COUPLEDFLOW: One-Step Neural Operators for Coupled Multi-Physics PDEs

Trong Khiem Tran*
Washington State University
khiem.tran@wsu.edu

Minh Long Bui
Washington State University
trongminhlong.bui@wsu.edu

Phi Le Nguyen
Hanoi University of Science and Technology
lenp@soict.hust.edu.vn

Mrinal Sen
University of Texas, Austin
mrinal@utexas.edu

Sanjay Srinivasan
Penn State University
szs27@psu.edu

Trong Nghia Hoang*
Washington State University
trongnghia.hoang@wsu.edu

Jana Doppa*
Washington State University
jana.doppa@wsu.edu

Abstract

This paper introduces COUPLEDFLOW, a new neural operator learning framework for multi-physics PDE systems that captures the co-evolution of multiple physical processes by modeling their cross-process interactions. Unlike existing neural operator methods that primarily target single-physics settings, COUPLEDFLOW employs a novel structure-constrained flow matching algorithm to learn process-specific mean velocity fields conditioned on the full multi-physics state. This enables fast and accurate one-step prediction of coupled PDE systems that would otherwise require expensive numerical simulations. A key challenge in training COUPLEDFLOW however lies in sampling structure-constrained conditional flow interpolants that capture cross-process correlations and serve as training examples for flow matching. To address this challenge, we derive a *multi-physics mean-flow consistency identity* that characterizes cross-process interactions and defines the structure-constrained mean-flow matching loss. To provide informative training samples for flow matching with this loss, we develop a principled method for sampling *correlated conditional interpolants based on posterior multi-output Gaussian processes*, which capture richer cross-process correlations than the commonly used endpoint-conditioned linear interpolants. We also establish a theoretical result showing that the Wasserstein-2 distance between the predicted and oracle PDE solution distributions is upper-bounded in terms of the structure-constrained flow matching loss and the smoothness of the learned flow. Experiments across diverse coupled PDE benchmarks demonstrate that COUPLEDFLOW consistently outperforms existing state-of-the-art neural operators and generative PDE solvers.

1 Introduction

Partial differential equations (PDEs) underpin the modeling of physical systems across science and engineering (Turing, 1952; Gao and Ghassemi, 2020). In particular, PDEs describe how physical quantities evolve and interact across space and time through differential equations, which can be solved using numerical simulations. However, these simulations can be computationally expensive,

*Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

especially for high-dimensional physical systems. To mitigate this computational cost, a prominent direction in scientific machine learning is to learn surrogate mappings directly from simulation data that approximate the underlying solution operators. Models that learn such mappings between input and solution function spaces are commonly referred to as neural operators. Notable state-of-the-art methods include FNO (Li et al., 2021), DeepONet (Lu et al., 2021) and CViT (Wang et al., 2025).

Research Gap. Existing neural operators have primarily been developed for single-physics PDEs, whereas many real-world systems involve interacting physical processes that co-evolve and influence one another. Examples include reaction-diffusion systems in chemistry and biology (Turing, 1952), which model non-linear interactions between multiple species, and thermo-hydro-mechanical systems (Gao and Ghassemi, 2020) where temperature, pressure, and deformation variables are tightly coupled. In such systems, the future state of each process depends on the current states of others and thus cannot be predicted accurately in isolation. Therefore, developing neural operators that explicitly capture these cross-physics dependencies is important for prediction accuracy. However, extending single-physics neural operators to coupled systems is challenging. For example, naively concatenating internal neural representations across different physical processes often leads to poor results (Xiao et al., 2023), as this approach does not explicitly model cross-physics interactions.

More recent work addresses this challenge by introducing explicit cross-physics information exchange into neural operator architectures. For example, CMWNO (Xiao et al., 2023) models coupling through multi-wavelet decomposition, while COMPOL (Sun et al., 2025) maintains process-specific FNO streams and exchanges information through recurrent or attention-based aggregation. Despite their improved performance, these approaches primarily capture cross-physics interactions through architectural design and endpoint regression which largely focuses on mapping initial states to final states without explicitly learning correlations among physical processes along their intermediate transport paths. This limitation can lead to suboptimal predictive performance (see Section 4).

New Perspective. To explicitly capture correlations among physical processes along intermediate transport paths, we revisit neural operator learning from the lens of a mass-transport flow. We view the mapping from initial conditions to PDE solutions as transporting a source distribution over initial states to a target distribution over solutions. Such transport flow can be learned through conditional flow (FM) (Lipman et al., 2022; Baldan et al., 2025; Utkarsh et al., 2025) which has recently emerged as a powerful approach for generative modeling. Importantly, by leveraging mean-flow parameterization (Geng et al., 2025), we can learn intermediate transport dynamics while preserving the one-step prediction capability of neural operators.

However, extending FM to coupled multi-physics systems is challenging as it requires accounting for cross-process dependencies in both the parameterized flow and its training examples. To understand this challenge, we view FM training from a *proposer-filter* perspective where (1) conditional interpolants propose transport paths connecting paired initial states and PDE solutions as training examples; and (2) the parameterized flow acts as a filter by imposing consistency constraints on the learned transport. Since FM amortizes information from these conditional paths into the parameterized flow, constructing structure-constrained interpolants that capture cross-process correlations is critical for effective learning of coupled multi-physics systems.

► In multi-physics modeling scenarios, two key challenges remain unresolved: (1) how to develop a mean-flow representation that explicitly captures interactions among dependent physical processes; and (2) how to construct correlated intermediate transport paths that provide informative supervision for learning these interactions. Our experiments show that a vanilla application of FM with independent linear interpolants is often suboptimal in coupled PDE systems (see Section 4).

Main Contributions. To address these challenges, we make the following technical contributions:

1. Algorithm Design. We develop CoupledFlow, a one-step multi-physics neural operator built on structure-constrained mean-flow matching to explicitly learn cross-process interactions. In particular, we derive a *multi-physics mean-flow consistency identity* that characterizes the coupling among process-specific transport flows. This identity enables us to impose structural constraints on the parameterized multi-physics mean-flow representation by penalizing constraint violations during flow matching (Section 3.1). To provide informative training samples for the structure-constrained flow matching loss, we develop a principled method for sampling *correlated conditional interpolants* from posterior multi-output Gaussian processes, which capture richer cross-process correlations than conventional endpoint-conditioned linear interpolants(Section 3.2).

2. Theoretical Analysis. We characterize test-time prediction performance in terms of the Wasserstein-2 distance between predicted and oracle PDE solution distributions and show that it is upper-bounded in terms of the coupled mean-flow matching loss (see Theorem 2, Section 3.3). This analysis also clarifies how the smoothness of the learned transport flow influences prediction error. We further empirically evaluate the resulting error bounds for correlated GP-based interpolants and standard linear interpolants (see Section 4).

3. Empirical Evaluation. We evaluate COUPLEDFLOW on diverse coupled PDE benchmarks spanning different physical systems. The results demonstrate significant improvements in prediction accuracy over existing (coupled) neural operators and generative PDE solvers. We also conduct ablation studies to isolate the contributions of the coupled mean-flow representation and GP-based correlated interpolants, demonstrating the importance of both components for accurate multi-physics prediction (see Section 4).

2 Background and Problem Setup

A. Coupled Physics Systems. We consider a multi-physics system consisting of m interacting physical processes $\{\boldsymbol{u}_i(\boldsymbol{x},t)\}_{i=1}^m$ modeled by coupled PDEs. Each PDE can be characterized as:

$$\boldsymbol{u}_r\big(\boldsymbol{x},\tau\big) \triangleq T_r\Big[\boldsymbol{x} \mid \mathcal{J}^\kappa \boldsymbol{u}_1(\boldsymbol{x}),\ldots,\mathcal{J}^\kappa \boldsymbol{u}_m(\boldsymbol{x})\Big] \text{ where } \mathcal{J}^\kappa \boldsymbol{u}_i(\boldsymbol{x}) \triangleq \Big\{\partial_{\boldsymbol{x}}^p \partial_t^q \boldsymbol{u}_i(\boldsymbol{x}) \mid p+q \le \kappa\Big\}, \quad (1)$$

Here, T_r maps from the initial spatiotemporal condition $\boldsymbol{x}$ at time $t=0$ to the corresponding solution $\boldsymbol{u}_r(\boldsymbol{x},\tau)$ at the target time $t=\tau$. As each operator T_r depends on differential information from other processes, each individual process $\boldsymbol{u}_r(\boldsymbol{x},t)$ cannot be solved in isolation. Furthermore, forward simulation of the coupled system can be computationally expensive. This motivates learning a data-driven approximation of the operator $\boldsymbol{u}_r(\boldsymbol{x},\tau) = T_r[\boldsymbol{x} \mid \ldots]$ using an efficient one-step predictor $\boldsymbol{u}_r(\boldsymbol{x},\tau) \simeq \boldsymbol{u}_r(\boldsymbol{x},\tau;\theta)$, thus avoiding repeated numerical simulation of T_r.

Existing neural operators differ in how they parameterize the surrogate $\boldsymbol{u}_r(\boldsymbol{x},\tau;\theta)$ and learn its parameters θ. However, most approaches learn to map between initial conditions and corresponding simulation outputs during training (Li et al., 2021; Lu et al., 2021; Huang et al., 2024) without explicitly modeling how interactions among physical processes drive the co-evolution of their intermediate states. For multi-physics systems, such interactions play a central role in determining the solution, motivating a learning framework that explicitly captures cross-process dependencies.

B. Flow-based Neural Operator. To explicitly model intermediate transport dynamics,, we can view the solution operator T_r as transporting a source distribution over initial conditions $\boldsymbol{x}$. The target distribution is implicitly defined as the push-forward of the source distribution π_s through T_r. We can then parameterize this transport using an ODE flow $\mathrm{d}\boldsymbol{x}(t)/\mathrm{d}t = \boldsymbol{u}(\boldsymbol{x}(t),t) \simeq \boldsymbol{u}_\theta(\boldsymbol{x}(t),t)$.

The neural velocity field $\boldsymbol{u}_\theta(\boldsymbol{x}(t),t)$ can be learned using conditional linear interpolants $\boldsymbol{x}(t \mid \boldsymbol{x},\boldsymbol{x}') = (1-t/\tau)\boldsymbol{x} + (t/\tau)\boldsymbol{x}'$[2] and amortizing them into $\boldsymbol{u}_\theta(\boldsymbol{x}(t),t)$ (Lipman et al., 2022):

$$\theta \triangleq \arg\min_\theta \mathbb{E}_{t\in\mathcal{U}(0,\tau)}\mathbb{E}_{\boldsymbol{x}\sim\pi_s}\mathbb{E}_{\boldsymbol{x}'\sim\pi_\tau}\big\|\boldsymbol{u}_\theta(\boldsymbol{x}(t),t) - \mathrm{d}\boldsymbol{x}(t \mid \boldsymbol{x},\boldsymbol{x}')/\mathrm{d}t\big\|^2, \quad (2)$$

where $\mathrm{d}\boldsymbol{x}(t \mid \boldsymbol{x},\boldsymbol{x}')/\mathrm{d}t = (1/\tau)(\boldsymbol{x}'-\boldsymbol{x})$. To enable one-step prediction during test time, the above can be further empowered with a mean flow parameterization (Geng et al., 2025) as detailed next.

C. Mean-Flow Neural Operator. We will use a direct parameterization for the mean or average velocities $\bar{\boldsymbol{u}}_\theta(\boldsymbol{x}(t),s,t) \simeq \bar{\boldsymbol{u}}(\boldsymbol{x}(t),s,t)$ over interval $[s,t]$ where

$$\bar{\boldsymbol{u}}_\theta(\boldsymbol{x}(t),s,t) \;=\; (t-s)^{-1}\int_s^t \boldsymbol{u}_\theta(\boldsymbol{x}(\varepsilon),\varepsilon)\mathrm{d}\varepsilon \;=\; (t-s)^{-1}(\boldsymbol{x}(t)-\boldsymbol{x}(s)), \quad (3)$$

via the fundamental theorem of calculus. By slight abuse of notation, the mean flow is parameterized as a function of the current state and time interval. Differentiating both sides w.r.t t, we obtain:

$$\boldsymbol{u}_\theta(\boldsymbol{x}(t),t) \;=\; \bar{\boldsymbol{u}}_\theta(\boldsymbol{x}(t),s,t) \;+\; (t-s)\cdot \mathrm{d}\bar{\boldsymbol{u}}_\theta(\boldsymbol{x}(t),s,t)/\mathrm{d}t, \quad (4)$$

which means we can either parameterize the instantaneous velocity $\boldsymbol{u}_\theta(\boldsymbol{x}(t),t)$ directly or compute it via the parameterization of the mean velocities $\bar{\boldsymbol{u}}_\theta(\boldsymbol{x}(t),s,t)$. The latter preserves the flow structure

[2] As we consider $t \in [0,\tau]$, this construction guarantees that the interpolation reaches $\boldsymbol{x}'$ when $t=\tau$.

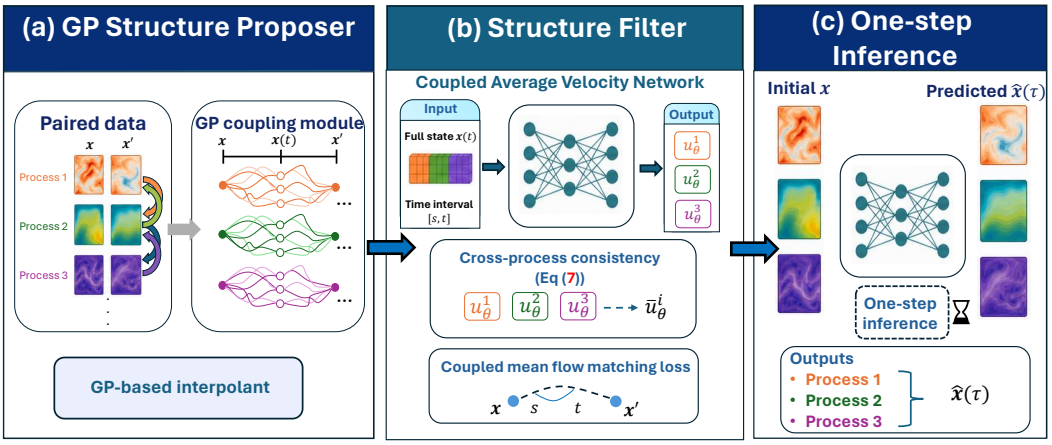

Figure 1: Workflow diagram of the proposed COUPLED-FLOW framework which consists of three main components: (a) a structure proposer that constructs Gaussian process-based conditional flow interpolants to capture cross-physics dependencies (Section 3.2); (b) a structure filter that constrains coupled transport dynamics through the mean-flow consistency identity (Section 3.1); and (c) a one-step inference procedure for efficient multi-physics prediction at testing time (Section 3.3).

while enabling fast one-step inference during test time. It can be learned via plugging Eq. (4) into Eq. (2) and taking expectation over $s \sim \mathcal{U}(0, \tau)$ and $t \sim \mathcal{U}(s, \tau)$ (Geng et al., 2025).

