PRETRAINING DESIGN DRIVES GAINS IN ZERO-SHOT PDE ROLLOUTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Generalization beyond the pretraining distribution remains a major challenge for pretrained neural PDE solvers. We study zero-shot out-of-distribution (OOD) rollout prediction in fluid dynamics, where a pretrained model must predict future trajectories from a single initial condition without downstream finetuning, contextual demonstrations, or test-time adaptation. We uncover a surprisingly large performance gap: a lightweight ViT-style spatiotemporal model with only about 25M parameters outperforms the 629M-parameter pretrained Poseidon-L on several OOD systems, with approximately 3–4$\times$ lower error on two of them, despite its weaker in-distribution accuracy. We examine how pretraining design affects zero-shot transfer through controlled comparisons of autoregressive rollout, teacher forcing, non-autoregressive full-trajectory prediction, and different temporal conditioning schemes. We find that training formulation can change OOD errors by large factors. With full-history attention, teacher forcing improves our ViT's source-domain accuracy and yields better overall source-domain and OOD performance than AR rollout training, despite the training-inference mismatch. Comparisons with FNO reveal similar benefits in source-domain prediction and on several OOD systems, showing that the advantages of teacher forcing extend beyond our ViT architecture.

1 INTRODUCTION

Data-driven neural solvers and neural operators for spatiotemporal dynamics governed by partial differential equations (PDEs, Evans, 2022) have shown strong promise for modeling complex physical systems (Li et al., 2021; Kovachki et al., 2023; Li et al., 2023; McCabe et al., 2024; Herde et al., 2024; Morel et al., 2025). However, generalization under distribution shift remains a major challenge (Koupaï et al., 2024; Serrano et al., 2025; Li et al., 2026), including changes in geometry, initial conditions, or physical parameters. Prior work has shown that large-scale PDE pretraining can substantially improve downstream transfer, sample efficiency, and performance after adaptation (Subramanian et al., 2023; Herde et al., 2024; McCabe et al., 2024; Morel et al., 2025). We study the performance and generalization of pretrained PDE models themselves. Specifically, we consider zero-shot OOD rollout prediction, where the model must generate a future spatiotemporal trajectory from a single initial condition without downstream finetuning.

Existing approaches build on PDE pretraining in different ways, either through downstream adaptation (Herde et al., 2024; McCabe et al., 2024; Morel et al., 2025; Wang et al., 2025) or through finetune-free mechanisms that learn from contexts or use test-time computation (Nzoyem et al., 2025; Serrano et al., 2025; Koupaï et al., 2025; Serrano et al., 2026). Our study initially followed the latter direction. However, we unexpectedly found that relatively modest changes to the pretraining formulation can lead to large differences in zero-shot OOD rollout performance, even when only a single initial condition is given at inference time. Remarkably, *with only about 25M parameters, a lightweight ViT-style (Dosovitskiy et al., 2021) spatiotemporal model outperforms a 629M-parameter pretrained Poseidon model (Herde et al., 2024) on several OOD systems, with approximately* $3\times$ *lower error on some of them.*

Recent work has increasingly examined how rollout training, temporal supervision, and alternative learning objectives affect multi-step neural PDE prediction (Brandstetter et al., 2022; Koehler et al., 2024; Koehler & Thuerey, 2025; Hou et al., 2026). Much of this work motivates rollout-aware

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.

training as a way to mitigate the training-inference mismatch between ground-truth inputs used during training and model-generated histories encountered during recursive inference (Brandstetter et al., 2022; Vlachas & Koumoutsakos, 2024; List et al., 2025). However, their role in zero-shot OOD rollout prediction remains much less understood. We therefore ask how strongly pretraining design can shape the pretrained model's zero-shot OOD rollout capability when only a single initial condition is available.

We summarize the resulting findings and contributions as follows:

- We identify unexpectedly strong zero-shot OOD rollout capabilities in a lightweight pretrained spatiotemporal ViT. With only about 25M parameters, our model outperforms the pretrained 629M-parameter Poseidon-L on several OOD fluid systems, with particularly large gains on some of them. These predictions require only a single initial condition, without downstream finetuning, contextual demonstrations, or test-time adaptation.
- Our experiments suggest that pretraining choices can substantially alter the zero-shot OOD generalization of the same model. Despite its training-inference mismatch, teacher forcing yields better overall source-domain and OOD performance than AR rollout training. These results indicate that matching training and inference histories is not necessarily preferable.
- Comparisons with FNO show that the benefits of teacher forcing extend beyond our ViT architecture. Teacher forcing improves source-domain accuracy and yields OOD gains on several systems for both backbones, with particularly large improvements on some dynamics, supporting it as a favorable pretraining choice across the evaluated architectures.

2 Related Work

Neural operators, PDE pretraining, and OOD generalization. Neural operators learn mappings between function spaces and provide a data-driven approach to approximating PDE solution operators (Li et al., 2021; Kovachki et al., 2023; Li et al., 2023; Boullé & Townsend, 2024). Recent work has extended this framework toward large-scale pretraining across multiple PDEs. Models such as MPP (McCabe et al., 2024), DPOT (Hao et al., 2024), Poseidon (Herde et al., 2024), and DISCO (Morel et al., 2025) pretrain on heterogeneous PDE datasets and show that shared pretraining can improve downstream prediction, sample efficiency, and transfer across physical systems (Subramanian et al., 2023). More recent work such as AOT-POT (Lv et al., 2026) further studies how the representation of solution operators during pretraining affects the ability to learn across heterogeneous PDEs.

Most existing studies evaluate OOD generalization of pretrained PDE models after downstream finetuning (Herde et al., 2024; McCabe et al., 2024; Morel et al., 2025; Lv et al., 2026). Another line of work considers generalization with frozen shared model weights through contextual information or additional test-time computation, including Neural Context Flows (Nzoyem et al., 2025), Zebra (Serrano et al., 2025), ENMA (Koupaï et al., 2025), and test-time neural-operator splitting (Serrano et al., 2026). In contrast, we study the OOD generalization of the pretrained model itself: the model directly rolls out unseen dynamics without downstream finetuning, contextual demonstrations, or test-time adaptation.

Training formulations for multi-step PDE prediction. Training and temporal prediction formulations are important considerations for long-horizon PDE prediction (Koehler et al., 2024; List et al., 2025). Early work explored autoregressive PDE prediction (Geneva & Zabaras, 2020) and solver-in-the-loop training to reduce the mismatch between training and recurrent inference (Um et al., 2020). Brandstetter et al. (2022) further introduced pushforward training and temporal bundling for autoregressive neural PDE solvers. Subsequent work has studied long-horizon rollout and iterative refinement in PDE-Refiner (Lippe et al., 2023), learning from model-generated predictions (Vlachas & Koumoutsakos, 2024), and unrolled training strategies in APEBench (Koehler et al., 2024) and List et al. (2025). More recent work has investigated stable autoregressive prediction of chaotic dynamics (Pedersen et al., 2025), the interplay between training objectives and numerical errors in multi-step prediction (Koehler & Thuerey, 2025), and continuous-time prediction through flow-matched neural operators (Hou et al., 2026). In Section 5, we examine whether the adverse effects of training-inference mismatch reported in prior work extend to zero-shot OOD prediction.

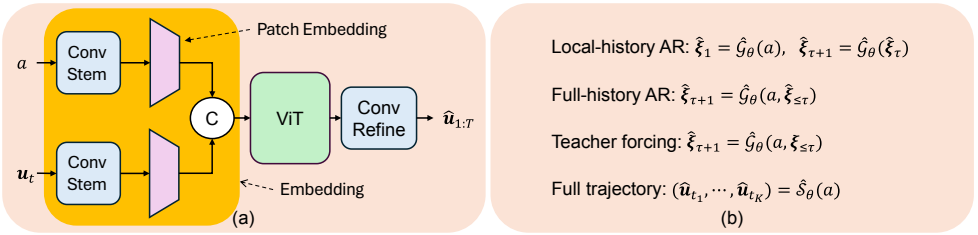


Figure 1: Overview of our model architecture and prediction formulations. (a) The ViT-style spatiotemporal model uses separate embeddings for the initial condition (IC) and subsequent evolution; C denotes concatenation. (b) The four prediction formulations considered in our controlled comparisons, where ξ_τ denotes the τ-th evolution block.

3 METHODS

Problem formalization. Consider a generic time-dependent PDE defined over a spatial domain $\Omega \subset \mathbb{R}^d$ and a time horizon $T > 0$:

$$\begin{aligned} \frac{\partial \boldsymbol{u}}{\partial t}(\boldsymbol{x}, t) &= \mathcal{F}(\boldsymbol{x}, t, \boldsymbol{u}, \nabla_{\boldsymbol{x}}\boldsymbol{u}, \nabla^2_{\boldsymbol{x}}\boldsymbol{u}, \ldots), && \forall (\boldsymbol{x}, t) \in \Omega \times (0, T], \\ \mathcal{B}(\boldsymbol{u})(\boldsymbol{x}, t) &= 0, && \forall (\boldsymbol{x}, t) \in \partial\Omega \times [0, T], \\ \boldsymbol{u}(\boldsymbol{x}, 0) &= a(\boldsymbol{x}), && \forall \boldsymbol{x} \in \Omega, \end{aligned} \tag{1}$$

where $\boldsymbol{u}(\boldsymbol{x}, t)$ denotes the solution field with $\boldsymbol{u}(\cdot, t) \in \mathcal{U} \subset L^p(\Omega; \mathbb{R}^m)$, and m denotes the number of state variables. The initial condition is given by $a \in \mathcal{U}$. The operators $\mathcal{F}$ and $\mathcal{B}$ specify the governing dynamics and boundary conditions, respectively. The solution to Equation 1 can be represented by the solution operator

$$\mathcal{S}_t : a \mapsto \boldsymbol{u}(\cdot, t). \tag{2}$$

Zero-shot rollout prediction. For discrete prediction times $0 = t_0 < t_1 < \cdots < t_K = T$, we write $\boldsymbol{u}_t := \boldsymbol{u}(\cdot, t)$. The solution operator induces a trajectory

$$\mathcal{S}(a) := (\boldsymbol{u}_{t_1}, \ldots, \boldsymbol{u}_{t_K}), \qquad \boldsymbol{u}_{t_i} := \mathcal{S}_{t_i}(a). \tag{3}$$

Our goal is to learn a neural approximation $\hat{\mathcal{S}}_\theta$ such that

$$\hat{\mathcal{S}}_\theta(a) = (\hat{\boldsymbol{u}}_{t_1}, \ldots, \hat{\boldsymbol{u}}_{t_K}) \approx \mathcal{S}(a). \tag{4}$$

We focus on zero-shot OOD rollout prediction. At test time, the pretrained model receives only the initial condition a of a trajectory drawn from an unseen PDE distribution and predicts its subsequent evolution without downstream finetuning, contextual demonstrations, or test-time adaptation.

