OPERATOR FLOW MATCHING FOR TIMESERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Forecasting high-dimensional, PDE-governed dynamics remains a core challenge for generative modeling. Existing autoregressive and diffusion-based approaches often suffer cumulative errors and discretisation artifacts that limit long, physically consistent forecasts. Flow matching offers a natural alternative, enabling efficient, deterministic sampling. We prove an upper bound on FNO approximation error and propose TempO, a latent flow matching model leveraging sparse conditioning with channel folding to efficiently process 3D spatiotemporal fields using time-conditioned Fourier layers to capture multi-scale modes with high fidelity. TempO outperforms state-of-the-art baselines across three benchmark PDE datasets, and spectral analysis further demonstrates superior recovery of multi-scale dynamics, while efficiency studies highlight its parameter- and memory-light design compared to attention-based or convolutional regressors.

1 INTRODUCTION

Generative artificial intelligence has brought unparalleled creative and scientific potential, with models capable of producing images (Hatamizadeh et al., 2025), video (Bar-Tal et al., 2024), audio (Ju et al., 2024), and text (Grattafiori et al., 2024) that rival human quality. From autoregressive transformers to diffusion models and energy-based approaches, the landscape of generative AI is rich and diverse, offering multiple pathways to model complex data distributions. At the core of this revolution are probabilistic generative models, which learn to sample from complex, high-dimensional distributions. Among these, flow matching models have emerged as a class of generative models which learn to transform a simple prior distribution to a more complex data distribution as a continuous transformation. This direct, simulation-free approach enables both efficiency and precision, offering a new lens on modeling complex systems (Lipman et al., 2023).

Despite recent advances, forecasting high-dimensional temporal dynamics remains challenging. Deep learning models are computationally expensive and often fail catastrophically after a few dozen timesteps due to compounding errors in autoregressive predictions (Ansari et al., 2024). Even with the advent of large language models and their remarkable ability to generate, models that attempt to leverage them for forecasting face limitations of discretisation and tokenisation (Ansari et al., 2024), offering little practical benefit relative to their computational cost (Tan et al., 2024). Modern generative models have been proven capable of generating visually compelling and coherent videos (John et al., 2024), but critically lack the fine-grained control required to be used in scientific and engineering contexts.

Recent foundation models for forecasting include GenCast for weather (Price et al., 2025) and Chronos for general time series (Ansari et al., 2024), demonstrate the promise of large-scale pretraining. These models leverage massive datasets across multiple domains resulting in strong zero- and few-shot transferability. Chronos captures coarse, long-range correlations remarkably long timespans; however, the granularity, i.e. prediction length still falls at an average of 22 across 55 datasets, with only 7 tasks exceeding 30 steps (Ansari et al., 2024). GenCast, likewise, can generate 15-day global weather forecasts, but at a granularity of 12 hours, around 30 steps. True progress requires models capable of deterministic yet flexible generation, able to explore plausible trajectories while respecting physical constraints to then select precise forecasts out of the space of plausible predictions (Guo et al., 2025). Although the short to mid term range is a popular horizon to explore (Lim

et al., 2025), the goal is to generate long-horizon predictions on the order of 30 timesteps or more, generating trajectories that are not just plausible, but physically consistent.

Fundamentally, models relying on discretisation or tokenisation are not ideal for continuous, Partial Differential Equation (PDE)-governed dynamics. Demonstrating smooth trajectories in state space which generalise to long forecasting horizons would show greater fidelity to the underlying physics. Other existing efforts which leverage diffusion (Molinaro et al., 2025; Yao et al., 2025; Huang et al., 2024) move toward more natural representations, but are themselves fundamentally tied to stochastic dynamics. Instead, a natural choice for such a problem is flow matching, where the vector field regression is closer to learning PDE operators which are themselves vector fields describing time-derivatives, and learn deterministic dynamics with potentially more efficient Ordinary Differential Equation (ODE)-based sampling in contrast to the denoising process of diffusion models. Existing flow matching methods have individually worked toward video generation (Davtyan et al., 2023; Jin et al., 2025) and PDE single-step prediction (Kerrigan et al., 2023), but thus far have not been thoroughly tested for long-horizon temporal forecasting and do not design for the deterministic and stable rollouts required for such tasks.

In this work, we propose Temporal Operator flow matching (TempO), a latent flow matching capable of forecasting physically meaningful fields over long time horizons with high fidelity in both spatial and spectral characteristics. We perform sparse conditioning for added computational- and data-efficiency, and channel folding to process spatiotemporal 3D data using conventionally 2D frameworks: We leverage recent advances in scientific machine learning by designing time-conditioned parameter-efficient shared Fourier layers within the vector field regressor, allowing for strong capture of global and local spatial modes. We derive theoretical error bounds that characterize the efficiency and expressivity of TempO, and showcase its performance on PDE benchmarking datasets accompanied with a spectral analysis showing a distinct advantage in capturing the essential dynamics required for forecasting. We see a 16% lower error when predicting vorticity of 2D incompressible Navier Stokes, with Pearson correlations remaining above 0.98 for a 40 step forecasting horizon, demonstrating its stable temporal forecasting and high quality generation capability.

2 Relevant Works

Interest in machine learning for physical systems has surged, with generative models being adapted for such tasks and borrowing features for broader generation. For example, Liu & Tang (2025); Li et al. (2021) integrate an Fourier Neural Operator (FNO) into a score-matching denoising network, leveraging its resolution-invariant properties to achieve state-of-the-art superresolution. Similarly, Fourier Neural ODE combines Fourier analysis with Neural ODEs, outperforming the original FNO, DeepONet (Lu et al., 2021), and Physics Informed Neural Networks (Raissi et al., 2019) for predicting time instances (Li et al., 2024). Operator learning has also been integrated with generative adversarial models to generalize to infinite-dimensional function spaces (Rahman et al., 2022). However, such approaches leverage desirable representation characteristics of Fourier embedded processing, which diverges from the focus of this work on spatio-temporal generation.

Application-specific models for scientific data have also seen development: GenCFD (Molinaro et al., 2025) proposes a conditional diffusion model to generate the underlying distributions of high fidelity flow fields. Kerrigan et al. (2023) propose the first extension of FNOs to flow matching tasks and predicts plausible fluid dynamic fields. Yao et al. (2025) leverages neural operators in an unconditional diffusion model to improve efficiency and sees state-of-the-art performance for multi-resolution PDE tasks, as compared to its competitor DiffusionPDE Huang et al. (2024) which originally demonstrated strong performance in solving PDEs with partial observations. Such methods have thus far focused on single-timeframe prediction, i.e., solving slices of 2D dynamic PDEs, rather than temporal rollouts as investigated here.

Models designed to predict sequences of future states include the aforementioned large-scale Chronos and GenCast (Ansari et al., 2024; Price et al., 2025). In addition, pyramidal flow matching (Jin et al., 2025) produces state-of-the-art video generation compared to leading models (**?**), representing a successful flow matching foundation model. (Tamir et al., 2024) present conditional flow matching for time series, succeeding in long 1D trajectories where neural ODEs fail, but has not scaled to 2D spatiotemporal data. Kollovieh et al. (2024) extends this with Gaussian processes

for forecasting tasks outside of scientific machine learning. We focus instead on models that fall between these two categories, scaling reasonably to 2D data to match common PDE settings.

3 METHOD

We begin by developing the background which is then used to construct our method. Flow matching learns a time-dependent velocity field $v_\theta(z, t)$ defining an ODE in the latent space:

$$\frac{dz(t)}{dt} = v_\theta(z(t), t), \quad z(0) \sim \pi_0, \tag{1}$$

where π_0 is a simple prior (e.g., Gaussian). Integrating this ODE transports samples to the latent data distribution π_1, see Appendix B. Training reduces to a regression objective that matches the model velocity field to a target velocity along interpolating probability paths (Lipman et al., 2023). This enables deterministic, simulation-free sampling from complex distributions.

Table 1: Representative Path Choices in Flow Matching Models.

Path	a_t	b_t	c_t	Parameter definitions
Affine-OT[1]	t	0	$(1-(1-\epsilon_{\min})t)^2$	$\epsilon_{\min} \geq 0$: min. noise level
RIVER[2]	$(1-(1-\sigma_{\min})t)$	t	σ^2	$\sigma \geq 0$: noise scale, $\sigma_{\min} \geq 0$: min. noise
SLP[3]	$(1-t)$	t	$\sigma_{\min}^2 + \sigma^2 t(1-t)$	$\sigma, \sigma_{\min} \geq 0$: variance parameters
VE-diff[4]	1	0	σ_t^2	σ_t: geometric schedule, $\sigma_{\min}, \sigma_{\max} > 0$
VP-diff[4]	$\exp(-\frac{1}{2}T(1-t))$	0	$1-\exp(-T(1-t))$	$\beta_{\min}, \beta_{\max} > 0, T(t) = \int_0^t \beta(s)\, ds$

[1] (Lipman et al., 2023), [2] (Davtyan et al., 2023), [3](Lim et al., 2025),[4] (Ryzhakov et al., 2024)

A key component of flow matching is the choice of the probability density path p_t interpolating between the reference distribution π_0 and the target π_1. We focus on Gaussian conditional paths with closed-form velocity fields:

$$p_t\Big(Z \mid \tilde{Z} := (Z_0, Z_1)\Big) = \mathcal{N}\Big(Z \,\Big|\, a_t Z_0 + b_t Z_1,\ c_t^2 I\Big),$$

where a_t, b_t, c_t define the path (Table 1). This pair-conditional path is defined for a specific transition (Z_0, Z_1), and the marginal interpolant is obtained by averaging over all pairs: $p_t(Z) = \mathbb{E}_{(Z_0,Z_1)}[p_t(Z \mid Z_0, Z_1)]$. While π_0 is typically a standard Gaussian, intermediate densities p_t can follow diffusion-inspired, optimal transport, or other custom schedules.

