COMPOSITIONAL NEURAL OPERATORS FOR MULTI-DIMENSIONAL FLUID DYNAMICS

Hamda Hmida
Centre for Material Forming
Mines Paris - PSL University
Valbonne, France

Hsiu-Wen CHANG
Centre for Robotics
Mines Paris - PSL University
Paris, France

Youssef MESRI *
Centre for Material Forming
Mines Paris - PSL University
Valbonne, France

ABSTRACT

Partial differential equations (PDEs) govern diverse physical phenomena, yet high-fidelity numerical solutions are computationally expensive and Machine Learning approaches lack generalization. While Scientific Foundation Models (SFMs) aim to provide universal surrogates, typical encoding-decoding approaches suffer from high pretraining costs and limited interpretability. In this paper, we propose Compositional Neural Operators (CompNO)[1] for 2D systems, a framework that decomposes complex PDEs into a library of Foundation Blocks. Each block is a specialized Neural Operator pretrained on elementary physics. This modular library contains convection, diffusion, and nonlinear convection blocks as well as a Poisson Solver, enabling the framework to address the pressure-velocity coupling. These experts are assembled via an Adaptation Block featuring an Aggregator. This aggregator learns nonlinear interactions by minimizing data loss and physics-based residuals driven from governing equations. The proposed approach has been evaluated on the Convection-Diffusion equation, the Burgers' equation, and the Incompressible Navier-Stokes equation. Our results demonstrate that learning from elementary operators significantly improves adaptability, enhances model interpretability and facilitates the reuse of pretrained blocks when adapting to new physical systems.

Key words. Scientific foundation models; Neural operators; PDE simulation; Computational fluid dynamics; Fourier neural operator, Physics-Informed Neural Networks.

1 INTRODUCTION

Solving partial differential equations (PDEs) is a cornerstone of modeling and simulating complex phenomena in science and engineering. In fields such as computational fluid dynamics (CFD), PDEs serve as the primary framework for describing fluid behavior, relating variables like velocity and pressure to the system's evolution over time. Traditional numerical methods, including finite difference, finite volume, and finite element methods, provide high-fidelity results but often require significant computational resources. This cost becomes particularly prohibitive when simulations must be repeated across wide parameter variations or for real-time applications. Recently, machine/deep learning approaches have emerged as promising alternatives. Some discrete data-driven methods represent PDE solutions directly with neural networks tailored to specific equations and discretization. Early work leveraged Convolutions (Zhu & Zabaras, 2018; Bhatnagar et al., 2019), Residual (Taylor et al., 2023), and Graph (Pelissier et al., 2024) network architectures to encode spatial information and evolve solutions over time, while physics-informed frameworks (PINNs (Raissi et al., 2019; Yang & Mesri, 2022), hPINN (Lv et al., 2023), PIKANs (Wang et al., 2025)) enabled unsupervised and semi-supervised recovery of single-instance solutions. The problem with these neural solvers is their limited ability to handle discretization effectively as well as their dependency on system configuration. Other continuous frameworks like neural operators (Kovachki

*Corresponding Author: youssef.mesri@minesparis.psl.eu

[1]Code and Data will be made available on request.

et al., 2023) and DeepONet (Lu et al., 2021) can learn mappings between function spaces by approximating operators instead of data distribution. DeepONet (Lu et al., 2021) employs a dual network design to map infinite-dimensional functions, while the Fourier Neural Operator (FNO) (Li et al., 2020) leverages the fast Fourier transform to achieve discretization invariance. Some extensions, like GINO (Li et al., 2023), were proposed to adapt it to complex geometries, while other work, such as PI-DeepONet (Wang et al., 2021) and PFNO (Yu & Hodzic, 2024), were trying to enhance the performance of neural operators, but they still face challenges related to limited generalization abilities.

Recent advancements have shifted focus toward the development of Scientific Foundation Models (SFMs). These are data-driven models trained on a broad distribution of physical systems, designed to exhibit wide generalization capabilities across different scientific problems and conditions without requiring retraining from scratch (Choi et al., 2025). SFMs (Totounferoush et al., 2025; Menon et al., 2026) serve as reusable bases that reduce the need for specific solvers for every unique PDE type. Researchers have explored several avenues for these models, including using language models to process symbolic PDE forms and utilizing simulation data to predict future states.

Despite their promise, current SFMs face notable challenges, such as the high computational expense of pretraining on massive, heterogeneous physics datasets and a lack of physical interpretability in their architectures. In this work, we present Compositional Neural Operators (CompNO), a foundation model approach that aims to address these limitations through the principle of compositionality. Building upon the modular framework introduced by Hmida et al. (2026), which demonstrated efficacy in one-dimensional settings, our work significantly extends the scope and utility of compositional neural operators in three key ways. First, we transition from one-dimensional setups to complex two-dimensional systems, specifically targeting the 2D Incompressible Navier-Stokes (INS) equations. Second, we introduce Physics-Informed Aggregation within the Adaptation Blocks to ensure that the assembly of Foundation Blocks remains consistent with underlying physical constraints. Finally, by constructing a library of specialized blocks pretrained on fundamental differential operators, CompNO demonstrates superior performance over well-established methods. This modular strategy improves data efficiency, enhances physical interpretability, and enables robust generalization across diverse, high-dimensional physical regimes.

2 Related Work

To develop foundation models capable of solving diverse PDE systems, recent work has explored two primary paradigms. The first paradigm leverages language models that condition solution prediction on the symbolic form of the governing PDE, treating mathematical expressions as structured sequences analogous to text. Representative examples include PROSE Liu et al. (2024b) and PROSE-FD (Liu et al., 2024a), which employ multimodal transformer architectures to jointly encode symbolic PDE descriptions and numerical solution data, enabling zero-shot or few-shot generalization across equation families. A related approach (Yang et al., 2025) fine-tunes large pretrained language models to incorporate textual or symbolic representations of PDEs as conditioning context for operator learning. Similarly, Unified Pretrained Solvers (UPS) (Shen et al., 2024) integrate symbolic and numerical modalities through large-scale pretraining, yielding a single model that transfers effectively across multiple PDE classes with limited task-specific data.

