SYMONET: EXPRESSION-DRIVEN FOUNDATION MODEL FOR MULTI-OPERATOR LEARNING WITH DATA-FREE FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Operator foundation models have advanced PDE solving through large-scale pre-training, enabling shared representations and transfer across physical systems. However, training and adapting these models typically require substantial high-fidelity solution data, which are expensive to obtain through numerical simulations or observations. To reduce this dependence during downstream adaptation, we introduce the Symbolic expression-driven Operator network foundation model (SymOnet) for data-free fine-tuning guided by governing equations, initial conditions, and boundary conditions. SymOnet learns a shared operator representation across heterogeneous PDE families through a token-query architecture that maps discretized input functions and space–time coordinates to solution values. Multi-PDE pre-training builds this representation, while downstream adaptation optimizes only lightweight adapters using expression-driven losses evaluated through automatic differentiation, keeping the shared backbone frozen. We establish a family-wise approximation theorem showing that a shared query basis can support operator approximation across a family of PDEs. At comparable model sizes, SymOnet outperforms all evaluated baselines on all 12 tasks under expression-driven fine-tuning, while also attaining the lowest L2RE on 9 tasks under data-driven fine-tuning. Additional experiments on three-dimensional compressible Navier–Stokes dynamics demonstrate transfer from two-dimensional pre-training to both fine-tuning regimes.

1 INTRODUCTION

Partial differential equations (PDEs) are central to scientific computing and beyond, yet repeatedly solving them from scratch remains expensive. Neural PDE solvers (Dissanayake & Phan-Thien, 1994; Chen & Chen, 1995) have been developed to address these challenges. Classical trial-function methods such as NNA-PDE (Dissanayake & Phan-Thien, 1994) and later physics-informed models such as PINNs (Raissi et al., 2019) and HFM (Raissi et al., 2020) embed governing equations directly into the objective and can be trained without high-fidelity solution data. However, they are typically optimized separately for each problem instance with relatively slow convergence and transfer poorly across PDE families. Operator learning addresses this limitation by learning mappings between function spaces. DeepONet (Lu et al., 2021), PI-DeepONet (Wang et al., 2021), FNO (Li et al., 2021), and PINO (Li et al., 2024) demonstrate that neural operators can generalize across input functions and deliver fast inference.

To improve generalization across varying coefficients, forcings, and initial or boundary conditions, attention-based neural operators such as the Galerkin Transformer (Cao, 2021), OFormer (Li et al., 2023), GNOT (Hao et al., 2023), and Transolver (Wu et al., 2024) model PDE operators on regular and irregular domains. Recent foundation-style models such as DPOT (Hao et al., 2024) and MoE-POT (Wang et al., 2025) further scale attention backbones to broad multi-operator pre-training, while parameter-efficient methods such as F-Adapter (Zhang et al., 2025) strengthen this direction (Hao et al., 2024; Zhang et al., 2025; Wang et al., 2025; Zhou et al., 2024). These methods reuse representations across PDEs, but their downstream adaptation generally depends on high-fidelity solution data. Physics-informed operator learning has also progressed toward multi-operator pre-training and data-free adaptation, including the LoRA-based fine-tuning of PI-DeepONet FM (Zhang et al., 2026).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

PIFT (Medvedev et al., 2026) further demonstrates physics-informed adaptation of a pre-trained Transformer operator through full-model updates and finite-difference residuals. These advances motivate a more specific question: how can a unified operator representation support adaptation across heterogeneous PDEs through a small set of task-specific parameters and differentiable governing-equation constraints? Throughout, *data-free* means that downstream adaptation uses no high-fidelity solution data from numerical simulations or observations; the governing equation and prescribed initial and boundary conditions define the problem and provide the supervision. The initial-condition expression is part of this problem specification, and the discrete model input is obtained by evaluating it directly at sensor coordinates, without numerical simulations or observations.

We propose the *Symbolic expression-driven Operator network foundation model* (SymOnet), combining multi-PDE pre-training with expression-driven adapter fine-tuning for this setting. SymOnet jointly pre-trains a shared input–query representation and places nonlinear residual adapters in the Patch Embedding, Input Encoder, and Fusion Block, allowing downstream constraints to adjust input features and their fusion with a fixed coordinate-query representation. The resulting space–time solution can be differentiated with respect to query coordinates to drive adaptation using governing equations and prescribed initial and boundary conditions. **The contributions of the work are as follows.**

- **Unified representation with expression-driven adaptation.** We jointly pre-train an operator representation across PDEs and adapt its input encoding and feature fusion through nonlinear residual modules, while keeping the query representation and backbone fixed and using no downstream high-fidelity solution data.
- **Family-wise shared-query approximation.** We prove that a common query basis with task-specific coefficient maps can uniformly approximate a compact family of solution operators, with an explicit separation of input discretization and representation errors. A Galerkin factorization connects this shared-query structure to SymOnet's attention mechanism.
- **Strong large-scale generalization.** Experiments across 12 tasks assess transfer to unseen parameter regimes, PDE families, geometries, and boundary configurations, while scaling studies and adapter ablations quantify the effects of pre-training and restricted downstream updates.

2 Related Work

We review the literature most relevant to SymOnet from two perspectives: physics-informed learning and foundation-style operator pre-training.

Physics-informed approaches from single-operator to multi-operator. Physics-informed learning reduces dependence on high-fidelity solution data by incorporating PDE constraints into training objectives (Dissanayake & Phan-Thien, 1994; Raissi et al., 2019; 2020). Operator-level variants such as PI-DeepONet (Wang et al., 2021) and PINO (Li et al., 2024) extend this principle from pointwise solutions to mappings between function spaces, while PICL (Lorsung & Barati Farimani, 2024) introduces equation-aware representation learning. MODNO (Zhang, 2024) and PI-DeepONet FM (Zhang et al., 2026) extend physics-informed learning to multiple operators. PI-DeepONet FM pre-trains a shared branch with operator-specific trunks and enables physics-informed LoRA adaptation while freezing the original weights. SymOnet jointly trains a common query representation across PDEs and adapts input features and feature fusion through nonlinear residual modules, preserving that query representation during downstream fine-tuning. PIFT (Medvedev et al., 2026) applies physics-informed fine-tuning to the pre-trained POSEIDON Transformer with all parameters trainable, evaluates PDE residuals by finite differences on structured output grids, and enforces Dirichlet boundaries through zero padding. This grid-based construction ties derivative accuracy to the output resolution, while its zero-padding treatment of boundaries does not directly support boundary conditions involving derivatives, such as Neumann conditions. SymOnet instead retains one shared non-adapter representation, updates only lightweight adapters, and evaluates the prescribed PDE residuals and boundary conditions through coordinate queries and automatic differentiation.

PITA (Zhu et al., 2025) targets temporal consistency in autoregressive prediction, whereas SymOnet targets adaptation to downstream PDEs through expression-driven supervision. Without assuming a known PDE, PITA uses finite differences and sparse regression to infer dynamics from observed trajectories and aligns equations inferred from observed and predicted temporal windows. SymOnet

instead uses the known governing equation and prescribed initial and boundary conditions, without downstream high-fidelity solution data.

Transformer-based foundation models. Attention-based operator architectures have become increasingly expressive and scalable. The Galerkin Transformer (Cao, 2021), OFormer (Li et al., 2023), WaveLiT (Sankaran et al., 2025), CoDA-NO (Rahman et al., 2024), ICON/ICON2 (Yang et al., 2023; Yang & Osher, 2024), GNOT (Hao et al., 2023), and Transolver (Wu et al., 2024) establish attention-based representations for operator learning. OFormer supports flexible evaluation at arbitrary spatial locations through attention-based coordinate queries. DPOT (Hao et al., 2024) and MoE-POT (Wang et al., 2025) scale this paradigm through broad pre-training on collections such as PDEBench (Takamoto et al., 2022) and CFDBench (Luo et al., 2023), while F-Adapter (Zhang et al., 2025) studies parameter-efficient transfer for operator foundation models. POSEIDON (Herde et al., 2024) learns from fluid-dynamics trajectories, WALRUS (McCabe et al., 2026) extends pre-training to heterogeneous 2D and 3D continuum systems, MORPH (Rautela et al., 2025) supports heterogeneous 1D–3D observations with full or LoRA fine-tuning, and PDE-FM (Soares et al., 2026) learns cross-physics representations from twelve datasets in The Well (Ohana et al., 2024). PDEformer-2 (Ye et al., 2025) encodes symbolic PDE graphs and supports coordinate queries, but relies on a 40-TB solution corpus and task-specific high-fidelity data for adaptation. Related directions include universal solvers (Zhou et al., 2025; Herde et al., 2024; Chen et al., 2025; Ye et al., 2025), multimodal and multi-operator formulations (Sun et al., 2025b;a; Negrini et al., 2025; Gupta & Brandstetter, 2023), transferable foundation models (Fan & Pang, 2025; Siddik et al., 2025), and pre-training or inference-time strategies (Subramanian et al., 2023; Chen et al., 2024; Totounferoush et al., 2025; Mansingh et al., 2025; Zhou et al., 2024). SymOnet complements these data-driven methods with expression-driven adapter fine-tuning, retaining a shared backbone while adapting to downstream PDEs without high-fidelity solution data.

3 METHODOLOGY

3.1 PROBLEM SETUP AND SYMONET OVERVIEW

Let $\mathcal{A} \subset C(D_a; \mathbb{R}^{c_{\text{in}}})$ and $\mathcal{U} \subset C(D_u; \mathbb{R}^{c_{\text{out}}})$ be the input- and solution-function spaces of a time-dependent PDE system on $D_a \cup D_u$, defined by $\mathcal{F}(u; a) = 0$ with solution operator $\mathcal{G}^* \colon \mathcal{A} \to \mathcal{U}$ such that $u = \mathcal{G}^*(a)$. SymOnet approximates $\mathcal{G}^*$ by a parametric operator $\mathcal{G}_\theta(a(X), X, Y)$ that maps the discrete representation $a(X)$ of an input function, together with its sensor coordinates X, to solution values at space–time query coordinates Y. In expression-driven fine-tuning, the input $a(X)$ is obtained by directly evaluating the prescribed initial-condition function a at sensor coordinates X, without using high-fidelity solution data.

The central design is to retain one query-conditioned operator representation across PDEs while localizing downstream changes to the adapters in the Patch Embedding, Input Encoder, and Fusion Block. This shared model supports data-driven pre-training and expression-driven downstream adaptation through the unified objective

$$\mathcal{L}(\theta;\, a, u) \;=\; \lambda_{\text{data}}\, \mathcal{L}_{\text{data}}(a, u) \;+\; \lambda_{\text{pde}}\, \mathcal{L}_{\text{pde}}(a), \tag{1}$$

Here, $\mathcal{L}_{\text{data}}$ measures the L_2 relative error over the prediction interval, while $\mathcal{L}_{\text{pde}}$ enforces the governing equations, initial conditions, and boundary conditions, with derivatives evaluated by automatic differentiation. Data-driven pre-training sets $\lambda_{\text{pde}} = 0$; expression-driven fine-tuning sets $\lambda_{\text{data}} = 0$.

3.2 SYMONET ARCHITECTURE

As illustrated in Figure 1, SymOnet comprises an Input Encoder, a Query Encoder, and a Fusion Block. Together, these components produce a solution representation that is differentiable with respect to space–time query coordinates, enabling PDE residuals to be evaluated by automatic differentiation.

1) Input Encoder The Input Encoder takes the discretized function $a(X)$ and its sensor coordinates X, with c_{in} denoting the number of input channels. SymOnet first applies a *convolutional patch embedding* (Dosovitskiy et al., 2021) to $a(X)$ and then processes the resulting tokens with Galerkin self-attention (Cao, 2021).

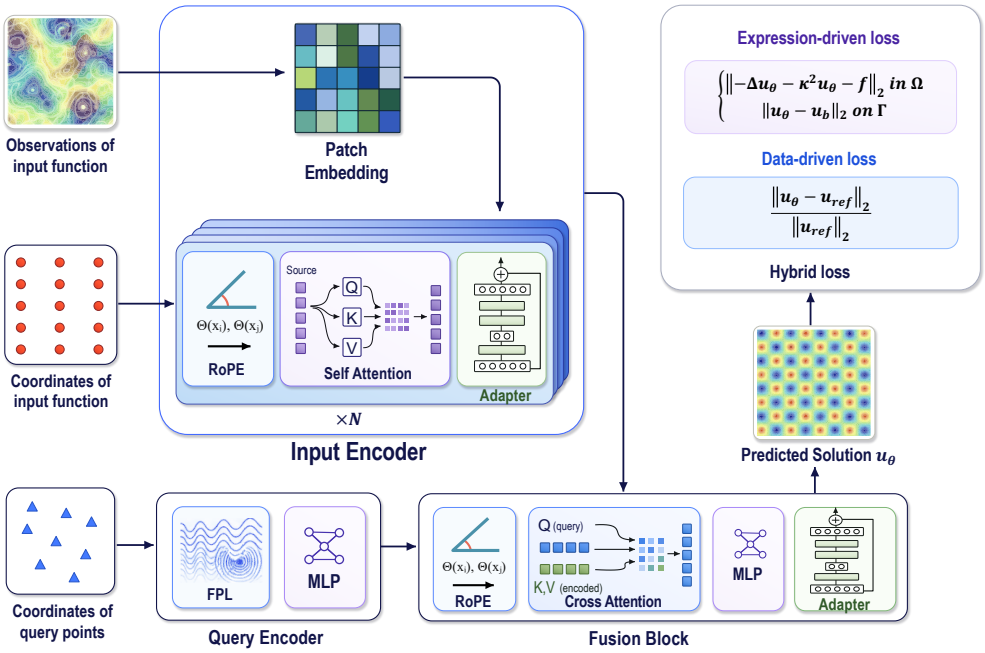

Figure 1: Overview of SymOnet. **Pre-training**: the Input Encoder and Query Encoder feed a cross-attention Fusion Block, trained with data-driven loss across multiple PDEs. **Data-free fine-tuning**: the backbone is frozen and adapters are updated using the governing equation and prescribed initial and boundary conditions, without downstream high-fidelity solution data.

The patch-embedding operator P_p applies a two-stage convolutional stack with patch size p along the spatial axes, reducing the spatial resolution from (h, w) to $(h^*, w^*) = (h/p, w/p)$. Flattening the resulting patch indices gives

$$Z^{(0)} \;=\; \mathsf{Flatten} \circ \mathsf{P}_p(a(X)) \;\in\; \mathbb{R}^{n_p \times d_{\mathrm{emb}}},$$

where n_p is the number of patch tokens. Using the corresponding patch-level coordinates pooled from X, Galerkin self-attention with Rotary Position Embedding (RoPE) (Su et al., 2024) produces

$$\mathrm{IE}(a(X), X) \;=\; \mathrm{SelfAttn}^{\mathrm{RoPE}}_{\mathrm{Gal}}\Big(Z^{(0)}, X\Big)\,. \tag{2}$$

Patch-level coordinates are obtained by stride-p spatial average-pooling of X, preserving the geometry of the sensor grid in the positional representation. This reduces attention cost while preserving local spatial structure and supporting later parameter-efficient fine-tuning.

2) Query Encoder Let $Y = \{(x_i, y_i, t_i)\}_{i=1}^{N}$ denote the query coordinate set. The Query Encoder maps each query coordinate to a feature vector via a Fourier Position Layer (FPL) (Tancik et al., 2020) followed by an MLP:

$$\mathrm{QE}(Y) \;=\; \mathrm{MLP}\big(\big[\cos(YB),\, \sin(YB)\big]\big) \;\in\; \mathbb{R}^{N \times d_{\mathrm{emb}}}, \tag{3}$$

where B is a Gaussian random projection matrix held fixed during training. The FPL enriches the query representation with high-frequency positional features and supports arbitrary query densities.

3) Fusion Block The Fusion Block combines the input-feature tokens $\mathrm{IE}(a(X), X)$ and the query features $\mathrm{QE}(Y)$ to produce the predicted solution.

Let W_Q, W_K, W_V be the query, key, and value projections of the cross-attention block. Galerkin cross-attention with RoPE produces

$$z(Y) \;=\; \mathrm{CrossAttn}^{\mathrm{RoPE}}_{\mathrm{Gal}}(W_Q\,\mathrm{QE}(Y),\; W_K\,\mathrm{IE}(a(X), X),\; W_V\,\mathrm{IE}(a(X), X))\,, \tag{4}$$

which is lifted by a linear map E and passed through a shared MLP,

$$\tilde{z}(Y) \;=\; \mathrm{MLP}_{\mathrm{shared}}(E\, z(Y))\,. \tag{5}$$

For channel-wise supervision, two disjoint linear projections W_{data} and W_{extra} map $\tilde{z}(Y)$ to the output channels with and without corresponding high-fidelity solution data, and their concatenation forms the full predicted solution:

$$u_\theta(Y) \;=\; \mathcal{G}_\theta(a(X), X, Y) \;=\; \mathrm{Concat}\big(W_{\mathrm{data}}\,\tilde{z}(Y),\; W_{\mathrm{extra}}\,\tilde{z}(Y)\big) \;\in\; \mathbb{R}^{N\times c_{\mathrm{out}}}. \tag{6}$$

Here, c_{out} denotes the number of solution functions predicted by SymOnet. The data-driven loss acts only on the channels associated with W_{data}, whereas the expression-driven loss constrains all channels.