To parameterize v_θ, we modify FNOs, which approximate mappings between functions via spectral convolution layers. Given input u, the FNO parameterizes an operator as $\mathcal{G}_\theta : u \mapsto \tilde{u}, \quad \tilde{u} : \mathcal{D} \to \mathbb{R}^{c_{\text{out}}}$, that maps u to an output function $\tilde{u}$. Iterative Fourier layers perform spectral transformations of the input $\hat{u}(k) = \mathcal{F}[u](k), \quad \hat{\tilde{u}}(k) = R_\theta(k) \cdot \hat{u}(k)$, followed by an inverse Fourier transform back to the spatial domain; $\tilde{u}(x) = \mathcal{F}^{-1}[\hat{\tilde{u}}](x)$, with $R_\theta(k)$ being learnable Fourier-mode weights and $\mathcal{F}$ denoting the Fourier transform. This spectral representation allows the FNO to efficiently capture long-range dependencies and global correlations in the data.

3.1 TEMPORAL OPERATOR FLOW MATCHING (TEMPO)

Using an FNO-inspired regressor to learn the vector field of a flow matching model has a number of benefits, namely, the added expressivity that the Fourier representation provides at a low computational cost thanks to highly optimised Fast Fourier Transform (FFT) operations. Building on prior analysis of FNOs for solving PDEs (Kovachki et al., 2021), we show that an FNO-inspired regressor can achieve an upper bound on approximation error for flow matching models and we provide a lower bound on the accuracy achievable by sampler-based methods (e.g., Transformer or U-Net) in relation to their number of parameters.

Theorem 3.1 (FNO regressor constructive upper bound)**.** *Let $\mathbb{T}^d$ be the d–torus. Fix $s, s' \geq 0$ and let $\mathcal{U} \subset H^s(\mathbb{T}^d)$ be compact. Suppose $\mathcal{G} : \mathcal{U} \to H^{s'}(\mathbb{T}^d)$ is continuous and satisfies $|\widehat{\mathcal{G}(u)}(k)| \leq C_\lambda(1+|k|)^{-p}$ for all $u \in \mathcal{U}$, $k \in \mathbb{Z}^d$, with constants $C_\lambda > 0$, $p > 0$. If $p > s' + \frac{d}{2}$ and we define $\alpha := p - s' - \frac{d}{2} > 0$, then for every $\varepsilon > 0$ there exists a Fourier Neural Operator $\mathcal{G}_\theta$ with*

$$P_{FNO}(\varepsilon) \lesssim \varepsilon^{-d/\alpha},$$

such that $\sup_{u\in\mathcal{U}} \|\mathcal{G}(u) - \mathcal{G}_\theta(u)\|_{H^{s'}} \le \varepsilon$. *The hidden constants depend only on* $d, s, s', \mathcal{U}, C_\lambda$ *and mild/logarithmic factors.*

This result is in line with the estimates and arguments made in (Kovachki et al., 2021).

Sketch of proof of Theorem 3.1. (Spectral truncation.) The Fourier decay assumption implies that high-frequency modes of $\mathcal{G}(u)$ contribute at most $O(K^{-2\alpha})$ to the $H^{s'}$-error. Choosing $K \asymp \varepsilon^{-1/\alpha}$ makes this truncation error $\le \varepsilon/2$.

(Finite-dimensional reduction.) For this cutoff K, the operator $\mathcal{G}_K$ is determined by $O(K^d)$ Fourier coefficients, and inputs can likewise be restricted to finitely many low modes without significant loss of accuracy. Thus the problem reduces to approximating a continuous map between compact subsets of $\mathbb{R}^{m_{\text{in}}}$ and $\mathbb{R}^{m_{\text{out}}}$, with $m_{\text{out}} \asymp K^d$.

(Approximation by networks.) Standard universal approximation results (or the constructive FNO design in (Kovachki et al., 2021)) ensure that such a finite map can be uniformly approximated by a network with $O(K^d)$ parameters, up to mild logarithmic factors.

(Conclusion.) Combining these errors yields an overall accuracy ε with parameter count $P \lesssim K^d \asymp \varepsilon^{-d/\alpha}$, proving the claim. □

Proposition 3.2 (Transformer/UNet Sampler-based lower bound)**.** *Under the assumptions of Theorem 3.1, consider any learner that observes each* $u \in \mathcal{U}$ *only through* n *fixed point evaluations and applies a parametric map with* P *real parameters, required in the worst case to reconstruct all Fourier modes up to radius* $K \asymp \varepsilon^{-1/\alpha}$. *Then necessarily*

$$n \gtrsim \varepsilon^{-d/\alpha}, \qquad P_{\text{sampler}}(\varepsilon) \gtrsim \varepsilon^{-\beta d/\alpha},$$

for some architecture–dependent $\beta \ge 1$ *(optimistically* $\beta = 1$ *when only diagonal mode-wise maps are needed, generically* $\beta = 2$ *for arbitrary dense linear maps). These bounds are information-theoretic and asymptotic, up to constants and mild/logarithmic factors.*

Sketch of proof of Proposition 3.2. (Sampling necessity.) The K–mode subspace V_K has dimension $D_K \asymp K^d$. Sampling at n points defines a linear map $S : V_K \to \mathbb{C}^n$. For S to be injective on V_K, its matrix must have rank D_K, hence $n \ge D_K \asymp K^d$.

(Parameter complexity.) After sampling, the learner implements a parametric map $M : \mathbb{C}^n \to \mathbb{C}^m$. To represent arbitrary linear maps on the D_K-dimensional coefficient space (e.g. arbitrary diagonal multipliers), the parameter family must have at least $P \gtrsim D_K$ degrees of freedom. For fully general dense linear maps one needs $P \gtrsim D_K^2$.

(Conversion.) Substituting $K \asymp \varepsilon^{-1/\alpha}$ (from the theorem) gives $n \gtrsim \varepsilon^{-d/\alpha}$ and $P \gtrsim \varepsilon^{-\beta d/\alpha}$ with $\beta = 1$ (optimistic) or $\beta = 2$ (dense case), establishing the lower bound, see Appendix A for the extended proof. □

Corollary 3.3 (FNO vs sampler scaling)**.** *From Theorem 3.1 and Proposition 3.2 one has*

$$P_{\text{FNO}}(\varepsilon) \lesssim \varepsilon^{-d/\alpha}, \qquad P_{\text{sampler}}(\varepsilon) \gtrsim \varepsilon^{-\beta d/\alpha}.$$

Hence, whenever $\beta > 1$, *FNOs achieve the same accuracy* ε *with asymptotically fewer parameters than sampler–based learners.*

TempO Consequently, we propose a novel generation model which capitalises on the FNO's expressivity and capacity to model complex velocity fields by designing a latent time-conditioned FNO vector field regressor using channel folding for both efficiency and enhanced temporal coherency. Together with temporal conditioning (Davtyan et al., 2023), these define a novel, end-to-end trainable model for predicting latent dynamics.

Let $f_\phi : \mathbb{R}^X \to \mathbb{R}^Z$ denote an encoder mapping data points x to latent embeddings $z = f_\phi(x)$. We can then define a latent-space velocity field described by 1 where v_θ is parameterized by an FNO.

To capture the temporal dependencies, we leverage *sparse conditioning* (Davtyan et al., 2023; Lim et al., 2025). For some discrete-time sequence $\{x_t\}_{t=1}^N$ with $x_t \in \mathcal{X}$, its latent representation

is given by $\{z_t\}_{t=1}^{N}$, where $z_t = f_\phi(x_t)$. For a prediction horizon $T \in \{L, \ldots, N-1\}$ with sequence length L, the objective is to predict the next latent embedding z_{T+1}. We define a reference embedding to be z_T, corresponding to the most recent observation prior to the prediction target, and a conditioning embedding as some observation selected at a timestep $\tau \in \{T-L, \ldots, T-1\}$. These two embeddings are concatenated with the temporal offset, defined as $\Delta = T-\tau$, which is the extent of temporal data the model is provided to predict the next-step embedding, $\hat{z}_{T+1} = f_\theta(z_T, z_\tau, \Delta)$.

To process the spatiotemporal input data and conditioning while preserving compatibility the 2D FNO, we then propose a *channel folding* scheme that merges the batch and channel dimensions (as opposed to the more conventional batch and time dimensions) to align with the original input ordering of the FNO). To match with the expected inputs of the form $\mathbb{R}^{B' \times T' \times H \times W}$, we collapse the batch and channel axes into a single "effective batch" dimension $u' \in \mathbb{R}^{(B \cdot C) \times T \times H \times W}$. This folding operation effectively treats each channel of each sample as an independent element within the extended batch. As a consequence, the FNO is applied identically across all channels but without cross-channel mixing at this stage.

This *time-conditioned FNO* then operates over latent temporal embeddings as functions on their spatial domain $v_\theta(z, t) = \mathcal{G}_\theta(z)$ to learn the time-dependent vector field that transports a reference latent distribution π_0 to the latent data distribution π_1. By leveraging the spectral inductive bias of FNOs, the learned velocity field can capture both local and long-range correlations efficiently, improving the expressivity and stability of flow matching in high-dimensional latent spaces.

4 Experiments

The TempO is evaluated with the goal of assessing its ability to learn accurate stochastic latent-space dynamics and forecast high-dimensional solution fields over medium to long time horizons. We test our method over PDE datasets which pose challenging spatio-temporal correlations and multiscale features, making them a natural testbed for latent flow-based modeling.

Our proposed TempO was set against five key methods. The state-of-the-art video generation method based on a U-Net shaped Vision Transformer (ViT) and modified optimal transport path Random frame conditioned flow Integration for VidEo pRediction (RIVER) proposed by Davtyan et al. (2023) matches or surpasses common video prediction benchmarks using 10x fewer computational resources (Davtyan et al., 2023). We also include the baseline conditional flow matching Lipman et al. (2023) which implements a U-Net trained using a theoretically optimal affine optimal transport (Affine-OT) path. The stochastic linear path (SLP) was proposed by Lim et al. (2025), tested with a ViT to directly address the challenges of spatiotemporal forecasting for PDE datasets. The Transformer-based latent space flow matching method with Affine-OT proposed by Dao et al. (2023) further demonstrates competitive performance in image generation using latent flow matching compared against both flow matching models and diffusion models (Phung et al., 2023; Ho et al., 2020) among others. We also evaluate both variance preserving diffusion (VP-diff) and variance exploding diffusion (VE-diff) paths which generalise the Denoising Diffusion Probabilistic noise perturbation model and a score-based model to flow matching paths, respectively (Ho et al., 2020; Song et al., 2021). Ryzhakov et al. (2024) establishes strong theoretical backing for both paths.