Spatiotemporal Transformer. Having defined the zero-shot rollout setting, we seek a lightweight backbone for studying pretrained PDE dynamics. Existing large-scale pretrained PDE models can incur substantial computational cost (Herde et al., 2024). We therefore explore a lightweight model based on the Vision Transformer (ViT) (Dosovitskiy et al., 2021) and its video extension ViViT (Arnab et al., 2021). ViViT represents dense videos through spatiotemporal patchification, jointly compressing the spatial and temporal dimensions into a relatively short sequence of tubelet tokens while retaining strong performance on large-scale video recognition tasks. This representation provides a natural fit for PDE trajectories, which similarly consist of dense spatial fields evolving over time. We also considered VICON (Cao et al., 2026) and Zebra (Serrano et al., 2025), two recent Transformer-based PDE models closely related to our setting. However, differences in their problem settings and design choices led us to favor a ViT/ViViT-based backbone, as discussed in Appendix A.

We adapt the ViViT backbone to PDE rollout prediction with several lightweight modifications. To complement the non-overlapping tubelet representation with local spatial processing, we use a lightweight convolutional stem (Wu et al., 2021) before patchification and a convolutional refinement (Ranftl et al., 2021) after patch recovery. To better align the patchification with PDE evolution, we use separate embeddings for the initial condition and subsequent evolution.

Dual temporal embedding. We embed the initial condition and subsequent evolution separately to reflect their distinct roles in PDE rollout prediction. While standard tubelet patchification jointly compresses consecutive states along the temporal dimension, the initial condition serves as the starting state from which the subsequent dynamics evolve. Our design therefore keeps the initial condition separate while retaining temporal compression for the evolution. The full architecture is as illustrated in Figure 1(a). Detailed architectural configurations are provided in Appendix B.

Temporal Prediction and Training Formulations. Prior work has shown that prediction and training formulations can substantially affect PDE rollout performance (Koehler et al., 2024; List et al., 2025). However, how these formulations affect the zero-shot OOD generalization of pretrained PDE models remains underexplored. Specifically, we consider four formulations: local-history autoregressive (AR) rollout, full-history AR rollout with and without teacher forcing (Bengio et al., 2015), and non-autoregressive full-trajectory prediction (Liu et al., 2024), as summarized in Figure 1(b).

For the three AR formulations in Figure 1(b), prediction is performed over the temporally compressed evolution blocks induced by the spatiotemporal embedding. Here, ξ_τ denotes the τ-th evolution block, corresponding to `tubelet_size` consecutive states at the original temporal resolution. We use $\hat{\mathcal{G}}\theta$ to denote the learned block-wise evolution operator and $\hat{\mathcal{S}}\theta$ to denote the learned full-trajectory operator. Local-history AR uses only the preceding predicted block, while full-history AR uses the initial condition and predicted history. Teacher forcing replaces the predicted history with ground truth during training. Full-trajectory prediction produces all future states jointly from the initial condition. At inference, all three AR variants generate trajectories autoregressively from the initial condition; teacher forcing is used only during training. In Section 5, we compare these formulations to examine how pretraining choices affect OOD generalization within the same backbone.

4 Experiments

Datasets. We use the datasets publicly released with Poseidon (Herde et al., 2024). We jointly pretrain each model on 6 datasets of compressible Euler (CE) gas dynamics and incompressible Navier-Stokes (NS) flow: CE-Gauss, CE-KH, CE-RP, CE-CRP, NS-Gauss, and NS-Sines. These datasets capture different forms of fluid evolution, including interacting vortices, layers moving at different velocities, and propagating shock waves.

For zero-shot OOD evaluation, we consider five additional settings: Richtmyer–Meshkov instability (CE-RM), Riemann problems with perturbed interfaces (CE-RKH), Brownian Bridge (NS-BB), Shear Layer (NS-SL), and Vortex Sheet (NS-VS). These tasks retain the governing equation families and physical variables used during pretraining but introduce different initial-condition distributions and resulting flow patterns. The prediction challenges include tracking spatially distributed swirling flows, shock-driven interface deformation, and the bending and roll-up of thin fluid layers.

Representative ground-truth animations from all six pretraining and five OOD datasets are provided in the supplementary material to illustrate these differences. Dataset details and the correspondence between our abbreviations and Poseidon's original names are provided in Appendix C.

Metrics. Consistent with the L_1-based evaluation adopted by Poseidon (Herde et al., 2024), we use the time-averaged relative L_1 error as our primary metric,

$$\mathcal{E} = \frac{1}{K} \sum_{k=1}^{K} \frac{|\hat{\boldsymbol{u}}_{t_k} - \boldsymbol{u}_{t_k}|_1}{|\boldsymbol{u}_{t_k}|_1}, \tag{5}$$

where each L_1 norm is computed over all evaluation trajectories, physical variables, and spatial locations. The same metric and evaluation procedure are used for all models. Corresponding relative L_2 results are provided in Appendix I.

Training. Our models are trained on N trajectories to predict the $K = 20$ future states following the initial condition. We minimize the mean squared error (MSE) averaged over all prediction times,

$$\mathcal{L} = \frac{1}{NK} \sum_{i=1}^{N} \sum_{k=1}^{K} \mathrm{MSE}\left(\hat{\boldsymbol{u}}_{t_k}^{(i)}, \boldsymbol{u}_{t_k}^{(i)}\right), \tag{6}$$

Table 1: Zero-shot OOD AR rollout performance from a single initial condition. Our ViT is pretrained for 40 epochs, while the Poseidon results use the released checkpoints with autoregressive rollout; Poseidon-L's stronger direct-prediction results appear in Table 6. GPU-hours per epoch are estimated from end-to-end wall-clock time under our A100 setup. Compute estimation and memory profiling details are provided in Appendix D. Bold indicates the lowest prediction error in each OOD column. Lower is better. Corresponding relative L_2 results are reported in Table 16.

				CE		NS		
Model	Params	Epochs	GPU-h/epoch	RM	RKH	BB	SL	VS
Poseidon-L	629M	20	≈ 15.4	**1.1561**	0.9814	1.2350	0.6578	1.4902
Poseidon-B	158M	40	≈ 6.8	1.3735	0.7249	1.6309	0.5628	1.9967
Poseidon-T	21M	40	≈ 2.1	2.2944	0.8913	3.3424	0.5155	3.7799
Our ViT	25M	40	≈ 0.29	1.2164	**0.6916**	**0.1416**	**0.4358**	**0.3020**

where the MSE is averaged over the physical variables and spatial dimensions. The same training loss is used for the FNO models in our controlled comparisons. All models in our controlled experiments are trained on a single NVIDIA A100-80GB GPU.

5 Results and Discussion

We begin with local-history AR rollout, a natural choice motivated by the common formulation of PDE evolution operators as mappings from the current state to the next. We compare the zero-shot OOD AR rollout performance and pretraining compute budgets of our model and Poseidon, and ask:

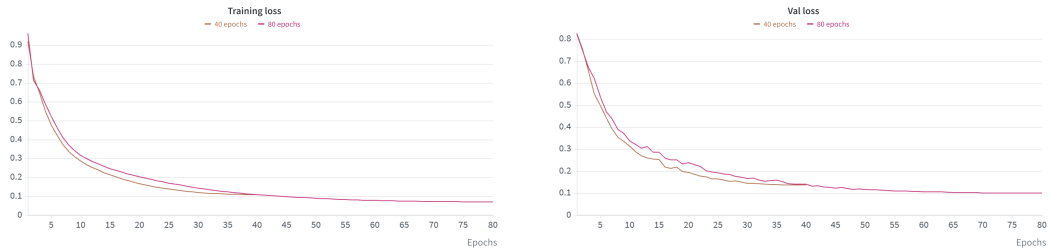
Why can a model that is substantially smaller and cheaper to pretrain achieve stronger overall source-domain and zero-shot OOD autoregressive rollout performance than Poseidon?

For comparability in pretraining duration, we train our 25M ViT for 40 epochs, matching the released Poseidon-B and Poseidon-T checkpoints; Poseidon-L uses its released 20-epoch checkpoint (Table 1). Despite substantially fewer estimated GPU-hours per epoch, our ViT achieves lower relative L_1 error than all three Poseidon checkpoints on most evaluated OOD dynamics. Our ViT's errors are about $9\times$ lower on Brownian Bridge and $5\times$ lower on Vortex Sheet than Poseidon-L's, with even larger gaps to Poseidon-B and Poseidon-T. Without downstream finetuning, errors of 0.1416 and 0.3020, respectively, indicate nontrivial predictive accuracy. Motivated by these results, we next investigate which pretraining choices contribute to the gains through controlled comparisons.

Pretraining Convergence Analysis. We first examine pretraining convergence and source-domain prediction accuracy to understand how continued optimization affects zero-shot OOD generalization. To this end, we examine Poseidon's pretraining curves reported in the original paper alongside our model's results after 40 epochs of pretraining, and further assess the effect of extending our pretraining to 80 epochs. Poseidon's curves indicate a relatively stable optimization regime toward the end of pretraining (Herde et al., 2024, Appendix D.2, Figure 22): both training and validation losses decrease substantially to the scale of 10^{-1} or below, with the curves gradually flattening toward the end. Our model exhibits a broadly similar trend, as shown in Figure 2.

We next examine how our model benefits from extending pretraining from 40 to 80 epochs. Figure 2 shows further reductions in training and validation losses, accompanied by improved source-domain accuracy across all six datasets (Table 2). However, these improvements do not translate uniformly to OOD prediction performance (Table 3). Although individual OOD errors change, the overall pattern remains consistent across the two training durations. We therefore use the 80-epoch setting for the controlled comparisons below, as it yields better source-domain accuracy without qualitatively changing the observed OOD behavior.

Training-Inference Mismatch. Prior work has shown that mismatches between training and autoregressive inference can substantially affect multi-step prediction accuracy (Brandstetter et al., 2022; Vlachas & Koumoutsakos, 2024; List et al., 2025). We therefore investigate whether Poseidon's relatively high source-domain errors in Table 2 arise from a mismatch between AR rollout and its



(a) Training loss.

(b) Validation loss.

Figure 2: Training and validation loss curves for 40- and 80-epoch pretraining.

Table 2: Source-domain prediction performance on the six pretraining datasets. † denotes training for 80 epochs, and ‡ denotes Poseidon-L evaluated using non-autoregressive direct prediction without recursive rollout. Lower is better. Corresponding relative L_2 results are reported in Table 17.

Model	CE-Gauss	CE-KH	CE-RP	CE-CRP	NS-Gauss	NS-Sines
Poseidon-L	0.8384	1.8891	0.6317	0.8654	0.4889	0.4197
Poseidon-B	0.6302	1.5531	0.9496	0.9598	0.1656	0.2886
Poseidon-T	0.7180	3.6888	1.4239	1.0131	1.3570	0.3888
Our ViT	0.2320	0.3262	0.2203	0.5484	0.1332	0.3866
Our ViT†	0.2289	0.2779	0.1710	0.4865	0.0999	0.3200
Poseidon-L‡	**0.0716**	**0.1208**	**0.0387**	**0.2744**	**0.0198**	**0.0527**

original training formulation[1]. Poseidon exploits the semigroup property of PDE solution operators (Evans, 2022) through all2all training, learning time-conditioned mappings between pairs of ground-truth states (Herde et al., 2024). During recursive rollout, however, later predictions depend on model-generated states. The same pretrained model also supports direct prediction, mapping $a \mapsto \hat{\boldsymbol{u}}_{t_k}\ \forall k \in [1, 2, \ldots, K]$. We therefore evaluate Poseidon-L by directly predicting every future state from the initial condition, assembling the complete trajectory without recursive rollout.

