Can we pre-train ICL-based SFMs for the zero-shot inference of the 1D CDR problem with noisy data?

Mingu Kang* **Dongseok Lee*** **Woojin Cho** **Kookjin Lee** **Anthony Gruber**

Nathaniel Trask **Youngjoon Hong†** **Noseong Park†**

Abstract

Recent advancements in scientific machine learning have begun to explore the potential of scientific foundation models (SFMs). Inspired by the in-context learning (ICL) framework of large language models (LLMs), we leverage prior data and pre-training techniques to construct our SFM. It has been demonstrated that ICL in LLMs can perform Bayesian inference, resulting in strong generalization capabilities. Furthermore, LLMs do not exhibit intrinsic inductive bias; rather, they inherit bias from the prior data, as confirmed experimentally. Building upon these insights, our methodology is structured as follows: (i) we collect prior data in the form of solutions of partial differential equations (PDEs) constructed by an arbitrary linear combination of mathematical dictionaries, (ii) we utilize Transformer architectures with self-attention and cross-attention mechanisms to predict PDE solutions without knowledge of the governing equations in a zero-shot setting, and (iii) we provide experimental evidence on the one dimensional convection-diffusion-reaction equation, which demonstrate that pre-training remains robust even with noisy prior data, with only marginal impacts on test accuracy. Notably, this finding opens the path to pre-training SFMs with realistic, low-cost data instead of, or in conjunction with, numerical high-cost data. These results support the conjecture that SFMs can improve in a manner similar to LLMs, where fully cleaning the vast set of sentences crawled from the Internet is nearly impossible.

1 Introduction

In recent years, large language models (LLMs) have revolutionized the field of natural language processing by introducing highly flexible and scalable architectures [5, 19, 37, 13, 10]. Notably, the in-context learning (ICL) paradigm has demonstrated powerful generalization capabilities, enabling LLMs to adapt to new tasks without explicit fine-tuning [5, 31, 11, 15]. This success has motivated the application of such foundation models across a variety of domains [43, 42, 44]. Scientific Machine Learning (SML) is one such emerging domain which merges physics-based models with machine learning methodologies [32, 41, 36, 21, 20, 9]. SML aims to leverage the power of machine learning to solve complex scientific problems, including those governed by partial differential equations (PDEs). Recent efforts in this direction have led to the development of foundation models specifically designed for scientific tasks, called Scientific Foundation Models (SFMs) [45, 42, 44, 24]. These models aim to generalize across a wide range of scientific problems using prior data, much like how LLMs generalize across various language tasks. For example, the versatility of in-context operator networks (ICONs), as illustrated in studies like [47] and [45], underscores their generalization capabilities in various PDE-related tasks, particularly in the context of few-shot learning. Moreover,

*Equal contribution, alphabetically ordered.

†Co-corresponding, alphabetically ordered.

Foundation Models for Science Workshop,38th Conference on Neural Information Processing Systems (NeurIPS 2024).

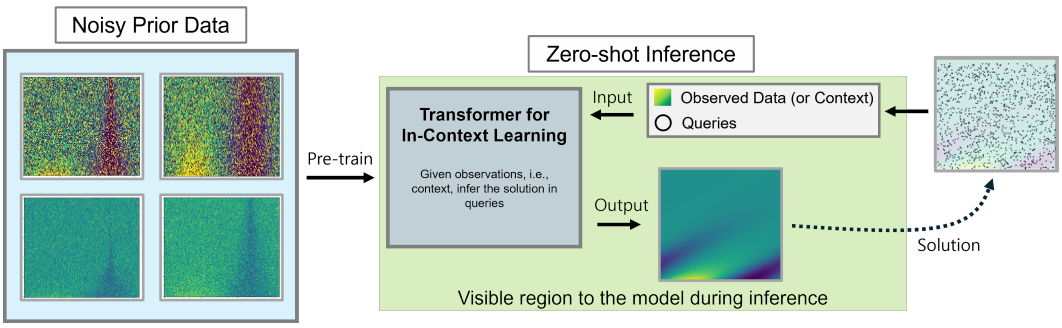


Figure 1: **End-to-end schematic diagram of our model.** Our model performs in-context learning based on the given observations, i.e., context, to infer the solution. Even when trained with noisy prior data, our model can obtain clean solutions due to its Bayesian inference capability.

the integration of in-context operator learning into multi-modal frameworks, as demonstrated by ICON-LM [46], has pushed the boundaries of traditional models by combining natural language with mathematical equations. Additionally, several other studies have focused on solving a family of PDEs with a single trained model [7]. However, all these studies are limited in their ability to fully harness the capabilities of large foundation models. Our methodology addresses these limitations and offers significant advantages in the following four aspects.

No prior knowledge of physical laws Our goal is to predict solutions from observed quantities, such as velocity and pressure, without relying on governing equations, a common challenge in many real-world scenarios [23, 28, 34, 3, 6]. In complex systems, such as those governing semiconductor manufacturing, the exact governing equations are often unknown and may change over time [6, 30]. Therefore, excluding these equations from the model input is a strategic choice aimed at enhancing the applicability of our method across various domains.

Zero-shot inference Our goal is to achieve zero-shot inference for predicting PDE solutions. For instance, ICON-LM requires few-shot "demos[3]" for an unknown target operator before making predictions. In contrast, our foundation model eliminates the need for such demos, as collecting them implies that inference cannot occur until these few-shot examples are available; see e.g., Figure 1. Our approach is designed to enable immediate inference as soon as the model is queried.

Bayesian inference We incorporate Bayesian inference into the prediction process by leveraging prior knowledge obtained from numerical solutions in PDE dictionaries. This approach allows the model to make more accurate and well-informed predictions by defining a prior distribution over unseen PDE coefficients. During training, the model learns to capture relationships among known data points using self-attention mechanisms, while cross-attention enables it to extrapolate and infer solutions for new, unseen points. When tested, the model utilizes this prior knowledge to generalize effectively to novel data points, achieving zero-shot predictions without the need for additional fine-tuning.

Noisy prior data For LLMs, one of the most challenging steps is gathering prior data, typically involving the crawling and cleaning of sentences from the Internet. However, this process is far from perfect due to two key issues: (i) the Internet, as a data source, is inherently unreliable, and (ii) cleaning such vast amounts of data requires significant manual effort. As a result, LLMs are often trained on incomplete or imperfect prior data. Remarkably, this realistic yet critical issue has been largely overlooked in the current literature related to SFMs despite their similarities to LLMs. For instance, when generating data using numerical solvers for PDEs whose analytical solutions are not known, it is inevitable to encounter numerical errors which present as a form of measurement noise. In this work, we are the first to explore the potential of pre-training SFMs using noisy data, since collecting high-fidelity solutions are frequently challenging for PDEs.

[3] In ICON and ICON-LM, a demo means a set of (input, output) pairs of an operator to infer.