D. Multi-Physics Challenge. At a conceptual level, the mean-flow representation admits a rich path-level structure that extends beyond direct matching between source and target states. In particular, the mean-flow relation in Eq. (3) constrains how the average velocity evolves along admissible transport paths, suggesting that the representation has the capacity to capture how multiple physical processes co-evolve in coupled systems. However, existing mean flow formulations (Geng et al., 2025) are developed almost exclusively for single-physics settings where the main goal is to enable one-step inference. Consequently, despite its structural capacity, the representation is ultimately learned via amortizing over conditional linear interpolations between source and target samples as in Eq. (2), leaving its path-level learning potential largely underutilized.

This is sufficient in single-physics settings where the only requirement on the interpolation is that it delivers a valid transport from source to target. In multi-physics settings, however, we need the transport to reflect how different components of the state are correlated along the path. The linear interpolation $\boldsymbol{x}(t \mid \boldsymbol{x}, \boldsymbol{x}') = (1 - t/\tau)\boldsymbol{x} + (t/\tau)\boldsymbol{x}'$ unfortunately offers no mechanism for this. It produces intermediate states as a coordinate-wise function of endpoints alone, leaving no room to capture any structural relationship among components of $\boldsymbol{x}(t)$. To address this limitation, we develop a coupled mean flow construction in which the linear interpolation is replaced by a correlated interpolation that acts as a learning device for proposing viable cross-physics interaction structures, which are subsequently amortized into the neuralized mean flow field (see Section 3).

3 COUPLEDFLOW: Learning One-Step Multi-Physics Neural Operators

High-Level Overview of COUPLEDFLOW. This section develops a coupled mean-flow framework for one-step multi-physics neural operators. A key starting point is the mean-flow identity in Eq. (3), which can be expanded to derive a multi-physics mean-flow consistency relation that constrains interactions among different physical processes along transport paths. The overall framework consists of two complementary components: (1) a structure proposer that generates candidate cross-physics interaction patterns through structure-constrained flow interpolations, and (2) a structure filter that uses the multi-physics consistency relation to constrain these patterns during learning (see Fig. 1).

That is, the correlated interpolations provide learning signals for the coupled mean-flow matching loss during training. The consistency identity on the other hand constrains the learned mean-flow representation. At inference time, the learned mean-flow operator can be used to predict the coupled multi-physics state in a single step without explicitly simulating the entire transport flow.

The remainder of this section makes each of these steps concrete. Section 3.1 derives the multi-physics consistency identity. Section 3.2 constructs the multi-output GP-based interpolation. Finally, Section 3.3 describes the high-level algorithm with practical training methods.

3.1 Structure Filter: Multi-Physics Mean Flow Consistency Identity

Let $\boldsymbol{x}(t) = (\boldsymbol{x}_1(t), \boldsymbol{x}_2(t), \ldots, \boldsymbol{x}_m(t))$ denote a multi-physics state comprising the states of m coupled physical processes at time t. Let $\bar{\boldsymbol{u}}^i_\theta(\boldsymbol{x}(t), s, t)$ denote the mean flow representation for the i-th physical process over the time interval $[s, t]$. The dependence among these physical processes is captured by conditioning each individual mean flow representation on the entire multi-physics state. The corresponding instantaneous velocity at time t can then be obtained from Eq. (4) as follows:

$$\boldsymbol{u}^i_\theta\big(\boldsymbol{x}(t), t\big) \;=\; \bar{\boldsymbol{u}}^i_\theta\big(\boldsymbol{x}(t), s, t\big) \;+\; (t-s)\cdot \mathrm{d}\bar{\boldsymbol{u}}^i_\theta\big(\boldsymbol{x}(t), s, t\big)/\mathrm{d}t\,. \tag{5}$$

Applying the chain rule to the derivative term on the RHS of Eq. (5) gives:

$$\mathrm{d}\bar{\boldsymbol{u}}^i_\theta\big(\boldsymbol{x}(t), s, t\big)/\mathrm{d}t \;=\; \partial_t \bar{\boldsymbol{u}}^i_\theta\big(\boldsymbol{x}(t), s, t\big) + \sum_{c=1}^{m} \boldsymbol{u}^c_\theta\big(\boldsymbol{x}(t), t\big)\partial_{\boldsymbol{x}_c(t)} \bar{\boldsymbol{u}}^i_\theta\big(\boldsymbol{x}(t), s, t\big)\,. \tag{6}$$

Substituting Eq. (6) into Eq. (5) yields the **multi-physics consistency identity**,

$$\boldsymbol{u}^i_\theta\big(\boldsymbol{x}(t), t\big) = \bar{\boldsymbol{u}}^i_\theta\big(\boldsymbol{x}(t), s, t\big) + (t-s)\left[\partial_t \bar{\boldsymbol{u}}^i_\theta\big(\boldsymbol{x}(t), s, t\big) + \sum_{c=1}^{m} \boldsymbol{u}^c_\theta\big(\boldsymbol{x}(t), t\big)\partial_{\boldsymbol{x}_c(t)} \bar{\boldsymbol{u}}^i_\theta\big(\boldsymbol{x}(t), s, t\big)\right] \tag{7}$$

Intuitively, this means that a learned mean-flow representation that accurately models the coupled PDE dynamics must satisfy this constraint. Therefore, the consistency identity can be used as a structure filter in conditional flow matching.

More specifically, this filter constrains how the cross-physics interaction patterns proposed by endpoint-conditioned interpolations are incorporated into the learned mean-flow representation. Interaction patterns consistent with Eq. (7) can be captured by the learned mean-flow representation, while inconsistent patterns are constrained by the coupled mean-flow matching objective. As a consequence, the learned representation captures cross-physics dependencies while satisfying the mean-flow consistency identity. Mathematically, the classical conditional flow matching approach Lipman et al. (2022) can be generalized as follows:

$$\theta \;=\; \arg\min_\theta \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{s,t}\mathbb{E}_{(\boldsymbol{x},\boldsymbol{x}')}\mathbb{E}_{\boldsymbol{x}_i(t)}\Big\|\boldsymbol{u}^i_\theta\big(\mathbb{E}[\boldsymbol{x}(t)], t\big) \;-\; \mathrm{d}\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}')/\mathrm{d}t\Big\|^2\,, \tag{8}$$

Unlike the original loss in Eq. (2), we do not parameterize $\boldsymbol{u}^i_\theta(\boldsymbol{x}(t), t)$ directly but instead use the form induced by the mean-flow representation in Eq. (7). This introduces an additional time variable $s \in [0, \tau]$ over which we average the loss in Eq. (8). The next key component of our framework is the construction of multi-physics interpolations that capture cross-physics dependencies and provide learning signals for the coupled mean-flow matching loss, as detailed in Section 3.2 below.

3.2 Structure Proposer: Conditional Flow Interpolation Construction

As discussed in Section 3.1, the mean-flow consistency identity in Eq. (7) acts as a structural filter that constrains the cross-physics interaction patterns learned from candidate flow interpolants. The effectiveness of the framework therefore depends on the quality of the endpoint-conditioned flow interpolations $\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}')$ for each process i, which propose how the intermediate states of different physical processes co-evolve along the transport path. As such, the interpolations must be capable of generating intermediate coupled states that capture dependencies among multiple processes.

A common choice in conditional flow matching (Lipman et al., 2022) is the deterministic linear interpolation $\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}') = (1 - t/\tau)\boldsymbol{x}_i + (t/\tau)\boldsymbol{x}'_i$ which reaches $\boldsymbol{x}'_i$ when $t = \tau$[3]. Although commonly used in single-physics settings, this construction can be restrictive in coupled systems because the intermediate states of different processes are determined by their endpoints without

[3]Here, $\boldsymbol{x}_i$ and $\boldsymbol{x}'_i$ denote the states of process i corresponding to the multi-physics states $\boldsymbol{x}$ and $\boldsymbol{x}'$, respectively.

explicitly modeling cross-physics dependencies along the transport path. As a result, it may fail to capture important cross-physics correlations, limiting the effectiveness of the structural filtering mechanism in Eq. (7). To address this limitation, we replace the independent linear interpolation with a *correlated interpolation induced from a multi-output Gaussian process (GP) posterior* conditioned on the endpoint pair $(\boldsymbol{x}, \boldsymbol{x}')$. In particular, we assume that the multi-physics path:

$$\boldsymbol{x}(t \mid \boldsymbol{x}, \boldsymbol{x}') \quad \triangleq \quad (\boldsymbol{x}_1(t \mid \boldsymbol{x}, \boldsymbol{x}'), \boldsymbol{x}_2(t \mid \boldsymbol{x}, \boldsymbol{x}'), \ldots, \boldsymbol{x}_m(t \mid \boldsymbol{x}, \boldsymbol{x}')) \,, \tag{9}$$

is stochastic and modeled using a multi-output GP (Alvarez et al., 2012) which provides a principled framework for capturing path-level correlations among dependent spatiotemporal processes. For two dependent processes $(\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}'), \boldsymbol{x}_c(t' \mid \boldsymbol{x}, \boldsymbol{x}'))$, the joint Gaussian distribution is given by:

$$\begin{bmatrix} \boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}') \\ \boldsymbol{x}_c(t' \mid \boldsymbol{x}, \boldsymbol{x}') \end{bmatrix} \quad \sim \quad \mathbb{N}\left(\begin{bmatrix} \boldsymbol{m}_i(t \mid \boldsymbol{x}, \boldsymbol{x}') \\ \boldsymbol{m}_c(t' \mid \boldsymbol{x}, \boldsymbol{x}') \end{bmatrix}, \begin{bmatrix} \boldsymbol{K}_{ii}(t, t' \mid \boldsymbol{x}, \boldsymbol{x}') & \boldsymbol{K}_{ic}(t, t' \mid \boldsymbol{x}, \boldsymbol{x}') \\ \boldsymbol{K}_{ci}(t, t' \mid \boldsymbol{x}, \boldsymbol{x}') & \boldsymbol{K}_{cc}(t, t' \mid \boldsymbol{x}, \boldsymbol{x}') \end{bmatrix} \right) \,, \tag{10}$$

whose mean and kernel functions determine the marginal correlation structure between these processes. At each time step t, we can thus generate the correlated state $\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}')$ given $\boldsymbol{x}_c(t \mid \boldsymbol{x}, \boldsymbol{x}')$ via constructing the posterior GP marginal:

$$\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}') \mid \boldsymbol{x}_c(t \mid \boldsymbol{x}, \boldsymbol{x}') \quad \sim \quad \mathbb{N}\left(\boldsymbol{m}_{i|c}(t), \boldsymbol{K}_{i|c}(t)\right) \,, \tag{11}$$

where the posterior mean and covariance functions are given by

$$\begin{aligned} \boldsymbol{m}_{i|c}(t) \quad &= \quad \boldsymbol{m}_i(t \mid \boldsymbol{x}, \boldsymbol{x}') + \boldsymbol{K}_{ic}(t \mid \boldsymbol{x}, \boldsymbol{x}') \boldsymbol{K}_{cc}(t \mid \boldsymbol{x}, \boldsymbol{x}')^{-1} \left(\boldsymbol{x}_c(t \mid \boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{m}_c(t \mid \boldsymbol{x}, \boldsymbol{x}')\right) \,, \\ \boldsymbol{K}_{i|c}(t) \quad &= \quad \boldsymbol{K}_{ii}(t \mid \boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{K}_{ic}(t \mid \boldsymbol{x}, \boldsymbol{x}') \boldsymbol{K}_{cc}(t \mid \boldsymbol{x}, \boldsymbol{x}')^{-1} \boldsymbol{K}_{ci}(t \mid \boldsymbol{x}, \boldsymbol{x}') \,. \end{aligned} \tag{12}$$

As such, $\boldsymbol{m}_{i|c}(t) = \mathbb{E}[\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}')]$ and we can use this posterior mean as the correlated multi-physics interpolation or structure proposer that supplies the learning signal for the flow matching loss in Eq. (8). This means that we optimize the flow matching loss using the average interpolation $\mathbb{E}[\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}')] = \boldsymbol{m}_{i|c}(t)$ rather than the sampled interpolation $\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}')$. Consequently, the resulting loss is given by:

$$\theta \quad = \quad \arg\min_{\theta} \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}_{s,t} \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{x}')} \mathbb{E}_{\boldsymbol{x}_i(t)} \left\| \boldsymbol{u}_\theta^i\big(\mathbb{E}[\boldsymbol{x}(t)], t\big) \quad - \quad \mathrm{d}\mathbb{E}[\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}')]/\mathrm{d}t \right\|^2 \,, \tag{13}$$

Although this changes the form of the loss in Eq. (8), it does not change its theoretical optimizer – see Theorem 1 below. An important advantage is that although $\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}')$ is stochastic and may not be differentiable, its posterior mean $\boldsymbol{m}_{i|c}(t)$ has a closed form in Eq. (12) allowing us to compute the corresponding instantaneous velocity $\mathrm{d}\mathbb{E}[\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}')]/\mathrm{d}t = \mathrm{d}\boldsymbol{m}_{i|c}(t)/\mathrm{d}t$ which is required for computing the flow matching loss in Eq. (8).

Theorem 1. *The theoretical optimizers of the conditional mean flow matching losses in Eq.* (8) *and Eq.* (13) *are the same. A detailed proof is provided in Appendix A.2.*

The key advantage of GP-based interpolation is its ability to mitigate the structural bias introduced by independent linear interpolation. In particular, linear interpolation determines each process's intermediate state solely from its own endpoints, without explicitly accounting for how other processes evolve along the transport path. This can be restrictive in coupled multi-physics systems, where the intermediate evolution of one process depends on the states of others.