This change substantially improves Poseidon-L's source-domain accuracy. As shown in Table 2, direct prediction yields much lower errors across all six datasets, outperforming even our 80-epoch ViT. Poseidon's original paper also compares direct prediction with AR rollout after downstream finetuning, without finding a consistently superior strategy across tasks (Herde et al., 2024, Appendix D.6.2). In our evaluation of the pretrained checkpoint without downstream adaptation, however, direct prediction has a clear and consistent advantage on the source-domain datasets.

However, Poseidon-L's strong source-domain accuracy under direct prediction does not translate into consistently better OOD performance. Our much smaller 40-epoch ViT achieves lower errors on three of the five systems, with substantial gains on NS-BB and NS-VS (Table 3). Both models remain inaccurate on CE-RM, while Poseidon-L performs moderately better on CE-RKH and substantially better on NS-SL. We therefore investigate how training formulations and temporal conditioning affect zero-shot OOD generalization through controlled comparisons.

Effects of Training Formulation and Temporal Conditioning. We next ask whether the observed OOD performance primarily reflects the ViT-style backbone or also depends on the temporal prediction and training formulation. To examine this, we compare the four formulations introduced in Figure 1(b) using the same 25M ViT backbone, with all variants pretrained for 80 epochs.

We first compare local-history AR rollout with non-autoregressive full-trajectory prediction, which predicts all future states jointly from the initial condition. As shown in Table 4, full-trajectory prediction achieves slightly lower errors on CE-RM and CE-RKH, with a small increase on NS-SL. However, switching from local-history AR to full-trajectory prediction increases the error on NS-BB

[1]For Poseidon's AR results, we use the authors' released pretrained checkpoints and official rollout pipeline, changing only the evaluation metric to Equation 5.

Table 3: Zero-shot OOD prediction performance of our ViT after 40 and 80 epochs of pretraining and Poseidon-L using non-autoregressive direct prediction without recursive rollout. Lower is better. Corresponding relative L_2 results are reported in Table 18.

Model	CE-RM	CE-RKH	NS-BB	NS-SL	NS-VS
Our ViT†	**1.1756**	0.7685	**0.1068**	0.4441	**0.2591**
Our ViT	1.2164	0.6916	0.1416	0.4358	0.3020
Poseidon-L‡	1.2821	**0.5809**	0.2576	**0.1694**	0.7152

Table 4: Zero-shot OOD prediction performance of the 25M ViT† under different training formulations. Lower is better. Corresponding relative L_2 results are reported in Table 19.

Training formulation	CE-RM	CE-RKH	NS-BB	NS-SL	NS-VS
Full-trajectory	1.1464	0.7589	0.2916	0.4534	0.4049
Local-history AR	1.1756	0.7685	0.1068	0.4441	0.2591
Full-history AR	**1.1240**	0.6925	0.0917	0.4649	0.2743
Teacher forcing	1.1971	**0.6252**	**0.0904**	**0.4435**	**0.2385**

from 0.1068 to 0.2916 (approximately $2.7\times$) and on NS-VS from 0.2591 to 0.4049 (approximately $1.6\times$). Thus, using the same ViT-style backbone does not automatically preserve the strong OOD performance observed under AR rollout. The corresponding training curves and source-domain prediction results are reported in Appendices E and F, respectively.

We further compare local-history and full-history AR rollout to examine whether access to longer temporal context improves generalization. Full-history conditioning reduces errors on CE-RM, CE-RKH, and NS-BB, but slightly increases errors on NS-SL and NS-VS. For example, the NS-BB error decreases from 0.1068 to 0.0917, whereas the NS-VS error increases from 0.2591 to 0.2743. The benefits are therefore mixed: conditioning on the complete prediction history does not consistently outperform using only the preceding block. These comparisons show that the temporal prediction formulation materially affects zero-shot OOD behavior even when the ViT backbone is held fixed. This sensitivity is consistent with prior findings for multi-step PDE prediction (Koehler et al., 2024; List et al., 2025).

Teacher Forcing Results. We next analyze teacher forcing separately, as its effect on our ViT differs from the trend commonly reported in prior PDE studies. Prior work (Brandstetter et al., 2022; Vlachas & Koumoutsakos, 2024; List et al., 2025) argues that training on ground-truth states, including teacher forcing, creates a distribution shift when the model is instead conditioned on its own predictions during autoregressive inference; accordingly, rollout-aware training on model-generated states has often been found to improve long-horizon PDE prediction. However, evidence from studies of autoregressive Transformer language models (He et al., 2021; Yee et al., 2024; Frydenlund, 2024; 2025) suggests that this mismatch is not necessarily detrimental, as models may recover from imperfect histories and the effects of teacher forcing can depend on the representation and supervision structure. We therefore directly compare full-history AR training with and without teacher forcing in our ViT, corresponding respectively to mismatched and matched training-inference conditions.

The overall comparison favors teacher forcing, although the improvements on individual OOD systems are generally moderate. Relative to full-history AR, teacher forcing reduces the error on four of the five dynamics: from 0.6925 to 0.6252 on CE-RKH, from 0.0917 to 0.0904 on NS-BB, from 0.4649 to 0.4435 on NS-SL, and from 0.2743 to 0.2385 on NS-VS. CE-RM is the exception, where the error increases from 1.1240 to 1.1971 and both formulations remain inaccurate. Thus, teacher forcing provides a more favorable overall OOD trade-off rather than a uniformly large improvement on every dynamic. The corresponding source-domain results further reinforce this conclusion, with teacher forcing achieving the lowest error across all six pretraining datasets under relative L_1 (Appendix F).

That teacher forcing remains the more favorable overall choice despite its training-inference mismatch contrasts with prior PDE results favoring rollout-aware training on model-generated states. This suggests that training-inference mismatch does not necessarily make teacher forcing inferior for our

Table 5: Zero-shot OOD prediction performance of approximately parameter-matched ViT-T and FNO models. Bold indicates the better formulation within each model family. Lower is better. Corresponding relative L_2 results are reported in Table 20.

Model	Training formulation	CE-RM	CE-RKH	NS-BB	NS-SL	NS-VS
ViT-T† (10M)	Full-history AR	1.1562	0.7256	0.1340	0.4743	0.2870
ViT-T† (10M)	Teacher forcing	**1.1167**	**0.6780**	**0.1110**	**0.4359**	**0.2798**
FNO† (10M)	AR rollout	1.2761	**0.7712**	0.7788	**0.4369**	0.6878
FNO† (10M)	Teacher forcing	**1.1103**	1.1325	**0.0305**	0.5031	**0.2370**

ViT. We next test whether this preference is specific to the ViT-style representation by repeating the same comparison with an approximately parameter-matched FNO.

Benefits of Teacher Forcing across Architectures. We next examine whether the benefits of teacher forcing depend on the model architecture. Evidence from autoregressive language modeling largely concerns standard Transformers, whose token-based full-history conditioning is closer to that used by our ViT (He et al., 2021; Yee et al., 2024; Frydenlund, 2024; 2025). Poseidon is also Transformer-based, but employs a hierarchical multiscale operator Transformer arranged in a U-Net-style encoder-decoder, resulting in a substantially different representation (Herde et al., 2024). We therefore introduce FNO (Li et al., 2021) as a contrasting non-Transformer backbone and compare AR rollout training with teacher forcing for this backbone:

$$\hat{\boldsymbol{u}}_{t_{k+1}} = \hat{\mathcal{G}}_\theta(\hat{\boldsymbol{u}}_{t_k}) \qquad \text{versus} \qquad \hat{\boldsymbol{u}}_{t_{k+1}} = \hat{\mathcal{G}}_\theta(\boldsymbol{u}_{t_k}). \tag{7}$$

We instantiate this comparison using ViT-T, a 10M variant of our ViT-style architecture, and an approximately parameter-matched FNO. For FNO, we follow the architectural settings specified in the Poseidon study, while doubling its reported channel width; in our implementation, this results in an approximately 10M-parameter model comparable in size to ViT-T. Both models are pretrained for 80 epochs on the same data using rollout-aware and teacher-forced training. We conduct the comparison at this scale because FNO is substantially more expensive to train in our setting: even the 10M model requires approximately 112 A100 GPU-hours for 80 epochs. Full implementation and training details are provided in Appendix B.

The results reveal that the benefits of teacher forcing persist across ViT model scales. ViT-T strengthens the pattern observed with the 25M ViT: teacher forcing reduces the OOD error on all five dynamics, with reductions ranging from approximately 2.5% on NS-VS to 17% on NS-BB. Thus, the favorable overall effect of teacher forcing is not specific to the 25M configuration and remains consistent after reducing the model to 10M parameters.

FNO also benefits substantially from teacher forcing, although its effects vary across dynamics. Teacher forcing reduces the error on NS-BB from 0.7788 to 0.0305 (approximately $25.5\times$ lower) and on NS-VS from 0.6878 to 0.2370 (approximately $2.9\times$ lower). However, teacher forcing increases the CE-RKH and NS-SL errors by approximately 47% and 15%, respectively; the CE-RKH regression is larger under relative L_2 (Appendix I). Teacher forcing also yields better overall source-domain performance for both backbones (Appendix F). These improvements and the large OOD gains on NS-BB and NS-VS support teacher forcing as a favorable default for both backbones in the evaluated setting, despite its training-inference mismatch and the trade-offs on individual OOD systems.

Final Zero-Shot OOD Comparison. We now return to our 25M ViT trained for 40 epochs with teacher forcing, comparing and visualizing its zero-shot predictions against those of the 629M Poseidon-L under direct prediction. Both models predict unseen dynamics from a single initial condition without downstream finetuning, contextual demonstrations, or test-time adaptation. As shown in Table 6, our ViT achieves lower error on three of the five OOD systems, including approximately $3.9\times$ and $2.8\times$ lower errors on NS-BB and NS-VS, respectively.

Figure 3 visualizes the predicted evolution on NS-BB. Our ViT captures the dominant spatial organization and evolution of the flow, whereas Poseidon-L increasingly deviates from the ground-truth dynamics. Visualizations for NS-VS and NS-SL are provided in Appendix G, showing that our ViT pretrained with teacher forcing broadly captures the principal evolution patterns across the three NS systems, including NS-SL where Poseidon-L achieves lower numerical error. The 10M FNO

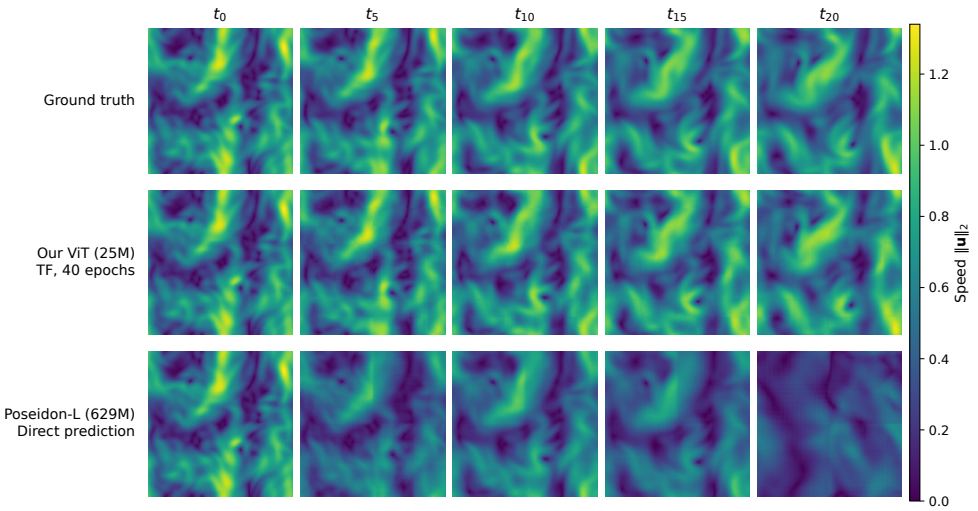


Figure 3: Visualization of NS-BB evolution using our ViT pretrained under teacher forcing (TF) and Poseidon-L with $a \mapsto \hat{\boldsymbol{u}}_{t_k}$ prediction, without any finetuning, contextual demonstrations, or test-time adaptation.