The second paradigm dispenses with explicit symbolic supervision and relies exclusively on simulation data, learning to predict future states from previous observations. Within this setting, VICON Cao et al. (2024) exploits in-context learning to infer solution operators directly from example trajectories, enabling rapid adaptation to new PDE instances. BCAT (Liu et al., 2025) introduces block-causal transformers for autoregressive prediction of high-dimensional fluid states, while Poseidon (Herde et al., 2024) employs multiscale operator transformers pretrained on fluid dynamics datasets to achieve strong cross-equation generalization. Multiple Physics Pretraining (MPP) (McCabe et al., 2024) further extends this approach by training a single model across datasets drawn from multiple PDE systems, encouraging the emergence of shared physical representations. Furthermore, PhysiX (Nguyen et al., 2025) adopts large-scale autoregressive modeling inspired by video transformers, demonstrating that foundation models trained purely on spatiotemporal simulation data can generalize across diverse physical systems without access to explicit governing equations.

Despite these advances, existing foundation models for PDEs exhibit several limitations. Many approaches require large volumes of expensive simulation data and substantial computational resources for pretraining, and often rely on opaque latent representations that limit interpretability. In contrast, CompNO (Hmida et al., 2026) overcomes these challenges by eschewing a monolithic latent space for heterogeneous physical systems. Instead, CompNO is first pretrained to learn dynamic representations of common differential operators. It is then fine-tuned to solve specific PDE problems by routing and composing the operators that govern the target system.

3 METHODOLOGY

3.1 PROBLEM SETTING

We consider parametric partial differential equations defined in a domain $D \subset \mathbb{R}^n$ and $P \subset \mathbb{R}^p$ is the parameter space where p is the number of parameters of this PDE. We define a parametric operator H as a time-stepping solver. Given the state of the system $u_t \in D$ at time t and a set of physical parameters $\gamma \in P$, the model is designed to predict the incremental change δu_t over a discrete time step Δt:

$$\delta u_t = H(u_t, \gamma) \tag{1}$$

then the state at the subsequent time step is determined by the update rule:

$$u_{t+1} = u_t + \delta u_t \tag{2}$$

This formulation allows the model to focus on learning the underlying dynamics rather than the absolute magnitude of the field.

We approximate the operator H using a neural operator H_θ, where $\theta \in \Theta$ is the space of all trainable parameters in the neural network. Formally, the neural operator represents a mapping between function spaces:

$$H_\theta : D \times P \to D$$

The theoretical objective is to identify the optimal parameter set θ^* in the parameter space Θ that minimizes the risk across the entire distribution of initial conditions and physical parameters:

$$\theta^* = \arg\min_{\theta \in \Theta} \mathbb{E}\left[\mathcal{L}(\mathcal{H}_\theta(u_t, \gamma), \delta u_t^{Truth})\right] \tag{3}$$

However, in practice, we have access to a finite dataset of N observations. Therefore, the training process yields an estimated parameter set $\hat{\theta}$, which is the minima of the empirical loss function:

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\mathcal{H}_\theta(u_t^{(i)}, \gamma^{(i)}), \delta u_t^{(i),Truth}) \tag{4}$$

While $\hat{\theta}$ is the optimal solution for the provided observations, it remains an approximation of the true optimal θ^*. The convergence of $\hat{\theta}$ to θ^* is governed by the diversity of the training data.

Then, the ultimate goal of $H_{\hat{\theta}}$ is to generalize from a known sequence of time steps to predict the long-term evolution of the system. Given initial observation up to time T_0:

$$\{u_i \mid 0 \leq i \leq T_0\}$$

the model performs an autoregressive rollout to predict the next T time steps.

$$\hat{\mathbf{U}} = \{\hat{u}_{i+1} \mid \hat{u}_{i+1} = \hat{u}_i + \mathcal{H}_{\hat{\theta}}(\hat{u}_i, \gamma) \text{ for } T_0 < i < T_0 + T\} \tag{5}$$

In this iterative process, each prediction $\hat{u}_{i+1}$ is fed back into the model as the input for the following inference cycle. This approach tests the model's ability to minimize error accumulation and maintain physical consistency over extended temporal horizons.

3.2 General Formulation of CompNO

Compositional Neural Operators is built on the principle of compositionality, which states that the behavior of complex physical systems emerges from the interactions of their individual components. This mathematical perspective allows us to translate physical modularity directly into our neural framework through a modular design consisting of Foundation Blocks and an Adaptation Block, as illustrated in Figure 1.

The translation of compositionality into our architecture follows a "pre-training assemble fine-tune" paradigm, where Foundation Blocks are pretrained to approximate elementary operators identified in the governing equations. Then, during fine-tuning, the Adaptation Block provides the physical context for the solutions generated by the Foundation Blocks.

