DTNO: DIRECT-TIME NEURAL OPERATORS FOR ONE-PASS SPATIOTEMPORAL FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Autoregressive neural PDE surrogates advance a system one or a few steps at a time, causing inference cost to grow with forecast horizon and allowing prediction errors to accumulate through repeated feedback. We study the Direct-Time Neural Operator (DTNO), a finite-time neural operator that predicts a requested future state in a single network evaluation. DTNO decomposes forecasting into a horizon-independent encoder, a lead-time-conditioned latent evolution operator, and a residual decoder. Lead time modulates the Fourier operator blocks through feature-wise linear modulation, enabling a single model to predict across multiple forecast horizons without autoregressive rollout.

We evaluate DTNO on five spatiotemporal systems ranging from reaction-diffusion dynamics to reacting flows, using a common FNO backbone, matched training splits, and the same evaluation protocol as an autoregressive counterpart. For a requested future state, DTNO uses a single forward pass rather than a number of rollout steps that grows with the forecast horizon. Across these systems, autoregressive forecasting is more accurate at short lead times, whereas DTNO consistently achieves superior accuracy at medium horizons, with accuracy transition points emerging between $h = 8.1$ and $h = 19.1$. At $h = 32$, DTNO achieves lower relative L_2 field error than the matched autoregressive baseline on all five datasets, while eliminating recursive error feedback and requiring no sequential rollout between the observed state and the queried horizon. DTNO also exhibits substantially lower variation across repeated training runs.

These results show that direct finite-time prediction provides a practical alternative to autoregressive rollout for medium- and long-horizon spatiotemporal forecasting. By conditioning a shared neural operator directly on forecast time, DTNO combines horizon-flexible prediction, constant-depth inference, and improved long-horizon stability within a single model.

1 INTRODUCTION

Learned surrogates can reduce the cost of repeatedly simulating spatiotemporal PDE systems, but most temporal predictors remain autoregressive: a model advances the state by one or a few steps and is applied repeatedly until the requested forecast horizon is reached. This design is natural for short-time dynamics, but it has two practical costs. First, the number of network evaluations grows linearly with lead time. Second, model outputs are fed back as future inputs, so approximation errors can accumulate and the inference distribution can drift away from the training distribution (Sanchez-Gonzalez et al., 2020; Brandstetter et al., 2022; Lippe et al., 2023).

We consider a different temporal parameterization. Instead of learning only a local transition and composing it at inference, we learn the finite-time map from an observed history and a requested lead time directly to the target state. We call the resulting model a *Direct-Time Neural Operator* (DTNO). DTNO separates the model into three parts: a horizon-independent encoder, a lead-time-conditioned latent evolution operator, and a horizon-independent residual decoder. The requested lead time modulates each latent Fourier block through FiLM. A single trained model can therefore answer many temporal queries, while every query uses one forward pass regardless of the number of physical time steps between input and target. This is a constant-depth claim with respect to forecast horizon; it does not change the spatial complexity of the neural operator.

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.

Direct querying of a future state is not new: Poseidon evaluates one lead-time-conditioned model directly at a requested time or composes it with itself (Herde et al., 2024), TC-UNet and time-as-coordinate DeepONets support arbitrary temporal queries (Ovadia et al., 2025; Lu et al., 2021), and CFO integrates a learned continuous-time vector field (Hou et al., 2026). We do not claim lead-time conditioning or one-pass querying as a contribution. What these works leave open is a forecasting question: when a direct model is compared with a *separately trained* autoregressive model of the same spatial backbone, from a history window taken mid-trajectory and over horizons of hundreds of frames, where does each win, why, and within what limits? Answering it requires the full error-versus-horizon curve rather than a final-time error, a climatology bound on that curve, and controls for the backbone and for how deeply the autoregressive model is trained on its own rollouts.

We instantiate DTNO with an FNO backbone (Li et al., 2021) and compare it with a matched rollout-trained FNO on five systems: Gray–Scott reaction–diffusion and Rayleigh–Bénard convection from THE WELL (Ohana et al., 2024), a cylinder wake and a swirl-flame combustor from RealPDEBench (Hu et al., 2026), and a lifted hydrogen jet from BLASTNet (Chung et al., 2022; Sharma et al., 2025). The two models share the encoder, backbone and decoder, the trajectory-level splits, the horizon grid and the metrics, and differ by 0.9% in parameter count. What differs is the temporal supervision—sampled target horizons for the direct model, rollouts of its own predictions for the autoregressive one—and that difference is the object of study rather than a confound.

Contributions. (1) DTNO, a lightweight, history-conditioned instance of time-conditioned operator learning on a standard FNO, together with an autoregressive counterpart that differs from it only in temporal parameterization. (2) A controlled comparison on five systems: the rollout is the better one-step predictor on all five, the direct model overtakes it between $h = 8.1$ and $h = 19.1$ despite a $38\times$ range of one-step error, is more accurate at $h = 32$ on all five, stays bounded where the rollout diverges, and varies by about one percent across seeds where the rollout's long-horizon error spans two orders of magnitude. (3) Controls that bound these findings: a U-Net backbone moves the crossover and on one dataset removes it, and this is not a capacity effect—an FNO widened to the U-Net's size neither improves nor stops diverging—while precision and Fourier-mode controls change nothing; two climatology baselines show where a bounded forecast stops carrying trajectory-specific information, which on one dataset is almost immediate. (4) Two negative design results: a latent semigroup loss improves composition consistency but not accuracy, and lead-time conditioning does not by itself imply interpolation to unseen horizons.

2 RELATED WORK

Neural PDE surrogates. Neural operators learn maps between function spaces and provide a common framework for surrogate modeling of parametrized PDEs (Lu et al., 2021; Li et al., 2021; Kovachki et al., 2023). Other learned simulators use graph message passing or related local transition models (Sanchez-Gonzalez et al., 2020; Brandstetter et al., 2022). Long autoregressive rollouts are known to be sensitive to error accumulation and distribution shift; training on model-generated states, temporal bundling, noise injection, and iterative refinement are among the strategies used to improve rollout stability (Sanchez-Gonzalez et al., 2020; Brandstetter et al., 2022; Lippe et al., 2023). Our autoregressive baseline is trained on four-step rollouts of its own predictions (Section 3.3), so the comparison is not against a purely teacher-forced one-step model.

Time-conditioned and continuous-time operators. Poseidon (Herde et al., 2024) is the closest prior work. It learns solution operators of time-dependent PDEs with a multiscale transformer whose layer normalizations are modulated by lead time, so one trained model can be queried directly at any time and can also be composed with itself at inference. It is trained on all ordered snapshot pairs within a trajectory, using the semigroup property to enlarge the training set, and targets pretraining and few-shot transfer across fifteen downstream tasks, evaluated by the relative error at a final time on trajectories of roughly ten to twenty snapshots; its FNO baselines are themselves time-conditioned. TC-UNet (Ovadia et al., 2025) and DeepONets with time as a trunk coordinate (Lu et al., 2021) also support direct temporal queries, while CFO (Hou et al., 2026) integrates a learned continuous-time vector field. DTNO uses the same idea—a lead-time-conditioned operator queried in one pass—and we make no claim to it. Our setting differs in what is measured: forecasting from a K-frame history window taken mid-trajectory, over horizons of up to several hundred frames on

records of 151 to 3990 frames, against a separately rollout-trained autoregressive model of matched backbone, so that the error-versus-horizon curve, its crossover and its convergence to climatology are the objects of study.

Structure and long-horizon evaluation. Poseidon uses the semigroup property to construct additional training pairs (Herde et al., 2024); we instead test it as an optional latent consistency loss. For long-horizon evaluation, field error alone can also be misleading: a deterministic squared-error predictor may remain bounded while approaching a conditional mean. We therefore report climatology baselines and spectral metrics in addition to relative field error. This evaluation is intended to separate numerical stability from trajectory-specific predictive information. Additional related literature is summarized in Appendix C.

3 METHOD

3.1 DIRECT FINITE-TIME PREDICTION

Let $U(t) \in \mathbb{R}^{H\times W\times C}$ denote the resolved physical state and $X_t = [U(t-K+1), \dots, U(t)]$ a K-frame history. For a requested lead time $\Delta T = h\Delta t$, DTNO learns

$$G_\theta(X_t, \Delta T) \approx U(t + \Delta T), \tag{1}$$

instead of recursively composing a short-step transition. Consequently, $N_{\text{eval}}^{\text{DTNO}} = 1$ for every query, whereas an autoregressive one-step model uses $N_{\text{eval}}^{\text{AR}} = h$. We use *one-pass* and *constant-depth* only in this temporal sense.

3.2 LEAD-TIME-CONDITIONED LATENT PROPAGATION

DTNO factorizes the finite-time map as

$$\boxed{G_\theta(X_t, \tau) = U(t) + D_\theta[\Phi_\theta(E_\theta(X_t), \tau)]\,,} \tag{2}$$

where $\tau = \Delta T/T_{\text{scale}}$. The encoder E_θ receives the history together with spatial coordinates and produces a latent state z_0. The propagator $\Phi_\theta(z_0, \tau)$ is the only component that receives the requested lead time. The decoder D_θ reconstructs a state increment, which is added to the latest observation. This factorization keeps representation and reconstruction shared across horizons and assigns temporal variation to the latent evolution operator. Since E_θ and D_θ are pointwise layers with under 15k parameters, the propagator carries essentially all capacity, and lead time effectively conditions the whole network, as in Poseidon.

We instantiate Φ_θ with residual Fourier-operator blocks. A Fourier embedding of τ is passed through a small MLP, and each block receives a separate pair of FiLM parameters (Perez et al., 2018). If s_ℓ is the spatial update in block ℓ, time conditioning has the form $\mathrm{FiLM}_\ell(s_\ell; \tau) = \gamma_\ell(\tau) \odot s_\ell + \beta_\ell(\tau)$. Thus lead time changes the latent propagation at every depth without changing the Fourier basis or the encoder/decoder. Full block equations are given in Appendix A.

