SPECTRAL-INSPIRED OPERATOR LEARNING WITH LIMITED DATA AND UNKNOWN PHYSICS

A PREPRINT

Han Wan, Rui Zhang, Hao Sun
Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
{wanhan2001, rayzhang, haosun}@ruc.edu.cn

September 29, 2025

ABSTRACT

Learning PDE dynamics from limited data with unknown physics is challenging. Existing neural PDE solvers either require large datasets or rely on known physics (e.g., PDE residuals or handcrafted stencils), leading to limited applicability. To address these challenges, we propose **S**pectral-**I**nspired **N**eural **O**perator (SINO), which can model complex systems from just 2-5 trajectories, without requiring explicit PDE terms. Specially, SINO automatically captures both local and global spatial derivatives from frequency indices, enabling a compact representation of the underlying differential operators in physics-agnostic regimes. To model nonlinear effects, it employs a Π-block that performs multiplicative operations on spectral features, complemented by a low-pass filter to suppress aliasing. Extensive experiments on both 2D and 3D PDE benchmarks demonstrate that SINO achieves state-of-the-art performance, with improvements of 1–2 orders of magnitude in accuracy. Particularly, with only 5 training trajectories, SINO outperforms data-driven methods trained on 1000 trajectories and remains predictive on challenging out-of-distribution cases where other methods fail.

1 Introduction

Partial differential equations (PDEs) govern the evolution of numerous physical systems, from heat transfer (Pletcher et al., 2012) and chemical reactions (Ghergu & Radulescu, 2011) to fluid dynamics (Holton & Hakim, 2013). Accurately modeling and simulating such PDE-governed systems is essential for the understanding of natural phenomena and engineering applications. However, in many scientific computing scenarios, we face two fundamental challenges: **(i) only a few trajectories of data are available**, and **(ii) the governing physics is unknown**. While traditional methods offer strong stability and interpretability, their reliance on a deep understanding of the physical system limits their application in scenarios where the underlying physics is unknown.

In recent years, with the development of deep learning, data-driven neural PDE solvers have been proposed to address the limitations of traditional numerical methods (Azizzadenesheli et al., 2024). Representative works include DeepONet (Lu et al., 2021), FNO (Li et al., 2021), and DPOT (Hao et al., 2024). By leveraging the expressive power of neural networks and large datasets, these methods learn mappings between function spaces. As a result, they can reduce computational costs under coarse spatiotemporal resolutions and do not require explicit knowledge of the underlying PDEs. While offering notable advantages, such approaches remain highly data-hungry. In many scientific and engineering domains, obtaining sufficient training data requires expensive physical experiments (e.g., wind tunnel testing or combustion experiments), which severely limits their practical applications (Parente & Swaminathan, 2024; Li et al., 2024a; Zhang et al., 2025b).

To reduce the data demands of data-driven solvers, physics-aware methods incorporate physical knowledge into deep learning models, yielding two main paradigms: physics-informed and physics-encoded (Faroughi et al., 2022). Physics-informed methods, such as PINNs (Raissi et al., 2019) and PINO (Li et al., 2024c), add soft constraints by including PDE residuals in the loss function. They require explicit forms of the governing PDEs and often suffer from training instability (Krishnapriyan et al., 2021). On the other hand, physics-encoded methods, such as PeRCNN (Rao et al., 2023, 2022), P^2C^2Net (Wang et al., 2024c), and TSM (Sun et al., 2023), embed physical rules directly into

the network architecture, e.g., hardcoding finite difference stencils as convolutional kernels (Rao et al., 2023). When designing these models, one still needs complete or partial PDE terms as prior knowledge. Moreover, since many architectures borrow from local numerical schemes such as the finite difference method (FDM) or the finite volume method (FVM), they inherit limitations in handling high-order derivatives and global interactions (McGreivy & Hakim, 2024).

To address these challenges, we propose **S**pectral-**I**nspired **N**eural **O**perator (SINO), which can accurately model complex systems from as few as 2-5 trajectories without any explicit PDE terms. The design of SINO is inspired by spectral methods, which are well known for their ability to capture global information with exponential accuracy (Jingrun et al., 2022). Specifically, we propose a Frequency-to-Vector (Freq2Vec) module that maps each frequency index into a learnable embedding, mimicking the derivative multipliers in spectral methods without relying on predefined operators. To further capture nonlinear dynamics, we employ a Π-block that models multiplicative interactions, together with a low-pass filter to mitigate aliasing effects. By inheriting the structure of spectral methods, SINO achieves stronger generalization from limited observations and demonstrates robust out-of-distribution (OOD) performance compared to data-driven approaches, while also handling global interactions and high-order derivatives more effectively than other physics-aware methods. **In summary, we make the following contributions**:

(1) Novel architecture. We propose SINO, a data-efficient neural operator for modeling complex spatiotemporal dynamics without requiring prior knowledge of PDE terms. We propose two key modules: Freq2Vec simulates frequency domain derivative multipliers, and the nonlinear operator block captures complex interactions through efficient multiplicative operations.

(2) State-of-the-art (SOTA) performance. SINO achieves SOTA performance across multiple 2D and 3D PDE benchmarks, delivering improvements of 1–2 orders of magnitude in accuracy compared to baselines. To our knowledge, SINO is among the first physics-aware methods capable of simulating globally coupled systems such as the NSEs without requiring explicit PDE terms.

(3) Data efficiency and OOD handling. Due to the spectral-inspired design, SINO performs genuine operator learning rather than data memorization. With only 5 training trajectories, SINO outperforms data-driven methods trained on 1000 trajectories and remains predictive in out-of-distribution (OOD) tests where other methods fail.

2 Related work

Data-driven neural PDE solvers. When provided with abundant data, data-driven methods learn mappings between function spaces, bypassing the need for explicit PDE formulation. Representative examples include the FNO (Li et al., 2021) and its variants (Tran et al., 2023; Wen et al., 2022), DeepONet (Lu et al., 2021) and its variants (Venturi & Casey, 2023; Lee et al., 2023). Transformer-based models (Wu et al., 2024; Li et al., 2024b) and graph-based architectures (Pfaff et al., 2021; Brandstetter et al., 2022) extend neural PDE solvers to handle complex geometries by operating on mesh or graph representations. Recently, PDE foundation models aim to solve multiple PDE families within a single framework, achieving broad applicability across different physical systems (Hao et al., 2024; Zhang et al., 2024; Hang et al., 2024). Thanks to their rapid inference ability, these data-driven solvers have found broad scientific computing applications, such as weather forecasting (Zhang et al., 2023), control systems (Hu et al., 2025), and geometric design (Wang et al., 2024b). However, a major drawback of data-driven PDE solvers is their heavy reliance on large datasets, which in many applications are costly to obtain through expensive physical experiments (Parente & Swaminathan, 2024; Li et al., 2024a). In data-scarce regimes, their generalization performance degrades substantially. In contrast, SINO uses a spectral-inspired framework to embed physical structure, enabling effective training from limited data and robust OOD generalization.

Physics-aware neural PDE solvers. Unlike data-driven methods, physics-aware methods aim to reduce data dependency by embedding physical priors into the learning process or architecture design (Faroughi et al., 2022). Based on how they incorporate physical laws, physics-aware methods can be categorized into physics-informed and physics-encoded methods. Physics-informed methods integrate PDE residuals (Wang et al., 2021) or numerical schemes (e.g., FDM (Huang et al., 2023), spectral methods (Du et al., 2024), particle methods (Zhang et al., 2025a)) into the loss function, but require complete knowledge of the governing PDEs and often suffer from training instability (Krishnapriyan et al., 2021). Physics-encoded methods incorporate physical structure in a hard-constrained manner, such as using FDM (Long et al., 2019; Rao et al., 2023) or FVM (Kochkov et al., 2021; Sun et al., 2023), yet depend on specific PDE terms and are constrained by their local receptive field. Additionally, recent physics-encoded approaches that incorporate spectral methods have shown promise in improving global modeling but still require full or partial knowledge of the governing PDE terms, especially when tackling complex systems such as the NSEs (Dresdner et al., 2023; Li et al., 2025). Unlike these hybrid approaches, SINO not only captures global information in the frequency domain but also requires no prior knowledge of specific PDE terms.

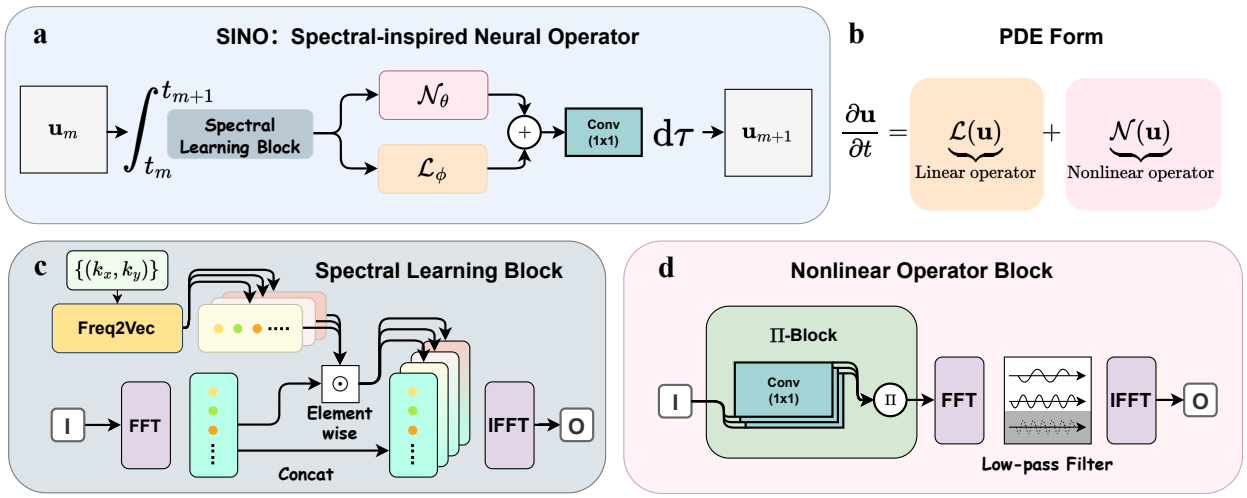


Figure 1: **The framework of SINO.** (a) The overall architecture. (b) The PDE formulation. (c) The structure of the Spectral Learning Block (SLB). (d) The structure of the Nonlinear Operator Block (Π-Block + low-pass filter).

3 Methodology

3.1 Problem setting and preliminary

This paper addresses the problem of modeling PDE-governed systems from limited data when PDE terms are unknown. Consider a system defined on $\Omega \subset \mathbb{R}^d$, whose state $\mathbf{u}(\mathbf{x}, t; \boldsymbol{\mu}) \in \mathbb{R}^C$ satisfies

$$\partial_t \mathbf{u}(\mathbf{x}, t; \boldsymbol{\mu}) = \mathcal{L}(\mathbf{u}(\mathbf{x}, t; \boldsymbol{\mu})) + \mathcal{N}(\mathbf{u}(\mathbf{x}, t; \boldsymbol{\mu})). \tag{1}$$

Here, $\mathbf{u}$ denotes the state field, $\boldsymbol{\mu}$ represents PDE parameters, and $\mathcal{L}(\cdot)$ and $\mathcal{N}(\cdot)$ are *unknown* linear and nonlinear differential operators, which include multiple spatial derivatives of various orders. When the PDE formulation in Eq. 1 is known, spectral methods are among the most accurate and fastest numerical methods on regular domains because they exhibit exponential convergence and capture global information (Shen et al., 2011; McGreivy & Hakim, 2024). Given the state $\mathbf{u}_m(\mathbf{x})$ at time t_m, a spectral solver advances to t_{m+1} in three steps.

Firstly, the state is transformed to frequency space, and all spatial derivatives are computed. In detail, the Fourier transform $\mathcal{F}$ converts $\mathbf{u}_m$ into spectral coefficients $\widehat{\mathbf{u}}_m$ as follows:

$$\widehat{\mathbf{u}_m}(\mathbf{k}) = \mathcal{F}\big[\mathbf{u}_m(\mathbf{x})\big] = \int_\Omega \mathbf{u}_m(\mathbf{x})\, e^{-\mathrm{i}\,\mathbf{k}\cdot\mathbf{x}}\, \mathrm{d}\mathbf{x}, \tag{2}$$

where $\mathbf{k}$ is the frequency indexing corresponding to different frequency components. In the frequency domain, each derivative operator can become an exact multiplier. For example, the Laplacian operator $\nabla^2 \mathbf{u}_m$ satisfies $\widehat{\nabla^2 \mathbf{u}_m}(\mathbf{k}) = -\|\mathbf{k}\|_2^2\, \widehat{\mathbf{u}_m}(\mathbf{k})$ in the frequency domain.

Secondly, each derivative is transformed back to the spatial domain via the inverse Fourier transform $\mathcal{F}^{-1}$ and combined to form the linear and nonlinear terms of the PDE. For linear terms, the corresponding spectral derivatives can be directly scaled by their coefficients and then summed. For example, the diffusion term $\nu \nabla^2 \mathbf{u}_m$ can be denoted as $\nu \mathcal{F}^{-1}[\widehat{\nabla^2 \mathbf{u}_m}]$. For nonlinear terms, the computation requires calculating element-wise products in the spatial domain. For instance, the convection term $\nabla \mathbf{u}_m \cdot \mathbf{u}_m$ can be calculated as $\mathcal{F}^{-1}[\widehat{\nabla \mathbf{u}_m}] \cdot \mathcal{F}^{-1}[\widehat{\mathbf{u}_m}]$.