We then ablate the specific implementation of the methods (consisting of a specific architecture and a specific probability path). In summary, the choice of regressor includes our proposed TempO regressor, and additionally implement a ViT regressor (Davtyan et al., 2023; Lim et al., 2025) and a classic U-Net regressor (Lipman et al., 2023). We pretrain a convolutional autoencoder with residual and attention blocks to obtain a compressed latent representation of the dynamics, see Appendix D.

All methods were conditioned using sparse conditioning. These models are then supervised by each probability density paths described in Table 1, with further details in Appendix E. The Adam optimiser was used with a learning rate of 1e-4 for the FNO, and 5e-5 for the ViT and U-Net regressors. Models are trained on an 80/10/10 training to validation to test data splits.

We evaluate our models on three spatiotemporal PDE datasets: the shallow water equation (SWE), which simulate 2D free-surface flows; 2D reaction diffusion (RD-2D) exhibiting multiscale nonlinear patterns; and 2D incompressible Navier-Stokes vorticity (NS-ω) dataset capturing chaotic turbulent dynamics. During training, models are sparsely conditioned on the first 15 frames and tasked with predicting the subsequent frame at resolutions of $1 \times 128 \times 128$ (shallow water equation

(SWE)), $2 \times 128 \times 128$ (2D reaction diffusion (RD-2D)), and $1 \times 64 \times 64$ (2D incompressible Navier-Stokes vorticity (NS-ω)), see Appendix F.

5 RESULTS

Table 2: NS-ω Results: Comparison of TempO, U-Net, and ViT models.

Regressor	Path	MSE ↓	SpectralMSE ↓	RFNE ↓	PSNR ↑	Pearson ↑	SSIM ↑
TempO	RIVER	**5.63e-02**	**3.84e-02**	**2.50e-01**	**25.19**	**0.969**	0.786
	Affine-OT	5.77e-02	3.98e-02	2.54e-01	25.08	0.968	**0.789**
	VP-diff	8.10e-02	5.34e-02	2.85e-01	23.61	0.955	0.731
	VE-diff	2.96e-01	1.73e-01	5.60e-01	17.98	0.821	0.373
ViT	Affine-OT[1]	6.75e-02	4.38e-02	2.72e-01	24.40	0.962	0.758
	RIVER[2]	6.88e-02	4.33e-02	2.73e-01	24.32	0.962	0.750
	VP-diff[3]	7.77e-02	4.95e-02	2.89e-01	23.79	0.956	0.729
	VE-diff[3]	1.63e+00	9.27e-01	1.35e+00	10.67	0.118	0.024
U-Net	VP-diff[4]	4.05e-01	3.26e-01	6.71e-01	16.62	0.756	0.323
	RIVER	4.08e-01	3.28e-01	6.74e-01	16.59	0.752	0.321
	Affine-OT[5]	4.10e-01	3.42e-01	6.76e-01	16.57	0.751	0.324
	VE-diff[4]	5.02e-01	3.70e-01	7.48e-01	15.68	0.694	0.263

[1] (Dao et al., 2023), [2](Davtyan et al., 2023), [3] (Lim et al., 2025; Song & Ermon, 2020), [4](Ryzhakov et al., 2024), [5](Lipman et al., 2023)

Table 3: SWE and RD-2D Results: Comparison of TempO, U-Net, and ViT models.

Dataset	Method	MSE ↓	SpectralMSE ↓	RFNE ↓	PSNR ↑	Pearson ↑	SSIM ↑
SWE	$\text{TempO}_{\text{Affine-OT}}$	**6.64e-05**	**5.65e-05**	**7.64e-03**	**46.5**	**0.998**	**0.997**
	$\text{ViT}_{\text{Affine-OT}}$[1]	9.59e-05	7.93e-05	9.06e-03	44.9	0.997	0.995
	$\text{ViT}_{\text{VP-diff}}$[2]	1.30e-04	8.81e-05	1.05e-02	43.6	0.996	0.993
	$\text{ViT}_{\text{RIVER}}$[3]	2.99e-04	1.67e-04	1.63e-02	40.0	0.992	0.981
	ViT_{SLP}[4]	6.60e-04	-	1.28e-01	36.1	-	0.93
RD-2D	$\text{TempO}_{\text{Affine-OT}}$	**2.76e-05**	**2.18e-05**	**3.29e-02**	**65.7**	**1.000**	**0.999**
	$\text{U-Net}_{\text{Affine-OT}}$[5]	3.09e-05	2.45e-05	3.57e-02	65.2	0.999	**0.999**
	$\text{ViT}_{\text{Affine-OT}}$	6.30e-04	4.40e-04	1.67e-01	52.2	0.987	0.986
	ViT_{SLP}[4]	3.56e-04	-	1.16e-01	34.3	-	0.90

[1] (Dao et al., 2023), [2](Lim et al., 2025; Song & Ermon, 2020), [3](Davtyan et al., 2023), [4](Lim et al., 2025); results reported from original paper trained on same dataset., [5](Lipman et al., 2023)

Overall, TempO outperforms the methods proposed by Lim et al. (2025); Song et al. (2022); Lipman et al. (2023) and Davtyan et al. (2023) as well as the ablated methods. For results predicting NS-ω in Table 2, we observe a 16% improvement in MSE and an 11.4% lower spectral MSE, producing spatially and spectrally accurate next steps. Its lower RFNE indicates reduced scale-independent error, while SSIM shows improved fidelity in local features, critical for the localized vorticity patterns where small spatial distortions significantly affect downstream evolution (Majda & Bertozzi, 2001). PSNR and Pearson see lower normalised ranges in values, indicating that large scale features like the vorticity intensity and global structure agreement, respectively, are more easily captured across all models, with a clear advantage by TempO; additional visualisations in Appendix H.

We select top performing comparisons for SWE and RD-2D, Table 3), where TempO maintains superior performance. In SWE, it achieves a 28.8% lower SpectralMSE and higher PSNR, indicating faithful amplitude, spectral content, and structural coherence with sharp boundaries preserved, see Appendix I for additional visualisations and ablated comparisons. Overall MSE is reduced by 30.8

In RD-2D, $\text{U-Net}_{\text{Affine-OT}}$ competes closely with TempO, benefiting from translation-equivariant convolutional layers that capture multi-scale dynamics and repeating local structures (Cohen & Welling, 2016). Both TempO and the U-Net have nearly matched PSNR, Pearon, and SSIM scores, with an improvement of 11% in SpectralMSE from the TempO. By contrast, the next best ViT regressor model is 95.6% drop in SpectralMSE, where attention might emphasize low-frequency global structures (Wang et al., 2022; Piao et al., 2024); see visual comparison in Appendix J.

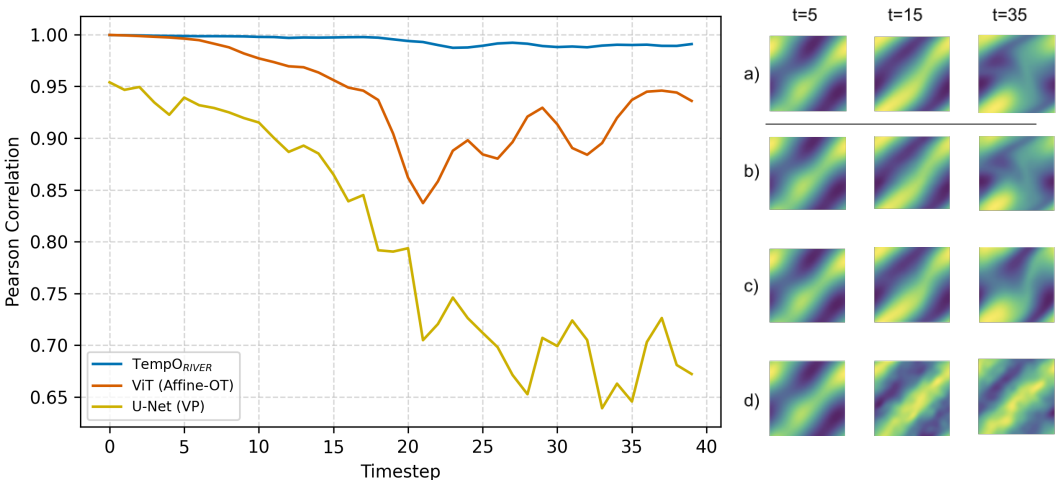

Figure 1: **Prediction performance comparison for NS-ω.** *Left: Pearson correlation across forecasted timesteps.* Forty timesteps are predicted by TempO, ViT, and U-Net conditioned on two preceding timesteps and sampled for each proceeding step. The Pearson correlation coefficient shows significant degradation for the U-Net, oscillatory behavior and degradation for the ViT, and consistently stable values above 0.98 for TempO. *Right: Predicted vorticity fields.* True data (a), TempO (b), ViT (c), and U-Net (d). At timesteps 5, 15, and 35 the ViT and U-Net models clearly diverge, with U-Net regressing to a noisy, while TempO maintaining excellent accuracy.

The timeseries forecasting task, see Fig. 1, evaluates how well models capture the underlying PDE. The model is provided two initial timeframes representing the conditioning and reference frames, respectively, and is then sampled for increasing temporal offsets with the reference set to be the most recent generation. TempO maintains Pearson correlation above 0.98 over 40 forecasted timesteps, indicating stable amplitude and phase tracking. The ViT regressor holds above 0.95 for 20 steps before degrading, while the flow matching baseline (Lipman et al., 2023) shows steady decline. This suggests TempO effectively mimics the dynamics without significant error accumulation. This is further demonstrated by visualisations of the vorticity field at key timesteps in Fig. 1 (right), where $t = 35$ most clearly shows TempO's faithful capture of turbulent eddies in comparison to the ViT regressor, which fails to predict the small vortical structure.