3.3 MATCHED AUTOREGRESSIVE MODEL

AR-FNO-R uses the same encoder, Fourier backbone, LayerScale residual blocks, and decoder, but removes the time embedding and FiLM heads and applies a local transition recursively. It is trained on rollouts of its own predictions, so that its training inputs include model-generated states. Each sample draws a depth $r \in \{1, \dots, 4\}$, but a batch is unrolled to the largest depth drawn within it, so at the per-device batch sizes used at least $1-(3/4)^8 \approx 90\%$ of batches are unrolled for all four steps; we therefore treat it as a four-step rollout-trained baseline. Section 5.3 varies this depth explicitly. On Gray–Scott, AR-FNO-R has $8{,}414{,}786$ parameters and the DTNO implementation (DT-FNO) $8{,}565{,}698$; the $150{,}912$ difference is exactly the four FiLM heads and the time-embedding MLP, or 1.8%. The comparison therefore keeps the dominant spatial capacity matched.

3.4 TRAINING PAIRS AND HORIZON SAMPLING

DT-FNO is trained on pairs $\big(X_t,\, h,\, U(t + h\Delta t)\big)$ drawn online from training trajectories, with a fixed budget of $20{,}000$ pairs per epoch. The horizon is drawn by choosing one of the octave bins

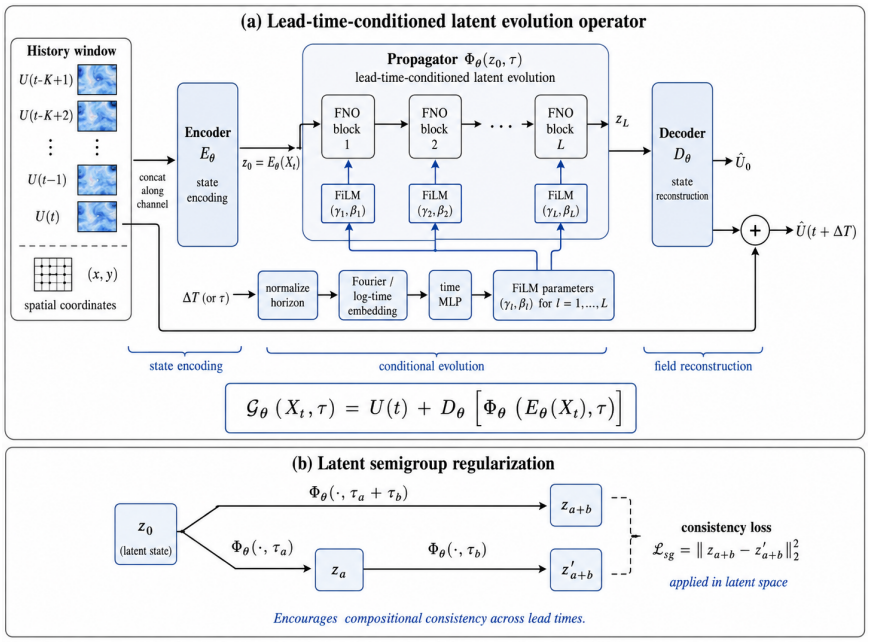

Figure 1: **DTNO architecture.** (a) A horizon-independent encoder maps the input history to z_0. The requested lead time modulates every latent Fourier block through FiLM, and a shared residual decoder predicts the future state. (b) The optional semigroup loss compares direct latent evolution with a composition that reaches the same lead time.

$[1, 2]$, $[3, 4]$, $[5, 8]$, $[9, 16]$, $[17, 32]$, $[33, 64]$, $[65, 128]$ uniformly at random and then h uniformly within it; bins lying above $h_{\max}$ are dropped, so there are seven at $h_{\max}$=128 and six on Rayleigh–Bénard, where $h_{\max}$=64. A valid history window and target are then drawn from a training trajectory. Uniform sampling on $[1, 128]$ would put 94% of pairs above h=8; this scheme puts 43% at or below it, so the short-horizon regime against which the crossover is measured is not starved. Unlike all2all training, whose pair count grows quadratically with record length, the budget is independent of trajectory length; as there, each pair costs one forward and backward pass whatever its horizon. The autoregressive arms do not use this sampler (Section 3.3).

3.5 Optional semigroup regularization

For autonomous dynamics finite-time evolution should compose. SG-DT-FNO adds $\lambda_{\mathrm{SG}}\mathcal{L}_{\mathrm{SG}} + \lambda_{\mathrm{id}}\mathcal{L}_{\mathrm{id}}$ to the prediction loss, with

$$\mathcal{L}_{\mathrm{SG}} = \frac{\|\Phi_\theta(z_0, \tau) - \Phi_\theta(\Phi_\theta(z_0, \tau_a), \tau_b)\|_2^2}{\|\Phi_\theta(z_0, \tau)\|_2^2}, \qquad \mathcal{L}_{\mathrm{id}} = \|\Phi_\theta(z_0, 0) - z_0\|_2^2, \tag{3}$$

where $\tau_a + \tau_b = \tau$ is the supervised horizon, split with h_a uniform on $\{1, \ldots, h-1\}$, and λ_{SG} is warmed up from zero. The inference architecture is identical to DT-FNO's, so this ablation tests only the loss.

4 Experimental setup

4.1 Datasets

Table 1 summarizes the five benchmarks. They were selected to cover distinct temporal regimes while retaining a common grid-based operator interface. Gray–Scott provides a fully observed autonomous periodic system; Cylinder and Rayleigh–Bénard add non-reacting transport; Lifted H_2 and RealPDEBench add reacting-flow dynamics and higher-dimensional state vectors.

Table 1: Five evaluation datasets. T_{pred} is the horizon at which a persistence forecast reaches E_{field}=0.3, measured from data before model training. The datasets cover reaction–diffusion, non-reacting fluid flow, buoyant transport, and two reacting-flow settings.

	mechanism added	traj.	frames $\times$ Δt	grid, ch.	held-out axis	T_{pred}
Experiment A — general PDE/CFD dynamics						
A1 Gray–Scott	reaction, no advection	1200	$1001 \times 10\,\mathrm{s}$	128^2, 2	initial condition	8.6
A2 Cylinder	advection, wake	92	$3990 \times 2.5\,\mathrm{ms}$	128×256, 3	Reynolds	8.1
A3 Rayleigh–Bénard	buoyant transport	1750	$200 \times 0.25\,\mathrm{s}$	128^2, 4	(Ra, Pr)	1.8
Experiment B — reacting flows						
B1 Lifted H_2	transport + chemistry	8	$151 \times 5\,\mu\mathrm{s}$	160×200, 5	Reynolds	—
B2 RealPDEBench	coupled multiscale	30	$2001 \times 250\,\mu\mathrm{s}$	128^2, 13	fuel composition	4–6

Sources. Gray–Scott and Rayleigh–Bénard come from THE WELL (Ohana et al., 2024), Cylinder and the combustion set from RealPDEBench (Hu et al., 2026), and Lifted H_2 from BLASTNet (Chung et al., 2022; Sharma et al., 2025). Channel selection, spatial subsampling, padding conventions and standardization are given in Appendix B; the one architectural setting that legitimately differs between datasets is the FNO padding, which is zero on the periodic Gray–Scott domain and replicate elsewhere, applied identically to all three models.

Split protocol. We hold out complete trajectories before any window construction, because random window splitting leaks strongly overlapping samples: $(U_{100:104} \to U_{120})$ and $(U_{105:109} \to U_{121})$ contain nearly identical states. This is not hypothetical for Cylinder, whose released index is window-level: under it all 92 trajectories appear in training and 43 also appear in validation and test, with only the (trajectory, start-frame) pairs disjoint. We therefore build our own trajectory-held-out split stratified over Reynolds-number quantile bins and do not report numbers under the released protocol; our Cylinder results are for this reason, and because we train at twice their resolution without their modality masking, not comparable with the RealPDEBench leaderboard. For THE WELL datasets we honour the published file-level split, stratifying over the six Gray–Scott (f, k) sets and over $(\log_{10}\mathrm{Ra}, \mathrm{Pr})$. RealPDEBench combustion is stratified by NH_3 fraction; for Lifted H_2, $Re_{\mathrm{jet}} = 7500$ is held out.

4.2 MODELS, BASELINES AND METRICS

Our principal learned baselines are AR-FNO-R, DT-FNO and SG-DT-FNO. AR-FNO-R is trained on four-step rollouts of its own predictions (Section 3.3); DT-FNO uses the same encoder–backbone–decoder with horizons sampled as in Section 3.4; SG-DT-FNO adds only the losses in Section 3.5. We also evaluate persistence, POD-DMD (Schmid, 2010), a *trajectory climatology* oracle (the time mean of the test trajectory), and a *nearest training climatology* (the mean field of the training trajectory whose history window is closest to the query history). The two climatology baselines are essential for interpreting Section 5.5: a stable, horizon-independent error can otherwise be mistaken for preserved dynamical skill.

The main accuracy metric is relative L_2 error per channel averaged across channels (E_{field}), with physically related groups reported separately; we also evaluate radial energy spectra and an integrated spectral error E_{spec}. Inference is reported as wall-clock latency and as the exact number of model evaluations. Model selection uses validation error integrated over horizons rather than one-step error, which would favour the rollout's training regime. Additional spectral and physics-oriented diagnostics are reported in the appendix where available; the main paper uses the same field-error metric across all five systems to keep the model comparison consistent.

5 RESULTS

5.1 MAIN ACCURACY–COST COMPARISON

DTNO changes the temporal cost from h model evaluations to one. Measured on the same hardware, DT-FNO takes approximately 2.62 ms per query from $h = 1$ to $h = 512$, while AR-FNO-R takes

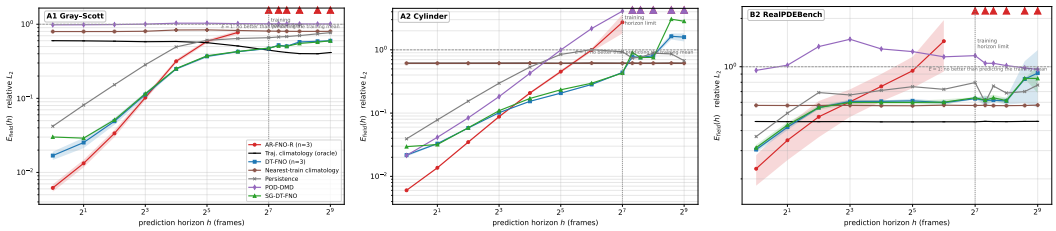

Figure 2: Field error versus forecast horizon for Gray–Scott, Cylinder, and RealPDEBench; Rayleigh–Bénard and Lifted H_2 are shown in Appendix D.3. Bands denote ± 1 s.d. over three seeds where repeated runs are available. Climatology baselines indicate when a bounded prediction no longer carries useful trajectory-specific information.

