PINN-ffusion: Physics-Informed Neural Network via Diffusion-based Adaptive Sampling and Anisotropic Domain Decomposition

Anonymous authors
Paper under double-blind review

Abstract

Physics-Informed Neural Networks (PINNs) provide a flexible framework for solving partial differential equations (PDEs) by incorporating physical constraints into neural network training. However, their performance is often limited by the non-uniform distribution of PDE residuals, which are typically concentrated in localized regions such as boundary layers and sharp gradients. Existing approaches address this issue through either *adaptive sampling* as a minimax optimization problem or *domain decomposition* based on fixed partitioning schemes, which lead to instability and limited flexibility. In this study, we propose PINN-ffusion, a unified framework that reformulates adaptive sampling and domain decomposition as a coupled problem of probability evolution and geometry learning. Instead of adversarial optimization, we construct the sampling distribution via a diffusion process, enabling stable adaptation while preserving the original PINN objective. In addition, we introduce a geometry-aware domain decomposition based on diversity and uncertainty metrics, generalizing isotropic partitions into anisotropic structures aligned with the residual landscape. The proposed framework allows sampling and domain geometry to co-evolve during training, improving the allocation of computational resources to complex regions. We provide a theoretical justification of consistency with the PINN objective and demonstrate through experiments that PINN-ffusion achieves improved accuracy and stability, particularly in problems with localized and anisotropic error structures.

1 Introduction

Partial Differential Equations (PDE) are fundamental tools for modeling physical systems across science and engineering. Although classical numerical methods such as Finite Element Method (FEM) (Ciarlet & Raviart, 1972; Xiong et al., 2025) and Finite Difference Method (FDM) (Courant et al., 1928; Lax & Richtmyer, 1956) provide effective approximations, they often require mesh generation and problem-specific design. Recently, Physics-Informed Neural Networks (PINNs) (Raissi et al., 2019) have emerged as a flexible, mesh-free alternative that directly incorporates governing equations into the learning process. PINNs consist of a single neural network that learns PDE solutions by minimizing residual and boundary losses over the domain. However, in practical PDE problems, residuals are highly non-uniform and often concentrated in localized regions such as boundary layers, shocks, and multi-scale structures. This heterogeneity makes it difficult for PINNs with uniform sampling and global models to accurately capture complex solution behaviors.

To address this limitation, various domain decomposition approaches have been proposed. eXtended Physics-Informed Neural Network (XPINN) (Jagtap & Karniadakis, 2020) partitions the domain into multiple subdomains with independent networks, while Augmented Physics-Informed Neural Network (APINN) (Hu et al., 2023) introduces a gating mechanism to enable soft partitioning. Although these methods improve local approximation, their reliance on fixed or pre-defined domain structures limits their ability to adapt to evolving residual patterns during training. PINN-ball (Bonfanti et al., 2025) addresses this issue by coupling

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.

adaptive sampling with dynamic domain decomposition, where the sampling distribution concentrates on regions with large residuals and induces ball-shaped subdomains. This allows the model to allocate computational resources to difficult regions more effectively.

Despite its effectiveness, PINN-ball has two fundamental limitations. First, its adaptive sampling is formulated as a minimax optimization problem, where the sampling distribution is learned adversarially. This leads to a degenerate solution where the distribution collapses to a Dirac delta, which results in introducing additional regularization that complicates both optimization and theoretical analysis. Second, the resulting domain decomposition is based on isotropic radial basis functions, which assume uniform importance in all directions. This limits the ability to capture anisotropic error structures such as thin layers and sharp interfaces, which are common in PDE solutions.

In this work, we propose PINN-ffusion, a framework that addresses these limitations of PINN-ball by reformulating adaptive sampling and domain decomposition as a coupled problem of probability evolution and geometry learning. Instead of solving a minimax problem for adaptive sampling, we construct the sampling distribution as the solution of a diffusion process (Song et al., 2021a), which provides a stable and analytically tractable way to evolve probability densities without collapse. At the same time, we generalize domain decomposition by introducing an anisotropic metric via Mahalanobis distance (McLachlan, 1999) to capture both the global structure of the sampling distribution and the local uncertainty of the residual field. This enables the model to align its representation with directional error patterns and efficiently focus on complex regions.