4) Autoregressive Rollout Protocol Both pre-training and fine-tuning use *autoregressive rollouts*, feeding predictions back into the model for subsequent prediction intervals. Here, T_{pre} denotes the number of future time steps predicted by one forward pass, and T_{out} is the total rollout length. At rollout step n, $\hat{\mathbf{A}}^{(n)}$ denotes the current input representation and its sensor coordinates, initialized as $\hat{\mathbf{A}}^{(0)} = (a(X), X)$. The rollout horizon is $N = \left\lceil \frac{T_{\text{out}}}{T_{\text{pre}}} \right\rceil$. At step n, the Query Encoder takes the coordinate set $Y^{(n)}$ of the next T_{pre} time steps, and SymOnet predicts

$$u_\theta^{(n)} \;=\; \mathcal{G}_\theta\big(\hat{\mathbf{A}}^{(n)},\, Y^{(n)}\big), \qquad \hat{\mathbf{A}}^{(n+1)} \;=\; \mathsf{Roll}\big(\hat{\mathbf{A}}^{(n)},\, (u_\theta^{(n)}, Y^{(n)})\big), \qquad n = 0, 1, \ldots, N-1, \tag{7}$$

The operator Roll constructs the next input from the predicted solution and its coordinates. For data-driven runs, the configured input window is retained throughout the rollout, as described in Appendix E. The per-step losses are summed over the rollout and used for a single parameter update at the end. When the data-driven term is active, $u_{\text{ref}}^{(n)}$ denotes the ground-truth reference for the next T_{pre} time steps. For expression-driven fine-tuning, the most recent predicted state is fed into the model as the input for the next autoregressive step, and all supervision comes from the governing equation together with the prescribed initial and boundary conditions.

3.3 Multi-PDE Pre-training

SymOnet is pre-trained jointly on multiple datasets spanning diverse PDE families to learn a shared operator representation. Let $\mathcal{S}_k = \{(a_i^{(k)}, u_i^{(k)})\}$ denote the high-fidelity training dataset for PDE family $\mathcal{F}_k$. The pre-training loss is

$$\mathcal{L}_{\text{pretrain}} = \mathbb{E}_{k\sim[K]}\left[\mathrm{L2RE}\big(u_\theta(Y), u(Y)\big)\right], \tag{8}$$

where the L_2 relative error is computed separately for each trajectory, time step, and channel, and then averaged. The resulting shared representation is transferred to downstream PDEs through expression-driven adapter fine-tuning.

3.4 Data-Free Expression-driven Fine-tuning

Given a governing equation together with prescribed initial and boundary conditions, SymOnet freezes the backbone and optimizes the adapters with

$$\mathcal{L}_{\text{finetune}} = \mathbb{E}_a\left[\lambda_{\mathrm{ic}}\mathcal{L}_{\mathrm{ic}}^{(0)} + \sum_{n=0}^{N-1}\Big(\lambda_{\mathrm{eq}}\mathcal{L}_{\mathrm{eq}}^{(n)} + \lambda_{\mathrm{bc}}\mathcal{L}_{\mathrm{bc}}^{(n)}\Big)\right]. \tag{9}$$

Here, $\mathcal{L}_{\mathrm{eq}}^{(n)}$, $\mathcal{L}_{\mathrm{ic}}^{(0)}$, and $\mathcal{L}_{\mathrm{bc}}^{(n)}$ enforce the governing equation, initial condition, and boundary conditions, respectively. The equation and boundary losses are computed from the corresponding expressions at collocation points, with all required derivatives obtained by automatic differentiation. The initial-condition loss is evaluated on the initial surface, and subsequent intervals use predicted states without accessing high-fidelity solution data. Specifically, we realize this restricted update through the lightweight *adapter modules* (Houlsby et al., 2019). Each adapter in SymOnet is the residual map:

$$\mathsf{A}(h) \;=\; h \;+\; s\cdot P_{\mathrm{up}}\big(\sigma(P_{\mathrm{down}}(h))\big), \qquad s \in \mathbb{R}_{>0}, \tag{10}$$

Here, h is an intermediate feature, $\sigma = \mathrm{GELU}$, and P_{down} and P_{up} are learned affine projections with intermediate width r, referred to as the adapter bottleneck width. On token sequences, $(P_{\mathrm{down}}, P_{\mathrm{up}})$ become the standard linear bottleneck $(W_{\mathrm{down}}, W_{\mathrm{up}})$; on patch-level feature maps, they are realized as 1×1 convolutions $(K_{\mathrm{down}}, K_{\mathrm{up}})$ that preserve lattice translation equivariance. Following (Houlsby et al., 2019), SymOnet inserts these adapters in the Patch Embedding, in the self-attention blocks of the Input Encoder, and in the cross-attention block of the Fusion Block.

Every up-projection P_{up} is initialized to zero ($W_{\mathrm{up}} = \mathbf{0},\ K_{\mathrm{up}} = \mathbf{0}$), so each adapter is initially an identity, and the pre-trained operator $\mathcal{G}_{\theta_0}$ is recovered exactly at the start of fine-tuning. During data-free fine-tuning, expression-driven losses update these residual modules while the pre-trained backbone remains fixed.

3.5 Theoretical Analysis

We establish a family-wise approximation result for a shared-query operator representation, providing a theoretical basis for sharing coordinate features across PDEs while adapting input-dependent coefficients. The formal setting and notation are given in Appendix A.1.

For a PDE task τ in a compact family $\mathfrak{P}$, let $A = S_h a$ denote samples of the input function a at resolution h. At a space–time query $q \in Q$, consider $\widehat{\mathcal{G}}_\tau(A, q) = F_{\alpha_\tau}(A)^\top \psi_{\theta_0}(q)$, where F_{α_τ} is a task-specific GELU coefficient network and ψ_{θ_0} is a query basis shared across the PDE family. The query basis is constructed from Gaussian Fourier features as specified in Appendix A.2.

Assumption A.1 requires continuous input reconstruction with error at most e_h and a unique, jointly continuous solution map over a compact set $\mathcal{E}$ of admissible task–input pairs. Its input-continuity modulus ω controls the solution error induced by discretizing a.

Theorem 3.1 (**Family-wise shared-query approximation**). *Under Assumption A.1, for any approximation tolerance $\varepsilon > 0$ and failure probability $\delta \in (0, 1)$, there exist finite choices of Fourier-feature count, basis dimension, and coefficient-network width such that, with probability at least $1 - \delta$ over the frequency draw, shared parameters θ_0 and task-specific parameters α_τ satisfy*

$$\sup_{(\tau,a)\in\mathcal{E},\, q\in Q} \left\| \widehat{\mathcal{G}}_\tau(S_h a, q) - \mathcal{G}^*_\tau(a)(q) \right\|_\infty \le \omega(e_h) + \varepsilon. \tag{11}$$

Here, $\|\cdot\|_\infty$ is the maximum absolute error over output channels.

The proof is given in Appendix A.4. The bound separates input discretization error from representation error and shows that a common query basis can approximate a compact family of solution operators, with task dependence carried by the coefficient maps. For SymOnet's Galerkin decoder, each attention head factors into shared query features and key–value moments of the adapted input tokens, linking this representation to the architecture (Appendix A.5). Appendices A.6–A.8 analyze local adapter updates, linearized gradient descent, autoregressive error propagation, and residual-to-solution stability.

4 Experiments

We evaluate SymOnet in a large-scale multi-PDE setting using 10+ datasets from FNO (Li et al., 2021), PDEBench (Takamoto et al., 2022), and CFDBench (Luo et al., 2023). We compare SymOnet against DPOT (Hao et al., 2024), F-Adapter (Zhang et al., 2025), MoE-POT (Wang et al., 2025), POSEIDON (Herde et al., 2024), WALRUS (McCabe et al., 2026), and PI-DeepONet FM (Zhang et al., 2026). DPOT, MoE-POT, POSEIDON, and WALRUS predict fields on discrete grids without a differentiable coordinate-query interface, so they do not directly support the automatic differentiation with respect to query coordinates used by SymOnet to evaluate PDE residuals. We therefore adopt PIFT's finite-difference scheme (Medvedev et al., 2026) to evaluate partial derivatives during expression-driven fine-tuning and denote the resulting baselines with the "-PIFT" suffix. All results are reported as L2 relative error (L2RE); detailed PDE expressions are provided in Appendix B.

4.1 Main Experiments

We evaluate multi-PDE transfer under data-driven and expression-driven fine-tuning, focusing on two-dimensional tasks and further examining transfer to three spatial dimensions.

Table 1: Datasets used for multi-PDE pre-training and downstream evaluation under data-driven and expression-driven fine-tuning.

Fine-tuning setting	Pre-training datasets	Single-dataset fine-tuning tasks	Downstream information available
Data-driven fine-tuning	FNO-ν (3 regimes) + PDEBench CNS (5 configurations)	Each pre-training dataset; OOD datasets: CNS $(1, 10^{-8})$, DR, SWE, and CFDBench	High-fidelity solution data
Expression-driven fine-tuning	FNO-ν (3 regimes) + PDEBench CNS (5 configurations)	Each pre-training dataset; OOD datasets: CNS $(1, 10^{-8})$, DR, SWE, and CFDBench	Governing equations and prescribed initial and boundary conditions

Multi-PDE pre-training and transfer. Table 1 summarizes the protocol for multi-PDE pre-training and single-dataset fine-tuning, and Table 2 reports the corresponding results. Under expression-driven fine-tuning, all methods receive the same governing equation and prescribed initial and boundary conditions. We jointly pre-train on the datasets in the first eight data columns of Table 2. The remaining four columns evaluate transfer beyond pre-training: CNS $(1, 10^{-8})$ tests an unseen parameter regime, DR and SWE test unseen PDE families, and CFDBench tests transfer across geometries and boundary conditions.

We compare all baselines and SymOnet-100M at approximately 100M total parameters under a common pre-training and downstream evaluation protocol. Method-specific architectures and adaptation settings are detailed in Appendix E, with expression-driven fine-tuning protocols summarized in Table 8. SymOnet-500M is included as a larger-capacity variant to assess scaling beyond the approximately 100M matched-scale comparison. The evaluation protocol is detailed in Appendix E.

Under data-driven fine-tuning, SymOnet-100M attains the lowest L2RE on 9 of the 12 tasks among the evaluated models at approximately 100M parameters. Scaling SymOnet from 100M to 500M parameters further reduces L2RE on 10 of the 12 tasks under data-driven fine-tuning. Relative to the best-performing non-SymOnet baseline on each task, SymOnet-100M reduces data-driven L2RE by 57.9% on CNS $(1, 10^{-8})$ and 33.3% on both DR and SWE, supporting transfer across parameter regimes and PDE families.

Under expression-driven fine-tuning, SymOnet-100M outperforms all evaluated baselines on all 12 tasks at the matched parameter scale. Increasing SymOnet's model size from 100M to 500M parameters further reduces L2RE on 11 of the 12 tasks under expression-driven fine-tuning. Relative to the strongest non-SymOnet baseline, SymOnet-100M reduces L2RE by 10.1%–86.3% on tasks associated with the pre-training datasets and by 47.0%–77.5% on the three PDEBench OOD tasks. At the matched parameter scale, SymOnet achieves lower errors than all evaluated baselines on DR and SWE, demonstrating effective expression-driven transfer to PDE families absent from pre-training. These results support the use of a fixed pre-trained representation with task-specific residual adaptation across the evaluated PDEs, without downstream high-fidelity solution data.

We further study scaling in two controlled settings. With a fixed 100M SymOnet, increasing the fraction of trajectories from the same eight pre-training datasets from 12.5% to 100% monotonically decreases L2RE by 50.7%, 53.5%, and 59.5% on CNS $(1, 10^{-8})$, DR, and SWE, respectively, as reported in Table 3; the subset construction and training controls are detailed in Appendix D. Figure 2 shows that, with 25% of pre-training data, scaling SymOnet from 10M to 500M reduces L2RE on all three tasks; it achieves the lowest error at every scale on CNS and DR and from 50M onward on SWE.

Transfer from two to three spatial dimensions. We evaluate two-to-three-dimensional transfer on PDEBench CNS by comparing training from scratch and fine-tuning after two-dimensional pre-training under both data-driven and expression-driven supervision. Each model follows the corresponding fine-tuning protocols from the main experiments in Table 2. Appendix B provides the PDE formulation and experimental details. SymOnet achieves the lowest L2RE under all four protocols in Table 4. Under expression-driven fine-tuning, SymOnet achieves a 20.8% lower L2RE than DPOT, the best-performing baseline in this setting. These results extend the evidence for effective expression-driven adaptation beyond the main 2D benchmarks to three-dimensional dynamics.

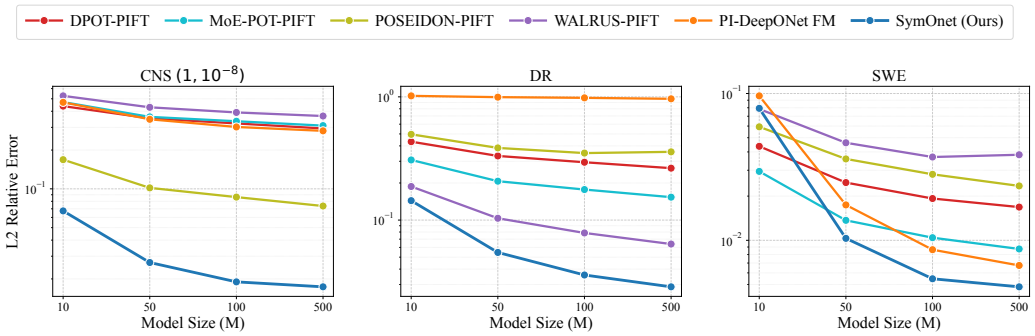

Figure 2: Controlled model-size scaling under expression-driven fine-tuning. All six methods are trained on the same randomly sampled 25% of trajectories from each pre-training dataset at the 10M, 50M, 100M, and 500M scales. Each curve reports L2RE; the three panels show the CNS $(1, 10^{-8})$, DR, and SWE OOD tasks.

Table 2: L2RE for multi-PDE pre-training with single-dataset fine-tuning. Within each fine-tuning setting and task, bold denotes the lowest error across all model sizes, and underlining denotes the lowest error among models with approximately 100M parameters. For the CNS columns, each pair denotes the Mach number and viscosity (M, η), with $\zeta = \eta$.

Model	FNO-ν			PDEBench CNS (M, η)						PDEBench		CFDBench
	10^{-5}	10^{-4}	10^{-3}	$(0.1, 10^{-1})$	$(0.1, 10^{-2})$	$(0.1, 10^{-8})$	$(1, 10^{-1})$	$(1, 10^{-2})$	$(1, 10^{-8})$	DR	SWE	–
Data-driven fine-tuning												
DPOT	0.05809	0.02659	0.00401	0.00786	0.02494	0.06562	0.01236	0.06874	0.19087	0.07503	0.00347	0.00444
F-Adapter	0.06381	0.04134	0.00749	0.00165	0.00937	0.06147	0.00386	0.06391	0.21280	0.08761	0.00039	0.00426
MoE-POT	0.05183	0.02473	0.00349	0.00161	0.00308	0.05045	0.00297	0.02585	0.18748	0.08089	0.00199	**0.00369**
POSEIDON	0.05234	0.02494	0.00686	0.00185	0.00322	0.07636	0.00255	0.01394	0.02409	0.09566	0.00340	0.02826
WALRUS	0.03172	0.00854	0.00243	**0.00043**	0.00219	0.04624	0.00136	0.01672	0.13757	0.02737	0.00097	0.01681
PI-DeepONet FM	0.10570	0.10932	0.03869	0.00309	0.02400	0.07169	0.01290	0.13515	0.23895	0.97396	0.00313	0.07208
SymOnet-100M	**0.02896**	0.00884	0.00196	0.00145	0.00128	0.03438	0.00117	**0.01045**	0.01014	0.01825	0.00026	0.00524
SymOnet-500M	0.02983	**0.00317**	**0.00143**	0.00055	**0.00114**	**0.02501**	**0.00060**	0.01192	**0.00942**	**0.01231**	**0.00022**	0.00496
Expression-driven fine-tuning												
DPOT-PIFT	0.10318	0.05126	0.00485	0.00455	0.02479	0.07244	0.01267	0.12337	0.25332	0.22613	0.01198	0.41069
MoE-POT-PIFT	0.08261	0.05271	0.00356	0.05965	0.06392	0.08252	0.04057	0.47882	0.25799	0.12481	0.00600	0.35632
POSEIDON-PIFT	0.07678	0.07610	0.00294	0.02423	0.02361	0.29788	0.02908	0.41483	0.05446	0.26593	0.01685	0.40588
WALRUS-PIFT	0.08376	0.07188	0.02434	0.07447	0.10049	0.10663	0.05008	0.19983	0.30803	0.04989	0.02115	0.17745
PI-DeepONet FM	0.24328	0.29380	0.11570	0.08127	0.25240	0.07208	0.17363	0.13816	0.24214	0.97575	0.00528	0.20603
SymOnet-100M	0.06899	0.03914	**0.00209**	0.00316	0.01806	0.03924	0.00173	0.07254	0.01223	0.02080	0.00280	0.02968
SymOnet-500M	**0.05690**	**0.01101**	0.00217	**0.00086**	**0.01554**	**0.03805**	**0.00113**	**0.01130**	**0.01181**	**0.01647**	**0.00253**	**0.01466**

Table 3: Pre-training-data scaling of SymOnet-100M under expression-driven fine-tuning. Each entry reports L2RE.

Pre-training data	CNS $(1, 10^{-8})$	DR	SWE
12.5%	0.02483	0.04472	0.00691
25%	0.01897	0.03584	0.00547
50%	0.01461	0.02591	0.00366
100%	0.01223	0.02080	0.00280

Table 4: L2RE on PDEBench 3D CNS for data-driven and expression-driven training from scratch or fine-tuning after 2D pre-training. Underlining identifies the lowest error in each protocol.