Table 2: Accuracy across the five datasets. Eval* is the mean of $\min(E_{\text{field}}, 1)$, which remains defined when an autoregressive rollout diverges. AR-FNO-R and DT-FNO are three-seed means except Rayleigh–Bénard (†, single seed); SG-DT-FNO is included for the two reacting-flow datasets shown here. Full baseline tables are in Appendix D.6.

		Eval*	$E_{\text{field}}(1)$	$E_{\text{field}}(8)$	$E_{\text{field}}(32)$	$E_{\text{field}}(128)$	T_{pred}
A1 Gray–Scott	AR-FNO-R	0.5926	**0.0062**	**0.1022**	0.5882	*div.*	15.4
	DT-FNO	**0.3468**	0.0169	0.1124	**0.3683**	**0.4703**	22.8
	clim. (oracle)	0.5027	0.6002	0.5812	0.5639	0.4470	—
A2 Cylinder	AR-FNO-R	0.5978	**0.0060**	**0.0878**	0.4523	2.724	22.0
	DT-FNO	**0.4340**	0.0217	0.1015	**0.2068**	**0.4387**	**70.9**
	clim. (oracle)	0.6140	0.6135	0.6145	0.6123	0.6136	—
A3 Rayleigh–Bénard†	AR-FNO-R	0.5494	**0.1852**	**0.4515**	0.7375	—	**3.0**
	DT-FNO	**0.4707**	0.2107	0.4743	**0.5694**	—	2.2
	clim. (oracle)	0.5286	0.6379	0.5583	0.4620	—	—
B1 Lifted H_2	AR-FNO-R	0.3835	**0.1106**	**0.2333**	0.5534	—	—
	DT-FNO	0.3403	0.1671	0.3510	**0.4150**	—	—
	SG-DT-FNO	**0.3397**	0.1684	0.3478	0.4191	—	—
	clim. (oracle)	0.3781	0.3818	0.3850	0.3728	—	—
B2 RealPDEBench	AR-FNO-R	0.7861	**0.2313**	0.6046	0.9439	4.367	1.6
	DT-FNO	0.5953	0.3046	0.6072	**0.6130**	**0.6321**	—
	SG-DT-FNO	0.6068	0.3128	0.5995	0.5992	0.6410	—
	clim. (oracle)	**0.4549**	0.4556	0.4550	0.4534	0.4538	—

1.992 ms per rollout step. At $h = 512$, this is $512\times$ fewer network evaluations and a $386.8\times$ wall-clock speedup. At $h = 1$, DT-FNO is $1.32\times$ slower because the time embedding and FiLM heads add a small fixed overhead. Timing details and the cost curve (Figure 4) are in Appendices D.6 and D.3. This reduction is shared by any direct model; the question the rest of this section addresses is where it comes with better accuracy.

Table 2 shows a consistent horizon-dependent trade-off. The autoregressive model has lower one-step error on all five datasets; Section 5.3 shows that this is a property of its four-step training and does not survive deeper rollouts. As the horizon grows, accumulated rollout error removes this advantage and DT-FNO becomes more accurate at approximately $h = 8$–20. The crossover is 9.1 on Gray–Scott, 9.6 on Cylinder, 11.5 on Rayleigh–Bénard, 19.1 on Lifted H_2, and 8.1 on RealPDEBench (Table 5). The range is relatively narrow despite a $38\times$ range in one-step autoregressive error and large differences in physical timestep. We treat this as an empirical property of the tested FNO family rather than a universal constant, and Appendix D.4 bounds it. Retaining more Fourier modes at matched size, or computing in `float32`, changes neither the divergence nor a crossover near ten. Changing the backbone does: a U-Net reaches $\text{Eval}^* = 0.295 \pm 0.019$ against 0.298 ± 0.008 for its direct counterpart on Gray–Scott, so the crossover there disappears, and 15.3 on RealPDEBench. That U-Net has 16.0M parameters against the FNO's 8.4M, but size explains neither the accuracy nor the stability: an FNO widened to 15.9M is indistinguishable from the 8.4M

Table 3: Autoregressive training depth, 100 epochs, batch 32; one seed except R=16 on Gray–Scott. Deeper rollouts improve accuracy and remove divergence but do not reach the direct model, whose supervised horizon is independent of its computational depth. $E>3$ is a blown-up rollout rather than a large error. The DT-FNO row is the three-seed result of Table 2, trained under the main protocol rather than re-run in this sweep.

		s/epoch	peak GB	best ep.	Eval*	$E_{\text{field}}(1)$	$E_{\text{field}}(32)$	$E_{\text{field}}(128)$
Gray–Scott	R=1	23	2.6	94	0.667	**0.0055**	1.206	1.4×10^4
	R=4	80	9.0	84	0.606	0.0068	0.674	0.911
	R=8	151	17.5	54	0.573	0.0115	0.585	0.844
	R=16 s0	299	34.6	19	0.525	0.0188	0.447	0.723
	R=16 s1	299	34.6	28	0.598	0.0136	0.523	709.9
	DT-FNO	—	—	—	**0.347**	0.0169	**0.368**	**0.470**
RealPDEBench	R=1	86	3.4	99	0.849	**0.212**	1.523	5.9×10^3
	R=4	111	11.8	84	0.842	0.284	1.217	289
	R=8	217	23.1	84	0.829	0.355	0.905	3.79
	R=16	436	45.8	24	0.804	0.361	0.726	1.071
	DT-FNO	—	—	—	**0.595**	0.305	**0.613**	**0.632**

The two R=16 seeds are shown separately: one diverges at no horizon and the other at six, so their mean would describe neither.

baseline (0.592 ± 0.017 against 0.593) and still diverges on two seeds of three, where the U-Net diverges on none of six runs across both datasets. The advantage of direct prediction is contingent on how good, and how stable, the rollout it is compared with happens to be.

5.2 Rollout stability and seed robustness

The difference at long horizons is not only a gradual increase in field error. AR-FNO-R diverges on Gray–Scott, Cylinder, and RealPDEBench in at least one reported long-horizon setting, whereas DT-FNO remains finite on every evaluated horizon of every dataset. For example, at $h = 128$ on Gray–Scott, the three AR seeds give $E_{\text{field}} = 0.84$, 9.0, and 113.2, and one seed becomes non-finite from $h \geq 256$. In contrast, DT-FNO has $\text{Eval}^* = 0.3468 \pm 0.0045$ on Gray–Scott and 0.4340 ± 0.0011 on Cylinder. The corresponding coefficients of variation are 1.3% and 0.25%. Qualitative examples (Figure 3) and spectral failure modes are in Appendix D.3.

These results support the main motivation for direct finite-time prediction: DTNO removes recursive feedback at inference, so an error at one queried horizon is not used as the input to the next horizon. This does not guarantee lower error at short lead times, but it prevents the specific rollout failure mode in which small local errors are repeatedly reintroduced into the model.

5.3 Autoregressive training depth

A natural objection to Table 2 is that the direct model was supervised at long horizons while the rollout was unrolled for four steps, so the gap might measure supervision rather than temporal parameterization. We retrained the autoregressive arm at fixed depths $R \in \{1, 4, 8, 16\}$, changing nothing else (Table 3); the R=4 arm reproduces AR-FNO-R to within 2% of Eval^* (0.606 against 0.593), anchoring the sweep to the headline numbers.

Deeper rollouts help on average; they do not buy reliability. Eval^* improves with R on both datasets, $0.667 \to 0.561$ on Gray–Scott (mean of two R=16 seeds) and $0.849 \to 0.804$ on RealPDEBench. Divergence is another matter: the single-seed counts fall with depth (five horizons at R=1, three at R=4, none at R=8), but the two R=16 seeds diverge at zero and six, reproducing the $0/5/6$ lottery the four-step baseline shows across its own seeds. Depth shifts the average and leaves the variance.

It costs activation memory, which is the binding constraint. Backpropagating through an R-step rollout stores activations for every step, so step time and memory grow linearly in R: 23 s per epoch and 2.6 GB at R=1 against 299 s and 34.6 GB at R=16. A microbenchmark on a fixed

batch isolates this from data loading: an autoregressive step costs 35.0 ms and 0.09 GB at $R{=}1$ and 678.5 ms and 4.22 GB at $R{=}32$, while a direct step costs 48.0 ms with identical memory at $h{=}1$ and at $h{=}128$. Its per-step memory scales to the training sweep exactly, 0.133 GB per step at batch 2 against 2.14 at batch 32, and both fits put $R{=}128$—the depth that would match the direct model's supervision—at about 270 GB, which is not trainable on one device.

And it does not close the gap. At $R{=}16$ the rollout reaches $\mathrm{Eval}^* = 0.561 \pm 0.037$ against 0.347 on Gray–Scott and 0.804 against 0.595 on RealPDEBench. The extra depth buys a trade rather than a gain: one-step error degrades monotonically with R ($0.0055 \to 0.0188$ and $0.212 \to 0.361$), so by $R{=}16$ the direct model is more accurate at every horizon tabulated for it, $h{=}1$ included, and the crossover of Section 5.1 has not moved outward but disappeared.

Caveats. Every arm but Gray–Scott $R{=}16$ is single-seed, and the two seeds there differ enough that the other divergence counts should be read as draws from a wide distribution. The direct reference is the main-protocol result, not a re-run at this batch size. The fixed epoch budget also overstates the wall-clock cost of depth, since the $R{=}16$ arms peak by epoch 28 of 100; activation memory does not depend on the budget.

5.4 Ablations: semigroup loss and lead-time embedding

The optional semigroup loss changes latent composition much more than forecast accuracy. Across the five datasets, the semigroup consistency residual decreases by roughly one to two orders of magnitude and the identity residual by two to three orders, while latent RMS does not collapse. Nevertheless, Eval^* changes by only -0.20% to $+2.47\%$ with no consistent sign. On Gray–Scott, where autonomous closed-state dynamics make the semigroup assumption especially appropriate, the accuracy change is 0.20%, smaller than the DT-FNO seed variation. We therefore keep the semigroup term optional: it improves structural consistency, but the current experiments do not support an accuracy claim (Figure 5 in Appendix D.3).

Lead-time conditioning also does not by itself guarantee generalization to unseen horizons. When trained on sparse horizon sets, Rayleigh–Bénard and Gray–Scott interpolate with mean error ratios of 1.06 and 1.24 relative to a local horizon interpolant, while Cylinder gives 3.62 and fails at all ten unseen horizons under the same criterion. A plain Fourier time embedding also has much smaller identity residual than the tested logarithmic embedding, whose $\log \tau$ parameterization is singular at $\tau = 0$. We therefore use the plain Fourier embedding in the main DTNO model. Table 6 and the corresponding curves are in Appendix D.3.