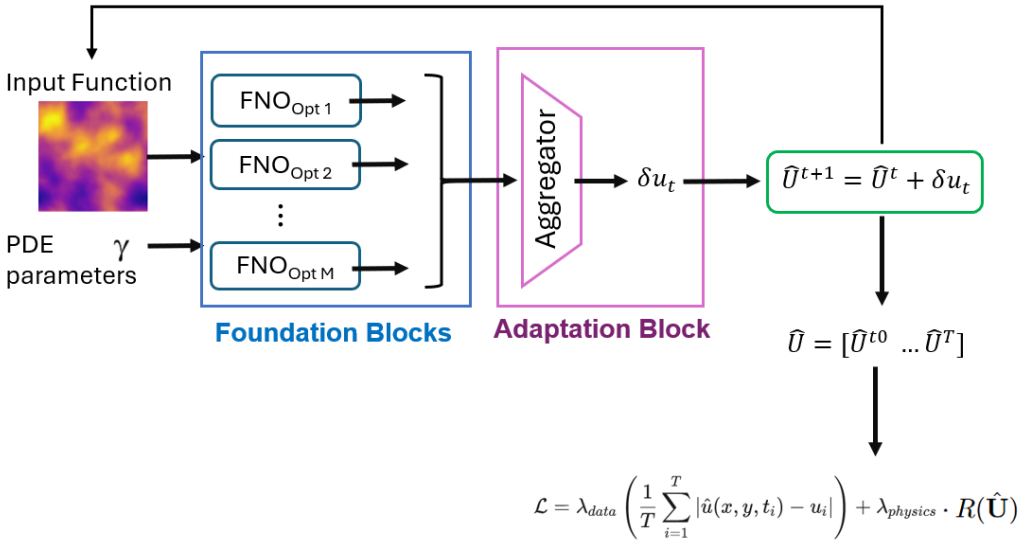

Figure 1: CompNO model overview: The input consists of the initial function state $\hat{U}^t$ with the physical parameter vector γ. The Foundation Blocks are pretrained Neural Operators that independently predict the time-evolution corresponding to specific elementary operators. The Aggregator is a neural network that combines their embeddings. The prediction $\hat{U}^{t+1}$ will be used as input for the next time step prediction. The training process incorporates data-driven loss as well as physics loss.

3.3 Foundation Blocks

In our architecture illustrated in Figure 1, the Foundation Blocks are a library consisting of expert modules pretrained to capture the dynamics of elementary physical operators in a 2D domain. Each block is a Parametric Fourier Neural Operator (PFNO), which enables discretization invariance and parametric dependence, or the original Fourier Neural Operator (FNO) if the governing equation lacks external parameters (architecture details in Appendix A).

The blocks are subjected to an autoregressive sequence-to-sequence training to learn the next time step prediction for a fundamental operator O_i. We define the pretraining task by the canonical form:

$$\frac{\partial \mathbf{u}}{\partial t} + \gamma \cdot O_i(\mathbf{u}) = 0, \tag{6}$$

This ensures that each foundation block $\hat{O}_i$ develops an exclusive representation of a unique elementary operator O_i.

Once pretraining is complete, the weights of these blocks are frozen, and the final projection layer of each i-th foundation block ($\hat{O}_i$) is removed to supply a high-dimensional embedded representation

of the solution ϵ_i, defined as:

$$\epsilon_i = \hat{O}_i^{emb}(\mathbf{u}_t, \gamma) \in \mathbb{R}^{d_{emb}} \tag{7}$$

where d_{emb} is the embedding dimension. This representation ϵ_i acts as a physical dictionary that captures structural patterns, spatial gradients, and parameter dependencies. By supplying these enriched embeddings rather than raw values, the foundation blocks provide the aggregator with a more informative basis for learning complex multi-physics interactions.

3.4 ADAPTATION BLOCKS

In our architecture illustrated in Figure 1, the Adaptation Block is the task-specific component of the framework designed to assemble the foundation experts into a unified solver for the target PDE. While the Foundation Blocks capture the physical behaviors of the operators separately, the Aggregator $\mathcal{A}$ learns the interactions, couplings, and nonlinearities unique to the system being solved.

The Aggregator $\mathcal{A}$ is a learnable multilayer perceptron (MLP) that processes the concatenated high-dimensional latent embeddings ϵ_i from the selected Foundation Blocks of each rollout step. We model the one-step incremental prediction as:

$$\delta\hat{\mathbf{u}}_t = \mathcal{A}(\epsilon_1 \oplus \cdots \oplus \epsilon_{opt}) \tag{8}$$

where $\oplus$ denotes the concatenation operator across the feature dimension. By operating in this enriched latent space rather than on raw physical values, the aggregator can exploit the pre-learned structural and gradient information encoded within each ϵ_i. The selection of specific foundation operators is guided by the mathematical structure of the target PDE. In practice, we manually assemble these blocks based on the structure of the target equations.

Mathematically, we view the target PDE as a composite operator O_{target}. The Adaptation Block effectively learns a mapping that approximates this composition:

$$O_{target} = \sum_{i=1}^{opt} \alpha_i \hat{O}_i^{emb} \tag{9}$$

where $\hat{O}_i^{emb}$ are foundation operators involved in the target physics and α_i are the sets of parameters of the aggregator neural network.

The training of the Aggregator A is driven by an optimization criterion designed to bridge the gap between individual operator dynamics and the full coupled system. The objective function of CompNO is a dual-term criterion:

$$\mathcal{J}(\alpha) = \mathcal{C}_{FB} + \mathcal{C}_{cross} \tag{10}$$

where:

- $\mathcal{C}_{FB}$ represents the direct approximation of the elementary operators within target PDE learned by the Foundation Blocks. In the CompNO framework, this term is assumed to be near-zero, as the core physical knowledge of the constituent operators is accurately captured and transferred by the frozen Foundation Blocks.
- $\mathcal{C}_{cross}$ is the interaction term, representing the inner coupling and nonlinear interactions between the operators. Mathematically, this can be viewed as the closure law of the system. While the Foundation Blocks provide the latent ϵ_i, the Aggregator must learn how these fields interfere, compete, or amplify each other. (More details in Appendix C)

To ensure the modeled interaction $\mathcal{C}_{cross}$ remains physically grounded, the aggregation is guided by a Physics-Informed loss function; The aggregator is trained to minimize the residual R derived from the target PDE as well as the data loss, balanced by coefficients $\lambda_{physics}$ and λ_{data}. The loss function can be defined as shown in the Figure 1:

$$\mathcal{L} = \lambda_{data} \cdot \mathcal{L}_{data} + \lambda_{physics} \cdot R(\hat{\mathbf{U}}) \tag{11}$$

with $\mathcal{L}_{data}$ is the Mean Absolute Error (MAE) loss function, defined as:

$$MAE = \frac{1}{T}\sum_{t\leq T}|u_t - \hat{u}_t|. \tag{12}$$

where T is the final time, t represents the time step $\in [0, ..., T]$, u_t is the ground truth target, and $\hat{u}_t$ represents the model's prediction.