Protocol	DPOT	MoE-POT	POSEIDON	WALRUS	PI-DeepONet FM	SymOnet-100M
3D data-driven training from scratch	0.04819	0.04865	0.26615	0.06240	0.04889	0.04812
2D pre-training → 3D data-driven fine-tuning	0.04783	0.04832	0.09737	0.05377	0.04859	0.04706
3D expression-driven training from scratch	0.08878	0.08596	0.58503	0.08442	0.29631	0.07459
2D pre-training → 3D Expression-driven fine-tuning	0.06646	0.32019	0.27868	0.07155	0.19765	0.05264

4.2 Ablation Studies

We ablate individual architecture components and the trainable parameter subset under expression-driven fine-tuning at approximately 100M parameters in Table 5, with detailed controls and analysis in Appendix F. Across CNS-OOD, DR, and SWE, the architecture ablations assess the representation used for adaptation, while the adapter ablations compare which parts of that representation are allowed

Table 5: Controlled ablations under expression-driven fine-tuning at approximately 100M total parameters. Entries report L2RE on CNS-OOD, DR, and SWE; the frozen model serves as a reference without downstream updates.

Variant	CNS-OOD	DR	SWE
Architecture components			
Full SymOnet	0.01223	0.02080	0.00280
Pointwise tokenizer (replace Patch Embedding)	0.02816	0.05793	0.01027
Absolute position encoding (replace RoPE)	0.02084	0.03672	0.00561
MLP token mixer (replace IE self-attention)	0.02437	0.04618	0.00783
Linear coordinate projection (replace Query Encoder)	0.01769	0.03142	0.00436
MLP fusion (replace FB cross-attention)	0.02258	0.03976	0.00614
Adapter subsets			
Frozen (no downstream adaptation)	0.11874	0.26418	0.01857
Patch adapters	0.03952	0.08164	0.00831
IE adapters	0.03168	0.05983	0.00672
FB adapters	0.03647	0.07325	0.00794
All except Patch adapter	0.02286	0.03592	0.00471
All except IE adapter	0.02847	0.04618	0.00583
All except FB adapter	0.02419	0.03974	0.00516
All adapters (full SymOnet)	0.01223	0.02080	0.00280
Full-model fine-tuning	0.01181	0.01936	0.00253

to change. Every architectural replacement increases L2RE, and updating all three adapter groups yields the lowest error among the tested adapter subsets. Adapter updates reduce the frozen model's L2RE by 84.9%–92.1% on these tasks, while full-model fine-tuning provides a further 3.4%–9.6% reduction relative to adapter tuning.

4.3 Discussions

The main comparison evaluates overall performance, while the ablation studies assess the contributions of individual components within SymOnet. At matched model sizes, SymOnet leads on all expression-driven tasks and most data-driven tasks, supporting the effectiveness of its shared representation. Results on the unseen CNS parameter regime and on DR and SWE, two PDE families absent from pre-training, support generalization within and across PDE families. The additional CFDBench task evaluates transfer from regular-domain pre-training to a unified benchmark with variable geometries, internal obstacles, and non-periodic inlet, outlet, and wall conditions. At a fixed downstream adaptation protocol, increasing the amount of pre-training data improves OOD accuracy, supporting the value of the learned representation for expression-driven transfer. The frozen, adapter-tuned, and fully tuned comparisons quantify the accuracy gained through restricted updates and the remaining benefit of releasing all parameters. The adapter-group ablations further support joint adaptation of input encoding and feature fusion, whose local update mechanism is analyzed in Appendix A.6.

5 Conclusion

We introduced SymOnet, an operator foundation model that combines a shared input–query representation learned through multi-PDE pre-training with expression-driven adapter fine-tuning. Downstream adaptation uses governing equations and prescribed initial and boundary conditions while keeping the backbone frozen, requiring no downstream high-fidelity solution data. Our family-wise approximation theorem establishes that a shared query basis can approximate a family of PDE solution operators. Across 12 downstream tasks, SymOnet-100M outperforms all evaluated baselines under expression-driven fine-tuning at comparable model sizes and attains the lowest L2RE on 9 tasks under data-driven fine-tuning. Together with the scaling studies and ablations, these results support the use of shared pre-trained representations for expression-driven transfer across parameter regimes, PDE families, geometries, and boundary conditions.

SymOnet builds on existing self-attention and cross-attention mechanisms, leaving room for designs tailored to PDE solution operators. Incorporating operator structure into attention could improve approximation accuracy and computational efficiency.

AI USE STATEMENT

In this work, we used generative AI tools solely for language and grammar editing. We did not use generative AI tools for research ideation, literature review, methodology development, mathematical derivations, code generation, experiment design or execution, data analysis, or the creation of figures, tables, results, claims, or other research artifacts; all other required-disclosure categories are not applicable to this work. No additional uses subject to recommended disclosure apply. We reviewed all AI-assisted work and manually checked every AI-edited sentence for grammatical correctness, clarity, and preservation of the intended technical meaning. We confirmed that the edits did not introduce or alter any scientific claim, citation, equation, result, or artifact. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Shuhao Cao. Choose a transformer: Fourier or Galerkin. In *Advances in Neural Information Processing Systems*, volume 34, pp. 24924–24940, 2021.

Tianping Chen and Hong Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks*, 6(4):911–917, 1995.

Tianyu Chen, Haoyi Zhou, Ying Li, Hao Wang, Chonghan Gao, Rongye Shi, Shanghang Zhang, and Jianxin Li. OmniArch: Building foundation model for scientific computing. In *proceeding of the 42nd International Conference on Machine Learning*, volume 267, pp. 9860–9887, 2025.

Wuyang Chen, Jialin Song, Pu Ren, Shashank Subramanian, Dmitriy Morozov, and Michael W. Mahoney. Data-efficient operator learning via unsupervised pretraining and in-context learning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 6213–6245, 2024.

M. W. M. Gamini Dissanayake and Nhan Phan-Thien. Neural-network-based approximations for solving partial differential equations. *Communications in Numerical Methods in Engineering*, 10 (3):195–201, 1994.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Peng Fan and Guofei Pang. Convolutional-neural-operator-based transfer learning for solving PDEs. *arXiv preprint arXiv:2512.17969*, 2025.

Jayesh K. Gupta and Johannes Brandstetter. Towards multi-spatiotemporal-scale generalized PDE modeling. *Transactions on Machine Learning Research*, 2023.

Zhongkai Hao, Zhengyi Wang, Hang Su, Chengyang Ying, Yinpeng Dong, Songming Liu, Ze Cheng, Jian Song, and Jun Zhu. GNOT: A general neural operator transformer for operator learning. In *proceeding of the 40th International Conference on Machine Learning*, volume 202, pp. 12556–12569, 2023.

Zhongkai Hao, Chang Su, Songming Liu, Julius Berner, Chengyang Ying, Hang Su, Anima Anandkumar, Jian Song, and Jun Zhu. DPOT: Auto-regressive denoising operator transformer for large-scale PDE pre-training. In *proceeding of the 41st International Conference on Machine Learning*, volume 235, pp. 17616–17635, 2024.

Maximilian Herde, Bogdan Raonic, Tobias Rohner, Roger Kaeppeli, Roberto Molinaro, Emmanuel de Bezenac, and Siddhartha Mishra. POSEIDON: Efficient foundation models for PDEs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 72525–72624, 2024.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *proceeding of the 36th International Conference on Machine Learning*, volume 97, pp. 2790–2799, 2019.

Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for partial differential equations' operator learning. *Transactions on Machine Learning Research*, 2023.

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

Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *ACM/IMS Journal of Data Science*, 1(3):9:1–9:27, 2024.

Cooper Lorsung and Amir Barati Farimani. PICL: Physics informed contrastive learning for partial differential equations. *APL Machine Learning*, 2(4):046107, 2024.

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.

Yining Luo, Yingfa Chen, and Zhen Zhang. CFDBench: A large-scale benchmark for machine learning methods in fluid dynamics. *arXiv preprint arXiv:2310.05963*, 2023.

Siddharth Mansingh, James Amarel, Ragib Arnab, Arvind Mohan, Kamaljeet Singh, Gerd J. Kunde, Nicolas Hengartner, Benjamin Migliori, Emily Casleton, Nathan A. DeBardeleben, Ayan Biswas, Diane Oyen, and Earl Lawrence. Towards reasoning for PDE foundation models: A reward-model-driven inference-time-scaling algorithm. *arXiv preprint arXiv:2509.02846*, 2025.

Michael McCabe, Payel Mukhopadhyay, Tanya Marwah, Bruno Régaldo-Saint Blancard, François Rozet, Cristiana Diaconu, Lucas Meyer, Kaze W. K. Wong, Hadi Sotoudeh, Alberto Bietti, Irina Espejo, Rio Fear, Siavash Golkar, Tom Hehir, Keiya Hirashima, Geraud Krawezik, François Lanusse, Rudy Morel, Ruben Ohana, Liam Parker, Mariel Pettee, Jeff Shen, Kyunghyun Cho, Miles Cranmer, and Shirley Ho. WALRUS: A cross-domain foundation model for continuum dynamics. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306, 2026.

Vlad Medvedev, Leon Armbruster, Christopher Straub, Georg Kruse, and Andreas Rosskopf. Physics-informed fine-tuning of foundation models for partial differential equations. In *International Conference on Learning Representations*, 2026.

Elisa Negrini, Yuxuan Liu, Liu Yang, Stanley J. Osher, and Hayden Schaeffer. A multimodal PDE foundation model for prediction and scientific text descriptions. *Journal of Machine Learning*, 4 (4):290–317, 2025.

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

Md Ashiqur Rahman, Robert Joseph George, Mogab Elleithy, Daniel Leibovici, Zongyi Li, Boris Bonev, Colin White, Julius Berner, Raymond A. Yeh, Jean Kossaifi, Kamyar Azizzadenesheli, and Anima Anandkumar. Pretraining codomain attention neural operators for solving multiphysics PDEs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 104035–104064, 2024.

Maziar Raissi, Paris Perdikaris, and George Em 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.

Maziar Raissi, Alireza Yazdani, and George Em Karniadakis. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 367(6481):1026–1030, 2020.

Mahindra Singh Rautela, Alexander Most, Siddharth Mansingh, Bradley C. Love, Ayan Biswas, Diane Oyen, and Earl Lawrence. MORPH: Shape-agnostic PDE foundation models. *arXiv preprint arXiv:2509.21670*, 2025.

Shyam Sankaran, Hanwen Wang, and Paris Perdikaris. WaveLiT: A parameter-efficient architecture for neural PDE solvers. In *NeurIPS Workshop on Machine Learning and the Physical Sciences*, 2025.

Abu Bucker Siddik, Diane Oyen, Alexander Most, Michal Kucer, and Ayan Biswas. SPUS: A lightweight and parameter-efficient foundation model for PDEs. *arXiv preprint arXiv:2510.01370*, 2025.

Eduardo Soares, Emilio Vital Brazil, Victor Shirasuna, Breno W. S. R. de Carvalho, and Cristiano Malossi. Towards a foundation model for partial differential equations across physics domains. In *AAAI 2026 Workshop*, 2026.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with Rotary Position Embedding. *Neurocomputing*, 568:127063, 2024.

Shashank Subramanian, Peter Harrington, Kurt Keutzer, Wahid Bhimji, Dmitriy Morozov, Michael W. Mahoney, and Amir Gholami. Towards foundation models for scientific machine learning: Characterizing scaling and transfer behavior. In *Advances in Neural Information Processing Systems*, volume 36, pp. 71242–71262, 2023.

Jingmin Sun, Yuxuan Liu, Zecheng Zhang, and Hayden Schaeffer. Towards a foundation model for partial differential equations: Multi-operator learning and extrapolation. *Physical Review E*, 111 (3):035304, 2025a.

Jingmin Sun, Zecheng Zhang, and Hayden Schaeffer. LeMON: Learning to learn multi-operator networks. *arXiv preprint arXiv:2408.16168*, 2025b.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. PDEBench: An extensive benchmark for scientific machine learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 1596–1611, 2022.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems*, volume 33, pp. 7537–7547, 2020.

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. *arXiv preprint arXiv:2503.19081*, 2025.

Hong Wang, Haiyang Xin, Jie Wang, Xuanze Yang, Fei Zha, Huanshuo Dong, and Yan Jiang. Mixture-of-experts operator transformer for large-scale PDE pre-training. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

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.

Haixu Wu, Huakun Luo, Haowen Wang, Jianmin Wang, and Mingsheng Long. Transolver: A fast transformer solver for PDEs on general geometries. In *proceeding of the 41st International Conference on Machine Learning*, volume 235, pp. 53681–53705, 2024.

Liu Yang and Stanley J. Osher. PDE generalization of in-context operator networks: A study on 1D scalar nonlinear conservation laws. *Journal of Computational Physics*, 519:113379, 2024.

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

Zhanhong Ye, Zining Liu, Bingyang Wu, Hongjie Jiang, Leheng Chen, Minyan Zhang, Xiang Huang, Qinghe Meng, Jingyuan Zou, Hongsheng Liu, and Bin Dong. PDEformer-2: A versatile foundation model for two-dimensional partial differential equations. *arXiv preprint arXiv:2507.15409*, 2025.

Hangwei Zhang, Chun Kang, Yan Wang, and Difan Zou. F-adapter: Frequency-adaptive parameter-efficient fine-tuning in scientific machine learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Zecheng Zhang. MODNO: Multi-operator learning with distributed neural operators. *Computer Methods in Applied Mechanics and Engineering*, 431:117229, 2024.

Zecheng Zhang, Christian Moya, Lu Lu, Guang Lin, and Hayden Schaeffer. DeepONet as a multi-operator extrapolation model: Distributed pretraining with physics-informed fine-tuning. *Journal of Computational Physics*, 547:114537, 2026.

Anthony Zhou, Cooper Lorsung, AmirPouya Hemmasian, and Amir Barati Farimani. Strategies for pretraining neural operators. *Transactions on Machine Learning Research*, 2024.

Hang Zhou, Yuezhou Ma, Haixu Wu, Haowen Wang, and Mingsheng Long. UniSolver: PDE-conditional transformers towards universal neural PDE solvers. In *proceeding of the 42nd International Conference on Machine Learning*, volume 267, pp. 79061–79088, 2025.

Congcong Zhu, Xiaoyan Xu, Jiayue Han, and Jingrun Chen. Physics-informed temporal alignment for auto-regressive PDE foundation models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, 2025.

A Detailed Theoretical Analysis and Proofs

Appendices A.1–A.4 develop the setting, shared query basis, and coefficient approximation needed to prove Theorem 3.1. The subsequent sections connect this representation to the Galerkin decoder and provide complementary analyses of local adapter updates, autoregressive propagation, residual-to-solution stability, and adapter dimension.

A.1 PDE-Family Setting and Shared-Query Model

Fix J equation families, indexed by $j = 1, \dots, J$. In each family, $\vartheta \in \Theta_j \subset \mathbb{R}^{k_j}$ specifies the physical parameters, boundary data, and any geometry parameters held fixed while the input function varies; Θ_j is compact and k_j is the parameter-vector dimension. Thus $\tau = (j, \vartheta)$ and $\mathfrak{P} = \bigsqcup_{j=1}^{J} \{j\} \times \Theta_j$. Equip each component with its Euclidean topology and the union with the disjoint-union topology. Padding ϑ with zeros and adjoining a one-hot family label defines an embedding $\iota : \mathfrak{P} \to \mathbb{R}^k$, with $k = J + \max_j k_j$, whose image is a finite union of compact sets. This proves compactness and specifies a Euclidean metric through ι; it imposes no continuity between different equation families.

For this analysis, inputs and outputs are expressed on fixed compact reference domains D_a and $Q \subset \mathbb{R}^{d_q}$, where d_q is the number of query coordinates, and c_{in} and $c = c_{\text{out}}$ are the input and output channel counts. Fixed coordinate transformations and zero padding of unused channels can provide these representations. For varying geometries, their pullbacks or extensions must additionally preserve the continuity conditions below. A family label can distinguish NS, SWE, DR, CNS, or a CFDBench geometry and boundary type; coefficients such as viscosity and reaction rates enter ϑ. The input function a represents the varying initial condition in expression-driven fine-tuning and is not duplicated in τ. This finite-dimensional parameterization applies to prescribed parameterized boundary and geometry families, not to all possible PDE specifications.

Assumption A.1 (Admissible inputs and stable reconstruction). Let K_τ contain the admissible continuous input functions for task τ. The set $\mathcal{E} = \{(\tau, a) : a \in K_\tau\}$ is a compact subset of $\mathfrak{P} \times C(D_a; \mathbb{R}^{c_{\text{in}}})$, where the function space has the supremum norm. For a fixed finite grid indexed by its resolution h, the continuous sampling map collects all grid values and input channels into $S_h a \in \mathbb{R}^{n_h}$, where n_h is the number of sampled scalar values, and defines the compact image $\mathcal{E}_h = \{(\tau, S_h a) : (\tau, a) \in \mathcal{E}\}$. There is a jointly continuous reconstruction $(\tau, A) \mapsto R_{h,\tau} A$ on $\mathcal{E}_h$ with $\|a - R_{h,\tau} S_h a\|_\infty \le e_h$. Both original and reconstructed inputs lie in a compact set $\widetilde{\mathcal{E}}$ on which a unique solution map $(\tau, b) \mapsto \mathcal{G}^*_\tau(b) \in C(Q; \mathbb{R}^c)$ is continuous in the supremum norm.

