3D-PINNS: A UNIFIED FRAMEWORK FOR DIMENSION-WISE INTERPRETABILITY AND ADAPTIVE DOMAIN DECOMPOSITION

Shuyuan Shang*
Illinois Institute of Technology

Ming Zhong
University of Houston

Ren Wang
Illinois Institute of Technology

ABSTRACT

Physics-informed neural networks (PINNs) provide a flexible framework for solving partial differential equations (PDEs), yet they often face difficulties in high-dimensional settings and in capturing solutions with localized or sharp features. In addition, most existing PINN-based approaches offer limited insight into how each dimensions contribute to the learned solution. We introduce Dimension-Domain Co-Decomposition (3D-PINNs), a structured and unified framework that combines dimension decomposition with mixture-of-experts-based domain decomposition. Within each expert, the solution is modeled through decoupled dimension components, while a router adaptively partitions the domains without requiring predefined subdomains or interface conditions. This formulation encourages structured representations reflecting dimension-wise characteristics of the underlying PDE solution. Empirical results on PDE benchmarks demonstrate that 3D improves solution accuracy and yields interpretable structure. The code and data used in this paper are available at `https://anonymous.4open.science/r/3D-PINNs_code`.

1 INTRODUCTION

Partial differential equations (PDEs) play a central role in modeling physical and engineering systems, yet their numerical solution becomes increasingly difficult as dimensionality grows or when solutions develop sharp features. While classical numerical solvers are effective in low-dimensional settings, their computational cost often grows prohibitively with dimension. Physics-informed neural networks (PINNs) provide a alternative by incorporating the governing equations into neural network training, alleviating limitations of mesh-based solvers (Raissi et al., 2019; Li et al., 2023; Meng et al., 2025; de Cominges Guerra et al., 2024). To further improve scalability, recent PINNs-based works have explored decomposition-based strategies. Dimension-wise decompositions reduce complexity by decoupling coordinate dependencies (Cho et al., 2023; Liu et al., 2024), but the resulting components are often difficult to interpret. Domain decomposition methods, on the other hand, partition the domains into multiple specialized subregions (Jagtap et al., 2020; Shukla et al., 2021; Hu et al., 2023), which is beneficial for capturing localized behaviors. However, existing domain decomposition methods require manually predefined subdomains and carefully designed interface conditions, making training problem-dependent.

To overcome these limitations, we propose Dimension-Domain Co-Decomposition (3D-PINNs), a unified framework that integrates dimension-wise representations with Mixture-of-Experts (MoE)-based domain decomposition. Within each expert, the solution is modeled through dimension-wise components via a shared Multilayer Perceptron (MLP), while an MoE router adaptively decomposes domain without predefined subdomains or explicit interface constraints. This design allows the model to better capture localized features while remaining parameter efficient. An overview of the proposed framework is illustrated in Figure 1.

We summarize our contributions as follows:

*Research intern at the Trustworthy and Intelligent Machine Learning Research Lab, ECE Department, Illinois Institute of Technology.

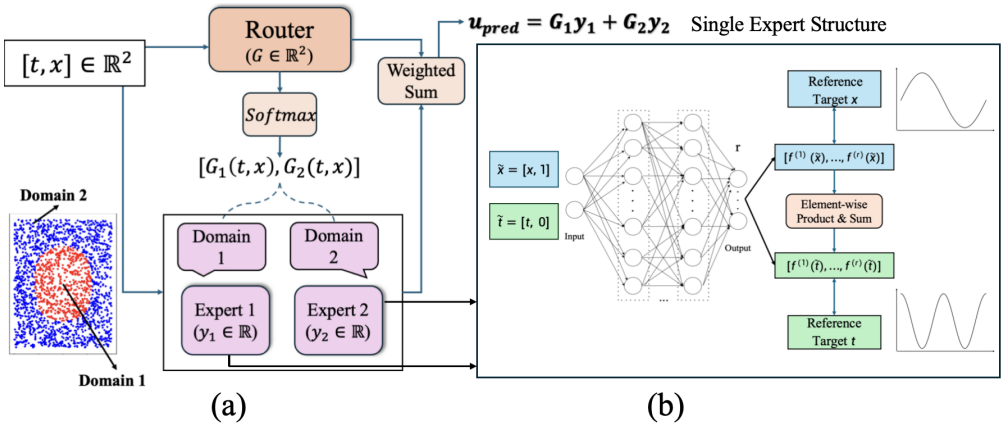

Figure 1: **Overview of the proposed 3D framework with two experts.** *(a) MoE Architecture:* An MoE router assigns soft weights to experts based on the input coordinates, inducing an automatic partition of the solution domain. *(b) Single Expert Structure:* Each expert adopts a dimension-decomposed architecture by a shared MLP that distinguishes input coordinates through indexing.