5.1 Spectral Analysis

The spectral analysis of TempO versus the top alternative ViT$_{\text{Affine-OT}}$ and the baseline U-Net$_{\text{Affine-OT}}$ (Lipman et al., 2023) in Fig. 2 examines the scale-resolved error via the energy per wavenumber k, or at the scale of $\frac{1}{k}$. This provides scale-resolved context to the SpectralMSE, which averages the MSE of the Fourier coefficients to a single metric. For NS-ω, the first 8 modes which cumulatively make up 99% of the total energy, beyond which the modes have negligible contributions to overall flow dynamics, see Appendix G. TempO closely follows the true spectrum compared to both ViT$_{\text{Affine-OT}}$ and U-Net$_{\text{Affine-OT}}$, though all three methods diverge past $k = 8$. We observe from the inset of Fig. 2 that TempO exhibits a small residual which fluctuates about 0 whereas the ViT$_{\text{Affine-OT}}$ has a negative and increasing error: the ViT regressor tends to capture the lower wavenumbers well, but then underestimates the higher wavelengths notably after $k = 4$.

Modes	SpectralMSE
1	8.57e-02
2	4.10e-02
4	3.98e-02
8	3.79e-02
16	3.74e-02

Figure 3: Ablation: Fourier mode cutoffs with TempO.

We observe also that the number of modes retained during the FFT of TempO in Fig. 3 follows the observation of a close spectral match up until $k = 8$, where the SpectralMSE sees the most improvement; however, from 8 modes to 16 modes, the performance appears to saturate. Fig. 3 demonstrates that up to 8 modes capture the essential dynamics, while the fundamental frequency alone is insufficient and likely under-represents necessary higher frequency components; adding more than 8 modes yields diminishing returns, matching the true spectral

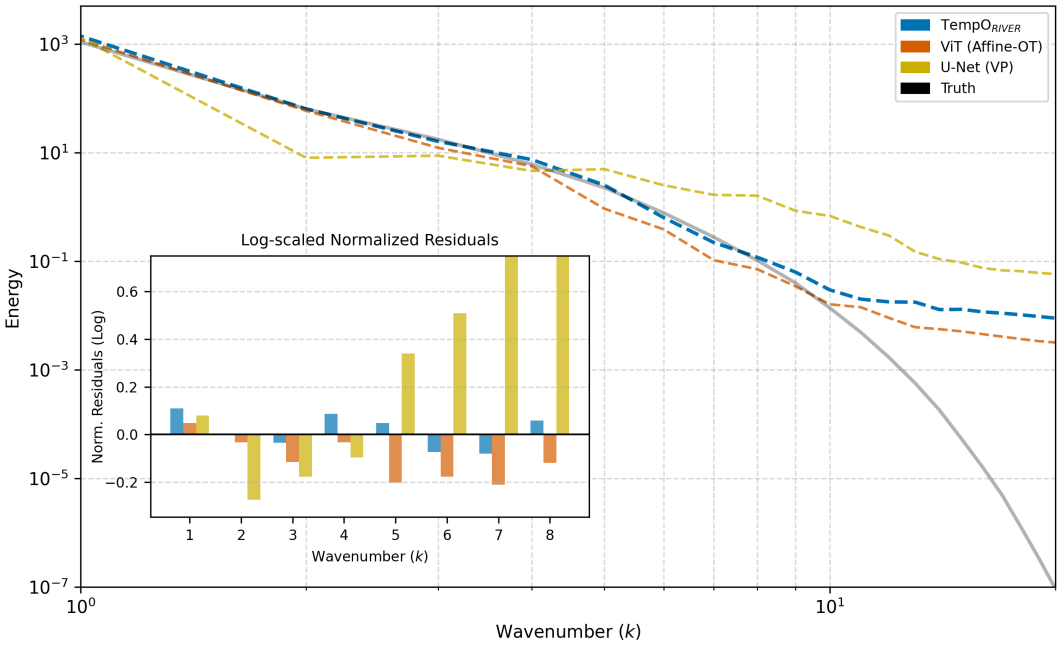

Figure 2: **Spectral graphs for NS-ω.** Comparison of energy spectra for TempO, a ViT-based model, and the U-Net baseline (Lipman et al., 2023). The first eight Fourier modes capture 99% of the energy, with higher modes negligible. TempO aligns closely, while the ViT underestimates energy beyond $k = 4$. The inset bar plot shows TempO oscillating tightly around zero with small deviations, the ViT producing larger negative deviations, and the U-Net performing markedly worse.

analysis; extended metrics support this trend in Appendix K. This empirical saturation beyond 8 modes is consistent with the theoretical expectation in Theorem 3.3, where FNOs are shown to achieve accuracy with asymptotically fewer parameters by leveraging only the most informative spectral modes.

5.2 Efficiency

Finally, we also train the models over varying sequence lengths and measuring next-step prediction error (MSE) and 40-step forecast error (MSE/time), shown in Table 5. MSE is lowest for shorter sequences, as the model learns from fewer choices of indices for sparse conditioning during training. Conversely, MSE/time decreases with longer sequences, reflecting better long-horizon performance. Notably, TempO's MSE/time drops faster and plateaus lower than the ViT, indicating better data efficiency to extrapolate from the same available sequence length.

Model	Params	FLOPs	Mem (MB)	NFEs
TempO	0.49M	208M	∼50	560
ViT	3.39M	10M	∼80	942
U-Net	14.0M	555M	∼68	728

Table 4: Model Complexity and Efficiency: number of function evaluationss (NFEs) are averaged from sampling performed for Table 2 for adaptive solver `dopri5` and tolerances of 1e-5.

TempO is the most lightweight model among the three choices of regressors, with $\tilde{7}$x fewer parameters than the ViT and $\tilde{2}8$x fewer than the U-Net. In addition, it sees a significantly lower memory usage compared to the ViT where attention has higher demands and the U-Net where skip-connections hold onto additional memory.

Method	Seq	MSE	MSE/time
TempO	2	4.92e-02	2.70e-01
	5	4.75e-02	3.41e-01
	10	5.04e-02	4.94e-02
	15	5.61e-02	3.83e-02
	25	6.26e-02	4.22e-02
ViT (Affine-OT)	2	6.75e-02	2.71e-01
	5	5.43e-02	3.59e-01
	10	6.01e-02	1.49e-01
	15	6.70e-02	4.53e-02
	25	7.68e-02	8.56e-02

Table 5: Ablation: Performance comparison scaling with sequence length.

While TempO has a moderate number of FLoating Point OPerations (FLOPs), landing between the ViT and U-Net, this may be offsetted by the NFEs seen during the ODE integration where TempO takes only 560 evaluations to meet the same tolerances. Beyond these empirical measures, TempO further benefits from its shared spatial Fourier layers. By folding the channel dimension and truncating higher modes, the spectral convolution scales as $O(N^2 \log N)$, in contrast to the naive $O(N^3 \log N)$ cost of a full 3D FFT. Also for reference, a ViT layer can scale as $O(N^4)$ in 2D 3.2, higher than the quasi-quadratic cost of the FNO.

6 Limitations

Flow matching models struggle with extreme data sparsity which can distort the distributions being learned, whereas hybrid models or models with explicitly defined conservations can fall back on injected physical knowledge. Additionally, similar to other generative models, adaptations, e.g. architectural modifications, would be necessary to extend the method towards a foundational model framework. Finally, while our stable and accurate 40-step forecasting represents the longer end time horizons, it remains an open question on how to forecast for much longer timeframes. Critical applications in science and engineering would require further study both experimentally and theoretically to establish statistically reliable forecasting.

7 Conclusions and Further work

In this work, we addressed the challenge of long-horizon PDE forecasting via our proposed method TempO. TempO consistently outperformed state-of-the-art baselines across three benchmark PDE datasets and achieves stable long-horizon 40 step forecasts with remarkable accuracy to the true trajectories as well as superior spectral fidelity. The modified time-conditioned FNO is parameter-efficient while improving the capture of both local and global spectral modes, resulting in improvements in both data- and compute- efficiency. Additionally, we establish that FNO can achieve an upper bound on approximation error that sampler-based architectures cannot reach without significantly more parameters, Corollary 3.3. These results highlight the importance of architectures that align with the continuous nature of PDE dynamics, enabling not only improved predictive accuracy but also physically consistent, long-horizon trajectories.

Consequently, TempO poses significant opportunity for further work in this field. Under typical real-world environments, PDE observations may come from irregularly sampled domains; since our method already demonstrates state-of-the-art generations using a simple autoencoder (AE) and the latent time-conditioned FNO which no longer relies on a regular grid as is a limitation of the original FNO (Li et al., 2021), one extension of our work is to then extend our method to real-world settings to forecast PDE over irregular domains.

References

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the Language of Time Series, November 2024.

Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Guanghui Liu, Amit Raj, Yuanzhen Li, Michael Rubinstein, Tomer Michaeli, Oliver Wang, Deqing Sun, Tali Dekel, and Inbar Mosseri. Lumiere: A Space-Time Diffusion Model for Video Generation. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–11, Tokyo Japan, December 2024. ACM. ISBN 979-8-4007-1131-2. doi: 10.1145/3680528.3687614.

Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International Conference on Machine Learning*, pp. 2990–2999. PMLR, 2016.

Quan Dao, Hao Phung, Binh Nguyen, and Anh Tran. Flow Matching in Latent Space, July 2023.

Aram Davtyan, Sepehr Sameni, and Paolo Favaro. Efficient Video Prediction via Sparsely Conditioned Flow Matching, August 2023.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia

Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The Llama 3 Herd of Models, November 2024.

Xingzhuo Guo, Yu Zhang, Baixu Chen, Haoran Xu, Jianmin Wang, and Mingsheng Long. Dynamical Diffusion: Learning Temporal Dynamics with Diffusion Models, March 2025.

Ali Hatamizadeh, Jiaming Song, Guilin Liu, Jan Kautz, and Arash Vahdat. DiffiT: Diffusion Vision Transformers for Image Generation. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, volume 15066, pp. 37–55. Springer Nature Switzerland, Cham, 2025. ISBN 978-3-031-73241-6 978-3-031-73242-3. doi: 10.1007/978-3-031-73242-3_3.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models, December 2020.

Jiahe Huang, Guandao Yang, Zichen Wang, and Jeong Joon Park. DiffusionPDE: Generative PDE-Solving Under Partial Observation, November 2024.

Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal Flow Matching for Efficient Video Generative Modeling, March 2025.

Richard John, Lukas Herron, and Pratyush Tiwary. A survey of probabilistic generative frameworks for molecular simulations, November 2024.