Our contributions are summarized as follows.

- We propose a diffusion-based adaptive sampling framework that replaces adversarial optimization with a probability evolution process, eliminating collapse while preserving the original PINN objective.
- We introduce a geometry-aware domain decomposition based on a combination of diversity and uncertainty metrics, enabling anisotropic partitioning aligned with residual structures.
- We show that the diffusion-based sampling is consistent with the original PINN objective, including on multi-dimensional and irregular domains, and that anisotropic partitions admit a smaller local approximation-error bound than isotropic ones. These analyses are supported by ablation, adaptive-sampling baselines, statistical tests, and cost comparisons.

2 Preliminaries

2.1 Physics-Informed Neural Networks and PINN-ball

Physics-Informed Neural Networks (PINNs) (Raissi et al., 2019) approximate solutions to partial differential equations (PDEs) by minimizing the residual of the governing equation together with boundary conditions using the loss:

$$\mathcal{L}(\theta) = \frac{1}{|N_\Omega|} \sum |\mathcal{R}u(x_r, \theta) - f(\hat{x}_r)|^2 + \frac{1}{N_{\partial\Omega}} \sum |u(x_b, \theta) - h(\hat{x}_b)|^2, \tag{1}$$

where u denotes the solution of the PDEs approximated by neural network parameterized by θ, Ω is the domain of u, $\mathcal{R}$ is the differential operator, f is the predicted residual points, h is the predicted boundary and initial points, and N_Ω and $N_{\partial\Omega}$ are the numbers of interior and boundary collocation points, respectively (see Table 4 in Appendix A for the full notation).

Table 1 provides an overview of several PINN-based methods, with procedures of the baselines provided in Appendix B.1 and detailed discussion in Appendix G.1. Of particular relevance to our comparison is PINN-ball (Bonfanti et al., 2025), which couples adaptive sampling with dynamic domain decomposition. It formulates the training objective as a minimax problem over model parameter θ and probability p:

$$\min_\theta \max_{p \in \mathcal{P}(\Omega)} \int_\Omega |r_\theta(x)|^2 p(x)\, dx - \beta \mathcal{M}(p), \tag{2}$$

Table 1: Qualitative comparison of domain-decomposition strategies among XPINN (Jagtap & Karniadakis, 2020), FBPINN (Moseley et al., 2023), APINN (Hu et al., 2023), AS-PINNs (Peng et al., 2025), and PINN-ball (Bonfanti et al., 2025).

	XPINN	FBPINN	APINN	AS-PINNs	PINN-ball	PINN-ffusion (Ours)
Dynamic domain	✗	✗	non-training	✓	✓	✓
Adaptive sampling	✗	✗	✗	non-adaptive	✓	✓
Geometry	arbitrary	rectangle	implicit gating	polygonal interface	isotropic ball	anisotropic ellipse

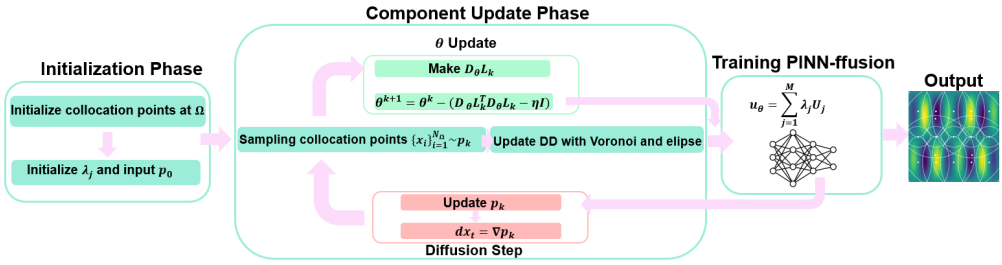


Figure 1: Overall Framework of PINN-ffusion

where $r_\theta(x) = |\mathcal{R}u(x_r, \theta) - f(\hat{x}_r)|$ denotes the PDE residual and $\mathcal{M}(p)$ is a regularization term to prevent the sampling distribution from collapsing into a Dirac delta function. The resulting sampling distribution concentrates on regions with large residuals and is approximated using Radial Basis Functions (RBFs), yielding a set of isotropic ball-shaped subdomains:

$$\lambda_j(x) = \frac{\phi_j(x)}{\sum_l \phi_l(x)}, \quad \phi_j(x) = \begin{cases} \left(1 - \frac{\|x - c_j\|^2}{s_j}\right)^2, & \|x - c_j\| \le s_j \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where the j-th subdomain is defined by $B_j := \{x \in \Omega \mid \phi_j(x) > 0\}$, with the radius of the subdomain denoted as s_j and λ_j denotes the subdomain weight. Since the decomposition may change in each epoch, the Levenberg-Marquardt (LM) algorithm (Marquardt, 1963) is employed. However, the minimax formulation requires additional regularization to avoid degenerate solutions, and the isotropic RBF-based decomposition cannot capture anisotropic structures such as thin layers and sharp interfaces.

2.2 Denoising Diffusion Implicit Models

Diffusion models (Ho et al., 2020) learn data distributions by simulating a stochastic forward process and its reverse-time dynamics (Grill et al., 2018; Travis & Ray, 2023). Denoising Diffusion Implicit Models (DDIM) (Song et al., 2021a) introduce a non-Markovian forward process that enables sampling with fewer reverse steps while preserving the same training objective. In the drift-free case considered in our method, the density evolution reduces to a heat equation (Section 3.1), which provides smooth density dynamics without the adversarial optimization of PINN-ball.

3 Methodology

In this section, we present PINN-ffusion, a framework that reformulates adaptive sampling and domain decomposition in PINN-ball as a coupled problem of probability evolution and geometry learning. Unlike prior approaches that rely on adversarial optimization and isotropic partitioning, our method replaces these components with diffusion-driven sampling dynamics (Section 3.1) and uncertainty-diversity-aware anisotropic domain decomposition (Section 3.2), respectively. Figure 1 illustrates the overall framework of PINN-ffusion.

3.1 Diffusion-based Adaptive Sampling

In PINN-ball (Bonfanti et al., 2025), adaptive sampling is formulated as an adversarial minimax problem, which collapses to a degenerate distribution unless an additional regularization term is introduced. We instead construct the sampling distribution as the solution of a diffusion process. Using the probability-flow ODE of the resulting heat equation, we obtain an explicit density evolution and a deterministic sampler, while preserving the original PINN residual objective.

Diffusion Formulation We consider the forward SDE with drift f, diffusion coefficient g, and Brownian motion w_t, whose marginal density evolves according to the Fokker–Planck equation (Song et al., 2021b; Haussmann & Pardoux, 1986):

$$dx_t = f(x_t, t)dt + g(t)dw_t, \tag{4}$$

$$\partial_t p(x_t, t) = -\nabla \cdot (f(x_t, t)p(x_t, t)) + \frac{1}{2}g(t)^2 \Delta p(x_t, t). \tag{5}$$

Our goal is twofold: 1) to guide samples toward regions with large PDE residuals, and 2) to keep the distribution smooth during training to prevent collapse or oscillation. If $f \neq 0$, the transport term can dominate and decouple the motion from the residual field, so we set $f = 0$ and the evolution is governed purely by diffusion:

$$\partial_t p(x_t, t) = \frac{1}{2}g(t)^2 \Delta p(x_t, t) = \kappa \Delta p(x_t, t), \tag{6}$$

which corresponds to the heat equation, with $\kappa = 1$. On $\Omega = [0, L]$ with Neumann conditions $\partial_x p(0, t) = \partial_x p(L, t) = 0$ and $p(x, 0) = p_0(x)$, separation of variables (Evans, 2022) (Appendix C.1.1) gives