Thirdly, a high-order time-stepping scheme (e.g., a fourth-order Runge-Kutta (RK4) scheme) is deployed to obtain $\mathbf{u}_{m+1}(\mathbf{x})$.

3.2 Overall architecture

Motivated by spectral methods, we propose SINO to model nonlinear systems from limited data when all PDE terms are unknown. Similar to a spectral solver, SINO follows three steps (Fig. 1a).

Firstly, SINO maps the input state $\mathbf{u}_m$ to the frequency domain via the Fourier transform and uses a Spectral Learning Block (SLB) to simulate various spatial derivatives (Fig. 1c). Because the exact PDE terms are unknown, we employ a

Frequency-to-Vector (Freq2Vec) module that learns how to represent each derivative from its frequency indexing $\mathbf{k}$ under the supervision of limited data.

Secondly, the learned representations are transformed through linear ($\mathcal{L}_\phi$) and nonlinear ($\mathcal{N}_\theta$) parts and then recombined by a 1×1 convolution, where ϕ and θ denote the parameters of SINO. Specifically, in the nonlinear branch, we use a Π-block to model the nonlinear interactions via element-wise products and a low-pass filter to prevent aliasing (Fig. 1d).

Thirdly, the combined right-hand side is advanced in time using an RK4 scheme to obtain $\mathbf{u}_{m+1}$. As a result, the overall framework of SINO can be described as

$$\mathbf{u}_{m+1} = \int_{t_m}^{t_{m+1}} \mathrm{Conv}_{1\mathrm{x}1}\left(\mathcal{L}_\phi(\mathrm{SLB}(\mathbf{u}_\tau)) + \mathcal{N}_\theta(\mathrm{SLB}(\mathbf{u}_\tau))\right) \mathrm{d}\tau. \tag{3}$$

3.3 Architecture components of SINO

We herein detail the network architecture of SINO, including the spectral learning block, the linear block, and the nonlinear block.

Spectral learning block. In spectral methods, spatial derivatives are computed exactly in the frequency domain, such as $\widehat{\partial_x^n \mathbf{u}}(\mathbf{k}) = (\mathrm{i}k_x)^n \cdot \widehat{\mathbf{u}}(k)$. When PDE terms are unknown, we design a spectral learning block to learn various spatial derivatives in the frequency domain. In detail, we propose a key module named Frequency-to-Vector (Freq2Vec). This module learns a latent representation $\psi(\mathbf{k})$ for each frequency indexing $\mathbf{k}$ via a shared MLP, i.e., $\psi(\mathbf{k}) = \mathrm{MLP}(\mathbf{k})$. The output $\psi(\mathbf{k})$ encodes the operator action at indexing $\mathbf{k}$ and plays a role analogous to the exact multipliers (e.g., $(\mathrm{i}k_x)^n$) used in classical spectral methods. To approximate the unknown derivative operator $\mathcal{D}$, SINO computes a element-wise product in the frequency domain, i.e., $\widehat{\mathcal{D}\mathbf{u}}(\mathbf{k}) = \psi(\mathbf{k}) \cdot \widehat{\mathbf{u}}(\mathbf{k})$, which mimics the way spectral methods apply multipliers to represent differential operators. The result is then transformed back to the spatial domain: $\mathcal{D}\mathbf{u}(\mathbf{x}) = \mathcal{F}^{-1}[\widehat{\mathcal{D}\mathbf{u}}(\mathbf{k})]$.

A key advantage of this spectral formulation is that SINO not only learns standard spatial differential operators with high accuracy, but also naturally captures globally coupled relations. For instance, the velocity-vorticity formulation in 2D NSE can be expressed in a simple closed form. Given vorticity $\omega = \partial_x \mathbf{u}_y - \partial_y \mathbf{u}_x$, the velocity can be recovered via the Biot–Savart law (Saffman, 1995):

$$\widehat{\mathbf{u}}(\mathbf{k}) = \frac{\mathrm{i}\,\mathbf{k}^\perp}{\|\mathbf{k}\|_2^2}\,\widehat{\omega}(\mathbf{k}), \quad \mathbf{k}^\perp := (-k_y,\, k_x), \tag{4}$$

which corresponds to a simple spectral multiplier. Such relations can in principle be approximated by Freq2Vec due to its expressive capacity. By contrast, other physics-encoded methods (Wang et al., 2024c; Yan et al., 2025) built on local discretization schemes must solve a Poisson equation in Eq. 4, thereby requiring detailed PDE terms.

Linear operator block. After computing the spectral derivatives via the spectral learning block, we use a linear operator $\mathcal{L}_\phi$ to combine them with simple linear channel mixing. Concretely, the set of derived features is passed through a 1×1 convolution that learns to weight and sum these channels.

Nonlinear operator block. Unlike the linear block's simple channel mixing, the nonlinear operator block uses multiplicative interactions to emulate nonlinear PDE terms (Fig. 1d). Given the calculated spatial derivative features $\mathcal{D}\mathbf{u}(\mathbf{x})$, we design the Π-block as follows. We first map the derivative features into P channels via a 1×1 convolution and then perform element-wise products across these channels to model nonlinear interactions:

$$\mathbf{v}_\Pi(\mathbf{x}) = \prod_{p=1}^{P} \left(W_p\, \mathcal{D}\mathbf{u}(\mathbf{x}) + b_p\right), \tag{5}$$

where W_p and b_p denote the weights and biases. This Π-block offers a lightweight yet powerful mechanism for modeling nonlinear interactions through element-wise products, which can naturally express various nonlinear terms (e.g., the convection term $\mathbf{u} \cdot \nabla \mathbf{u}$). As shown later in Fig. 5, feature maps from the Π-block closely align with ground-truth differential operators of the underlying PDEs, suggesting that SINO implicitly learns physically meaningful operators.

However, nonlinear interactions can create frequencies above the Nyquist limit, causing aliasing, where high frequencies fold into low ones and introduce spurious noise (Kravchenko & Moin, 1997). To mitigate this effect, we perform de-aliasing in the frequency domain using the classical low-pass filter with a $2/3$-rule. Specifically, we apply a Fourier transform to the nonlinear output and zero out high-frequency modes exceeding the $2/3$ cutoff:

$$\widetilde{\mathbf{v}}_\Pi(\mathbf{k}) = \mathbf{M}(\mathbf{k}) \cdot \mathcal{F}[\mathbf{v}_\Pi(\mathbf{x})](\mathbf{k}), \quad \mathbf{M}(\mathbf{k}) = \begin{cases} 1, & \|\mathbf{k}\|_\infty \le 2/3 k_{\max}, \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

where $k_{\max}$ denotes the highest frequency mode. This ensures that the nonlinear output does not distort the solution with aliased frequencies. Finally, we map the de-aliased representation back to the spatial domain. By combining Π-block with the low-pass filter, the nonlinear block in SINO captures complex operator interactions efficiently and enjoys strong numerical stability.

4 Theoretical Insights of SINO

We now provide theoretical insights into why SINO performs well with limited data, focusing on its approximation ability and generalization behavior. In essence, SINO is a **compact yet expressive** neural operator that balances approximation accuracy and generalization.

For a physical field $\mathbf{u}$, one step of SINO corresponds to predicting the right-hand side (RHS) of the PDE, which involves linear and nonlinear combinations of spatial derivatives of $\mathbf{u}$. The following theorem reveals that SINO can approximate a broad class of such operators (proof in Appendix A):

Theorem 1 (Approximation ability of SINO) *Let $\mathbf{u} : \Omega \to \mathbb{R}^C$ be a periodic function on the torus $\Omega = [0, 1)^d$, bandlimited to $\|\mathbf{k}\|_\infty \leq k_{\max}$. Suppose the RHS of the PDE can be written as*

$$\partial_t \mathbf{u} = \mathcal{R}(\mathbf{u}) := \sum_{j=1}^{J} \prod_{i=1}^{n_j} \mathcal{D}_{j,i}\mathbf{u},$$

where each $\mathcal{D}_{j,i}$ is a linear differential operator of finite order. Then, for any $\epsilon > 0$, there exists a SINO model $\mathcal{S}_\theta$ such that

$$\|\mathcal{S}_\theta(\mathbf{u}) - \mathcal{R}(\mathbf{u})\|_{L^2} < \epsilon.$$

Theorem 1 shows that SINO can approximate nonlinear PDE operators constructed from linear derivatives and finite multiplicative interactions, covering a wide range of reaction-advection-diffusion systems. Beyond approximation, classical learning theory (Mohri et al., 2018) implies that the generalization gap scales as $\mathcal{O}(\sqrt{C(\mathcal{H})/N})$, where $C(\mathcal{H})$ denotes a complexity measure of the hypothesis class and N is the number of training samples. Since SINO achieves low training error with substantially fewer parameters than other neural operators, it induces a less complex hypothesis class, which intuitively explains its strong performance in low-data regimes.

5 Experiments

Given 2-5 observed trajectories, we evaluate SINO against baselines across various 2D and 3D scenarios without any known PDE terms. We also assess its OOD generalization and conduct ablation studies. **Source code will be released upon acceptance.**

5.1 Datasets and baseline models

Datasets. We evaluate SINO on three PDE classes, each with distinct computational challenges.

(1) Kuramoto-Sivashinsky Equation (KSE). We consider the 2D KSE (Jayaprakash et al., 1993):

$$\partial_t u = -\nabla^2 u - \nabla^4 u - 0.5|\nabla u|^2, \quad \mathbf{x} \in [0, 12\pi)^2, \ t \in [0, 5]. \tag{7}$$

The KSE is known for its stiffness due to a biharmonic term $\nabla^4 u$ and the emergence of spatiotemporal chaos. We use this PDE to evaluate the performance of SINO in handling high-order derivatives and denote the corresponding experiment as E1.

(2) Navier-Stokes Equation (NSE). To examine globally coupled nonlinear dynamics and assess extrapolation ability, we adopt the 2D incompressible NSE in vorticity form as follows:

$$\partial_t \omega = \nu \nabla^2 \omega - (\mathbf{u} \cdot \nabla)\omega + f, \quad \mathbf{x} \in [0, 1)^2, \ t \in [0, 15], \tag{8}$$

where ω is the vorticity and $\mathbf{u}$ is the velocity. NSE presents several challenges like multi-scale turbulence and nonlocal coupling between ω and $\mathbf{u}$. To assess performance under varying tasks, we consider two viscosities ($\nu \in \{10^{-4}, 10^{-5}\}$) and two forcings ($f_1(\mathbf{x}) = 0.1\cos(8\pi x_1)$, $f_2(\mathbf{x}) = 0.1\sqrt{2}\,\sin\big(2\pi(x_1 + x_2) + \frac{\pi}{4}\big)$), yielding four setups (E2-E5). To evaluate extrapolation ability, we provide the first 10 s of data during training, while testing extends to 15 s.

(3) Burgers' equation. We consider Burgers' equation, a canonical model for nonlinear advection and shock formation:

$$\partial_t \mathbf{u} = 0.01\nabla^2 \mathbf{u} - (\mathbf{u} \cdot \nabla)\mathbf{u}, \quad \mathbf{x} \in [0, 2\pi)^d, \ t \in [0, 2], \tag{9}$$