5.5 Predictability limits

One-pass stability is not forecasting skill. On four of five datasets DT-FNO approaches a climatology-like field at long horizons, as expected for a deterministic squared-error model once the history carries little information about a specific realization; on RealPDEBench the climatology oracle is outright better (0.4549 against 0.5953). Cylinder is the exception, with DT-FNO below the climatology line past $h = 100$ and $T_{\mathrm{pred}} = 70.9$ frames against 22.0. DTNO removes autoregressive cost and feedback but cannot recover information no longer in the history. Spectral and horizon-range controls are in the appendix.

6 Discussion and conclusion

We studied DTNO, a lightweight lead-time-conditioned operator that answers each temporal query in one forward pass with a small conditioning overhead on a standard FNO, against a matched autoregressive counterpart. Temporal inference cost falls from h model evaluations to one and recursive error feedback is removed—properties any direct model shares; what the study adds is where they translate into accuracy.

Across five spatiotemporal systems, the empirical picture is consistent. The rollout-trained baseline is stronger at one step, while DTNO becomes more accurate at medium horizons and is substantially more reliable at long horizons. The measured crossover lies between roughly eight and twenty steps

for the FNO family, and the direct model remains bounded throughout; with a U-Net backbone the advantage shrinks or disappears, and widening the FNO to that size reproduces neither its accuracy nor its stability, so it is the backbone rather than the capacity. Training the rollout more deeply has the same character: it improves on average, does not reach the direct model, and does not remove the divergence, while its activation memory grows linearly with depth, so the supervision the direct model gets for free is not affordable autoregressively. The climatology comparisons then show that bounded predictions are not necessarily informative once the flow has lost memory of the observed history.

The ablations show that the semigroup regularizer is not needed for accurate forecasts and that conditioning and interpolation are different properties. Direct finite-time propagation is most useful when medium- or long-horizon queries are needed repeatedly and the autoregressive alternative is unstable.

Design implications. The experiments also clarify what is gained by the particular DTNO factorization. First, temporal conditioning is cheap: FiLM heads and a time-embedding MLP totalling 151k parameters rather than a network per horizon, which is why the two models can be matched to within two percent in size. Because the encoder and decoder are pointwise, the conditioning reaches effectively the whole network; the factorization organizes the model but does not by itself save computation across queries.

Second, DTNO predicts a finite-time state rather than a local transition, which is responsible for both its main advantage and its main limitation: a query at $h = 128$ does not depend on predictions at $h < 128$, removing error feedback and giving constant temporal depth at inference *and* in training. However, independently queried states are not guaranteed to form a dynamically consistent trajectory, and the semigroup result suggests that composition constraints do not automatically help pointwise prediction. Applications that need dense, strictly consistent trajectories may prefer a continuous-time or autoregressive formulation despite its cost.

Third, the requested horizon is part of the supervised input distribution: the Cylinder ablation shows that a time embedding does not by itself guarantee interpolation, so training should cover the range that will be queried, which the sampler of Section 3.4 does at no extra cost per pair.

Limitations. SG-DT-FNO is single-seed on three datasets, Rayleigh–Bénard is also single-seed in the main comparison, Cylinder is quasi-periodic rather than chaotic, and the temporal-cadence intervention could not be completed on the available record. The evidence is limited to the tested backbones and resolutions. We do not include a fine-tuned Poseidon baseline: it conditions on a single state, is pretrained at a scale our per-dataset protocol does not match, and targets initial-value operator learning rather than history-conditioned forecasting. These limitations are detailed in Appendices D.1, D.7, and D.5.

Reproducibility statement

Trajectory splits are frozen before window construction, and normalization statistics are fitted only on the training split. Each results file records the training configuration, and non-finite autoregressive horizons are counted and reported rather than silently removed. Three seeds are used for the headline AR-FNO-R and DT-FNO comparisons on Gray–Scott, Cylinder, RealPDEBench, and Lifted H_2; single-seed results are marked explicitly. Additional implementation details, split rules, hyperparameters, and the anonymous code repository link are provided in the appendix.

AI use statement

Generative AI tools were used to assist with drafting and editing parts of the manuscript, including English-language revision and structural reorganization. They were not used to generate experimental data or reported numerical results. The authors reviewed the AI-assisted text, checked technical claims and references against primary sources where available, and take responsibility for the final manuscript.

REFERENCES

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

Wai Tong Chung, Ki Sung Jung, Jacqueline H. Chen, and Matthias Ihme. BLASTNet: A call for community-involved big data in combustion machine learning. *Applications in Energy and Combustion Science*, 12:100087, 2022. doi: 10.1016/j.jaecs.2022.100087.

Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for PDEs. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2311.

Xianglong Hou, Xinquan Huang, and Paris Perdikaris. CFO: Learning continuous-time PDE dynamics via flow-matched neural operators. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=IQhaeSzyup.

Peiyan Hu, Haodong Feng, Hongyuan Liu, Tongtong Yan, Wenhao Deng, Tianrun Gao, Rong Zheng, Haoren Zheng, Chenglei Yu, Chuanrui Wang, Kaiwen Li, Zhi-Ming Ma, Dezhi Zhou, Xingcai Lu, Dixia Fan, and Tailin Wu. RealPDEBench: A benchmark for complex physical systems with real-world data. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=y3oHMcoItR.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to PDEs. *Journal of Machine Learning Research*, 24(89):1–97, 2023. URL https://jmlr.org/papers/v24/21-1524.html.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=c8P9NQVtmnO.

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. doi: 10.52202/075280-2946.

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

Runze Mao, Rui Zhang, Xuan Bai, Tianhao Wu, Teng Zhang, Zhenyi Chen, Minqi Lin, Bocheng Zeng, Yangchen Xu, Yingxuan Xiang, Haoze Zhang, Shubham Goswami, Pierre A. Dawe, Yifan Xu, Zhenhua An, Mengtao Yan, Xiaoyi Lu, Yi Wang, Rongbo Bai, Haobu Gao, Xiaohang Fang, Han Li, Hao Sun, and Zhi X. Chen. Benchmarking neural surrogates on realistic spatiotemporal multiphysics flows. *arXiv preprint arXiv:2512.18595*, 2025. doi: 10.48550/arXiv.2512.18595.

Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina J. Agocs, Miguel Beneitez, Marsha Berger, Blakesley Burkhart, Stuart B. Dalziel, Drummond B. Fielding, Daniel Fortunato, Jared A. Goldberg, Keiya Hirashima, Yan-Fei Jiang, Rich R. Kerswell, Suryanarayana Maddu, Jonah Miller, Payel Mukhopadhyay, Stefan S. Nixon, Jeff Shen, Romain Watteaux, Bruno Régaldo-Saint Blancard, François Rozet, Liam H. Parker, Miles Cranmer, and Shirley Ho. The well: a large-scale collection of diverse physics simulations for machine learning. In *Advances in Neural Information Processing Systems*, volume 37 of *Datasets and Benchmarks Track*, pp. 44989–45037, 2024. doi: 10.52202/079017-1430. arXiv:2412.00568.

Vivek Oommen, Aniruddha Bora, Zhen Zhang, and George Em Karniadakis. Integrating neural operators with diffusion models improves spectral representation in turbulence modelling. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 481(2309): 20240819, 2025. doi: 10.1098/rspa.2024.0819.

Oded Ovadia, Vivek Oommen, Adar Kahana, Ahmad Peyvan, Eli Turkel, and George Em Karniadakis. Real-time inference and extrapolation with time-conditioned UNet: Applications in hypersonic flows, incompressible flows, and global temperature forecasting. *Computer Methods in Applied Mechanics and Engineering*, 441:117982, 2025. doi: 10.1016/j.cma.2025.117982.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pp. 3942–3951, 2018. doi: 10.1609/AAAI.V32I1.11671.

Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R. Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, Remi Lam, and Matthew Willson. Probabilistic weather forecasting with machine learning. *Nature*, 637(8044): 84–90, 2025. doi: 10.1038/s41586-024-08252-9.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8459–8468. PMLR, 2020. URL `https://proceedings.mlr.press/v119/sanchez-gonzalez20a.html`.

Peter J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010. doi: 10.1017/S0022112010001217.

Pushan Sharma, Wai Tong Chung, and Matthias Ihme. A physics-informed machine learning approach for predicting dynamic behavior of reacting flows with application to hydrogen jet flames. *Combustion and Flame*, 277:114190, 2025. doi: 10.1016/j.combustflame.2025.114190.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. PDEBench: An extensive benchmark for scientific machine learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-0117.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 32–42, 2021. doi: 10.1109/ICCV48922.2021.00010.

A Conditional spectral propagation and time conditioning

The propagator is a stack of L residual Fourier-operator blocks (Li et al., 2021). For block ℓ, the spatial update is

$$s_\ell = \mathcal{F}^{-1}(R_\ell \odot \mathcal{F}(z_\ell)) + W_\ell z_\ell, \qquad z_{\ell+1} = z_\ell + \alpha_\ell \, \mathcal{A}_\ell(\mathrm{FiLM}_\ell(s_\ell; \tau)), \tag{4}$$

where R_ℓ contains learned weights on retained Fourier modes, W_ℓ is the local branch, α_ℓ is a LayerScale coefficient (Touvron et al., 2021), and $\mathcal{A}_\ell$ denotes the block's activation operations. Time changes the modulation of latent features, not the Fourier basis itself. Specifically,

$$(\gamma_\ell(\tau), \beta_\ell(\tau)) = M_\ell(e(\tau)), \qquad \mathrm{FiLM}_\ell(s; \tau) = \gamma_\ell(\tau) \odot s + \beta_\ell(\tau), \tag{5}$$

so the same horizon representation conditions every propagation depth (Perez et al., 2018). This yields a family $\{\Phi_\theta(\cdot, \tau)\}_\tau$ of finite-time latent maps sharing one spatial backbone, rather than a set of separate horizon-specific networks.