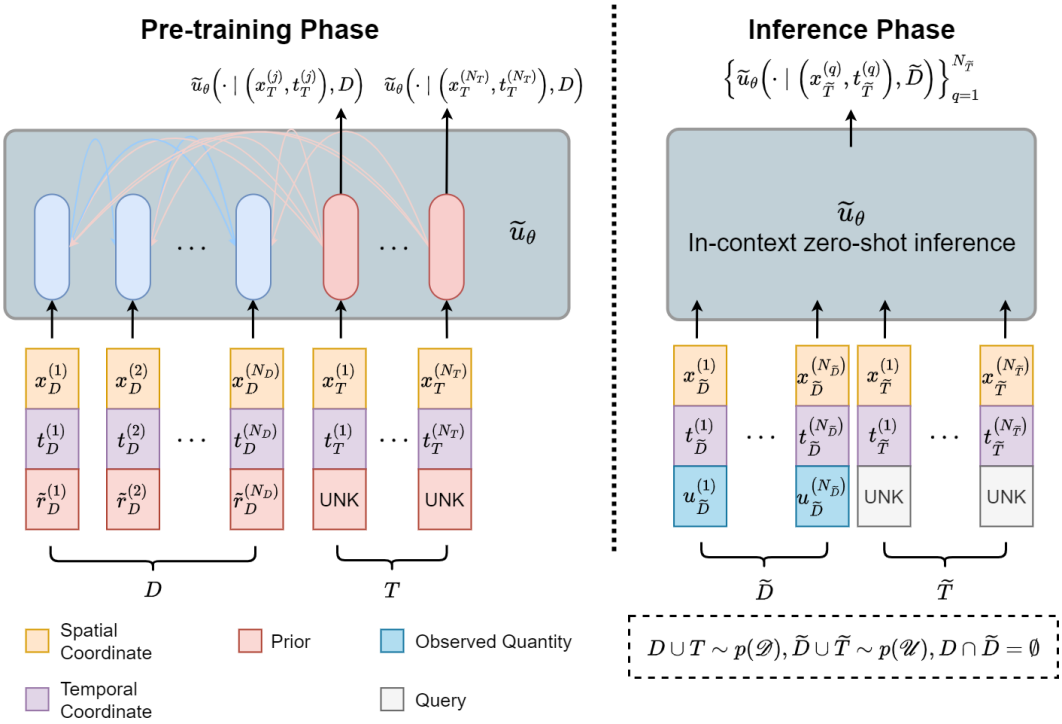


Figure 2: **Schematic diagram of Transformer.** *(Left)* The Transformer $\tilde{u}_\theta$ takes prior of solution-known D and querying task T drawn from the prior distribution $\mathcal{D}$ and infers solutions of the queried points in the training phase. ICL is leveraged with a self-attention among D (blue rods) and a cross-attention from T to D (red rods). *(Right)* In the testing phase, $\tilde{u}_\theta$ takes an input of unseen data $\widetilde{D}$ and $\widetilde{T}$ drawn from the ground truth distribution $\mathcal{U}$, and the model predicts the queried points $\widetilde{T}$.

1D CDR Study For our empirical studies, we utilize the convection-diffusion-reaction (CDR) equation and compare our approach with two state-of-the-art methods for solving parameterized PDEs. Additionally, we introduce three different types of noise into the numerical solutions of the CDR equation. Our method not only outperforms the two baseline methods but also demonstrates stable performance, even when noise is added to the prior data during pre-training.

2 Problem Setup

Benchmark PDE The one-dimensional convection-diffusion-reaction (CDR) equation (1) with a Fisher reaction term is used for the benchmark PDE,

$$\text{1D CDR: } u_t + \beta u_x - \nu u_{xx} - \rho u(1-u) = 0. \quad x \in [0, 2\pi], t \in [0, 1]. \tag{1}$$

This equation consists of three key terms with distinct properties, i.e., convective, diffusive and reactive, making it an ideal benchmark paradigm. It is commonly employed in the PINN literature due to the diverse dynamics introduced by three parameters: β, ν, and ρ, including various failure modes [22]. In this paper, we will use the following representation of the CDR equation (2):

$$u_t = \mathcal{N}(\cdot), \quad \mathcal{N}(t, x, u, u_x, u_{xx}, \alpha) = \beta u_x - \nu u_{xx} - \rho u(1-u), \tag{2}$$

where $\alpha := (\beta, \nu, \rho)$ according to [35].

Prior of PDE Solution Space We can then construct a parameter space, Ω, which is the collection of α values [33]. Consequently, the target exact prior $\mathcal{U}$ represents the collection of solutions $u(\alpha)$ for each parameter $\alpha \in \Omega$, where $\mathcal{X}$ and $\mathcal{T}$ correspond to the spatial and temporal domains of interest, respectively

$$\mathcal{U} = \bigcup_{\alpha \in \Omega} \{u(\alpha) \,|\, u_t = N(t, x, u, u_x, u_{xx}, \alpha)\}, \quad \mathcal{U} : \mathcal{X} \times \mathcal{T} \to \mathbb{R}. \tag{3}$$

Since the target exact prior data $\mathcal{U}$ is hard to obtain, we instead use a prior $\mathcal{D}$ that closely approximates $\mathcal{U}$ as follows. The prior $\mathcal{D}$ consists of the approximated solutions $\tilde{r}(\alpha)$ for each $\alpha \in \Omega$,

$$\mathcal{D} = \bigcup_{\alpha \in \Omega} \{\tilde{r}(\alpha)\}, \quad p(\mathcal{D}) \sim p(\mathcal{U}). \tag{4}$$

Subsequently, the model learns the posterior predictive distribution (PPD) of the generated prior $p(\mathcal{D})$ through ICL.

ICL of Transformer We use the capability of Transformers to perform Bayesian inference of a PPD [25] through ICL on prior data. At inference time, the non-overlapping N_D data points $D := \{(x_D^{(i)}, t_D^{(i)}, \tilde{r}(x_D^{(i)}, t_D^{(i)}))\}_{i=1}^{N_D}$ and N_T data points $\{(x_T^{(j)}, t_T^{(j)})\}_{j=1}^{N_T}$ are drawn i.i.d. from $\mathcal{D}$ and given as an input to the Transformer. The task of the Transformer is to predict $\{\tilde{r}(x_T^{(j)}, t_T^{(j)})\}_{j=1}^{N_T}$ based on the dataset $D \cup \{(x_T^{(j)}, t_T^{(j)})\}_{j=1}^{N_T}$. The context of the input data is interpreted by an encoder attention mask with two attentions: a self-attention among D and a cross-attention from T to D (Figure 2). Thus, the model can capture specific contextual information from the data points with known solutions, enabling it to make an inference based on this learned context.

Zero-shot learning The attention-based ICL of the Transformer enables zero-shot learning, allowing the model to make predictions when presented with unseen data. This is achieved through the model's Bayesian inference of the prior, which helps identify the relevant context from the new, unseen data.

