped $K = 16$ trajectory. Curves and shaded regions show the test-set median and interquartile range, respectively. The horizontal reference is the frozen FNO error, and the vertical dashed line at $k = 2$ marks the training refinement horizon.

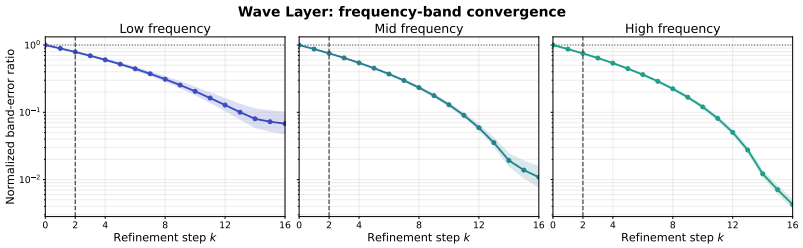


Figure 11: Wave Layer frequency-band error along a single terminal-capped $K = 16$ trajectory. Curves and shaded regions show the test-set median and interquartile range, respectively. The horizontal reference is the frozen FNO error, and the vertical dashed line at $k = 2$ marks the training refinement horizon. In all four figures, $k > 2$ evaluates additional test-time computation without retraining.

G.10 ERA5 Interpolation and Temporal Extrapolation Cases

We complement the aggregate ERA5 results with two examples from the interpolation test year (2009) and one example from the temporal-extrapolation years (2015). Let $u_0 = B_\phi(x)$ and $u_1 = u_0 + G_\theta(x, u_0)$ denote the frozen-FNO and one-step FOCO predictions. For each case, we visualize the target and both predictions, together with the required correction $y - u_0$, the correction produced by FOCO, $u_1 - u_0$, and the remaining residual $y - u_1$. These quantities obey

$$y - u_0 = (u_1 - u_0) + (y - u_1). \tag{104}$$

The water-vapor zoom is selected from the target gradient and is therefore independent of either model. Within each case and physical variable, all residual maps use the same zero-centered color scale, permitting direct comparison among the three residual columns. Scales are selected separately across cases and variables and should not be compared between figures.

Figures 12– 13 show that one FOCO correction restores spatial structures smoothed by the frozen FNO, including fine kinetic-energy and water-vapor features. The predicted update follows the location and sign of the required correction and leaves a visibly weaker remaining residual. Across the four cases, one correction reduces physical relative L_2 error by 40.5–41.5% for 10-m kinetic energy, 48.6–53.3% for 2-m temperature, and 37.1–42.1% for total-column water vapor. The two extrapolation examples retain improvements comparable to those from the interpolation year. These selected cases provide qualitative evidence; the dataset-level comparison is reported in the main results.

G.11 Active Matter One-Step Refinement

We visualize six test trajectories selected *a priori* from their physical coefficients, spanning $\zeta \in [3, 17]$ and $|\alpha| \in [1, 5]$ with $L = 10$. All cases use frames 35–38 to predict frame 39. We show orientation anisotropy $D_{xx} - D_{yy}$ and strain-rate shear E_{xy}. Each row compares the ground truth, frozen FNO, and two-step FOCO prediction, followed by their absolute errors. State and error limits are shared across methods within each row and case.

Across these coefficient-stratified examples, FOCO reduces both the $D_{xx} - D_{yy}$ and E_{xy} errors in all four cases. The cases were not selected by sample-level refinement gain.

G.12 TR-2D Refinement and Autoregressive Rollouts

We visualize four TR-2D test trajectories spanning different cooling times, $t_{\mathrm{cool}} \in \{0.06, 0.18, 0.56, 1.78\}$. The cases are selected *a priori* using evenly spaced trajectory indices $\{1, 3, 5, 7\}$, rather than according to refinement performance. All cases use frames 48–51 as the initial history. The teacher-forced comparison predicts frame 52, whereas the rollout recursively predicts frames 52–58 without injecting ground-truth states after initialization. At every temporal transition, FOCO applies two correction updates ($K = 2$); the symbol h denotes temporal forecast horizon.

Within each teacher-forced panel, the target, FNO, and FOCO predictions share one color scale for each physical variable, while the two absolute-error maps share a second scale. The rollout panels show density at $h \in \{1, 3, 5, 7\}$ and the terminal errors at $h = 7$. Display ranges are shared within each case but selected separately across cases.