To solve the case of INS, the Aggregator $\mathcal{A}$ handles the coupling required to satisfy the incompressibility equation inspired by the classical Projection Method (Guermond et al., 2006), which decomposes the velocity update into a convection-diffusion step followed by a pressure-gradient correction needed to satisfy the divergence-free constraint:

$$\nabla \cdot (\hat{\mathbf{u}}_t + \delta\hat{\mathbf{u}}_t) = 0 \tag{13}$$

We provide the Aggregator with the latent embeddings from the relevant experts: the nonlinear convection embedding ϵ_{conv}, the diffusion embedding ϵ_{diff}, and the gradient of the pressure embedding $\nabla\epsilon_p$ derived from the Poisson Foundation Block. To maintain numerical consistency with the underlying physics, we apply the gradient operator ∇ on the embedded features ϵ_p using central differences. The Aggregator then approximates the following multi-stage physical process:

- Intermediate State Construction: The foundation blocks $\hat{\mathcal{O}}_{conv}$ and $\hat{\mathcal{O}}_{diff}$ provide the latent representation of the intermediate velocity $\mathbf{u}^*$, which accounts for advection and viscosity but is not necessarily divergence-free:

$$\mathbf{u}^* \approx \mathbf{u}_t + \mathcal{A}_{sub}(\epsilon_{conv} \oplus \epsilon_{diff}) \tag{14}$$

- Pressure-Gradient Correction: To satisfy the incompressibility constraint, the Aggregator utilizes the pressure information ϵ_p from the Poisson block. In classical CFD, the update follows $\mathbf{u}_{t+1} = \mathbf{u}^* - \frac{\Delta t}{\rho}\nabla p$. In our framework, the Aggregator learns to perform this projection in the latent space:

$$\delta\hat{\mathbf{u}}_t = \mathcal{A}\left((\epsilon_{conv} \oplus \epsilon_{diff}) \oplus \nabla\epsilon_p\right) \tag{15}$$

By providing $\nabla\epsilon_p$ as a direct input, the Aggregator is equipped with the necessary geometric information to correct the velocity field. Instead of forcing a single network to learn the entire projection, CompNO routes the required physical components through the Aggregator, which then learns the closure law ($\mathcal{C}_{cross}$) that balances momentum and mass conservation.

4 RESULTS

In this section, we evaluate the performance of CompNO in two-dimensional domains, assessing its ability to reconstruct complex fluid dynamics from the knowledge transferred from elementary operators. We specifically investigate the framework's capability to maintain physical consistency quantified via the physics residual loss compared to the monolithic PFNO baseline. All datasets utilized for training and evaluation were generated using high-fidelity numerical methods to serve as the ground-truth physics; for a comprehensive description of data generation methods, the reader is referred to Appendix B.

4.1 PRETRAINING PERFORMANCE

The first stage of the proposed framework involves the independent pretraining of the foundation library. To ensure the robustness of the experts, the Foundation Blocks were trained on a diverse set of parameters and initial conditions, including the Taylor–Green Vortex, Shear Layer, and Isotropic Turbulence. The Foundation Blocks were constructed using four Fourier integral operator layers with the GELU activation function. The embedding dimension used for these experiments is $d_{emb} = 128$. These blocks are trained using only the MAE loss function.

For the current experiments, we constructed a library of four Foundation Blocks:

- Convection Block: Trained on the linear convection equation with the parameter β taking values from the set $\{0.5, 1.0, 1.5, 2.0, 2.5\}$.
- Diffusion Block: Trained on the diffusion equation where the parameter ν is sampled from the set $\{1, 0.1, 0.01, 0.001\}$.
- Nonlinear Convection Block: Trained on the inviscid Burgers' equation to capture nonlinear convection dynamics.
- Poisson Block: Trained on the pressure-Poisson equation to provide the pressure field required for incompressible flows.

Table 1 reports the MAE values obtained during the pretraining phase. The dataset is split into 80% of (128×128) grid and 20% of finer resolution with (256×256) grid.

Table 1: MAE losses for different pretrained models.

Model	Scalar Training Loss	Vectorial Training Loss
PFNO Convection	1×10^{-5}	-
PFNO Diffusion	2×10^{-5}	8×10^{-5}
FNO NL Convection	5×10^{-5}	2×10^{-4}
FNO Poisson	-	2×10^{-5}

As shown in Table 1, our pretrained blocks achieve a great performance minimizing the MAE in a good way. The low error rates across both scalar and vectorial operators confirm that our Foundation Blocks successfully learn the underlying operators in isolation, providing a rich latent basis for subsequent aggregation.

4.2 Scalar Aggregation

The scalar track evaluates the model on coupled 2D Convection-Diffusion and Viscous Burgers'. For these experiments, the models are provided with an initial state $T_0 = 1$ and tasked with an autoregressive rollout of $T = 30$ time steps.

Table 2: Temporal evolution training MAE of loss for different metrics.