We choose T_{scale} so the largest training horizon has $\tau = 1$. The default embedding contains the scalar horizon and Fourier features, $e_F(\tau) = [\tau, \sin(2^j \pi \tau), \cos(2^j \pi \tau)]_{j \in \mathcal{J}}$, followed by a small time MLP and block-specific FiLM heads. The sparse-horizon ablation replaces this periodic representation with a log-scaled construction while leaving the spatial model unchanged; Section 5.4 shows that the embedding controls interpolation between supervised lead times. Residual propagation and the residual decoder also make near-identity behavior accessible at short lead times, which is consistent with the structure of a finite-time flow map.

B Dataset preprocessing

Gray–Scott and Rayleigh–Bénard come from The Well (Ohana et al., 2024), Cylinder and the combustion set from RealPDEBench (Hu et al., 2026), and Lifted H_2 from the BLASTNet lifted-flame family (Chung et al., 2022; Sharma et al., 2025), using the eight reacting Reynolds-number cases and excluding the two inert-mixing cases; the original 1600×2000 fields are used in their 160×200 reduced representation. Gray–Scott has periodic boundaries on a uniform grid and therefore uses zero padding, where every other dataset uses replicate padding. Rayleigh–Bénard is stored at 512×128 and subsampled to 128×128 by taking every fourth point along the periodic horizontal axis, which preserves periodicity exactly. Cylinder is used at its native 128×256. All channels are standardized with training-set statistics only. Resampling the RealPDEBench combustion fields onto the uniform grid produces Gibbs undershoot, mole fractions reach -6.0×10^{-2} and temperature -94 K, so we avoid logarithmic or Box–Cox transforms despite the large raw dynamic ranges.

Channel selection. We retain 13 of the 15 released RealPDEBench numerical channels: `Absolute_Pressure`, `Chemistry_Heat_Release_Rate`, the mole fractions of CH_4, CO, CO_2, H_2O, NH_2, NH_3 and OH, `Temperature`, and the three velocity components. Two are excluded because an automated redundancy check found them to be functions of channels already in the set. `Pressure` equals `Absolute_Pressure` $-$ 101325 with a measured correlation of 1.000000, and standardization erases the offset, so after normalization the two are literally the same field. `Velocity_Magnitude` agrees with $\sqrt{u^2+v^2+w^2}$ to 2.45% relative error, not an exact identity, and the residual is plausibly a resampling artefact since $|\mathrm{resample}(u)| \neq \mathrm{resample}(|u|)$, but 97.5% of it is reconstructible from channels already present, and retaining it would give velocity four of fourteen slots in the channel-averaged E_{field} while diluting its group in the per-variable breakdown. A duplicated quantity is not merely wasteful: it receives double weight in the loss *and* in the channel-averaged error, so a model that happens to fit it well is rewarded twice for one capability. Cylinder is treated the same way: the released `vo` (vorticity) channel is a function of u and v and is excluded, leaving u, v and p.

The Rayleigh–Bénard vertical grid is not uniform. The Well releases this dataset sampled at Chebyshev nodes along the wall-normal axis, alongside a resampled `rayleigh_benard_uniform` variant. We use the Chebyshev version. An FNO's FFT and our spectral error both assume uniform spacing, so E_{field} remains exact, it is a pointwise error on the points the data actually has, but the *vertical* wavenumber axis of a spectrum computed on this grid is not a physical wavenumber axis. We therefore make no spectral claim for Rayleigh–Bénard; the spectral results in this paper (Figure 8, Table 8) are for Lifted H_2 and RealPDEBench, both on uniform grids. A study needing the vertical spectrum should use the uniform variant. We did not resample, because an interpolation inserted beneath the normalization statistics would change every number reported here while leaving no trace in the results files.

C Extended related work

Neural operators, PDE surrogates and benchmarks. DeepONet (Lu et al., 2021) and neural operators (Li et al., 2021; Kovachki et al., 2023) learn maps between function spaces and are standard surrogates for parametrized PDEs. PDEBench (Takamoto et al., 2022) systematized comparison across canonical families, The Well (Ohana et al., 2024) extends this to sixteen spatiotemporal systems, and more recent datasets move toward realistic multiphysics regimes (Hu et al., 2026; Chung et al., 2022; Mao et al., 2025). We draw from three of these and select for *mechanism* rather than coverage: reaction without advection, advection without chemistry, buoyant transport, and two reacting flows. The resulting set spans chaotic decorrelation, a quasi-periodic limit cycle and a quasi-periodic wake, with predictability horizons differing by four orders of magnitude in physical time.

Temporal prediction and rollout stability. Autoregressive simulators reuse a local transition model at arbitrary rollout length, but their errors become model inputs. Noise injection (Sanchez-Gonzalez et al., 2020), the pushforward trick and temporal bundling (Brandstetter et al., 2022), and iterative refinement (Lippe et al., 2023) are representative stabilizers, and our AR-FNO-R baseline

follows the same principle by training on four-step rollouts of its own predictions. Our goal is not another stabilizer but a comparison against a predictor whose inference depth does not grow with the horizon.

Time-conditioned and continuous-time operators. Several operator formulations already support non-autoregressive temporal queries. DeepONet can place time in its trunk coordinates (Lu et al., 2021); Poseidon uses lead-time-conditioned normalization to learn continuous-in-time solution operators and exploits the semigroup property during training (Herde et al., 2024); and TC-UNet conditions a U-Net on time for arbitrary temporal-resolution inference (Ovadia et al., 2025). CFO instead learns continuous-time PDE dynamics and integrates a learned vector field at inference (Hou et al., 2026). Poseidon describes both direct evaluation and autoregressive composition of the same trained model; neither involves a separately rollout-trained autoregressive model. It also distinguishes its initial-value operator-learning task from next-step prediction given a context window, the setting of MPP and DPOT; ours is closer to the latter, forecasting from a K-frame history window. Our contribution is complementary: we hold the spatial backbone fixed, compare against a rollout-trained model, and evaluate the full error-versus-horizon curve, its crossover, climatology-relative predictability, the effect of backbone, precision and Fourier bandwidth on the comparison, semigroup regularization as a controlled loss ablation, and interpolation between sparsely supervised horizons.

Long-horizon uncertainty and generative forecasting. Once multiple futures are plausible, deterministic squared-error predictors tend toward conditional means and lose small-scale energy. GenCast shows that conditional generative models retain sharp samples at long lead times (Price et al., 2025), and diffusion correction improves spectral fidelity for turbulent-flow operators (Oommen et al., 2025). Both motivate probabilistic extensions of our operator and highlight the trade-off: iterative sampling surrenders part of the one-pass cost advantage.

D Supplementary material

D.1 Additional experimental checks

The following are not run in this paper. They are ordered by what they would change, and we list them because each bears on a claim we do make.

Would change a claim. The *lead-time mechanism* is the largest gap: two runs would settle whether quasi-periodicity explains Cylinder's failure to interpolate—retraining it with horizons spaced below half its 174.5-frame shedding period, and running the sparse-τ arm on an advective but aperiodic dataset. If either contradicts the prediction, the explanation in Section 5.4 becomes "dataset-dependent, cause unknown". An external *Poseidon/scOT* baseline would place the matched results against the strongest pretrained time-conditioned model; it would not be a matched comparison (single-state input, large-scale pretraining) and is not included. A *temporal-cadence ablation* by striding RealPDEBench is the one direct intervention on the crossover of Section 5.1: if striding moves the crossover in frames but not in seconds, it is set by physical time rather than by composition. It was attempted and could not be completed on this record (Appendix D.5).

Would strengthen a conservative claim. SG-DT-FNO has three seeds on Lifted H_2 and RealPDEBench and one on the three Experiment A datasets, so on those three the semigroup null is stated against DT-FNO's measured spread rather than its own; two more Gray–Scott seeds would close that. A POD-DMD *rank sweep*, with the rank-selection rule fixed before test evaluation, would report the classical baseline at its ceiling rather than at rank 256.

Completeness. Physics-facing metrics (reaction-zone IoU, gradient error, integrated heat release) are reported for RealPDEBench in Table 4 where available; extending them to Lifted H_2 across all horizons would complete the picture. Qualitative fields for Gray–Scott and Rayleigh–Bénard would complete the set begun by Figures 7, 10 and 3; the reaction-zone overlay does not apply to non-reacting flows and is omitted rather than substituted.

D.2 Reproducibility details

Implementation and Reproducibility. The source code, training scripts, and experimental configurations to reproduce our results are available at the anonymous repository: `https://anonymous.4open.science/r/DTNO-6236`

Seeds. Three seeds are used wherever a comparison in the main text depends on the spread; single-seed entries are marked † in Table 2 and listed here so the count does not have to be assembled from the text.

dataset	AR-FNO-R	DT-FNO	SG-DT-FNO
Gray–Scott	3	3	1
Cylinder	3	3	1
Rayleigh–Bénard	1	1	1
Lifted H_2	3	3	3
RealPDEBench	3	3	3

The baselines (persistence, both climatologies, POD-DMD) are deterministic given the split and are run once. The sparse-horizon and time-embedding ablations of Section 5.4 are single-seed.

Shared hyperparameters. All five datasets use the same architecture and optimizer: FNO width 64, four blocks, $K = 4$ history frames, LayerScale 0.1, GELU, AdamW at lr $= 10^{-3}$ with 500 warm-up steps and cosine decay, batch 64 per GPU on four GPUs, 20,000 online pairs per epoch, and 100 epochs (120 on Rayleigh–Bénard). Four settings differ, each for a stated reason: the mode counts follow the grid (16×16 on the three 128×128 datasets, 16×32 on Cylinder's 128×256, 16×20 on Lifted H_2's 160×200, all at one quarter of Nyquist on both axes); the FNO padding is zero on the periodic Gray–Scott domain and replicate (8) elsewhere; $h_{\max}$ is 128 except on Rayleigh–Bénard, whose 200-frame records cap it at 64; and the evaluation grid is truncated to match. Channel transforms are z-score throughout—the resampling undershoot documented above rules out logarithmic or Box–Cox transforms on RealPDEBench, and no channel of the other four datasets spans decades. Training horizons are sampled as described in Section 3.4; every integer horizon up to $h_{\max}$ can be drawn, horizons beyond it are evaluated but never supervised, and the sparse-horizon arm of Section 5.4 is the deliberate exception. The pool of available pairs is itself not uniform in h: a trajectory of length T with K history frames admits about $T - K - h$ anchors at horizon h, so the longest horizons are the scarcest. On Lifted H_2 ($T = 151$) this falls from about 147 anchors at h=1 to about 20 at h=128, which is why its evaluation grid stops short of the other datasets'.