Zeqian Ju, Yuancheng Wang, Kai Shen, Xu Tan, Detai Xin, Dongchao Yang, Yanqing Liu, Yichong Leng, Kaitao Song, Siliang Tang, Zhizheng Wu, Tao Qin, Xiang-Yang Li, Wei Ye, Shikun Zhang, Jiang Bian, Lei He, Jinyu Li, and Sheng Zhao. NaturalSpeech 3: Zero-Shot Speech Synthesis with Factorized Codec and Diffusion Models, April 2024.

Gavin Kerrigan, Giosue Migliorini, and Padhraic Smyth. Functional Flow Matching, December 2023.

Marcel Kollovieh, Marten Lienen, David Lüdke, Leo Schwinn, and Stephan Günnemann. Flow Matching with Gaussian Process Priors for Probabilistic Time Series Forecasting. In *The Thirteenth International Conference on Learning Representations*, October 2024.

Nikola Kovachki, Samuel Lanthaler, and Siddhartha Mishra. On Universal Approximation and Error Bounds for Fourier Neural Operators. *Journal of Machine Learning Research*, 22(290): 1–76, 2021. ISSN 1533-7928.

Xin Li, Jingdong Zhang, Qunxi Zhu, Chengli Zhao, Xue Zhang, Xiaojun Duan, and Wei Lin. From Fourier to Neural ODEs: Flow Matching for Modeling Complex Systems, May 2024.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier Neural Operator for Parametric Partial Differential Equations, May 2021.

Soon Hoe Lim, Yijin Wang, Annan Yu, Emma Hart, Michael W. Mahoney, Xiaoye S. Li, and N. Benjamin Erichson. Elucidating the Design Choice of Probability Paths in Flow Matching for Forecasting, July 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling, February 2023.

Xiaoyi Liu and Hao Tang. DiffFNO: Diffusion Fourier Neural Operator, April 2025.

Lu Lu, Pengzhan Jin, and George Em Karniadakis. DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, March 2021. ISSN 2522-5839. doi: 10.1038/s42256-021-00302-5.

Andrew J. Majda and Andrea L. Bertozzi. *Vorticity and Incompressible Flow*. Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge, 2001. ISBN 978-0-521-63057-3. doi: 10.1017/CBO9780511613203.

Roberto Molinaro, Samuel Lanthaler, Bogdan Raonić, Tobias Rohner, Victor Armegioiu, Stephan Simonis, Dana Grund, Yannick Ramic, Zhong Yi Wan, Fei Sha, Siddhartha Mishra, and Leonardo Zepeda-Núñez. Generative AI for fast and accurate statistical computation of fluids, February 2025.

Hao Phung, Quan Dao, and Anh Tran. Wavelet Diffusion Models are fast and scalable Image Generators, March 2023.

Xihao Piao, Zheng Chen, Taichi Murayama, Yasuko Matsubara, and Yasushi Sakurai. Fredformer: Frequency Debiased Transformer for Time Series Forecasting. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2400–2410, August 2024. doi: 10.1145/3637528.3671928.

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, January 2025. ISSN 1476-4687. doi: 10.1038/s41586-024-08252-9.

Md Ashiqur Rahman, Manuel A. Florez, Anima Anandkumar, Zachary E. Ross, and Kamyar Azizzadenesheli. Generative Adversarial Neural Operators, October 2022.

M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, February 2019. ISSN 0021-9991. doi: 10.1016/j.jcp.2018.10.045.

Gleb Ryzhakov, Svetlana Pavlova, Egor Sevriugov, and Ivan Oseledets. Explicit Flow Matching: On The Theory of Flow Matching Algorithms with Applications. In *ICOMP 2024*. arXiv, July 2024. doi: 10.48550/arXiv.2402.03232.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models, October 2022.

Yang Song and Stefano Ermon. Generative Modeling by Estimating Gradients of the Data Distribution, October 2020.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations, February 2021.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Dan MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. PDEBENCH: An Extensive Benchmark for Scientific Machine Learning, October 2022.

Ella Tamir, Najwa Laabid, Markus Heinonen, Vikas Garg, and Arno Solin. Conditional Flow Matching for Time Series Modelling. In *ICML 2024 Workshop on Structured Probabilistic Inference {\&} Generative Modeling*, July 2024.

Mingtian Tan, Mike A. Merrill, Vinayak Gupta, Tim Althoff, and Thomas Hartvigsen. Are Language Models Actually Useful for Time Series Forecasting?, October 2024.

Peihao Wang, Wenqing Zheng, Tianlong Chen, and Zhangyang Wang. Anti-Oversmoothing in Deep Vision Transformers via the Fourier Domain Analysis: From Theory to Practice, March 2022.

Jiachen Yao, Abbas Mammadov, Julius Berner, Gavin Kerrigan, Jong Chul Ye, Kamyar Azizzadenesheli, and Anima Anandkumar. Guided Diffusion Sampling on Function Spaces with Applications to PDEs, May 2025.

A PROOFS

Proof of Theorem 3.1. Step 1: Spectral truncation. By assumption the Fourier coefficients of $\mathcal{G}(u)$ satisfy

$$\left|\widehat{\mathcal{G}(u)}(k)\right| \;\le\; C_\lambda(1+|k|)^{-p}, \qquad \forall u \in \mathcal{U},\;\; k \in \mathbb{Z}^d.$$

If we keep only the modes $|k| \le K$ and set

$$\mathcal{G}_K(u)(x) \;:=\; \sum_{|k|\le K} \widehat{\mathcal{G}(u)}(k) e^{ik\cdot x},$$

then the error lives in the high modes:

$$\|\mathcal{G}(u) - \mathcal{G}_K(u)\|_{H^{s'}}^2 \;=\; \sum_{|k|>K} (1+|k|^2)^{s'} \left|\widehat{\mathcal{G}(u)}(k)\right|^2.$$

Using the decay bound gives

$$\|\mathcal{G}(u) - \mathcal{G}_K(u)\|_{H^{s'}}^2 \;\le\; C_\lambda^2 \sum_{|k|>K} (1+|k|)^{2(s'-p)}.$$

A standard counting argument (comparing the lattice sum with a radial integral) shows this tail is $\lesssim K^{-2\alpha}$, with

$$\alpha \;:=\; p - s' - \tfrac{d}{2} \;>\; 0.$$

This is exactly the pseudo-spectral tail estimate also used in (Kovachki et al., 2021, Thm. 40). Hence choosing

$$K \;\asymp\; \varepsilon^{-1/\alpha}$$

ensures $\|\mathcal{G} - \mathcal{G}_K\|_{H^{s'}} \le \varepsilon/2$.

Step 2: Reduction to a finite-dimensional map. The truncated operator $\mathcal{G}_K$ is determined by finitely many Fourier coefficients $\{\widehat{\mathcal{G}(u)}(k)\}_{|k|\le K}$, with output dimension $m_{\text{out}} \asymp K^d$. To apply a neural network, we also restrict the input to finitely many low modes. By compactness of $\mathcal{U} \subset H^s$ and continuity of the projection P_M, there exists M such that

$$\|\mathcal{G}_K(u) - \mathcal{G}_K(P_M u)\|_{H^{s'}} \;\le\; \varepsilon/6 \qquad \forall u \in \mathcal{U}.$$

This is the same finite-dimensional reduction used in the universal approximation argument of (Kovachki et al., 2021, Thm. 15). Thus it suffices to approximate the finite-dimensional continuous map

$$F:\; (\widehat{u}(k))_{|k|\le M} \;\longmapsto\; (\widehat{\mathcal{G}(u)}(k))_{|k|\le K},$$

between compact subsets of Euclidean spaces.

Step 3: Approximation of the finite map. Classical universal approximation theorems (and the constructive Ψ–FNO realization in (Kovachki et al., 2021, Def. 11, Thm. 15)) ensure that for any desired accuracy $\delta > 0$, one can build a neural network (or FNO block) approximating F uniformly to error δ on each retained coefficient. To control the $H^{s'}$–norm it suffices to achieve coefficient accuracy

$$\delta \;\lesssim\; \frac{\varepsilon}{K^{s'+d/2}}.$$

This choice ensures $\|P_K\mathcal{G}(u) - \widetilde{\mathcal{G}}_\theta(u)\|_{H^{s'}} \le \varepsilon/3$. Constructive approximation bounds then give a parameter count

$$P \;\lesssim\; K^d \cdot \mathrm{polylog}(1/\varepsilon),$$

where the extra logarithmic factor reflects standard overheads in coefficient quantization and network approximation (Kovachki et al., 2021, Remark 22).

Step 4: Assemble errors and conclude. Adding the contributions: - spectral truncation error $\le \varepsilon/2$ (Step 1), - input-projection error $\le \varepsilon/6$ (Step 2), - finite-map approximation error $\le \varepsilon/3$ (Step 3),

we obtain

$$\sup_{u\in\mathcal{U}} \|\mathcal{G}(u) - \mathcal{G}_\theta(u)\|_{H^{s'}} \;\leq\; \varepsilon.$$

Substituting $K \asymp \varepsilon^{-1/\alpha}$ into the parameter bound gives

$$P_{\mathrm{FNO}}(\varepsilon) \;\lesssim\; \varepsilon^{-d/\alpha},$$

up to the mild logarithmic factors discussed above. □

Proof of Proposition 3.2. Step 1: Finite-dimensional subspace and sampling. Consider the K-mode Fourier subspace

$$V_K := \mathrm{span}\{e^{ik\cdot x} : |k| \leq K\} \subset L^2(\mathbb{T}^d), \qquad \dim V_K =: D_K \asymp K^d.$$

Any sampler-based learner observes an input $u \in V_K$ only through n fixed points $(u(x_1), \ldots, u(x_n))$. This defines a linear map

$$S : V_K \to \mathbb{C}^n, \quad S(u) = (u(x_1), \ldots, u(x_n)).$$

Step 2: Nyquist / injectivity argument. To reconstruct all Fourier modes up to radius K, the sampling map S must be injective on V_K. In matrix terms, S is represented by an $n \times D_K$ Vandermonde-like matrix. To have full rank D_K, we require

$$n \;\geq\; D_K \;\asymp\; K^d.$$

If $n < D_K$, there exists a nonzero $u \in V_K$ vanishing on all sample points, so the learner cannot distinguish it from zero. This is the standard Nyquist/dimension-counting requirement: at least as many samples as degrees of freedom.