PDE	Pe / Re	Model	Trainable Parameters	MAE Velocity Loss	Physics Loss
Convection-Diffusion	$Pe \in [1, 250]$	CompNO	41 K	3×10^{-4}	0.05
		PFNO	465 K	9×10^{-3}	24
Burgers'	$Re \in [1, 100]$	CompNO	75 K	2×10^{-3}	0.6
		PFNO	465 K	0.019	19.94

As summarized in Table 2, CompNO consistently outperforms the monolithic PFNO when trained on the exact same amount of data. A critical observation is the Physics Loss discrepancy; while PFNO achieves reasonable velocity accuracy, its inability to strictly respect the PDE constraints leads to a significantly higher residual error. In contrast, our modular approach preserves the learned dynamics of the frozen experts through Physics-Informed Aggregation, ensuring a physically consistent temporal evolution that adheres more closely to the governing equations.

4.3 Vectorial Aggregation

The vectorial track addresses the bi-variable version of the Viscous Burgers' equation and the Incompressible Navier-Stokes equations, where the model must handle coupled velocity components (u,v) with the pressure for the INS case. The models are provided with a sequence $T_0 = 10$ and perform a rollout of $T = 50$ time steps.

Table 3: Temporal evolution training loss for different metrics.

PDE	Re	Model	Trainable Parameters	MAE	
				Velocity Loss	Pressure Loss
Burgers'	$Re \in [1, 100]$	CompNO	75 K	4×10^{-3}	-
		PFNO	466 K	8×10^{-3}	-
Incomp Navier-Stokes	$Re \in [1, 100]$	CompNO	140 K	4×10^{-3}	6×10^{-3}
		PFNO	466 K	0.01	0.012

Table 3 highlights the clear advantage of CompNO over the PFNO when constrained to an identical data budget. For the Burgers' equation, CompNO achieves better accuracy in terms of velocity vector prediction. However for the INS test case, the superiority of our model is more clear in both velocity and pressure predictions. Notably, this performance gain is achieved with a considerable reduction in trainable parameters, highlighting the parameter efficiency and superior scaling of our modular, physics-informed architecture compared to traditional monolithic operators.

4.4 Inference Speed

A primary motivation for utilizing Scientific Foundation Models is the reduction of computational overhead associated with traditional numerical solvers. Table 4 compares the time required to generate a single time step per trajectory across different physical tracks. Traditional simulations were performed on a uniform 128×128 Cartesian grid using the numerical schemes detailed in Appendix B. Benchmarking was conducted by comparing the traditional solver's execution on an AMD EPYC 7502P 32-Core Processor against the CompNO's inference on a single NVIDIA A100.

Table 4: Computational Time Comparison (Per time step)

	Convection-Diffusion	Scalar Burgers'	Vectorial Burgers'	INS
Numerical Solver	0.064 s	0.1 s	0.22 s	0.35 s
CompNO	0.0052 s	0.0052 s	0.0065 s	0.01 s
Speed-up	$\times$ 12	$\times$ 19	$\times$ 33	$\times$ 35

For the scalar track, CompNO achieves a speed-up of 12$\times$ for Convection-Diffusion and 19$\times$ for the Viscous Burgers' equation. The efficiency gains are even more pronounced in the Vectorial Track, where CompNO achieves speed-ups of 33$\times$ for Vectorial Burgers' and 35$\times$ for the INS equations. The INS equations necessitate a multi-step Projection Method, involving the calculation of an intermediate velocity and the iterative solution of a global Pressure Poisson Equation. In contrast, CompNO bypasses these iterations through its modular architecture; by utilizing the pretrained Poisson Foundation Block and a single forward pass of the Physics-Informed Aggregator, the model enforces incompressibility and momentum transfer simultaneously. These results are effectively decoupled from the iterative complexity of the underlying physics.

4.5 Discussion

The evaluation of CompNO across the Convection-Diffusion, Burgers', and Incompressible Navier-Stokes equations showed promising results for the framework. The model acts as an aggregator that profits from the specialized knowledge transferred by the pretrained Foundation Blocks to solve new tasks. This confirms that the "Pre-train, Assemble, and Fine-tune" paradigm is a promising path for building scalable Scientific Foundation Models.

A critical observation in this 2D extension is the behavior of temporal error accumulation during autoregressive rollouts. As shown in Figure 2, the error increases throughout the prediction sequence but remains within the same order of magnitude. This indicates that the model is numerically stable, as the error grows slowly and linearly, rather than exhibiting the exponential divergence as is often seen in monolithic solvers. By decomposing the system into modular pieces, the framework ensures

that fundamental mechanisms are handled by experts who already understand those dynamics. This structural integrity prevents small temporal errors from collapsing into non-physical noise, keeping the overall solution physically valid even at the end of the rollout process.

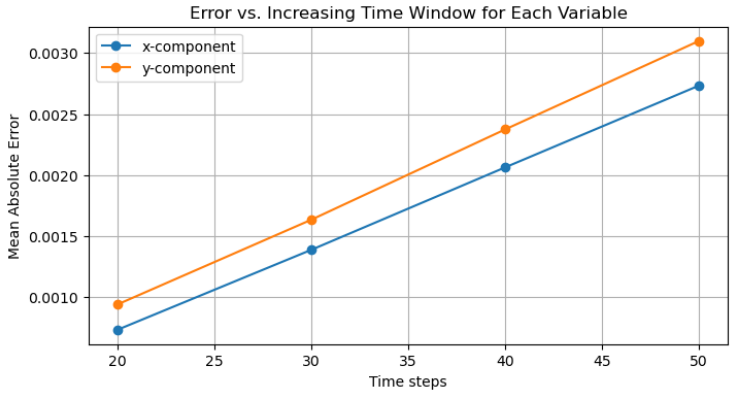

Figure 2: Comparative MAE for INS variables during the inference phase (T=50).