The solution map is required to be well posed and continuous in the task parameters and inputs over the chosen time interval, including for reconstructed inputs. Define its uniform input-continuity modulus by

$$\omega(s) = \sup_{\substack{(\tau,b),(\tau,b') \in \widetilde{\mathcal{E}} \\ \|b - b'\|_\infty \le s}} \|\mathcal{G}^*_\tau(b) - \mathcal{G}^*_\tau(b')\|_\infty. \tag{12}$$

Compactness and continuity imply $\omega(s) \to 0$ as $s \downarrow 0$: otherwise a sequence of pairs at vanishing input distance has convergent subsequences with the same limit and nonvanishing output distance, a contradiction. The sampled target $g_h(\tau, A, q) = \mathcal{G}^*_\tau(R_{h,\tau} A)(q)$ is jointly continuous on $\mathcal{D}_h = \mathcal{E}_h \times Q$, since reconstruction, the solution map, and evaluation on $C(Q)$ are continuous. Both $\mathcal{E}_h$ and $\mathcal{D}_h$ are compact, and

$$\sup_{(\tau,a) \in \mathcal{E},\, q \in Q} \|g_h(\tau, S_h a, q) - \mathcal{G}^*_\tau(a)(q)\|_\infty \le \omega(e_h). \tag{13}$$

A concrete sufficient sampling condition is a uniform Hölder bound $\|a(x) - a(y)\|_\infty \le H \|x - y\|_2^\alpha$, $0 < \alpha \le 1$, on a rectangular reference domain. Multilinear interpolation on a grid whose cells have diameter at most h gives $e_h \le H h^\alpha$: the interpolant is a convex combination of vertex values, each at distance at most h from the evaluation point. A closed uniformly bounded set with this modulus is compact by Arzelà–Ascoli. If the solution map is L_G-Lipschitz on the required inputs, Eq. (13) becomes $L_G H h^\alpha$. Reconstruction must still remain within its well-posedness domain. These conditions do not give a uniform supremum-norm interpolation bound for unresolved shocks.

For a query basis with p components, the shared-query operator used in Theorem 3.1 is defined by

$$\widehat{\mathcal{G}}_\tau(A,q) := F_{\alpha_\tau}(A)^\top \psi_{\theta_0}(q), \qquad F_{\alpha_\tau}(A) \in \mathbb{R}^{p\times c}, \quad \psi_{\theta_0}(q) \in \mathbb{R}^p. \tag{14}$$

Here θ_0 denotes the shared query parameters and α_τ the task-specific coefficient-network parameters; their dependence is suppressed in $\widehat{\mathcal{G}}_\tau$ for brevity. We write $F_\tau = F_{\alpha_\tau}$ and $\psi_B = \psi_{\theta_0} = T_B\gamma_B$ in the construction below. The readout in this representation is the identity; the nonlinear decoder is defined in Eq. (21). For vector-valued functions, the supremum norm is taken over the domain and output components; $\|C\|_{\max}$ denotes the largest absolute matrix entry. Euclidean, spectral, and Frobenius norms are denoted by $\|\cdot\|_2$, $\|\cdot\|_{\mathrm{op}}$, and $\|\cdot\|_F$ where needed below.

A.2 Shared Query Basis

Lemma A.2 (Uniform shared-basis decomposition). *For every $\eta > 0$, there are continuous functions $\phi_1, \dots, \phi_p \geq 0$ on Q with $\sum_i \phi_i = 1$ and a continuous map $C : \mathcal{E}_h \to \mathbb{R}^{p\times c}$ such that $\sup_{\mathcal{D}_h} \|g_h(\tau, A, q) - C(\tau, A)^\top \Phi(q)\|_\infty < \eta$, where $\Phi = (\phi_1, \dots, \phi_p)^\top$.*

Proof. Uniform continuity on $\mathcal{D}_h$ gives $r > 0$ such that

$$\sup_{\substack{(\tau,A)\in\mathcal{E}_h,\ q,q'\in Q \\ \|q-q'\|_2<r}} \|g_h(\tau,A,q) - g_h(\tau,A,q')\|_\infty \leq \eta/2.$$

Choose a finite $r/2$-net $\{q_i\}_{i=1}^p \subset Q$ and define

$$w_i(q) = (r - \|q - q_i\|_2)_+, \qquad \phi_i(q) = \frac{w_i(q)}{\sum_{j=1}^p w_j(q)}, \qquad C_{i,:}(\tau,A) = g_h(\tau,A,q_i)^\top,$$
$$\sum_i w_i(q) \geq r/2, \qquad \sum_i \phi_i(q) = 1, \qquad \phi_i(q) > 0 \implies \|q - q_i\|_2 < r.$$

Thus ϕ_i and C are continuous, and for every $(\tau, A, q) \in \mathcal{D}_h$,

$$\begin{aligned} &\|g_h(\tau,A,q) - C(\tau,A)^\top\Phi(q)\|_\infty \\ &= \left\| \sum_{i=1}^p \phi_i(q)\big[g_h(\tau,A,q) - g_h(\tau,A,q_i)\big] \right\|_\infty \\ &\leq \sum_{i=1}^p \phi_i(q)\|g_h(\tau,A,q) - g_h(\tau,A,q_i)\|_\infty \leq \eta/2 < \eta. \end{aligned}$$

Finally, $C(\mathcal{E}_h)$ is the continuous image of a compact set, so $M_C = \sup_{\mathcal{E}_h}\|C\|_{\max} < \infty$. □

Lemma A.3 (Shared Fourier features). *Let $\Phi : Q \to \mathbb{R}^p$ be continuous and let $\gamma_B(q) = (\sin(2\pi b_\ell^\top q), \cos(2\pi b_\ell^\top q))_{\ell=1}^M$, where b_ℓ are independent $N(0, \sigma_B^2 I)$ draws with $\sigma_B > 0$. For any $\eta_q > 0$ and $0 < \delta < 1$, there are finite M and, with probability at least $1 - \delta$, a linear map T_B with $\sup_Q \|T_B\gamma_B - \Phi\|_\infty < \eta_q$.*

Proof. Fix $q_0 \in Q$, write $\bar{q} = q - q_0$, and let $D = \mathrm{diam}(Q)$. Real trigonometric polynomials with arbitrary real frequencies form an algebra containing constants and separating points of Q. Stone–Weierstrass gives approximants

$$P_i(q) = \sum_{k=1}^K \{a_{ik}\cos(2\pi\xi_k^\top\bar{q}) + d_{ik}\sin(2\pi\xi_k^\top\bar{q})\}, \quad \sup_Q \|P - \Phi\|_\infty < \eta_q/2, \tag{15}$$

using a common finite frequency list, including zero for constants. Let $A_* = \max_i \sum_k (|a_{ik}| + |d_{ik}|)$, and choose

$$r = \frac{\eta_q}{4\pi\max\{1,D\}\max\{1,A_*\}}, \quad \mu_* = \min_{k\leq K}\Pr\{\|b - \xi_k\|_2 < r\} > 0, \quad M \geq \left\lceil \mu_*^{-1}\log\frac{K}{\delta}\right\rceil. \tag{16}$$

Let $E_k = \{\min_{\ell \le M} \|b_\ell - \xi_k\|_2 \ge r\}$ and $\mu_k = \Pr\{\|b - \xi_k\|_2 < r\}$. Independence and the union bound give

$$\Pr\left(\bigcup_{k=1}^{K} E_k\right) \le \sum_{k=1}^{K} (1-\mu_k)^M \le K e^{-M\mu_*} \le \delta.$$

On $\bigcap_k E_k^c$, select $\widetilde{\xi}_k = b_{\ell(k)}$ with $\|\widetilde{\xi}_k - \xi_k\|_2 < r$, and let $\widetilde{P}$ be P with these frequencies. The mean-value inequality gives, for $f \in \{\sin, \cos\}$,

$$|f(2\pi \widetilde{\xi}_k^\top \bar{q}) - f(2\pi \xi_k^\top \bar{q})| \le 2\pi \|\widetilde{\xi}_k - \xi_k\|_2 \|\bar{q}\|_2 \le 2\pi D r,$$
$$\sup_Q \|\widetilde{P} - P\|_\infty \le 2\pi D r A_* \le \eta_q/2.$$

For $t_\ell = 2\pi b_\ell^\top q_0$, the identity

$$\begin{pmatrix} \sin(2\pi b_\ell^\top (q - q_0)) \\ \cos(2\pi b_\ell^\top (q - q_0)) \end{pmatrix} = \begin{pmatrix} \cos t_\ell & -\sin t_\ell \\ \sin t_\ell & \cos t_\ell \end{pmatrix} \begin{pmatrix} \sin(2\pi b_\ell^\top q) \\ \cos(2\pi b_\ell^\top q) \end{pmatrix}$$

expresses $\widetilde{P}$ as $T_B \gamma_B$, with repeated selected frequencies combined in the same columns. Therefore

$$\sup_Q \|T_B \gamma_B - \Phi\|_\infty \le \sup_Q \|\widetilde{P} - P\|_\infty + \sup_Q \|P - \Phi\|_\infty < \eta_q.$$

The Gaussian density gives the following lower bound, where v_{d_q} is the volume of the unit ball:

$$\mu_* \ge v_{d_q} r^{d_q} (2\pi\sigma_B^2)^{-d_q/2} \exp\left[-\frac{(\max_k \|\xi_k\|_2 + r)^2}{2\sigma_B^2}\right]. \tag{17}$$

The sufficient feature count in Eq. (16) depends on the approximating polynomial through K, A_*, and $\{\xi_k\}$, and on d_q, D, σ_B, η_q, and δ. □

A.3 Task-Specific Coefficient Approximation

Lemma A.4 (Coefficient networks of a common finite width). *For continuous $C : \mathcal{E}_h \to \mathbb{R}^{p\times c}$ and any $\eta_C > 0$, a finite-width, single-hidden-layer GELU network N on $(\iota(\tau), A)$ satisfies $\sup_{\mathcal{E}_h} \|N(\iota(\tau), A) - C(\tau, A)\|_{\max} < \eta_C$. Consequently $F_\tau(A) = N(\iota(\tau), A)$ is a family of coefficient networks with one common finite width.*

Proof. The embedded set $\{(\iota(\tau), A) : (\tau, A) \in \mathcal{E}_h\}$ is compact in a finite-dimensional Euclidean space, and C is continuous on it. Let σ denote GELU. It is smooth and nonpolynomial, so for every integer $m \ge 1$ there is a scalar b with $\sigma^{(m)}(b) \ne 0$. Taylor expansion, uniformly for bounded z, gives

$$\frac{\sum_{j=0}^{m} (-1)^{m-j} \binom{m}{j} \sigma(b + jt\, w^\top z)}{t^m \sigma^{(m)}(b)} \longrightarrow (w^\top z)^m \qquad (t \to 0). \tag{18}$$

The cancellation and polarization identities used here are

$$\sum_{j=0}^{m} (-1)^{m-j} \binom{m}{j} j^\ell = \begin{cases} 0, & 0 \le \ell < m, \\ m!, & \ell = m, \end{cases}$$
$$\prod_{\ell=1}^{m} x_\ell = \frac{1}{m!} \sum_{S \subseteq \{1,\dots,m\}} (-1)^{m-|S|} \left(\sum_{\ell \in S} x_\ell\right)^m.$$

Taylor's remainder divided by t^m is $O(|t|)$ uniformly on the compact domain. Thus every monomial, including repeated variables, lies in the uniform closure of the GELU ridge-function span. Constants are supplied by the output bias. By Stone–Weierstrass, choose a matrix polynomial P with $\sup \|P - C\|_{\max} < \eta_C/2$, and then a finite GELU network N with $\sup \|N - P\|_{\max} < \eta_C/2$. Consequently

$$\sup_{\mathcal{E}_h} \|N - C\|_{\max} \le \sup_{\mathcal{E}_h} \|N - P\|_{\max} + \sup_{\mathcal{E}_h} \|P - C\|_{\max} < \eta_C.$$

Concatenating the hidden units for the pc entries gives one finite width. Partitioning the input weights as (W_{task}, W_A) yields

$$F_\tau(A) = U\sigma(W_A A + b_\tau) + v, \qquad b_\tau = b + W_{\text{task}} \iota(\tau),$$

which has that same width for every τ. □

The coefficient networks have a common width that may depend on η_C. Their inputs are sampled functions A; approximation of corrections on hidden-state domains is considered in Appendix A.9.

A.4 Proof of Theorem 3.1

Proof. Apply Lemma A.2 with $\eta = \varepsilon/3$ and retain its p and finite bound M_C. Choose $\eta_q < \varepsilon/[3p\max\{1, M_C\}]$. Lemma A.3 gives, with probability at least $1-\delta$, one $\psi_B = T_B\gamma_B$ satisfying $\|\psi_B - \Phi\|_\infty < \eta_q$. Since $0 \le \phi_i \le 1$, $\sup_Q \|\psi_B\|_\infty \le 1 + \eta_q$. Lemma A.4 supplies coefficient networks with $\eta_C < \varepsilon/[3p(1+\eta_q)]$. For every $(\tau, A, q) \in \mathcal{D}_h$,

$$\begin{aligned}\|F_\tau(A)^\top \psi_B(q) - g_h(\tau, A, q)\|_\infty &\le \|C^\top\Phi - g_h\|_\infty + \|C^\top(\psi_B - \Phi)\|_\infty + \|(F_\tau - C)^\top\psi_B\|_\infty \\ &< \varepsilon/3 + pM_C\eta_q + p\eta_C(1+\eta_q) < \varepsilon.\end{aligned} \tag{19}$$

The same frequency draw and query map apply to every task. Substituting $A = S_h a$ and using Eq. (13) yields

$$\begin{aligned}&\sup_{(\tau,a)\in\mathcal{E},\, q\in Q} \|\widehat{\mathcal{G}}_\tau(S_h a, q) - \mathcal{G}^*_\tau(a)(q)\|_\infty \\ &\le \sup_{(\tau,A,q)\in\mathcal{D}_h} \|\widehat{\mathcal{G}}_\tau(A, q) - g_h(\tau, A, q)\|_\infty + \omega(e_h) < \varepsilon + \omega(e_h).\end{aligned}$$

□

A.5 Galerkin Factorization and Stability

To relate the shared-query representation in Theorem 3.1 to SymOnet, we factor each Galerkin attention head into shared query features and input-dependent key–value moments, then describe their composition with the decoder's nonlinear maps. Let $x : Q \to \mathbb{R}^d$ be the shared query map and $Z_\alpha(A) \in \mathbb{R}^{n\times d}$ the input-token map. Fix H attention heads of width d_h, token coordinates $s_1, \dots, s_n$, and orthogonal RoPE matrices $R_h(q) \in \mathbb{R}^{d_h\times d_h}$. For $v \in \mathbb{R}^n$, define normalization across token positions by

$$N_\epsilon(v) = \frac{P_n v}{(n^{-1}\|P_n v\|_2^2 + \epsilon)^{1/2}}, \qquad P_n = I_n - \frac{\mathbf{1}\mathbf{1}^\top}{n}, \qquad \epsilon > 0.$$

On matrices, N_ϵ acts columnwise; $\mathcal{R}_h$ rotates the ith row by $R_h(s_i)^\top$. A Galerkin head maps query u and token rows $k_i^\top, v_i^\top$ to $n^{-1}\sum_i (u^\top k_i)v_i$. Let $\mathcal{A}_{\alpha,\mathrm{ca}}$ and $\mathcal{A}_{\alpha,\mathrm{ff}}$ be residual adapter maps on $\mathbb{R}^d$, $\mathcal{H}(h) = h + \mathrm{FFN}(h)$, and $\rho : \mathbb{R}^d \to \mathbb{R}^c$ the shared readout.

Proposition A.5 (Galerkin query–moment identity). *Let $W_{Q,h} \in \mathbb{R}^{d_h\times d}$, $W_{K,h}, W_{V,h} \in \mathbb{R}^{d\times d_h}$, $W_O \in \mathbb{R}^{d\times Hd_h}$, and $b_O \in \mathbb{R}^d$ be fixed. Set $N_K = N_{\epsilon_K}$, $N_V = N_{\epsilon_V}$, and define*

$$\begin{aligned}u_h(q) &= R_h(q)W_{Q,h}x(q), \quad \overline{K}_h(Z) = \mathcal{R}_h N_K(ZW_{K,h}), \\ \overline{V}_h(Z) &= N_V(ZW_{V,h}), \qquad M_h(Z) = \frac{1}{n}\overline{K}_h(Z)^\top\overline{V}_h(Z) \in \mathbb{R}^{d_h\times d_h}.\end{aligned} \tag{20}$$

Writing $\mathcal{A}_{\alpha,\mathrm{ca}}$ and $\mathcal{A}_{\alpha,\mathrm{ff}}$ for the two residual adapters, the decoder output is exactly

$$\begin{aligned}h_0(A, q) &= x(q) + W_O\,\mathrm{concat}_{h=1}^H\{M_h(Z_\alpha(A))^\top u_h(q)\} + b_O, \\ h_1(A, q) &= \mathcal{A}_{\alpha,\mathrm{ca}}(h_0(A, q)), \\ h_2(A, q) &= \mathcal{A}_{\alpha,\mathrm{ff}}\big(h_1(A, q) + \mathrm{FFN}(h_1(A, q))\big), \\ \widehat{\mathcal{G}}_{\theta_0,\alpha}(A, q) &= \rho(h_2(A, q)).\end{aligned} \tag{21}$$

Here ρ is the frozen composition of feature expansion, LayerNorm, the shared output MLP, and the concatenated output heads. The query MLP, projection matrices, normalization rules, RoPE, FFN, and ρ belong to θ_0. Task dependence enters through the adapters in Z_α and the two decoder adapters.