Training From a given parameter space Ω, the parameter α is randomly drawn i.i.d. from Ω. This method is adopted from meta learning [12] which optimizes the model parameter to adapt to various tasks, in our case the prediction over wide prior space $\mathcal{D}$ expressed as a dictionary of α. After that, the prior $\tilde{r}(\alpha)$ is then given as an input of Transformer $\tilde{u}_\theta$ to minimize the mean squared error (MSE) on predicted points (5). The MSE loss criterion is proposed as the Transformer's task is to perform regression of solution over spatial and temporal domain for given $\tilde{r}(\alpha)$,

$$L_\alpha = \frac{1}{N_T} \sum_{j=1}^{N_T} \left[\tilde{u}(x_T^{(j)}, t_T^{(j)}) - \tilde{r}(x_T^{(j)}, t_T^{(j)})\right]^2. \tag{5}$$

Evaluation After training, we assess the model's performance using data sampled i.i.d. from $\mathcal{U}$, ensuring no overlap with the training set $D \cup T$, to illustrate the model's zero-shot learning capability in scenarios commonly encountered in practical applications. For evaluation, we employ both L_1 mean absolute error and L_2 relative errors between the model's predicted solutions for test queries and the numerically computed ground truth. These errors are then averaged over the target parameter space Ω used during training.

3 Scientific Foundation Model via Bayesian Methods

When modeling complex systems, it is often impractical to assume that data originates from predefined parametric equations. Real-world data exhibits complexity and variability that is difficult to capture with strict assumptions. This has led to the development of more flexible Bayesian methods, which do not rely on specific parametric forms but instead allow for adaptable modeling to better reflect the underlying data distribution.

Consider a sequence of pairs $(X_1, Y_1), (X_2, Y_2), \ldots$, each within the measurable space $(\mathcal{X} \times \mathcal{Y}, \mathcal{B})$, where X_i represents the spatiotemporal coordinate, Y_i denotes the corresponding solution in this paper's context and $\mathcal{B}$ denotes the Borel σ-algebra on the measurable space $\mathcal{X} \times \mathcal{Y}$. For simplicity, we adopt this notation in this section. These pairs are drawn from an unknown true density function π. Lacking information about π, we adopt a Bayesian framework to establish a prior distribution Π over the space H of density functions on $(\mathcal{X} \times \mathcal{Y}, \mathcal{B})$. This prior is updated with the observed data to form the posterior distribution Π^n, which is defined as

$$\Pi^n(A) = \frac{\int_A L_n(q)\Pi(\mathrm{d}q)}{\int_H L_n(q)\Pi(\mathrm{d}q)}, \tag{6}$$

where $L_n(q) = \prod_{i=1}^{n} \frac{q(X_i,Y_i)}{\pi(X_i,Y_i)}$ for $A \subset H$. The resulting posterior density is

$$q_n(X,Y) = \int_H q(X,Y)\Pi^n(\mathrm{d}q). \tag{7}$$

Adopting the notation $\Pi^n(\mathrm{d}q) = \mathrm{d}\Pi(q \mid D_n)$, the posterior predictive distribution (PPD) is formulated as

$$\pi(y \mid x, \mathcal{D}_n) = \int_H q(y \mid x)\, \mathrm{d}\Pi(q \mid x, \mathcal{D}_n). \tag{8}$$

The behavior of D_n plays a crucial role in this formulation. As noted by [40, 39, 4, 38, 27], for a well-behaved prior, PPD converges toward π as n increases. This aligns with findings in [4], demonstrating that in well-specified scenarios, strong consistency is achieved as

$$\Pi^n\{q : H(\pi, q) > \epsilon\} \to 0 \quad \text{almost surely}, \tag{9}$$

for any $\epsilon > 0$. This indicates that the posterior distribution becomes concentrated in a small Hellinger neighborhood around the true density function π.

Theorem 3.1. *Suppose that for any $n \in \mathbb{N}$ and $\epsilon > 0$, there exists a Transformer parameterized by $\hat{\theta}$ such that*

$$\hat{\theta} = \arg\min_{\theta} \mathbb{E}_x \left[KL\left(p_\theta(\cdot \mid x, D_n), \pi(\cdot \mid x, D_n)\right)\right] < \epsilon.$$

If the posterior consistency condition holds, and for any $q \in \mathcal{Q}$, $q(x) = \pi(x)$ almost everywhere on $\mathcal{X}$, then the following holds almost surely then the following holds

$$\mathbb{E}_x \left[H\left(p_{\hat{\theta}}(\cdot \mid x, D_n), \pi(\cdot \mid x)\right)\right] \xrightarrow{n\to\infty} 0 \quad \textit{almost surely}.$$

Proof. The proof of Theorem 3.1 is provided in Appendix A. □

This result demonstrates that as the amount of data increases, the neural network close to the posterior distribution converges to the expected value under the prior distribution, highlighting the consistency and robustness of the Bayesian inference process. Building upon the theoretical foundation established in the previous theorem, our model performs the Bayesian inference with prior data. Ultimately, our model's goal is to read some ground-truth spatiotemporal points and infer an appropriate PDE solution that accurately describes the dynamics under the given spatiotemporal conditions.

Suppose dataset $D_n = \{(X_i, Y_i)\}_{i=1}^n$ is independently and identically distributed (i.i.d.) and sampled from some distribution $q_\alpha \sim \pi$. Specifically, $Y_i \sim u(X_i \mid \alpha) + \text{noise}$, where noise is a small Gaussian noise. The PPD is then given by

$$q(y \mid x, \mathcal{D}_n) = \int_H q_\alpha(y \mid x)\, d\Pi(q_\alpha \mid x, \mathcal{D}_n), \tag{10}$$

where we consider $q(y \mid X, D_n)$ as the solution likelihood distribution given D_n, representing the distribution most likely to select the correct solution. Inspired by [26, 1, 27], we approximate the PPD by minimizing the Kullback—Leibler (KL) divergence between the true distribution $q(\cdot \mid x, D_n)$ and the approximating model $p_\theta(\cdot \mid x, D_n)$. To achieve this, we adjust our loss function as follows:

$$l_{\theta,n} := -\mathbb{E}_\alpha \mathbb{E}_{D_n \sim q_\alpha} \mathbb{E}_{y \sim u_\alpha(x) + \text{noise}} \left[\log p_\theta(y \mid x, D_n)\right].$$

4 Experiments

In this section, we study the model with the following four prior distributions $p(\mathcal{D})$:

P1 (noiseless) : $p(\mathcal{D}) = p(\mathcal{U})$, **P2** (Gaussian noise) : $p(\mathcal{D}) \sim \mathcal{N}(\mathcal{U}, \sigma^2\mathbf{I})$,

P3 (salt-and-pepper noise) : $p(\mathcal{D}) \sim p(s \cdot \mathcal{U})$ where $s = \begin{cases} \min(\mathcal{U}) & \text{with probability } \frac{\gamma}{2}, \\ \max(\mathcal{U}) & \text{with probability } \frac{\gamma}{2}, \\ 1 & \text{with probability } 1-\gamma, \end{cases}$

P4 (uniform noise) : $p(\mathcal{D}) \sim p(\mathcal{U} + U(-\epsilon, \epsilon))$ (U: uniform distribution).

Table 1: Major comparisons between Hyper-LR-PINN, P^{2}INN, and our model. While both Hyper-LR-PINN and P^{2}INN require the knowledge of governing equation, our model only needs observed quantities. The notations used in the table are fully aligned with those in Figure 2.