- We introduce *Dimension-Domain Co-Decomposition (3D-PINNs)*, a structured learning framework that combines dimension decomposition with adpative domain decomposition for PDE learning.
- We design dimension decomposition architecture using a shared MLP within each expert, allowing direct visualization of each coordinate contribution.
- We incorporate MoE routing to achieve automatic domain decomposition, enabling the model to capture localized solution features without manually specified subdomains or explicit interface conditions.

2 Dimension Domain Co-Decomposition

We consider a neural approximation of a target solution $u(\mathbf{x})$ defined over an input $\mathbf{x} \in \mathbb{R}^d$. The proposed Dimension Domain Co-Decomposition (3D-PINNs) framework models the solution by combining dimension decomposition within each expert and domain decomposition across multiple experts. Within each expert, the solution is represented in a dimension-wise form, where a shared MLP produces latent components for each dimension. These components are then combined to form the expert output. Multiple experts are aggregated through a MoE mechanism with a router assigning input-dependent weights to combine expert outputs, enabling adaptive domain decomposition. Furthermore, our framework provides inherent interpretability by providing direct visualizations between the learned dimension-wise components and the underlying solution components.

2.1 Dimension Decomposition

We describe the structure of a single expert in the proposed 3D-PINNs framework. Conventional methods mix all dimensions in a single network, coupling dimensional behaviors and causing great computational costs in high-dimensional settings. In contrast, we adopt dimension decomposition in single expert to decouple coordinates and simplify both forward propagation and derivative computation. To further decrease parameters, we employ a shared MLP across all dimensions instead of introducing separate networks per dimension within each expert. Concretely, for the input $\mathbf{x} = (x_1, \ldots, x_d)$, we construct d two-dimensional vectors $(x_1, 0) \ \ldots \ (x_d, d-1)$, and apply the same MLP $f : \mathbb{R}^2 \to \mathbb{R}^r$ to each vector. This design enables parameter sharing across dimensions and reduces the total number of training parameters compared to using independent networks. Formally, the output of a single expert $E(\mathbf{x})$ is represented as

$$E(\mathbf{x}) = \sum_{i=1}^{r} \prod_{j=1}^{d} f^{(i)}(x_j, j-1), \tag{1}$$

where $f^{(i)}(\cdot)$ denotes the i-th component of the MLP output, and r is a latent rank controlling the expressivity of the representation.

2.2 MoE-Driven Domain Decomposition

To effectively resolve localized solution structures, we advance domain decomposition by incorporating a MoE routing mechanism (Zhou et al., 2022; Zhang et al., 2025). Traditional decomposition strategies necessitate pre-defined subdomains and complex interface conditions to ensure global continuity. In constrast, our framework achieves automatic decomposition thorugh a routing network $G : \mathbb{R}^d \to \mathbb{R}^K$ that maps the input coordinates $\mathbf{x} \in \mathbb{R}^d$ directly to gating weights. These weights function as soft-partition indicators, dynamically directing each expert E_i to focus on modeling local behaviors. Each expert remains locally smooth while capturing distinct, complementary features of the PDE solution. Their weighted combination yields a global approximation

$$\hat{u}(\mathbf{x}) = \sum_{i=1}^{K} G_i(\mathbf{x})\, E_i(\mathbf{x}).$$

This MoE-based design allows automatic domain decomposition during training without requiring predefined subdomains or explicit interface constraints.

3 Experiments

3.1 Experiment Setup

We evaluate the proposed framework across two distinct scenarios to assess the efficacy of its components: (i) **Dimension decomposition** using a single-expert configuration on the Poisson and Wave equations and (ii) **MoE-driven domain decomposition** for the Viscous Burgers equation, where each expert employs dimension decomposition architecture. The models are optimized in an end-to-end pipeline using standard physics-informed losses. We quantify the approximation accuracy using the relative ℓ_2 error between the predicted and reference solutions.