In contrast, the GP posterior mean $\boldsymbol{m}_{i|c}(t)$ explicitly conditions the interpolation of process i on the intermediate state of process c through the cross-covariance operators $\boldsymbol{K}_{ic}$ and $\boldsymbol{K}_{ci}$. Consequently, the proposed intermediate states can reflect cross-process dependencies beyond those implicitly encoded by the endpoint pairs. In practice, we can initialize the interpolation for one process $\boldsymbol{x}_c(t \mid \boldsymbol{x}, \boldsymbol{x}')$ using standard linear interpolation between its endpoints and subsequently generate correlated interpolations for the remaining processes using the posterior GP marginals in Eq. (11). This provides a practical mechanism for constructing coupled intermediate states while incorporating cross-physics correlations through the GP covariance operators. Moreover, when conditioned on the endpoint observations, the GP-based interpolations preserve both endpoint conditions, as theoretically established in Proposition 3.

For practical implementation, we use zero-mean Gaussian processes (Rasmussen and Williams, 2006), a standard choice such that the prior mean is zero: $\boldsymbol{m}_i(t \mid \boldsymbol{x}, \boldsymbol{x}') = 0$. The covariance operators $\boldsymbol{K}_{ic}(t, t' \mid \boldsymbol{x}, \boldsymbol{x}')$ are defined using smooth kernels, such as the squared-exponential kernel (Rasmussen and Williams, 2006), which encourage temporally smooth interpolations while capturing correlations among dependent physical processes.

3.3 Coupled Mean-Flow Matching: Algorithms and Guarantees

A. Workflow. The two key components developed in Section 3.1 and Section 3.2 are integrated into the proposed COUPLEDFLOW algorithm for learning a multi-physics neural operator (see Alg. 1) that enables one-step prediction at inference time (see Alg. 2).

Algorithm 1 Constructing Multi-Physics Neural Operator

input: simulation dataset D, number of training epochs n, prediction time $\tau > 0$
output: optimized coupled mean flow parameter θ.
for $i = 1$ to n **do**
  sample $s \in \mathcal{U}(0, \tau)$ and $t \in \mathcal{U}(s, \tau)$
  sample $(\boldsymbol{x}, \boldsymbol{x}') \sim D$
  sample randomly $c \in [m]$ and construct $\boldsymbol{x}_c(t \mid \boldsymbol{x}, \boldsymbol{x}') = (1 - t/\tau)\boldsymbol{x}_c + (t/\tau)\boldsymbol{x}'_c$
  compute $\mathbb{E}[\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}')]$ for each $i \in [m], i \neq c$ via Eq. (12)
  compute the learning signal $\mathrm{d}\mathbb{E}[\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}')]/\mathrm{d}t$ for Eq. (13)
  update θ via running Adam optimizer (Kingma and Ba, 2015) on Eq. (13)
end for
return the optimized coupled mean flow parameterization θ

Algorithm 2 One-Step Prediction with Coupled Mean Flow

input: initial multi-physics state $\boldsymbol{x}(0) = (\boldsymbol{x}_1(0), \ldots, \boldsymbol{x}_m(0))$
output: predicted multi-physics solution $\hat{\boldsymbol{x}}(\tau) = (\hat{\boldsymbol{x}}_1(\tau), \ldots, \hat{\boldsymbol{x}}_m(\tau))$
for $i = 1$ to m **do**
  $\hat{\boldsymbol{x}}_i(\tau) \leftarrow \boldsymbol{x}_i(0) \;+\; \tau\, \bar{\boldsymbol{u}}^i_\theta(\boldsymbol{x}(0), 0, \tau)$
end for
return $\hat{\boldsymbol{x}}(\tau) = (\hat{\boldsymbol{x}}_1(\tau), \ldots, \hat{\boldsymbol{x}}_m(\tau))$

B. Performance Guarantee. We establish a theoretical guarantee relating the proposed algorithm's training objective to its test-time prediction error. Given a test-time distribution $\boldsymbol{x}_i(0) \sim \pi^i_s(\boldsymbol{x})$, let $\pi^i_\tau(\boldsymbol{x})$ denote the marginal distribution over the corresponding (oracle) output $\boldsymbol{x}(\tau)$. Similarly, let $\bar{\pi}^i_\tau(\boldsymbol{x}')$ denote the marginal distribution over the predicted outputs of our multi-physics neural operator COUPLEDFLOW when testing inputs are sampled from $\pi^i_s(\boldsymbol{x})$.

We measure test-time prediction error $\mathrm{ERR}(\theta)$ using the Wasserstein-2 distance (on ℓ_2 cost metric) between π^i_τ and $\bar{\pi}^i_\tau$, averaged across all physical processes: $\mathrm{ERR}(\theta) = (1/m)\sum_{i=1}^m W_2(\pi^i_\tau, \bar{\pi}^i_\tau)$. The following theorem establishes an upper bound on this test-time prediction error in terms of the coupled mean-flow matching loss in Eq. (13).

Theorem 2. *Let ℓ_r denote the Lipschitz constant of the instantaneous multi-physics velocity $\boldsymbol{u}_\theta(\boldsymbol{x}(r), r) = (\boldsymbol{u}^1_\theta(\boldsymbol{x}(r), r), \ldots, \boldsymbol{u}^m_\theta(\boldsymbol{x}(r), r))$. Let $\kappa \triangleq \exp(\int_0^\tau \ell_r \mathrm{d}r)$, it follows that*

$$\mathrm{ERR}(\theta) \quad \leq \quad \kappa\tau\sqrt{L(\theta)}, \tag{14}$$

where $L(\theta)$ denotes the coupled mean flow matching loss in Eq. (13) *which is restated below:*

$$L(\theta) \quad \triangleq \quad \frac{1}{m}\sum_{i=1}^m \mathbb{E}_{s,t}\mathbb{E}_{(\boldsymbol{x},\boldsymbol{x}')}\mathbb{E}_{\boldsymbol{x}_i(t)}\Big\|\boldsymbol{u}^i_\theta\big(\mathbb{E}[\boldsymbol{x}(t)], t\big) \;-\; \mathbb{E}[\mathrm{d}\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}')/\mathrm{d}t]\Big\|^2. \tag{15}$$

A detailed proof of this theorem is provided in Appendix B. Here, τ denotes the prediction time.

This result implies that the generalization error can be small when (1) θ is well optimized, resulting in a small training loss $L(\theta)$; and (2) the Lipschitz constant ℓ_r of the instantaneous multi-physics velocity $u_\theta(\boldsymbol{x}(r), r)$ is small. This reflects the interplay between the algorithm and the intrinsic complexity of the physics system. For PDE systems with rapidly varying dynamics, the Lipschitz constant ℓ_r can be large due to the sensitivity of the instantaneous velocity to changes in the system state. As a consequence, learning accurate neural operators for such systems can be more challenging, and the resulting bound on test-time prediction error may be less informative. The bound becomes less informative for longer prediction horizons τ, as expected. The tightness of this error bound is also validated empirically in Appendix C and Figure 3.

4 Empirical Evaluation and Analysis

This section presents empirical results comparing CoupledFlow and SOTA baselines. The official code repository is provided at `https://github.com/khiembk/CoupledFlow`.

4.1 Experimental Setup

Benchmarks: Following prior work (Sun et al., 2025; Xiao et al., 2023), we evaluate Coupled-Flow on five coupled multi-physics PDE benchmarks: 1-D Lotka-Volterra equations (Murray, 2002) (2 processes), 1-D Belousov-Zhabotinsky equations (Petrov et al., 1993) (3 processes), 2-D Gray-Scott equations (Pearson, 1993) (2 processes), 2-D multiphase flow of oil and water in geological storage (Bear and Cheng, 2010) (Hashemi et al., 2021) Abou-Kassem et al. (2013) (2 processes), and a 2-D Thermal-Hydro-Mechanical problem (Gao and Ghassemi, 2020) (5 processes). We also evaluate on two PDE benchmarks: Diffusion Reaction-2D (2 processes) from PDEBench (Takamoto et al., 2024) and Navier-Stokes-2D (3 processes) from PDEArena (Gupta and Brandstetter, 2022).

Baselines: We compare CoupledFlow against state-of-the-art neural operators and generative PDE solvers: COMPOL (Sun et al., 2025), FNO (Li et al., 2021), CViT (Wang et al., 2025), UFNO (Wen et al., 2022), DiffusionPDE (Huang et al., 2024), Transolver (Wu et al., 2024), and CMWNO (Xiao et al., 2023). COMPOL and CMWNO are specifically designed for coupled PDEs, while the remaining baselines are not explicitly designed for multi-physics modeling.

Evaluation Methodology: We evaluate prediction accuracy using the relative L_2 error between predicted and ground-truth PDE solutions, defined as

$$\text{Error}_{L_2} = \frac{\|\hat{\boldsymbol{z}} - \boldsymbol{z}\|_2}{\|\boldsymbol{z}\|_2}\,, \tag{16}$$

where $\hat{\boldsymbol{z}}$ and $\boldsymbol{z}$ denote the predicted and ground-truth solution vectors obtained by concatenating the outputs of all physical processes following prior work (Sun et al., 2025; Xiao et al., 2023). We consider two training-data regimes with 512 and 1024 samples to assess performance under different levels of data availability. All experiments are repeated over five independent runs, and we report the mean prediction error, with standard deviations in Appendix C. We also evaluate long-horizon rollout performance, robustness to distribution shifts, and computational efficiency. Experimental configurations and hyperparameters are provided in Appendix D.

4.2 Main Results

Predictive Performance. The results in Table 1 and Table 13 show the predictive performance of Coupled-Flow against baseline neural operators (COMPOL, FNO, UFNO, Transolver, CMWNO) and recent approaches based on vision transformers, CViT (Wang et al., 2025) and diffusion models, DiffusionPDE (Huang et al., 2024). We evaluate all methods with 512 and 1024 training samples across five datasets (Lotka- Volterra, Belousov-Zhabotinsky, Gray- Scott, multiphase flow, and Thermal-Hydro-Mechanical). We make the following observations.

1. Across both training-data regimes, CoupledFlow with the GP interpolant (coupled-flow-GP) achieves the best predictive performance on 7 out of 10 tasks and the second-best performance on the remaining 3 tasks.

2. CoupledFlow with the linear interpolant (coupled-flow-Linear) also performs competitively, achieving second-best performance on 7 out of 10 tasks. The improvements are particularly pronounced on benchmarks with strong cross-process interactions.

To evaluate the generality of CoupledFlow beyond the coupled PDE benchmarks considered in prior coupled-operator work, we conduct additional experiments on two standard benchmarks: Reaction-Diffusion-2D from PDEBench (Takamoto et al., 2024) and Navier-Stokes-2D from PDEArena (Gupta and Brandstetter, 2022). Table 3 reports the relative L_2 prediction errors. We include CViT-B (Wang et al., 2025) as a strong baseline, given its recent state-of-the-art performance across several operator-learning benchmarks.

3) On Reaction-Diffusion-2D, CoupledFlow-GP achieves performance close to CViT-B, the strongest baseline, while substantially outperforming the other neural-operator and coupled-operator baselines. On Navier-Stokes-2D, CoupledFlow-GP achieves the best predictive performance.

Method	Lotka-V.	Belousov-Z.	Gray-Scott	Multiphase	THM
FNO_c	9.55	0.16	0.55	2.91	0.11
CMWNO	6.65	0.14	0.43	1.93	0.09
UFNO_c	8.73	0.17	0.36	3.87	0.08
Transolver_c	17.67	1.88	0.49	0.88	0.16
DiffusionPDE_c	3.10	0.53	0.21	2.45	0.16
MeanFlow_c	13.21	0.17	0.29	3.33	0.13
CViT-B	**0.77**	0.31	0.74	1.39	0.51
COMPOL-RNN	8.02	0.15	0.50	6.44	0.09
COMPOL-ATN	8.58	0.15	0.48	6.76	0.10
COUPLEDFLOW-Linear	1.29	**0.09**	**0.18**	**0.09**	**0.08**
COUPLEDFLOW-GP	**1.21**	**0.06**	**0.15**	**0.11**	**0.05**

Table 1: Comparison of relative L_2 prediction errors ($\times 10^{-2}$) across five multi-physics benchmark datasets with 1024 training samples. **Bold** and **blue** values indicate the best and second-best performance, respectively. Detailed results with standard deviations are provided in Table 13.

4.3 Ablation Studies

Rollout Prediction. We further evaluate whether the learned one-step operator remains stable when recursively applied for long-horizon prediction. Tables 15 and 14 report rollout errors on Lotka–Volterra and Gray–Scott using 512 training samples and 256 independent test samples, over 10 rollout steps. COUPLEDFLOW consistently achieves the best or second-best performance across all rollout horizons, indicating that the proposed transport formulation not only improves single-step prediction but also supports stable multi-step forecasting.

Component-wise Contribution. Table 2 evaluates the contribution of the coupled transport formulation relative to the vanilla MeanFlow baseline. **(1)** Both COUPLEDFLOW-GP and COUPLEDFLOW-Linear substantially outperform MeanFlow_c across all data regimes, demonstrating that conditioning process-specific average velocities on the full coupled state is important for multi-physics prediction. **(2)** The GP variant further improves over the linear variant. This suggests that GP-based coupled paths provide more informative intermediate supervision than independent linear interpolation.

Method	Lotka-V.		Belousov-Z.	
	512	1024	512	1024
MeanFlow_c	16.52	13.21	0.17	0.17
COUPLEDFLOW-GP	**2.16**	**1.21**	**0.11**	**0.06**
COUPLEDFLOW-Linear	**2.34**	**1.29**	**0.15**	**0.09**

Table 2: Comparison of relative L_2 errors ($\times 10^{-2}$) on Lotka–Volterra and Belousov–Zhabotinsky. Detailed results with standard deviations are provided in Table 13 of Appendix C.

Method	RD-2D	NS-2D
MeanFlow_c	9.51	16.13
FNO_c	12.34	9.12
UFNO_c	12.33	9.47
COMPOL-RNN	8.28	8.25
COMPOL-ATN	8.27	8.21
CMWNO	8.45	8.42
DiffusionPDE	12.65	8.84
CViT-B	**1.58**	**3.95**
COUPLEDFLOW-GP	**1.63**	**3.45**
COUPLEDFLOW-Linear	1.74	3.98

Table 3: Relative L_2 prediction error ($\times 10^{-2}$) on Reaction-Diffusion-2D (RD-2D) and Navier-Stokes-2D (NS-2D). Detailed results with standard deviations are provided in Appendix C.