Proof. Write $k_{h,i}^\top$ and $v_{h,i}^\top$ for the ith rows of $\overline{K}_h$ and $\overline{V}_h$. For each output coordinate $b = 1, \dots, d_h$, the Galerkin sum is

$$[o_h(q, Z)]_b = \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{a=1}^{d_h} [u_h(q)]_a [k_{h,i}]_a \right) [v_{h,i}]_b$$
$$= \sum_{a=1}^{d_h} [u_h(q)]_a \left(\frac{1}{n} \sum_{i=1}^{n} [k_{h,i}]_a [v_{h,i}]_b \right) = [M_h(Z)^\top u_h(q)]_b.$$

Let $U_h(Y)$ have rows $u_h(q_j)^\top$ for queries $Y = \{q_j\}$. The same identity in matrix form is

$$O_h(Y, Z) = \frac{1}{n} \big(U_h(Y) \overline{K}_h(Z)^\top\big) \overline{V}_h(Z) = U_h(Y) M_h(Z).$$

With $\mathcal{O}(q, Z) = W_O \operatorname{concat}_h o_h(q, Z) + b_O$, the decoder composition is

$$\widehat{\mathcal{G}}_{\theta_0,\alpha}(A, q) = \big(\rho \circ \mathcal{A}_{\alpha,\mathrm{ff}} \circ \mathcal{H} \circ \mathcal{A}_{\alpha,\mathrm{ca}}\big)\big(x(q) + \mathcal{O}(q, Z_\alpha(A))\big).$$

Substitution of $o_h = M_h^\top u_h$ gives each line of Eq. (21). □

The moments also satisfy $\|M_h(Z)\|_F \le d_h$ and, at fixed token positions,

$$\|M_h(Z) - M_h(Z')\|_F \le \sqrt{\frac{d_h}{n}} \left(\frac{\|W_{K,h}\|_{\mathrm{op}}}{\sqrt{\epsilon_K}} + \frac{\|W_{V,h}\|_{\mathrm{op}}}{\sqrt{\epsilon_V}} \right) \|Z - Z'\|_F. \tag{22}$$

Proof. Set $y = P_n v$ and $s^2 = \|y\|_2^2/n + \epsilon$. Direct differentiation gives

$$DN_\epsilon(v) = \frac{P_n}{s} - \frac{yy^\top}{ns^3}, \qquad DN_\epsilon(v)\mathbf{1} = 0, \qquad DN_\epsilon(v)y = \frac{\epsilon}{s^3} y,$$
$$DN_\epsilon(v)w = \frac{1}{s} w \quad (w \perp \mathbf{1},\ w \perp y), \qquad \|DN_\epsilon(v)\|_{\mathrm{op}} \le \epsilon^{-1/2}.$$

The bound also holds when $y = 0$, since then $DN_\epsilon(v) = P_n/\sqrt{\epsilon}$. Integration along the segment between v and v' gives

$$\|N_\epsilon(v) - N_\epsilon(v')\|_2 \le \int_0^1 \|DN_\epsilon(v' + t(v - v'))\|_{\mathrm{op}} \|v - v'\|_2 \, dt \le \epsilon^{-1/2} \|v - v'\|_2,$$
$$\|N_\epsilon(v)\|_2^2 = \frac{\|y\|_2^2}{\|y\|_2^2/n + \epsilon} \le n.$$

Columnwise application and orthogonality of each $R_h(s_i)$ imply

$$\|\overline{K}_h(Z)\|_F,\ \|\overline{V}_h(Z)\|_F \le \sqrt{nd_h}, \qquad \|M_h(Z)\|_F \le n^{-1} \|\overline{K}_h(Z)\|_F \|\overline{V}_h(Z)\|_F \le d_h,$$
$$\|\overline{K}_h(Z) - \overline{K}_h(Z')\|_F \le \epsilon_K^{-1/2} \|W_{K,h}\|_{\mathrm{op}} \|Z - Z'\|_F,$$
$$\|\overline{V}_h(Z) - \overline{V}_h(Z')\|_F \le \epsilon_V^{-1/2} \|W_{V,h}\|_{\mathrm{op}} \|Z - Z'\|_F.$$

Writing $K = \overline{K}_h(Z)$, $K' = \overline{K}_h(Z')$ and similarly for V, V', we obtain

$$M_h(Z) - M_h(Z') = n^{-1} \big[(K - K')^\top V + K'^\top (V - V')\big],$$
$$\|M_h(Z) - M_h(Z')\|_F \le n^{-1} \big(\|K - K'\|_F \|V\|_F + \|K'\|_F \|V - V'\|_F\big).$$

Substituting the preceding bounds proves Eq. (22). □

For a readout of the form $\rho(v) = W_3 \sigma(W_2 \sigma(W_1 \operatorname{LN}(Ev + e))) + b_3$, with LayerNorm gain g and denominator offset $\epsilon_{\mathrm{LN}} > 0$, the normalization derivative bound and the Lipschitz composition inequality give

$$\operatorname{Lip}(\mathrm{LN}) \le \frac{\|g\|_\infty}{\sqrt{\epsilon_{\mathrm{LN}}}}, \qquad \operatorname{Lip}(\rho) \le \|W_3\|_{\mathrm{op}} \|W_2\|_{\mathrm{op}} \|W_1\|_{\mathrm{op}} \|E\|_{\mathrm{op}} L_\sigma^2 \frac{\|g\|_\infty}{\sqrt{\epsilon_{\mathrm{LN}}}},$$

where $L_\sigma = \sup_t |\sigma'(t)| < \infty$ for GELU. For a smooth gated FFN and a bounded convex hidden-state set B, $\operatorname{Lip}(\mathcal{H}|_B) \le 1 + \sup_{h \in B} \|D \operatorname{FFN}(h)\|_{\mathrm{op}} < \infty$.

The identity exposes two adaptation routes: changes in Z_α alter the key–value moments, and the two Fusion Block adapters transform the fused features before the fixed readout. Differentiating this composition yields the adapter residual Jacobian used in Theorem A.6, with the query map, attention projections, and shared readout fixed.

A.6 Expression-driven Adaptation with a Fixed Backbone

Complementing the representation result in Theorem 3.1, this section studies how adapter updates affect expression-driven residuals for a given pre-trained model. Fix a pre-trained backbone θ_0, the adapter widths, and a downstream task τ, and write the prediction as $u_\alpha(a, q) = \mathcal{G}_{\theta_0,\alpha}(a)(q)$. The adapter parameter vector α includes the affine projections of the Patch Embedding, Input Encoder, and Fusion Block residual modules. Its identity initialization α_0 has zero up-projections and recovers the pre-trained predictor.

Residual map and query derivatives. For fixed input functions and interior, initial, and boundary collocation points, define $R(\alpha)$ by stacking the corresponding constraint residuals, scaled by fixed positive weights and sampling normalizations. For instance, an equation residual contributes $\sqrt{\lambda_{\mathrm{eq}}/m_{\mathrm{eq}}}\,\mathcal{F}_\tau(u_\alpha; a)(q_i)$ when m_{eq} such scalar residuals are sampled. Writing $\widetilde{R}(\alpha)$ for the unweighted residual vector and D_w for the fixed diagonal matrix of square-root loss weights and sampling normalizations gives $R(\alpha) = D_w\widetilde{R}(\alpha)$ and $J_0 = D_w D\widetilde{R}(\alpha_0)$. The singular values in Proposition A.7 therefore reflect the physical residual sensitivities together with the loss weights and sampling normalizations. All required derivatives of u_α with respect to q are evaluated by automatic differentiation. The construction applies to a single autoregressive step with its input fixed; propagation across steps is treated in Appendix A.7. The Jacobian DR differentiates these physical residuals with respect to adapter parameters and therefore includes their effect on both solution values and coordinate derivatives.

For a fixed input A and fixed sensor positions, the Galerkin moments do not depend on the query coordinate, so a head satisfies

$$D_q^\beta\big[M_h(Z_\alpha(A))^\top u_h(q)\big] = M_h(Z_\alpha(A))^\top D_q^\beta u_h(q). \tag{23}$$

Thus Patch and Input Encoder adapters change the coefficients acting on the fixed query features and their derivatives. Fusion Block adapters also change the nonlinear maps applied to these features, with their contribution to coordinate derivatives determined by the chain rule. This relation connects the architectural factorization to expression-driven residual adaptation.

Let $J_0 = DR(\alpha_0)$ denote the residual Jacobian at the identity initialization.

Theorem A.6 (Local residual adaptation with a fixed backbone). *Suppose R is continuously differentiable on a neighborhood of $B_b = \{\alpha : \|\alpha - \alpha_0\|_2 \le b\}$ and its Jacobian is L_R-Lipschitz on B_b. Define $\eta_b = \min_{\|\Delta\|_2 \le b} \|R(\alpha_0) + J_0\Delta\|_2$. Then the best residual attainable within this adapter parameter ball satisfies*

$$\min_{\alpha \in B_b} \|R(\alpha)\|_2 \le \eta_b + \tfrac{1}{2}L_R b^2. \tag{24}$$

If $\|J_0^\dagger R(\alpha_0)\|_2 \le b$, then $\eta_b = \|(I - J_0J_0^\dagger)R(\alpha_0)\|_2$, where $\dagger$ denotes the Moore–Penrose pseudoinverse.

Proof. Write $R_0 = R(\alpha_0)$ and take any Δ with $\|\Delta\|_2 \le b$. The fundamental theorem of calculus and the Lipschitz assumption give

$$\begin{aligned} R(\alpha_0 + \Delta) &= R_0 + J_0\Delta + e(\Delta), \\ e(\Delta) &= \int_0^1 [DR(\alpha_0 + t\Delta) - J_0]\Delta \,\mathrm{d}t, \\ \|e(\Delta)\|_2 &\le \int_0^1 L_R t\|\Delta\|_2^2 \,\mathrm{d}t = \tfrac{1}{2}L_R\|\Delta\|_2^2. \end{aligned} \tag{25}$$

The linearized residual norm is continuous on the compact parameter ball, so a minimizer Δ_b defining η_b exists. Evaluating the nonlinear residual at this minimizer yields

$$\min_{\alpha \in B_b} \|R(\alpha)\|_2 \le \|R(\alpha_0 + \Delta_b)\|_2 \le \eta_b + \tfrac{1}{2}L_R b^2.$$

The unconstrained linear least-squares problem has minimum-norm solution $\Delta_* = -J_0^\dagger R_0$. If $\|\Delta_*\|_2 \le b$, it is also feasible for the constrained problem and leaves residual $(I - J_0J_0^\dagger)R_0$, proving the second statement. □

The bound relates local residual reduction to the initial residual, its alignment with the adapter update directions within radius b, and the nonlinear remainder $\frac{1}{2}L_R b^2$.

Proposition A.7 (Local linearized residual descent). *Let $R_0 = R(\alpha_0)$, $J_0 \neq 0$, and $P = J_0 J_0^\dagger$ be the orthogonal projector onto $\mathrm{range}(J_0)$. Consider gradient descent on the linearized objective $\phi(\Delta) = \frac{1}{2}\|R_0 + J_0\Delta\|_2^2$, initialized at $\Delta_0 = 0$, with step size $0 < \gamma \leq \|J_0\|_{\mathrm{op}}^{-2}$:*

$$\Delta_{k+1} = \Delta_k - \gamma J_0^\top (R_0 + J_0\Delta_k). \tag{26}$$

Writing $r_k = R_0 + J_0\Delta_k$ and letting $\sigma_{\min}^+$ denote the smallest nonzero singular value of J_0, for every $k \geq 1$ we have

$$\|r_k\|_2^2 \leq \|(I-P)R_0\|_2^2 + \left(1 - \gamma(\sigma_{\min}^+)^2\right)^{2k} \|PR_0\|_2^2. \tag{27}$$

Moreover, $r_k \to (I-P)R_0$ and $\Delta_k \to -J_0^\dagger R_0$. Under the assumptions of Theorem A.6, if $\|J_0^\dagger R_0\|_2 \leq b$, then all $\alpha_0 + \Delta_k$ lie in B_b and the nonlinear residual at these linearized iterates satisfies

$$\|R(\alpha_0 + \Delta_k)\|_2 \leq \left[\|(I-P)R_0\|_2^2 + \left(1 - \gamma(\sigma_{\min}^+)^2\right)^{2k} \|PR_0\|_2^2\right]^{1/2} + \tfrac{1}{2}L_R b^2. \tag{28}$$

Proof. Let $J_0 = U\Sigma V^\top$ be a compact singular value decomposition, with positive singular values σ_i. The recurrence gives $r_{k+1} = (I - \gamma J_0 J_0^\top) r_k$, hence

$$r_k = (I-P)R_0 + U(I - \gamma\Sigma^2)^k U^\top R_0.$$

The two terms are orthogonal, and $0 \leq 1 - \gamma\sigma_i^2 \leq 1 - \gamma(\sigma_{\min}^+)^2 < 1$, proving Eq. (27) and the residual limit. Solving the parameter recurrence from $\Delta_0 = 0$ yields

$$\Delta_k = -V\Sigma^{-1}\left[I - (I - \gamma\Sigma^2)^k\right] U^\top R_0.$$

Each diagonal factor in brackets lies in $[0,1]$ and tends to one, so $\Delta_k \to -J_0^\dagger R_0$ and $\|\Delta_k\|_2 \leq \|J_0^\dagger R_0\|_2$. If this last norm is at most b, the Taylor remainder in Eq. (25) gives $\|R(\alpha_0 + \Delta_k)\|_2 \leq \|r_k\|_2 + \frac{1}{2}L_R\|\Delta_k\|_2^2$, which proves Eq. (28). □

With identity-initialized adapters, the pre-trained predictor sets the initial residual R_0, while $\mathrm{range}(J_0)$ identifies the residual directions accessible to local adapter updates. Proposition A.7 quantifies how the nonzero singular values of this weighted Jacobian govern descent of the reducible residual component in the linearized model. With $\gamma = \|J_0\|_{\mathrm{op}}^{-2}$, the contraction factor is $1 - (\sigma_{\min}^+/\|J_0\|_{\mathrm{op}})^2$, linking linearized convergence to the conditioning on the accessible residual space.

Joint adapter groups. At the common identity initialization, partition the residual Jacobian as $J_0 = [J_{\mathrm{Patch}}, J_{\mathrm{IE}}, J_{\mathrm{FB}}]$. For a subset S of adapter groups, let J_S contain its columns and define $\eta_b(S) = \min_{\|\Delta_S\|_2 \leq b} \|R_0 + J_S\Delta_S\|_2$. If $S \subseteq T$, extending Δ_S by zeros preserves its norm and residual, hence

$$\eta_b(T) \leq \eta_b(S). \tag{29}$$

The local linearized residual can therefore benefit from the combined directions of multiple adapter groups. When the unconstrained least-squares solutions are feasible, a strict improvement occurs precisely when the additional span captures a nonzero component of the residual left by J_S. This characterization identifies residual alignment as the relevant quantity for joint adaptation. At identity initialization, the down-projection derivatives of a residual adapter vanish because its up-projection is zero, whereas the up-projection derivatives act on its initialized nonlinear features. Subsequent updates change these features and the local Jacobian; Theorem A.6 accounts for this nonlinearity through its remainder term.

Connection to continuous residuals and solution error. Let $\mathcal{R}_\tau(u_\alpha)$ denote the continuous equation and initial and boundary residuals in a normed space $\mathcal{Y}$. Suppose an admissible solution class satisfies a stability estimate and the chosen samples satisfy a uniform residual approximation bound on B_b:

$$\begin{aligned} \|u_\alpha - u_\tau^*\|_{\mathcal{X}} &\leq C_\tau \|\mathcal{R}_\tau(u_\alpha)\|_{\mathcal{Y}}, \\ \|\mathcal{R}_\tau(u_\alpha)\|_{\mathcal{Y}} &\leq \|R(\alpha)\|_2 + \varepsilon_{\mathrm{samp}}. \end{aligned} \tag{30}$$

For an attained parameter vector $\widehat{\alpha} \in B_b$ satisfying $\|R(\widehat{\alpha})\|_2 \le \min_{\alpha \in B_b} \|R(\alpha)\|_2 + \varepsilon_{\mathrm{opt}}$, Theorem A.6 then gives

$$\|u_{\widehat{\alpha}} - u_\tau^*\|_{\mathcal{X}} \le C_\tau \left(\eta_b + \tfrac{1}{2} L_R b^2 + \varepsilon_{\mathrm{opt}} + \varepsilon_{\mathrm{samp}}\right). \tag{31}$$

The estimate separates adapter accessibility, local nonlinearity, optimization accuracy, and sampling accuracy. For operator predictions, these assumptions can be imposed uniformly over admissible input functions, or in corresponding norms over an input distribution. Appendix A.8 provides a concrete stability estimate for a coercive parabolic subclass, and Appendix A.7 describes how a uniform one-step solution bound propagates through autoregressive prediction.

A.7 Autoregressive Error Propagation

This analysis relates one-step approximation error to accumulated prediction error over a finite autoregressive horizon. Let $\mathcal{F}_\tau$ and $\widehat{\mathcal{F}}_\tau$ map the current input representation to the next block of T_{pre} steps. The input represents the initial state of the current interval in expression-driven fine-tuning or the configured temporal window in data-driven training. Use the maximum norm over time and channels, and suppose all exact and predicted input representations belong to a common set $\mathcal{W}$. Assume

$$\|\mathcal{F}_\tau(W) - \mathcal{F}_\tau(V)\| \le L\|W - V\| \quad (W, V \in \mathcal{W}),$$
$$\sup_{W \in \mathcal{W}} \|\widehat{\mathcal{F}}_\tau(W) - \mathcal{F}_\tau(W)\| \le \varepsilon_1.$$

Let Roll select the most recent state or configured temporal window after appending the predicted block, and define $W_{n+1} = \mathsf{Roll}(W_n, \mathcal{F}_\tau(W_n))$ and $\widehat{W}_{n+1} = \mathsf{Roll}(\widehat{W}_n, \widehat{\mathcal{F}}_\tau(\widehat{W}_n))$. With $E_n = \|\widehat{W}_n - W_n\|$ and $\Lambda = \max\{1, L\}$,

$$\begin{aligned} E_{n+1} &\le \max\{E_n, LE_n + \varepsilon_1\}, \\ E_N &\le \Lambda^N E_0 + \varepsilon_1 \sum_{j=0}^{N-1} \Lambda^j, \qquad N = \lceil T_{\mathrm{out}}/T_{\mathrm{pre}} \rceil . \end{aligned} \tag{32}$$

Proof. The appended blocks satisfy

$$\begin{aligned} &\|\widehat{\mathcal{F}}_\tau(\widehat{W}_n) - \mathcal{F}_\tau(W_n)\| \\ &\le \|\widehat{\mathcal{F}}_\tau(\widehat{W}_n) - \mathcal{F}_\tau(\widehat{W}_n)\| + \|\mathcal{F}_\tau(\widehat{W}_n) - \mathcal{F}_\tau(W_n)\| \le \varepsilon_1 + LE_n. \end{aligned}$$

Restriction to the retained steps is nonexpansive in the maximum norm. Hence

$$\begin{aligned} E_{n+1} &\le \max\{E_n, LE_n + \varepsilon_1\} \le \Lambda E_n + \varepsilon_1, \\ E_N &\le \Lambda^k E_{N-k} + \varepsilon_1 \sum_{j=0}^{k-1} \Lambda^j \quad (1 \le k \le N). \end{aligned}$$

Taking $k = N$ gives the stated bound. □

The bound separates the propagation of initial input error from the accumulation of one-step approximation errors.