3.2 Dimension Decomposition on Poisson and Wave

Compact Architecture via Shared MLP. Empirical results demonstrate that the proposed shared-MLP design improves parameter efficiency without degrading accuracy. Testing the 5d Poisson equation with a single expert, we compare a shared MLP architecture against an independent-MLPs variant with the same depth and width. In this setting, the shared MLP uses approximately $5\times$ fewer trainable parameters than independent MLPs. Figure 2 shows the training curves of the shared MLP, independent MLPs, and a vanilla PINN baseline. The shared and independent MLPs exhibit comparable convergence behavior and both substantially outperform vanilla PINNs. At convergence, the shared MLP and independent MLPs achieve relative ℓ_2 errors of 1.84×10^{-4} and 3.26×10^{-4}, respectively. In contrast, vanilla PINNs require substantially more training iterations and still converge to a much higher error of 7.55×10^{-3}. These results indicate that parameter sharing across dimensions preserves expressive power while significantly reducing model size, supporting the use of shared MLPs for high-dimensional PDEs.

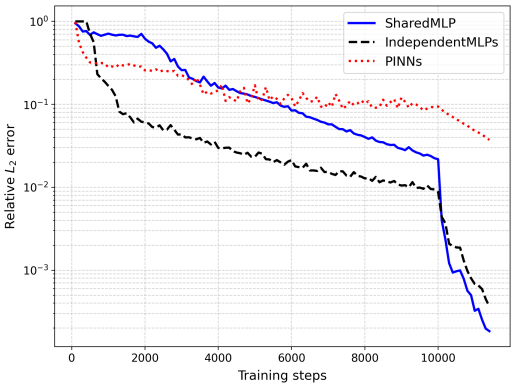

Figure 2: The shared MLP architecture achieves comparable accuracy to independent MLPs with substantially fewer parameters, and both clearly outperform vanilla PINNs.

Interpretability. To evaluate the interpretability of our framework, we visualize the evolution of learned dimension-wise components for the 1D Wave equation with the analytical solution $u(t, x) = \sin(\pi x)\cos(c\pi t)$. As illustrated in Figure 3, the model progressively recovers the decoupled analytical components, $f_t(t) = \cos(c\pi t)$ and $f_x(x) = \sin(\pi x)$ during training. Significantly, the spatial component f_x is resolved within the first 1000 iterations, whereas the higher-frequency temporal component f_t requires approximately 4000 steps to converge. This discrepancy is consistent with the spectral bias typically observed in PINNs, where high-frequency modes require more

optimization effort. By explicitly separating dimensions, our architecture allows us to monitor this convergence behavior for each component individually–a process that remains hidden in standard neural networks.

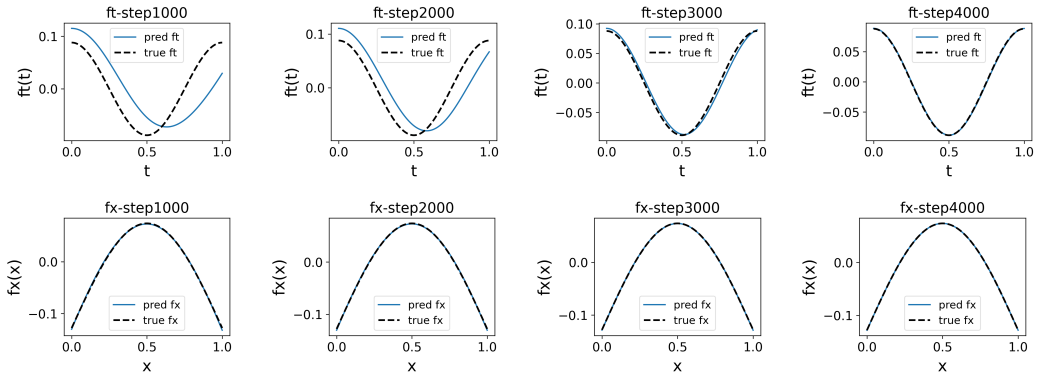

Figure 3: **Evolution of dimension-wise components for the 1D Wave equatio when $c = 2$.** Top and bottom rows show temporal f_t and spatial f_x components, respectively. As training progresses (1k to 4k steps), the model accurately recovers the analytical functions (black dotted lines), with the lower-frequency f_x converging faster than f_t.