Parameter counts. The three arms differ by exactly 150,912 parameters regardless of dataset—four FiLM heads ($4 \times 33{,}024$) and the time-embedding MLP (18,816)—because neither scales with the channel count, and the 1×1 lift and projection together stay under 15k. Dataset to dataset the count is set by the retained modes: the spectral weights dominate at $2w^2 m_1 m_2$ real parameters per block. The values below are the ones the training logs report, and a formula built from the block structure reproduces all five arms measured here (Gray–Scott AR and DT, the width-88 arm, and both mode-count arms) to the parameter.

dataset	channels	modes	AR-FNO-R	DT-FNO / SG-DT-FNO
Gray–Scott	2	16×16	8.41 M	8.57 M
Cylinder	3	16×32	16.80 M	16.95 M
Rayleigh–Bénard	4	16×16	8.42 M	8.57 M
Lifted H_2	5	16×20	10.51 M	10.66 M
RealPDEBench	13	16×16	8.42 M	8.57 M

What is and is not reported. Per-group errors are produced for every dataset by the same script (Figure 12 shows RealPDEBench; the other four are in the code release). Seed-level results are given wherever more than one seed exists—AR-FNO-R and DT-FNO on all five datasets, SG-DT-FNO on Lifted H_2 and RealPDEBench—and the tables quote per-seed values wherever the spread straddles

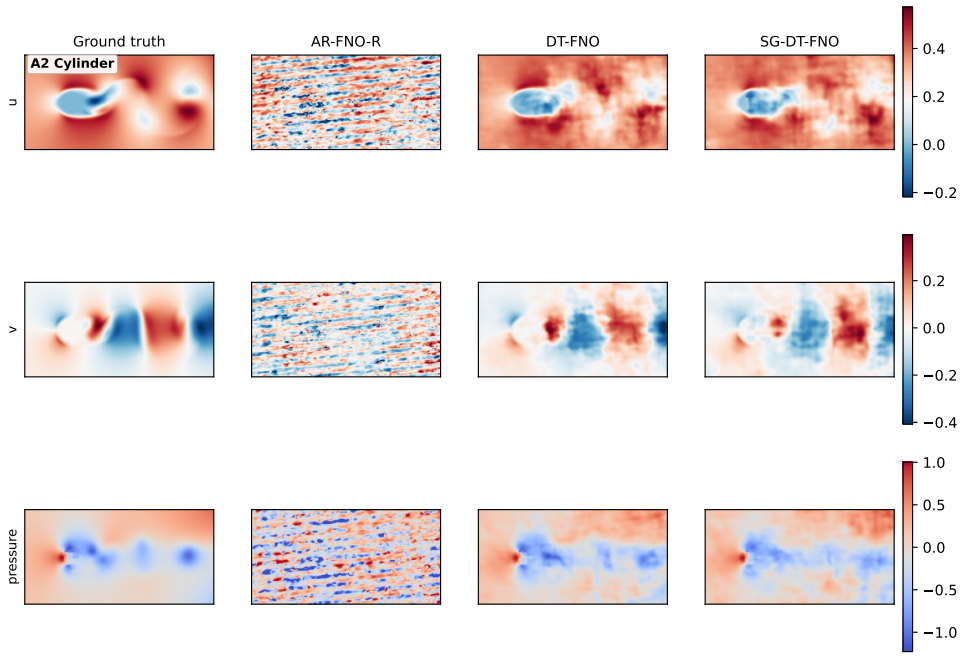

Figure 3: Cylinder prediction at $h = 128$ on a held-out trajectory. AR-FNO-R has developed high-frequency artifacts, while DT-FNO preserves the recirculation region and vortex train. The direct prediction is smoother than the reference, consistent with conditional-mean behavior at long horizons.

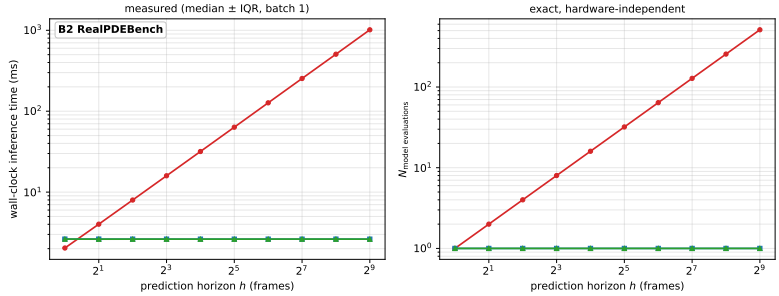

Figure 4: Inference cost versus horizon. Left: measured wall-clock (median ± IQR, batch 1, idle GPU). Right: the exact number of model evaluations, which is hardware-independent. Timing depends on the architecture, not on the autoregressive training variant, so a single AR curve is shown.

divergence, rather than a mean that describes no seed. Physics-facing metrics are in Table 4 where available.

Lifted-flame spatial reduction. The BLASTNet lifted-flame fields are released on a 1600×2000 grid; we use the 160×200 reduced representation distributed with the dataset, a $10\times$ decimation in each direction. This is a decimation rather than a filtered downsampling, so the reduced fields alias structure above the reduced Nyquist wavenumber, and the spectral results for this dataset (Figure 8) are statements about the resolved band of the reduced fields rather than about the original DNS. The same distinction applies to Cylinder, where the released benchmark protocol decimates by a further factor of two and we do not.

D.3 Additional figures

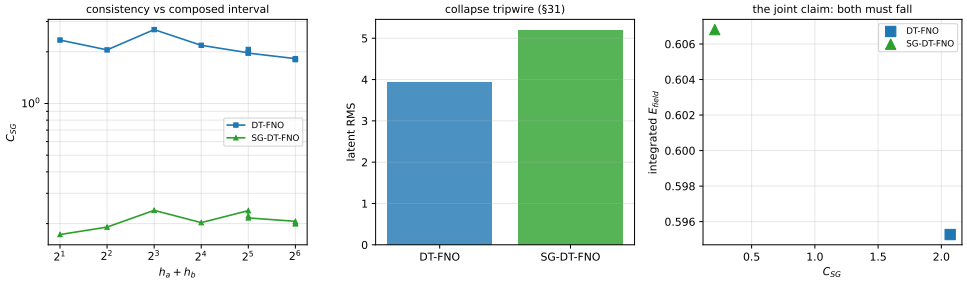

Figure 5: Semigroup consistency must be read jointly: C_{SG} falls by an order of magnitude (left) while the latent RMS *rises* (centre), ruling out representational collapse—and the integrated field error does not move (right).

Table 4: Physics-facing metrics on RealPDEBench (`scripts/physics_metrics_table.py`): reaction-zone IoU (higher is better), gradient error and integrated heat-release error (lower is better). The autoregressive model loses the flame position at long horizons while the direct model retains it; the trajectory climatology remains best, consistent with Section 5.5.

[place tab_physics.tex next to main.tex]

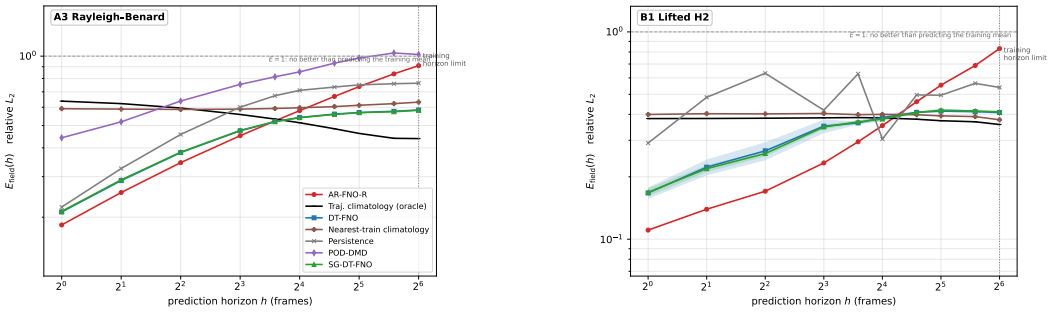

Figure 6: Field error versus horizon on the two datasets omitted from Figure 2 for space: A3 Rayleigh–Bénard (left) and B1 Lifted H_2 (right).

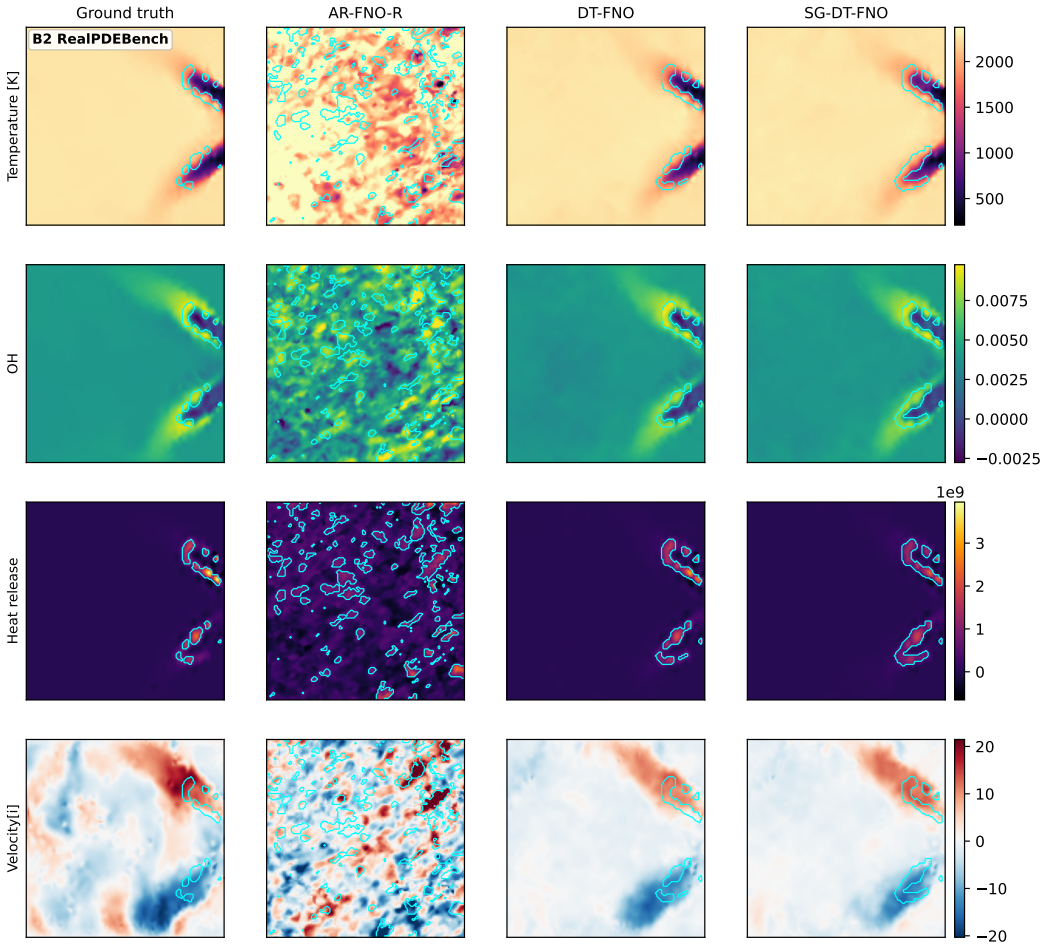

Figure 7: Fields at h=128 (32 ms) on a held-out RealPDEBench trajectory: ground truth against AR-FNO-R, DT-FNO and SG-DT-FNO. The cyan contour is the reaction zone at α=0.2 of peak heat release. The autoregressive rollout has become high-frequency noise; both direct models still place the two reaction zones, the OH streak and the velocity structure.