Properties	**Hyper-LR-PINN**	**P^{2}INN**	**Ours**
Target function	$u(x,t;\alpha\in\Omega)$	$u(x,t;\alpha\in\Omega)$	$u(x,t)\|_{\mathcal{D}}$
Governing equation $\mathcal{N}(\cdot)$ given	✓	✓	✗
Train dataset	$D\cup T\cup\widetilde{D}$	$D\cup T\cup\widetilde{D}$	$D\cup T$
Test dataset	$\widetilde{T}$	$\widetilde{T}$	$\widetilde{D}\cup\widetilde{T}$
Dataset with a solution	None	None	$D,T,\widetilde{D}$

Algorithm 1 Training a Transformer

1: **Input:** A prior dataset $D\cup T$ drawn from prior $p(\mathcal{D})$
2: **Output:** A Transformer $\tilde{u}_\theta$ which can approximate the PPD
3: Initialize the Transformer $\tilde{u}_\theta$
4: **for** $i=1$ to n **do**
5: Sample $\alpha\in\Omega$ and $D\cup T\subseteq\tilde{r}(\alpha)\sim p(\mathcal{D})$
6: $(D:=\{(x_D^{(i)},t_D^{(i)})\}_{i=1}^{N_D}, T:=\{(x_T^{(j)},t_T^{(j)})\}_{j=1}^{N_T})$
7: Compute loss $L_\alpha=\frac{1}{N_T}\sum_{j=1}^{N_T}\left\{\tilde{u}(x_T^{(j)},t_T^{(j)})-\tilde{r}(x_T^{(j)},t_T^{(j)})\right\}^2$.
8: Update parameters θ with an Adam optimizer
9: **end for**

The study of models in the noiseless case **P1** verifies the ICL capabilities of the Transformer with various tasks such as predicting seen/unseen PDE solutions (5.2 $\sim$ 5.3) and extrapolating solutions in the temporal domain (5.4). After that, we study three different noisy prior distributions **P2**, **P3** and **P4** (5.5) to reinforce the ICL property of our model by guaranteeing its capability for zero-shot learning.

4.1 Experimental Setup

Baseline methods We compare our model with 2 baselines: Hyper-LR-PINN [8] and P^{2}INN without fine tuning [7]. Both models are parametrized physics-informed neural networks (PINNs) designed to learn parameterized PDEs. Hyper-LR-PINN emphasizes a low-rank architecture with a parameter hypernetwork, while P^{2}INN focuses on a parameter-encoding scheme based on the latent space of the parameterized PDEs.

Following this, as shown in Figure 2, the model takes $D\cup T\sim p(\mathcal{D})$ in training phase and $\widetilde{D}\cup\widetilde{T}\sim p(\mathcal{U})$ in testing phase. In addition, the dataset $D\cup T$ requires the prior $\widetilde{r}$, and $\widetilde{D}$ requires the solution u. For a fair comparison, we use D, T, and $\widetilde{D}$ as the training dataset for both Hyper-LR-PINN and P^{2}INN. Notably, while Hyper-LR-PINN and P^{2}INN do not rely on solution points during training and testing, our model operates without any knowledge of the governing equation $\mathcal{N}(\cdot)$. This setup ensures a valid and balanced comparison. (Table 1).

Training algorithm The concrete flow of training phase is described in Algorithm 1.

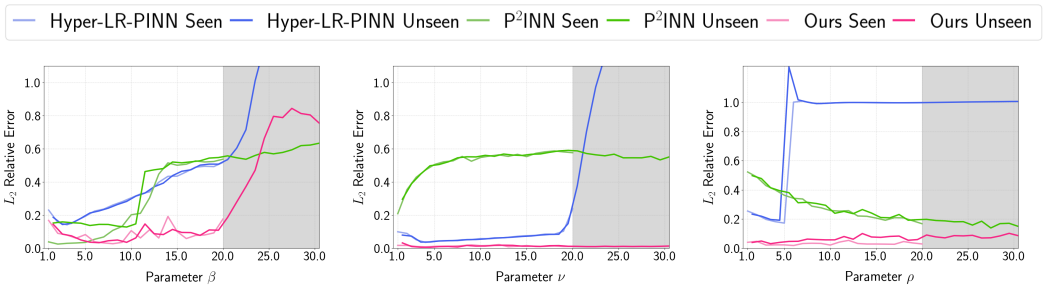
4.2 Time Domain Interpolation for Seen PDE Parameters

In this section, we employ 6 different dynamics derived from 1D CDR equation. For each dynamic, we set the parameter space Ω with three different coefficient (β,ν,ρ) range: $([1,5]\cap\mathbb{Z})^m$, $([1,10]\cap\mathbb{Z})^m$, and $([1,20]\cap\mathbb{Z})^m$ where m is the number of nonzero coefficients. The Transformer $\widetilde{u}_\theta$ is trained with $D\cup T\subseteq r(\alpha)$ where $\alpha\in\Omega$ is selected at least once and uniformly at random manner for each epoch. After that, we test $\widetilde{u}_\theta$ with $\widetilde{D}\cup\widetilde{T}\subseteq u(\alpha)$ for all $\alpha\in\Omega$ and evaluate average L_1 mean absolute and L_2 relative error (Table 2).

We can point out two notable facts: The model outperforms on diffusion, reaction, reaction-diffusion, and convection-diffusion-reaction system, and it shows stable performance through the value of coefficients. For instance, all baselines show difficulties in predicting accurate solutions for high

Table 2: The L_1 mean absolute and L_2 relative errors over the 1D-CDR equation using **P1** prior. P^{2}INN is tested without fine-tuning, and *-marked cases are evaluated with a reduced number of parameters due to the extensive computational requirements.