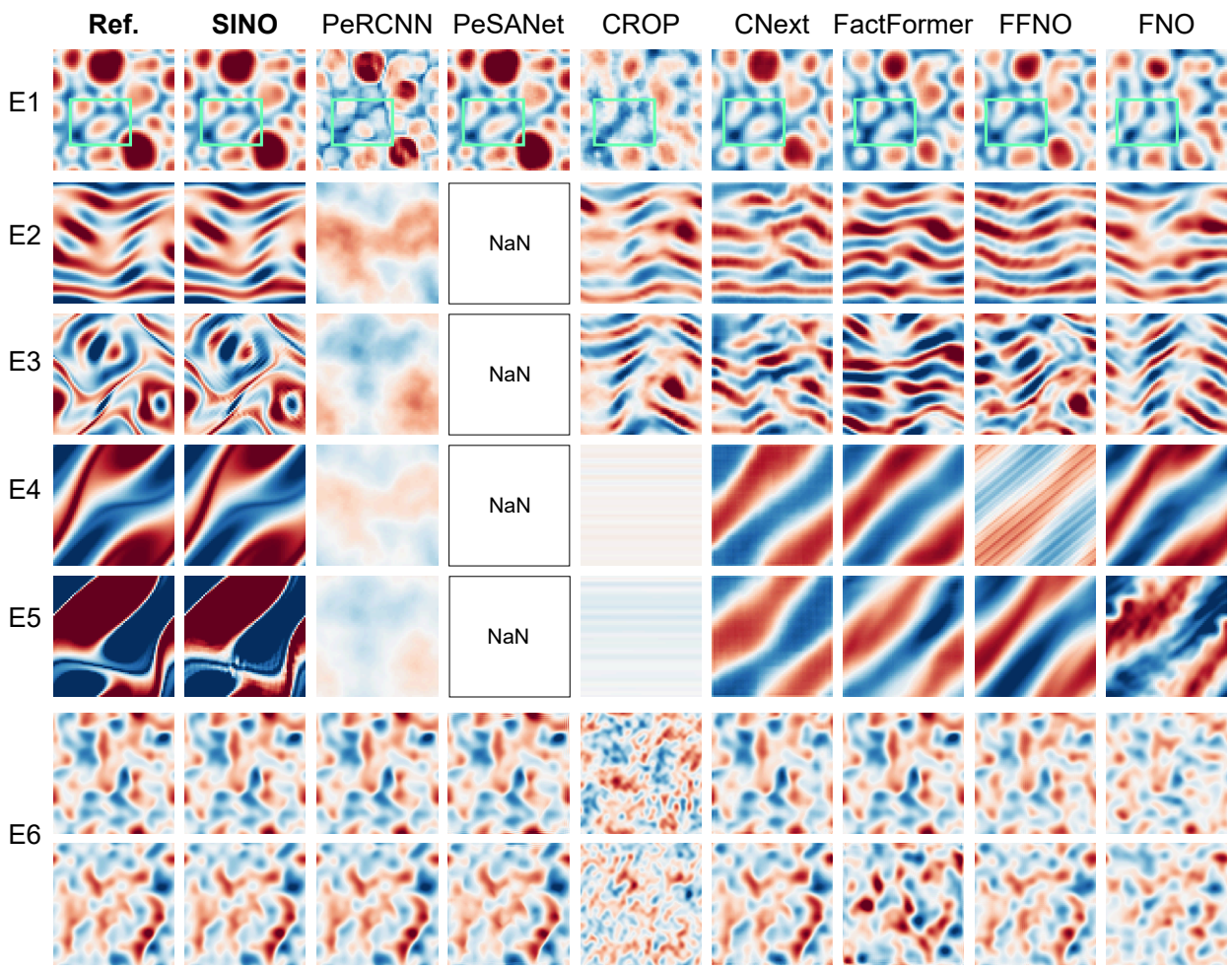


Figure 2: **Predicted snapshots of SINO and other baselines on 2D cases.** From E1 to E6, we show the final prediction plots for each method. For the 2D Burgers' equation (E6), the two rows denote the two components of $\mathbf{u} = [u, v]$, respectively.

where d denotes the spatial dimension. In our experiments, we consider both the 2D case ($d = 2$, E6) and the 3D case ($d = 3$, E7).

All datasets are simulated with high-resolution spectral solvers. We subsample 2-5 low-resolution trajectories per case for training. Dataset details are in Appendix 3.

Baseline models. We compare SINO with data-driven baselines including **FNO** (Li et al., 2021), **FFNO** (Tran et al., 2023), **FactFormer** (Li et al., 2024b), **CNext** (Liu et al., 2022; Ohana et al., 2024), and **CROP** (Gao et al., 2025). We also include physics-encoded models such as **PeRCNN** (Rao et al., 2023) and **PeSANet** (Wan et al., 2025).

5.2 Main Results

We herein present the experimental results, with additional results provided in Appendix C.

KSE (E1). For KSE cases, PeRCNN suffers from discretization error when approximating fourth-order derivatives on coarse grids due to its FD-based stencils. PeSANet partially alleviates this limitation by incorporating frequency information, achieving the second-best result. However, its error remains $6.8\times$ larger than that of SINO (Table 1). Furthermore, data-driven methods miss the major peaks of the solution field, resulting in inaccurate predictions (Fig. 1).

NSE (E2-E5). For the NSE cases, the challenge lies not only in capturing the nonlinear coupling between velocity and vorticity without explicit priors, but also in extrapolating beyond the training horizon. SINO maintains high correlation with the ground-truth over long rollouts (Fig. 4) and preserves coherent vortex structures even at the final timestep (Fig. 1). By contrast, PeRCNN and PeSANet collapse in the absence of sufficient physical priors, while

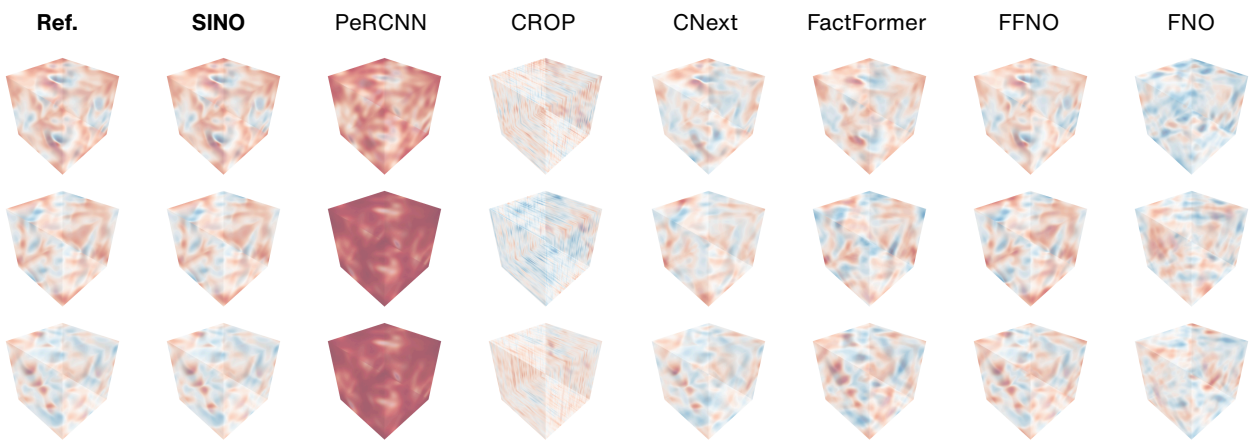


Figure 3: **Predicted snapshots of SINO and other baselines on 3D Burgers cases (E7).** The three rows denote the three components of $\mathbf{u} = [u, v, w]$, respectively.

Table 1: **Relative ℓ_2 error for seven cases.** The best results are shown in **bold**, and the second-best are underlined. NaN denotes 'Not a Number', and NA denotes 'Not Applicable'.

Model	KSE	NSE				Burgers	
	E1	E2 $10^{-4}, f_1$	E3 $10^{-5}, f_1$	E4 $10^{-4}, f_2$	E5 $10^{-5}, f_2$	E6 2D	E7 3D
PeRCNN (Rao et al., 2023)	0.4027	0.8161	0.8341	0.9136	0.9127	0.0174	0.9530
PeSANet (Wan et al., 2025)	0.0833	NaN	NaN	NaN	NaN	0.0974	NA
CROP (Gao et al., 2025)	0.4541	0.9925	1.0021	0.9982	0.9939	0.9864	1.0826
CNext (Ohana et al., 2024)	0.2256	0.7060	0.7388	0.2877	0.3153	0.2007	0.2098
FactFormer (Li et al., 2024b)	0.2962	0.8102	0.9249	0.2879	0.3287	0.5446	0.4842
FFNO (Tran et al., 2023)	0.2136	0.9484	1.0218	1.1791	1.1565	0.2374	0.1678
FNO (Li et al., 2021)	0.3179	0.5210	0.6253	0.2282	0.2574	0.4157	0.9032
SINO (ours)	**0.0122**	**0.0110**	**0.0303**	**0.0031**	**0.0171**	**0.0008**	**0.0097**

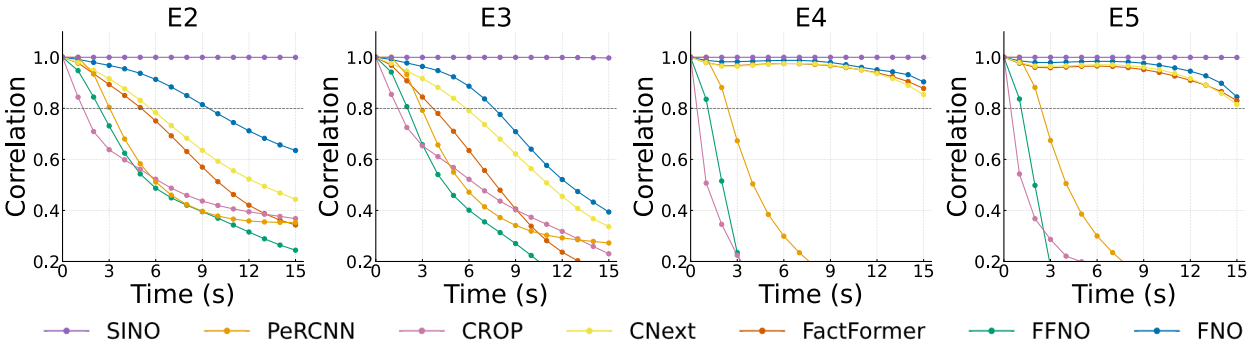
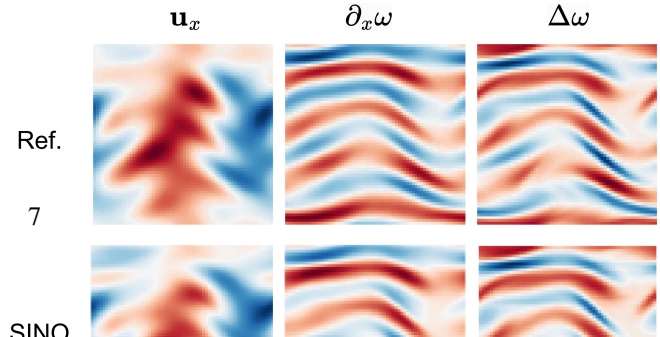


Figure 4: **Propagation curves for correlation across E2-E5 on NSEs.** Here, we provide the first 10 s of data during training, while testing extends to 15 s.

FNO and CNext perform best among the data-driven baselines but still cannot resolve vortices across multiple scales. Quantitatively, SINO improves upon these methods by 1-2 orders of magnitude (Table 1). Furthermore, we find that certain feature maps from SINO's Π-block and linear block align closely with ground-truth operators in NSEs (e.g., $\partial_x \omega$, $\Delta\omega$), providing strong evidence that SINO implicitly learns physically meaningful differential operators within the spectral learning block (Fig. 5).

Burgers (E6-E7). For the 2D Burgers' equation, we observe that PeRCNN and PeSANet perform reasonably well due to their hard-coded local operator representations, which align naturally with the equation's local advection-diffusion structure. Nevertheless, thanks



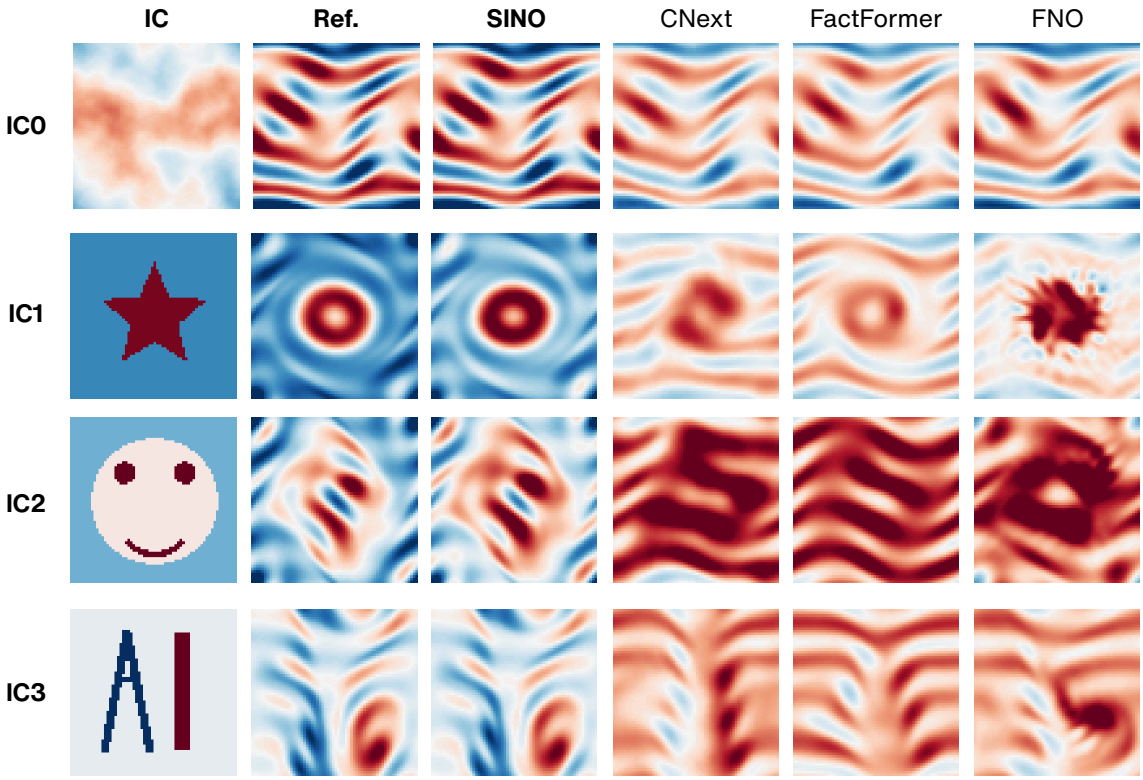


Figure 6: **In-distribution vs. OOD generalization on NSEs (E2).** Comparison between SINO trained with only **5** trajectories and data-driven baselines trained with **1000** trajectories. IC0 is sampled from the same distribution as the training set (in-distribution), while IC1-IC3 are sampled from OOD initial conditions. Shown are the vorticity fields after 15 s of evolution.

to the spectral accuracy inherited from its design, SINO achieves nearly two orders of magnitude lower error (Table 1). In the 3D case, only FFNO and SINO are able to preserve all three velocity components in close agreement with the reference (Fig. 3). However, SINO better captures small-scale shocks and dissipation, producing solutions that remain accurate and physically consistent across all three components.