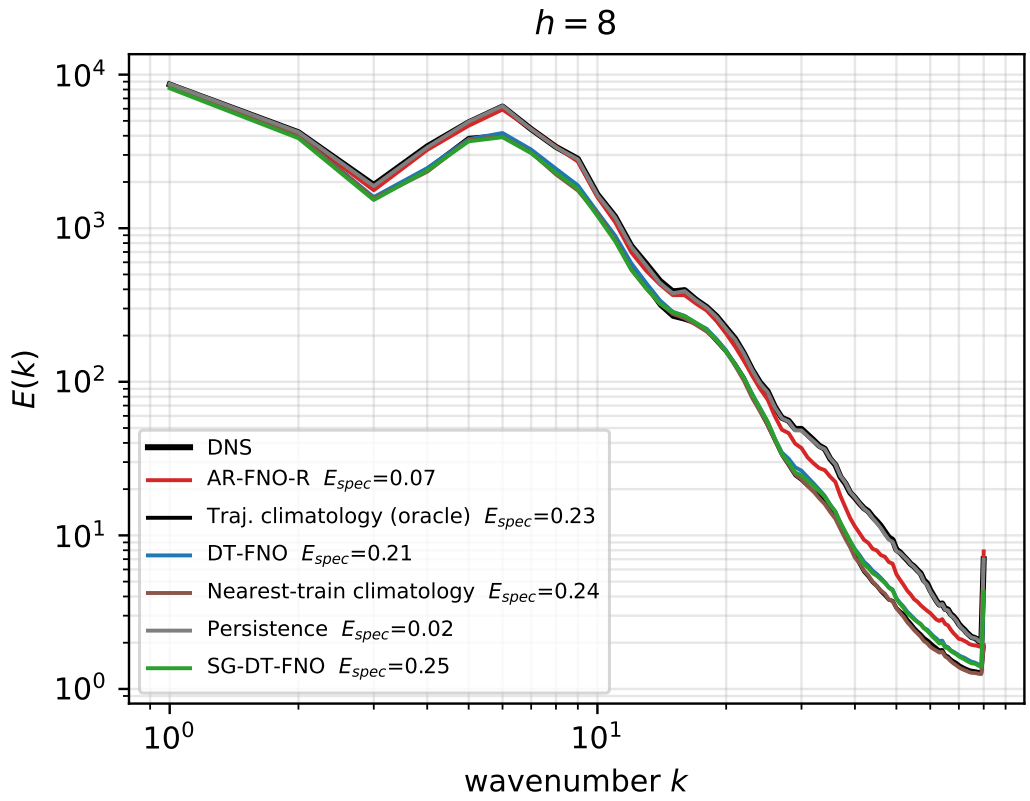

Figure 8: Radial energy spectra on Lifted H_2 at $h = 8, 32, 64$. The direct operator's spectrum coincides with the climatology's at every horizon; the autoregressive model is far better at short lead time and injects energy above DNS by h=64.

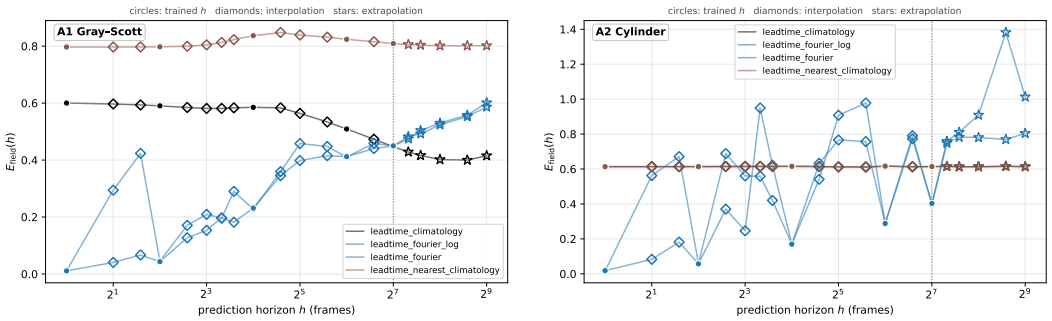

Figure 9: Lead-time generalization from the sparse training set $h \in \{1, 4, 16, 64, 128\}$ on Gray–Scott (left) and Cylinder (right). Trained horizons are marked; everything between them is unseen. Gray–Scott traces a smooth curve through the unseen horizons; Cylinder returns to the trained anchors and rises between them. Reproduced from the sparse-horizon runs of Section 5.4.

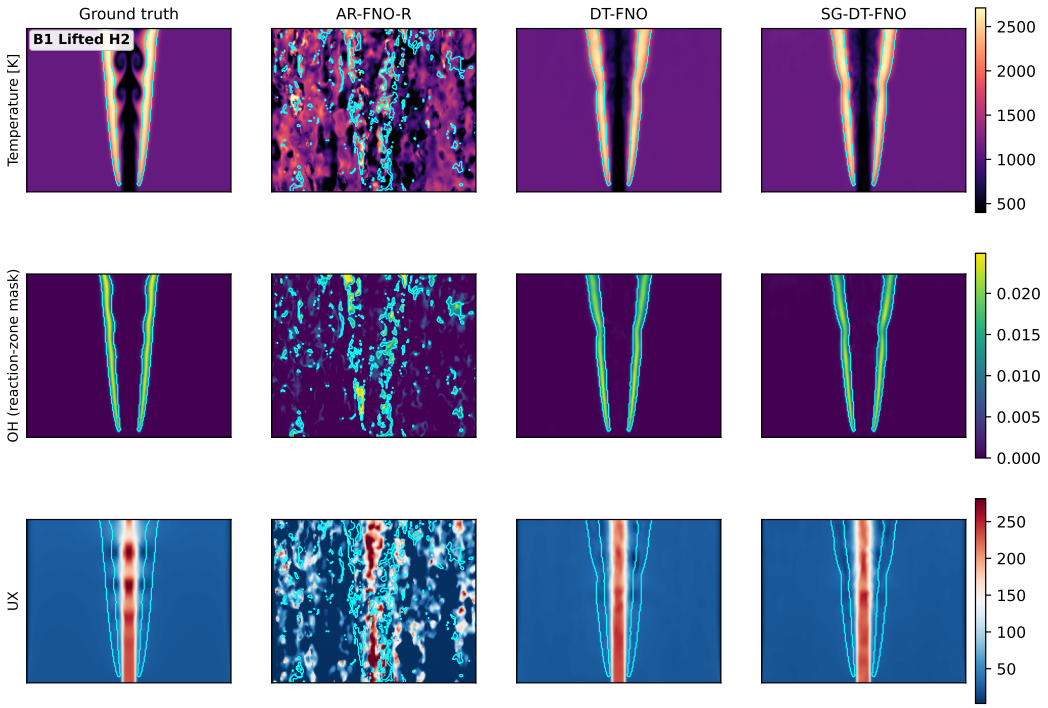

Figure 10: Fields at h=128 (0.64 ms) on a held-out BLASTNet's Lifted H_2 trajectory: ground truth against AR-FNO-R, DT-FNO and SG-DT-FNO.

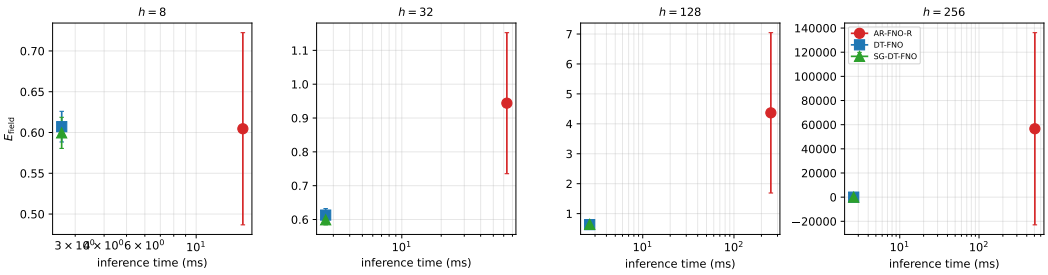

Figure 11: Accuracy against measured inference cost at four horizons. The claim is not that the direct operator is uniformly more accurate—it is not—but that it occupies a different point on the trade-off, and that the autoregressive model leaves the plot entirely once its rollout diverges.

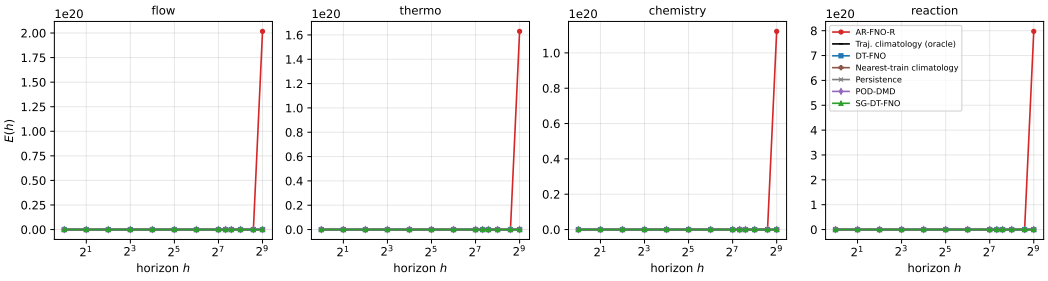

Figure 12: Per-variable error on RealPDEBench. A channel-averaged score can hide a model that tracks velocity while misplacing the reaction zone; we therefore retain grouped errors in addition to the aggregate metric.