System	Coefficient range	Hyper-LR-PINN		P^{2}INN		Ours	
		Abs.err	Rel.err	Abs.err	Rel.err	Abs.err	Rel.err
Convection	$\beta \in [1,5] \cap \mathbb{Z}$	**0.0104**	**0.0119**	0.0741	0.1020	0.0192	0.0184
	$\beta \in [1,10] \cap \mathbb{Z}$	**0.0172**	**0.0189**	0.1636	0.1801	0.0250	0.0251
	$\beta \in [1,20] \cap \mathbb{Z}$	**0.0340**	**0.0368**	0.2742	0.2743	0.0764	0.0864
Diffusion	$\nu \in [1,5] \cap \mathbb{Z}$	0.0429	0.0570	0.3201	0.3652	**0.0096**	**0.0120**
	$\nu \in [1,10] \cap \mathbb{Z}$	0.0220	0.0282	0.3550	0.4029	**0.0108**	**0.0137**
	$\nu \in [1,20] \cap \mathbb{Z}$	0.1722	0.1991	0.4553	0.5166	**0.0095**	**0.0134**
Reaction	$\rho \in [1,5] \cap \mathbb{Z}$	0.0124	0.0428	0.0109	0.0354	**0.0102**	**0.0154**
	$\rho \in [1,10] \cap \mathbb{Z}$	0.2955	0.3562	0.0192	0.0708	**0.0129**	**0.0202**
	$\rho \in [1,20] \cap \mathbb{Z}$	0.7111	0.7650	0.1490	0.2915	**0.0160**	**0.0322**
Convection-Diffusion	$\beta, \nu \in [1,5] \cap \mathbb{Z}$	**0.0046**	**0.0055**	0.1329	0.1554	0.0195	0.0231
	$\beta, \nu \in [1,10] \cap \mathbb{Z}$	0.0268	0.0295	0.1609	0.1815	**0.0211**	**0.0274**
	$\beta, \nu \in [1,20] \cap \mathbb{Z}$	*0.1487	*0.1629	0.1892	0.2044	**0.0226**	**0.0305**
Reaction-Diffusion	$\nu, \rho \in [1,5] \cap \mathbb{Z}$	0.0817	0.1160	0.0579	0.1346	**0.0139**	**0.0189**
	$\nu, \rho \in [1,10] \cap \mathbb{Z}$	0.0317	0.0446	0.4398	0.5457	**0.0122**	**0.0189**
	$\nu, \rho \in [1,20] \cap \mathbb{Z}$	*0.3228	*0.3844	0.1513	0.2955	**0.0165**	**0.0331**
Convection-Diffusion-Reaction	$\beta, \nu, \rho \in [1,5] \cap \mathbb{Z}$	0.0231	0.0307	0.0418	0.0595	**0.0143**	**0.0209**
	$\beta, \nu, \rho \in [1,10] \cap \mathbb{Z}$	*0.3135	*0.3732	0.0367	0.0624	**0.0276**	**0.0411**
	$\beta, \nu, \rho \in [1,20] \cap \mathbb{Z}$	*0.9775	*0.9958	0.0446	0.1211	**0.0159**	**0.0310**
Statistics	Average	0.1805	0.2033	0.1709	0.2222	**0.0196**	**0.0267**
	Standard Deviation	0.2581	0.2727	0.1423	0.1549	**0.0147**	**0.0164**



(a) convection (b) diffusion (c) reaction

Figure 3: L_2 relative error measured at unseen parameters for (a) convection, (b) diffusion, and (c) reaction. The result of seen parameters are plotted together. The grey area indicates the region where the model extrapolates the coefficient β, ν, or ρ.

coefficient especially in diffusion and reaction system while ours do not. When we measure the standard deviation of L_2 relative error over three coefficient range for diffusion system, ours have 9.1×10^{-4} while others show 10^{-2} scale value. These observations not only verify the effectiveness of the Transformer's ICL capability, but also suggest its potential to handle larger parameter space Ω.

4.3 Time Domain Interpolation for Unseen PDE Parameters

We test our model with unseen parameters at convection, diffusion, and reaction systems. For each system, the model is trained with $[1,20] \cap \mathbb{Z}$ range coefficients and tested with unseen coefficient $1.5, 2.5, \cdots, 19.5$ which is included in interval $[1,20]$ and $20.5, 21.5, 22.5, \cdots, 30.5$ which is not in range of $[1,20]$. The measured L_2 relative error for each coefficient value is plotted in Figure 3 with baselines Hyper-LR-PINN and P^{2}INN. Both baselines are not fine-tuned for each parameter to compare with our zero-shot learned model.

Over the trained coefficient range, our model effectively interpolates the coefficients β, ν, and ρ, achieving performance comparable to that seen with known coefficients. Moreover, the model demonstrates stable extrapolation in diffusion and reaction systems. Compared to the baselines, our model significantly outperforms it, particularly in diffusion and reaction systems. This result indicates that the Transformer can effectively learn the posterior predictive distribution (PPD) of the prior space $\mathcal{D}$, even without observing the complete prior.

4.4 Time Domain Extrapolation for Seen PDE Parameters

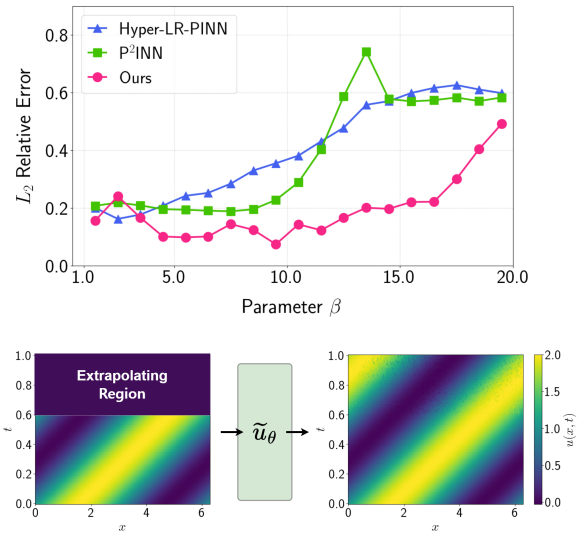


Figure 4: (*upper*) The L_2 relative error is evaluated for each convection coefficient $\beta = 1.5, 2.5, \cdots, 19.5$ as an extrapolation task. (*lower*) The graph illustrates the extrapolation of convection equation with $\beta = 5.5$ at $0.6 \leq t \leq 1.0$.

One major limitation of the PINN is an extrapolation at the temporal domain that infer solutions at unknown points. Our model demonstrates extrapolation capability in the 1D convection equation, where the solution exhibits wave-like fluctuations in the inference region. In particular, the model trained with the **P1** prior over the coefficient range $\beta \in [1, 20] \cap \mathbb{Z}$ can predict β values in $1.5, 2.5, \cdots, 19.5$ for equations where the test points $\widetilde{T}$ fall within $t \in (0.6, 1.0]$, even though $\widetilde{D}$ is only distributed within $t \in [0.0, 0.6]$. We then evaluate the relative L_2 error and plot for each coefficient β with our baselines. Both baselines are not fine-tuned for each test β to make a fair comparison with our zero-shot model. (Figure 4, *upper*).

In practice, extrapolation is performed in a section-by-section manner. The extrapolation interval $(0.6, 1.0]$ is divided into 10 consecutive sections: $(0.6, 0.64]$, $(0.64, 0.68]$, $\cdots$, $(0.96, 1.0]$. For each section, the model output from the previous sections is added to $\widetilde{D}$ to infer the current section.

As a result, our model demonstrates effective extrapolation capabilities in convection equation (Figure 4). In addition, our model outperforms both Hyper-LR-PINN and P^{2}INN across most values of β, while maintaining a stable L_2 relative error over a wider range. The diagram at Figure 4, *lower* presents the detailed performance at $\beta = 5.5$. This capability emphasizes our model's potential for advancing solutions to PDEs in unknown spatial regions and for enhancing time series predictions.

4.5 In-Context Learning of Transformers with Noisy Prior

The use of **P1** prior shows the capability of ICL of Transformer. Building upon the result, the introduction of **P2**, **P3**, and **P4** priors, which inject noises, further highlights the capability of our model's zero-shot inference.

In this section, we sample $D \cup T \sim p(\mathcal{D})$, where $\mathcal{D}$ is a noisy prior, and train the Transformer $\widetilde{u}_\theta$. We then test $\widetilde{u}_\theta$ with $\widetilde{D} \cup \widetilde{T} \sim p(\mathcal{U})$, demonstrating that the model can predict the true solution even when trained on noisy prior data. The experiment is conducted on reaction and convection-diffusion-reaction equations, which outperform other baselines, under three different noises: the Gaussian noise (**P2**), the salt-and-pepper noise (**P3**), and the uniform noise (**P4**). The standard deviation σ of Gaussian noise is set to 1%, 5%, and 10% of the mean value of the ground truth solution. Additionally, for the experiment, the probe γ for salt-and-pepper noise and the range ϵ for uniform noise are also set to 1%, 5%, and 10%.