$$p(x, t) = \int_0^L K_k(x, y, t) p_0(y) dy, \tag{7}$$

where $K_k(x, y, t) = \frac{1}{L} + \frac{2}{L}\sum_{n=1}^{\infty} e^{-\frac{n^2\pi^2}{L^2}t} \cos\frac{n\pi}{L}x \cos\frac{n\pi}{L}y$. Since the residual-based sampling study of (Wu et al., 2023) defines the sampling density as $p(x) \propto \epsilon^k(x)/\mathbb{E}(\epsilon^k) + c$ where c is a constant, we define $p_0^k(x) = \frac{r_{\theta_k}^2}{\int_\Omega r_{\theta_k}^2 dx}$ by choosing a residual exponent of 2 and setting $c = 0$ (if $\int_\Omega r_{\theta_k}^2 dx = 0$, we set p_0^k to the uniform density). Since each mode of Eq. 7 decays as $e^{-\frac{n^2\pi^2}{L^2}t}$, high frequencies vanish rapidly and distribution updates become smooth, mitigating the unstable sampling observed in prior work.

Sampling Dynamics To generate collocation points from the evolving density, we use the probability-flow ODE associated with the drift-free diffusion,

$$dx_s = -\nabla \log p_{s,k}(x_s)\, ds, \qquad s \in [0, t], \tag{8}$$

whose marginals satisfy the heat equation in Eq. 6 (Appendix C.1.1). Integrating Eq. 8 from $s = 0$ to $s = t$ therefore transports samples $x_0 \sim p_0^k$ to $x_t \sim p_{t,k}$, with the score available in closed form from Eq. 7. The two ingredients play complementary roles: the concentration on high-residual regions is inherited from the residual-informed initial density p_0^k, while the flow spreads the samples by exactly the amount prescribed by the heat kernel. This keeps $p_{t,k}$ bounded away from zero (Theorem 3.1) and prevents collapse without adversarial updates.

Consistency with PINN Objective An adaptive sampling strategy must remain consistent with the original PINN objective of minimizing the residual over the entire domain. Theorem 3.1 shows that driving the sampling-weighted loss to zero also drives the unweighted residual to zero on the whole domain, i.e., the reweighting creates no blind regions.

Theorem 3.1 (Convergence of the sampling-weighted residual)**.** *Fix a diffusion time $t > 0$ and let $r_{\theta_k}(x) = \mathcal{R}[u_{\theta_k}](x) - f(x)$ be the PDE residual at the k-th training iteration. Define the sampling-weighted residual loss $J(\theta_k, p_{t,k}) = \int_\Omega r_{\theta_k}^2(x)\, p_{t,k}(x)\, dx$, where $p_{t,k}(x) = \int_\Omega K_k(x, y, t)\, p_0^k(y)\, dy$ is the diffusion-based sampling density. Then there exists $m(t) > 0$, independent of k and θ_k, such that*

$$\|r_{\theta_k}\|_{L^2(\Omega)}^2 \leq \frac{J(\theta_k, p_{t,k})}{m(t)}. \tag{9}$$

In particular, $\lim_{k\to\infty} J(\theta_k, p_{t,k}) = 0$ *implies* $\lim_{k\to\infty} \|r_{\theta_k}\|_{L^2(\Omega)} = 0$. *If, in addition,* $\{r_{\theta_k}\}_k$ *is uniformly Lipschitz on* Ω*, the convergence is uniform on* $\bar{\Omega}$.

Proof. See Appendix C.1.2. □

Multi-dimensional and Irregular Domains Since multidimensional and irregular domains frequently arise in practical PDE problems, we extend our formulation to these settings. On a box $\Omega_{\text{box}} = \prod_{i=1}^{d}[0, L_i]$, the Neumann heat kernel separates across coordinates. For a non-rectangular domain $\Omega_{\text{eff}} \subseteq \Omega_{\text{box}}$, the resulting density is restricted to Ω_{eff} and renormalized as:

$$K_k(\mathbf{x}, \mathbf{y}, t) = \prod_{i=1}^{d} K_{k,i}(x_i, y_i, t), \qquad \tilde{p}_{t,k}(\mathbf{x}) = \frac{p_{t,k}(\mathbf{x})\, \mathbf{1}_{\Omega_{\text{eff}}}(\mathbf{x})}{\int_{\Omega_{\text{eff}}} p_{t,k}(\mathbf{x}')\, d\mathbf{x}'}, \tag{10}$$

where each $K_{k,i}$ is the one-dimensional kernel of Eq. 7. Hence, p and its score remain in closed form at $O(dN)$ cost for N collocation points. In Navier–Stokes for example, Ω_{eff} excludes the cylinder. Notably, Theorem 3.1 above continues to hold on Ω_{eff}, with Lemma C.3 and Corollary C.4 provided in Appendix C.1.3.