Sensitivity and Robustness Analysis. We examine the sensitivity of COUPLEDFLOW-GP to the GP kernel and length-scale. **(1)** As shown in Table 11, performance remains stable across a broad range of ℓ, indicating that the method does not require precise length-scale tuning. **(2)** Table 12 shows that several standard kernels, including squared-exponential, Matérn-$3/2$, Matérn-$5/2$, and rational quadratic, perform close to or better than the linear interpolation baseline. **(3)** Since different kernels perform best on different datasets, the gains appear to come from correlated GP-based intermediate paths rather than reliance on a specific covariance function. **(4)** Additional OOD coefficient-shift results are provided in Table 17 of Appendix C. COUPLEDFLOW-GP consistently achieves the best or second-best generalization performance, highlighting the robustness

Method	Lotka-V.	Belousov-Z.	Gray-Scott	Multiphase
FNO_c	0.70	10.15	0.20	2.04
$UFNO_c$	0.65	10.04	0.25	5.27
$CMWNO_c$	0.62	7.11	0.22	2.36
COMPOL-ATN_c	0.48	8.74	0.18	2.67
COMPOL-RNN_c	0.52	8.52	0.19	1.87
CViT	0.42	3.70	1.13	3.26
CoupledFlow-Linear	0.23	2.30	0.15	0.90
CoupledFlow-GP	**0.19**	**2.10**	**0.11**	**0.61**

Table 4: Comparison of relative L_2 prediction errors ($\times 10^{-2}$) with approximately matched parameter counts. The parameter budgets are 10M, 9M, 13M, and 10M for Lotka–Volterra, Belousov–Zhabotinsky, Gray–Scott, and Multiphase, respectively. The best results are highlighted in bold.

of our method. Table 8 also reports experiments under varying initial and boundary conditions on MPF with matched model capacities, where CoupledFlow-GP outperforms DiffusionPDE in both ID and OOD settings. **(5)** Results for boundary and initial-condition prediction errors are provided in Table 9 of Appendix C. CoupledFlow achieves the lowest overall and boundary errors on both MPF and THM benchmarks, further demonstrating its effectiveness in capturing coupled PDE solution behavior.

Model Scale and Computational Efficiency. To investigate whether the performance gains of CoupledFlow are attributable to larger model capacity, we conduct experiments with approximately matched parameter counts across methods. Specifically, we consider parameter budgets of 10M, 9M, 13M, and 10M for Lotka–Volterra, Belousov–Zhabotinsky, Gray–Scott, and Multiphase, respectively. As shown in Table 4, CoupledFlow-GP consistently achieves the lowest prediction error across all four benchmarks, reducing the error by 38.9%–67.4% relative to the strongest competing baseline on each dataset. In terms of computational efficiency, our method introduces a small inference-time overhead compared to FNO and CViT, averaging approximately 3.8 ms and 3.7 ms, respectively, across the five benchmarks reported in Table 6. Importantly, the GP-based interpolations are used only during training, while inference remains a one-step operation. Overall, the substantial improvements in prediction accuracy come at a modest additional inference cost, indicating a favorable accuracy–efficiency trade-off.

Runtime/Speed Results. Due to space constraints, we report these results in Table 6 of Appendix. CoupledFlow achieves substantially faster inference than iterative generative baselines such as DiffusionPDE and is consistently faster than CMWNO. Its inference runtime is also competitive with standard neural operators and transformer-based methods.

5 Conclusion

We introduced CoupledFlow, a one-step flow-based neural operator for coupled multi-physics PDE systems. Unlike standard endpoint-regression neural operators, our method learns transport dynamics through process-specific average velocity fields conditioned on the full coupled state, capturing cross-process interactions during training. To provide informative path-level supervision, we constructed Gaussian process-based correlated interpolants that capture cross-process dependencies more effectively than independent linear interpolants. More broadly, our results show that intermediate transport paths can guide the learning of cross-physics dependencies without requiring their explicit simulation at inference time. Our theoretical analysis established that the prediction error of the learned operator can be controlled by the coupled mean-flow matching objective and the smoothness of the induced transport. Empirically, CoupledFlow outperformed state-of-the-art baselines across diverse coupled PDE benchmarks.

Acknowledgement

This research is supported by the National Science Foundation CAREER Award IIS-2544071. Phi Le Nguyen is supported by the Vietnam National Foundation for Science and Technology Development (NAFOSTED) under project code 24/2025/HĐ-TN.01.

References

Jamal Hussein Abou-Kassem, Syed Mohammad Farouq-Ali, and M Rafiq Islam. *Petroleum Reservoir Simulations*. Elsevier, 2013. ISBN 978-0-128-00414-2.

Mauricio A. Alvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for vector-valued functions: a review, 2012. URL https://arxiv.org/abs/1106.6251.

Giacomo Baldan, Qiang Liu, Alberto Guardone, and Nils Thuerey. Flow matching meets pdes: A unified framework for physics-constrained generation. *arXiv preprint arXiv:2506.08604*, 2025.

Jan-Hendrik Bastek, WaiChing Sun, and Dennis M Kochmann. Physics-informed diffusion models. *arXiv preprint arXiv:2403.14404*, 2024.

Jacob Bear and Alexander H-D Cheng. *Modeling Groundwater Flow and Contaminant Transport*, volume 23. Springer, 2010.

Chaoran Cheng, Boran Han, Danielle C Maddix, Abdul Fatir Ansari, Andrew Stuart, Michael W Mahoney, and Yuyang Wang. Gradient-free generation for hard-constrained systems. *arXiv preprint arXiv:2412.01786*, 2024.

Qian Gao and Ahmad Ghassemi. Three-dimensional thermo-poroelastic modeling and analysis of flow, heat transport and deformation in fractured rock with applications to a lab-scale geothermal system. *Rock Mechanics and Rock Engineering*, 53(4):1565–1586, 2020. doi: 10.1007/s00603-019-01989-0.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=uWj4s7rMnR.

Wolfgang Gröbner. *Die Lie-Reihen und ihre Anwendungen*, volume 3 of *Mathematische Monographien*. VEB Deutscher Verlag der Wissenschaften, Berlin, 1960.

Jayesh K Gupta and Johannes Brandstetter. Towards multi-spatiotemporal-scale generalized pde modeling. *arXiv preprint arXiv:2209.15616*, 2022.

Leila Hashemi, Martin Blunt, and Hadi Hajibeygi. Pore-scale modelling and sensitivity analyses of hydrogen-brine multiphase flow in geological porous media. *Scientific Reports*, 11(1):8348, 2021.

Quang Minh Hoang, Trong Nghia Hoang, and Kian Hsiang Low. A generalized stochastic variational Bayesian hyperparameter learning framework for sparse spectrum Gaussian process regression. In *AAAI Conference on Artificial Intelligence*, pages 2007–2014, 2017.

Quang Minh Hoang, Trong Nghia Hoang, Hai Pham, and David Woodruff. Revisiting the sample complexity of sparse spectrum approximation of gaussian processes. In *Advances in Neural Information Processing Systems*, 2020.

Trong Nghia Hoang, Quang Minh Hoang, and Kian Hsiang Low. A unifying framework of anytime sparse Gaussian process regression models with stochastic variational inference for big data. In *International Conference on Machine Learning*, pages 569–578, 2015.

Trong Nghia Hoang, Quang Minh Hoang, and Kian Hsiang Low. A distributed variational inference framework for unifying parallel sparse Gaussian process regression models. In *International Conference on Machine Learning*, pages 382–391, 2016.

Trong Nghia Hoang, Quang Minh Hoang, Kian Hsiang Low, and Jonathan P. How. Collective online learning of Gaussian processes in massive multi-agent systems. In *Proceedings of theAAAI Conference on Artificial Intelligence*, pages 7850–7857, 2019.

Jiahe Huang, Guandao Yang, Zichen Wang, and Jeong Joon Park. Diffusionpde: Generative pde-solving under partial observation. *Advances in Neural Information Processing Systems*, 37: 130291–130323, 2024.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

Nikita Kornilov, Petr Mokrov, Alexander Gasnikov, and Alexander Korotin. Optimal flow matching: Learning straight trajectories in just one step. *Advances in Neural Information Processing Systems*, 37:104180–104204, 2024.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Multipole graph neural operator for parametric partial differential equations, 2020a. URL https://arxiv.org/abs/2006.09535.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations, 2020b. URL https://arxiv.org/abs/2003.03485.

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

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

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.

James D. Murray. *Mathematical Biology I. An Introduction*, volume 17 of *Interdisciplinary Applied Mathematics*. Springer, New York, 3 edition, 2002. doi: 10.1007/b98868.

John E. Pearson. Complex patterns in a simple system. *Science*, 261(5118):189–192, July 1993. ISSN 1095-9203. doi: 10.1126/science.261.5118.189. URL http://dx.doi.org/10.1126/science.261.5118.189.

Valery Petrov, Vilmos Gáspár, Jonathan Masere, and Kenneth Showalter. Controlling chaos in the belousov—zhabotinsky reaction. *Nature*, 361(6409):240–243, 1993. doi: 10.1038/361240a0. URL https://doi.org/10.1038/361240a0.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Salva Rühling Cachay, Bo Zhao, Hailey Joren, and Rose Yu. Dyffusion: A dynamics-informed diffusion model for spatiotemporal forecasting. *Advances in neural information processing systems*, 36:45259–45287, 2023.

Kevin Scaman and Aladin Virmaux. Lipschitz regularity of deep neural networks: analysis and efficient estimation, 2019. URL https://arxiv.org/abs/1805.10965.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. 2023.

Yifei Sun, Tao Wang, Junqi Qu, Yushun Dong, Hewei Tang, and Shibo Li. Compol: A unified neural operator framework for scalable multi-physics simulations, 2025. URL https://arxiv.org/abs/2501.17296.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Dan MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. Pdebench: An extensive benchmark for scientific machine learning, 2024. URL https://arxiv.org/abs/2210.07182.

Alan Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society B*, 237:37–72, 1952.

Utkarsh Utkarsh, Pengfei Cai, Alan Edelman, Rafael Gomez-Bombarelli, and Christopher Vincent Rackauckas. Physics-constrained flow matching: Sampling generative models with hard constraints. 2025.

Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

Sifan Wang, Jacob H Seidman, Shyam Sankaran, Hanwen Wang, George J. Pappas, and Paris Perdikaris. Cvit: Continuous vision transformer for operator learning, 2025. URL https://arxiv.org/abs/2405.13998.

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, 2022. URL https://arxiv.org/abs/2109.03697.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for pdes on general geometries, 2024. URL https://arxiv.org/abs/2402.02366.

Xiongye Xiao, Defu Cao, Ruochen Yang, Gaurav Gupta, Gengshuo Liu, Chenzhong Yin, Radu Balan, and Paul Bogdan. Coupled multiwavelet operator learning for coupled differential equations. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=kIo_C6QmMOM.

A Proof of Theorem 1

A.1 Endpoint Consistency of GP interpolation

Proposition 3. *Let $\boldsymbol{x}(t \mid \boldsymbol{x}, \boldsymbol{x}')$ be a multi-output Gaussian process conditioned on the endpoint observations $\boldsymbol{x}(0) = \boldsymbol{x}$ and $\boldsymbol{x}(1) = \boldsymbol{x}'$. Then the GP posterior mean satisfies*

$$\boldsymbol{m}_{i|c}(0) = \boldsymbol{x}_i, \qquad \boldsymbol{m}_{i|c}(1) = \boldsymbol{x}'_i,$$

for every physical process i, where $\boldsymbol{m}_{i|c}(t)$ denotes the posterior mean of process i conditioned on process c.

Proof. We consider a zero-mean multi-output Gaussian process with covariance kernel $\boldsymbol{K}_{ic}(t, s \mid \boldsymbol{x}, \boldsymbol{x}')$. Let the endpoint observations be

$$\boldsymbol{y}_i = \begin{bmatrix} \boldsymbol{x}_i \\ \boldsymbol{x}'_i \end{bmatrix}, \qquad \mathcal{T} = \{0, 1\}.$$

Conditioning the GP on these observations yields the posterior mean

$$\boldsymbol{m}_{i|c}(t) = \boldsymbol{K}_{ic}(t, \mathcal{T} \mid \boldsymbol{x}, \boldsymbol{x}')\boldsymbol{K}_{cc}(\mathcal{T}, \mathcal{T} \mid \boldsymbol{x}, \boldsymbol{x}')^{-1}\boldsymbol{y}_i, \tag{17}$$

and posterior covariance

$$\boldsymbol{K}_{i|c}(t) = \boldsymbol{K}_{ic}(t \mid \boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{K}_{ic}(t, \mathcal{T} \mid \boldsymbol{x}, \boldsymbol{x}')\boldsymbol{K}_{cc}(\mathcal{T}, \mathcal{T} \mid \boldsymbol{x}, \boldsymbol{x}')^{-1}\boldsymbol{K}_{ci}(\mathcal{T}, s \mid \boldsymbol{x}, \boldsymbol{x}'). \tag{18}$$

At $t = 0$, since $\boldsymbol{K}_{ic}(0, \mathcal{T} \mid \boldsymbol{x}, \boldsymbol{x}')$ corresponds to the first block row of $\boldsymbol{K}_{cc}(\mathcal{T}, \mathcal{T} \mid \boldsymbol{x}, \boldsymbol{x}')$, we obtain

$$\boldsymbol{m}_{i|c}(0) = \boldsymbol{K}_{ic}(0, \mathcal{T} \mid \boldsymbol{x}, \boldsymbol{x}')\boldsymbol{K}_{cc}(\mathcal{T}, \mathcal{T} \mid \boldsymbol{x}, \boldsymbol{x}')^{-1}\boldsymbol{y}_i = \boldsymbol{x}_i \tag{19}$$

Similarly, at $t = 1$,

$$\boldsymbol{m}_{i|c}(1) = \boldsymbol{K}_{ic}(1, \mathcal{T} \mid \boldsymbol{x}, \boldsymbol{x}')\boldsymbol{K}_{cc}(\mathcal{T}, \mathcal{T} \mid \boldsymbol{x}, \boldsymbol{x}')^{-1}\boldsymbol{y}_i = \boldsymbol{x}'_i \tag{20}$$

Furthermore, the posterior covariance vanishes at the observed endpoints:

$$\boldsymbol{K}_{i|c}(0) = \boldsymbol{K}_{i|c}(1) = 0 \tag{21}$$

Therefore, conditioning on an additional physical process c cannot alter the deterministic endpoint values of process i. This establishes that the GP-based correlated interpolants preserve both prescribed endpoints. □

A.2 Proof of Theorem 1

Theorem. *Theoretic optimizers of the conditional mean flow matching losses in Eq.* (8) *and Eq.* (13) *are the same.*