Our model demonstrates robust performance across different types of noise injection as shown in Table 3. For Gaussian noise, neither the L_1 mean absolute nor L_2 relative errors are significantly

Table 3: The L_1 mean absolute error and L_2 relative errors for the reaction and convection-diffusion-reaction systems using the **P2** prior with varying levels of Gaussian noise σ, **P3** prior with varying levels of noise probe γ, and **P4** prior with varying levels of noise ϵ (1%, 5%, and 10%). For a comparison, the result of using **P1** prior is notated.

System	Prior Type	Noisy Prior with a Noise Level						P1 Prior	
		1% Noise		5% Noise		10% Noise			
		Abs. err	Rel. err	Abs. err	Rel. err	Abs. err	Rel. err	Abs. err	Rel. err
Reaction	**P2**	0.0210	0.0392	0.0213	0.0399	0.0210	0.0392	0.0160	0.0322
	P3	0.0309	0.0598	0.0286	0.0517	0.0354	0.0619		
	P4	0.0285	0.0568	0.0293	0.0583	0.0306	0.0607		
Convection -Diffusion -Reaction	**P2**	0.0175	0.0296	0.0220	0.0431	0.0235	0.0431	0.0159	0.0310
	P3	0.0246	0.0459	0.0263	0.0453	0.0267	0.0496		
	P4	0.0210	0.0420	0.0215	0.0422	0.0230	0.0426		

influenced by the noise level. Notably, the model with 10% prior noise consistently outperforms other baselines, as shown in Table 2. Similarly, although salt-and-pepper noise is the most challenging and our model encounters greater difficulty compared to **P2** and **P4** priors, it still achieves superior performance even with 10% prior noise. For uniform noise, while the L_1 mean absolute and L_2 relative errors increase progressively with the noise level ϵ, our model with 10% prior noise remains consistently better than the baselines. It shows our Transformer can perform ICL with zero-shot learning even if it is trained with inaccurate or noisy prior $D \cup T \sim p(\mathcal{D})$.

5 Related Works

In-context learning Transformers have shown remarkable ICL abilities across various studies. They can generalize to unseen tasks by emulating Bayesian predictors [29] and linear models [49], while also efficiently performing Bayesian inference through Prior-Data Fitted Networks (PFNs) [26]. Their robustness extends to learning different function classes, such as linear and sparse linear functions, decision trees, and two-layer neural networks even under distribution shifts [14]. Furthermore, Transformers can adaptively select algorithms based on input sequences, achieving near-optimal performance on tasks like noisy linear models [2]. They are also highly effective ans fast for tabular data classification [18].

Foundation model Recent studies have advanced in-context operator learning and PDE solving through Transformer-based models. [48] introduces PDEformer, a versatile model for solving 1D PDEs with high accuracy and strong performance in inverse problems. In-context operator learning has also been extended to multi-modal frameworks, as seen in [46], where ICON-LM integrates natural language and equations to outperform traditional models. Additionally, [47] and [45] demonstrate the generalization capabilities of In-Context Operator Networks (ICON) in solving various PDE-related tasks, highlighting ICON's adaptability and potential for few-shot learning across different differential equation problems. Several other studies have addressed the problem of solving various PDEs using a single trained model [16, 17] . However, many of these approaches rely on symbolic PDE information, true or near-true solutions and/or do not support zero-shot in-context learning, making their objectives different from ours.

6 Conclusions

In this work, we presented a foundation model for scientific machine learning that integrates in-context learning and Bayesian inference for predicting PDE solutions. Our results demonstrate that Transformers, equipped with self-attention and cross-attention mechanisms, can effectively generalize from prior data, even in the presence of noise, and exhibit robust zero-shot learning capabilities. These findings suggest that foundation models in SML have the potential to follow the development

trajectory similar to that of natural language processing foundation models, offering new avenues for further exploration and advancement in the field.

Limitations and future work In this workshop version of our on-going research, only 1D CDR problems have been used for experiments. They have terms with different characteristics and have been used widely in SML. In the future, however, we will extend our SFM for learning various PDEs at the same time. Therefore, the prior data collection step should be extended as well.

Acknowledgments and Disclosure of Funding

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. This article has been co-authored by an employee of National Technology & Engineering Solutions of Sandia, LLC under Contract No. DE-NA0003525 with the U.S. Department of Energy (DOE). The employee owns all right, title and interest in and to the article and is solely responsible for its contents. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this article or allow others to do so, for United States Government purposes. The DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan *https://www.energy.gov/downloads/doe-public-access-plan*. K. Lee acknowledges support from the U.S. National Science Foundation under grant IIS 2338909. The work of A.G. is partially supported by the U.S. Department of Energy, Office of Advanced Scientific Computing Research under the "Scalable, Efficient, and Accelerated Causal Reasoning Operators, Graphs, and Spikes for Earth and Embedded Systems (SEA-CROGS)" project. The work of Y.H. was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2021R1A2C1093579) and by the Korea government (MSIT) (RS-2023-00219980). N. Park was partly supported by Samsung Electronics Co., Ltd. (No. G01240136, KAIST Semiconductor Research Fund (2nd)).

References

[1] Steven Adriaensen, Herilalaina Rakotoarison, Samuel Müller, and Frank Hutter. Efficient bayesian learning curve extrapolation using prior-data fitted networks. *Advances in Neural Information Processing Systems*, 36, 2024.

[2] Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 57125–57211. Curran Associates, Inc., 2023.

[3] Andrea Beck and Marius Kurz. A perspective on machine learning methods in turbulence modeling. *GAMM-Mitteilungen*, 44(1):e202100002, 2021.

[4] Pierpaolo De Blasi and Stephen G. Walker. Bayesian asymptotics with misspecified models. *Statistica Sinica*, 23(1):169–187, 2013.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[6] Chen-Fu Chien, Chia-Yu Hsu, and Chih-Wei Hsiao. Manufacturing intelligence to forecast and reduce semiconductor cycle time. *Journal of Intelligent Manufacturing*, 23:2281–2294, 2012.

[7] Woojin Cho, Minju Jo, Haksoo Lim, Kookjin Lee, Dongeun Lee, Sanghyun Hong, and Noseong Park. Parameterized physics-informed neural networks for parameterized PDEs. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 8510–8533. PMLR, 21–27 Jul 2024.

[8] Woojin Cho, Kookjin Lee, Donsub Rim, and Noseong Park. Hypernetwork-based meta-learning for low-rank physics-informed neural networks. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 11219–11231. Curran Associates, Inc., 2023.

[9] Junho Choi, Taehyun Yun, Namjung Kim, and Youngjoon Hong. Spectral operator learning for parametric pdes without data reliance. *Computer Methods in Applied Mechanics and Engineering*, 420:116678, 2024.

[10] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

[11] Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can GPT learn in-context? language models implicitly perform gradient descent as meta-optimizers. In *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2023.