3.3 Domain Decomposition on Viscous Burgers.

We evaluate the MoE-based domain decomposition on the Viscous Burgers equation with viscosity $\nu = \frac{0.01}{\pi}$, which features a sharp shock at $x = 0$. Multiple experts are employed with dimension decomposition retained within each expert, and a router assigns soft weights over the domain. As shown in Figure 4, the router network automatically identifies the shock as a natural interface, effectively partitioning the domain without prior knowledge. In this scenario, two experts are sufficient to resolve the solution: each specializes in the distinct regional dynamics on either side of the shock. This adaptive specialization leads to a substantial performance gain, with the relative ℓ_2 error dropping from 0.2108 ± 0.1252 (single expert) to 0.0011 ± 0.0005 ($K = 2$) under five random seeds, showing the precision of MoE-based domain decomposition.

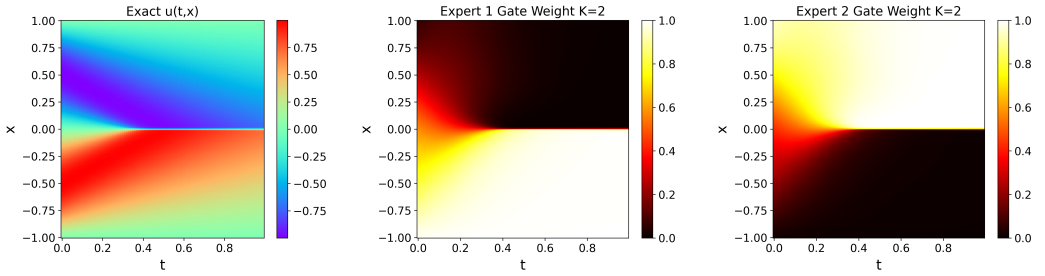

Figure 4: **Automatic domain decomposition for $K = 2$ on the Viscous Burgers equation.** The router assignments align with the dominant shock structure of the solution.

4 Conclusion

In this work, we introduce Dimension Domain Co-Decomposition (3D-PINNs) framework that synergizes dimension decomposition with MoE-driven domain decomposition. By leveraging a shared MLP across decoupled coordinates and an adaptive routing mechanism, 3D-PINNs achieves high-fidelity PDE approximations with significant parameter efficiency. Our results on PDE benchmarks demonstrate that this co-decomposition strategy not only captures localized features but also offers a scalable and interpretable alternative to neural PDE solvers. This approach establishes a principled foundation for balancing model expressivity with computational efficiency.

REFERENCES

Claude Basdevant, Michel Deville, Pierre Haldenwang, J.-M. Lacroix, Jalil Ouazzani, Roger Peyret, Paolo Orlandi, and Anthony T. Patera. Spectral and finite difference solutions of the burgers equation. *Computers & Fluids*, 14(1):23–41, 1986. doi: 10.1016/0045-7930(86)90036-8.

Junwoo Cho, Seungtae Nam, Hyunmo Yang, Seok-Bae Yun, Youngjoon Hong, and Eunbyung Park. Separable physics-informed neural networks. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS 2023)*, 2023.

Ignacio de Cominges Guerra, Wenting Li, and Ren Wang. A comprehensive analysis of pinns for power system transient stability. *Electronics*, 13(2):391, 2024.

Zheyuan Hu, Ameya D. Jagtap, George Em Karniadakis, and Kenji Kawaguchi. Augmented physics-informed neural networks (apinns): A gating network-based soft domain decomposition methodology. *Engineering Applications of Artificial Intelligence*, 113:107183, 2023.

Ameya D. Jagtap, Kenji Kawaguchi, and George Em Karniadakis. Extended physics-informed neural networks (xpinns): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. *Communications in Computational Physics*, 28(5):2002–2041, 2020.

Wenting Li, Deepjyoti Deka, Ren Wang, and Mario R Arrieta Paternina. Physics-constrained adversarial training for neural networks in stochastic power grids. *IEEE Transactions on Artificial Intelligence*, 5(3):1121–1131, 2023.

Youqiong Liu, Li Cai, and Yaping Chen. Variable separated physics-informed neural networks based on adaptive weighted loss functions for blood flow model. *Computers & Mathematics with Applications*, 153, 2024. doi: 10.1016/j.camwa.2023.11.018.

Chuizheng Meng, Sam Griesemer, Defu Cao, Sungyong Seo, and Yan Liu. When physics meets machine learning: A survey of physics-informed machine learning. *Machine Learning for Computational Science and Engineering*, 1(1):20, 2025.