Step 3: Parameter lower bound. After sampling, the learner applies a parametric map $M : \mathbb{C}^n \to \mathbb{C}^m$ (e.g., a neural network) to produce either output samples or coefficients. To implement arbitrary linear transformations on the D_K retained modes (e.g., arbitrary Fourier multipliers), the parametric map must have at least D_K free parameters. For fully general dense linear maps (no structural constraints), one needs

$$P \;\gtrsim\; D_K^2 \;\asymp\; K^{2d}.$$

Step 4: Conversion to accuracy ε. From the FNO upper bound analysis, achieving accuracy ε requires

$$K \;\asymp\; \varepsilon^{-1/\alpha}, \qquad \alpha = p - s' - d/2 > 0.$$

Substituting this into the previous bounds gives the scaling

$$n \;\gtrsim\; \varepsilon^{-\,d/\alpha}, \qquad P_{\mathrm{sampler}}(\varepsilon) \;\gtrsim\; \varepsilon^{-\,\beta d/\alpha},$$

with $\beta = 1$ for minimal mode-wise maps and $\beta = 2$ for fully dense maps.

Step 5: Conclusion. Hence any sampler-based architecture that must reconstruct all modes up to radius K requires asymptotically more parameters than an FNO whenever $\beta > 1$, justifying the lower bound in the proposition. □

B Flow Matching Background

Flow matching The core idea of flow matching is to learn a time-dependent velocity field, $v_\theta(z, t)$, which defines an ODE in the latent space:

$$\frac{dz(t)}{dt} = v_\theta(z(t), t), \quad z(0) \sim \pi_0, \tag{2}$$

where π_0 is a simple reference distribution (e.g., Gaussian). Integrating this ODE transports samples to the latent data distribution π_1, such that $z(1) \sim \pi_1$ and $p_1(z) \approx f_\phi \# \mathcal{D}_{\text{data}}$, where $f_\phi \# \mu$ denotes the pushforward measure of a distribution μ under f_ϕ, i.e., $(f_\phi \# \mu)(A) = \mu(f_\phi^{-1}(A))$ for

measurable sets A. The corresponding time-dependent probability density, $p_t(z)$, evolves according to the continuity equation:

$$\frac{\partial p_t(z)}{\partial t} + \nabla_z \cdot \big(p_t(z)\, v_\theta(z,t)\big) = 0. \tag{3}$$

In practice, the target velocity field $u(t,z)$ and the full marginal density $p_t(z)$ are generally unknown and intractable. Flow matching sidesteps this issue by directly supervising the model to match the instantaneous vector field along interpolating paths between the reference π_0 and the target π_1, allowing for deterministic, efficient sampling. Different choices of paths lead to different training dynamics and inductive biases, as they implicitly define the target velocity field $u(t,z)$ that the model regresses against.

Integrating this ODE from $t=0$ to $t=1$ transports the reference distribution π_0 to the latent data distribution π_1, so that $z(1) \sim \pi_1$ and $p_1(z) \approx f_\phi \# \mathcal{D}_{\text{data}}$.

Latent Flow Matching. We now instantiate the general flow matching framework in the latent space. Let $z_\tau = f_\phi(x_\tau)$ for $\tau = 1, \ldots, m$, where f_ϕ is a pretrained encoder mapping from the data space to the lower-dimensional latent space. Our objective is to approximate the ground-truth latent distribution $q(z_\tau \mid x_1, \ldots, x_{\tau-1})$ by a parametric distribution $p(z_\tau \mid z_{\tau-1})$, which can later be decoded back to the data space via $x_\tau = g_\psi(z_\tau)$ using a decoder g_ψ.

The latent dynamics can be expressed by the ODE:

$$\dot{z}_t = u_t(z_t), \tag{4}$$

where u_t denotes the (true) time-dependent velocity field. Learning these dynamics amounts to approximating u_t with a neural parameterization. Following the flow matching framework, we introduce a model velocity field $v_\theta : [0,1] \times \mathbb{R}^Z \to \mathbb{R}^Z$ and consider the ODE

$$\dot{\phi}_t(z) = v_\theta(\phi_t(z), t), \quad \phi_0(z) = z, \tag{5}$$

which defines a time-dependent diffeomorphism ϕ_t pushing forward an initial reference distribution p_0 (often chosen as $\mathcal{N}(0, I)$) to a target distribution $p_1 \approx q$ along the density path p_t:

$$p_t = (\phi_t)_\# p_0, \tag{6}$$

where $(\cdot)_\#$ denotes the pushforward. In other words, the goal of flow matching is to learn a deterministic coupling between p_0 and q by training v_θ so that the solution satisfies $z_0 \sim p_0$ and $z_1 \sim q$.

Given a probability path p_t and its associated velocity field u_t, flow matching reduces to a least-squares regression problem:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t \sim U[0,1],\, z \sim p_t}\, \omega(t)\, \|v_\theta(z,t) - u_t(z)\|_2^2, \tag{7}$$

where $\omega(t) > 0$ is a weighting function, often taken as $\omega(t) = 1$ (Lipman et al., 2022). This formulation ensures that the learned velocity field aligns with the target field u_t at all times, thereby generating the desired marginal probability path.

C Fourier Neural Operator Background

An FNO is designed to learn a mapping between function spaces, rather than between finite-dimensional vectors. Consider a function $u : \mathbb{R}^d \to \mathbb{R}^c$ representing data, for example in $\mathbb{R}^X$, with samples $x \in \mathbb{R}^X$. Then, an FNO parameterizes an operator as

$$\mathcal{G}_\theta : u \mapsto \tilde{u}, \quad \tilde{u} : \mathcal{D} \to \mathbb{R}^{c_{\text{out}}},$$

that maps u to an output function $\tilde{u}$ (e.g., a solution field of a PDE or a transformed spatial signal).

This mapping is implemented via iterative Fourier layers which perform spectral transformations of the input:

$$\hat{u}(k) = \mathcal{F}[u](k), \quad \hat{\tilde{u}}(k) = R_\theta(k) \cdot \hat{u}(k), \tag{8}$$

followed by an inverse Fourier transform back to the spatial domain:

$$\tilde{u}(x) = \mathcal{F}^{-1}[\hat{\tilde{u}}](x), \tag{9}$$

with $R_\theta(k)$ being learnable Fourier-mode weights and $\mathcal{F}$ denoting the Fourier transform. This spectral representation allows the FNO to efficiently capture long-range dependencies and global correlations in the data.

D AUTOENCODER DETAILS

Residual blocks throughout the architecture consist of two 3×3 convolutions with ReLU activation and group normalization (8 groups) in between, with the input added back to the output. Attention blocks are implemented using PyTorch's `nn.MultiheadAttention`, with embeddings reshaped from $[B, C, H, W]$ to $[B, HW, C]$.

The autoencoder is initialised with a depth of $d = 2$ resulting in a factor $2^d = 4$ compression for all datasets.

E MODEL HYPERPARAMETERS

We initialised the probability paths with the following hyperparameters. RIVER was defined with variance parameters $\sigma = 0.1$ and $\sigma_{\min} = 10^{-7}$. SLP used $\sigma = 0.1$ and $\sigma_{\min} = 0.01$. We further considered the VE-diff path with $\sigma_{\min} = 0.01$ and $\sigma_{\max} = 0.1$ and the VP-diff path initialized with $\beta_{\min} = 0.1$ and $\beta_{\max} = 20.0$ per (Lim et al., 2025).

We provide details for the vector field regressors' width and depth hyperparameters as per Table 6.

Model	**Parameter**	**Value**
TempO	n_{modes}	20
	Hidden channels	64
	Projection channels	64
	Depth	4
U-Net	Hidden channels	64
	Attention resolutions	(1, 2, 2)
	Channel multiplier	(1, 2, 4)
	Depth	3
ViT	Hidden channels	256
	Depth	4
	Mid-depth	5
	Output normalization	LayerNorm

Table 6: Descriptions of hyperparameters across TempO, U-Net, and ViT architectures.

F DATASET DETAILS

Table 7: Dataset sizes and trajectory lengths used in evaluation.

Dataset	# Trajectories	Timeseries Length
SWE	1000	100
RD-2D	1000	100
NS-ω	5000	50

Shallow water equation (SWE)

The SWEs are derived from the compressible Navier–Stokes equations and model free-surface flow problems in 2D. The system of hyperbolic PDEs is given by:

$$\partial_t h + \partial_x(hu) + \partial_y(hv) = 0, \tag{10}$$

$$\partial_t(hu) + \partial_x\left(u^2 h + \frac{1}{2} g_r h^2\right) + \partial_y(uvh) = -g_r h\, \partial_x b, \tag{11}$$

$$\partial_t(hv) + \partial_y\left(v^2 h + \frac{1}{2} g_r h^2\right) + \partial_x(uvh) = -g_r h\, \partial_y b, \tag{12}$$

where u, v are the horizontal and vertical velocities, h is the water height, b represents spatially varying bathymetry, and g_r is gravitational acceleration. The quantities hu and hv correspond to directional momentum components.

The dataset ((Takamoto et al., 2022)) simulates a 2D radial dam break scenario on a square domain $\Omega = [-2.5, 2.5]^2$. The initial water height is a circular bump in the center of the domain:

$$h(t=0,x,y) = \begin{cases} 2.0, & \text{if } r < r_0, \\ 1.0, & \text{if } r \geq r_0, \end{cases} \quad r = \sqrt{x^2+y^2}, \quad r_0 \sim \mathcal{U}(0.3, 0.7).$$

2D reaction diffusion (RD-2D)

The RD-2D dataset models two non-linearly coupled variables: the activator $u = u(t,x,y)$ and the inhibitor $v = v(t,x,y)$. The system of PDEs is:

$$\partial_t u = D_u\, \partial_{xx} u + D_u\, \partial_{yy} u + R_u(u,v), \tag{13}$$
$$\partial_t v = D_v\, \partial_{xx} v + D_v\, \partial_{yy} v + R_v(u,v), \tag{14}$$

where D_u and D_v are diffusion coefficients, and $R_u(u,v)$, $R_v(u,v)$ are the reaction functions. Specifically, the FitzHugh–Nagumo model defines the reactions as:

$$R_u(u,v) = u - u^3 - k - v, \tag{15}$$
$$R_v(u,v) = u - v, \tag{16}$$

with $k = 5 \times 10^{-3}$, $D_u = 1 \times 10^{-3}$, and $D_v = 5 \times 10^{-3}$.