5.3 OOD generalization to different ICs

OOD generalization is a fundamental open challenge in operator learning, as models trained on limited distributions often fail when confronted with novel initial conditions or other new physics (Wang et al., 2024a; Zhang et al., 2025b). To further evaluate SINO's generalization ability, we compare it with data-driven baselines on the NSE case (E2). SINO is trained with only **5** trajectories, while baseline methods (including CNext, FactFormer, and FNO) are trained with **1000** trajectories. We test on four types of initial conditions: IC0, sampled from the same Gaussian random field distribution as the training set (in-distribution), and three OOD conditions, including IC1 (star), IC2 (smiley face), and IC3 (the pattern 'AI'). As shown in Fig. 6, all models perform reasonably well on the in-distribution case (IC0), but baselines fail to produce meaningful predictions on OOD cases, collapsing into unrealistic fields. In contrast, SINO accurately tracks the long-term dynamics for all OOD conditions, with results closely aligned with ground-truth even at the final timestep. This demonstrates that SINO does not rely on data memorization but genuinely learns the underlying operator mapping functions to functions. Additional rollout trajectories are provided in Appendix C.6.

5.4 Ablation studies

We conduct ablation experiments on three systems to evaluate the contributions of individual modules in Table 2. The experiments are conducted on three tasks: KSE (E1), NSE (E4), and Burgers (E6). Replacing the Π-block with a linear combination (**SINO \ Π-block**) weakens overall performance, particularly for the NSE, which relies on crucial nonlinear terms. Removing the low-pass filter (**SINO \ Filter**) introduces instability in the model's rollout, with aliasing effects becoming most pronounced in the NSE system. Additionally, substituting the Freq2Vec module with a learnable vector (**SINO \ Freq2Vec**) or removing the linear block (**SINO \ Linear**) also significantly degraded the model's performance. Replacing the RK4 scheme with an Euler integrator (**SINO \ RK4**) further reduces accuracy and long-term stability, confirming that high-order temporal integration is critical for preserving accurate dynamics over extended rollouts. These experiments demonstrate that every component of the SINO design is indispensable, and it is their joint effect that endows SINO with high accuracy and strong generalization performance.

Table 2: **Relative ℓ_2 error for ablation studies.**

Model	KSE	NSE	Burgers
SINO \ Π-block	0.4911	0.4028	0.2287
SINO \ Filter	0.0203	NaN	0.0063
SINO \ Freq2Vec	NaN	NaN	0.0093
SINO \ Linear	NaN	NaN	NaN
SINO \ RK4	0.0287	NaN	0.0049
SINO	**0.0122**	**0.0031**	**0.0008**

5.5 Additional numerical results

In Appendix C, we provide further discussions and experiments beyond the main results. These include: (i) analysis of how training data quantity affects SINO's performance, (ii) parameter sensitivity studies, (iii) evaluations of zero-shot discretization invariance, (iv) inference speed benchmarks, (v) evolution of training loss and validation error, and (vi) additional trajectory predictions of physical fields for more comprehensive visualization.

6 Conclusion

We presented SINO, a spectral-inspired neural operator designed to model PDE-governed systems from limited data without requiring explicit knowledge of governing equations. By leveraging frequency-domain embeddings through the proposed Freq2Vec module and incorporating multiplicative interactions via the Π-block with de-aliasing, SINO provides a compact yet expressive architecture that balances accuracy and generalization. Theoretically, we established its universal approximation ability for a broad class of PDE operators. Empirically, SINO consistently outperforms both data-driven and physics-encoded baselines across diverse benchmarks, achieving 1-2 orders of magnitude lower error while maintaining robust OOD generalization. Our current study mainly focuses on Cartesian grids, and one promising direction is to extend SINO beyond regular domains to more complex geometries, which may be achieved through coordinate transformations that map irregular domains into regular ones, or by selecting appropriate basis functions tailored to the corresponding geometry.

Ethics Statement

This work focuses on developing a general neural operator framework (SINO) for learning PDE-governed dynamics from scarce data. Our study does not involve human subjects, sensitive personal data, or applications with immediate societal harm.

Reproducibility Statement

We have made extensive efforts to ensure the reproducibility of our results. The model architecture and training settings are described in Section 3, with dataset generation details provided in Appendix B.1. Theorem proofs are included in Appendix A, and extended experimental results are provided in Appendix C. Our model follows standard machine learning practices, where all hyperparameters are tuned based on validation performance, ensuring fair and consistent model selection. **Source codes will be released publicly upon acceptance.**

LLM Usage Statement

Large Language Models (LLMs) such as ChatGPT were used in this work for language editing (e.g., grammar checking, improving readability, and converting text to LaTeX format) and for generating initial code templates in Python.

All scientific ideas, model designs, theoretical results, experiments, and analysis were conceived, implemented, and validated solely by the authors.

References

Kamyar Azizzadenesheli, Nikola Kovachki, Zongyi Li, Miguel Liu-Schiaffini, Jean Kossaifi, and Anima Anandkumar. Neural operators for accelerating scientific simulations and design. *Nature Reviews Physics*, 6(5):320–328, 2024.

Johannes Brandstetter, Daniel E. Worrall, and Max Welling. Message passing neural PDE solvers. In *International Conference on Learning Representations*, pp. 1–9, 2022. URL https://openreview.net/forum?id=vSix3HPYKSU.

Gideon Dresdner, Dmitrii Kochkov, Peter Christian Norgaard, Leonardo Zepeda-Nunez, Jamie Smith, Michael Brenner, and Stephan Hoyer. Learning to correct spectral methods for simulating turbulent flows. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=wNBARGxoJn.

Yiheng Du, Nithin Chalapathi, and Aditi S. Krishnapriyan. Neural spectral methods: Self-supervised learning in the spectral domain. In *International Conference on Learning Representations*, pp. 1–9, 2024. URL https://openreview.net/forum?id=2DbVeuoa6a.

Salah A Faroughi, Nikhil Pawar, Celio Fernandes, Maziar Raissi, Subasish Das, Nima K Kalantari, and Seyed Kourosh Mahjour. Physics-guided, physics-informed, and physics-encoded neural networks in scientific computing. *arXiv preprint arXiv:2211.07377*, 2022.

Wenhan Gao, Ruichen Xu, Yuefan Deng, and Yi Liu. Discretization-invariance? on the discretization mismatch errors in neural operators. In *International Conference on Learning Representations*, pp. 1–9, 2025. URL https://openreview.net/forum?id=J9FgrqOOni.

Marius Ghergu and Vicentiu Radulescu. *Nonlinear PDEs: Mathematical models in biology, chemistry and population genetics*. Springer Science & Business Media, 2011.

Zhou Hang, Yuezhou Ma, Haixu Wu, Haowen Wang, and Mingsheng Long. Unisolver: Pde-conditional transformers are universal pde solvers. *arXiv preprint arXiv:2405.17527*, 2024.

Zhongkai Hao, Chang Su, Songming Liu, Julius Berner, Chengyang Ying, Hang Su, Anima Anandkumar, Jian Song, and Jun Zhu. DPOT: Auto-regressive denoising operator transformer for large-scale PDE pre-training. In *International Conference on Machine Learning*, volume 703, pp. 17616 – 17635. PMLR, 2024.

James R Holton and Gregory J Hakim. *An introduction to dynamic meteorology*, volume 88. Academic Press, 2013.

Peiyan Hu, Rui Wang, Xiang Zheng, Tao Zhang, Haodong Feng, Ruiqi Feng, Long Wei, Yue Wang, Zhi-Ming Ma, and Tailin Wu. Wavelet diffusion neural operator. In *International Conference on Learning Representations*, pp. 1–9, 2025. URL https://openreview.net/forum?id=FQhDIGuaJ4.

Xinquan Huang, Wenlei Shi, Qi Meng, Yue Wang, Xiaotian Gao, Jia Zhang, and Tie-Yan Liu. Neuralstagger: accelerating physics-constrained neural pde solver with spatial-temporal decomposition. In *International Conference on Machine Learning*, pp. 13993–14006. PMLR, 2023.

C Jayaprakash, F Hayot, and Rahul Pandit. Universal properties of the two-dimensional kuramoto-sivashinsky equation. *Physical Review Letters*, 71(1):12, 1993.

Chen Jingrun, Chi Xurong, E Weinan, and Yang Zhouwang. Bridging traditional and machine learning-based algorithms for solving pdes: the random feature method. *Journal of Machine Learning*, 1(3):268–298, 2022. ISSN 2790-2048.

Dmitrii Kochkov, Jamie A Smith, Ayya Alieva, Qing Wang, Michael P Brenner, and Stephan Hoyer. Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21):e2101784118, 2021.

AG Kravchenko and Parviz Moin. On the effect of numerical errors in large eddy simulations of turbulent flows. *Journal of Computational Physics*, 131(2):310–322, 1997.

Aditi Krishnapriyan, Amir Gholami, Shandian Zhe, Robert Kirby, and Michael W Mahoney. Characterizing possible failure modes in physics-informed neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 26548–26560, 2021.

Jae Yong Lee, SungWoong CHO, and Hyung Ju Hwang. HyperdeepONet: learning operator with complex target function space using the limited resources via hypernetwork. In *International Conference on Learning Representations*, pp. 1–9, 2023.

Xin Li, Chengli Zhao, Xue Zhang, and Xiaojun Duan. Symbolic neural ordinary differential equations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18511–18519, 2025.

Zeyu Li, Wang Han, Yue Zhang, Qingfei Fu, Jingxuan Li, Lizi Qin, Ruoyu Dong, Hao Sun, Yue Deng, and Lijun Yang. Learning spatiotemporal dynamics with a pretrained generative model. *Nature Machine Intelligence*, 6(12): 1566–1579, 2024a.

Zijie Li, Dule Shu, and Amir Barati Farimani. Scalable transformer for pde surrogate modeling. In *Advances in Neural Information Processing Systems*, volume 1216, pp. 28010 – 28039, 2024b.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, pp. 1–9, 2021.

Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *ACM/JMS Journal of Data Science*, 1(3):1–27, 2024c.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11976–11986, 2022.

Zichao Long, Yiping Lu, and Bin Dong. Pde-net 2.0: Learning pdes from data with a numeric-symbolic hybrid deep network. *Journal of Computational Physics*, 399:108925, 2019. ISSN 0021-9991. doi:10.1016/j.jcp.2019.108925. URL http://dx.doi.org/10.1016/j.jcp.2019.108925.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021.

Nick McGreivy and Ammar Hakim. Weak baselines and reporting biases lead to overoptimism in machine learning for fluid-related partial differential equations. *Nature Machine Intelligence*, 6(10):1256–1269, 2024.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina Agocs, Miguel Beneitez, Marsha Berger, Blakesly Burkhart, Stuart Dalziel, Drummond Fielding, et al. The well: a large-scale collection of diverse physics simulations for machine learning. *Advances in Neural Information Processing Systems*, 37:44989–45037, 2024.

Alessandro Parente and Nedunchezhian Swaminathan. Data-driven models and digital twins for sustainable combustion technologies. *Iscience*, 27(4), 2024.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W Battaglia. Learning Mesh-Based Simulation with Graph Networks. In *International Conference on Learning Representations*, pp. 1–9, 2021.

Richard H Pletcher, John C Tannehill, and Dale Anderson. *Computational fluid mechanics and heat transfer*. CRC Press, 2012.

Maziar Raissi, Paris Perdikaris, and George 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, 2019.

Chengping Rao, Pu Ren, Yang Liu, and Hao Sun. Discovering nonlinear PDEs from scarce data with physics-encoded learning. In *International Conference on Learning Representations*, pp. 1–9, 2022.

Chengping Rao, Pu Ren, Qi Wang, Oral Buyukozturk, Hao Sun, and Yang Liu. Encoding physics to learn reaction–diffusion processes. *Nature Machine Intelligence*, 5(7):765–779, 2023.

Philip G Saffman. *Vortex dynamics*. Cambridge University Press, 1995.

Jie Shen, Tao Tang, and Li-Lian Wang. *Spectral methods: algorithms, analysis and applications*, volume 41. Springer Science & Business Media, 2011.

Zhiqing Sun, Yiming Yang, and Shinjae Yoo. A neural pde solver with temporal stencil modeling. In *International Conference on Machine Learning*, pp. 33135–33155. PMLR, 2023.

Alasdair Tran, Alexander Mathews, Lexing Xie, and Cheng Soon Ong. Factorized fourier neural operators. In *International Conference on Learning Representations*, pp. 1–9, 2023. URL https://openreview.net/forum?id=tmIiMPl4IPa.