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

Khemraj Shukla, Ameya D. Jagtap, and George Em Karniadakis. Parallel physics-informed neural networks via domain decomposition. *Journal of Computational Physics*, 447:110683, 2021.

Xu Zhang, Kaidi Xu, Ziqing Hu, and Ren Wang. Optimizing robustness and accuracy in mixture of experts: A dual-model approach. In *Forty-second International Conference on Machine Learning*, 2025.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114, 2022.

A DETAILS OF PDE EXAMPLES

In this appendix, we detail the PDE setups used in the main paper: Poisson, Wave and Viscous Burgers

A.1 POISSON EQUATION

We consider the Poisson problem with homogeneous Dirichlet boundary conditions:

$$\begin{cases} -\Delta u(\mathbf{x}) = f(\mathbf{x}) & \mathbf{x} \in \Omega, \\ u(\mathbf{x}) = 0 & \mathbf{x} \in \partial\Omega. \end{cases} \tag{2}$$

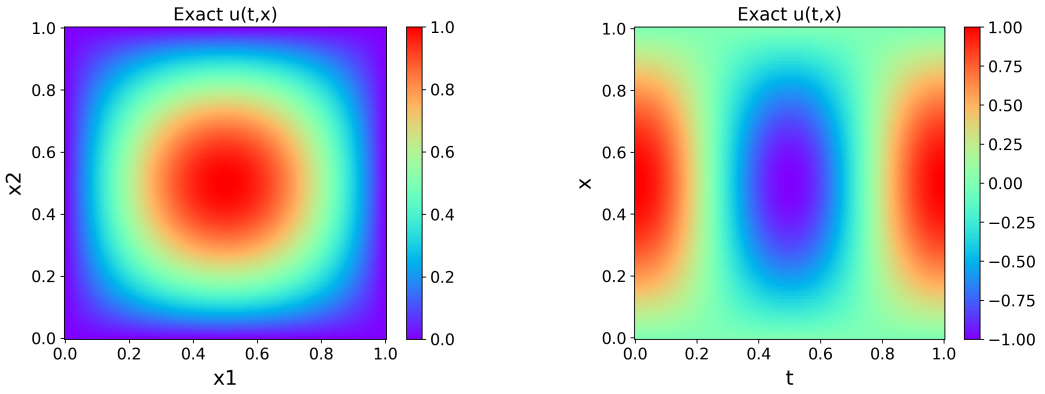

Figure 5: **Ground truths for 5d Poisson and Wave examples.** The left figure is the ground truth of 5d Poisson slice at $(x_3, x_4, x_5) = (0.5, 0.5, 0.5)$. The right one is the ground truth of 1d wave equation when $c = 2.0$.

where $\Omega = [0, 1]^5$ and $\mathbf{x} = (x_1, \ldots, x_5)$. We use the manufactured solution

$$u(\boldsymbol{x}) \;=\; \prod_{i=1}^{5} \sin(\pi x_i), \tag{3}$$

for which

$$-\Delta u \;=\; 5\pi^2 \prod_{i=1}^{5} \sin(\pi x_i) \;=\; f(\boldsymbol{x}). \tag{4}$$

Figure 5 shows a 2D slice of 5d Poisson u with respect to (x_1, x_2) while fixing $(x_3, x_4, x_5) = (0.5, 0.5, 0.5)$.

A.2 Wave Equation

Wave equation is a time-dependent PDE that takes the form:

$$\begin{cases} u_{tt}(t,x) = c^2 \Delta u & x \in (0,1), t \in [0,1] \\ u(0,t) = u(1,t) = 0 & t \in [0,1] \\ u(x,0) = \sin(\pi x), u_t(x,0) = 0 & x \in [0,1] \end{cases} \tag{5}$$

where c is the wave speed. In our experiments, we test $c = 2.0$. The analytical form of Wave equation is $u(t,x) = \sin(\pi x)\cos(\pi c t)$. Figure 5 shows the ground truth figure.