Table 6: Final zero-shot OOD comparison between our 25M ViT after 40 epochs of teacher-forced pretraining and Poseidon-L‡ under direct prediction. The error ratio is Poseidon-L‡ / Our ViT; values above one favor our ViT. Lower is better. Corresponding relative L_2 results are reported in Table 21.

Model	CE-RM	CE-RKH	NS-BB	NS-SL	NS-VS
Our ViT (teacher forcing)	**1.1304**	0.6746	**0.0666**	0.4450	**0.2561**
Poseidon-L‡	1.2821	**0.5809**	0.2576	**0.1694**	0.7152
Poseidon-L‡ / Our ViT	1.13×	0.86×	3.87×	0.38×	2.79×

pretrained for 40 epochs with teacher forcing also achieves approximately 7.3× lower error than Poseidon-L on NS-BB (Table 14), with corresponding visualizations provided in Appendix H. These results further support the benefits of teacher-forced pretraining for zero-shot OOD generalization. Complete rollout animations for all five OOD systems are provided in the supplementary material.

6 Conclusion

We empirically show that pretraining design substantially affects the zero-shot OOD generalization of pretrained PDE models. With an appropriate training formulation, our 25M spatiotemporal ViT outperforms the 629M Poseidon-L on several unseen fluid systems, using only a single initial condition without downstream finetuning, contextual demonstrations, or test-time adaptation. Controlled comparisons highlight the roles of training formulation and temporal conditioning. In particular, full-history attention combined with teacher-forced pretraining improves source-domain accuracy and yields better overall OOD performance than AR rollout training despite the training-inference mismatch. This configuration achieves approximately 3-4× lower error than Poseidon-L on two OOD systems. Results with FNO under teacher forcing further reinforce this finding, suggesting that training-inference mismatch is not necessarily detrimental to predictive performance.

Limitations. Some supplementary rollout videos exhibit grid-aligned spatial artifacts despite capturing coherent large-scale evolution. Spatiotemporal patchification and temporal compression may contribute to these artifacts. Future work will investigate reduced temporal compression, overlapping patchification (Wang et al., 2022), overlap-averaged prediction fusion (Cho et al., 2024), and stronger spatial refinement (Ranftl et al., 2021) to improve fine-scale reconstruction and zero-shot OOD accuracy.

AI USE STATEMENT

Generative AI tools were used to assist with literature discovery, drafting and editing manuscript text, manuscript organization, and the drafting and debugging of auxiliary visualization and data-processing code. They were also used to provide feedback on descriptions and interpretations of experimental results. All suggested references were manually verified against their original sources, all AI-assisted code was reviewed and tested by the authors, and all analyses and claims were checked against the experimental outputs. Generative AI was not used to generate datasets or reported experimental results. The authors take full responsibility for the content of this work.

ETHICS STATEMENT

This work uses publicly available numerical PDE simulation datasets and released model checkpoints. It involves no human participants, personal data, or sensitive attributes. The evaluated models are research prototypes and have not been validated for safety-critical or consequential real-world applications; their predictions should not be treated as substitutes for verified numerical solvers in such settings. We are not aware of other significant ethical concerns arising from this work.

REPRODUCIBILITY STATEMENT

The datasets, official data splits, preprocessing procedure, evaluation protocol, and training objective are described in Section 4. Architectural and implementation details are provided in Appendix B, and the compute estimation procedure is documented in Appendix D. Additional source-domain results, training curves, and qualitative rollouts are reported in Appendices F, E, and G. We use the publicly released Poseidon datasets, checkpoints, and preprocessing pipeline wherever applicable. Our implementation and experiment configurations will be released upon publication.

REFERENCES

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, pp. 6816–6826, 2021. URL https://doi.org/10.1109/ICCV48922.2021.00676.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/e995f98d56967d946471af29d7bf99f1-Paper.pdf.

Nicolas Boullé and Alex Townsend. A mathematical guide to operator learning. In *Handbook of Numerical Analysis*, volume 25, pp. 83–125. Elsevier, 2024.

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.

Yadi Cao, Yuxuan Liu, Liu Yang, Rose Yu, Hayden Schaeffer, and Stanley Osher. VICON: Vision in-context operator networks for multi-physics fluid dynamics prediction. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=6V3YmHULQ3.

Seokju Cho, Heeseong Shin, Sunghwan Hong, Anurag Arnab, Paul Hongsuck Seo, and Seungryong Kim. Cat-seg: Cost aggregation for open-vocabulary semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4113–4123, June 2024.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale.

In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Lawrence C Evans. *Partial differential equations*, volume 19. American mathematical society, 2022.

Arvid Frydenlund. The mystery of the pathological path-star task for language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 12493–12516, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.695. URL https://aclanthology.org/2024.emnlp-main.695/.

Arvid Frydenlund. Language models, graph searching, and supervision adulteration: When more supervision is less and how to make more more. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29011–29059, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1409. URL https://aclanthology.org/2025.acl-long.1409/.

Nicholas Geneva and Nicholas Zabaras. Modeling the dynamics of pde systems with physics-constrained deep auto-regressive networks. *Journal of Computational Physics*, 403:109056, 2020.

Zhongkai Hao, Chang Su, Songming Liu, Julius Berner, Chengyang Ying, Hang Su, Anima Anandkumar, Jian Song, and Jun Zhu. DPOT: Auto-regressive denoising operator transformer for large-scale PDE pre-training. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=X7UnDevHOM.

Tianxing He, Jingzhao Zhang, Zhiming Zhou, and James Glass. Exposure bias versus self-recovery: Are distortions really incremental for autoregressive text generation? In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5087–5102, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.415. URL https://aclanthology.org/2021.emnlp-main.415/.

Maximilian Herde, Bogdan Raonic, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bezenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for PDEs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=JC1VKK3UXk.

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

Felix Koehler and Nils Thuerey. Neural emulator superiority: When machine learning for PDEs surpasses its training data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=g9olLDlaqH.

Felix Koehler, Simon Niedermayr, Rüdiger Westermann, and Nils Thuerey. Apebench: A benchmark for autoregressive neural emulators of pdes. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 120252–120310. Curran Associates, Inc., 2024. doi: 10.52202/079017-3822. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/d9875ebcf74bccdc5076acab0dbee62c-Paper-Datasets_and_Benchmarks_Track.pdf.

Armand Kassaï Koupaï, Jorge Mifsut Benet, Yuan Yin, Jean-Noël Vittaut, and Patrick Gallinari. Boosting generalization in parametric PDE neural solvers through adaptive conditioning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=GuY0zB2xVU.

Armand Kassaï Koupaï, Lise Le Boudec, Louis Serrano, and Patrick Gallinari. ENMA: Tokenwise autoregression for continuous neural PDE operators. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=3CYXSMFv55.

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 http://jmlr.org/papers/v24/21-1524.html.

Siyang Li, Yize Chen, Yan Guo, Ming Huang, and Hui Xiong. Towards generalizable pde dynamics forecasting via physics-guided invariant learning. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 72050–72079, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/754612bde73a8b65ad8743f1f6d8ddf6-Paper-Conference.pdf.

Zongyi Li, Nikola Borislavov 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.

Zongyi Li, Nikola Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, and Animashree Anandkumar. Geometry-informed neural operator for large-scale 3d pdes. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 35836–35854. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/70518ea42831f02afc3a2828993935ad-Paper-Conference.pdf.

Phillip Lippe, Bastiaan S. Veeling, Paris Perdikaris, Richard E Turner, and Johannes Brandstetter. PDE-refiner: Achieving accurate long rollouts with neural PDE solvers. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=Qv6468llWS.

Bjoern List, Li-Wei Chen, Kartik Bali, and Nils Thuerey. Differentiability in unrolled training of neural physics simulators on transient dynamics. *Computer Methods in Applied Mechanics and Engineering*, 433:117441, 2025.

Yuxuan Liu, Jingmin Sun, Xinjie He, Griffin Pinney, Zecheng Zhang, and Hayden Schaeffer. PROSE-FD: A multimodal PDE foundation model for learning multiple operators for forecasting fluid dynamics. In *Neurips 2024 Workshop Foundation Models for Science: Progress, Opportunities, and Challenges*, 2024. URL https://openreview.net/forum?id=7tEA7ENjLd.

Qitan Lv, Hong Wang, Zhongkai Hao, Wen Wu, Xuenan Xu, Bowen Zhou, Feng Wu, and Chao Zhang. Aot-pot: Adaptive operator transformation for large-scale pde pre-training. *arXiv preprint arXiv:2605.15793*, 2026.

Michael McCabe, Bruno Régaldo-Saint Blancard, Liam Holden Parker, Ruben Ohana, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Siavash Golkar, Geraud Krawezik, Francois Lanusse, Mariel Pettee, Tiberiu Tesileanu, Kyunghyun Cho, and Shirley Ho. Multiple physics pretraining for spatiotemporal surrogate models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=DKSI3bULiZ.

Rudy Morel, Jiequn Han, and Edouard Oyallon. DISCO: learning to DISCover an evolution operator for multi-physics-agnostic prediction. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=6EZ3MDDf6p.

Roussel Desmond Nzoyem, David A.W. Barton, and Tom Deakin. Neural context flows for meta-learning of dynamical systems. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=8vzMLo8LDN.

Christian Pedersen, Laure Zanna, and Joan Bruna. Thermalizer: Stable autoregressive neural emulation of spatiotemporal chaos. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=73zrUyI5kB.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12159–12168, 2021. doi: 10.1109/ICCV48922.2021.01196.

Louis Serrano, Armand Kassaï Koupaï, Thomas X Wang, Pierre ERBACHER, and Patrick Gallinari. Zebra: In-context generative pretraining for solving parametric PDEs. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=22kNOkkokU.

Louis Serrano, Jiequn Han, Edouard Oyallon, Shirley Ho, and Rudy Morel. Test-time generalization for physics through neural operator splitting. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=gLLcjBJCXZ.

Shashank Subramanian, Peter Harrington, Kurt Keutzer, Wahid Bhimji, Dmitriy Morozov, Michael W. Mahoney, and Amir Gholami. Towards foundation models for scientific machine learning: Characterizing scaling and transfer behavior. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=zANxvzflMl.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Kiwon Um, Robert Brand, Yun (Raymond) Fei, Philipp Holl, and Nils Thuerey. Solver-in-the-loop: Learning from differentiable physics to interact with iterative pde-solvers. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6111–6122. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/43e4e6a6f341e00671e123714de019a8-Paper.pdf.