A.8 Residual-to-solution error in a well-posed subclass

Let Ω be a bounded Lipschitz domain and u solve $u_t - \nu\Delta u + cu = f$ with homogeneous Dirichlet data, where $\nu > 0$ and $c \ge 0$ are constants. For a prediction $\widetilde{u}$ satisfying the same boundary data, set $e = u - \widetilde{u}$ and $r = f - (\widetilde{u}_t - \nu\Delta\widetilde{u} + c\widetilde{u})$. Assume $e \in L^2(0,T; H_0^1(\Omega))$, $e_t \in L^2(0,T; H^{-1}(\Omega))$, and $r \in L^2(0,T; L^2(\Omega))$. Let $\lambda_1 > 0$ satisfy $\|\nabla v\|_{L^2}^2 \ge \lambda_1 \|v\|_{L^2}^2$ for $v \in H_0^1(\Omega)$. Then

$$\begin{aligned} \|e(t)\|_{L^2} &\le e^{-\lambda t}\|e(0)\|_{L^2} + \int_0^t e^{-\lambda(t-s)}\|r(s)\|_{L^2}\, ds, \\ \lambda &= \nu\lambda_1 + c > 0, \qquad 0 \le t \le T. \end{aligned} \tag{33}$$

Proof. Testing $e_t - \nu\Delta e + ce = r$ with e and applying Poincaré's inequality gives, almost everywhere in time,

$$\frac{1}{2}\frac{d}{dt}\|e\|_{L^2}^2 + \nu\|\nabla e\|_{L^2}^2 + c\|e\|_{L^2}^2 = \langle r, e\rangle,$$
$$\frac{1}{2}\frac{d}{dt}\|e\|_{L^2}^2 + \lambda\|e\|_{L^2}^2 \le \|r\|_{L^2}\|e\|_{L^2}.$$

For $\kappa > 0$, define $y_\kappa(t) = (\|e(t)\|_{L^2}^2 + \kappa^2)^{1/2}$. Since $y_\kappa \ge \kappa$,

$$y_\kappa' + \lambda y_\kappa \le \|r\|_{L^2}\frac{\|e\|_{L^2}}{y_\kappa} + \frac{\lambda\kappa^2}{y_\kappa} \le \|r\|_{L^2} + \lambda\kappa,$$
$$y_\kappa(t) \le e^{-\lambda t}y_\kappa(0) + \int_0^t e^{-\lambda(t-s)}\|r(s)\|_{L^2}\,ds + \kappa(1 - e^{-\lambda t}).$$

Letting $\kappa \downarrow 0$ proves the estimate. □

Nonhomogeneous boundary errors require a lifting and its corresponding residual estimate. A collocation loss controls the continuous residual only with an additional quadrature or sampling bound. Extension to nonlinear or shock-forming equations requires the corresponding stability or entropy estimates.

A.9 Adapter Dimension

Proposition A.8 (Correction subspace and approximation error). *For a fixed task τ, compact hidden-state set $C_{\tau,\ell}$, and continuous correction $\Delta_{\tau,\ell}$, define $d_r(\tau) = \inf_{\dim V \le r}\sup_{h\in C_{\tau,\ell}}\inf_{v\in V}\|\Delta_{\tau,\ell}(h) - v\|_2$. For $h \in \mathbb{R}^d$ and intermediate width m, a residual adapter $h \mapsto h + U\sigma(Wh + b) + c$ has its correction in $c + \mathrm{range}(U)$, whose affine dimension is at most $\min\{d, m\}$. For every $\zeta > 0$, a width-variable adapter constrained to an r-dimensional output subspace achieves uniform error below $d_r(\tau) + \zeta$. The hidden width may exceed r.*

Proof. Write $\Delta = \Delta_{\tau,\ell} : C_{\tau,\ell} \to \mathbb{R}^d$. Choose a subspace $V \subseteq \mathbb{R}^d$ of dimension $s \le r$ with $\sup_h \mathrm{dist}(\Delta(h), V) < d_r(\tau) + \zeta/2$. If $s = 0$, the zero correction already satisfies the claim. Otherwise let $U \in \mathbb{R}^{d\times s}$ have orthonormal columns spanning V. Then

$$P_V = UU^\top, \qquad c(h) = U^\top\Delta(h), \qquad \Delta(h) - Uc(h) = (I - P_V)\Delta(h).$$

The coordinate map c is continuous on the compact hidden-state set. The GELU approximation argument in Lemma A.4 gives a finite width m and a network $\widetilde{c}(h) = B\sigma(Wh + b) + v$ satisfying $\sup_h \|\widetilde{c}(h) - c(h)\|_2 < \zeta/2$. Therefore

$$\sup_h \|\Delta(h) - U\widetilde{c}(h)\|_2$$
$$\le \sup_h \|(I - P_V)\Delta(h)\|_2 + \|U\|_{\mathrm{op}}\sup_h \|c(h) - \widetilde{c}(h)\|_2 < d_r(\tau) + \zeta,$$
$$U\widetilde{c}(h) = (UB)\sigma(Wh + b) + Uv \in V, \qquad \mathrm{rank}(UB) \le s \le r.$$

Here m is the hidden bottleneck width, whereas r bounds the output subspace dimension; the construction does not require $m \le r$. For multiple nonlinear adapter layers, a single fixed correction subspace need not exist. □

For the implemented modules, the correction dimension is bounded by $\min\{d, r\}$ with the configured intermediate width r, which may exceed the feature width in some modules. The fixed-backbone residual analysis applies to these widths without imposing a strict low-rank constraint on every adapter.

A.10 Summary of the Theoretical Analysis

The analysis characterizes shared operator representations and expression-driven adaptation. Under the stated compactness and continuity assumptions, Theorem 3.1 establishes that a common query basis and task-specific coefficient maps can approximate a family of PDE solution operators. The

Galerkin factorization identifies how input features determine key–value moments and how Fusion Block adapters transform the resulting query-conditioned features.

For a given backbone and fixed adapter widths, Theorem A.6 bounds the best locally attainable residual under the stated differentiability and Jacobian regularity assumptions. Proposition A.7 gives a gradient-descent rate for the local linearized objective and connects its iterates to the nonlinear residual through the same Taylor remainder. At a common initialization and update radius, combining adapter groups enlarges the accessible linearized residual space, yielding the comparison in Eq. (29). Under the sampling and PDE stability assumptions in Eq. (30), Eq. (31) connects adaptation to solution accuracy through adapter accessibility, local nonlinearity, optimization accuracy, and sampling accuracy. The autoregressive analysis then quantifies how uniform one-step errors propagate under Lipschitz continuity on the reachable input representations.

B Details of Benchmark PDEs

For the FNO, PDEBench, and CFDBench benchmarks also evaluated by DPOT, we adopt the same dataset construction, preprocessing, data splits, and evaluation protocols. For expression-driven fine-tuning, the initial conditions specified below define the input function a, which is discretized at sensor coordinates X to form $a(X)$. The governing equations and prescribed initial and boundary conditions provide the training constraints.

Navier–Stokes Equations in Vorticity Form. For the NS2d benchmarks, we consider the two-dimensional incompressible Navier–Stokes equations on the unit torus $(0,1)^2$ with periodic boundary conditions. We adopt the *vorticity–stream function* formulation rather than the primitive velocity–pressure system. This reformulation is motivated by two considerations central to expression-driven operator learning: (i) applying the curl to the momentum equation eliminates the pressure gradient ∇p entirely, thereby removing the incompressibility constraint as an implicit algebraic condition and yielding a self-contained evolution system; (ii) the spatial derivatives of the stream function ψ recover the velocity components (u, v), enabling data-driven supervision on velocity fields and per-component visualization within our architecture.

Denoting the scalar vorticity by $\omega = \partial_x v - \partial_y u$ and the stream function by ψ, the governing system consists of the vorticity transport equation, the kinematic relation between vorticity and stream function, and the velocity recovery:

$$\frac{\partial \omega}{\partial t} + u\frac{\partial \omega}{\partial x} + v\frac{\partial \omega}{\partial y} = \nu\left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2}\right) + f, \tag{34}$$

$$\omega + \frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = 0, \tag{35}$$

$$u = \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x}, \tag{36}$$

where $\nu > 0$ is the kinematic viscosity and f is a prescribed time-independent external forcing that drives the flow. Note that the velocity field defined via Eq. (36) automatically satisfies the divergence-free condition $\partial_x u + \partial_y v = 0$, so incompressibility is structurally enforced without an additional constraint.

Since the equations are posed on the periodic torus $(0,1)^2$, no explicit boundary conditions need to be imposed. For expression-driven operator learning, the prescribed initial vorticity defines the input function a through

$$\omega(x, y, 0) = a(x, y), \qquad (x, y) \in (0,1)^2. \tag{37}$$

During expression-driven fine-tuning, the residuals in Eqs. (34)–(36) constrain the dynamics, while the initial-condition loss enforces Eq. (37).

Shallow Water Equations. For the SWE2d benchmark we adopt the two-dimensional nonlinear shallow water system in conservative form, following the radial dam-break configuration of PDEBench (Takamoto et al., 2022). Let $h(x, y, t) \geq 0$ denote the fluid height above the flat bottom

and $\mathbf{u} = (u, v)^\top$ the depth-averaged horizontal velocity. On a rectangular domain $\Omega \subset \mathbb{R}^2$ with reflective (no-flux) boundary conditions, the governing conservation laws read

$$\frac{\partial h}{\partial t} + \frac{\partial (hu)}{\partial x} + \frac{\partial (hv)}{\partial y} = 0, \tag{38}$$

$$\frac{\partial (hu)}{\partial t} + \frac{\partial}{\partial x}\left(hu^2 + \tfrac{1}{2}\, g\, h^2\right) + \frac{\partial (huv)}{\partial y} = 0, \tag{39}$$

$$\frac{\partial (hv)}{\partial t} + \frac{\partial (huv)}{\partial x} + \frac{\partial}{\partial y}\left(hv^2 + \tfrac{1}{2}\, g\, h^2\right) = 0, \tag{40}$$

where $g > 0$ is the gravitational acceleration (set to $g = 1$ in the benchmark). Equation (38) is the continuity equation expressing mass conservation, while Eqs. (39)–(40) encode conservation of momentum under the hydrostatic pressure assumption (whence the $\frac{1}{2}gh^2$ flux terms). We emphasise two structural features of this system that shape the operator learning problem. First, the flux Jacobian is genuinely non-linear in the conserved variables $(h, hu, hv)^\top$, which permits wave steepening and discontinuity formation even from smooth initial data—a qualitatively different regime from the viscosity-regularised NS2d flows. Second, in the absence of bottom-friction or topographic source terms the momentum equations are in *pure divergence form*, so each residual in Eqs. (38)–(40) admits a direct automatic-differentiation evaluation without further closure modelling.

The benchmark prescribes a radially symmetric dam-break initial condition: the fluid is at rest with a step profile in the height field,

$$h(x, y, 0) = \begin{cases} h_{\text{in}}, & r(x, y) \le r_0, \\ h_{\text{out}}, & r(x, y) > r_0, \end{cases} \qquad \mathbf{u}(x, y, 0) = \mathbf{0}, \tag{41}$$

where $r(x, y) = \sqrt{(x - x_c)^2 + (y - y_c)^2}$ and (x_c, y_c) is the domain centre. The release of the circular reservoir drives an outwardly propagating bore, producing sharp radial wavefronts that test the ability of an operator-learning model to track near-discontinuous dynamics.

In expression-driven fine-tuning, a is the prescribed initial height in Eq. (41), and its samples $a(X)$ form the encoder input. The governing-equation residuals and prescribed initial and boundary conditions constrain the predicted dynamics without using high-fidelity solution data.

Diffusion–Reaction Equations. For the DR benchmark, we follow the PDEBench `2D_diff-react_NA_NA` generator. The state consists of two scalar fields $u(x, y, t)$ and $v(x, y, t)$, which can be interpreted as two reacting species or concentrations, on the square domain $\Omega = [-1, 1]^2$. The governing equations represented by the finite-volume data generator are

$$\frac{\partial u}{\partial t} = \nu_u \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + u - u^3 - k - v, \tag{42}$$

$$\frac{\partial v}{\partial t} = \nu_v \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) + u - v, \tag{43}$$

where $\nu_u = 10^{-3}$ and $\nu_v = 5 \times 10^{-3}$ are the diffusion coefficients of the two species, and $k = 5 \times 10^{-3}$ is a constant reaction offset. Equivalently, the reaction functions are $R_u(u, v) = u - u^3 - k - v$ and $R_v(u, v) = u - v$. No additional external forcing or spatially varying material coefficient is used in this PDEBench subset.

The boundary condition is homogeneous Neumann, i.e. no diffusive flux leaves the domain:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial x} = 0 \quad \text{on } x = \pm 1, \qquad \frac{\partial u}{\partial y} = \frac{\partial v}{\partial y} = 0 \quad \text{on } y = \pm 1, \tag{44}$$

In the PDEBench implementation, this condition is encoded by modifying the boundary entries of the sparse finite-volume second-derivative stencil.

Each trajectory is initialized by sampling independent standard-normal values at all cell centres,

$$u_{ij}(0) \sim \mathcal{N}(0, 1), \qquad v_{ij}(0) \sim \mathcal{N}(0, 1), \qquad i, j = 1, \ldots, 128, \tag{45}$$

with independent samples for the two variables and a separate random seed for each trajectory. The original PDEBench trajectories are stored on a 128×128 cell-centred grid with $\Delta x = \Delta y = 0.015625$ over $t \in [0, 5]$, with 101 snapshots spaced by $\Delta t = 0.05$, and are resampled to 64×64 for our experiments. The metadata specifies a finite-volume spatial discretization and an RK45 time integrator, while the generator calls `solve_ivp` on the sampled time grid to produce the HDF5 trajectories.

Compressible Navier–Stokes Equations. For the PDEBench CNS benchmark, we consider the two-dimensional compressible Navier–Stokes equations on the unit square $\Omega = [0, 1]^2$. The primitive variables are the mass density $\rho(x, y, t)$, the velocity components $u(x, y, t)$ and $v(x, y, t)$, and the gas pressure $p(x, y, t)$. The total energy density is defined by the ideal-gas closure

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho\left(u^2 + v^2\right), \tag{46}$$

and is therefore not an additional unknown. The benchmark system contains four scalar governing equations for the four unknown fields (ρ, u, v, p): mass conservation, two momentum equations, and total-energy conservation. Written explicitly in partial-derivative form, the equations are

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} = 0, \tag{47}$$

$$\rho\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}\right) = -\frac{\partial p}{\partial x} + \eta\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + \left(\zeta + \frac{\eta}{3}\right)\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 v}{\partial x\,\partial y}\right), \tag{48}$$

$$\rho\left(\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y}\right) = -\frac{\partial p}{\partial y} + \eta\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) + \left(\zeta + \frac{\eta}{3}\right)\left(\frac{\partial^2 u}{\partial x\,\partial y} + \frac{\partial^2 v}{\partial y^2}\right), \tag{49}$$

$$\frac{\partial E}{\partial t} + \frac{\partial}{\partial x}\left((E + p)u - u\tau_{xx} - v\tau_{xy}\right) + \frac{\partial}{\partial y}\left((E + p)v - u\tau_{xy} - v\tau_{yy}\right) = 0, \tag{50}$$

where the viscous stress components in Eq. (50) are

$$\begin{aligned}
\tau_{xx} &= 2\eta\frac{\partial u}{\partial x} + \left(\zeta - \frac{2\eta}{3}\right)\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right), \\
\tau_{xy} &= \eta\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right), \\
\tau_{yy} &= 2\eta\frac{\partial v}{\partial y} + \left(\zeta - \frac{2\eta}{3}\right)\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right).
\end{aligned} \tag{51}$$

Here η and ζ denote the shear and bulk viscosities, respectively, and γ is the adiabatic index of the ideal gas. In the PDEBench CNS subsets used in our experiments, $\gamma = 5/3$; different datasets are generated by varying the Mach number M and the viscosity setting (η, ζ), with the released subsets we use taking $\eta = \zeta$.