The dataset ((Takamoto et al., 2022)) uses a simulation domain $x, y \in (-1, 1)$ and $t \in (0, 5]$ with initial condition set as standard normal random noise: $u(0,x,y) \sim \mathcal{N}(0, 1.0)$.

2D incompressible Navier-Stokes vorticity (NS-ω)

The NS-ω ((Li et al., 2021)) models 2D incompressible fluid flow on the unit torus. The system of equations is:

$$\partial_t w(x,t) + u(x,t) \cdot \nabla w(x,t) = \nu\, \Delta w(x,t) + f(x), \quad x \in (0,1)^2,\, t \in (0,T], \tag{17}$$
$$\nabla \cdot u(x,t) = 0, \tag{18}$$
$$w(x,0) = w_0(x), \tag{19}$$

where $w(x,t)$ is the vorticity, $u(x,t)$ is the velocity field, ν is viscosity, and $f(x)$ is a fixed forcing term:

$$f(x) = 0.1\Big(\sin(2\pi(x_1+x_2)) + \cos(2\pi(x_1+x_2))\Big).$$

The initial condition is sampled from a Gaussian measure:

$$w_0 \sim \mu, \quad \mu = \mathcal{N}\Big(0,\, \big(-\Delta + 49I\big)^{-2.5}\, 7^{3/2}\Big),$$

with periodic boundary conditions.

G Spectral Analysis of Ground Truth NS-ω

Fig. 4 shows how the quality of spectral truncations of the true Navier–Stokes vorticity field depends on the cutoff wavenumber k_{cut}. Given the full Fourier spectrum $\hat{\omega}(k_x, k_y)$, we apply a mask that retains only modes with $|k_x| + |k_y| \leq k_{\text{cut}}$, reconstruct the signal by inverse FFT, and compute three quantities as functions of k_{cut}:

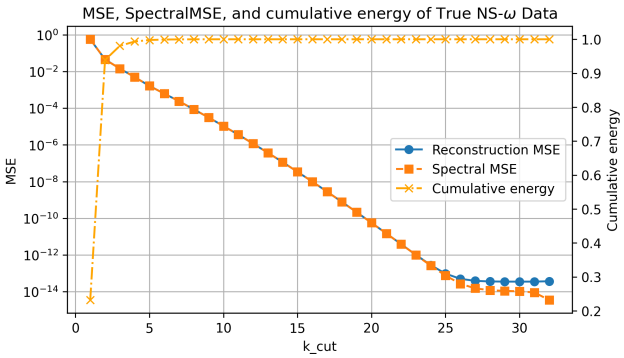

Figure 4: **Spectral Analysis of True Vorticity**: Reconstruction MSE, spectral MSE, and cumulative enstrophy fraction of true Navier–Stokes vorticity data as functions of cutoff wavenumber k_{cut}.

1. Reconstruction MSE: the mean squared error between the original and truncated fields in physical space.
2. Spectral MSE: the mean squared error in Fourier space, quantifying lost spectral content.
3. Cumulative energy fraction: the fraction of total energy $\sum |\hat{\omega}|^2$ retained by the truncated spectrum.

As k_{cut} increases, both reconstruction and spectral errors decrease, while the retained energy approaches unity.

H Extended Results for Navier–Stokes Vorticity

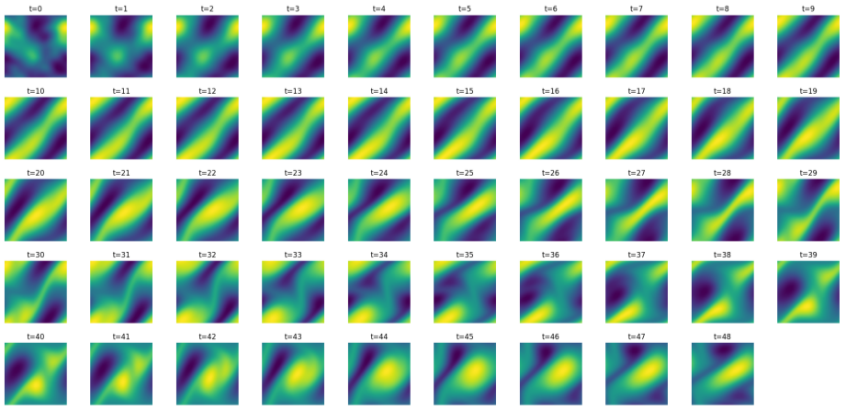

Figure 5: **Navier–Stokes Vorticity (Original).** Ground-truth timeseries across 40 timesteps.

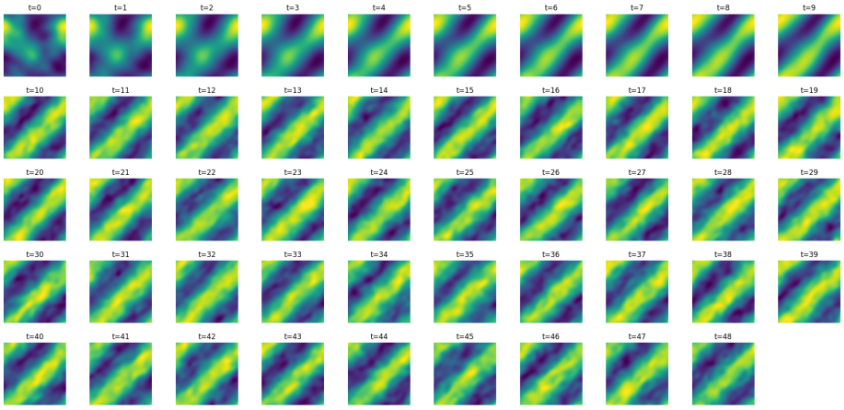

Figure 6: **Navier–Stokes Vorticity (U-Net).** Forecasted timeseries across 40 timesteps.

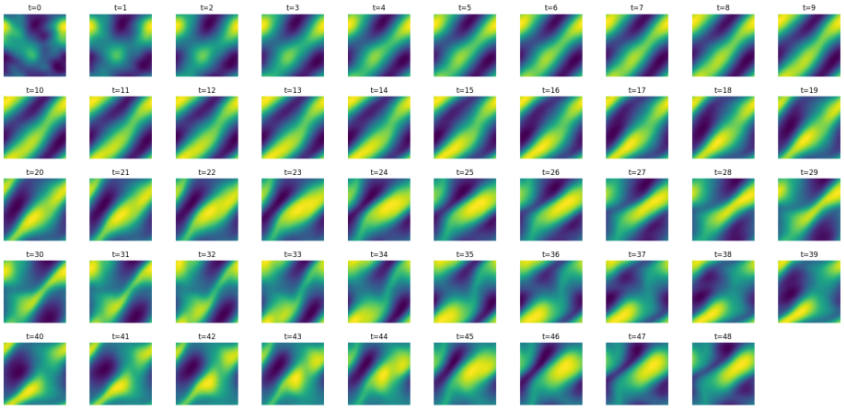

Figure 7: **Navier–Stokes Vorticity (ViT).** Forecasted timeseries across 40 timesteps.

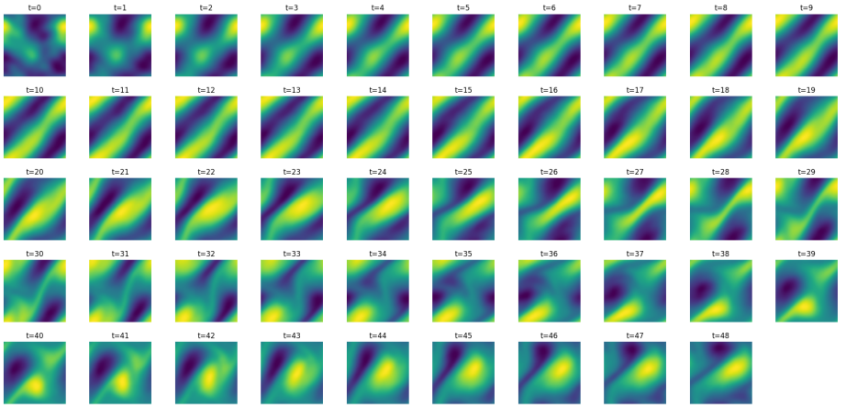

Figure 8: **Navier–Stokes Vorticity (TempO).** Forecasted timeseries across 40 timesteps.

I Extended Results for Shallow Water Equation

Regressor	Path	MSE ↓	SpectralMSE ↓	RFNE ↓	PSNR ↑	Pearson ↑	SSIM ↑
TempO	Affine-OT	**6.64e-05**	**5.65e-05**	**7.64e-03**	**46.5**	**0.998**	**0.997**
	RIVER	4.04e-04	2.33e-04	1.89e-02	38.7	0.989	0.976
	VE-diff	9.37e-04	8.22e-04	2.89e-02	35.2	0.994	0.977
	VP-diff	4.41e-03	2.51e-03	4.31e-02	28.3	0.872	0.857
ViT	Affine-OT	9.59e-05	7.93e-05	9.06e-03	44.9	0.997	0.995
	VP-diff	1.30e-04	8.81e-05	1.05e-02	43.6	0.996	0.993
	RIVER	2.99e-04	1.67e-04	1.63e-02	40.0	0.992	0.981
	SLP[1]	6.60e-04	-	1.28e-01	36.1	-	0.93
	VE-diff	1.28e-03	1.01e-03	3.38e-02	33.7	0.985	0.960
U-Net	VP-diff	1.37e-02	8.26e-03	1.10e-01	23.4	0.546	0.627
	RIVER	1.61e-02	1.00e-02	1.20e-01	22.7	0.437	0.610
	Affine-OT	1.68e-02	1.01e-02	1.22e-01	22.5	0.435	0.593

Table 8: Comparison of TempO, U-Net, and ViT models under different probability paths for the SWE. The best value for each metric is highlighted in bold.