Aaron van den Oord, Oriol Vinyals, and koray kavukcuoglu. Neural discrete representation learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/7a98af17e63a0ac09ce2e96d03992fbc-Paper.pdf.

Pantelis Rafail Vlachas and Petros Koumoutsakos. Learning on predictions: Fusing training and autoregressive inference for long-term spatiotemporal forecasts. *Physica D: Nonlinear Phenomena*, 470:134371, 2024.

Hong Wang, Haiyang Xin, Jie Wang, Xuanze Yang, Fei Zha, huanshuo dong, and Yan Jiang. Mixture-of-experts operator transformer for large-scale pde pre-training. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 31498–31527. Curran Associates, Inc., 2025. doi: 10.52202/085713-1057. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/2d23a9991a6f64482bf395628e279f5f-Paper-Conference.pdf.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022. doi: 10.1007/s41095-022-0274-8.

Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22–31, 2021. doi: 10.1109/ICCV48922.2021.00009.

Evelyn Yee, Alice Li, Chenyu Tang, Yeon Ho Jung, Ramamohan Paturi, and Leon Bergen. Faithful and unfaithful error recovery in chain of thought. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=IPZ28ZqD4I.

APPENDIX

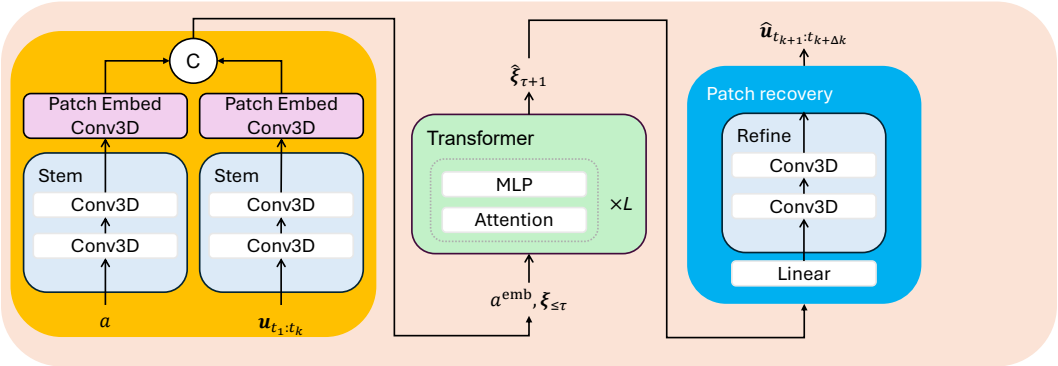


Figure 4: Detailed architecture of our spatiotemporal ViT under full-history teacher-forced training. Separate convolutional stems and patch embeddings encode the initial condition and ground-truth history as a^{emb} and $\xi_{\leq\tau}$. These tokens are concatenated (C) and processed by L Transformer blocks. The patch-recovery module maps the predicted block representation $\hat{\xi}_{\tau+1}$ to the next $\Delta k = \texttt{tubelet_size}$ states, where $k = \tau\Delta k$. At inference, model-generated histories replace ground-truth histories. Activation functions, normalization, residual connections, and tensor reshaping are omitted for clarity.

A Comparison with Existing Transformer-Based PDE Models

Recent Transformer-based approaches to PDE prediction include VICON (Cao et al., 2026) and Zebra (Serrano et al., 2025). VICON adopts a sequence-to-sequence formulation that differs substantially from our initial-condition-to-trajectory prediction setting, making its architecture less directly applicable to our study. Zebra, in contrast, addresses a closely aligned PDE prediction problem, but presents two limitations for our study. First, Zebra combines a VQ-VAE (van den Oord et al., 2017) representation with a Llama-based autoregressive model (Touvron et al., 2023), with both the representation and prediction formulated over a quantized vocabulary. This makes the overall framework less straightforward to modify than the continuous-valued ViT-style architecture used here, particularly when varying the temporal prediction objective while retaining the same underlying backbone. We therefore adopt the ViViT-based architecture described in Section 3.

The other, and more substantial, challenge in adopting Zebra for our setting is scalability, particularly its sequence length and associated training cost. In Zebra's validated 2D configuration, the VQ-VAE uses three spatial compression stages and two codebooks. At our 128×128 spatial resolution, each frame is therefore represented by a 16×16 latent grid with two codebook tokens per spatial location, resulting in

$$16 \times 16 \times 2 = 512$$

tokens per frame. A trajectory consisting of one initial condition and 20 future states would consequently require

$$21 \times 512 = 10{,}752$$

tokens. This already exceeds the maximum sequence length of 8,192 used in Zebra's validated configuration, before considering longer trajectories or additional contextual information.

The computational cost of the two-stage training pipeline is also substantial in our multi-PDE setting. Using Zebra's 2D VQ-VAE configuration on the six Poseidon pretraining datasets, we measure approximately 5.4 hours per VQ-VAE training epoch on our A100. Zebra trains its VQ-VAE for 300 epochs in the reported experiments. Although the difference in dataset size means that the same number of epochs is not necessarily required in our setting, the VQ-VAE alone introduces a substantial additional pretraining stage. In addition to the VQ-VAE, the subsequent Llama-based dynamics model must be trained separately, requiring approximately an additional 2.5 hours per epoch in our setting. Together with the rapid growth in sequence length, these costs make directly scaling the Zebra pipeline to our multi-PDE pretraining setting considerably more expensive than the lightweight spatiotemporal backbone used in this work.

Algorithm 1 Prediction under the four ViT formulations

Require: Initial condition a; block count $T^{\mathrm{emb}} = K/\Delta k$; formulation f; training or inference mode
Require: Ground-truth blocks $\xi_{1:T^{\mathrm{emb}}}$ for teacher-forced training
Ensure: Predicted future trajectory $\hat{\boldsymbol{u}}$

if $f =$ Full-trajectory **then**
  return $\hat{\mathcal{S}}_\theta(a)$ ▷ Future positions use learned mask embeddings
end if
$a^{\mathrm{emb}} \leftarrow \mathrm{Embed}(a)$
$\hat{\xi}_1 \leftarrow \hat{\mathcal{G}}_\theta(a^{\mathrm{emb}})$ ▷ $\hat{\xi}_1$ is patch-recovered and re-embedded before feedback
for $\tau = 1, \ldots, T^{\mathrm{emb}} - 1$ **do**
  if $f =$ Local-history AR **then**
    $\hat{\xi}_{\tau+1} \leftarrow \hat{\mathcal{G}}_\theta(\hat{\xi}_\tau)$ ▷ $\hat{\xi}_{\tau+1}$ is patch-recovered and re-embedded before feedback
  else if training **and** $f =$ Teacher forcing **then**
    $\hat{\xi}_{\tau+1} \leftarrow \hat{\mathcal{G}}_\theta(a^{\mathrm{emb}}, \xi_{1:\tau})$ ▷ Ground-truth embedded history
  else
    $\hat{\xi}_{\tau+1} \leftarrow \hat{\mathcal{G}}_\theta(a^{\mathrm{emb}}, \hat{\xi}_{1:\tau})$ ▷ $\hat{\xi}_{\tau+1}$ is patch-recovered and re-embedded before feedback
  end if
end for
$\hat{\boldsymbol{u}} \leftarrow \textsc{RecoverTrajectory}(\hat{\xi}_{1:T^{\mathrm{emb}}})$
return $\hat{\boldsymbol{u}}$

B Implementation Details

Spatiotemporal Embedding. Figure 4 illustrates the architecture under full-history teacher-forced training. The initial condition and evolution states are processed by independently parameterized convolutional stems. Each stem contains two Conv3D layers with kernel size $(1, 3, 3)$, stride $(1, 2, 2)$, and padding $(0, 1, 1)$, each followed by GELU. These layers reduce the spatial resolution by a factor of 4 while preserving the temporal dimension. Their temporal kernel size of 1 makes them equivalent to spatial convolutions applied independently at each time step with shared weights. The stem width is $\max(32, \lfloor D/8 \rfloor)$, where D is the Transformer embedding dimension.

Separate patch embeddings map the stem outputs to tokens. The initial condition is embedded separately as a^{emb}, while the evolution branch produces blocks ξ_τ, each corresponding to $\Delta k =$ `tubelet_size` consecutive states. The number of future evolution blocks is therefore $T^{\mathrm{emb}} = K/\Delta k$, excluding the initial-condition tokens.

Positional Embedding. We use fixed sinusoidal positional encodings.

Transformer and Patch Recovery. The backbone contains L Transformer blocks with pre-normalization, self-attention, MLPs, and residual connections. For AR prediction, only the history available at the current prediction step is included in the input. Attention is bidirectional within this available history; future blocks are excluded from the input sequence, so no triangular causal attention mask is used.

A linear prediction head maps each output token to $C_{\mathrm{patch}} = C_{\mathrm{out}} \Delta k p^2$ values, where C_{out} is the number of predicted physical variables and p is the spatial side length of a reconstructed patch. Convolutional refinement operates on these patch-grid predictions using two Conv3D layers with kernel size $(1, 3, 3)$, unit stride, and padding $(0, 1, 1)$. The channel dimensions follow $C_{\mathrm{patch}} \to C_{\mathrm{refine}} \to C_{\mathrm{patch}}$, with GELU between the two layers. The resulting patch predictions are rearranged into physical states. When predictions are fed back during rollout, they are recovered to physical fields and re-embedded as evolution inputs.

Training Formulations and Gradient Flow. Algorithm 1 summarizes the four formulations at the block level. Local-history AR uses only the preceding predicted block, whereas full-history AR retains the initial condition and the complete predicted history. During AR rollout training, gradients propagate through the predicted feedback without detachment. Teacher forcing replaces predicted histories with ground-truth histories during training; inference uses full-history AR rollout. Full-trajectory prediction uses the embedded initial condition and learnable mask tokens to predict all future blocks jointly in a single forward pass.

Table 7: Model configurations and training batch sizes of the two ViT variants. Both ViT variants are trained on a single GPU without gradient accumulation.

Configuration	ViT-T	Our ViT
Embedding dimension	256	384
Transformer blocks	8	12
Attention heads	4	6
MLP hidden dimension	1024	1536
Convolutional stem width	32	48
Refinement hidden channels	32	32
Spatial patch size ($p \times p$)	16×16	16×16
Temporal tubelet size (Δk)	4	4
Batch size: local-history AR	280	180
Batch size: full-history AR	192	125
Batch size: teacher forcing	192	125
Batch size: full-trajectory	240	140

We use full-history conditioning for teacher-forced ViT training. Applying teacher forcing to the local-history variant would make each prediction after the first depend only on a ground-truth evolution block. Consequently, later prediction losses would no longer propagate to the separately parameterized initial-condition stem and embedding. Full-history conditioning retains a direct dependence on the initial condition and available history at every prediction step. It does not restore gradient paths through previous predictions, which are not used as feedback during teacher-forced training.

ViT configurations. Table 7 summarizes the configurations and batch sizes of our two ViT variants. Both use a spatial patch size of 16×16 and a temporal tubelet size of $\Delta k = 4$. Each Transformer block uses pre-layer normalization with $\epsilon = 10^{-6}$, scaled dot-product multi-head self-attention, and a GELU MLP with an expansion ratio of 4. Query and value biases are enabled, while the key bias is fixed to zero. Dropout and stochastic depth are disabled. For both CE and NS, we use $C_{\text{in}} = C_{\text{out}} = m = 4$, where m is the state-channel dimension used in the main text.