Proof. We can express the objective function in Eq. (8) as

$$\begin{aligned}
L(\theta) &= \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{(s,t)}\mathbb{E}_{(\boldsymbol{x},\boldsymbol{x}')}\mathbb{E}_{\boldsymbol{x}_i(t)}\Big\|\boldsymbol{u}^i_\theta\big(\mathbb{E}[\boldsymbol{x}(t)], t\big) - d\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}')/dt\Big\|^2 \qquad (22)\\
&= \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{(s,t)}\mathbb{E}_{(\boldsymbol{x},\boldsymbol{x}')}\mathbb{E}_{\boldsymbol{x}_i(t)}\Big\|\boldsymbol{u}^i_\theta\big(\mathbb{E}[\boldsymbol{x}(t)], t\big) - d\mathbb{E}[\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}')]/dt\Big\|^2\\
&+ \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{(s,t)}\mathbb{E}_{(\boldsymbol{x},\boldsymbol{x}')}\mathbb{E}_{\boldsymbol{x}_i(t)}\Big\|d\mathbb{E}[\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}')]/dt - d\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}')/dt\Big\|^2\\
&+ \frac{2}{m}\sum_{i=1}^{m} \mathbb{E}_{(s,t)}\mathbb{E}_{(\boldsymbol{x},\boldsymbol{x}')}\mathbb{E}_{\boldsymbol{x}_i(t)}\langle \boldsymbol{u}^i_\theta\big(\mathbb{E}[\boldsymbol{x}(t)], t\big) - d\mathbb{E}[\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}')]/dt, d\mathbb{E}[\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}')]/dt - d\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}')/dt\rangle
\end{aligned}$$

Because $\mathbb{E}_{\boldsymbol{x}_i(t)}[d\mathbb{E}[\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}')]/dt - d\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}')/dt] = 0$, we have

$$\begin{aligned} L(\theta) &= \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{(s,t)}\mathbb{E}_{(\boldsymbol{x},\boldsymbol{x}')}\mathbb{E}_{\boldsymbol{x}_i(t)}\left\|\boldsymbol{u}_\theta^i\big(\mathbb{E}[\boldsymbol{x}(t)], t\big) - d\mathbb{E}[\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}')]/dt\right\|^2 \\ &+ \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{(s,t)}\mathbb{E}_{(\boldsymbol{x},\boldsymbol{x}')}\mathbb{E}_{\boldsymbol{x}_i(t)}\left\|d\mathbb{E}[\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}')]/dt - d\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}')/dt\right\|^2 \\ &= \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_{(s,t)}\mathbb{E}_{(\boldsymbol{x},\boldsymbol{x}')}\mathbb{E}_{\boldsymbol{x}_i(t)}\left\|\boldsymbol{u}_\theta^i\big(\mathbb{E}[\boldsymbol{x}(t)], t\big) - d\mathbb{E}[\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}')/dt]\right\|^2 + \text{const} \end{aligned} \tag{23}$$

Here, the second-to-last equality follows from the interchangeability between the expectation $\mathbb{E}$ and the derivative d/dt of $\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}')$ when $\boldsymbol{x}_i(t \mid \boldsymbol{x}, \boldsymbol{x}')$ follows the law of GPs with a squared exponential kernel, which we used in our experiments. This shows that Eq. (8) and (13) only differ by a constant. □

B Proof of the Theorem 2

B.1 Key lemmas

We need the following lemmas to prove Theorem 2.

Lemma 4. *Let ℓ_r denote the Lipschitz constant of $\boldsymbol{u}_\theta(\boldsymbol{x}(r), r)$ with respect to its spatial argument, and define*

$$\kappa \triangleq \exp\left(\int_0^\tau \ell_r \, dr\right). \tag{24}$$

Then the operator norm of the Jacobian of the state $\boldsymbol{x}(\tau)$ with respect to the initial condition $\boldsymbol{x}(0)$ satisfies

$$\|\nabla_{\boldsymbol{x}(0)}\boldsymbol{x}(\tau)\|_{\text{op}} \leq \kappa. \tag{25}$$

Proof. Let $\boldsymbol{x}(r)$ be the solution to the ODE

$$\frac{d}{dr}\boldsymbol{x}(r) = \boldsymbol{u}_\theta(\boldsymbol{x}(r), r), \qquad \boldsymbol{x}(0) = \boldsymbol{x}. \tag{26}$$

Define the Jacobian matrix

$$J(r) \triangleq \nabla_{\boldsymbol{x}(0)}\boldsymbol{x}(r). \tag{27}$$

Differentiating the dynamics with respect to the initial condition $\boldsymbol{x}(0)$ yields the variational equation

$$\frac{d}{dr}J(r) = \nabla_{\boldsymbol{x}}\boldsymbol{u}_\theta(\boldsymbol{x}(r), r)\, J(r), \qquad J(0) = I. \tag{28}$$

Taking operator norms and using submultiplicativity,

$$\frac{d}{dr}\|J(r)\|_{\text{op}} \leq \|\nabla_{\boldsymbol{x}}\boldsymbol{u}_\theta(\boldsymbol{x}(r), r)\|_{\text{op}}\, \|J(r)\|_{\text{op}}. \tag{29}$$

Since $\boldsymbol{u}_\theta(\cdot, r)$ is ℓ_r-Lipschitz,

$$\|\nabla_{\boldsymbol{x}}\boldsymbol{u}_\theta(\boldsymbol{x}(r), r)\|_{\text{op}} \leq \ell_r. \tag{30}$$

Therefore,

$$\frac{d}{dr}\|J(r)\|_{\text{op}} \leq \ell_r \|J(r)\|_{\text{op}}. \tag{31}$$

Applying Grönwall's inequality gives

$$\|J(\tau)\|_{\text{op}} \quad \leq \quad \exp\left(\int_0^\tau \ell_r \, dr\right) = \kappa. \tag{32}$$

Hence,

$$\|\nabla_{\boldsymbol{x}(0)} \boldsymbol{x}(\tau)\|_{\text{op}} \quad \leq \quad \kappa. \tag{33}$$

□

Lemma 5. *Let $L(\theta)$ be defined as in Eq.* (15)*. Let $\boldsymbol{x}^\star(r)$ denote the oracle trajectory satisfying*

$$\frac{d}{dr}\boldsymbol{x}^\star(r) \quad = \quad \boldsymbol{u}(\boldsymbol{x}^\star(r), r), \qquad \boldsymbol{x}^\star(0) = \boldsymbol{x}. \tag{34}$$

For any $\boldsymbol{x} \sim \pi_0(\boldsymbol{x})$, we have the one-step prediction $\hat{\boldsymbol{x}}(\tau) = \boldsymbol{x} + \tau\, \bar{\boldsymbol{u}}_\theta(\boldsymbol{x}, 0, \tau)$ and the true target sample $\boldsymbol{x}'(\tau) = \boldsymbol{x} + \int_0^\tau \boldsymbol{u}(\boldsymbol{x}^\star(r), r)dr$, then

$$\mathbb{E}_{\boldsymbol{x}\sim\pi_0(\boldsymbol{x})}\|\hat{\boldsymbol{x}}(\tau) - \boldsymbol{x}'(\tau)\|^2 \quad \leq \quad \kappa^2 m \tau^2 L(\theta) \tag{35}$$

Proof. Define the trajectory error

$$\boldsymbol{e}(r) \quad \triangleq \quad \hat{\boldsymbol{x}}(r) - \boldsymbol{x}^\star(r). \tag{36}$$

Note that the predicted trajectory satisfies

$$\frac{d}{dr}\hat{\boldsymbol{x}}(r) \quad = \quad \hat{u}_\theta(\hat{x}(r), r), \qquad \hat{\boldsymbol{x}}(0) = \boldsymbol{x}. \tag{37}$$

Substracting the two dynamics gives

$$\frac{d}{dr}\boldsymbol{e}(r) \quad = \quad \hat{u}_\theta(\hat{\boldsymbol{x}}(r), r) - \boldsymbol{u}(\boldsymbol{x}^\star(r), r). \tag{38}$$

Applying Alekseev-Gröbner identity (Gröbner, 1960), we obtain

$$\boldsymbol{e}(\tau) \quad = \quad \int_0^\tau \nabla_{\boldsymbol{x}(0)}\boldsymbol{x}(\tau)\Big(\hat{u}_\theta(\hat{\boldsymbol{x}}(r), r) - \boldsymbol{u}(\boldsymbol{x}^\star(r), r)\Big)\, dr, \tag{39}$$

Applying Lemma 4, we have

$$\|\nabla_{\boldsymbol{x}(0)}\boldsymbol{x}(\tau)\|_{\text{op}} \quad \leq \quad \kappa. \tag{40}$$

Therefore,

$$\|\hat{\boldsymbol{x}}(\tau) - \boldsymbol{x}'(\tau)\| \quad = \quad \|\boldsymbol{e}(\tau)\| \;\leq\; \kappa \int_0^\tau \|\hat{u}_\theta(\hat{\boldsymbol{x}}(r), r) - \boldsymbol{u}(\boldsymbol{x}^\star(r), r)\|\, dr \tag{41}$$

Applying Cauchy-Schwarz inequality gives

$$\|\boldsymbol{e}(\tau)\|^2 \quad \leq \quad \kappa^2 \tau \int_0^\tau \|\hat{u}_\theta(\hat{\boldsymbol{x}}(r), r) - \boldsymbol{u}(\boldsymbol{x}^\star(r), r)\|^2\, dr \tag{42}$$

Taking expectation with respect to $\boldsymbol{x} \sim \pi_0(\boldsymbol{x})$, we obtain

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{x}\sim\pi_0(\boldsymbol{x})}\|\hat{\boldsymbol{x}}(\tau) - \boldsymbol{x}'(\tau)\|^2 \quad &\leq \quad \kappa^2\tau \int_0^\tau \mathbb{E}_{\boldsymbol{x}\sim\pi_0(\boldsymbol{x})}\Big\|\boldsymbol{u}_\theta(\boldsymbol{x}(r), r) - \frac{d}{dt}\boldsymbol{x}^\star(r)\Big\|_2^2 dr \\
&= \quad \kappa^2\tau^2 \sum_{i=1}^m \mathbb{E}_{t\sim\mathcal{U}(0,\tau)}\mathbb{E}_{(\boldsymbol{x},\boldsymbol{x}')}\Big\|\boldsymbol{u}_\theta^i(\boldsymbol{x}(t), t) - \frac{d}{dt}\boldsymbol{x}_i^\star(t)\Big\|_2^2
\end{aligned} \tag{43}$$

We now need to bound the above sum in terms of $L(\theta)$. For any fixed $t \in [0, \tau]$, the pair (s, t) with $s \in [0, t]$ is a subset of all pairs sampled in $L(\theta)$. Therefore, for a process i and fixed t:

$$\mathbb{E}_{s\sim\mathcal{U}(0,t)}\mathbb{E}_{(\boldsymbol{x},\boldsymbol{x}')}\Big\|\boldsymbol{u}_\theta^i(\boldsymbol{x}(t), t) - \frac{d}{dt}\boldsymbol{x}_i^\star(t)\Big\|_2^2 \quad \leq \quad \mathbb{E}_{\substack{s\sim\mathcal{U}(0,\tau)\\ t'\sim\mathcal{U}(s,\tau)}}\mathbb{E}_{(\boldsymbol{x},\boldsymbol{x}')}\Big\|\boldsymbol{u}_\theta^i(\boldsymbol{x}(t'), t') - \frac{d}{dt}\boldsymbol{x}_i^\star(t')\Big\|_2^2$$

Integrating over t from 0 to τ and summing over processes $i = 1, \ldots, m$ yields

$$\int_0^\tau \mathbb{E}_{\boldsymbol{x}\sim\pi_0(\boldsymbol{x})}\Big\|\boldsymbol{u}_\theta(\boldsymbol{x}(t), t) - \frac{d}{dt}\boldsymbol{x}^\star(t)\Big\|_2^2 dt \quad \leq \quad m\tau\, L(\theta). \tag{44}$$

Substituting this into (43) implies

$$\mathbb{E}_{\boldsymbol{x}\sim\pi_0(\boldsymbol{x})}\|\hat{\boldsymbol{x}}(\tau) - \boldsymbol{x}'(\tau)\|^2 \quad \leq \quad \kappa^2 m\tau^2 L(\theta) \tag{45}$$

□

Model	Grey-Scot	Multiphase	Lotka-V.	Belousov-Z	THM
Transolver	0.42	0.42	0.42	0.42	0.42
FNO	4.47	4.47	0.58	0.58	8.41
UFNO	6.73	6.73	1.31	1.31	10.04
CMWNO	2.0	2.0	1.0	1.5	5.0
COMPOL-RNN	2.42	2.42	2.50	3.07	10.62
COMPOL-ATN	2.40	2.40	1.49	2.10	10.57
DiffusionPDE	15.5	15.5	10.9	10.9	15.5
CViT-B	23.5	23.5	21.5	21.5	23.6
Ours	10.4	10.4	8.9	13.4	25.5

Table 5: Number of model parameters (M) of baselines across different benchmark datasets.

Model	Grey-Scot	Multiphase	Belousov-Z	Lotka-V.	THM
FNO	0.278	0.308	0.132	0.142	0.271
UFNO	0.836	0.839	0.492	0.497	0.835
COMPOL-RNN	0.910	0.880	0.608	0.502	1.818
COMPOL-ATN	1.226	1.190	0.661	0.468	2.725
CViT-B	0.539	0.214	0.149	0.208	0.210
CMWNO	11.471	9.449	7.912	11.284	19.870
DiffusionPDE	58.013	93.442	78.458	66.134	60.065
Ours	3.117	2.926	5.328	3.601	4.943

Table 6: The wall clock time (ms) of baselines across different benchmark datasets.

B.2 Proof of Theorem 2

We now ready to prove Theorem 2

Proof of Theorem 2. By properties of the Wasserstein-2 distance on product measures (Villani et al., 2009), we have

$$\sum_{i=1}^{m} W_2^2(\pi_\tau^i, \bar{\pi}_\tau^i) \quad \leq \quad W_2^2(\pi_\tau, \bar{\pi}_\tau), \tag{46}$$

where π_τ and $\bar{\pi}_\tau$ denote the joint distributions over the full multi-physics state.