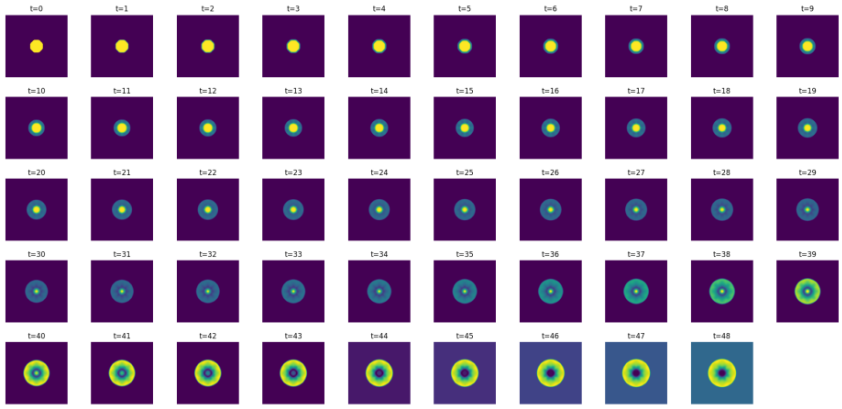

Figure 9: **SWE (Original).** Ground-truth timeseries across 40 timesteps.

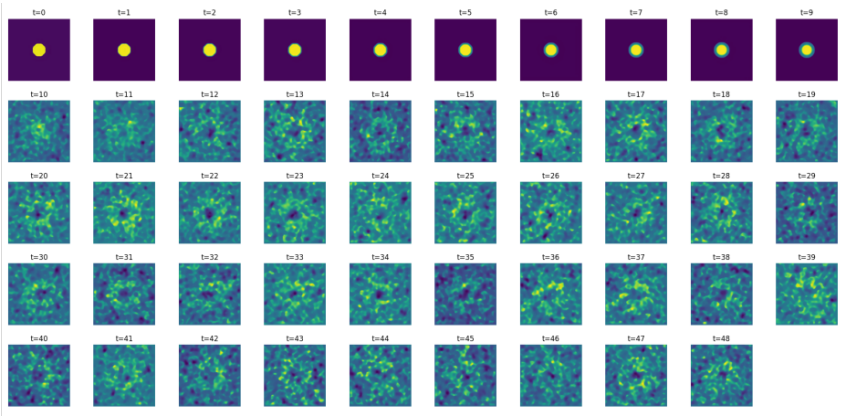

Figure 10: **SWE (U-Net).** Forecasted timeseries across 40 timesteps.

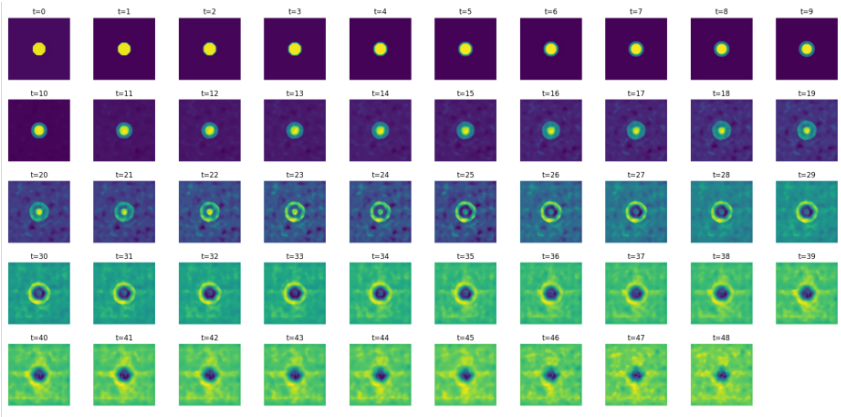

Figure 11: **SWE (ViT).** Forecasted timeseries across 40 timesteps.

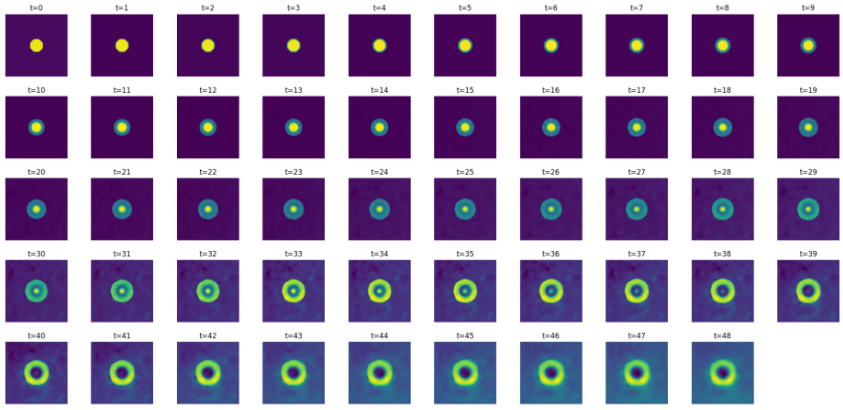

Figure 12: **SWE (TempO).** Forecasted timeseries across 40 timesteps.

J Extended Results for 2D Reaction Diffusion

Regressor	Path	MSE ↓	SpectralMSE ↓	RFNE ↓	PSNR ↑	Pearson ↑	SSIM ↑
TempO	Affine-OT	**2.76e-05**	**2.18e-05**	**3.29e-02**	**65.7**	**1.000**	**0.999**
	RIVER	9.36e-04	5.47e-04	2.08e-01	50.4	0.975	0.978
	VE-diff	1.58e-03	1.38e-03	2.70e-01	48.2	0.990	0.977
	VP-diff	1.24e-02	1.01e-02	4.95e-01	39.2	0.714	0.862
ViT	Affine-OT	6.30e-04	4.40e-04	1.67e-01	52.2	0.987	0.986
	SLP^2	3.56e-04	-	1.16e-01	34.3	-	0.90
	RIVER	1.00e-03	5.89e-04	2.16e-01	50.1	0.973	0.977
	VE-diff	3.54e-03	2.23e-03	4.06e-01	44.7	0.915	0.946
U-Net	Affine-OT	3.09e-05	2.45e-05	3.57e-02	65.2	0.999	**0.999**
	RIVER	1.02e-03	5.49e-04	2.17e-01	50.1	0.972	0.976
	VE-diff	9.03e-03	6.07e-03	6.42e-01	40.6	0.820	0.860
	VP-diff	2.09e-02	1.66e-02	6.81e-01	37.0	0.574	0.792

Table 9: Comparison of TempO, U-Net, and ViT models under different probability paths for the RD-2D. The best value for each metric is highlighted in bold.

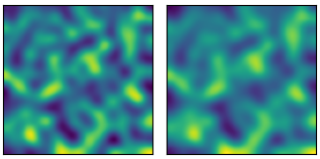

Figure 13: **Reaction Diffusion (Original).** Ground-truth end sample, from initial conditions of randomly sampled noise.

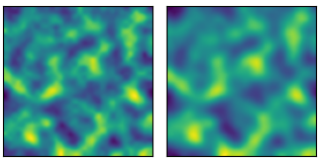

Figure 14: **Reaction Diffusion (U-Net).** Forecasted end sample, from initial conditions of randomly sampled noise.

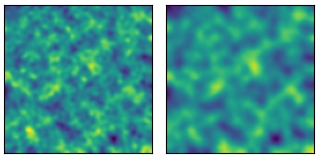

Figure 15: **Reaction Diffusion (ViT).** Forecasted end sample, from initial conditions of randomly sampled noise.

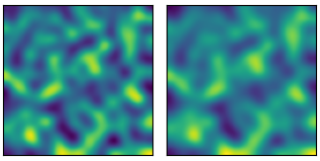

Figure 16: **Reaction Diffusion (TempO).** Forecasted end sample, from initial conditions of randomly sampled noise.

K Extended Ablation Results

Table 10: Ablation over different training sequence lengths on the NS-ω dataset. TempO and the top performing alternative are trained while varying sequence lengths and evaluated on 10 timesteps to predict the next step.

Method	Seq. Len.	MSE	DensityMSE	SpectralMSE	RFNE	PSNR	Pearson	SSIM	NFE
TempO	3	4.924e-02	7.685e-05	3.531e-02	2.328e-01	25.769	0.973	0.803	74
	6	4.753e-02	1.133e-04	3.394e-02	2.276e-01	25.923	0.974	0.800	608
	11	5.036e-02	1.055e-04	3.620e-02	2.352e-01	25.672	0.972	0.800	842
	16	5.607e-02	1.282e-04	3.821e-02	2.497e-01	25.205	0.969	0.786	938
	26	6.255e-02	7.487e-05	3.726e-02	2.541e-01	24.730	0.968	0.765	1070
ViT (Affine-OT)	3	6.748e-02	1.414e-04	4.652e-02	2.678e-01	24.401	0.963	0.766	116
	6	5.434e-02	1.239e-04	3.727e-02	2.416e-01	25.341	0.970	0.783	1766
	11	6.014e-02	1.376e-04	4.067e-02	2.546e-01	24.901	0.967	0.777	1712
	16	6.701e-02	1.093e-04	4.428e-02	2.680e-01	24.431	0.963	0.764	1622
	26	7.682e-02	8.104e-05	4.468e-02	2.778e-01	23.838	0.960	0.741	1100

Table 11: Ablation of the TempO model on the NS-ω dataset by varying the number of modes. Models are trained with different numbers of Fourier modes and evaluated on 10 timesteps to predict the next step.

Modes	MSE	DensityMSE	SpectralMSE	RFNE	PSNR	Pearson	SSIM	NFE
1	1.409e-01	1.075e-04	8.566e-02	3.947e-01	21.204	0.921	0.588	5798
2	6.103e-02	8.928e-05	4.096e-02	2.596e-01	24.837	0.966	0.765	1688
4	5.789e-02	8.361e-05	3.978e-02	2.538e-01	25.066	0.968	0.776	1058
8	5.528e-02	8.498e-05	3.788e-02	2.481e-01	25.267	0.969	0.788	800
16	5.471e-02	8.757e-05	3.742e-02	2.467e-01	25.312	0.970	0.787	884

L Use of Large Language Models (LLMs)

We acknowledge the use of ChatGPT to make suggestions on how to polish the text, correct grammar, and ensure clarity in writing. No results, code, or data were created or altered by the model.