[12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR, 2017.

[13] Simon Frieder, Luca Pinchetti, , Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Petersen, and Julius Berner. Mathematical capabilities of chatgpt. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 27699–27744. Curran Associates, Inc., 2023.

[14] Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 30583–30598. Curran Associates, Inc., 2022.

[15] Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. Large language models are zero-shot time series forecasters. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 19622–19635. Curran Associates, Inc., 2023.

[16] 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.

[17] Maximilian Herde, Bogdan Raonić, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for pdes. *arXiv preprint arXiv:2405.19101*, 2024.

[18] Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *NeurIPS 2022 First Table Representation Workshop*, 2022.

[19] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[20] Namjung Kim, Dongseok Lee, and Youngjoon Hong. Data-efficient deep generative model with discrete latent representation for high-fidelity digital materials. *ACS Materials Letters*, 5(3):730–737, 2023.

[21] Namjung Kim, Dongseok Lee, Chanyoung Kim, Dosung Lee, and Youngjoon Hong. Simple arithmetic operation in latent space can generate a novel three-dimensional graph metamaterials. *npj Computational Materials*, 10(1), October 2024.

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

[23] Chin Yik Lee and Stewart Cant. A grid-induced and physics-informed machine learning cfd framework for turbulent flows. *Flow, Turbulence and Combustion*, 112(2):407–442, 2024.

[24] Michael Moor, Oishi Banerjee, Zahra Shakeri Hossein Abad, Harlan M. Krumholz, Jure Leskovec, Eric J. Topol, and Pranav Rajpurkar. Foundation models for generalist medical artificial intelligence. *Nature*, 616(7956):259–265, April 2023.

[25] Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations*, 2022.

[26] Samuel Müller, Noah Hollmann, Sebastian Pineda-Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. *CoRR*, abs/2112.10510, 2021.

[27] Thomas Nagler. Statistical foundations of prior-data fitted networks. In *International Conference on Machine Learning*, pages 25660–25676. PMLR, 2023.

[28] Zachary G Nicolaou, Guanyu Huo, Yihui Chen, Steven L Brunton, and J Nathan Kutz. Data-driven discovery and extrapolation of parameterized pattern-forming dynamics. *Physical Review Research*, 5(4):L042017, 2023.

[29] Madhur Panwar, Kabir Ahuja, and Navin Goyal. In-context learning through the bayesian prism. In *The Twelfth International Conference on Learning Representations*, 2024.

[30] Michael Quirk and Julian Serda. *Semiconductor manufacturing technology*, volume 1. Prentice Hall Upper Saddle River, NJ, 2001.

[31] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[32] 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, 2019.

[33] Maziar Raissi. Deep hidden physics models: Deep learning of nonlinear partial differential equations. *Journal of Machine Learning Research*, 19(25):1–24, 2018.

[34] Nusrat Rouf, Majid Bashir Malik, Tasleem Arif, Sparsh Sharma, Saurabh Singh, Satyabrata Aich, and Hee-Cheol Kim. Stock market prediction using machine learning techniques: a decade survey on methodologies, recent developments, and future directions. *Electronics*, 10(21):2717, 2021.

[35] Samuel H. Rudy, Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, 2017.

[36] Shashank Subramanian, Peter Harrington, Kurt Keutzer, Wahid Bhimji, Dmitriy Morozov, Michael W. Mahoney, and Amir Gholami. Towards foundation models for scientific machine learning: Characterizing scaling and transfer behavior. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

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

[38] Stephen Walker. On sufficient conditions for bayesian consistency. *Biometrika*, 90(2):482–488, 2003.

[39] Stephen Walker. New approaches to bayesian consistency. *The Annals of Statistics*, 32(5), October 2004.

[40] Stephen G Walker. Modern bayesian asymptotics. *Statistical Science*, pages 111–117, 2004.

[41] Jared Willard, Xiaowei Jia, Shaoming Xu, Michael Steinbach, and Vipin Kumar. Integrating scientific knowledge with machine learning for engineering and environmental systems. *ACM Comput. Surv.*, 55(4), nov 2022.

[42] Qianqian Xie, Qingyu Chen, Aokun Chen, Cheng Peng, Yan Hu, Fongci Lin, Xueqing Peng, Jimin Huang, Jeffrey Zhang, Vipina Keloth, et al. Me llama: Foundation large language models for medical applications. *arXiv preprint arXiv:2402.12749*, 2024.

[43] Zhuoyan Xu, Zhenmei Shi, Junyi Wei, Fangzhou Mu, Yin Li, and Yingyu Liang. Towards few-shot adaptation of foundation models via multitask finetuning. In *The Twelfth International Conference on Learning Representations*, 2024.

[44] Dongchao Yang, Jinchuan Tian, Xu Tan, Rongjie Huang, Songxiang Liu, Xuankai Chang, Jiatong Shi, Sheng Zhao, Jiang Bian, Xixin Wu, et al. Uniaudio: An audio foundation model toward universal audio generation. *arXiv preprint arXiv:2310.00704*, 2023.

[45] Liu Yang, Siting Liu, Tingwei Meng, and Stanley J. Osher. In-context operator learning with data prompts for differential equation problems. *Proceedings of the National Academy of Sciences*, 120(39):e2310142120, 2023.

[46] Liu Yang, Siting Liu, and Stanley J Osher. Fine-tune language models as multi-modal differential equation solvers. *arXiv preprint arXiv:2308.05061*, 2023.

[47] Liu Yang and Stanley J. Osher. Pde generalization of in-context operator networks: A study on 1d scalar nonlinear conservation laws. *Journal of Computational Physics*, page 113379, 2024.

[48] Zhanhong Ye, Xiang Huang, Leheng Chen, Hongsheng Liu, Zidong Wang, and Bin Dong. Pdeformer: Towards a foundation model for one-dimensional partial differential equations. *arXiv preprint arXiv:2402.12652*, 2024.

[49] Ruiqi Zhang, Spencer Frei, and Peter L. Bartlett. Trained transformers learn linear models in-context. *Journal of Machine Learning Research*, 25(49):1–55, 2024.