Simone Venturi and Tiernan Casey. Svd perspectives for augmenting deeponet flexibility and interpretability. *Computer Methods in Applied Mechanics and Engineering*, 403:115718, 2023.

Han Wan, Rui Zhang, Qi Wang, Yang Liu, and Hao Sun. Pesanet: Physics-encoded spectral attention network for simulating pde-governed complex systems. In *International Joint Conference on Artificial Intelligence*, pp. 1–7, 2025.

Haixin Wang, Yadi Cao, Zijie Huang, Yuxuan Liu, Peiyan Hu, Xiao Luo, Zezheng Song, Wanjia Zhao, Jilin Liu, Jinan Sun, et al. Recent advances on machine learning for computational fluid dynamics: A survey. *arXiv preprint arXiv:2408.12171*, 2024a.

Haixin Wang, Jiaxin LI, Anubhav Dwivedi, Kentaro Hara, and Tailin Wu. BENO: Boundary-embedded neural operators for elliptic PDEs. In *International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=ZZTkLDRmkg.

Qi Wang, Pu Ren, Hao Zhou, Xin-Yang Liu, Zhiwen Deng, Yi Zhang, Ruizhi Chengze, Hongsheng Liu, Zidong Wang, Jian-Xun Wang, Ji-Rong Wen, Hao Sun, and Yang Liu. P^2C^2Net: PDE-Preserved Coarse Correction Network for efficient prediction of spatiotemporal dynamics. In *Advances in Neural Information Processing Systems*, volume 38, 2024c.

Sifan Wang, Hanwen Wang, and Paris Perdikaris. Learning the solution operator of parametric partial differential equations with physics-informed deeponets. *Science Advances*, 7(40):eabi8605, 2021.

Gege Wen, Zongyi Li, Kamyar Azizzadenesheli, Anima Anandkumar, and Sally M Benson. U-fno—an enhanced fourier neural operator-based deep-learning model for multiphase flow. *Advances in Water Resources*, 163:104180, 2022.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for pdes on general geometries. In *International Conference on Machine Learning*, volume 2200, pp. 53681 – 53705, 2024.

Mengtao Yan, Qi Wang, Haining Wang, Ruizhi Chengze, Yi Zhang, Hongsheng Liu, Zidong Wang, Fan Yu, Qi Qi, and Hao Sun. Learnable-differentiable finite volume solver for accelerated simulation of flows. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 3471–3482, 2025.

Rui Zhang, Qi Meng, and Zhi-Ming Ma. Deciphering and integrating invariants for neural operator learning with various physical mechanisms. *National Science Review*, 11(4):nwad336, 2024.

Rui Zhang, Qi Meng, Rongchan Zhu, Yue Wang, Wenlei Shi, Shihua Zhang, Zhi-Ming Ma, and Tie-Yan Liu. Monte carlo neural pde solver for learning pdes via probabilistic representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(6):5059–5075, 2025a.

Xuan Zhang, Limei Wang, Jacob Helwig, Youzhi Luo, Cong Fu, Yaochen Xie, Meng Liu, Yuchao Lin, Zhao Xu, Keqiang Yan, et al. Artificial intelligence for science in quantum, atomistic, and continuum systems. *Foundations and Trends® in Machine Learning*, 18(4):385–912, 2025b.

Yuchen Zhang, Mingsheng Long, Kaiyuan Chen, Lanxiang Xing, Ronghua Jin, Michael I Jordan, and Jianmin Wang. Skilful nowcasting of extreme precipitation with nowcastnet. *Nature*, 619(7970):526–532, 2023.

Appendix Overview

A Proof of Theorem 1

Theorem 1 (Approximation ability of SINO) Let $\mathbf{u} : \Omega \to \mathbb{R}^C$ be a periodic function on the torus $\Omega = [0, 1)^d$, bandlimited to $\|\mathbf{k}\|_\infty \leq k_{\max}$. Suppose the right-hand side (RHS) of the PDE can be expressed as

$$\partial_t \mathbf{u} = \mathcal{R}(\mathbf{u}) := \sum_{j=1}^{J} \prod_{i=1}^{n_j} \mathcal{D}_{j,i}\mathbf{u},$$

where each $\mathcal{D}_{j,i}$ is a linear differential operator of finite order. Then for any $\epsilon > 0$, there exists a SINO model $\mathcal{S}_\theta$ such that

$$\|\mathcal{S}_\theta(\mathbf{u}) - \mathcal{R}(\mathbf{u})\|_{L^2} < \epsilon.$$

Proof 1 *Let us define a truncated version of* $\mathbf{u}$ *in the Fourier domain:*

$$\mathbf{u}_{k_{\max}}(\mathbf{x}) = \sum_{\|\mathbf{k}\|_\infty \leq k_{\max}} \hat{\mathbf{u}}(\mathbf{k}) e^{2\pi i \mathbf{k}\cdot\mathbf{x}}.$$

Since $\mathbf{u}$ *is bandlimited, we have* $\mathbf{u} = \mathbf{u}_{k_{\max}}$.

Each linear differential operator $\mathcal{D}_{j,i}$ *corresponds to a spectral multiplier:*

$$\mathcal{D}_{j,i}\mathbf{u}(\mathbf{x}) = \sum_{\|\mathbf{k}\|_\infty \leq k_{\max}} \psi_{j,i}(\mathbf{k})\hat{\mathbf{u}}(\mathbf{k}) e^{2\pi i \mathbf{k}\cdot\mathbf{x}},$$

where $\psi_{j,i}(\mathbf{k})$ *is a function of* $\mathbf{k}$.

Let $\widehat{\mathcal{D}}_{j,i}$ *denote the learned spectral operator in SINO. By the universal approximation theorem for neural networks, we can construct a neural network approximator* $\hat{\psi}_{j,i}(\mathbf{k})$ *for any* $\delta > 0$, *such that*

$$\sup_{\|\mathbf{k}\|_\infty \leq k_{\max}} \left|\hat{\psi}_{j,i}(\mathbf{k}) - \psi_{j,i}(\mathbf{k})\right| < \delta.$$

This implies element-wise approximation of Fourier multipliers, and consequently

$$\left\|\widehat{\mathcal{D}}_{j,i}\mathbf{u} - \mathcal{D}_{j,i}\mathbf{u}\right\|_{L^2} \leq C_{j,i}\delta,$$

Table 3: **Summary of experimental settings for different cases.** For the NSE case, models were trained on trajectories spanning 10 s and evaluated on prediction rollouts of 15 s. "Gen." denotes the resolution/time step used for generating high-fidelity data with spectral solvers. "DOL" and "POL" denotes the resolution/time step used for data-driven and physics-encoded operator learning.

Setting	**KSE**	**NSE**	**2D Burgers**	**3D Burgers**
Numerical Method	spectral method	spectral method	spectral method	spectral method
Spatial Domain	$[0, 12\pi)^2$	$[0, 1)^2$	$[0, 2\pi)^2$	$[0, 2\pi)^3$
Temporal Domain	$[0, 5]$	$[0, 10/15]$	$[0, 2]$	$[0, 5]$
Training Trajectories	2	5	5	5
Validation Trajectories	2	2	2	2
Test Trajectories	5	5	5	5
Gen. Grid / Δt	108^2, 1×10^{-4}	256^2, 1×10^{-4}	512^2, 1×10^{-3}	128^3, 5×10^{-3}
DOL Grid / Δt	54^2, 1	64^2, 1	128^2, 0.2	64^3, 0.5
POL Grid / Δt	54^2, 1×10^{-3}	64^2, 5×10^{-3}	128^2, 5×10^{-3}	64^3, 5×10^{-2}

where the constant $C_{j,i}$ depends on $\|\hat{\mathbf{u}}(\mathbf{k})\|_{L^2}$, which is finite due to bandlimiting.

Now, consider the nonlinear composition $\prod_{i=1}^{n_j} \mathcal{D}_{j,i}\mathbf{u}$ approximated by $\prod_{i=1}^{n_j} \widehat{\mathcal{D}}_{j,i}\mathbf{u}$. Because $\mathbf{u}$ is smooth and bounded, multiplication is continuous in L^2, implying that the error of the product can be controlled linearly in δ:, i.e.,

$$\left\| \prod_{i=1}^{n_j} \widehat{\mathcal{D}}_{j,i}\mathbf{u} - \prod_{i=1}^{n_j} \mathcal{D}_{j,i}\mathbf{u} \right\|_{L^2} \leq C_j\delta,$$

for some constant C_j depending on $\mathbf{u}$ and n_j.

Aggregating over all $j = 1, \ldots, J$ gives

$$\|\mathcal{S}_\theta(\mathbf{u}) - \mathcal{R}(\mathbf{u})\|_{L^2} \leq \sum_{j=1}^{J} C_j\delta = C'\delta.$$

Therefore, for any prescribed $\epsilon > 0$, choosing $\delta < \epsilon/C'$ and increasing the neural network capacity accordingly, we ensure

$$\|\mathcal{S}_\theta(\mathbf{u}) - \mathcal{R}(\mathbf{u})\|_{L^2} < \epsilon. \quad \blacksquare$$

On unbounded Fourier bandwidth. The theorem assumes that the input function $\mathbf{u}$ is bandlimited with $\|\mathbf{k}\|_\infty \leq k_{\max}$. In practice, many physical fields are not exactly bandlimited. However, if $\mathbf{u} \in C^m(\Omega)$, then its Fourier coefficients decay at the rate

$$|\hat{\mathbf{u}}(\mathbf{k})| \lesssim \frac{1}{\|\mathbf{k}\|^m},$$

which implies that the tail $\|\mathbf{u} - \mathbf{u}_{k_{\max}}\|_{L^2}$ decays as $O(k_{\max}^{-(m-d/2)})$. Thus, the total approximation error of SINO in this case can be bounded by

$$\|\mathcal{S}_\theta(\mathbf{u}) - \mathcal{R}(\mathbf{u})\|_{L^2} \leq \epsilon + C_2 k_{\max}^{-(m-d/2)}.$$

By choosing a sufficiently large $k_{\max}$ and network capacity, this bound can still be made arbitrarily small.

Supported PDE types. The class of PDEs covered by this theorem includes a broad family of physical systems, as long as the RHS can be expressed as a sum of products of linear differential operators. This encompasses reaction-diffusion equations, convection-diffusion equations, the incompressible Navier-Stokes equations, and other general polynomial-type PDEs.

B Experimental Details

B.1 Data Generation

Initial conditions for all PDE systems were sampled from a Gaussian Random Field (GRF), following the standard practice introduced in the classical FNO work (Li et al., 2021). We target the data-scarce scenario: each experiment

includes only 2-5 trajectories in the training set, as summarized in Table 3. The table specifies the numerical scheme used for data generation, the spatial and temporal domains, and the number of training and testing trajectories for each PDE case. For the NSE case, training trajectories span 10 s, while evaluation involves rollout prediction over 15 s, as indicated in the temporal domain row of Table 3. Numerical solutions were obtained using a $2/3$ dealiased pseudo-spectral method with periodic boundary conditions, implemented in Python with PyTorch, consistent with prior classical papers (Li et al., 2021; Tran et al., 2023). In particular, the NSE dataset was generated using an open-source implementation from (Li et al., 2021). To ensure reproducibility, we fixed random seeds for dataset splitting: seed 0 for training, seed 1 for validation, and seed 2 for testing.

B.2 Baseline Models

We provide detailed introductions to the baseline models used for comparison with SINO. These baselines include data-driven approaches such as FNO (Li et al., 2021), FFNO (Tran et al., 2023), FactFormer (Li et al., 2024b), CNext (Liu et al., 2022; Ohana et al., 2024) and CROP (Gao et al., 2025), as well as physics-encoded models including PeRCNN (Rao et al., 2023) and PeSANet (Wan et al., 2025). Details of baseline models are provided as follows:

Fourier Neural Operator (FNO) (Li et al., 2021). FNO is one of the most classical data-driven neural operator model that captures features in the frequency domain. Its ability to capture information in the frequency domain allows it to effectively utilize global information, and it also possesses a degree of resolution invariance.

For hyperparameter tuning, we conducted a grid search over the number of channels (12, 36, 64) and Fourier modes (12, 16), and selected the best configuration according to the performance on the validation set.

Factorized Fourier Neural Operator (FFNO) (Tran et al., 2023). FFNO is an improved neural operator model based on FNO, and its core involves the introduction of factorized Fourier representation. This factorization method and improved network structure allow FFNO to employ deeper network architectures and demonstrate performance superior to standard FNO in simulating various partial differential equations.

For hyperparameter tuning, we performed a grid search over the number of channels (16, 32) and Fourier modes (12, 16), selecting the best setting based on validation performance.

FactFormer (Li et al., 2024b). FactFormer is a transformer-based model designed for multi-dimensional settings that leverages an axial factorized kernel integral, implemented via a learnable projection operator that decomposes the input function into one-dimensional sub-functions.

For hyperparameter tuning, we searched over the hidden dimension (64, 128, 256) and network depth (4, 6), and selected the optimal configuration according to the validation set.