The boundary condition is periodic in both spatial directions. For every primitive variable $q \in \{\rho, u, v, p\}$,

$$q(0, y, t) = q(1, y, t), \qquad q(x, 0, t) = q(x, 1, t), \qquad (x, y) \in \Omega. \tag{52}$$

The initial condition is sampled from the PDEBench random-field generator. Let $\Phi_\rho, \Phi_p, \Phi_u, \Phi_v$ be independent random Fourier fields on Ω, normalized by spatial averages. A representative trajectory is initialized as

$$\rho(x, y, 0) = \rho_0\left(1 + \delta_\rho\frac{\Phi_\rho(x, y)}{\langle|\Phi_\rho|\rangle}\right), \qquad p(x, y, 0) = p_0\left(1 + \delta_p\frac{\Phi_p(x, y)}{\langle|\Phi_p|\rangle}\right), \tag{53}$$

$$u(x, y, 0) = U_0\frac{\Phi_u(x, y)}{\left\langle\sqrt{\Phi_u^2 + \Phi_v^2}\right\rangle}, \qquad v(x, y, 0) = U_0\frac{\Phi_v(x, y)}{\left\langle\sqrt{\Phi_u^2 + \Phi_v^2}\right\rangle}, \tag{54}$$

where $p_0 = \rho_0 T_0$, $c_s = \sqrt{\gamma T_0}$, and $U_0 = M c_s$. Thus the Mach number M is not a material coefficient in Eqs. (47)–(50); instead, it controls the amplitude of the initial velocity field and hence the compressibility level of the generated trajectory. The original PDEBench CNS trajectories are stored on a 128×128 grid over $t \in [0, 1]$, with 21 snapshots spaced by $\Delta t = 0.05$, and are resampled to 64×64 for our experiments.

Three-dimensional Compressible Navier–Stokes Equations. We consider the three-dimensional compressible Navier–Stokes system on the periodic unit cube $\Omega = [0,1]^3$ for $t \in [0,1]$, following PDEBench (Takamoto et al., 2022). Let $\mathbf{x} = (x, y, z)$, $\mathbf{v} = (u, v, w)^\top$, and let ρ and p denote density and pressure. The shear and bulk viscosities are $\eta = \zeta = 10^{-8}$, and the nominal initial Mach number is $M = 0.1$. Conservation of mass, momentum, and total energy gives

$$\partial_t \rho + \partial_x(\rho u) + \partial_y(\rho v) + \partial_z(\rho w) = 0, \tag{55}$$

$$\rho(\partial_t u + u\partial_x u + v\partial_y u + w\partial_z u) = -\partial_x p + \eta\Delta u + (\zeta + \eta/3)\partial_x d, \tag{56}$$

$$\rho(\partial_t v + u\partial_x v + v\partial_y v + w\partial_z v) = -\partial_y p + \eta\Delta v + (\zeta + \eta/3)\partial_y d, \tag{57}$$

$$\rho(\partial_t w + u\partial_x w + v\partial_y w + w\partial_z w) = -\partial_z p + \eta\Delta w + (\zeta + \eta/3)\partial_z d, \tag{58}$$

$$\partial_t E + \sum_{j=1}^{3} \partial_{x_j}\left[(E+p)v_j - \sum_{i=1}^{3} v_i \tau_{ij}\right] = 0, \tag{59}$$

Here $d = \nabla \cdot \mathbf{v}$, $\Delta = \partial_{xx} + \partial_{yy} + \partial_{zz}$, $(x_1, x_2, x_3) = (x, y, z)$, and $(v_1, v_2, v_3) = (u, v, w)$. The ideal-gas closure and Newtonian viscous stress are

$$E = \frac{p}{\gamma - 1} + \frac{\rho}{2}(u^2 + v^2 + w^2), \qquad \tau_{ij} = \eta(\partial_{x_j} v_i + \partial_{x_i} v_j) + (\zeta - 2\eta/3) d\, \delta_{ij}, \qquad \gamma = \frac{5}{3}. \tag{60}$$

Here E denotes the total energy density. Body forces and heat conduction are neglected. All primitive fields satisfy periodic conditions in each spatial direction: for $q \in \{\rho, u, v, w, p\}$,

$$q(0, y, z, t) = q(1, y, z, t), \quad q(x, 0, z, t) = q(x, 1, z, t), \quad q(x, y, 0, t) = q(x, y, 1, t). \tag{61}$$

Initial conditions follow the PDEBench random-field construction. For $a \in \{\rho, p, 1, 2, 3\}$, define

$$f_a(\mathbf{x}) = \sum_{\mathbf{k} \in \mathcal{K}} \frac{\sin(2\pi \mathbf{k} \cdot \mathbf{x} + \phi_{a,\mathbf{k}})}{2\pi \|\mathbf{k}\|_2}, \qquad \mathcal{K} = \{\mathbf{k} \in \mathbb{Z}^3 : 1 \le |k_j| \le 4,\ j = 1, 2, 3\}, \tag{62}$$

where the phases are sampled uniformly from $[0, 2\pi)$. With $\mathbf{f} = (f_1, f_2, f_3)^\top$ and $\langle \cdot \rangle$ denoting the spatial grid average, the initial fields are

$$\begin{aligned} \rho(\mathbf{x}, 0) &= \rho_0 \left(1 + W(\mathbf{x})\delta_\rho \frac{f_\rho(\mathbf{x})}{\langle |f_\rho| \rangle}\right), \qquad p(\mathbf{x}, 0) = p_0 \left(1 + W(\mathbf{x})\delta_p \frac{f_p(\mathbf{x})}{\langle |f_p| \rangle}\right), \\ \mathbf{v}(\mathbf{x}, 0) &= W(\mathbf{x}) M \sqrt{\gamma T_0}\, \frac{\mathbf{f}(\mathbf{x})}{\langle \|\mathbf{f}\|_2 \rangle}, \qquad p_0 = \rho_0 T_0. \end{aligned} \tag{63}$$

Here $\rho_0, T_0 \sim \mathcal{U}(0.1, 10)$ and $\delta_\rho, \delta_p \sim \mathcal{U}(0.01, 0.2)$. With equal probability, $W \equiv 1$ or

$$\begin{aligned} W(\mathbf{x}) &= \prod_{j=1}^{3} \frac{1}{2}\left[\tanh\left(\frac{x_j - \ell_j}{0.01}\right) - \tanh\left(\frac{x_j - r_j}{0.01}\right)\right], \\ \ell_j &\sim \mathcal{U}(0.1, 0.45), \qquad r_j \sim \mathcal{U}(0.55, 0.9). \end{aligned} \tag{64}$$

Thus $M = 0.1$ specifies the mean-speed Mach number before windowing.

The three-dimensional experiments use 90 training and 10 test trajectories to compare training from scratch and fine-tuning after two-dimensional pre-training under both data-driven and expression-driven supervision. For both fine-tuning regimes, each method reuses its pre-trained model from the main experiments, trained on the eight datasets corresponding to the first eight dataset columns of Table 2: the three FNO viscosity regimes and the five PDEBench CNS configurations. For fine-tuning, all parameters compatible with each method's three-dimensional architecture are loaded from its two-dimensional pre-trained checkpoint. In both fine-tuning regimes, each method retains the parameter update strategy used in the multi-PDE transfer experiments in Section 4.1. Methods using full-model fine-tuning update all parameters, whereas methods with restricted updates retain their original trainable parameter subsets and additionally optimize all new parameters introduced by extending the model from two to three spatial dimensions, including new non-adapter parameters. Expression-driven fine-tuning retains each method's loss formulation and derivative computation from the main experiments, with the parameter update scope specified above. As in the two-dimensional experiments, expression-driven fine-tuning takes the discretized initial-condition function as input and uses the governing equations and prescribed initial and boundary conditions for supervision.

CFDBench Flow Problems. We follow the CFDBench construction used by DPOT (Hao et al., 2024): the cavity, tube, and cylinder trajectories, including their property-, boundary-condition-, and geometry-varying subsets, are combined into one CFDBench task and evaluated with a single aggregate L2RE. For data-driven training, the released DPOT preprocessing partitions each trajectory into 20-frame segments, resamples every segment to a 64×64 grid, represents each input frame by the two velocity components and a geometry mask, and evaluates the two predicted velocity channels with one aggregate L2RE. The dam problem is not part of this DPOT-compatible aggregate.

All three problems use the benchmark-level two-dimensional incompressible Newtonian-flow model specified by CFDBench (Luo et al., 2023). Let $\mathbf{u} = (u, v)^\top$ be the velocity, p the pressure, $\rho > 0$ the constant density, and $\mu > 0$ the dynamic viscosity for one case. On the case-dependent fluid domain Ω, the governing equations are

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \tag{65}$$

$$\rho\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}\right) = -\frac{\partial p}{\partial x} + \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right), \tag{66}$$

$$\rho\left(\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y}\right) = -\frac{\partial p}{\partial y} + \mu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right). \tag{67}$$

The prescribed initial velocity defines the input function a through

$$\mathbf{u}(x, y, 0) = a(x, y), \qquad (x, y) \in \Omega, \tag{68}$$

Expression-driven fine-tuning uses $a(X)$ at sensor coordinates X as the encoder input and enforces the prescribed initial and boundary conditions. For expression-driven fine-tuning, Eqs. (65)–(67) are evaluated only at collocation points inside the geometry-defined fluid domain. The auxiliary pressure channel is used in the momentum and pressure-outlet residuals but is excluded from the velocity L2RE, matching DPOT's reported CFDBench quantity.

Lid-driven cavity. For a cavity of width L_c and height H_c, the fluid occupies the rectangle

$$\Omega_c = (0, L_c) \times (0, H_c). \tag{69}$$

The geometry subset varies (L_c, H_c), while the BC subset varies the top-wall speed U_{top}. Writing $\Gamma_{\text{top}} = \{(x, H_c) : 0 < x < L_c\}$ and $\Gamma_{\text{wall}} = \partial\Omega_c \setminus \Gamma_{\text{top}}$, the no-penetration/no-slip conditions are

$$\mathbf{u} = (U_{\text{top}}, 0)^\top \quad \text{on } \Gamma_{\text{top}}, \qquad \mathbf{u} = \mathbf{0} \quad \text{on } \Gamma_{\text{wall}}. \tag{70}$$

Because only ∇p enters the momentum equations, the pressure gauge is fixed by $\int_{\Omega_c} p \, \mathrm{d}x\mathrm{d}y = 0$.

Tube flow. The tube is represented by the two-dimensional channel

$$\Omega_t = (0, L_t) \times (0, H_t), \tag{71}$$

where the geometry subset varies the channel length and height (equivalently, its aspect ratio). Define the inlet, outlet, and wall boundaries by

$$\Gamma^t_{\text{in}} = \{0\} \times (0, H_t), \qquad \Gamma^t_{\text{out}} = \{L_t\} \times (0, H_t), \qquad \Gamma^t_{\text{wall}} = (0, L_t) \times \{0, H_t\}.$$

The benchmark prescribes a uniform velocity inlet, stationary no-slip walls, and a zero-gauge-pressure outlet:

$$\mathbf{u} = (U_{\text{in}}, 0)^\top \qquad \text{on } \Gamma^t_{\text{in}}, \tag{72}$$

$$\mathbf{u} = \mathbf{0} \qquad \text{on } \Gamma^t_{\text{wall}}, \tag{73}$$

$$p = p_{\text{out}} = 0 \qquad \text{on } \Gamma^t_{\text{out}}. \tag{74}$$

Here and below, the released cases use zero gauge pressure at the outlet.

Flow around a cylinder. After translating the coordinate system so that the cylinder centre is the origin, let L_{in}, L_{out}, H_b, and H_t denote its distances to the inlet, outlet, bottom wall, and top wall, respectively, and let r be the cylinder radius supplied in the case metadata. The fluid domain is the rectangular channel with the closed circular obstacle removed:

$$\Omega_{\text{cyl}} = ((-L_{\text{in}}, L_{\text{out}}) \times (-H_b, H_t)) \setminus \overline{B_r(0, 0)}. \tag{75}$$

Thus the geometry subset changes both the outer channel dimensions and the size of the internal curved boundary. With

$$\Gamma^c_{\text{in}} = \{-L_{\text{in}}\} \times (-H_b, H_t), \qquad \Gamma^c_{\text{out}} = \{L_{\text{out}}\} \times (-H_b, H_t),$$
$$\Gamma^c_{\text{wall}} = (-L_{\text{in}}, L_{\text{out}}) \times \{-H_b, H_t\}, \qquad \Gamma^c_{\text{obs}} = \partial B_r(0,0),$$

the boundary conditions are

$$\mathbf{u} = (U_{\text{in}}, 0)^\top \quad \text{on } \Gamma^c_{\text{in}}, \tag{76}$$
$$\mathbf{u} = \mathbf{0} \quad \text{on } \Gamma^c_{\text{wall}} \cup \Gamma^c_{\text{obs}}, \tag{77}$$
$$p = p_{\text{out}} = 0 \quad \text{on } \Gamma^c_{\text{out}}. \tag{78}$$

The geometry mask supplied to every model distinguishes the circular obstacle from the fluid region; the PDE residual is imposed only in the fluid interior, while the reported velocity error follows the aggregate DPOT evaluation described above.

C L_2 Relative Error as the Data-Driven Metric

All data-driven terms appearing in this paper are measured with the L_2 *relative error* operator L2RE rather than with the squared Euclidean distance. We define L2RE at the granularity of a single (trajectory, time, channel) triplet and then average.

Definition. Consider a prediction $\hat{u} \in \mathbb{R}^{B\times n\times C}$ and a ground-truth reference $u \in \mathbb{R}^{B\times n\times C}$, where the n spatio-temporal samples are the Cartesian product of n_s spatial grid points and T time steps, i.e. $n = n_s\,T$. After reshaping both tensors to the separated layout $\mathbb{R}^{B\times n_s\times T\times C}$ and applying a binary validity mask $M \in \{0,1\}^{B\times n_s\times T\times C}$ (which gates padded positions), the per-triplet relative error is

$$e_{b,t,c}(\hat{u}, u) \;\coloneqq\; \sqrt{\frac{\displaystyle\sum_{i=1}^{n_s} M_{b,i,t,c}\big(\hat{u}_{b,i,t,c} - u_{b,i,t,c}\big)^2}{\displaystyle\sum_{i=1}^{n_s} M_{b,i,t,c}\,u^2_{b,i,t,c} \;+\; \varepsilon}}, \qquad \varepsilon = 10^{-8}, \tag{79}$$

and the L_2 relative error is the masked arithmetic mean over all active triplets,

$$\text{L2RE}(\hat{u}, u) \;\coloneqq\; \frac{1}{|\mathcal{T}|} \sum_{(b,t,c)\,\in\,\mathcal{T}} e_{b,t,c}(\hat{u}, u), \qquad \mathcal{T} \;=\; \big\{(b,t,c) \,:\, \textstyle\sum_i M_{b,i,t,c} > 0\big\}. \tag{80}$$

Rationale for L2RE. In multi-PDE pre-training, an unnormalized squared-error loss can be dominated by fields with larger solution amplitudes. The normalization in Eq. (79) reduces this sensitivity by measuring prediction errors relative to the corresponding solution norms. Averaging these relative errors over trajectories, time steps, and channels in Eq. (80) gives each active triplet equal weight in the aggregate metric.

Relation to the expression-driven term. The governing-equation residual $\|\mathcal{F}_k(\cdot)\|^2$ is evaluated in the dimensional units of the prescribed equation during expression-driven fine-tuning, together with the boundary-condition and initial-condition constraints in Eq. (9).

Numerical remark. The $\varepsilon = 10^{-8}$ regulariser in Eq. (79) protects the denominator against triplets where the reference solution happens to be identically zero (e.g. quiescent initial conditions of SWE2d). Triplets whose validity mask sums to zero are excluded from the outer average via the set $\mathcal{T}$ in Eq. (80), ensuring that padded or masked-out coordinates do not bias the loss.

D Pre-training-Data Scaling

To measure how reducing the amount of pre-training data affects performance, we fix the SymOnet architecture at 100M parameters and construct progressively larger training subsets containing 12.5%,

25%, 50%, and 100% of the trajectories from each of the eight pre-training datasets. The smaller subsets are contained in the larger ones, and all variants are pre-trained for 100 epochs and use the same expression-driven fine-tuning protocol on the unseen CNS parameter regime, DR, and SWE.

Table 3 shows that, at fixed model capacity, the L2RE decreases monotonically with the amount of multi-PDE pre-training data on all three OOD tasks. Increasing the data fraction from 12.5% to 100% reduces L2RE by 50.7% on the unseen CNS parameter regime, 53.5% on DR, and 59.5% on SWE, demonstrating improved cross-parameter and cross-family transfer.

E Training Details

SymOnet architecture. SymOnet-100M uses $d_{\text{in}} = 576$, $d_{\text{latent}} = 384$, four input encoder layers with four attention heads per layer, patch size 8, and four output channels. SymOnet-500M increases the embedding dimensions to $d_{\text{in}} = 1728$ and $d_{\text{latent}} = 1152$ and the input encoder depth to eight layers, retaining four attention heads per layer and patch size 8. SymOnet-500M uses convolutional patch embedding, GeGELU activations in the attention feed-forward blocks, and a Gaussian Fourier-feature scale of 8 in the Query Encoder. For both SymOnet variants, all adapters are disabled during pre-training and enabled during fine-tuning. All models are trained and evaluated on a common 64×64 spatial grid in the two-dimensional experiments.

Adapter configuration and parameter counts. Table 6 summarizes the adapter configurations for the two-dimensional SymOnet-100M and SymOnet-500M models. Within each model, all adapters share the bottleneck width r in Eq. (10), set to 352 for SymOnet-100M and 328 for SymOnet-500M. The Patch Embedding contains two 1×1 convolutional adapters, one after each convolution, acting on feature maps with 32 and d_{in} channels, respectively. Each Input Encoder layer contains two linear adapters, placed after the self-attention and feed-forward residual updates, while the Fusion Block contains two linear adapters, placed after its cross-attention and feed-forward residual updates. The linear adapters operate at feature widths d_{in} in the Input Encoder and $d_{\text{latent}}/2$ in the Fusion Block. All adapters use GELU activation, a fixed residual scale $s = 1$, and zero-initialized up-projection weights and biases. Including both projection biases, each adapter with feature width d and bottleneck width r contains $2dr + d + r$ parameters. The same intermediate width is used across modules with different feature dimensions, and parameter efficiency is measured by the aggregate adapter count relative to the backbone.

Table 6: Adapter configurations and parameter counts for the two-dimensional SymOnet variants.