Remark 3.2 (Novelty of Diffusion-based Adaptive Sampling)**.** Unlike adversarial sampling, $p_{t,k}$ is not the maximizer of a minimax objective but is constructed explicitly by heat-flow smoothing of a residual-informed initial density, so no sampling parameters are optimized and no coupled adversarial updates arise. Moreover, since the heat kernel is strictly positive for $t > 0$, $p_{t,k} \geq m(t) > 0$ on Ω at every iteration regardless of how concentrated p_0^k is, so every region of the domain retains a nonzero sampling probability and the sampling cannot collapse to a Dirac measure. Note that p_0^k is invariant to rescaling of the residual, so the concentration on high-residual regions is maintained even as the residual decreases. The diffusion time t controls this trade-off: a small t preserves the concentration of p_0^k, whereas as $t \to \infty$, $p_{t,k}$ converges to the uniform distribution, the invariant density of the Neumann heat flow (Fulks, 1952).

3.2 Geometry-aware Anisotropic Domain Decomposition

While adaptive sampling determines where to allocate training samples, domain decomposition determines how the model represents the solution across the domain. In PINN-ball, subdomains are isotropic balls, whose partition learning can be sensitive to initialization and optimization imbalance, and which assume that all directions are equally important, an assumption often violated by PDE error structures with strong directional characteristics. We therefore introduce a geometry-aware anisotropic domain decomposition.

Anisotropic Decomposition via Mahalanobis Distance We generalize the domain decomposition framework by introducing an anisotropic distance measure:

$$d_{k,j}^2(z) = (z - c_{k,j})^T G_{k,j} (z - c_{k,j}), \tag{11}$$

where $c_{k,j}$ denotes the centroid of the j-th subdomain at iteration k, and $G_{k,j} \in \mathbb{S}_{++}^d$ is a symmetric positive definite matrix that defines the local geometry of each subdomain; setting $G_{k,j} = I$ recovers the isotropic balls of PINN-ball. This Mahalanobis distance yields ellipsoidal regions whose shape and orientation are determined by G, enabling the model to capture elongated or thin structures that are common in PDE solutions. **Uncertainty-Diversity-aware Geometry** We construct $G_{k,j}$ to reflect both global diversity and local uncertainty:

$$G_{k,j} = (1 - \alpha_k) G_{k,j}^{div} + \alpha_k G_{k,j}^{unc}, \tag{12}$$

where $G_{k,j}^{div}$ and $G_{k,j}^{unc}$ are built from the sample covariance and the squared-residual gradient in each subdomain, respectively:

$$G_{k,j}^{div} = \frac{d\,(\Sigma_{k,j} + \epsilon I)^{-1}}{\operatorname{tr}\big((\Sigma_{k,j} + \epsilon I)^{-1}\big)}, \; G_{k,j}^{unc} = I + \xi \frac{S_{k,j}}{\operatorname{tr}(S_{k,j})}, \tag{13}$$

$$S_{k,j} = \frac{\sum_n \lambda_{k,j}(x_n)\, \nabla_x r_{\theta_k}^2(x_n)\, \nabla_x r_{\theta_k}^2(x_n)^\top}{\sum_n \lambda_{k,j}(x_n)} \tag{14}$$

Here $\Sigma_{k,j}$ is the $\lambda_{k,j}$-weighted covariance of the collocation points $\{x_n\}$ in the j-th subdomain, $\epsilon, \xi > 0$ are stability and scaling constants, and $G^{unc}_{k,j} = I$ if $S_{k,j} = 0$. The outer product avoids the cancellation of $\nabla_x r^2_{\theta_k}$ across a thin layer. By trace normalization, the eigenvalues are independent of the residual magnitude, and $G^{unc}_{k,j}$ has eigenvalue up to $1 + \xi$ along the dominant direction of residual variation and 1 elsewhere; each ellipse thus contracts across the high-gradient direction and elongates along the level sets of $r^2_{\theta_k}$, with its size set by s_j. The warm-up coefficient $\alpha_k = \min(1, \frac{k}{K_w})$ balances the two geometries, so the residual geometry increasingly determines the orientation without fading as the residual decreases.