CNext (Liu et al., 2022; Ohana et al., 2024). CNext is a family of modernized ConvNet models, inspired by the design principles of Vision Transformers (ViTs) while retaining the efficiency and simplicity of convolutional architectures. In the latest benchmark study (Ohana et al., 2024), CNext achieved SOTA results across a wide range of PDE learning tasks.

For hyperparameter tuning, we conducted a grid search over the network width (16, 32, 64) and number of layers (2, 3, 4), and selected the best configuration according to validation performance.

Cross-Resolution Operator-learning Pipeline (CROP) (Gao et al., 2025). CROP is a method proposed to address problems with generalization and discretization mismatch error in existing neural operators across different data resolutions. Equipped with a specific pipeline design, CROP is able to eliminate aliasing effects and discretization mismatch errors, thus enabling efficient learning and inference across different resolutions.

For hyperparameter tuning, we conducted a grid search over the number of channels (8, 12, 36) and Fourier modes (12, 16), and selected the best configuration according to validation performance.

Physics-embedded Recurrent-Convolutional Neural Network (PeRCNN) (Rao et al., 2023). PeRCNN is a physics-encoded learning methodology that directly embeds physical laws into the neural network architecture. It employs multiple parallel convolutional neural network and leverages feature map multiplication to simulate polynomial equations, thereby enhancing the model's extrapolation and generalization capabilities.

For hyperparameter tuning, we performed a grid search over the network width (32, 64, 128) and input kernel size (5, 7), selecting the best configuration based on validation performance.

Physics-encoded Spectral Attention Network (PeSANet) (Wan et al., 2025). PeSANet includes a physics-encoded block for approximating local differential operators and a spectral-enhanced block which, combined with spectral

Table 4: **Parameter counts of different models.** NaN denotes 'Not a Number', and NA denotes 'Not Applicable'.

Model	KSE	NSE				Burgers	
	E1	E2 $10^{-4}, f_1$	E3 $10^{-5}, f_1$	E4 $10^{-4}, f_2$	E5 $10^{-5}, f_2$	E6 2D	E7 3D
FNO (Li et al., 2021)	0.59M	0.33M	9.46M	0.33M	3.00M	9.46M	127.41M
FFNO (Tran et al., 2023)	17.85M	0.48M	0.48M	13.65M	0.48M	4.33M	62375
FactFormer (Li et al., 2024b)	11.26M	11.26M	2.95M	1.81M	2.95M	0.65M	0.30M
CNext (Ohana et al., 2024)	50977	51.35M	51.35M	12.84M	0.84M	0.22M	4.16M
CROP (Gao et al., 2025)	2.76M	0.56M	0.56M	0.56M	0.56M	2.76M	22.13M
PeSANet (Wan et al., 2025)	89162	NaN	NaN	NaN	NaN	89262	NA
PeRCNN (Rao et al., 2023)	10116	10116	10116	10116	10116	19716	1924
SINO (ours)	**1708**	**9278**	**9278**	**9278**	**9278**	**11133**	**2951**

attention, captures global features in the frequency domain, allowing it to perform excellently, especially in long-term forecasting accuracy, under scarce data and incomplete physical priors.

For hyperparameter tuning, we searched over the network width (32, 64, 128) and input kernel size (5, 7), and selected the optimal configuration according to validation performance.

For the proposed **SINO**, we conducted a grid search over the number of channels (16, 32, 64) and the dimension of $\psi(\mathbf{k})$ in the Freq2Vec (4, 6, 8), selecting the best configuration according to validation performance.

Our model follows standard machine learning practices, where all hyperparameters are tuned based on validation performance with fixed seed, ensuring fair and consistent model selection. The parameter sizes of all baseline models and SINO are summarized in Table 4.

B.3 Metrics

To assess the performance of our proposed method, we utilize two evaluation metrics: relative ℓ_2 error and correlation. The relative ℓ_2 error measures the ratio of the ℓ_2 norm of the error vector to that of the ground-truth vector, providing a dimensionless measure of the prediction error relative to the true scale. Correlation is quantified using the Pearson correlation coefficient (PCC), which measures the linear dependence between predicted and ground-truth solutions over time.

The definitions are as follows:

$$\begin{aligned} \text{Relative } \ell_2 \text{ Error:} \quad & \frac{\|\mathbf{y}-\tilde{\mathbf{y}}\|_2}{\|\mathbf{y}\|_2} = \sqrt{\frac{\sum_{i=1}^{n}(y_i-\tilde{y}_i)^2}{\sum_{i=1}^{n} y_i^2}}, \\ \text{Correlation (PCC):} \quad & \text{PCC}(\mathbf{y},\tilde{\mathbf{y}}) = \frac{\text{Cov}(\mathbf{y},\tilde{\mathbf{y}})}{\sigma_{\mathbf{y}}\sigma_{\tilde{\mathbf{y}}}}. \end{aligned} \tag{10}$$

Here, n denotes the number of evaluation points, y_i and $\tilde{y}_i$ are the i-th components of the ground-truth vector $\mathbf{y}$ and the predicted vector $\tilde{\mathbf{y}}$, respectively, Cov is the covariance, and σ denotes the standard deviation.

B.4 Training Details

All experiments were conducted on a single NVIDIA A100 GPU (80GB) with an Intel(R) Xeon(R) Platinum 8380 CPU (2.30GHz, 64 cores). For 2D cases, models were trained for 20000 iterations, while for 3D cases, 5000 iterations were used. We adopt a OneCycle learning rate scheduler with a maximum learning rate selected from $\{0.01, 0.001\}$, scheduled over the entire training horizon.

During training, we randomly sample a starting position along each trajectory. From this state, the model first evolves forward n steps without gradients (warm-up), where n is randomly sampled from $\{0, \ldots, n_1\}$ at each iteration, and then predicts the following n_2 steps with gradients, which are compared against the ground truth. For physics-encoded methods, we fix $n_1 = 4$ and $n_2 = 8$; for data-driven methods, due to their larger effective step size, we set $n_1 = 2$ and $n_2 = 2$. This scheme provides stable backpropagation and encourages the models to generalize across different rollout horizons (Brandstetter et al., 2022).

C Additional Numerical Results

C.1 Data Efficiency

To evaluate the data efficiency and scalability of SINO, we conduct experiments on KSE (E1) with varying numbers of training trajectories. The results in Table 5 reveal two key observations. First, SINO achieves strong performance even when trained on a single trajectory, with a relative ℓ_2 error of 0.0430. This highlights the model's ability to capture complex dynamics from extremely limited data, which is crucial in data-scarce scientific domains. Second, SINO scales effectively with additional data: as the number of training trajectories increases from 1 to 8, the relative ℓ_2 error decreases monotonically from 0.0430 to 0.0122. Although performance slightly saturates beyond 8 trajectories, the trend confirms that SINO can leverage more data to enhance predictive accuracy. Overall, these results demonstrate that SINO is both data-efficient and scalable: it delivers accurate predictions from minimal data while benefiting from larger training sets.

Table 5: **Relative ℓ_2 error of SINO with varying training dataset sizes on KSE (E1).**

Number of Trajectories	1	2	4	8	16
Relative ℓ_2 Error	0.0430	0.0285	0.0175	0.0122	0.0139

C.2 Hyperparameter Sensitivity Analysis

Table 6 reports the sensitivity of SINO to two key hyperparameters: the channel size ($C \in \{64, 32, 16\}$) and the embedding dimension of $\psi(\mathbf{k})$ in the Freq2Vec module ($K \in \{8, 6, 4\}$). As expected, reducing either the number of channels or the embedding dimension degrades accuracy. For example, with $K = 8$, the relative ℓ_2 error increases from 0.0109 at $C = 64$ to 0.0593 at $C = 16$, while reducing K from 8 to 4 also significantly worsens performance. Nevertheless, across all tested settings, SINO maintains errors within a reasonable range, demonstrating robustness to hyperparameter choices.

Table 6: **Hyperparameter sensitivity.** Relative ℓ_2 error for different configurations.

Model	$C = 64$	$C = 32$	$C = 16$
$K = 8$	0.0109	0.0245	0.0593
$K = 6$	0.0392	0.0382	0.0457
$K = 4$	0.1089	0.1216	0.1406

C.3 Zero-shot Super-resolution

We further evaluate the zero-shot super-resolution capability of SINO, where the model is trained on the lowest-resolution data and directly tested on higher resolutions without retraining. As summarized in Table 7, SINO trained on 36×36 (KSE) or 32×32 (NSE) generalizes robustly to finer grids. In the KSE case (E1), the relative ℓ_2 error drops significantly from 0.1169 at low resolution to 0.0297 and 0.0283 when applied to 54×54 and 108×108 grids, respectively. In the NSE case, the model achieves errors of 0.0523, 0.0427, and 0.0424 when evaluated at 32×32, 64×64, and 128×128, respectively. These results highlight that SINO not only achieves discretization invariance but also benefits from higher-resolution inputs at inference, where the Freq2Vec module effectively extracts fine-scale spectral information to improve prediction accuracy.

Table 7: **Relative ℓ_2 error for different resolutions.**

Equation	**Low**	**Medium**	**High**
KSE (E1)	0.1169	0.0297	0.0283
NSE (E4)	0.0523	0.0427	0.0424

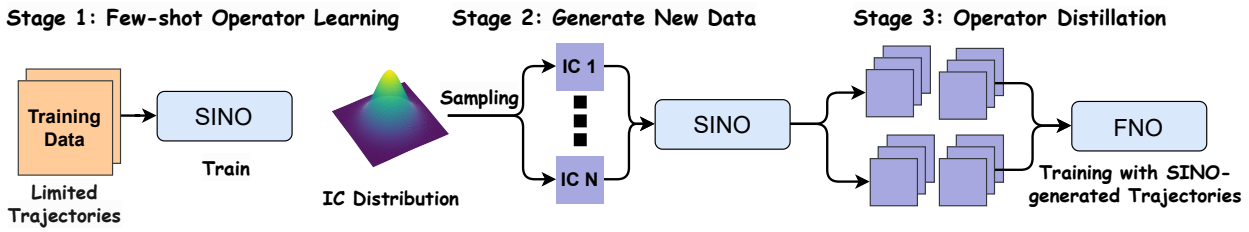


Figure 7: **Operator distillation.** The three stages of operator distillation include few-shot operator learning, synthetic data generation, and the training of SINO-FNO.

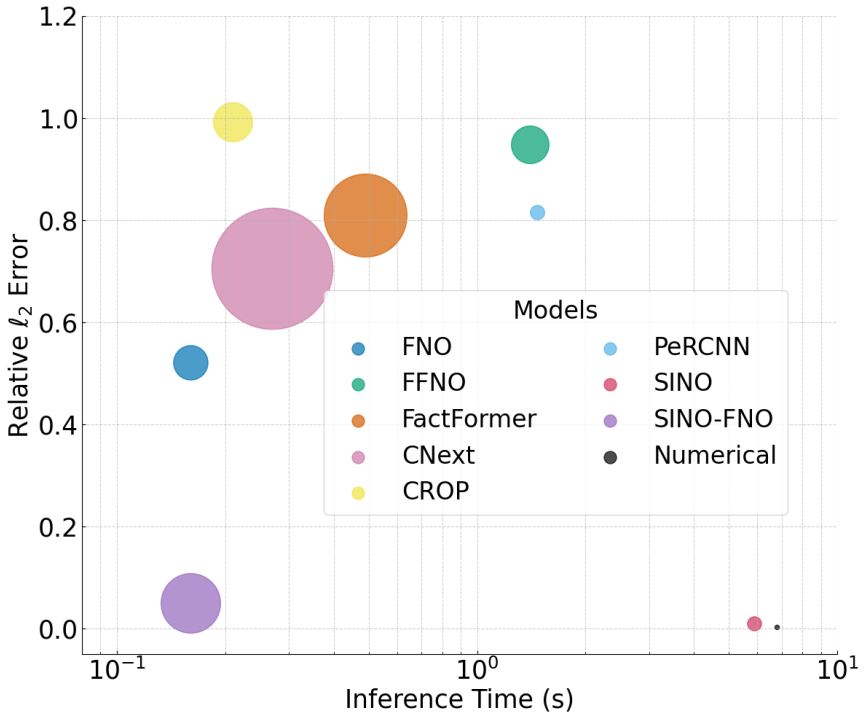


Figure 8: **Inference time vs. accuracy trade-off on NSE (E2).** Each bubble denotes a model, where the x-axis shows inference time (s, log scale), the y-axis shows relative ℓ_2 error, and the bubble size corresponds to parameter count.

C.4 Inference Speed and Operator Distillation

In this section, we discuss the inference speed of SINO. Similar to other physics-encoded methods (McGreivy & Hakim, 2024), the speed of SINO is constrained by its intrinsic physical structure. Since the focus of this work is on modeling complex systems under physics-agnostic settings, computational efficiency is not the primary objective. Nevertheless, it is possible to alleviate this bottleneck with a simple yet effective strategy.