Across these four parameter-stratified cases, two-step FOCO refinement reduces the sample-mean one-step VRMSE by 1.4%, 24.3%, 43.6%, and 51.7%, respectively. At rollout horizon $h = 7$, the corresponding reductions are 15.0%, 21.7%, 40.1%, and 53.4%. The first case shows that a modest one-step change can still reduce accumulated rollout error, while the remaining cases exhibit increasingly visible recovery of fine spatial structure. These examples complement the aggregate results and illustrate that FOCO remains effective when its corrected outputs are recursively fed back into the frozen surrogate.

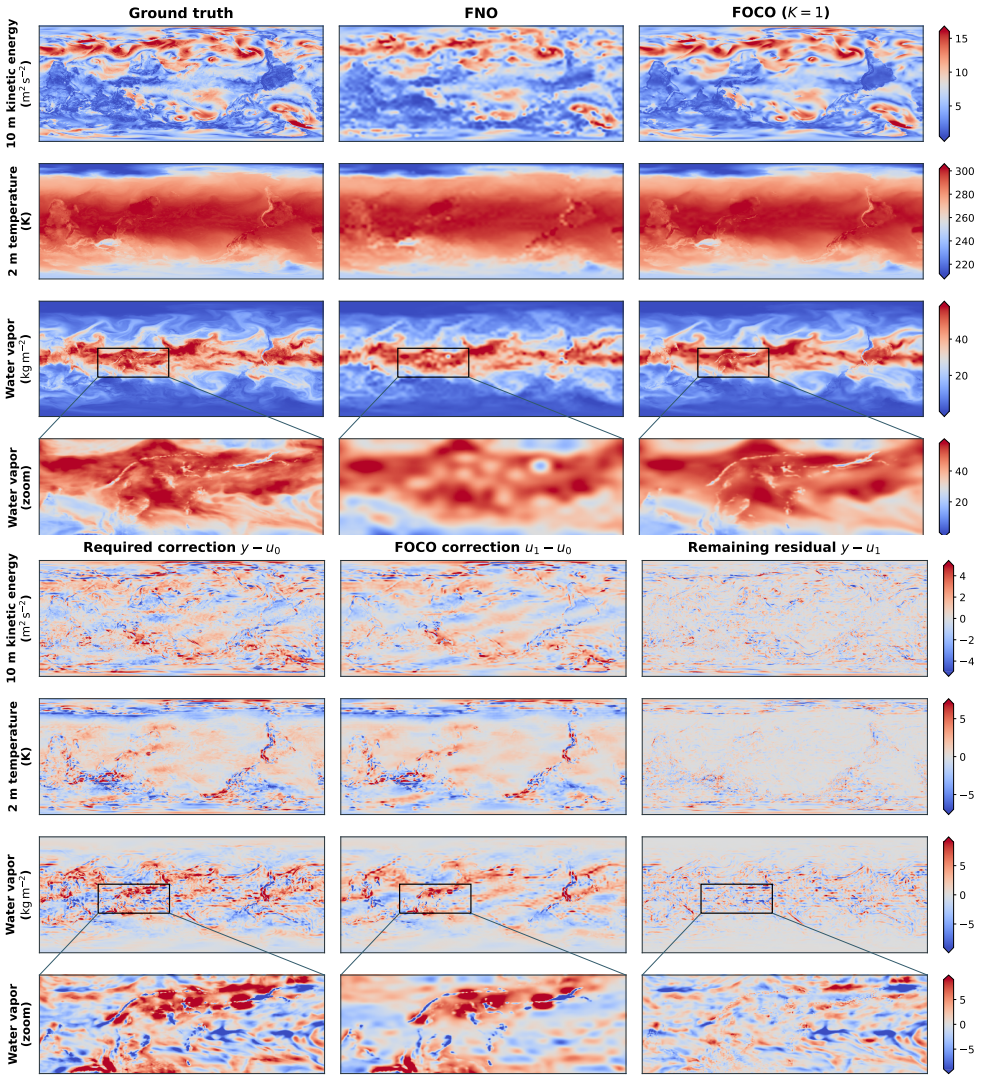


Figure 12: ERA5 16× one interpolation example from 2009 . Top: ground truth, frozen FNO, and one-step FOCO prediction for the three physical variables, including the water-vapor zoom. Bottom: required correction, FOCO correction, and remaining residual.