A.3 Viscous Burgers

The Burgers equation is a fundamental nonlinear PDE combining advection and diffusion, used as a prototype for shock formation and turbulence modeling. We consider the following Viscous Burgers:

$$\begin{cases} u_t + uu_x = \nu u_{xx} & x \in [-1,1], t > 0 \\ u(-1,t) = 0, u(1,t) = 0 & t \geq 0 \\ u(x,0) = -\sin(\pi x) & x \in [-1,1] \end{cases} \tag{6}$$

where viscosity $\nu = \frac{0.01}{\pi}$. With such small viscosity, the solution behaves almost inviscid: gradients steepen rapidly and form very thin viscous layers (shock transitions). Similarly, we set $T = 1$ and $t \in [0,1]$. Analytical solution is introduced in (Basdevant et al., 1986). Gound truth figure is shown in main text, see Figure 4.

B TRAINING DETAILS

Data and seed. For each PDE, we randomly sample according to Gaussian distribution N_f collocation points in the interior domain and N_b points on the boundary. For time-dependent PDEs, we additionally sample N_{ic} points from the initial condition. The specific collocation point counts are summarized in Table 1. For Poisson problems, we normalize data into $[-1, 1]$ before sending into the model.

To ensure robustness, all experiments are conducted using five random seeds: $\{1234, 42, 4, 2, 1\}$. Results reported in the main text use **seed 1234** as the default.

Table 1: Task-specific experimental configurations and hyperparameters.

PDE Case	**Points** (N_f, N_b, N_{ic})	**Adam LR**	**Weights** (w_f, w_{bc}, w_{ic})
Viscous Burgers	$(10000, 200, 256)$	2×10^{-3}	$(1, 1, 10)$
Poisson	$(8192, 2048, 0)$	5×10^{-4}	$(1, 5000, \text{N/A})$
Wave Equation	$(8192, 1024, 1024)$	10^{-3}	$(1, 100, 100)$

Loss Function We follow the standard PINN framework (Raissi et al., 2019) to train our model u_θ. For a generic time-dependent PDE $\mathcal{F}(u) = 0$ with boundary conditions $\mathcal{B}[u] = 0$ and initial condition $u(x, 0) = u_0$, the total loss is defined as:

$$\mathcal{L}(\theta) = w_f \mathcal{L}_{\text{PDE}} + w_{bc} \mathcal{L}_{\text{BC}} + w_{ic} \mathcal{L}_{\text{IC}} \tag{7}$$

The individual components are computed as mean squared residuals over collocation points:

$$\mathcal{L}_{\text{PDE}} = \frac{1}{N_f} \sum_{i=1}^{N_f} \left| \mathcal{F}(x_f^{(i)}, t_f^{(i)}, u_\theta, \dots) \right|^2 \tag{8}$$

$$\mathcal{L}_{\text{BC}} = \frac{1}{N_{bc}} \sum_{j=1}^{N_{bc}} \left| \mathcal{B}[u_\theta](x_{bc}^{(j)}, t_{bc}^{(j)}) \right|^2 \tag{9}$$

$$\mathcal{L}_{\text{IC}} = \frac{1}{N_{ic}} \sum_{k=1}^{N_{ic}} \left| u_\theta(x_{ic}^{(k)}, 0) - u_0(x_{ic}^{(k)}) \right|^2 \tag{10}$$

where all spatial and temporal derivatives are obtained via automatic differentiation. The scalar weights $\{w_f, w_{bc}, w_{ic}\}$ are task-specific and detailed in Table 1.

Two-stage Optimization We employ a two-stage optimization strategy: an Adam warm-up followed by L-BFGS refinement.

- **Adam Stage:** We use cosine learning rate annealing (down to 10^{-6}) and linearly anneal the PDE residual weight w_f from $w_f^{\text{init}} = 0.01$ to $w_f^{\text{final}} = 1.0$ over $T_{\text{anneal}} = 0.75 n_{\text{Adam}}$ steps:
$$w_f(e) = w_f^{\text{init}} + (w_f^{\text{final}} - w_f^{\text{init}}) \cdot \min(e / T_{\text{anneal}}, 1) \tag{11}$$
where e is the current step. We set $n_{\text{Adam}} = 10000$ and task-specific learning rates are listed in Table 1.
- **L-BFGS Stage:** We use `torch.optim.LBFGS` with `max_iter`=20,000, `history_size`=100, and strong-Wolfe line search. The optimizer terminates early if the gradient norm falls below 10^{-9} or the change in loss is less than 10^{-12}. During this stage, collocation points are fixed (20k interior, 5k boundary, 5k initial) for all cases except Viscous Burgers, which maintains the Adam-stage sampling.