5 Conclusion

In this work, we have extended the Compositional Neural Operators framework to two-dimensional systems, demonstrating a scalable and physically consistent approach to building Scientific Foundation Models. By leveraging the principle of compositionality, we decomposed complex fluid dynamics into a library of Foundation Blocks, each expert in a fundamental physical mechanism such as convection, diffusion, or nonlinear momentum transfer. These modules serve as a reusable base that can be assembled into task-specific solvers through an Adaptation Block. This ”pre-training assemble fine-tune” paradigm improves data efficiency, enhances interpretability, and enables compositional generalization to coupled physics not seen during pretraining.

CompNO demonstrates good performance on two-dimensional fluid dynamics systems such as convection–diffusion, Burgers', and INS equations. These results indicate that our approach is a promising way towards a scalable, physically consistent, and reusable foundation models. While the current operator library focuses on fluid flow equations, the proposed framework is inherently extensible. Additional physical dynamics can be incorporated by training new elementary operators, allowing the system to address new classes of problems without retraining the existing experts. Future work will concentrate on testing the model on more challenging configurations and geometries, and enriching the operator library to cover a broader range of PDEs.

References

Saakaar Bhatnagar, Yaser Afshar, Shaowu Pan, Karthik Duraisamy, and Shailendra Kaushik. Prediction of aerodynamic flow fields using convolutional neural networks. *Computational Mechanics*, 64(2):525–545, 2019. ISSN 1432-0924.

Yadi Cao, Yuxuan Liu, Liu Yang, Rose Yu, Hayden Schaeffer, and Stanley Osher. Vicon: Vision in-context operator networks for multi-physics fluid dynamics prediction. *arXiv preprint arXiv:2411.16063*, 2024.

Youngsoo Choi, Siu Wun Cheung, Youngkyu Kim, Ping-Hsuan Tsai, Alejandro N. Diaz, Ivan Zanardi, Seung Whan Chung, Dylan Matthew Copeland, Coleman Kendrick, William Anderson, Traian Iliescu, and Matthias Heinkenschloss. Defining foundation models for computational science: A call for clarity and rigor, 2025.

J.L. Guermond, P. Minev, and Jie Shen. An overview of projection methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 195(44):6011–6045, 2006. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2005.10.010.

Maximilian Herde, Bogdan Raonic, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bezenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for PDEs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Hamda Hmida, Hsiu-Wen Chang Joly, and Youssef Mesri. Compno: A novel foundation model approach for solving partial differential equations. *Applied Sciences*, 16(2), 2026. ISSN 2076-3417. doi: 10.3390/app16020972.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

Zongyi Li, Nikola Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, et al. Geometry-informed neural operator for large-scale 3d pdes. *Advances in Neural Information Processing Systems*, 36:35836–35854, 2023.

Yuxuan Liu, Jingmin Sun, Xinjie He, Griffin Pinney, Zecheng Zhang, and Hayden Schaeffer. PROSE-FD: A multimodal pde foundation model for learning multiple operators for forecasting fluid dynamics. *arXiv preprint arXiv:2409.09811*, 2024a.

Yuxuan Liu, Zecheng Zhang, and Hayden Schaeffer. PROSE: Predicting multiple operators and symbolic expressions using multimodal transformers. *Neural Networks*, 180:106707, 2024b.

Yuxuan Liu, Jingmin Sun, and Hayden Schaeffer. Bcat: A block causal transformer for pde foundation models for fluid dynamics, 2025. URL https://arxiv.org/abs/2501.18972.

Lu Lu, Pengzhan Jin, and George Em Karniadakis. DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021. ISSN 2522-5839. arXiv:1910.03193.

Chunyue Lv, Lei Wang, and Chenming Xie. A hybrid physics-informed neural network for nonlinear partial differential equation. *International Journal of Modern Physics C*, 34(06):2350082, 2023.

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

Sidharth S. Menon, Trishit Mondal, Shuvayan Brahmachary, Aniruddha Panda, Subodh M. Joshi, Kaushic Kalyanaraman, and Ameya D. Jagtap. On scientific foundation models: Rigorous definitions, key applications, and a comprehensive survey. *Neural Networks*, 198:108567, 2026. ISSN 0893-6080.

Tung Nguyen, Arsh Koneru, Shufan Li, and Aditya Grover. Physix: A foundation model for physics simulations, 2025. URL https://arxiv.org/abs/2506.17774.

Ugo Pelissier, Augustin Parret-Fréaud, Felipe Bordeu, and Youssef Mesri. Graph neural networks for mesh generation and adaptation in structural and fluid mechanics. *Mathematics*, 12(18):2933, 2024.

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

Junhong Shen, Tanya Marwah, and Ameet Talwalkar. UPS: Efficiently building foundation models for PDE solving via cross-modal adaptation. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=0r9mhjRv1E.

Jamie M. Taylor, David Pardo, and Ignacio Muga. A deep fourier residual method for solving pdes using neural networks. *Computer Methods in Applied Mechanics and Engineering*, 405:115850, 2023. ISSN 0045-7825.

Amin Totounferoush, Serge Kotchourko, Michael W. Mahoney, and Steffen Staab. Paving the way for scientific foundation models: enhancing generalization and robustness in pdes with constraint-aware pre-training, 2025. URL https://arxiv.org/abs/2503.19081.

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

Yizheng Wang, Jia Sun, Jinshuai Bai, Cosmin Anitescu, Mohammad Sadegh Eshaghi, Xiaoying Zhuang, Timon Rabczuk, and Yinghua Liu. Kolmogorov–arnold-informed neural network: A physics-informed deep learning framework for solving forward and inverse problems based on kolmogorov–arnold networks. *Computer Methods in Applied Mechanics and Engineering*, 433: 117518, 2025. ISSN 0045-7825.

Liu Yang, Siting Liu, and Stanley J Osher. Fine-tune language models as multi-modal differential equation solvers. *Neural Networks*, pp. 107455, 2025.