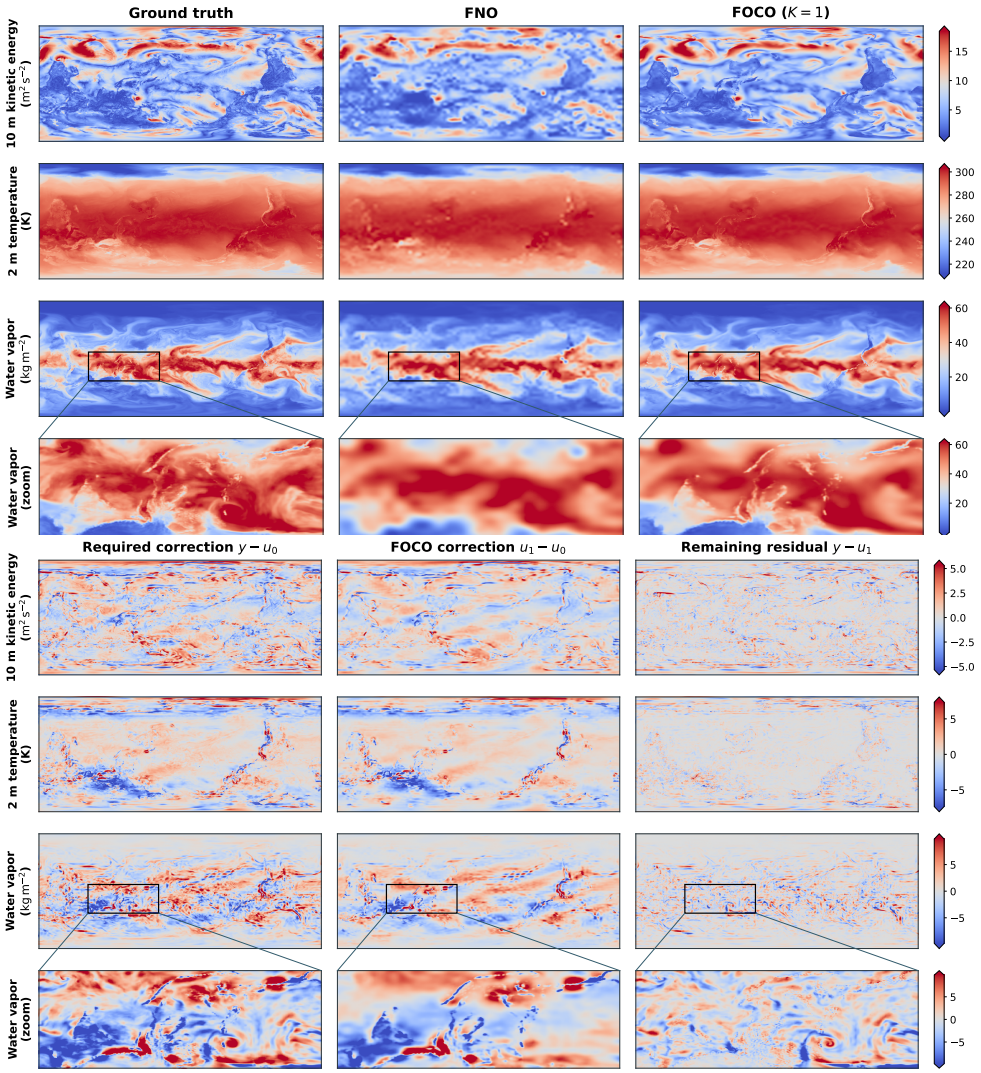


Figure 13: ERA5 16× temporal-extrapolation example from 2015

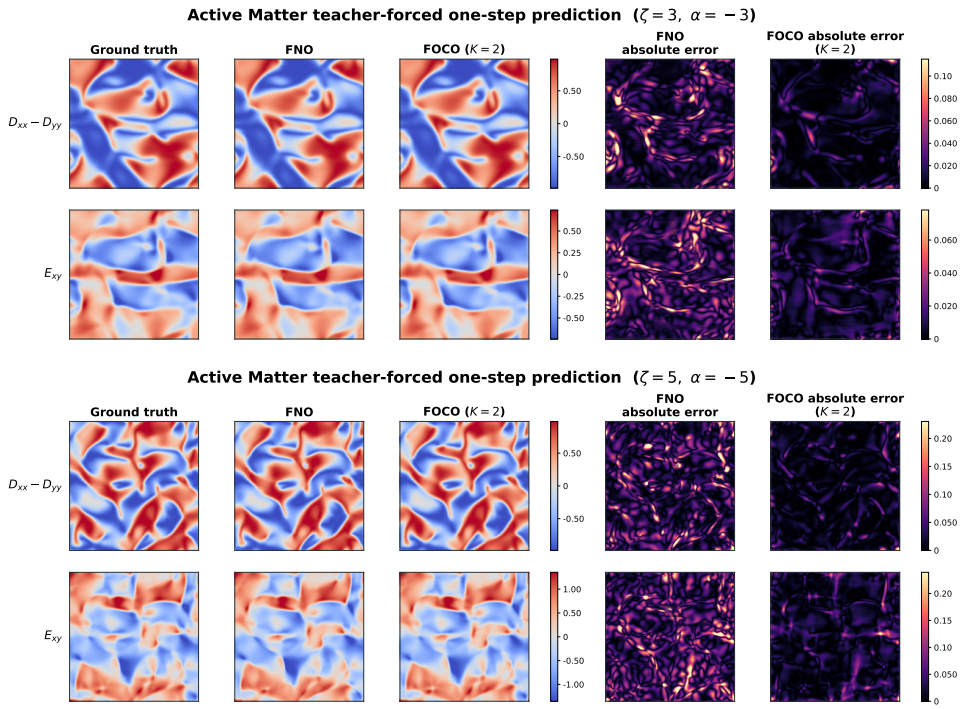


Figure 14: Active Matter one-step refinement for trajectories 15 (top; $\zeta = 3$, $\alpha = -3$) and 21 (bottom; $\zeta = 5$, $\alpha = -5$).

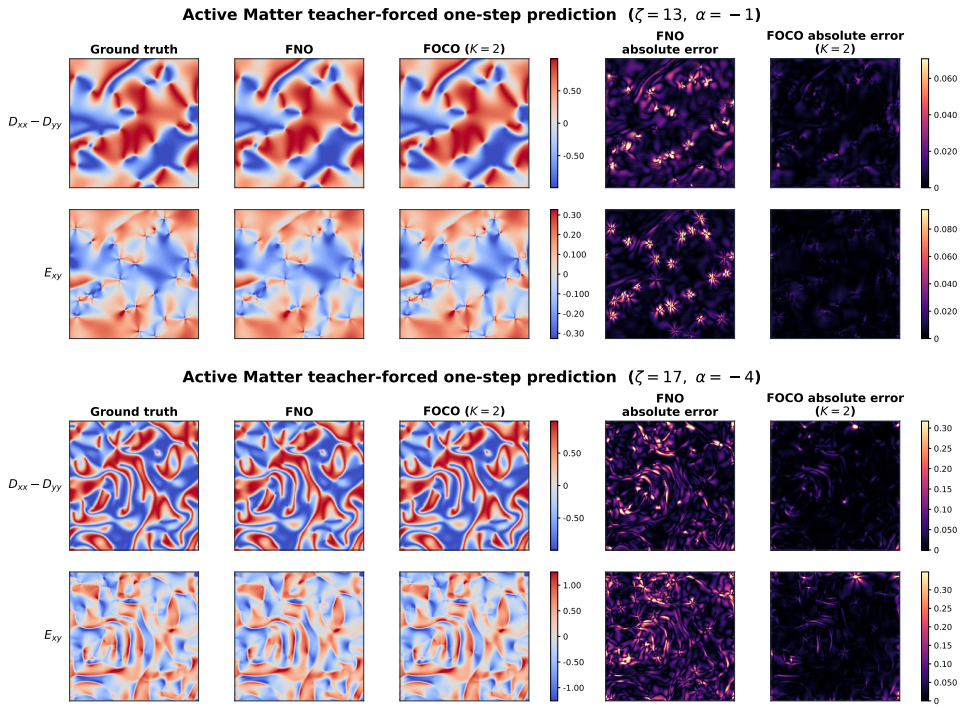


Figure 15: Active Matter one-step refinement for trajectories 8 (top; $\zeta = 13$, $\alpha = -1$) and 14 (bottom; $\zeta = 17$, $\alpha = -4$).