To this end, we introduce an *operator distillation* approach that transfers the knowledge of a fine-step-size SINO model into a coarse-step-size neural operator, termed **SINO-FNO** (Fig. 7). The procedure consists of three stages. **(i)** We first train a SINO model from limited observations and use the resulting surrogate operator as the teacher network. **(ii)** Equipped with this teacher, we sample new initial conditions from the underlying distribution and generate a large synthetic dataset by rolling out trajectories with the SINO teacher. **(iii)** We then train an FNO as the student network on this synthetic dataset. Importantly, in the distillation loss we supervise the FNO to predict system states over a much coarser time step than the fine resolution used in the teacher SINO.

It is worth noting that, in the physics-agnostic scenarios, numerical solvers cannot be used to generate training data. Instead, SINO itself serves as a proxy simulator. Since SINO achieves 1-2 orders of magnitude lower error than data-driven neural operators, it provides a sufficiently accurate teacher, making this distillation strategy both feasible and effective.

We conduct an inference time analysis of SINO and SINO-FNO, comparing them against baseline neural operators and the traditional spectral method (chosen as a representative numerical solver due to its accuracy and efficiency on regular domains (McGreivy & Hakim, 2024)). The analysis is performed on the NSE (E2) case. All models are tested on the same spatial resolution; step sizes are kept consistent across data-driven approaches (including SINO-FNO), while physics-aware models follow the same setup as SINO. For the spectral solver, we adopt the largest stable time step to ensure fairness.

As shown in Fig. 8, SINO achieves the lowest relative ℓ_2 error among all neural operators, reaching accuracy comparable to the spectral method. Through operator distillation, SINO-FNO effectively reduces inference cost by more than two orders of magnitude (down to $\sim$0.2s), while retaining errors close to zero. This corresponds to a $\sim 36\times$ speedup compared to the spectral method, with accuracy far surpassing purely data-driven baselines.

C.5 Evolution of Training Loss and Validation Error

In this subsection, we present the training dynamics of SINO and other data-driven baselines on the NSE case (E3). As shown in Fig. 9, SINO maintains a consistent trend between training loss and validation error throughout the entire training process, both decreasing steadily without divergence. In contrast, data-driven baselines suffer from severe overfitting: while their training loss continues to decrease, the validation error increases, indicating poor generalization. These results further corroborate our discussion in Section 4, highlighting that SINO achieves robust generalization rather than mere data memorization.

C.6 Additional Rollout Trajectories

In this section, we provide additional rollout trajectories to complement the additional results in the main text. Fig. 10 illustrates long-term predictions on the NSE cases (E2-E5). Figs. 11-14 further examine generalization across different initial conditions, including both in-distribution (IC0) and out-of-distribution (IC1-IC3) scenarios.

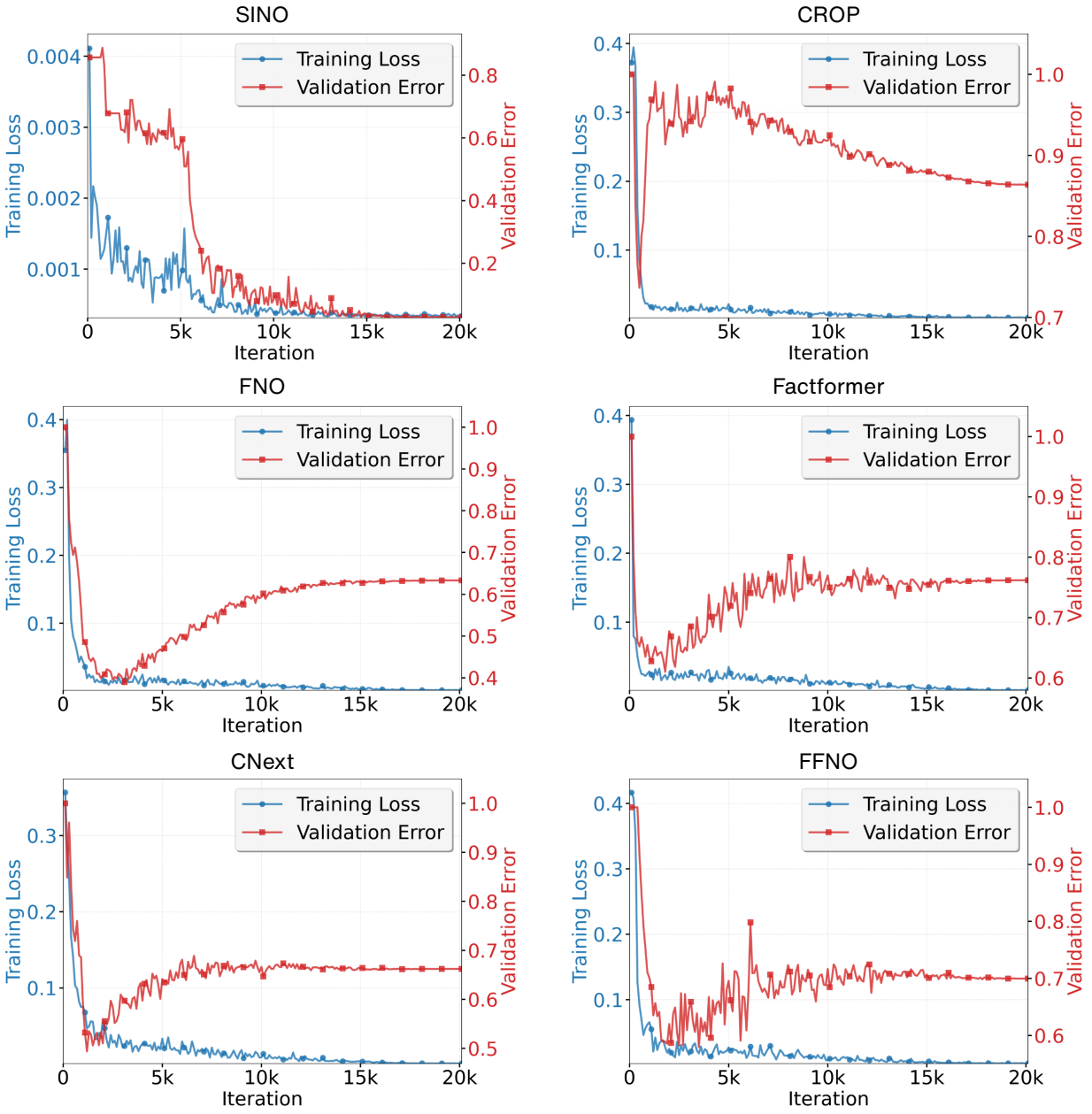


Figure 9: **Training and validation dynamics on NSE (E3).** Comparison of SINO with other data-driven baselines.

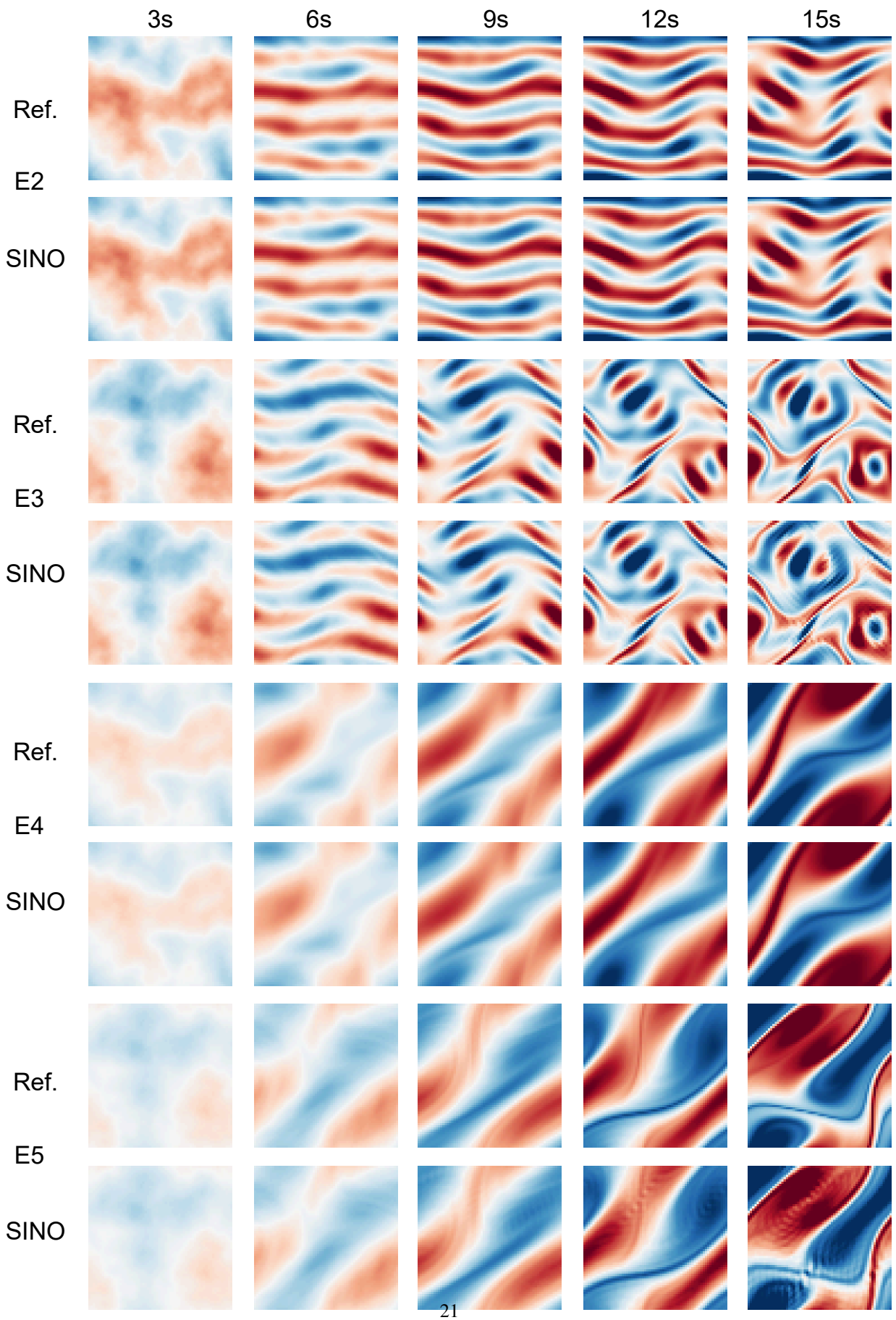


Figure 10: **Predicted trajectory of SINO.** Comparison of the vorticity field prediction trajectories for different scenarios (E2 to E5). The columns represent different prediction time steps at 3 s, 6 s, 9 s, 12 s, and 15 s.

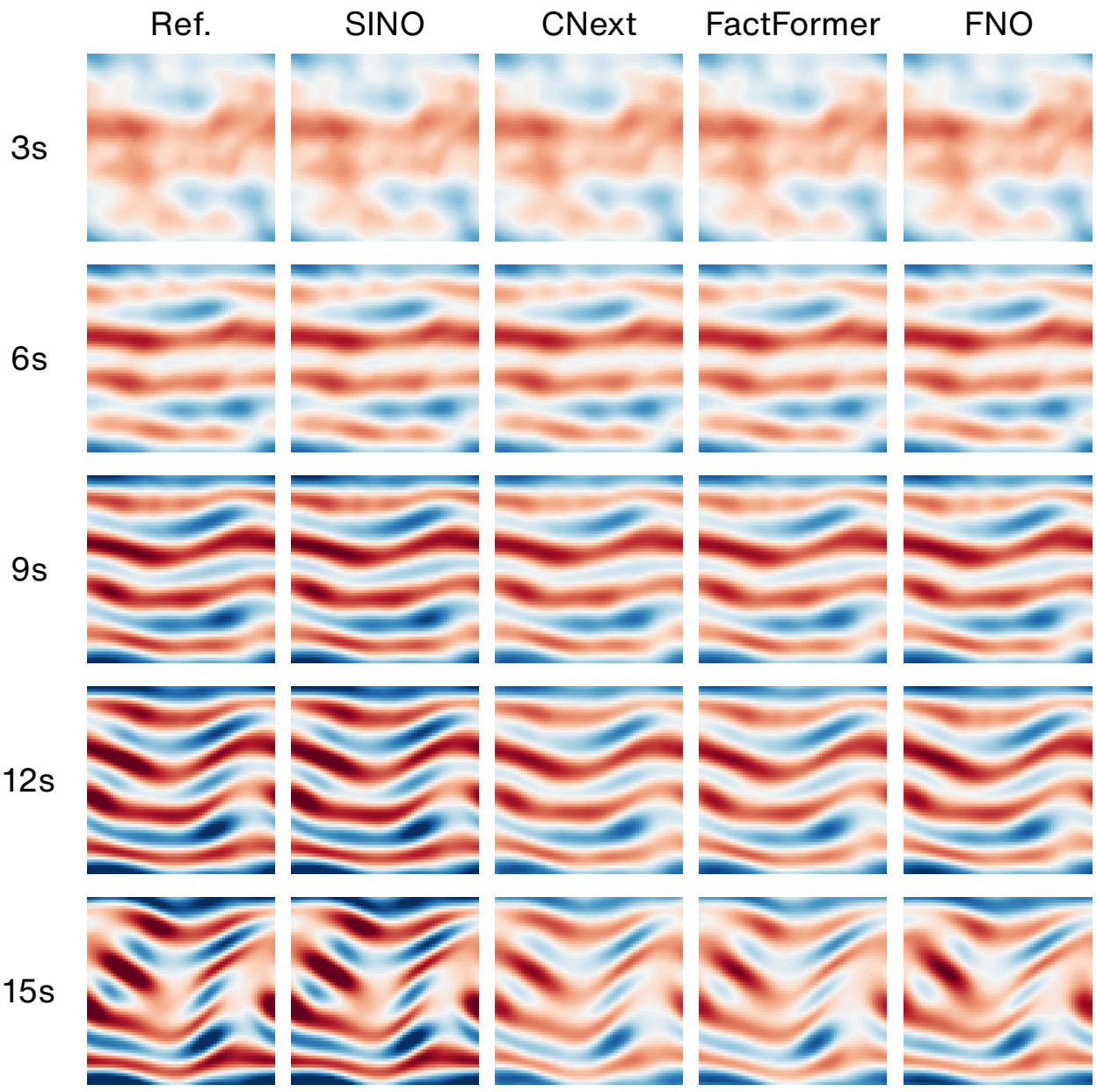


Figure 11: **In-distribution trajectories on IC0.** Comparison between SINO trained with only **5** trajectories and data-driven baselines trained with **1000** trajectories.