Yuekun Yang and Youssef Mesri. Learning by neural networks under physical constraints for simulation in fluid mechanics. *Computers & Fluids*, 248:105632, 2022.

Rixin Yu and Erdzan Hodzic. Parametric learning of time-advancement operators for unstable flame evolution. *Physics of Fluids*, 36(4), 2024.

Yinhao Zhu and Nicholas Zabaras. Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. *Journal of Computational Physics*, 366:415–447, 2018. ISSN 0021-9991.

A Architecture of Fourier Neural Operators

A.1 The Fourier Neural Operator (FNO)

The Fourier Neural Operator (FNO) (Li et al., 2020) learns a mapping between functional spaces by parameterizing the integral kernel in the Fourier domain. Given an input $f(x)$, the FNO architecture consists of an initial lifting layer P that maps the input to a high-dimensional representation ε_0, followed by L Fourier layers, and a final projection layer Q that maps the hidden representation to the output.

The core of the architecture is the Fourier Layer. For a hidden representation ε_ℓ at layer ℓ, the update is defined as:

$$\varepsilon_{\ell+1} = \sigma\left(\mathcal{F}^{-1}\left\{\mathfrak{R}_\ell\left(\mathcal{F}\{\varepsilon_\ell\}\right)\right\} + W_\ell(\varepsilon_\ell)\right), \tag{16}$$

where $\epsilon_\ell \in \mathbb{R}^{d_h}$ denotes the feature embedding at layer l, σ is a nonlinear activation function, $\mathcal{F}$ and $\mathcal{F}^{-1}$ are the Fast Fourier Transform and the inverse one, W_ℓ performs a channel-wise linear transformation, and $\mathfrak{R}_\ell$ is a linear transformation function defined as:

$$\mathfrak{R}_\ell(\mathcal{F}\{\varepsilon\})_{\kappa,i} = \sum_{j=1}^{d_h}(R_\ell)_{\kappa,i,j}\mathcal{F}\{\varepsilon\}_{\kappa,j}, \quad \kappa = 1,\ldots,\kappa^{\max} \quad \text{and} \quad i = 1,\ldots,d_h, \tag{17}$$

with R_ℓ is a trainable weight tensor and $\kappa^{\max}$ is the maximum number of frequency modes.

A.2 The Parametric Fourier Neural Operator (PFNO)

While the standard FNO learns a global operator for a fixed physical system, the Parametric FNO (PFNO) (Yu & Hodzic, 2024) generalizes across varying system parameters γ (e.g., varying viscosity ν or velocity β).

As showed in Figure 3, PFNO incorporates parameter influence by appending each parameter value $\gamma \in \mathbb{R}^p$ to the codomain of the input function $f(x) \in \mathbb{R}^d$. The modified function $f'(x) \in \mathbb{R}^{d+p}$ is then projected to a higher dimension using the same lifting operator P.

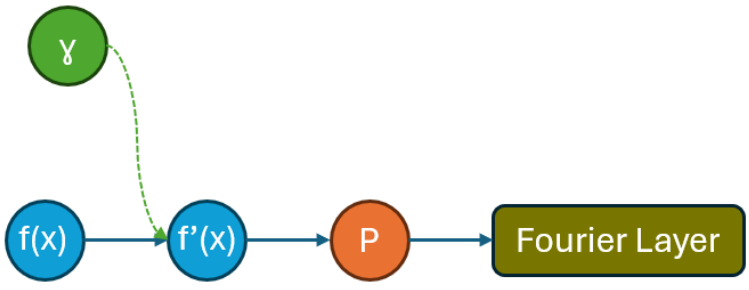

Figure 3: Architecture of the PFNO. Illustration of the parameter-dependent input transformation.

B Data Generation and Numerical Schemes

This appendix provides the technical details of the numerical methodologies used to generate the 2D datasets. All simulations were performed on uniform Cartesian grids with periodic boundary conditions, utilizing specific discretization schemes to ensure stability and physical accuracy.

B.1 Elementary Operators (Foundation Blocks)

The Foundation Blocks are trained on isolated physical mechanisms to learn disentangled representations of convection, diffusion, and nonlinear interactions.

B.1.1 Scalar Foundation Blocks

The **Scalar Velocity Track** library contains expert modules specialized in the fundamental transport mechanisms of a scalar field $u(x, y, t)$. Each block is pretrained on the following governing equations:

- **Linear Convection Block:** Captures the transport of u in a constant velocity field β:

$$\frac{\partial u}{\partial t} + \beta \frac{\partial u}{\partial x} + \beta \frac{\partial u}{\partial y} = 0 \tag{18}$$

- **Diffusion Block:** Learns the isotropic spreading of u governed by the Laplacian operator and diffusion coefficient ν:

$$\frac{\partial u}{\partial t} = \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \tag{19}$$

- **Nonlinear Convection Block:** Derived from the 2D inviscid Burgers' equation, this block models self-advection where the field u acts as its own transport velocity:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + u \frac{\partial u}{\partial y} = 0 \tag{20}$$

B.1.2 Vectorial Foundation Blocks

The **Vectorial Velocity Track** addresses the velocity vector field $\mathbf{u} = (u, v)$. These blocks are essential for assembling the Incompressible Navier-Stokes equations, with component-wise dynamics defined as:

- **Vectorial Diffusion:** Applies the Laplacian operator independently to each velocity component:

$$\begin{cases} \frac{\partial u}{\partial t} = \nu \nabla^2 u \\ \frac{\partial v}{\partial t} = \nu \nabla^2 v \end{cases} \tag{21}$$

- **Nonlinear Vector Convection:** Captures the momentum transfer where the vector field $\mathbf{u}$ advects itself:

$$\begin{cases} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = 0 \\ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = 0 \end{cases} \tag{22}$$

- **Poisson Foundation Block:** Solves the elliptic pressure-Poisson equation to relate the scalar pressure field p to the velocity divergence source term f:

$$\nabla^2 p = \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} = f \tag{23}$$

B.2 Coupled Physical Systems (Adaptation Blocks)

The Adaptation Blocks are trained on complex systems where multiple elementary operators interact simultaneously.