Since W_2^2 is the infimum over all couplings of the expected squared Euclidean cost, for the particular coupling induced by the one-step predictor we obtain

$$W_2^2(\pi_\tau, \bar{\pi}_\tau) \quad \leq \quad \mathbb{E}_{\boldsymbol{x}\sim\pi_0(\boldsymbol{x})}\left[\|\hat{\boldsymbol{x}}(\tau) - \boldsymbol{x}'(\tau)\|_2^2\right]. \tag{47}$$

From Lemma 5, we have

$$\mathbb{E}_{\boldsymbol{x}\sim\pi_0}\left[\|\hat{\boldsymbol{x}}(\tau) - \boldsymbol{x}'(\tau)\|_2^2\right] \quad \leq \quad \kappa^2 m \tau^2 L(\theta). \tag{48}$$

Combining (46), (47), and (48) yields

$$\frac{1}{m}\sum_{i=1}^{m} W_2^2(\pi_\tau^i, \bar{\pi}_\tau^i) \quad \leq \quad \kappa^2 \tau^2 L(\theta). \tag{49}$$

Using Jensen's inequality gives

$$\textsc{err}(\theta) \quad \leq \quad \sqrt{\frac{1}{m}\sum_{i=1}^{m} W_2^2(\pi_\tau^i, \bar{\pi}_\tau^i)} \;\leq\; \kappa\tau\sqrt{L(\theta)}. \tag{50}$$

This concludes the proof. □

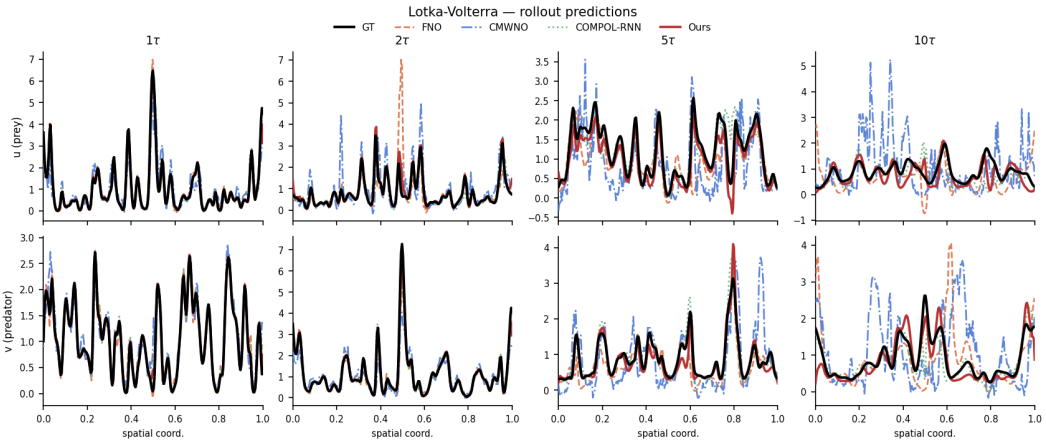

Figure 2: The prediction of COUPLED-FLOW-GP (Ours) and other baselines vs. Ground Truth (GT) on Lotka–Volterra across different rollout prediction steps.

Hyperparameter	Grey-Scott	Lotka-V.	Multiphase	Belousov-Z	THM
Architecture	Unet	Unet	Unet	Unet	Unet
# nets	2	2	2	3	5
Params/net	∼5.1M	∼4.5M	∼5.1M	∼4.5M	∼5.1M
Total params	∼10.2M	∼9M	∼10.2M	∼13.5M	∼25.5M
Batch size	32	32	32	32	32
Learning rate	6×10^{-4}	6×10^{-4}	6×10^{-4}	6×10^{-4}	6×10^{-4}
Epochs	500	500	500	500	500
Warmup epochs	100	100	100	100	100
LR schedule	linear warmup	linear warmup	linear warmup	linear warmup	linear warmup
Optimizer	Adam	Adam	Adam	Adam	Adam
Dropout	0.2	0.2	0.2	0.2	0.2
EMA decay	0.9999	0.9999	0.9999	0.9999	0.9999
GP kernel	SE	SE	SE	SE	SE
GP length scale ℓ_0	1	1	1	1	1
Train ratio	0.8	0.8	0.8	0.8	0.8
Val ratio	0.2	0.2	0.2	0.2	0.2

Table 7: Hyperparameter settings across benchmark datasets. The Adam (Kingma and Ba, 2017) optimizer is referred to as Adam with $(\beta_1 = 0.9, \beta_2 = 0.999)$. SE is referred to as the Squared-Exponential kernel.

C Detailed Experiment Results and Ablation Studies

In this section, we provide the detailed experiment results and additional ablation studies that support the empirical analysis in Section 4. Unless otherwise stated, all reported numbers are relative L_2 prediction errors averaged over five independent runs, and the uncertainty term denotes the standard deviation across runs. The main results in Section 4 are reported in the scaled form $(\times 10^{-2})$, while the tables in this appendix report the corresponding raw relative L_2 values.

Detailed single-step prediction results. Table 13 reports the complete single-step prediction results on the five coupled multi-physics benchmarks, including Lotka–Volterra, Belousov–Zhabotinsky, Gray–Scott, Multiphase, and Thermal-Hydro-Mechanical systems. The results provide the mean and standard deviation corresponding to Table 1. Across both training-data regimes, COUPLED-FLOW-GP consistently achieves the best or second-best performance. In particular, COUPLED-FLOW-GP obtains the strongest performance on most settings, while COUPLED-FLOW-Linear also remains highly competitive. This confirms that the improvement does not only come from a larger neural architecture, but from the coupled transport formulation itself.

Method	MPF-ID	MPF-OOD
DiffusionPDE	0.0241	0.0318
CoupledFlow-Linear	0.0090	0.0240
CoupledFlow-GP	**0.0061**	**0.0190**

Table 8: Comparison of relative L_2 errors under in-distribution (ID) and out-of-distribution (OOD) settings on Multiphase Flow (MPF), using approximately matched model capacity (10M parameters).

Results on standard PDE benchmarks. Table 16 provides the complete results with standard deviations on Reaction-Diffusion-2D and Navier-Stokes-2D. These benchmarks complement the coupled PDE datasets used in prior coupled-operator work and evaluate whether the proposed formulation remains effective on standardized operator-learning benchmarks. On Reaction-Diffusion-2D, COUPLED-FLOW-GP achieves performance close to the strongest CViT-B baseline while substantially improving over FNO, UFNO, CMWNO, COMPOL, DiffusionPDE, and MeanFlow. On Navier-Stokes-2D, COUPLED-FLOW-GP achieves the best performance among all compared methods. COUPLED-FLOW-Linear is also competitive with CViT-B, suggesting that the one-step coupled average-velocity parameterization is already effective even without the GP path, while the GP path provides additional gains.

Rollout prediction. Tables 14 and 15 report recursive rollout prediction on Lotka–Volterra and Gray–Scott. At each rollout step, the model prediction is recursively fed back as the next input, making this evaluation more challenging than single-step prediction. On Lotka–Volterra, COUPLED-FLOW-GP achieves the best performance at 1τ, 2τ, and 10τ, while COUPLED-FLOW-Linear obtains the best performance at 5τ. On Gray–Scott, COUPLED-FLOW-GP consistently achieves the lowest error across all rollout horizons. These results indicate that the learned one-step transport is stable under recursive application and does not merely optimize short-horizon prediction.

Length-scale Sensitivity. Table 11 studies the sensitivity of COUPLED-FLOW-GP to the squared-exponential kernel length-scale. We vary the length-scale around the default value $\ell_0 = 1$ and evaluate the resulting performance on Lotka–Volterra and Gray–Scott. The performance remains stable across a broad range of length scales. For Lotka–Volterra, the error changes only slightly across the tested range, while for Gray–Scott the default setting achieves the best result. This shows that COUPLED-FLOW-GP does not require delicate length-scale tuning to obtain competitive performance.

Kernel Sensitivity. Table 12 compares different GP kernels, including squared-exponential, Matérn-$3/2$, Matérn-$5/2$, and rational quadratic kernels. All GP kernels perform competitively with or better than the linear interpolant. Different kernels achieve the best results on different datasets: Matérn-$3/2$ performs best on Lotka–Volterra, while the rational quadratic kernel performs best on Gray–Scott. This indicates that the empirical gains are not tied to one specific covariance function. Instead, the improvement mainly comes from replacing independent linear interpolation with a correlated GP-based path construction.

Coefficient-shift generalization. Table 17 evaluates out-of-distribution generalization under shifted coefficient conditions on Lotka–Volterra and Gray–Scott. COUPLED-FLOW-GP achieves the best OOD performance on both datasets, while COUPLED-FLOW-Linear remains the second-best method in most settings. On Lotka–Volterra, COUPLED-FLOW-GP improves over CViT-B under the OOD coefficient setting. On Gray–Scott, both COUPLED-FLOW variants substantially outperform the other baselines under coefficient shift. These results suggest that the proposed coupled transport objective improves robustness when the test dynamics deviate from the training coefficient distribution.

Empirical Verification of the Generalization Bound. Theorem 2 establishes an upper bound on the average Wasserstein-2 prediction error of CoupledFlow:

$$\mathrm{ERR}(\theta) \quad \leq \quad \kappa\tau\sqrt{L(\theta)},$$

where $\kappa = \exp(\int_0^\tau \ell_r\, dr)$ and ℓ_r denotes the Lipschitz constant of the induced instantaneous velocity. Importantly, the objective function $L(\theta)$ in Eq. (15) is not restricted to linear interpolants. In the GP variant, we use the coupled transport paths constructed in Algorithm 1: one randomly selected process is initialized using linear interpolation, while the remaining processes are constructed using the GP conditional means. To assess the informativeness of this bound, we empirically estimate its left-hand side (LHS) and right-hand side (RHS) on five multi-physics benchmarks. The path-matching loss

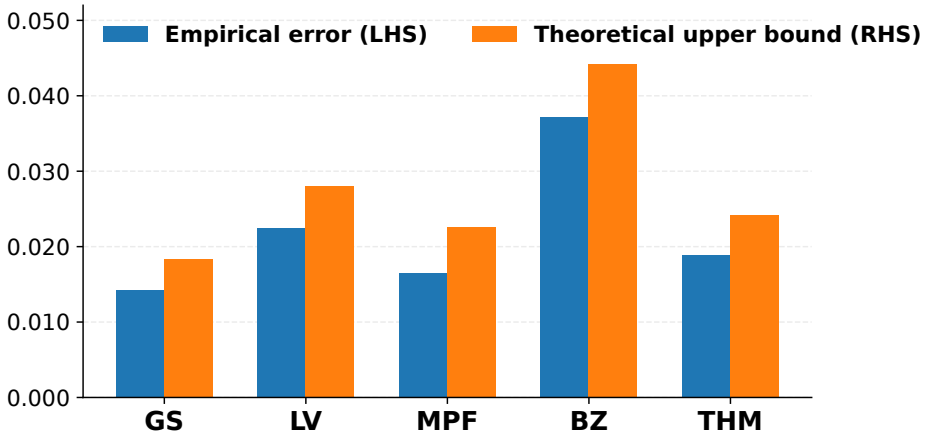

Figure 3: Empirical evaluation of the generalization bound in Theorem 2 across five coupled multi-physics benchmarks. We compare the estimated prediction error $\mathrm{ERR}(\theta)$ (LHS) with its theoretical upper bound $\kappa\tau\sqrt{\mathcal{L}(\theta)}$ (RHS). The empirical results satisfy the inequality across all datasets, with relatively small differences between the two quantities, suggesting that the bound is informative.

$L(\theta)$ is estimated with respect to the learned theta using Eq. (15) with 1000 sampled time points on a test set with 256 data points. The computation is also averaged over 5 independent runs. For each sampled time r, we estimate the local Lipschitz constant as $\ell_r = \max_{\boldsymbol{x}} \|\nabla_{\boldsymbol{x}\in\mathcal{S}_r} \boldsymbol{u}_\theta(\boldsymbol{x}, r)\|_2$, where $\mathcal{S}_r$ contains the states sampled from the held-out trajectories at time r. We then estimate the Jacobian spectral norm $|\cdot|_2$ following standard practice (Scaman and Virmaux, 2019) for estimating the network's local Lipschitz constant. Fig. 3 visualizes their comparison, which shows that across all five datasets, the empirical LHS remains below the estimated RHS. The relative gaps, defined as $(\mathrm{RHS} - \mathrm{LHS})/\mathrm{LHS}$, range from 18.8% to 37.0%, with an average of 27.6%. These results suggest that the bound remains informative for the GP-based formulation across tested coupled PDE systems.

Sparse Gaussian Process Baseline. To further investigate whether the performance gains of CoupledFlow-GP arise simply from the use of Gaussian Processes, we compare our method against a sparse Gaussian Process (SGP) (Hoang et al., 2015, 2016, 2017, 2019, 2020) baseline across five coupled multi-physics benchmarks. Specifically, the SGP directly learns the mapping from the initial coupled state $\boldsymbol{x}$, obtained by concatenating all physical processes, to the corresponding future state $\boldsymbol{x}'$ at the target prediction horizon. We adopt a sparse GP instead of a full GP due to the large number of training pairs in the Belousov–Zhabotinsky, Multiphase Flow, and THM datasets, which makes exact GP inference computationally expensive. For these three datasets, we use $M = 1500$ inducing points, yielding a computational complexity of $\mathcal{O}(M^3 + M^2N)$, where N denotes the number of training samples. As shown in Table 10, CoupledFlow-GP consistently outperforms the SGP baseline across all five benchmarks. Moreover, even CoupledFlow-Linear, which does not employ GP-based interpolants, substantially outperforms SGP. These results suggest that the superior performance of CoupledFlow-GP cannot be attributed solely to the use of Gaussian Processes. Instead, the proposed coupled mean-flow formulation, together with GP-based correlated intermediate paths, provides a more effective mechanism for learning cross-physics dependencies than direct GP regression.

Boundary and Initial-Condition Prediction Errors. To further assess the predictive performance of CoupledFlow, we evaluate additional metrics, including boundary condition (BC) and initial condition (IC) errors, alongside the overall relative L_2 prediction error. We conduct this evaluation on the Multiphase Flow (MPF) and Thermal-Hydro-Mechanical (THM) benchmarks using 512 training samples, with fresh training runs for each method. As shown in Table 9, CoupledFlow consistently achieves the lowest overall prediction error and boundary condition error on both benchmarks. On MPF, our method also obtains the lowest IC error, substantially outperforming all competing baselines across the three metrics. On THM, CoupledFlow achieves the lowest overall and boundary errors, while Transolver obtains the lowest IC error. These results demonstrate that the predictive improvements of CoupledFlow extend beyond the overall solution field to boundary

System	Method	Overall L_2	BC Error	IC Error
MPF	FNO	0.02687	0.04363	0.02766
	CViT	0.14251	0.16921	0.11824
	Transolver	0.01325	0.02340	0.01618
	COMPOL	0.09545	0.12070	0.09705
	CoupledFlow	**0.00575**	**0.00741**	**0.00539**
THM	FNO	0.00198	0.00216	0.00223
	CViT	0.02945	0.03293	0.03389
	Transolver	0.00184	**0.00183**	**0.00071**
	COMPOL	0.00171	0.00187	0.00170
	CoupledFlow	**0.00108**	**0.00130**	0.00123

Table 9: Comparison of overall relative L_2 prediction error, boundary condition (BC) error, and initial condition (IC) error on Multiphase Flow (MPF) and Thermal-Hydro-Mechanical (THM) benchmarks using 512 training samples. The best results are highlighted in bold.