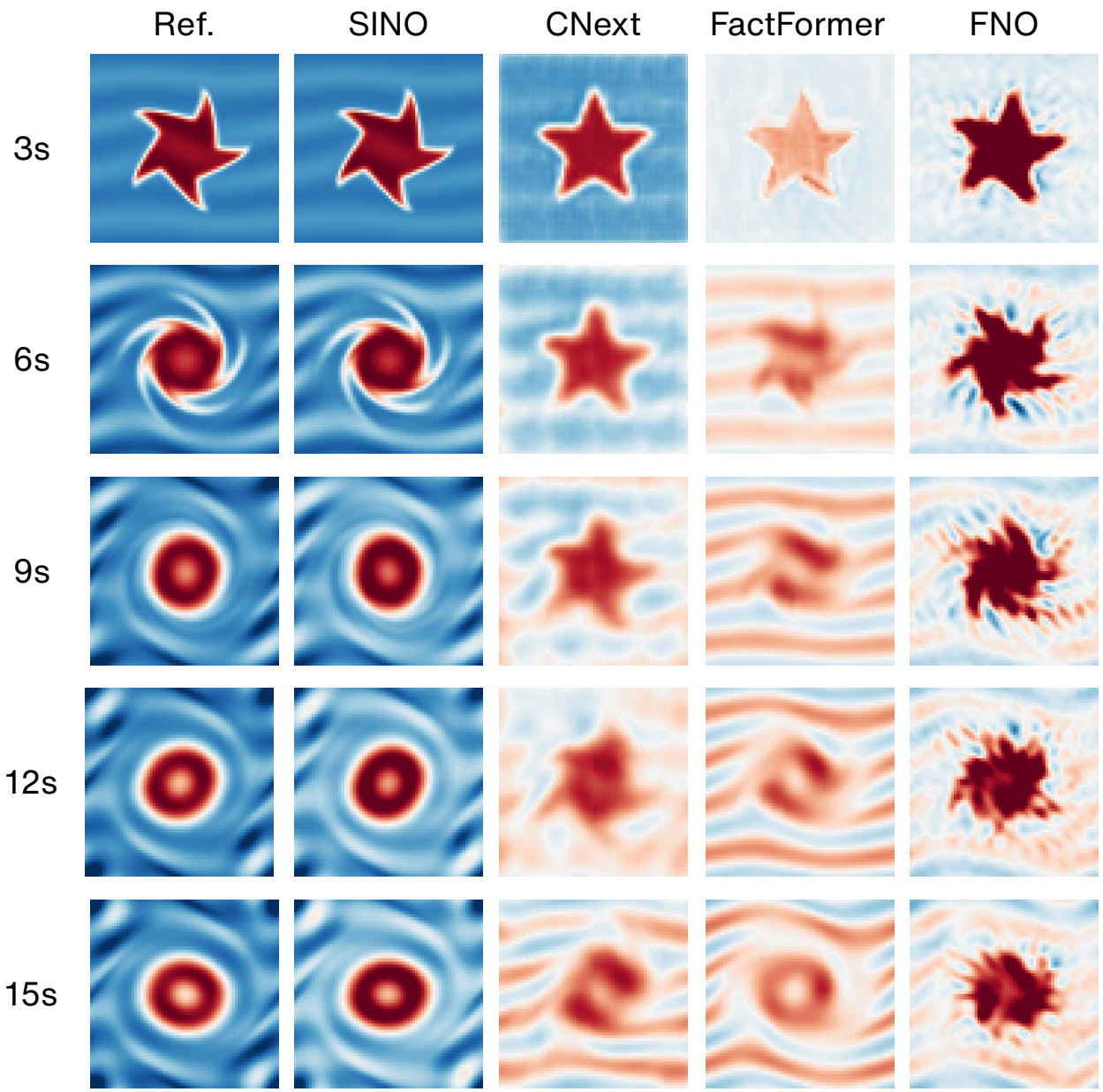


Figure 12: **OOD generalization trajectories on IC1 (star).** Comparison between SINO trained with only **5** trajectories and data-driven baselines trained with **1000** trajectories.

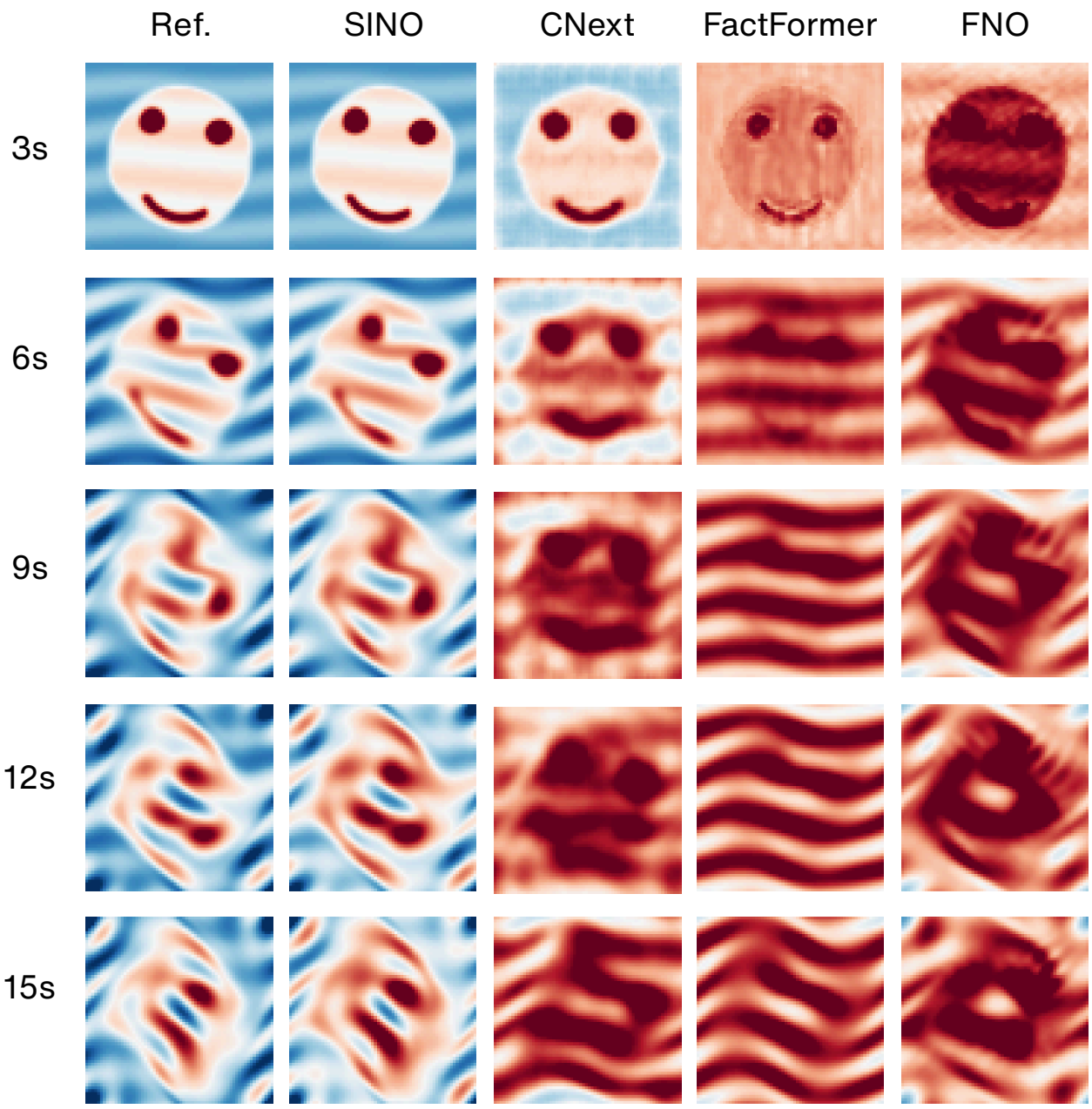


Figure 13: **OOD generalization trajectories on IC2 (smiley face).** Comparison between SINO trained with only **5** trajectories and data-driven baselines trained with **1000** trajectories.

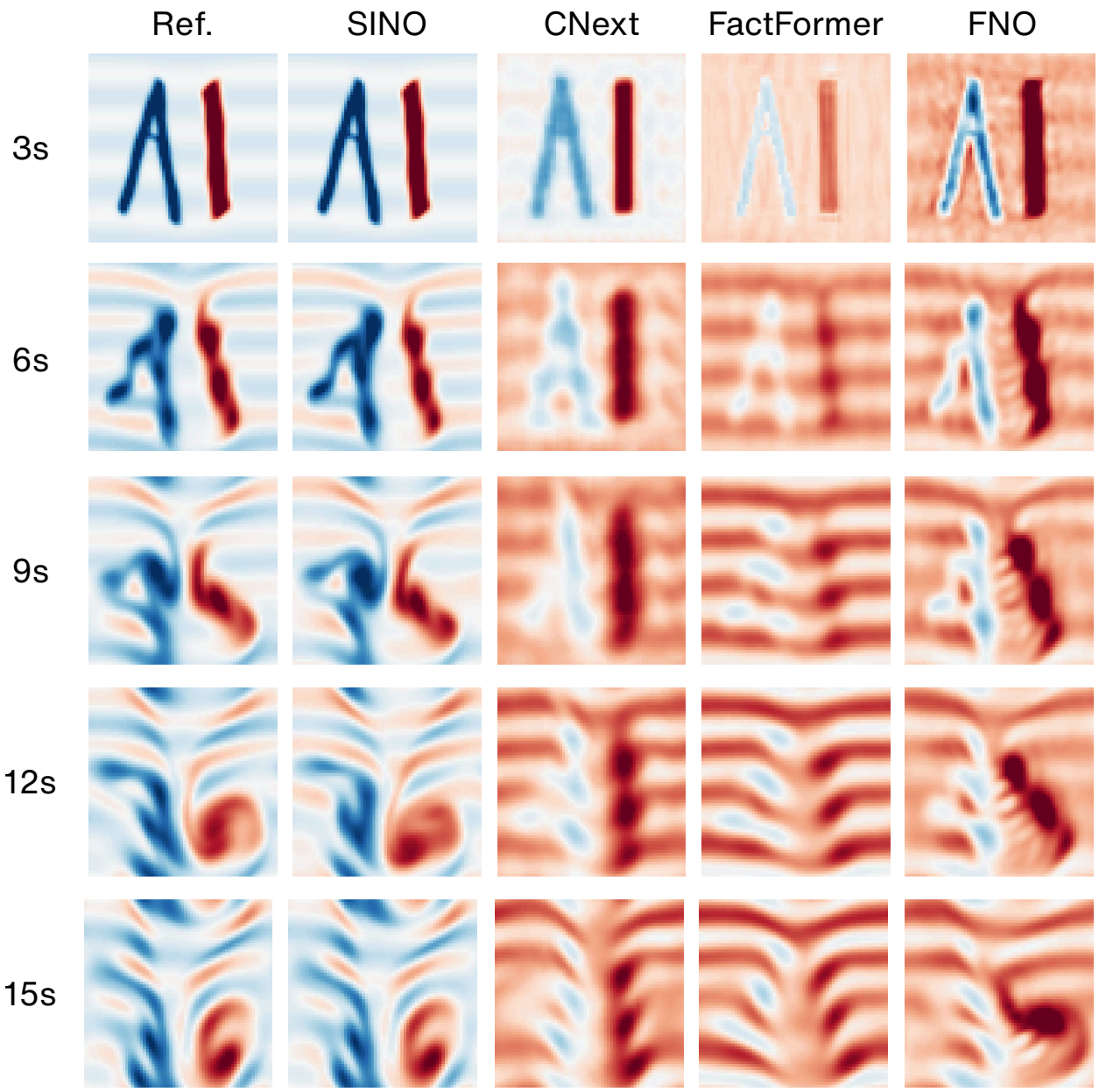


Figure 14: **OOD generalization trajectories on IC3 (the pattern 'AI').** Comparison between SINO trained with only **5** trajectories and data-driven baselines trained with **1000** trajectories.