B.2.1 2D Convection-Diffusion

This system couples linear transport with molecular dissipation:

$$\frac{\partial u}{\partial t} + \beta \cdot \nabla u = \nu \nabla^2 u \tag{24}$$

We implement an **IMEX (Implicit-Explicit) scheme**. Convection is treated explicitly using a first-order upwind scheme for stability, while the diffusion term is treated implicitly via Crank-Nicolson. This hybrid approach ensures stability across a wide range of Péclet numbers (Pe) defined as:

$$\mathrm{Pe} = \frac{\beta \text{ x L}}{\nu}$$

where L is the characteristic length of the domain.

B.2.2 2D VISCOUS BURGERS' EQUATION

The viscous Burgers' equation introduces coupling between nonlinear convection and diffusion:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = \nu \nabla^2 \mathbf{u} \tag{25}$$

Data generation utilizes an **IMEX-RK3** approach. Nonlinear convection terms are computed explicitly via upwind discretization to prevent numerical oscillations, while viscous terms are handled implicitly. This equation is characterized by The Reynolds number (Re) defined as:

$$\mathrm{Re} = \frac{U \text{ x L}}{\nu}, \tag{26}$$

where L is the characteristic length of the domain and U is the characteristic value of $\mathbf{u}$. This dimensionless parameter help predict the behavior of fluids in the Burgers' equation as well as the Navier-Stokes equations.

B.2.3 2D INCOMPRESSIBLE NAVIER-STOKES

The most complex coupled system involves momentum transport and the divergence-free constraint:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u}, \quad \nabla \cdot \mathbf{u} = 0 \tag{27}$$

We implement a **Chorin-type Projection Method** consisting of three steps:

1. **Tentative Velocity:** A velocity $\mathbf{u}^*$ is computed by solving the momentum equation (excluding pressure) via IMEX.
2. **Pressure Poisson:** The Pressure Poisson Equation (PPE), $\nabla^2 p = \frac{\rho}{\Delta t}\nabla \cdot \mathbf{u}^*$, is solved to find the pressure field.
3. **Correction:** The final velocity is updated via $\mathbf{u}_{t+1} = \mathbf{u}^* - \frac{\Delta t}{\rho}\nabla p$.

C CLOSURE LAW

- Convection-diffusion: Mathematically, let's denote u the solution of the convection equation and v the solution of diffusion equation. The aggregated output $w = \alpha_1 u + \alpha_2 v$ can be a solution of the convection-diffusion equation, where α_1 and α_2 represent mixing coefficients. By substituting the aggregated solution into the target equation:

$$\frac{\partial(\alpha_1 u + \alpha_2 v)}{\partial t} + \beta \cdot \nabla(\alpha_1 u + \alpha_2 v) - \nu \cdot \Delta(\alpha_1 u + \alpha_2 v) = 0, \tag{28}$$

  that refers to :

$$\alpha_1(\frac{\partial u}{\partial t} + \beta \cdot \nabla u) + \alpha_2(\frac{\partial v}{\partial t} - \nu \cdot \Delta v) + \alpha_2 \beta \Delta v - \alpha_1 \nu \nabla u = 0. \tag{29}$$

  We define $\mathcal{C}_{CD}$ as the residual of the convection-diffusion elementary operators:

$$\mathcal{C}_{CD} = \alpha_1(\frac{\partial u}{\partial t} + \beta \cdot \nabla u) + \alpha_2(\frac{\partial v}{\partial t} - \nu \cdot \Delta v) \approx 0 \tag{30}$$

  and by simplifying $\mathcal{C}_{CD}$ as the knowledge transferred by the foundation blocks (equal to zero), we define close law of the convection-diffusion equation $\mathcal{C}_{cross}$ as:

$$\mathcal{C}_{cross} = \alpha_2 \beta \nabla v - \alpha_1 \nu \Delta u \tag{31}$$

- Burgers' equation: Mathematically, let's denote u as the solution of the inviscid Burgers' equation and v as the solution from the diffusion equation. By substituting a point-wise linearized aggregation $w = \alpha_1 u + \alpha_2 v$ into the viscous Burgers' equation we obtain:

$$\frac{\partial(\alpha_1 u + \alpha_2 v)}{\partial t} + (\alpha_1 u + \alpha_2 v) \cdot \nabla(\alpha_1 u + \alpha_2 v) - \nu \cdot \Delta(\alpha_1 u + \alpha_2 v) = 0, \tag{32}$$

that refers to :

$$\mathcal{C}_B + \alpha_1^2 u \cdot \nabla u + \alpha_2^2 v \cdot \nabla v + \alpha_1 \alpha_2 (v \cdot \nabla u + u \cdot \nabla v) - \alpha_1 \nu \cdot \Delta u = 0, \tag{33}$$

where $\mathcal{C}_B$ is the residual of the Burgers' equation elementary operators:

$$\mathcal{C}_B = \alpha_1 (\frac{\partial u}{\partial t} + u \cdot \nabla u) + \alpha_2 (\frac{\partial v}{\partial t} - \nu \cdot \Delta v) \approx 0, \tag{34}$$

By simplifying $\mathcal{C}_B$, we define close law of the Burgers' equation $\mathcal{C}_{cross}$ as: :

$$\mathcal{C}_{cross} = \alpha_1^2 u \cdot \nabla u + \alpha_2^2 v \cdot \nabla v + \alpha_1 \alpha_2 (v \cdot \nabla u + u \cdot \nabla v) - \alpha_1 \nu \cdot \Delta u \tag{35}$$