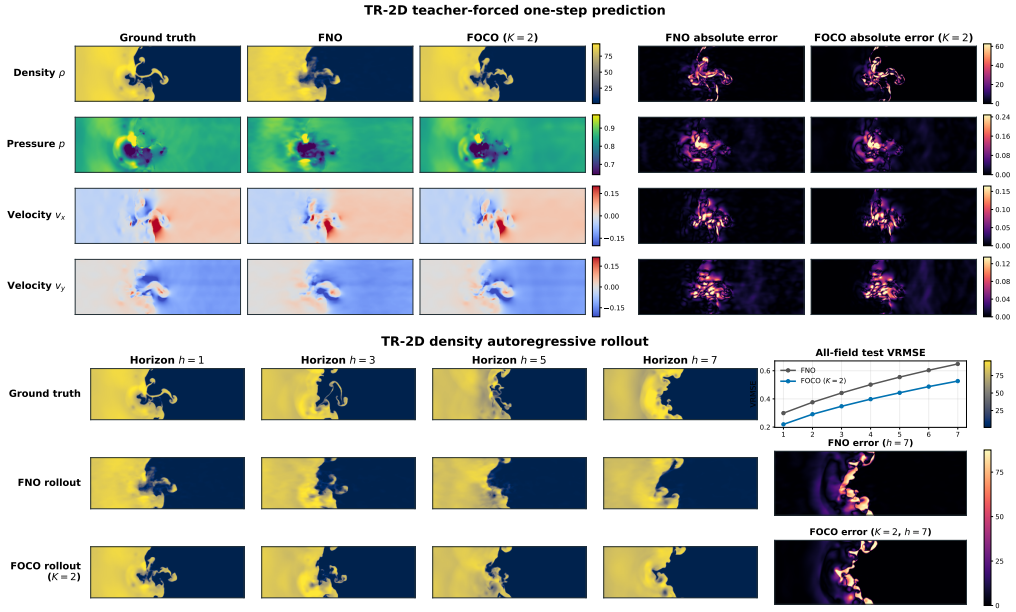


Figure 16: TR-2D test trajectory 1 ($t_{\text{cool}} = 0.06$). Top: one-step teacher-forced prediction for density, pressure, and both velocity components. Bottom: recursive density rollout through $h = 7$. Two FOCO updates are applied after every FNO forecast. The inset reports the dataset-level, all-field rollout VRMSE.

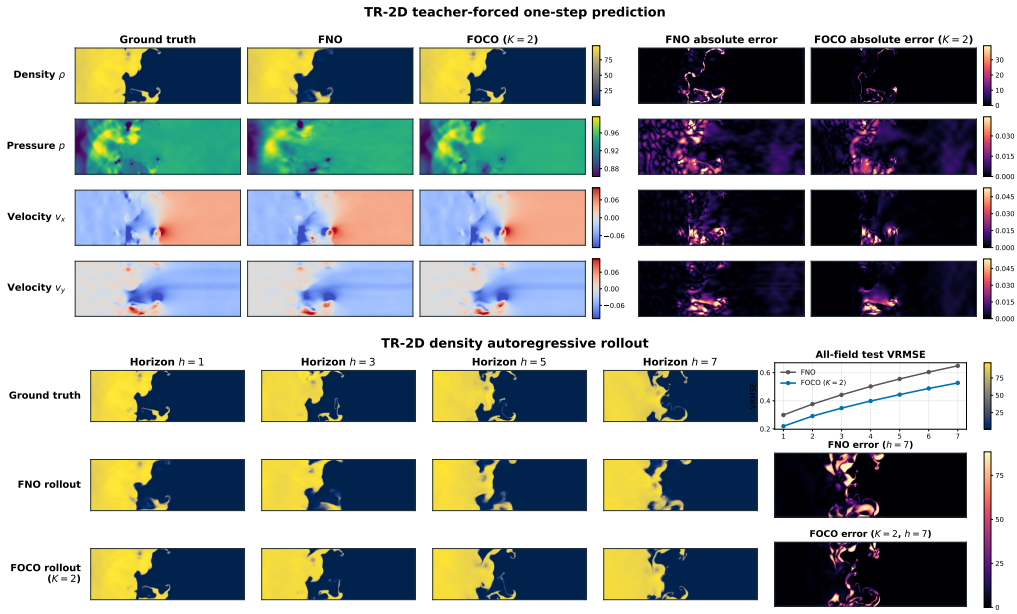


Figure 17: TR-2D test trajectory 3 ($t_{\text{cool}} = 0.18$), using the same teacher-forced and recursive rollout protocols as Figure 16.