Configuration	SymOnet-100M	SymOnet-500M
Input embedding width d_{in}	576	1,728
Latent width d_{latent}	384	1,152
Input Encoder depth	4	8
Adapter bottleneck width r	352	328
Patch Embedding adapter parameters	429,344	1,156,976
Input Encoder adapter parameters	3,251,456	18,169,984
Fusion Block adapter parameters	271,424	757,520
Total adapter parameters	3,952,224	20,084,480

Unless otherwise stated, SymOnet is trained with the AdamW optimizer, momentum parameters $(\beta_1, \beta_2) = (0.9, 0.9)$, and weight decay 10^{-6}. For pre-training, we use an initial learning rate of 10^{-3} and train for 100 epochs with a cycle learning-rate schedule, where the first 10 epochs are used for warm-up. For downstream fine-tuning, we use an initial learning rate of 10^{-3} and train for 200 epochs with a plateau schedule, using decay factor 0.5 and threshold 0.005. For SymOnet, the patch size is fixed to 8, with $T_{\text{pre}} = 1$ and $T_{\text{out}} = 5$. Data-driven pre-training and fine-tuning use $T_{\text{in}} = 10$, taking the first ten time steps of each input–output sample as the initial history. Expression-driven fine-tuning instead takes $a(X)$, the discretization of the prescribed initial-condition function at sensor coordinates X, as its input. All training runs are conducted on eight A800 GPUs, each with 80 GB memory. SymOnet uses a batch size of 64 trajectories.

Information available during expression-driven adaptation. Each downstream problem is specified by its governing equation, coefficients, geometry, and initial and boundary conditions. The prescribed initial function supplies $a(X)$ for both the encoder input and the initial-condition constraint; subsequent autoregressive inputs are formed from predicted states. All expression-driven methods are trained solely from this problem information, without access to high-fidelity solution data.

Computational cost. At matched model size, expression-driven fine-tuning based on automatic differentiation requires approximately $1.5\times$ the wall-clock time and $2\times$ the peak GPU memory of data-driven fine-tuning, because computing PDE derivatives requires retaining additional computation graphs.

Dataset sizes. Table 7 reports the numbers of training and test trajectories for every dataset in the main experiments. To reduce sampling imbalance across datasets of different sizes, we use the integer sampling weights in Table 7; a weight of w repeats each training trajectory w times in the multi-dataset sampler. Single-dataset fine-tuning uses the same split as the corresponding dataset in pre-training.

Table 7: Dataset sizes and sampling weights used in the main experiments.

Statistic	FNO-ν			PDEBench CNS (M, η)						PDEBench		CFDBench
	10^{-5}	10^{-4}	10^{-3}	$(0.1, 10^{-1})$	$(0.1, 10^{-2})$	$(0.1, 10^{-8})$	$(1, 10^{-1})$	$(1, 10^{-2})$	$(1, 10^{-8})$	DR	SWE	–
Train	1000	5000	4800	2700	2700	900	2700	2700	900	900	900	5000
Sampling weight	5	1	1	2	2	6	2	2	6	6	6	1
Test	200	1000	200	60	60	60	60	60	60	60	60	1000

Physics-informed sampling and loss weights for the main experiments. For SymOnet, the weights of the data, governing-equation, boundary-condition, and initial-condition losses are 1000, 1, 10, and 100, respectively, whenever the corresponding loss is active. For SymOnet in Table 2, we sample 1000 interior points per trajectory and autoregressive rollout step to evaluate the governing-equation residuals. The initial-condition loss compares predictions at sensor coordinates X on the initial surface with the prescribed values $a(X)$, as in Eq. (9). The data-loss weight is zero during expression-driven fine-tuning, so the objective uses only governing-equation, initial-condition, and boundary-condition constraints. For the non-periodic downstream benchmarks, we sample 508, 252, and 252 boundary-condition points for PDEBench DR, PDEBench SWE, and CFDBench, respectively; the periodic FNO-ν and PDEBench CNS problems require no separately sampled boundary points.

Baseline architectures and hyperparameters. We detail the architectural and training hyperparameters of the six baselines for the two-dimensional experiments, each configured with approximately 100M parameters.

DPOT. DPOT (Hao et al., 2024) uses 8×8 patches, embedding width 1864, and five Fourier operator layers, each with four channel blocks and four retained Fourier modes along each spatial axis. The MLP expansion ratio is 1, the decoder intermediate width is 56, and the operator blocks use GELU activations and GroupNorm with eight groups. In data-driven training, the exponential MLP temporal aggregator combines ten input frames; input normalization is disabled.

F-Adapter. F-Adapter (Zhang et al., 2025) uses the same pretrained DPOT backbone, with base adapter width 32 and residual scale 1. The frequency allocation uses four bands, exponent 1.5, and minimum and maximum width factors of 0.75 and 2.25, yielding bottleneck widths $(72, 55, 40, 30)$ from low to high frequencies. Spectral adapters are inserted at the input, intermediate, and output stages of each Fourier channel block; the patch-embedding and MLP adapters have bottleneck width 32. Only adapter parameters are updated during fine-tuning, with the pretrained backbone frozen.

MoE-POT. MoE-POT (Wang et al., 2025) uses embedding width 1376 and four Fourier operator layers, with patch size 8, four Fourier channel blocks, four retained modes per spatial axis, MLP expansion ratio 1, and decoder intermediate width 56. Each mixture layer contains two routed experts and one shared expert, with top-1 routing and an auxiliary routing-loss weight of 0.1. The model uses GELU activations, eight-group GroupNorm, and exponential MLP temporal aggregation, with ten input frames in data-driven training and no input normalization. In our configuration, both the router and expert parameters remain trainable during fine-tuning.

POSEIDON. POSEIDON (Herde et al., 2024) uses a four-level ScOT hierarchy with embedding widths $(75, 150, 300, 600)$, transformer depths $(9, 10, 9, 8)$, and attention-head counts $(3, 6, 12, 24)$, listed from the finest to the coarsest level. The decoder mirrors these levels in reverse order, and the skip connections contain $(2, 2, 2, 0)$ ConvNeXt blocks. We use patch size 4, nominal attention-window size 16, MLP expansion ratio 4, GELU activations, and layer-normalization epsilon 10^{-5}. Time conditioning and query/key/value biases are enabled, while absolute positional embeddings, dropout, and stochastic depth are disabled. At each autoregressive step, ScOT receives the most recent field and a lead-time input of 1.

WALRUS. Our WALRUS implementation (McCabe et al., 2026) uses 12 processor blocks of width 768, each applying spatial attention, causal temporal attention, and a SwiGLU channel MLP with hidden width 1536. Both attention modules use 12 heads, and RMSGroupNorm uses 12 groups with epsilon 10^{-5}. The encoder and decoder use a convolution and a transposed convolution, respectively, both with kernel size and stride 8. The model uses learned spatial positional embeddings and relative temporal biases, with a maximum supported history length of 16 and ten input frames in data-driven training. Dropout is disabled, per-trajectory RMS normalization uses epsilon 10^{-6}, and the output represents the increment from the most recent state. Patch jittering is enabled for CNS and CFDBench and disabled for the FNO-ν, DR, and SWE configurations.

PI-DeepONet FM. We follow the multi-operator pre-training and physics-informed LoRA fine-tuning framework of PI-DeepONet FM (Zhang et al., 2026), with the architecture and adaptation settings specified below for the common benchmark. Pre-training uses a shared branch network and dataset-specific trunks. Our implementation uses 512 basis features, with four linear layers in both the branch and trunk networks, GELU activations, and no dropout. The branch and trunk hidden widths are fixed at 588 and 320, respectively, across all datasets. The branch encodes inputs on the 64×64 grid, using ten frames in data-driven training and the discretized initial-condition function in expression-driven fine-tuning, while the trunk takes three-dimensional space–time coordinates. The elementwise product of the branch and trunk features is scaled by $1/\sqrt{512}$ and passed through a width-512 feature projection and a shared output MLP with hidden widths $(256, 256)$. The pre-trained branch is retained, and the averaged parameters of the dataset-specific trunks initialize the downstream trunk. The CNS, DR, and FNO-ν/SWE fine-tuning configurations use trunk LoRA factors with dimension $r = 1400$ and scaling parameter $\alpha = 1400$. For a layer with weight matrix $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, the update is $\Delta W = (\alpha/r)BA$, where $B \in \mathbb{R}^{d_{\text{out}} \times r}$ and $A \in \mathbb{R}^{r \times d_{\text{in}}}$, so r denotes the factor dimension and $\operatorname{rank}(\Delta W) \leq \min\{d_{\text{out}}, d_{\text{in}}, r\}$. We freeze the branch and the trunk's pre-trained weights, and optimize the trunk LoRA factors, feature projection, and output heads. The same trainable modules are used in both fine-tuning regimes, with their respective input specifications and objectives defined above.

Baseline optimization. The baseline configurations use Adam with $(\beta_1, \beta_2) = (0.9, 0.9)$, weight decay 10^{-6}, and gradient-norm clipping at 10^4. Pre-training runs for 100 epochs with a one-cycle schedule, peak learning rate 10^{-3}, ten warm-up epochs, and initial and final division factors of 10^4. Fine-tuning runs for 200 epochs from learning rate 10^{-3}, using a plateau scheduler with decay factor 0.5, patience 10, and relative improvement threshold 0.005. All baselines use a batch size of 64 throughout pre-training and fine-tuning. F-Adapter reuses DPOT pre-training, and the PIFT variants retain the architecture and optimizer settings of their corresponding baselines.

Table 8 summarizes the expression-driven fine-tuning protocols used in the main two-dimensional comparison. All methods use 100 pre-training epochs and 200 fine-tuning epochs with batch size 64; F-Adapter reuses the pre-trained DPOT checkpoint. The comparison matches total model scale and uses a common training schedule, with each method's parameter update rule specified explicitly. For each training run, we compute test L2RE using the checkpoint with the lowest training loss. L2RE in Tables 2, 3, and 5 and Figure 2 is averaged over five random seeds. The SymOnet adapter parameter counts for the two model scales are given in Table 6.

Table 8: Expression-driven fine-tuning protocols for the main two-dimensional comparison. AD and FD denote automatic differentiation and finite differences, respectively.

Method	Derivative computation	Parameters updated during fine-tuning
DPOT-PIFT	FD	All model parameters
MoE-POT-PIFT	FD	All model parameters
POSEIDON-PIFT	FD	All model parameters
WALRUS-PIFT	FD	All model parameters
PI-DeepONet FM	AD	Trunk LoRA factors, feature projection, and output heads
SymOnet	AD	Patch Embedding, Input Encoder, and Fusion Block adapters

SymOnet-500M follows the same architecture and training protocol as SymOnet-100M and represents the larger-capacity regime beyond the matched-scale comparison. For the model-size scaling study in Figure 2, DPOT-PIFT, MoE-POT-PIFT, POSEIDON-PIFT, WALRUS-PIFT, PI-DeepONet FM, and SymOnet are instantiated at approximately 10M, 50M, 100M, and 500M total parameters according to their respective architectural scaling rules. We randomly sample 25% of the trajectories from each pre-training dataset and use the same subsets for all six methods and four model scales. Within each model family, the optimizer, learning-rate schedule, training length, downstream tasks, and expression-driven fine-tuning protocol remain fixed across parameter scales, with only the scale-defining architectural hyperparameters changed. The PIFT variants use finite-difference residuals at every scale, while PI-DeepONet FM and SymOnet use automatic-differentiation residuals.

F Ablation Details

The controlled ablations use the same approximately 100M total-parameter scale as the main experiments. SymOnet uses $d_{\text{in}} = 576$, $d_{\text{latent}} = 384$, four encoder layers with four heads, patch size 8, and four output channels. These settings are held fixed across the ablations, while each parameter-matched replacement is adjusted to preserve the total model scale.

All downstream updates use only the governing equation and prescribed initial and boundary conditions, with $a(X)$ as the input and no high-fidelity solution data. CNS-OOD denotes CNS $(M, \eta) = (1, 10^{-8})$, while DR and SWE represent PDE families absent from pre-training. Within each comparison, we fix the dataset splits, pre-training objective and budget, downstream loss weights, collocation-point budgets, optimizer, learning-rate schedule, and training length.

For architecture ablations, each row in Table 5 replaces only the named component with a parameter-matched alternative and is pretrained independently. Downstream fine-tuning updates all adapter groups present in each variant while freezing the backbone, so the adaptation rule is held fixed across architectural replacements. For adaptation ablations, all variants retain the full architecture and share the pretrained initialization within each comparison. Single-group variants update only the named adapter group, leave-one-out variants freeze that group and update the other two, and the all-adapter variant updates all three groups while freezing the backbone. The frozen reference performs no parameter updates, whereas full-model fine-tuning updates all parameters using the same expression-driven objective.

Table 5 shows that every architectural replacement increases L2RE on all three tasks, supporting the contribution of each component under the matched-parameter protocol. Replacing patch embedding with a pointwise tokenizer causes the largest degradation, raising errors to 2.30, 2.79, and 3.67 times those of Full SymOnet on CNS-OOD, DR, and SWE, respectively. Replacing IE self-attention with an MLP produces the next-largest increase on every task, highlighting the value of both patch embedding and attention-based input encoding. Linear coordinate projection incurs the smallest penalty, but still raises L2RE by 44.6%–55.7%, indicating that the query encoder also contributes consistently.

Every single adapter group improves over the frozen reference, and updating all three groups reduces its L2RE by 89.7%, 92.1%, and 84.9% on CNS-OOD, DR, and SWE, respectively. Under the reported configurations, IE adapters perform best among single-group variants, and excluding them causes the largest degradation among leave-one-out variants on every task. Removing any adapter group

increases L2RE relative to joint adaptation at the patch, input-encoding, and fusion stages. Full-model fine-tuning achieves the lowest errors, improving on all-adapter fine-tuning by a further 3.4%, 6.9%, and 9.6%, respectively; the remaining gap quantifies the accuracy tradeoff of keeping the backbone frozen.

The attention feed-forward blocks use GeGELU. Adapter modules use scale 1.0 and GELU activation, with the up-projection initialized to zero. The cross-attention adapter is disabled during ablation pre-training and enabled during downstream fine-tuning.

G Unified Training and Fine-Tuning Algorithm

Algorithm 1 Unified SymOnet training and expression-driven fine-tuning with autoregressive rollout

Input: PDE family $\mathcal{F}$, sampled functions $a(X)$ and sensor coordinates X, prescribed initial and boundary conditions, loss weights, trainable parameters Θ_{train}, lengths $(T_{\text{pre}}, T_{\text{out}})$, learning rate η
In expression-driven fine-tuning, $a(X)$ discretizes the prescribed initial-condition function
Solution data in the prediction interval: available for optimization only when $\lambda_{\text{data}} > 0$
Initialise or restore SymOnet parameters θ; freeze $\theta \setminus \Theta_{\text{train}}$
$N \leftarrow \lceil T_{\text{out}}/T_{\text{pre}} \rceil$ *// rollout steps*
repeat
  Select $a(X)$ with coordinates X according to the training regime and set $\hat{\mathbf{A}}^{(0)} = (a(X), X)$
  Compute $\mathcal{L}_{\text{ic}}^{(0)}$ against $a(X)$ on the prescribed channels if $\lambda_{\text{pde}}\lambda_{\text{ic}} > 0$; otherwise set it to zero
  $\mathcal{L}_{\text{roll}} \leftarrow \lambda_{\text{pde}}\lambda_{\text{ic}}\mathcal{L}_{\text{ic}}^{(0)}$ *// initial-condition constraint*
  for $n = 0, 1, \ldots, N-1$ **do**
    Construct $Y^{(n)}$ for the next $\min(T_{\text{pre}}, T_{\text{out}} - nT_{\text{pre}})$ prediction steps
    $u_\theta^{(n)} \leftarrow \mathcal{G}_\theta(\hat{\mathbf{A}}^{(n)}, Y^{(n)})$
    $\mathcal{L}_{\text{pde}}^{(n)} \leftarrow \lambda_{\text{eq}}\mathcal{L}_{\text{eq}}^{(n)} + \lambda_{\text{bc}}\mathcal{L}_{\text{bc}}^{(n)}$ if $\lambda_{\text{pde}} > 0$; otherwise set it to zero
    if $\lambda_{\text{data}} > 0$ **then**
      Read $u_{\text{ref}}^{(n)} \leftarrow u_{\text{ref}}(Y^{(n)})$ *// data-driven regime only*
      $\mathcal{L}_{\text{data}}^{(n)} \leftarrow \text{L2RE}\big(u_\theta^{(n)},\, u_{\text{ref}}^{(n)}\big)$
    else
      $\mathcal{L}_{\text{data}}^{(n)} \leftarrow 0$
    end if
    $\mathcal{L}^{(n)} \leftarrow \lambda_{\text{pde}}\,\mathcal{L}_{\text{pde}}^{(n)} + \lambda_{\text{data}}\,\mathcal{L}_{\text{data}}^{(n)}$
    $\mathcal{L}_{\text{roll}} \leftarrow \mathcal{L}_{\text{roll}} + \mathcal{L}^{(n)}$
    $\hat{\mathbf{A}}^{(n+1)} \leftarrow \mathsf{Roll}\big(\hat{\mathbf{A}}^{(n)},\ (u_\theta^{(n)}, Y^{(n)})\big)$
    Use the latest predicted state for the next expression-driven interval, or retain the configured window in data-driven training
  end for
  Update Θ_{train} using $\nabla_{\Theta_{\text{train}}}\mathcal{L}_{\text{roll}}$ and learning rate η
until convergence
Data-driven pre-training: $\Theta_{\text{train}} = \theta$ and $\lambda_{\text{pde}} = 0$
Expression-driven fine-tuning: $\Theta_{\text{train}} =$ Adapter parameters, $\lambda_{\text{data}} = 0$, and $\lambda_{\text{pde}} = 1$