A The Proof of Theorem 3.1

Proof. For any n, ϵ, we derive that

$$\begin{aligned}
\mathbb{E}_x\left[H\left(p_{\hat{\theta}}(\cdot \mid x, D_n), \pi(\cdot \mid x)\right)\right] &\leq \mathbb{E}_x\left[H\left(p_{\hat{\theta}}(\cdot \mid x, D_n), \pi(\cdot \mid x, D_n)\right)\right] && (1)\\
&\quad + \mathbb{E}_x\left[H\left(\pi(\cdot \mid x, D_n), \pi(\cdot \mid x)\right)\right] \\
&\leq \sqrt{\frac{1}{2}\mathbb{E}_x\left[KL\left(p_{\hat{\theta}}(\cdot \mid x, D_n), \pi(\cdot \mid x, D_n)\right)\right]} && (2)\\
&\quad + \mathbb{E}_x\left[H\left(\pi(\cdot \mid x, D_n), \pi(\cdot \mid x)\right)\right] \\
&\leq \sqrt{\frac{\epsilon}{2}} + \mathbb{E}_x\left[1 - \int_{\mathcal{Y}} \sqrt{\int q(y \mid x)\pi(y \mid x)\mathrm{d}\Pi^n(q)}\mathrm{d}y\right]^{1/2} && (3)\\
&\leq \sqrt{\frac{\epsilon}{2}} + \left[1 - \int_{\mathcal{X}} \pi(x) \int_{\mathcal{Y}} \frac{1}{\pi(x)} \sqrt{\int q(y, x)\pi(y, x)\mathrm{d}\Pi^n(q)}\mathrm{d}y\mathrm{d}x\right]^{1/2} \\
&\leq \sqrt{\frac{\epsilon}{2}} + \left[1 - \int_{\mathcal{X}} \int_{\mathcal{Y}} \int \sqrt{q(y, x)\pi(y, x)}\mathrm{d}\Pi^n(q)\mathrm{d}y\mathrm{d}x\right]^{1/2} \\
&= \sqrt{\frac{\epsilon}{2}} + \left[\int H\left(q, \pi\right)^2 \mathrm{d}\Pi^n(q)\right]^{1/2} \\
&\leq \sqrt{\frac{\epsilon}{2}} + \left[\int H\left(q, \pi\right) d\Pi^n(q)\right]^{1/2} \\
&= \sqrt{\frac{\epsilon}{2}} + \left[\int_{\{q: H(\pi, q) > \epsilon\}} H\left(q, \pi\right) \mathrm{d}\Pi^n(q)\right]^{1/2} \\
&\quad + \left[\int_{\{q: H(\pi, q) \leq \epsilon\}} H\left(q, \pi\right) \mathrm{d}\Pi^n(q)\right]^{1/2} && (4)\\
&= \sqrt{\frac{\epsilon}{2}} + (\Pi^n(\{q : H(\pi, q) > \epsilon\}) + \epsilon)^{1/2} \to \sqrt{\frac{\epsilon}{2}} + \sqrt{\epsilon} \quad \text{a.s.} \\
& && (5)
\end{aligned}$$

The first inequality (1) is derived from the triangle inequality for the Hellinger distance, which states that for any intermediate distribution $q(\cdot \mid x, D_n)$, we have

$$H\left(p_{\hat{\theta}}(\cdot \mid x, D_n), \pi(\cdot \mid x)\right) \leq H\left(p_{\hat{\theta}}(\cdot \mid x, D_n), q(\cdot \mid x, D_n)\right) + H\left(q(\cdot \mid x, D_n), \pi(\cdot \mid x)\right).$$

The second inequality (2) uses the fact that the Hellinger distance $H(p, q)$ is bounded above by the square root of the KL divergence $KL(p \parallel q)$, such that

$$H(p, q)^2 \leq \frac{1}{2} KL(p \parallel q).$$

Thus, we can bound the Hellinger distance by the KL divergence. In the third inequality (3), we make use of assumption

$$\mathbb{E}_x\left[KL\left(p_{\hat{\theta}}(\cdot \mid x, D_n), \pi(\cdot \mid x, D_n)\right)\right] < \epsilon,$$

and utilize the definition of the Hellinger distance. In (4), we partition the domain into two regions– one where the Hellinger distance $H(\pi, q)$ exceeds ϵ and another where it is less than or equal to ϵ–and use this partitioning to demonstrate the inequality.

Finally, in (5), by posterior consistency, the region where the Hellinger distance is greater than ϵ vanishes as $n \to \infty$ such that

$$\Pi^n\left\{q : H(\pi, q) > \epsilon\right\} \to 0 \quad \text{almost surely.}$$

Since ϵ is arbitrary, we can conclude that

$$\mathbb{E}_x\left[H\left(p_{\hat{\theta}}(\cdot \mid x, D_n), \pi(\cdot \mid x)\right)\right] \xrightarrow{n \to \infty} 0 \quad \text{almost surely.}$$

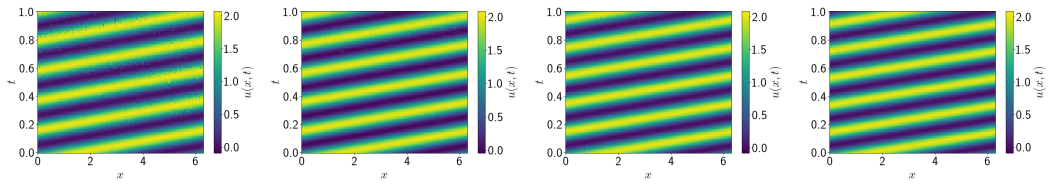
□

B Experiments at PINN Failure Modes

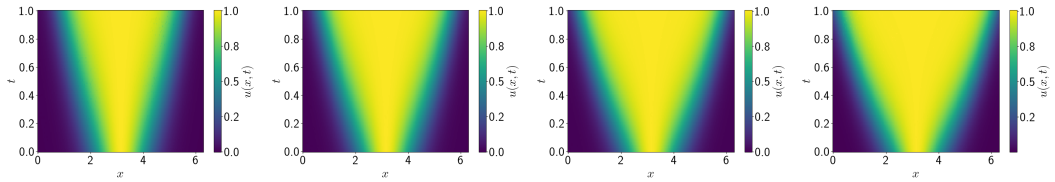
Referring to [7] and [22], we test our method on PINN's major failure modes: $\beta \in [30, 40]$ with an initial condition $1 + \sin(x)$ and $\rho \in [1, 10]$ with an initial condition $\mathcal{N}\left(\pi, \left(\frac{\pi}{2}\right)^2\right)$. We have trained our model with this range with **P1** prior and evaluate L_1 mean absolute and L_2 relative errors. The following are major results and solution profiles at failure modes.

Table 4: The L_1 mean absolute and L_2 relative error at PINN failure modes.

Trained Coefficient Range	**Test Coefficient Value**	L_2 **Error Type**		**Average Error**	
		Abs.err	Rel.err	Abs.err	Rel.err
$\beta \in [30, 40]$	$\beta = 30$	0.2483	0.2516	0.1280	0.1328
	$\beta = 31$	0.1029	0.1111		
	$\beta = 32$	0.0803	0.0882		
	$\beta = 33$	0.0806	0.0801		
$\rho \in [1, 10]$	$\rho = 4$	0.0071	0.0160	0.0048	0.0097
	$\rho = 5$	0.0029	0.0054		
	$\rho = 6$	0.0033	0.0063		
	$\rho = 7$	0.0058	0.0112		



(a) $\beta = 30$ (b) $\beta = 31$ (c) $\beta = 32$ (d) $\beta = 33$



(e) $\rho = 4$ (f) $\rho = 5$ (g) $\rho = 6$ (h) $\rho = 7$

Figure 5: The solution profiles at PINN failure modes: (a), (b), (c) and (d) for $\beta \in [30, 40]$ with initial condition $1 + \sin(x)$ and (e), (f), (g) and (h) for $\rho \in [1, 10]$ with initial condition $\mathcal{N}\left(\pi, \left(\frac{\pi}{2}\right)^2\right)$. The solution profile is constructed using the union of 1,000 test prediction points and the remaining ground truth points.