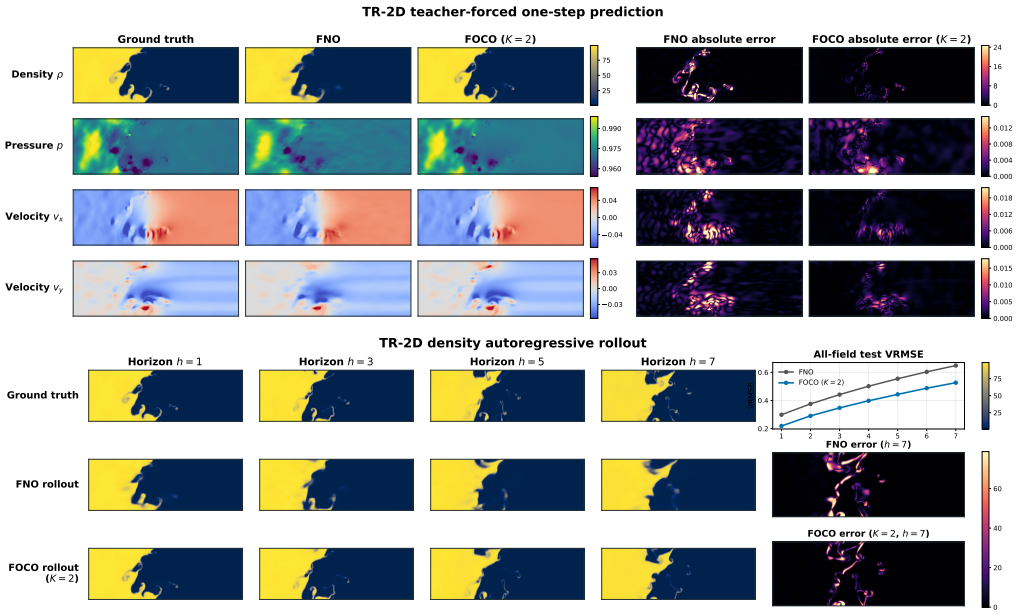


Figure 18: TR-2D test trajectory 5 ($t_{\text{cool}} = 0.56$). FOCO sharpens the one-step prediction and reduces error accumulation over the recursive rollout.

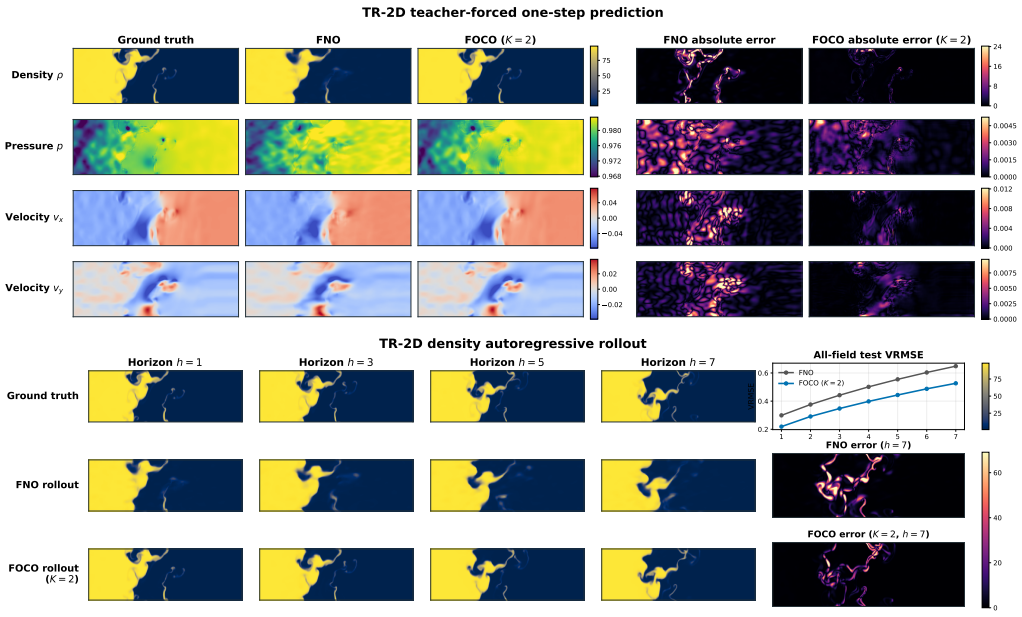


Figure 19: TR-2D test trajectory 7 ($t_{\text{cool}} = 1.78$), evaluated with two FOCO updates per temporal transition and no teacher forcing during rollout.