Method	Lotka-V.	Belousov-Z.	Gray-Scott	Multiphase	THM
CoupledFlow-Linear	0.0234	0.0015	0.0023	0.0090	0.0011
CoupledFlow-GP	**0.0216**	**0.0011**	**0.0019**	**0.0061**	**0.0006**
Sparse GP	0.3554	0.0222	0.1162	0.0781	0.1443

Table 10: Comparison of relative L_2 prediction errors between CoupledFlow and the sparse Gaussian Process (SGP) baseline across five coupled multi-physics benchmarks. The best results are highlighted in bold.

and initial-condition regions. In particular, the consistently lower boundary errors suggest that our method more accurately captures solution behavior near domain boundaries.

D Detailed Experiment setup and hyperparameters

The experiments are conducted with 5 independent runs on the NVIDIA H100. The five simulated datasets were stimulated with numerical solvers using 256-point meshes for 1-D cases and 64×64 meshes for 2-D cases. As prior practice (Sun et al., 2025), we also trained the models on two dataset sizes (512 and 1024 samples) and evaluated performance on an independent test set of 200 samples. Coefficients and boundary conditions remained constant to isolate the effects of varying initial conditions. The number of parameters for each baseline across different datasets is provided in Tab. 5. The wall-clock time for each baseline across different datasets is provided in Tab. 6. The details of the hyperparameters are provided in Tab 7. The official code repository is provided at `https://github.com/khiembk/CoupledFlow`.

E Limitations

Our method has some limitations that are potential research directions for future work. First, COUPLED-FLOW assumes access to paired endpoint data $(\boldsymbol{x}, \boldsymbol{x}')$, where $\boldsymbol{x}'$ denotes the initial coupled state and $\boldsymbol{x}'$ denotes the corresponding future solution state. This setting is standard in neural operator learning, but it still requires high-fidelity numerical simulations to construct the training dataset. In practical scientific applications, however, one may only have sparse trajectories, noisy sensor measurements, or partially observed physical fields. Extending COUPLED-FLOW to learn from incomplete trajectories or partially observed coupled states is therefore an important direction for future work. Potential solutions include combining the proposed coupled transport objective with data-assimilation methods, uncertainty-aware training, or weakly supervised objectives that do not require fully paired endpoint observations. Second, COUPLED-FLOW is designed for efficient one-step inference, but its training procedure is more expensive than standard endpoint-regression neural operators. A useful future direction is to develop more efficient approximations of the local consistency objective, such as lightweight directional-derivative estimators, reduced-frequency

Method	Lotka-V.	Gray-Scott
COUPLED-FLOW-Linear	0.0234 ± 1e-4	0.0023 ± 1e-5
COUPLED-FLOW-GP ($0.25\ell_0$)	0.0226 ± 2e-4	0.0022 ± 3e-5
COUPLED-FLOW-GP ($0.5\ell_0$)	0.0218 ± 8e-5	0.0023 ± 2e-5
COUPLED-FLOW-GP ($1.0\ell_0$)	**0.0216 ± 2e-5**	**0.0019 ± 3e-5**
COUPLED-FLOW-GP ($2.0\ell_0$)	**0.0215 ± 1e-4**	0.0021 ± 4e-5
COUPLED-FLOW-GP ($4.0\ell_0$)	0.0217 ± 5e-5	**0.0020 ± 5e-5**

Table 11: Sensitivity of COUPLED-FLOW-GP to the squared-exponential kernel length-scale. We report relative L_2 errors ($\times 10^{-2}$) using 512 training samples with $\ell_0 = 1$ as the practical implementation.

Kernel	Lotka-V.	Gray-Scott
Our-Linear	0.0234 ± 1e-4	0.0023 ± 1e-5
Squared-Exponential	**0.0216 ± 2e-5**	**0.0019 ± 3e-5**
Matérn-3/2	**0.0214 ± 3e-5**	0.0022 ± 2e-5
Matérn-5/2	0.0230 ± 1e-4	0.0023 ± 7e-5
Rational Quadratic	0.0223 ± 2e-5	**0.0018 ± 1e-4**

Table 12: Kernel sensitivity of COUPLED-FLOW-GP. We report relative L_2 errors (mean ± std) using 512 training samples. All GP kernels use the same default length-scale $\ell_0 = 1$.

Method	*Train Samples*	*Lotka-V.*	*Belousov-Z.*	*Grey-Scott*	*Multiphase*	*THM*
FNO_c	512	0.1229 ±4e-5	0.0023 ±3e-5	0.0088 ±4e-5	0.0229 ±2e-4	0.0015 ±2e-5
	1024	0.0955 ±6e-5	0.0016 ±3e-5	0.0055 ±3e-6	0.0291 ±1e-5	0.0011 ±1e-6
CMWNO	512	0.0735 ±3e-5	0.0021 ±2e-5	0.0083 ±5e-4	0.0256 ±2e-5	0.0013 ±4e-5
	1024	0.0665 ±1e-5	0.0014 ±6e-5	0.0043 ±3e-5	0.0193 ±6e-5	0.0009 ±5e-5
$UFNO_c$	512	0.1163 ±1e-4	0.0026 ±6e-6	0.0055 ±5e-5	0.0448 ±1e-5	0.0013 ±3e-5
	1024	0.0873 ±8e-5	0.0017 ±2e-5	0.0036 ±6e-6	0.0387 ±1e-5	0.0008 ±4e-5
$Transolver_c$	512	0.1819 ±1e-5	0.0597 ±7e-6	0.0054 ±9e-6	0.0128 ±4e-5	0.0018 ±1e-5
	1024	0.1767 ±1e-5	0.0188 ±2e-5	0.0049 ±2e-5	0.0088 ±3e-5	0.0016 ±5e-5
$DeepONet_c$	512	0.5546 ±7e-5	0.0111 ±1e-5	0.3130 ±1e-5	0.3120 ±1e-5	0.3136 ±1e-5
	1024	0.5483 ±6e-5	0.0099 ±1e-5	0.3099 ±1e-5	0.2819 ±7e-5	0.3150 ±1e-5
CViT-B	512	**0.0163±6e-4**	0.0104±7e-4	0.0209±2e-3	0.0212±4e-4	0.0127±6e-4
	1024	**0.0077±3e-4**	0.0031±1e-5	0.0074±5e-4	0.0139±4e-4	0.0051±1e-5
$DiffusionPDE_c$	512	0.0441 ± 3e-4	0.0033 ± 2e-5	0.0026± 1e-4	0.0361 ± 5e-5	0.0021 ± 3e-5
	1024	0.0310 ± 3e-5	0.0053 ± 7e-5	0.0021 ± 5e-4	0.0025 ± 5e-5	0.0016 ± 2e-5
COMPOL-RNN	512	0.1043 ±2e-5	0.0017 ±2e-5	0.0066 ±1e-4	0.0221 ±1e-5	0.0014 ±3e-5
	1024	0.0802 ±4e-5	0.0015 ±8e-6	0.0050 ±4e-5	0.0644 ±3e-5	0.0009 ±4e-5
COMPOL-ATN	512	0.1132 ±1e-4	0.0018 ±4e-5	0.0061 ±3e-5	0.0217 ±3e-5	0.0014 ±3e-5
	1024	0.0858 ±5e-5	0.0015 ±3e-5	0.0048 ±2e-5	0.0676 ±3e-5	0.0010 ±2e-5
$MeanFlow_c$	512	0.1650 ±4e-3	0.0018 ±6e-5	0.0033 ±5e-5	0.0112 ±4e-5	0.0017 ±3e-5
	1024	0.1320 ±5e-3	0.0017 ±2e-5	0.0029 ±6e-5	0.0333 ± 7e-5	0.0013 ±4e-5
COUPLED-FLOW-Linear	512	0.0234 ±1e-4	**0.0015 ±5e-5**	**0.0023 ±1e-5**	**0.0090 ±5e-5**	**0.0011 ±6e-5**
	1024	0.0129 ±4e-5	**0.0009 ±2e-5**	**0.0018 ±5e-5**	**0.0009 ± 4e-5**	**0.0009 ±2e-5**
COUPLED-FLOW-GP	512	**0.0216 ±2e-5**	**0.0011 ±4e-5**	**0.0019 ±3e-5**	**0.0061 ±1e-5**	**0.0006 ±2e-5**
	1024	**0.0121 ±3e-5**	**0.0006 ±2e-5**	**0.0015 ±5e-5**	0.0011 ±1e-5	**0.0005 ±2e-5**

Table 13: Comparison of relative L_2 prediction errors (mean ± std) of our method with different neural operators and generative baselines across different coupled physics datasets. **Bold** and **Blue** values indicate the best and second-best performance, respectively.

consistency regularization, or shared low-rank coupling structures. Such improvements could further improve the scalability of COUPLED-FLOW while preserving its one-step inference advantage.

Method	Time-Step			
	1 τ	**2** τ	**5** τ	**10** τ
MeanFlow$_c$	0.7420 ± 2e-3	1.3310 ± 5e-3	2.3890 ± 1e-2	4.939 ± 3e-2
FNO$_c$	0.1240 ± 3e-4	0.2610 ± 8e-4	0.4000 ± 1e-3	0.791 ± 2e-3
COMPOL-RNN	0.1090 ± 2e-4	0.2260 ± 5e-4	0.3780 ± 1e-3	0.533 ± 3e-3
COMPOL-ATN	0.1180 ± 2e-4	0.2890 ± 7e-4	0.3640 ± 2e-3	0.580 ± 4e-3
CMWNO	0.0735 ± 1e-4	0.4528 ± 8e-4	0.7176 ± 2e-3	1.030 ± 5e-3
COUPLED-FLOW-GP	**0.0216 ± 2e-4**	**0.1330 ± 5e-4**	0.2510 ± 7e-4	**0.3650 ± 1e-3**
COUPLED-FLOW-Linear	0.0340 ± 1e-4	0.1450 ± 3e-4	**0.2351 ± 7e-4**	0.3772 ± 2e-3

Table 14: Comparison of relative L_2 error rollout prediction (mean ± std) of our method with different baselines on Lotka-V dataset. **Bold** and **Blue** values indicate the best and second-best performance, respectively.

Method	Time-Step			
	1τ	**2**τ	**5**τ	**10**τ
MeanFlow$_c$	0.0033 ± 1e-4	0.0102 ± 2e-4	0.0262 ± 7e-4	0.0591 ± 1e-3
FNO$_c$	0.0088 ± 2e-4	0.0139 ± 7e-4	0.0184 ± 1e-3	0.0368 ± 3e-3
COMPOL-RNN	0.0066 ± 1e-4	0.0134 ± 4e-4	0.0121 ± 1e-3	0.0287 ± 2e-3
COMPOL-ATN	0.0051 ± 1e-4	0.0133 ± 5e-4	0.0119 ± 1e-3	0.0211 ± 2e-3
CMWNO	0.0083 ± 2e-4	0.0143 ± 6e-4	0.0196 ± 2e-3	0.0226 ± 4e-3
COUPLED-FLOW-GP	**0.0019 ± 1e-4**	**0.0025 ± 3e-4**	**0.0069 ± 8e-4**	**0.0167 ± 1e-3**
COUPLED-FLOW-Linear	0.0023 ± 1e-4	0.0029 ± 2e-4	0.0071 ± 1e-3	0.0168 ± 2e-3

Table 15: Comparison of relative L_2 error rollout prediction (mean ± std) of our method with different baselines on GrayScott dataset. **Bold** and **Blue** values indicate the best and second-best performance, respectively.

F Broader Impacts

This work develops a one-step neural operator for coupled multi-physics PDE systems. The main positive impact is that accurate and efficient surrogate models can reduce the cost of repeatedly solving expensive coupled PDEs. This can benefit scientific and engineering applications such as subsurface flow modeling, thermal-hydro-mechanical simulation, climate and environmental modeling, materials design, and other domains where high-fidelity numerical simulation is costly. Because COUPLED-FLOW performs one-step inference after training, it may also reduce inference-time energy consumption compared with iterative generative PDE solvers or repeated numerical simulations.

At the same time, surrogate PDE solvers can have negative consequences if used without proper validation. In safety-critical settings, such as geomechanics, energy systems, environmental forecasting, or infrastructure design, inaccurate predictions could lead to incorrect scientific conclusions or poor engineering decisions. The proposed method should therefore be viewed as an acceleration tool rather than a replacement for validated numerical solvers. For high-stakes applications, users should perform uncertainty quantification, out-of-distribution testing, and domain-specific physical validation before deployment.

The work does not involve human subjects, personal data, private information, or crowdsourced annotations. The experiments are conducted on simulated and publicly available scientific benchmark datasets. As a result, we do not anticipate direct privacy, fairness, or human-subject risks. However, like other scientific machine-learning tools, the method could be applied in domains with environmental or industrial consequences. Responsible use requires transparent reporting of model assumptions, data-generation procedures, and failure cases.

Method	Reaction-Diffusion-2D	Navier-Stokes-2D
MeanFlow$_c$	0.0951 ± 4e-4	0.1613 ± 1e-3
FNO$_c$	0.1234 ± 1e-3	0.0912 ± 7e-4
UFNO$_c$	0.1233 ± 7e-4	0.0947 ± 3e-4
COMPOL-RNN	0.0828 ± 3e-4	0.0825 ± 6e-4
COMPOL-ATN	0.0827 ± 4e-4	0.0821 ± 1e-3
CMWNO	0.0845 ± 7e-4	0.0842 ± 2e-4
DiffusionPDE	0.1265 ±1e-3	0.0884 ± 6e-4
CViT-B	**0.0158 ± 5e-4**	**0.0395 ± 2e-4**
COUPLED-FLOW-GP	**0.0163 ± 3e-4**	**0.0345 ± 4e-4**
COUPLED-FLOW-Linear	0.0174 ± 1e-4	0.0398 ± 5e-4

Table 16: Comparison of relative L_2 prediction error (mean ± std) on Reaction-Diffusion-2D and Navier-Stokes-2D. **Bold** values indicate the best performance for each case. **Blue** values indicate the second-best performance for each case.