D.4 Architecture and bandwidth controls

	backbone / modes	params	AR diverged	crossover
Gray–Scott	FNO 16×16 (quarter Nyquist)	8.41 M	$0/5/6$	9.1
	FNO 32×32 (half)	8.40 M	4	10.5
	FNO 64×65 (no truncation)	8.52 M	5	8.2
	FNO 16×16, width 88	15.90 M	$0/3/2$	—
	U-Net	16.01 M	$0, 0, 0$	none
RealPDEBench	FNO 16×16	8.42 M	4	8.1
	U-Net	16.01 M	$0, 0, 0$	15.3

Three seeds for every U-Net entry, for the Gray–Scott FNO baseline and for the width-88 arm; the mode-count arms are single-seed, so their divergence counts should be read against the baseline's $0/5/6$ spread rather than as values. Parameter counts are read from the training logs; the direct counterparts add 150,912 parameters for the FiLM heads and time embedding in the FNO arms, and about 0.59M in the U-Net arms, where the conditioned channel widths are larger.

Bandwidth and capacity. The three mode-count arms hold the parameter count fixed to within 1.3% by reducing the width as the mode count rises ($64 \to 32 \to 16$), separating truncation from capacity. In the untruncated arm `SpectralConv2d` resolves to m_1=64, m_2=65 on a 128×128 grid and its two writes tile all 128 rows exactly once, discarding nothing, and it diverges slightly *more* than the baseline. We proposed spectral truncation as the mechanism behind the architecture gap and report it as ruled out. The width-88 arm instead holds bandwidth fixed and raises the parameter count to the U-Net's 16M. Over three seeds it is indistinguishable from the 8.4M baseline ($\mathrm{Eval}^* = 0.592 \pm 0.017$ against 0.593) and diverges at 0, 3 and 2 horizons against the baseline's 0, 5 and 6, while the U-Net diverges at none in six runs across two datasets. Doubling the parameter count therefore buys the FNO neither accuracy nor stability: both the accuracy gap to the U-Net and its freedom from divergence are properties of the backbone.

"None" in the Gray–Scott U-Net row means the two error curves lie within each other's seed spread over the whole evaluated range, so the horizon at which they cross is not a meaningful quantity; a single seed puts it at 194, but that is where two coincident curves happen to touch. The autoregressive and direct U-Nets there reach Eval^* of 0.2952 ± 0.0191 and 0.2980 ± 0.0077.

Precision. The U-Net arms compute in `float32`. Under `bfloat16` they produce a non-finite loss within the first dozen epochs, at a different epoch for each seed, which the FNO arms never do; the fix is an explicit upcast inside the backbone, matching what `SpectralConv2d` already did. A control confirms this is not what drives the comparison: running the Gray–Scott FNO arms in `float32` leaves the autoregressive divergence in place (1 horizon, inside the 0–6 seed range) and moves the crossover from 9.1 to 11.4. No wall-clock comparison between the two backbones is reported, since their precisions differ.

D.5 Temporal cadence: attempted, not obtained

The intervention that would turn Section 5.1 from a cross-dataset correlation into a within-dataset causal test is to stride RealPDEBench in time, holding flow, architecture, protocol and split fixed. We attempted it and report that it is not feasible on this record. Striding shortens the usable trajectory while $h_{\max}$ is held fixed in *frames*, so the evaluation anchors collapse: 150 at stride 1, 55 at stride 2, 5 at stride 4. Five anchors cannot support a crossover estimate, and lowering $h_{\max}$ with the stride would change the horizon grid, which is the one thing the experiment must hold fixed. A record several times longer, or a flow whose predictability horizon is a larger fraction of its length, would be needed. We flag this as the most important open check rather than reporting the numbers the truncated sweep produced.

D.6 Additional quantitative details

Semigroup, per dataset. C_{SG} falls $4.34 \to 0.12$ (Gray–Scott), $4.21 \to 0.09$ (Cylinder) and $3.31 \to 0.08$ (Rayleigh–Bénard), a factor of 36–46; on Lifted H_2 the identity residual falls from

Table 5: Observed accuracy crossover between AR-FNO-R and DT-FNO. The narrow range in step count is empirical for this FNO family; it is not assumed to be universal across architectures.

Dataset	channels	AR $E_{\mathrm{field}}(1)$	crossover h	Δt
Gray–Scott	2	0.0062	9.1	10 s
Cylinder	3	0.0060	9.6	2.5 ms
Rayleigh–Bénard	4	0.1852	11.5	0.25 s
Lifted H_2	5	0.111	19.1	5 μs
RealPDEBench	13	0.231	8.1	250 μs

Table 6: Lead-time generalization from sparse training horizons. Ratio of E_{field} at unseen horizons to the interpolant of the trained neighbours, over the unseen horizons below $h_{\max}$. Trained set $\{1, 4, 16, 64, 128\}$ on Gray–Scott and Cylinder, $\{1, 4, 16, 48, 64\}$ on Rayleigh–Bénard.

dataset	embedding	mean ratio	max	$> 2\times$
Rayleigh–Bénard	Fourier	**1.06**	1.10	0/8
Gray–Scott	Fourier	**1.24**	1.83	0/10
Gray–Scott	log-Fourier	3.24	11.03	2/10
Cylinder	Fourier	3.62	7.22	10/10
Cylinder	log-Fourier	5.90	14.74	10/10

10.8–27.7 to 0.0026. Eval* changes by -0.20%, $+2.47\%$, -0.06%, -0.18% and $+1.93\%$ on Gray–Scott, Cylinder, Rayleigh–Bénard, Lifted H_2 and RealPDEBench respectively. Three of the five compare three seeds against three (Table D.2); on Rayleigh–Bénard the two models agree to three decimals at every horizon.

Horizon-range ablation. This ablation argues against a capacity-allocation explanation. Training with $h_{\max} \in \{16, 32, 64, 128\}$ leaves the in-range error within 4% ($E(8) = 0.616$–0.642) and the ordering runs the wrong way, with larger training horizons better overall (Eval* $0.771/0.761/0.692/0.630$); the same sweep shows no extrapolation past the training range ($h_{\max}$=16 reaches $E = 3.51$ at h=128). The training horizon range *defines* the usable range and costs nothing to widen. Widening it does dilute the short end, because the number of surviving bins grows with $h_{\max}$: the $[1, 2]$ bin receives a quarter of the pairs at $h_{\max}$=16 and a seventh at $h_{\max}$=128, a 43% relative cut in its share of a fixed budget. Short-horizon accuracy is nonetheless insensitive to it, as the near-constant $E(8)$ above shows.

D.7 Additional limitations

Seed depth. SG-DT-FNO is single-seed on Gray–Scott, Cylinder and Rayleigh–Bénard, so on those three the semigroup null is stated against DT-FNO's measured three-seed spread rather than its own interval. On Lifted H_2 and RealPDEBench all three arms have three seeds.

Cylinder is quasi-periodic, not chaotic. Its long predictable window follows from the periodicity of vortex shedding and should not be read as evidence that direct-time prediction extends the predictable window of a chaotic flow. A chaotic flow sampled finely enough that h=32–128 lies inside its predictable window is still missing from this study; Cylinder is the closest available proxy.

Backbones. The headline results use an FNO backbone, and the U-Net control of Appendix D.4 shows that the crossover is *not* architecture-insensitive: it moves on RealPDEBench and disappears on Gray–Scott. Claims about the crossover are therefore restricted to the FNO family. The U-Net carries 16.0M parameters against the FNO's 8.4M, but widening the FNO to 15.9M reproduces neither its accuracy nor its freedom from divergence over three seeds, so both are architectural. Spectral truncation and numerical precision are also ruled out. What is *not* explained is why the U-Net rollout is stable.

Lifted H_2 has only one held-out reacting-flow trajectory at $Re_{\mathrm{jet}} = 7500$, so its curves do not estimate variability across independent flow realizations. The RealPDEBench case decorrelates

Table 7: Inference cost on an idle A800 (batch 1, fp32, 3 s sustained GPU warm-up, median of 30 repeats, model loading excluded). AR's per-step cost is 1.992 ms with 3.1% spread over a $512\times$ range; the direct operator is flat to within 0.01 ms.

h	AR-FNO-R	DT-FNO	speedup	N_{eval} ratio
1	2.04 ms	2.62 ms	$0.8\times$	$1\times$
32	63.44	2.62	$24.2\times$	$32\times$
128	253.74	2.62	$96.8\times$	$128\times$
512	1014.06	2.62	$386.8\times$	$512\times$

Table 8: Spectral error E_{spec} on Lifted H_2.

model	h=8	h=32	h=64
AR-FNO-R	**0.07**	0.28	0.48
DT-FNO	0.21	0.23	0.24
climatology	0.23	0.23	0.23

chaotically, whereas Lifted H_2 is quasi-periodic; their predictability horizons should therefore not be interpreted as the same physical timescale. Absolute pressure is effectively unpredictable at the available 4 kHz cadence for all tested methods (persistence 1.93, POD-DMD 1.91), and its DT-FNO spectral error can dominate channel-averaged spectral summaries. Finally, RealPDEBench uses a heat-release reaction-zone mask while Lifted H_2 uses OH because no heat-release channel is available; these diagnostics are related but not interchangeable.

Table 9: The constraint is satisfied by one to two orders of magnitude with the latent RMS *rising*, so the improvement is not representational collapse, and forecast accuracy is unchanged on both datasets.

dataset	model	C_{SG}	latent RMS	Eval*
RealPDEBench	DT-FNO	2.05	3.95	0.6106
	SG-DT-FNO	**0.21**	5.19	0.6084
Lifted H_2	DT-FNO	3.1–11.2	2.0–2.5	0.3403
	SG-DT-FNO	**0.11–0.18**	10.1–13.0	0.3403

Table 10: Training on sparse anchors $\{1, 4, 16, 64, 128\}$ and evaluating at unseen horizons. Curve roughness (mean absolute second difference) is 0.228 versus 0.067; the worst unseen horizon is 0.939 versus 0.626.

h	trained	Fourier	log-Fourier
1	yes	0.252	0.247
2	no	**0.895**	0.574
3	no	**0.888**	0.547
4	yes	0.554	0.549
10	no	**0.939**	0.603