The initial condition and evolution history are processed by separate convolutional stems with independent parameters. Each stem contains two Conv3D–GELU stages, with convolution kernel $(1, 3, 3)$, stride $(1, 2, 2)$, and padding $(0, 1, 1)$. The resulting features are projected into tokens using separate Conv3D layers for the initial condition and evolution blocks. Their kernel sizes and strides are $(1, 4, 4)$ and $(4, 4, 4)$, respectively, accounting for the preceding spatial downsampling. Convolution tuples are ordered as time, height, and width.

The prediction head maps each output token to $C_{\text{patch}} = C_{\text{out}} \Delta k p^2 = 4096$ values, where C_{out} is the number of output channels and $p = 16$. Before reconstruction at the original resolution, these predictions are refined on the spatial patch grid using a Conv3D-GELU-Conv3D module with 32 hidden channels, kernel $(1, 3, 3)$, unit stride, and padding $(0, 1, 1)$.

FNO Training and Inference. Algorithm 2 summarizes the corresponding FNO implementation, with FNO_θ denoting the complete state-to-state model. The initial condition initializes the first input and is not supplied as an additional input at subsequent steps. AR rollout training feeds predictions back without detachment, whereas teacher forcing uses the corresponding ground-truth states. Unlike our ViT, FNO does not use a separately parameterized initial-condition embedding: the same operator processes both the initial state and subsequent states. Both training formulations use AR rollout at inference and the loss defined in Equation 6.

FNO configuration. We implement FNO using `neuraloperator` version 2.0.0. We use the same FNO architecture for AR rollout training and teacher forcing:

- Fourier-mode setting: `n_modes` $= (20, 20)$.
- Fourier layers: 5.
- Hidden channels: 96.

Algorithm 2 FNO prediction under AR rollout training and teacher forcing

Require: Initial condition a; prediction horizon K; formulation f; training or inference mode
Require: Ground-truth states $\boldsymbol{u}_{t_1:t_K}$ for teacher-forced training
Ensure: Predicted future trajectory $\hat{\boldsymbol{u}}_{t_1:t_K}$

$x \leftarrow a$
for $k = 0, \ldots, K-1$ **do**
  $\hat{\boldsymbol{u}}_{t_{k+1}} \leftarrow \mathrm{FNO}_\theta(x)$
  if training **and** $f =$ Teacher forcing **then**
    $x \leftarrow \boldsymbol{u}_{t_{k+1}}$ ▷ Ground-truth feedback
  else
    $x \leftarrow \hat{\boldsymbol{u}}_{t_{k+1}}$ ▷ Predicted feedback; no detachment during training
  end if
end for
return $\hat{\boldsymbol{u}}_{t_1:t_K}$

- Lifting and projection channel ratios: 2, corresponding to hidden widths of 192.
- Mini-batch size: 18 for both training formulations.

Optimization. Both ViT and FNO use AdamW with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, learning-rate warmup over the first 5% of training, and a cosine learning-rate schedule. The minimum learning rate is 10^{-6} for 40-epoch training and 10^{-7} for 80-epoch training. Gradients are clipped to a maximum norm of 1.0. Each configuration is pretrained once.

ViT:

- Initial learning rate: 3×10^{-4}.
- Weight decay: 0.01.
- Batch sizes follow Table 7, with no gradient accumulation.

FNO:

- Initial learning rate: 5×10^{-4}.
- Weight decay: 10^{-6}.
- Mini-batch size: 18, with gradients accumulated over four mini-batches, giving an effective batch size of 72.

We use a fixed base seed and, when training is resumed across cluster jobs, deterministically offset the data-loader seed by the starting epoch. For each fixed 40- or 80-epoch training schedule, we select the checkpoint with the lowest validation loss and evaluate it once on the test sets; OOD test performance is not used for checkpoint selection.

C Dataset Details and Task Selection

We use the datasets publicly released with Poseidon (Herde et al., 2024), downloaded from the PDEgym collection on Hugging Face[2]. Preprocessing, including normalization, and dataset splits follow the authors' released code without modification. We combine the 6 pretraining datasets into a single shuffled dataset without dataset-level balancing or reweighting. Consequently, each dataset contributes training samples in proportion to its size. We use the authors' released preprocessing, including normalization, without modification. The model receives no explicit dataset or PDE identifier and conditions only on the input physical fields and available trajectory history.

We select five OOD tasks that retain the governing equation families, predicted physical fields, and computational domain of the pretraining datasets. These tasks test generalization to different initial-condition distributions and flow dynamics without introducing additional physical fields or changes

[2] https://huggingface.co/collections/camlab-ethz/pdegym

Table 8: Dataset abbreviations used in this work and their counterparts in Poseidon (Herde et al., 2024). Subsection numbers refer to Appendix B of the Poseidon paper.

				Number of trajectories		
Setting	This work	Poseidon	Subsection	Train	Val.	Test
Pretraining	CE-Gauss	CE-Gauss	B.1.6	9,640	120	240
	CE-KH	CE-KH	B.1.5	9,640	120	240
	CE-RP	CE-RP	B.1.3	9,640	120	240
	CE-CRP	CE-CRP	B.1.4	9,640	120	240
	NS-Gauss	NS-Gauss	B.1.2	19,640	120	240
	NS-Sines	NS-Sines	B.1.1	19,640	120	240
OOD evaluation	CE-RM	CE-RM	B.2.8	1,030	100	130
	CE-RKH	CE-RPUI	B.2.7	9,640	120	240
	NS-BB	NS-BB	B.2.2	19,640	120	240
	NS-SL	NS-SL	B.2.3	39,640	120	240
	NS-VS	NS-SVS	B.2.4	19,640	120	240

in domain geometry. This allows us to evaluate the pretrained models directly while keeping their input–output interfaces unchanged. Tasks requiring additional physical fields or geometry-dependent inputs may require model modifications and further training, and are outside the scope of this study.

Spatially distributed vortical flows are represented by Gaussian vorticity initializations (CE-Gauss and NS-Gauss), sinusoidal velocity fields (NS-Sines), and flows derived from Brownian bridges (NS-BB). CE-KH and NS-SL describe the evolution of fluid layers moving at different velocities, while NS-VS follows an initially sinusoidal sheet of concentrated vorticity as it deforms and rolls up. CE-RP and CE-CRP involve interacting waves arising from discontinuities across straight or curved interfaces. CE-RKH uses perturbed interfaces and different initial-state distributions, whereas CE-RM describes shock waves interacting with density interfaces.

Table 8 maps our dataset abbreviations to the names and corresponding appendix subsections in Herde et al. (2024), where full mathematical definitions and simulation details are provided. In particular, our CE-RKH corresponds to CE-RPUI (Riemann problems with uncertain interfaces), and our NS-VS corresponds to NS-SVS (sinusoidal vortex sheet). These naming differences refer to the same released datasets. The train, validation, and test split sizes are also reported in Table 8. We use the training and validation splits only for the six pretraining datasets, while all source-domain and zero-shot OOD results are evaluated on the corresponding test splits. Each trajectory contains one initial condition and $K = 20$ future states on a 128×128 spatial grid.

D Training Cost and Memory Estimation

GPU-hour estimation. Each job on our cluster is limited to 8 hours. For each model, we run an 8-hour training job and estimate the full training cost from the number of iterations completed during this period and the corresponding runtime. We extrapolate this estimate to 40 epochs, except for Poseidon-L, for which we use the published 20-epoch pretraining schedule.

Peak GPU memory. Peak memory is measured during training in the same A100 environment. For Poseidon, we use four times the per-RTX4090 batch size reported in the original paper; increasing this multiplier to five results in out-of-memory errors. The comparatively low GPU-memory usage of our ViT is primarily due to the limited CPU memory available on our cluster, which constrains the input pipeline and prevents us from using larger batch sizes. As a result, our ViT runs remain input-pipeline bound and do not saturate GPU memory, indicating additional throughput headroom through systems optimization.

The relatively low peak memory usage of our model is mainly due to CPU and RAM constraints rather than GPU memory limitations. Using a larger batch size requires more host memory for data loading and prefetching, which can cause RAM out-of-memory errors or increase the risk of job preemption on our cluster. We therefore use a smaller batch size and limited data-loader prefetching.

Table 9: Training-resource measurements under our A100 setup. Total GPU-hours are estimated for the listed training schedules. Peak memory is measured using the run-specific batch size of each model and is reported as a diagnostic rather than as a fixed-batch memory-efficiency comparison.

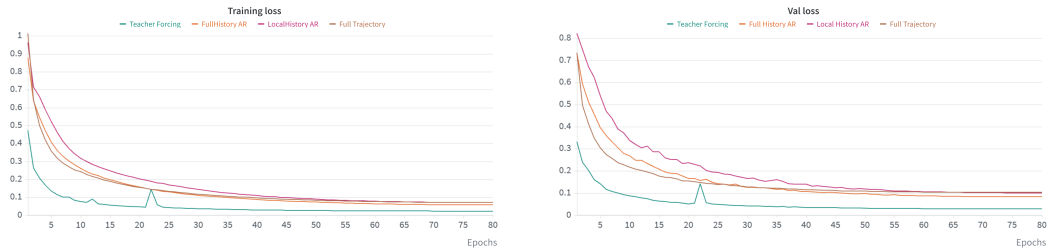
Model	Epochs	GPU-h/epoch	Total GPU-hours	Peak memory (%)
Poseidon-L	20	≈ 15.4	≈ 307.5	70.77
Poseidon-B	40	≈ 6.8	≈ 271.0	99.58
Poseidon-T	40	≈ 2.1	≈ 82.8	79.46
Our ViT	40	≈ 0.29	≈ 11.7	41.49

As a result, the A100 sometimes waits for data loading, and we observe substantially lower GPU utilization for our model than for Poseidon. Despite this limitation, our model still requires much less estimated training time while achieving the zero-shot OOD improvements reported in Table 1.

E Training Curves

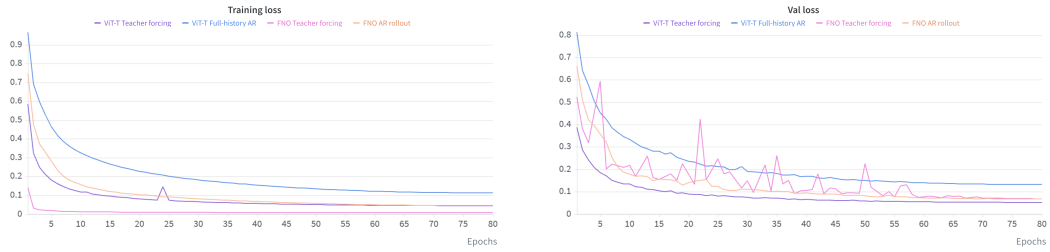
Figure 5 shows the training and validation losses of the 25M ViT under the four training formulations. All formulations exhibit sustained decreases in both losses and reach relatively stable regimes toward the end of pretraining. Teacher forcing achieves the lowest final training and validation losses, consistent with its stronger source-domain prediction performance. The transient increase near epoch 22 quickly recovers, and none of the formulations exhibits persistent instability or a growing gap between training and validation loss.