Method	Lotka–Volterra		Gray–Scott	
	ID coeff.	OOD coeff.	ID coeff.	OOD coeff.
FNO$_c$	0.1229 ±4e-5	0.1815 ± 1e-4	0.0088 ± 4e-5	0.1015 ± 1e-4
CMWNO	0.0735 ± 3e-5	0.1606 ±2e-4	0.0083 ±5e-4	0.1139 ± 3e-4
COMPOL-ATN	0.1132 ± 1e-4	0.1523 ± 3e-4	0.0061 ±3e-5	0.0888 ± 2e-4
COMPOL-RNN	0.1143 ±1e-4	0.1627 ± 7e-5	0.0066 ±1e-4	0.1136 ± 1e-4
DiffusionPDE	0.0441 ± 3e-4	0.1098 ± 1e-4	0.0026 ± 1e-4	0.0067 ± 2e-4
CViT-B	**0.0163 ± 6e-4**	**0.0759 ± 6e-5**	0.0209 ±2e-3	0.1055 ± 1e-4
COUPLED-FLOW-Linear	0.0234 ± 1e-4	0.0798 ± 3e-5	**0.0023 ±1e-5**	**0.0061 ± 3e-4**
COUPLED-FLOW-GP	**0.0216 ±2e-5**	**0.0731 ± 7e-5**	**0.0019 ±3e-5**	**0.0052 ±2e-4**

Table 17: Coefficient-shift generalization on Lotka–Volterra and Gray–Scott. We report relative L_2 errors (mean ± std) under in-distribution (ID) and out-of-distribution (OOD) coefficient settings.

G Details of Multi-Physics Datasets

1-D Lotka–Volterra. The Lotka–Volterra system models predator-prey dynamics with spatial diffusion. It contains two coupled fields $u(x,t)$ and $v(x,t)$:

$$\begin{aligned} \frac{\partial u}{\partial t} &= D_u \nabla^2 u + au - buv, \\ \frac{\partial v}{\partial t} &= D_v \nabla^2 v + cuv - dv. \end{aligned}$$

Follow the prior setting (Sun et al., 2025), the initial conditions are sampled from a one-dimensional Gaussian random field with length-scale $l = 0.1$ and amplitude $\sigma = 1$. The interaction parameters are fixed as $a = b = c = d = 0.01$, and the diffusion coefficients are fixed as $D_u = D_v = 0.01$. The dataset is discretized on a 1-D mesh, and the learning task is to predict the later coupled state (u, v) from the earlier coupled state.

1-D Belousov–Zhabotinsky. The Belousov–Zhabotinsky system describes nonlinear chemical oscillations with three coupled reactants $u(x,t)$, $v(x,t)$, and $w(x,t)$:

$$\begin{aligned} \frac{\partial u}{\partial t} &= \epsilon_1 \nabla^2 u + u + v - uv - u^2, \\ \frac{\partial v}{\partial t} &= \epsilon_2 \nabla^2 v + w - v - uv, \\ \frac{\partial w}{\partial t} &= \epsilon_3 \nabla^2 w + u - w. \end{aligned}$$

Follow the prior setting (Sun et al., 2025), we set $\epsilon_1 = 10^{-2}$, $\epsilon_2 = 10^{-2}$, and $\epsilon_3 = 5 \times 10^{-3}$. The system is simulated from $t = 0$ to $t = 0.5$.

2-D Gray–Scott. The Gray–Scott system models two-species reaction-diffusion pattern formation. It contains an activator $u(x, y, t)$ and an inhibitor $v(x, y, t)$:

$$\begin{aligned}\frac{\partial u}{\partial t} &= D_u \nabla^2 u - uv^2 + F(1-u),\\ \frac{\partial v}{\partial t} &= D_v \nabla^2 v + uv^2 - (F+k)v.\end{aligned}$$

Follow prior practice (Sun et al., 2025), we use the parameter setting $D_u = 0.12$, $D_v = 0.06$, $F = 0.054$, and $k = 0.063$. The initial state is sampled from a two-dimensional Gaussian random field and evolved until $T = 20$. The learning task is to predict the future coupled field (u, v) from the earlier field.

2-D multiphase flow. The multiphase benchmark describes oil-water two-phase flow in porous media. Let α and β denote the water and oil components. The governing equations follow component-wise mass conservation:

$$\begin{aligned}\frac{\partial M^\alpha}{\partial t} &= -\nabla \cdot F_a^\alpha + q^\alpha,\\ \frac{\partial M^\beta}{\partial t} &= -\nabla \cdot F_a^\beta + q^\beta,\end{aligned}$$

where M^α, M^β are component mass accumulations, F_a^α, F_a^β are advective fluxes, and q^α, q^β are source or sink terms. Following the simulation setup, diffusive fluxes are neglected. The phase velocity is given by Darcy's law:

$$u_p = -k\left(\nabla P_p - \rho_p g\right)\frac{k_{rp}}{\mu_p},$$

where k is permeability, P_p is phase pressure, ρ_p is phase density, k_{rp} is relative permeability, and μ_p is viscosity. Follow (Sun et al., 2025), the simulations are performed with GEOS on a 64×64 two-dimensional grid. Water is injected from two randomly placed sources on the left boundary, and oil is produced from two randomly placed sinks on the right boundary. The permeability field satisfies $k_x = k_y$ and is generated by a fractal algorithm with $k_{\min} = 1$mD, $k_{\max} = 1000$mD, and $k_{\text{base}} = 100$mD. The dataset contains 1024 scenarios, each simulated for 15 time steps with step size 5×10^5 seconds. The stored variables are phase pressure P_p and phase saturation S_p, resulting in data of size $1024 \times 15 \times 64 \times 64 \times 2$.

2-D thermo-hydro-mechanical system. The thermo-hydro-mechanical benchmark models coupled pressure diffusion, heat transfer, and mechanical deformation in porous media. The governing equations consist of momentum balance, fluid mass balance, and heat transfer:

$$\begin{aligned}Gu_{i,jj} + \frac{G}{1-2\nu}u_{j,ji} - \alpha p_{,i} - \frac{2G\alpha_m^T(1+\nu)}{3(1-2\nu)}T_{,i} + F_i = 0,\\ \frac{\partial p}{\partial t} = M\left[\frac{k}{\mu}p_{,jj} - \alpha\frac{\partial \epsilon_{kk}}{\partial t} + \left(\alpha\alpha_m^T + \phi_0(\alpha_f^T - \alpha_m^T)\right)\frac{\partial T}{\partial t}\right] + \gamma,\\ \frac{\partial T}{\partial t} = \frac{1}{\rho_t C_t}(\kappa^T T_{,i})_{,i} - \frac{\rho_f C_f}{\rho_t C_t}v_i T_{,i}.\end{aligned}$$

Here, G is the shear modulus, ν is Poisson's ratio, α is Biot's coefficient, p is pore pressure, T is temperature, F_i is the body force, M is the Biot modulus, k/μ is hydraulic conductivity, ϵ_{kk} is volumetric strain, ϕ_0 is initial porosity, α_m^T and α_f^T are thermal expansion coefficients of porous media and fluid, γ is the fluid source or sink term, ρ_t, C_t are total density and heat capacity, κ^T is the thermal conductivity tensor, and ρ_f, C_f are fluid density and heat capacity.

2-D Reaction-Diffusion from PDEBench. We also evaluate on the public Reaction-Diffusion-2D benchmark from PDEBench. This system contains two coupled variables, an activator $u(t, x, y)$ and an inhibitor $v(t, x, y)$:

$$\begin{aligned}\frac{\partial u}{\partial t} &= D_u\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + R_u(u, v),\\ \frac{\partial v}{\partial t} &= D_v\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) + R_v(u, v).\end{aligned}$$

The reaction terms follow the FitzHugh–Nagumo form:

$$R_u(u,v) = u - u^3 - k - v, \qquad R_v(u,v) = u - v.$$

The PDEBench release provides trajectories with two variables over time, and we use the same preprocessing and train/test split protocol as described in our experimental setup.

2-D Navier–Stokes from PDEArena. We further evaluate on the public Navier-Stokes-2D benchmark from PDEArena. This dataset contains a scalar advected field $c(t,x,y)$ and an incompressible velocity field $v(t,x,y) = (v_x(t,x,y), v_y(t,x,y))$, giving three coupled channels (c, v_x, v_y). The dynamics can be written as an incompressible buoyancy-driven flow:

$$\begin{aligned} \frac{\partial c}{\partial t} + v \cdot \nabla c &= 0, \\ \frac{\partial v}{\partial t} + (v \cdot \nabla) v &= -\nabla p + \nu \Delta v + cb, \\ \nabla \cdot v &= 0, \end{aligned}$$

where p is the pressure, ν is the viscosity, and $b = (0, b_y)$ is the buoyancy direction. The PDEArena generator initializes the scalar field with a smooth random noise field and initializes the velocity field to zero. At each solver step, the scalar field is advected by the velocity, the velocity is advected by itself, buoyancy forcing is added, viscosity is applied through diffusion, and the velocity is projected to an incompressible field. The official HDF5 trajectories store the scalar channel c and the velocity channels v_x, v_y, which are used as the three coupled processes in our experiments.

H Related Work

H.1 Flow-based Generative Models

Generative models based on flow matching (Lipman et al., 2022, 2024) offer a scalable alternative to diffusion methods by defining a continuous normalizing flows through the means of ordinary differential equations (ODEs). Flow matching transport the samples from a source distribution to a target distribution by learning a time-dependent velocity field using a regression target without the need to simulate during training. During inference, a new sample from the target distribution is generated by integrating the ODEs. Recent advancements aim to reduce the ODE integration costs by straightening flow trajectories such as ReFlow (Liu et al., 2022), Optimal FLow Matching (Kornilov et al., 2024). Other methods such as MeanFlow (Geng et al., 2025) or Consistency Models Song et al. (2023) directly learn one-step mappings between the source to target distributions, thus entirely avoiding the costly ODE integration.

H.2 Single Operator Learning

Neural operator methods, which leverage neural networks as their foundational architecture, present an innovative approach to surrogate modeling that maps input functions to output functions. FNO (Li et al., 2021) formulates a neural operator by parameterizing the integral kernel directly in Fourier space, allowing for an expressive and efficient architecture. Building on these innovations, researchers introduced the GNO (Li et al., 2020b) and MGNO (Li et al., 2020a), which combine Nystrom approximation with graph neural networks to approximate function convolution. Another significant contribution emerged with the DeepONet (Lu et al., 2021), which employs a dual-network architecture combining a branch network for input functions and a trunk network for sampling locations. Recently, Wang et al. (2025) propose CVIT, which combines a vision transformer encoder, a novel grid-based coordinate embedding, and a query-wise cross-attention mechanism to capture multi-scale dependencies effectively.

Generative models have gained increasing interests in recent years to solve PDEs. DiffusionPDE (Huang et al., 2024) uses denoising diffusion probabilistic models to jointly model the distributions of the coefficient and solution spaces in order to solve forward as well as backward PDE problems under missing data regimes. DYffusion (Rühling Cachay et al., 2023) leverages dynamics of the spatiotemporal data to train diffusion models, facilitating multi-step and long-range PDE predictions. Physics-Based Flow Matching (PBFM) (Baldan et al., 2025) and Physics-Informed Diffusion Models

(PBDM) (Bastek et al., 2024) combine generative loss and physical residual in the objective functions to learn the underlying physics of the data. ECI (Cheng et al., 2024) and PCFM (Utkarsh et al., 2025) incorporate additional projection step to the generated solution samples to enforce hard physical constraints. Although these methods have shown great potential in modeling single-physics systems, extending them to multiphysics systems remains a non-trivial challenge , requiring further advances in modeling cross-physics interactions and capturing complex coupling dynamics.

H.3 Coupled Operator Learning

Coupled operator learning extends standard neural operator learning from single-field systems to multi-physics systems in which several physical variables evolve jointly and influence one another. Rather than learning a single operator from one input field to one output field, the goal is to learn a family of operators whose predictions for each process depend on the full coupled state. CMWNO (Xiao et al., 2023) addresses this problem by decomposing coupled integral kernels in the multiwavelet domain, allowing cross-process interactions to be represented during wavelet decomposition and reconstruction. While effective, this coupling mechanism is tied to a particular wavelet-space factorization and predefined decomposition structure, which may limit flexibility when the number of processes, coupling patterns, or spatial resolutions varies. More recently, COMPOL (Sun et al., 2025) builds on Fourier neural operators by maintaining process-specific operator streams and exchanging information through recurrent or attention-based aggregation modules. This architecture provides a flexible way to model inter-process dependencies in latent feature space and has shown strong empirical performance on several coupled PDE datasets. Related attention-based neural operators also suggest that explicit information exchange across physical variables can improve multi-physics prediction. However, these approaches remain primarily *architecture-centric*: they model coupling through feature aggregation inside the neural operator, while the learning objective is still based on direct endpoint regression from initial states to future states.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: see Section 3

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: see Appendix E

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: see Appendix A.2 and Appendix B

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: see Appendix D

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: see Appendix D

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: see Section 4 and Appendix D

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: see Appendix C

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: see Appendix D

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work focuses on learning surrogate neural operators for coupled multiphysics PDE systems using simulated and publicly available scientific benchmarks. It does not involve human subjects, personally identifiable information, sensitive data, or deployment in safety-critical real-world settings. We have cited the datasets and prior methods used in the experiments and followed standard scientific reporting practices.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: see Appendix F

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release data or models with a high risk for misuse. The work focuses on surrogate modeling for coupled multi-physics PDE systems using simulated and publicly available scientific benchmark datasets, and does not involve scraped datasets, human-subject data, large pretrained generative models, or systems intended for safety-critical deployment.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

    Justification: We use existing public benchmark datasets and baseline implementations only for research and reproducibility purposes. The original creators are credited through citations to the corresponding papers and official repositories. For each dataset and codebase used in our experiments, we report the source, version or access date when applicable. We respect the stated terms of use and do not repackage or redistribute existing assets beyond what is allowed by their licenses.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We will release the implementation of our method, including training and inference scripts, configuration files, and instructions for reproducing the reported experiments. The released assets will be documented with usage instructions, required dependencies, hyperparameter settings, and limitations. Our work does not introduce a new dataset involving human subjects, and all benchmark datasets used in the paper are existing public datasets. The release will be anonymized during the review period.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing, user studies, or research with human subjects. All experiments are conducted on simulated and publicly available PDE benchmark datasets, so there are no participant instructions, screenshots, or compensation details to report.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve research with human subjects, crowdsourcing, user studies, personally identifiable information, or participant data. All experiments are conducted on simulated and publicly available PDE benchmark datasets, so IRB approval or equivalent human-subjects review is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.