Figure 6 reports the corresponding curves for ViT-T and FNO under AR rollout training and teacher forcing. Both backbones converge under both formulations. The validation loss of teacher-forced FNO is more variable during the earlier epochs but stabilizes as training proceeds. These curves provide no evidence that the backbone-dependent OOD results arise from a failure to optimize either training formulation.



(a) Training loss. (b) Validation loss.

Figure 5: Training and validation loss curves for 80-epoch pretraining of different formulations.



(a) Training loss. (b) Validation loss.

Figure 6: Training and validation loss curves for 80-epoch pretraining of ViT-T and FNO under AR rollout and teacher-forced training.

F Source-Domain Prediction Performance

We report the source-domain prediction results corresponding to the OOD comparisons in Section 5. All models in this section are trained for 80 epochs.

F.1 25M ViT source-domain results

Table 10 compares the four training formulations for the 25M ViT. Full-history AR improves upon local-history AR across all six datasets, while teacher forcing generally achieves the lowest error. In contrast, full-trajectory prediction yields mixed results relative to local-history AR. Table 11 reports the relative L_2 error.

F.2 10M ViT-T vs FNO source-domain results

Table 12 compares approximately parameter-matched ViT-T and FNO models. Teacher forcing improves four of the six source-domain datasets for ViT-T and all six for FNO in relative L_1 error, with the clearest and most consistent gains occurring for FNO. These consistent source-domain improvements contrast with the dynamics-dependent effects observed in the OOD comparisons. Table 13 reports the relative L_2 error.

Table 10: Source-domain prediction performance of the 25M ViT† under different training formulations. Lower is better.

Training formulation	CE-Gauss	CE-KH	CE-RP	CE-CRP	NS-Gauss	NS-Sines
Full-trajectory	0.2257	0.2527	0.1406	0.4914	0.1154	0.3728
Local-history AR	0.1878	0.2779	0.1710	0.4865	0.0999	0.3200
Full-history AR	0.1704	0.2587	0.1507	0.4658	0.0831	0.2743
Teacher forcing	**0.1219**	**0.2495**	**0.1239**	**0.4244**	**0.0692**	**0.2559**

Table 11: Same source-domain comparison as Table 10, but reporting relative L_2 error. Lower is better.

Training formulation	CE-Gauss	CE-KH	CE-RP	CE-CRP	NS-Gauss	NS-Sines
Full-trajectory	0.2738	0.3668	0.1686	0.5453	0.1394	0.4747
Local-history AR	0.2289	0.3870	0.2000	0.5425	0.1138	0.3967
Full-history AR	0.2087	**0.3664**	0.1777	0.5243	0.0957	**0.3444**
Teacher forcing	**0.1557**	0.3841	**0.1564**	**0.4947**	**0.0912**	0.3449

Table 12: Source-domain prediction performance of approximately parameter-matched ViT-T† (10M) and FNO† (10M) models. Lower is better.

Model	Formulation	CE-Gauss	CE-KH	CE-RP	CE-CRP	NS-Gauss	NS-Sines
ViT-T†	Full-history AR	0.2461	**0.3180**	0.2206	0.5431	**0.1445**	0.4271
ViT-T†	Teacher forcing	**0.2126**	0.3296	**0.2097**	**0.5165**	0.1627	**0.4223**
FNO†	AR rollout	0.1214	0.2661	0.1573	0.4170	0.0527	0.2529
FNO†	Teacher forcing	**0.0829**	**0.2391**	**0.1470**	**0.4139**	**0.0370**	**0.1902**

Table 13: Same source-domain comparison as Table 12, but reporting relative L_2 error. Bold indicates the better formulation within each model family. Lower is better.

Model	Formulation	CE-Gauss	CE-KH	CE-RP	CE-CRP	NS-Gauss	NS-Sines
ViT-T†	Full-history AR	0.2920	**0.4234**	0.2449	0.5947	**0.1591**	**0.5125**
ViT-T†	Teacher forcing	**0.2591**	0.4723	**0.2435**	**0.5749**	0.2004	0.5409
FNO†	AR rollout	0.1519	**0.3643**	**0.1795**	**0.4788**	0.0623	0.3245
FNO†	Teacher forcing	**0.1090**	0.4056	0.1940	0.5008	**0.0523**	**0.2663**

G Additional Visualization

This section provides additional qualitative comparisons between our 25M ViT after 40 epochs of teacher-forced pretraining and the 629M Poseidon-L under direct prediction. Both models predict unseen dynamics from a single initial condition without downstream finetuning, contextual demonstrations, or test-time adaptation. As shown in Table 6, our ViT achieves lower error on three of the five OOD systems.

The largest improvements occur on NS-BB and NS-VS. On NS-BB, our ViT reduces the error from 0.2576 to 0.0666, corresponding to a factor of 3.87. On NS-VS, the error decreases from 0.7152 to 0.2561, corresponding to a factor of 2.79. Our model also reduces the CE-RM error from 1.2821 to 1.1304, although both models remain inaccurate on this system. Poseidon-L performs better on CE-RKH, with errors of 0.5809 and 0.6746 for Poseidon-L and our ViT, respectively. NS-SL remains the clearest exception: Poseidon-L achieves an error of 0.1694, whereas our ViT obtains 0.4450.

The qualitative comparisons across the three NS systems further contextualize these results. On NS-BB, our ViT preserves the spatial organization and motion of the dominant structures, whereas Poseidon-L becomes progressively diffuse and loses much of the velocity variation at later times (Figure 3). On NS-VS, our ViT captures the dominant counterclockwise deformation and large-scale roll-up, while Poseidon-L increasingly departs from the ground-truth evolution; however, our prediction does not fully recover the finer vortical structures at later times (Figure 8). On NS-SL, where Poseidon-L is substantially more accurate, our ViT still captures the progressive deformation of the two shear layers but underestimates their roll-up and produces more diffuse and spatially misaligned structures (Figure 7). Complete rollout animations for all five OOD systems, including multi-field visualizations of CE-RM and CE-RKH, are provided in the supplementary material.

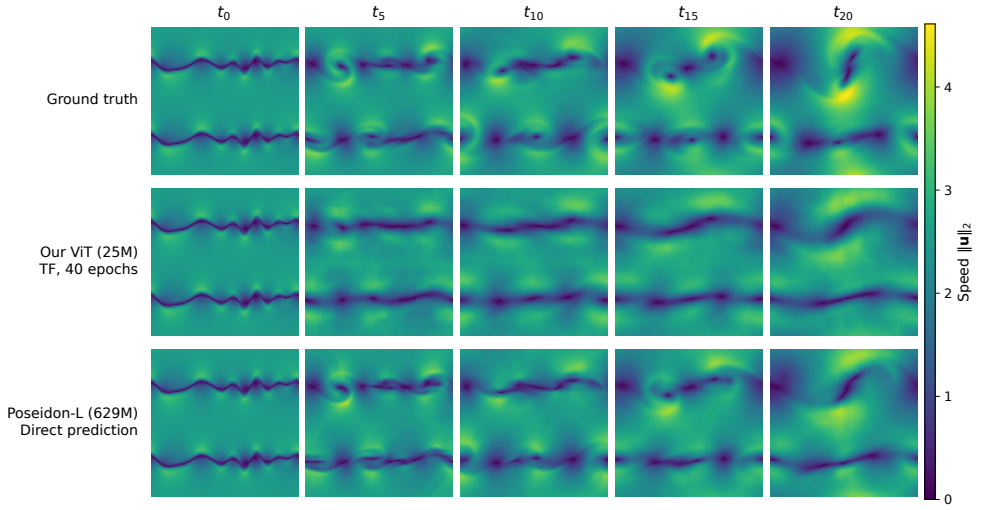


Figure 7: Visualization of NS-SL evolution using our ViT pretrained under teacher forcing (TF) and Poseidon-L with $a \mapsto \hat{\boldsymbol{u}}_{t_k}$ prediction, without any finetuning, contextual demonstrations, or test-time adaptation.

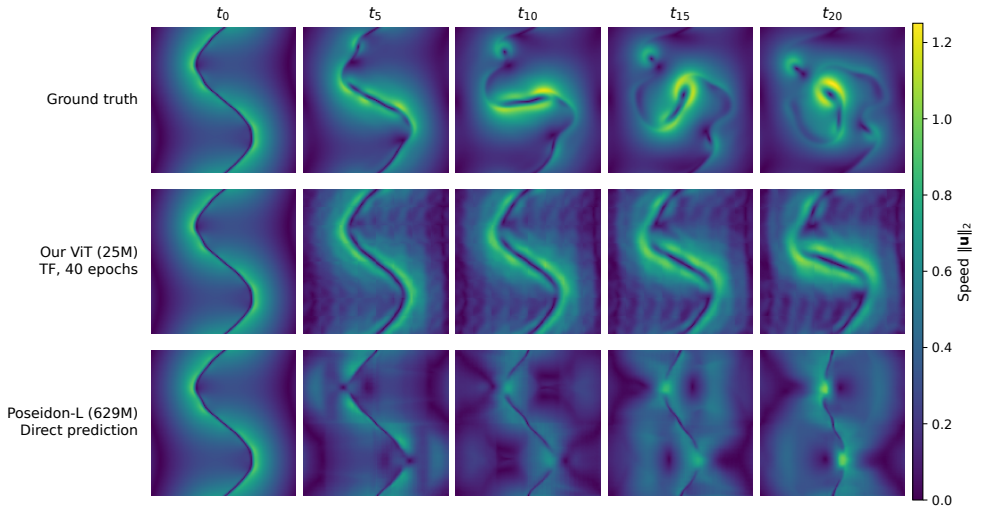


Figure 8: Visualization of NS-VS evolution using our ViT pretrained under teacher forcing (TF) and Poseidon-L with $a \mapsto \hat{\boldsymbol{u}}_{t_k}$ prediction. Both models perform zero-shot OOD prediction without any finetuning, contextual demonstrations, or test-time adaptation.

H FNO VISUALIZATION AND OOD PREDICTION

Having shown the strong zero-shot OOD performance of our teacher-forced ViT in Table 6 and the accompanying evolution visualizations, we now provide an analogous quantitative and qualitative comparison for FNO. Specifically, we compare the 10M FNO after 40 epochs of teacher-forced pretraining with the 629M Poseidon-L under direct prediction. FNO generates the trajectory through AR rollout, whereas Poseidon-L directly predicts each future state from the initial condition. Both models receive only a single initial condition, without any downstream finetuning, contextual demonstrations, or test-time adaptation.

As shown in Table 14, FNO achieves lower relative L_1 error on three of the five OOD systems. On NS-BB, it reduces the error from 0.2576 to 0.0354, corresponding to a factor of 7.28. On NS-VS, the error decreases from 0.7152 to 0.2368, corresponding to a factor of 3.02. FNO also improves CE-RM from 1.2821 to 1.1610, although both models remain inaccurate on this system. Poseidon-L performs better on CE-RKH and NS-SL.

The qualitative comparisons are consistent with the numerical results. On NS-BB, FNO closely preserves the spatial organization, motion, and velocity variation of the ground-truth trajectory, whereas Poseidon-L becomes increasingly diffuse at later times (Figure 9). On NS-VS, FNO captures the dominant deformation and global rotational evolution of the vortex sheet, although it does not fully recover the finer rolled-up structures. Poseidon-L instead departs substantially from the ground-truth evolution (Figure 11). NS-SL remains the main exception: Poseidon-L more accurately follows the formation of the larger vortical structures, while FNO preserves the two-layer organization but underestimates the subsequent roll-up and produces more diffuse predictions (Figure 10). Together with the results in Section 5, these comparisons show that strong zero-shot OOD prediction after teacher-forced pretraining is not exclusive to the ViT backbone.

Table 14: Zero-shot OOD comparison between the 10M FNO after 40 epochs of teacher-forced pretraining and Poseidon-L‡ under direct prediction. The error ratio is Poseidon-L‡ / FNO; values above one favor FNO. Lower is better. Corresponding relative L_2 results are reported in Table 15.

Model	CE-RM	CE-RKH	NS-BB	NS-SL	NS-VS
FNO (teacher forcing)	**1.1610**	0.7611	**0.0354**	0.4589	**0.2368**
Poseidon-L‡	1.2821	**0.5809**	0.2576	**0.1694**	0.7152
Poseidon-L‡ / FNO	1.10×	0.76×	7.28×	0.37×	3.02×

Table 15: Same FNO comparison as Table 14, but reporting relative L_2 error. The error ratio is Poseidon-L‡ / FNO; values above one favor FNO. Lower is better.

Model	CE-RM	CE-RKH	NS-BB	NS-SL	NS-VS
FNO (teacher forcing)	**1.1636**	0.7224	**0.0450**	0.5444	**0.3610**
Poseidon-L‡	1.2625	**0.5888**	0.3456	**0.1942**	0.7707
Poseidon-L‡ / FNO	1.08×	0.82×	7.68×	0.36×	2.13×

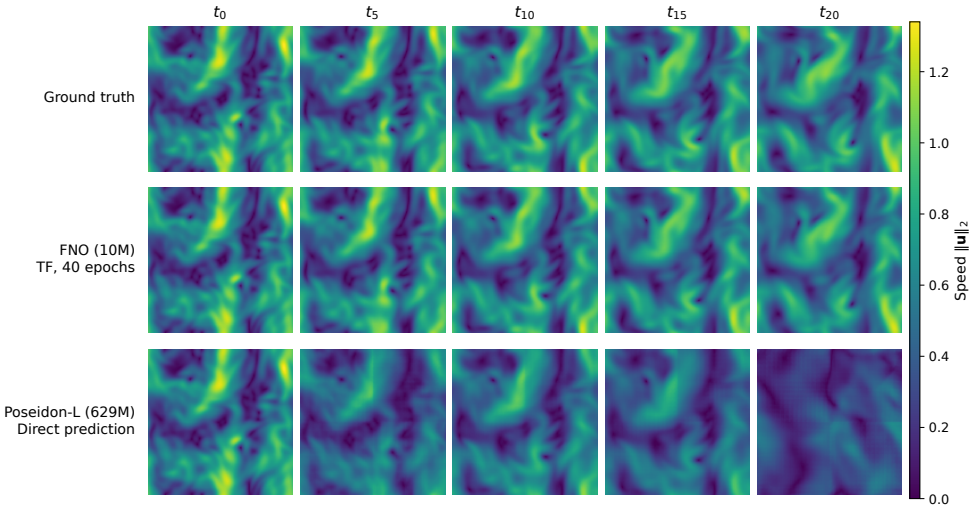


Figure 9: Zero-shot OOD speed evolution on NS-BB. FNO is pretrained for 40 epochs with teacher forcing and evaluated through AR rollout, while Poseidon-L uses direct prediction $a \mapsto \hat{\boldsymbol{u}}_{t_k}$.

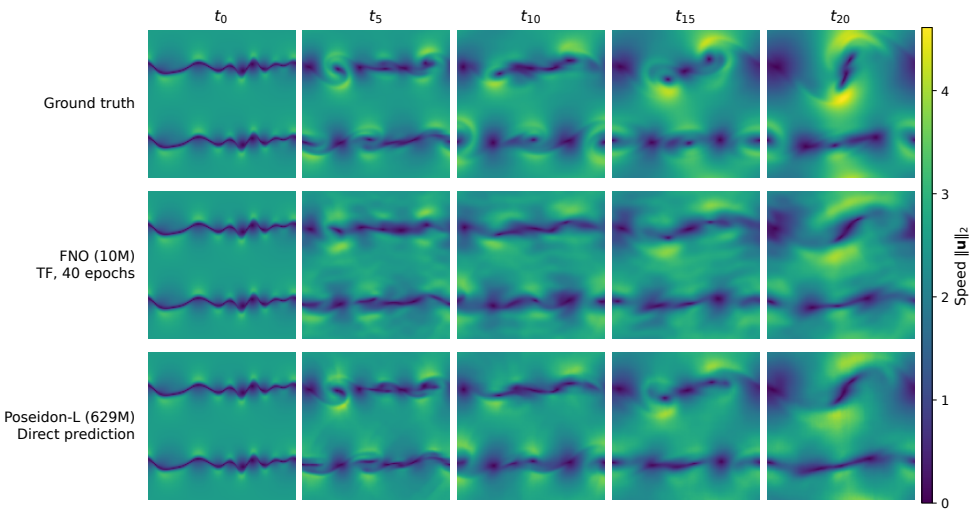


Figure 10: Zero-shot OOD speed evolution on NS-SL. FNO is pretrained for 40 epochs with teacher forcing and evaluated through AR rollout, while Poseidon-L uses direct prediction $a \mapsto \hat{\boldsymbol{u}}_{t_k}$.

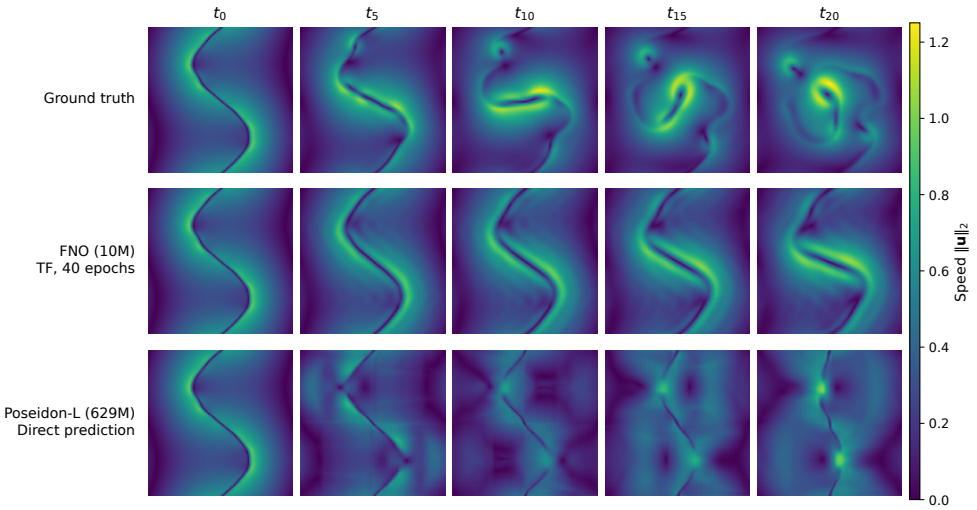


Figure 11: Zero-shot OOD speed evolution on NS-VS. FNO is pretrained for 40 epochs with teacher forcing and evaluated through AR rollout, while Poseidon-L uses direct prediction $a \mapsto \hat{\boldsymbol{u}}_{t_k}$.

I RESULTS OF RELATIVE L_2 ERROR

We report the relative L_2 counterparts of all relative L_1 comparisons in the main text. The relative L_2 results support the same principal conclusions, although individual effects are more metric-dependent. In particular, teacher forcing preserves strong FNO gains on NS-BB and NS-VS but amplifies the regression on CE-RKH, where the error increases from 0.7554 to 1.7229.

Table 16: Same comparison as Table 1, but reporting relative L_2 error. Lower is better.

Model	CE-RM	CE-RKH	NS-BB	NS-SL	NS-VS
Poseidon-L	**1.1374**	0.9238	1.9087	0.7803	1.8582
Poseidon-B	1.4341	0.6980	1.9208	0.6600	2.3386
Poseidon-T	2.7371	0.8278	4.4731	0.6164	4.9047
Our ViT	1.2096	**0.6323**	**0.1480**	**0.4441**	**0.3857**

Table 17: Same source-domain comparison as Table 2, but reporting relative L_2 error. The symbols retain the same meanings. Lower is better.

Model	CE-Gauss	CE-KH	CE-RP	CE-CRP	NS-Gauss	NS-Sines
Poseidon-L	0.9493	2.1321	0.6384	0.9003	0.5907	0.5603
Poseidon-B	0.7446	1.3142	0.8383	1.0018	0.2121	0.3823
Poseidon-T	0.8320	4.4619	1.3953	1.0521	1.4153	0.5126
Our ViT	0.2792	0.4369	0.2489	0.5993	0.1513	0.4743
Our ViT†	0.1878	0.3870	0.2000	0.5425	0.1138	0.3967
Poseidon-L‡	**0.0942**	**0.1993**	**0.0660**	**0.3446**	**0.0282**	**0.0730**

Table 18: Same pretraining-duration comparison as Table 3, but reporting relative L_2 error. The symbols retain the same meanings. Lower is better.

Model	CE-RM	CE-RKH	NS-BB	NS-SL	NS-VS
Our ViT†	**1.1642**	0.7561	**0.1171**	0.4876	**0.3540**
Our ViT	1.2096	0.6323	0.1480	0.4441	0.3857
Poseidon-L‡	1.2625	**0.5888**	0.3456	**0.1942**	0.7707

Table 19: Same training-formulation comparison as Table 4, but reporting relative L_2 error. Lower is better.

Training formulation	CE-RM	CE-RKH	NS-BB	NS-SL	NS-VS
Full-trajectory	1.1538	0.7999	0.3210	**0.4679**	0.4311
Local-history AR	1.1642	0.7561	0.1171	0.4876	0.3540
Full-history AR	**1.1317**	0.6803	**0.0989**	0.5229	0.3658
Teacher forcing	1.1911	**0.6030**	0.1134	0.5198	**0.3449**

Table 20: Same ViT-T and FNO comparison as Table 5, but reporting relative L_2 error. Bold indicates the better formulation within each model family. Lower is better.

Model	Training formulation	CE-RM	CE-RKH	NS-BB	NS-SL	NS-VS
ViT-T† (10M)	Full-history AR	1.1445	0.6712	0.1423	0.4991	**0.3763**
ViT-T† (10M)	Teacher forcing	**1.1144**	**0.6150**	**0.1289**	**0.4586**	0.3765
FNO† (10M)	AR rollout	1.2520	**0.7554**	0.7883	**0.5101**	0.6542
FNO† (10M)	Teacher forcing	**1.1418**	1.7229	**0.0401**	0.5954	**0.3651**

Table 21: Same final comparison as Table 6, but reporting relative L_2 error. The error ratio is Poseidon-L‡ / Our ViT; values above one favor our ViT. Lower is better.

Model	CE-RM	CE-RKH	NS-BB	NS-SL	NS-VS
Our ViT (teacher forcing)	**1.1408**	0.6169	**0.0752**	0.5026	**0.3693**
Poseidon-L‡	1.2625	**0.5888**	0.3456	**0.1942**	0.7707
Poseidon-L‡ / Our ViT	1.11×	0.95×	4.60×	0.39×	2.09×