Anisotropic Decomposition Finally, we redefine the domain partitioning weights as:

$$\lambda_j(x) = \frac{\phi_{k,j}(x)}{\sum_l \phi_{k,l}(x)}, \qquad \phi_{k,j}(x) = \begin{cases} \left(1 - \dfrac{d^2_{k,j}(z)}{s_j}\right)^2, & d^2_{k,j}(z) \le s_j, \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

The matrix G thus distorts the balls into ellipses, so the model focuses more heavily on regions that are difficult to learn. For instance, the Helmholtz solution consists of vertically elongated oscillatory lobes; the isotropic balls of PINN-ball cover the domain with overlapping circles regardless of the lobe shape, whereas the ellipsoidal subdomains of PINN-ffusion stretch along the lobes and follow the solution structure (Figure 7 in Appendix D). Although Euclidean k-means can be interpreted as a special case of Lloyd iteration (Lloyd, 1982; Du et al., 1999), our method departs from plain k-means in two aspects: it employs p_k weighted anisotropic Voronoi (Voronoi, 1908; Aurenhammer, 1991) partitioning and jointly updates not only the centroids but also the local geometry through $G_{k,j}$. Figure 3 in Section 4.4 further shows that the resulting subdomains align with elongated error structures during training. Theorem 3.3 shows that anisotropic geometry has a smaller upper bound on the learning error than isotropic geometry.

Theorem 3.3. (*Approximation Error Advantage of Anisotropic Decomposition*) *Suppose domain Ω is divided into M subdomains $\{\lambda_j\}_{j=1}^M$, and suppose that subnetwork U_j provides a local approximation of the solution in each subdomain. For the principal direction $i = 1, ..., d$ of the subdomain, let h be the magnitude of λ_j in the direction of axis i, $\mathbf{g}_{j,i} = ||\partial_i u||_{L^2(\lambda_j)}$ be the change in the solution along the axis, and $V_j = \prod_{i=1}^d h_{j,i}$ be the volume of λ_j. When the volume V_j is fixed and the upper bound error function is defined as $\Phi(h_{j,1}, ..., h_{j,d}) = \sum_{i=1}^d h^2_{j,i}\mathbf{g}^2_{j,i}$, $E^2 \le C \sum_{i=1}^d h^2_{j,i}\mathbf{g}^2_{j,i}$ (Formaggia & Perotto, 2001; Huang, 2005) with a constant C, E^2 is minimized for the anisotropic choice $h_{j,i} \propto \mathbf{g}^{-1}_{j,i}$, and the isotropic choice $h_{j,i} = V_j^{\frac{1}{d}}$ is always inferior, except when $\mathbf{g}_{j,1} = ... = \mathbf{g}_{j,d}$. Thus, it satisfies*

$$\Phi_{aniso} \le \Phi_{iso}. \tag{16}$$

Proof. See, Appendix C.2 □

Remark 3.4 (Novelty of Anisotropic Geometry Learning)**.** Unlike isotropic ball-shaped partitions, our decomposition learns the orientation, shape, and concentration of each subdomain from both the sampling density and the residual geometry, and re-adjusts them as the error pattern evolves during training.

3.3 Overall PINN-ffusion Framework

Combining the proposed diffusion-based adaptive sampling and anisotropic domain decomposition, we define the PINN-ffusion model as a pair (u_θ, p_k):

$$u_\theta(x) = \sum_{j=1}^{M} \lambda_{k,j}(x)\, U_j(x; \theta_j), \tag{17}$$

where p_k denotes the sampling distribution obtained from the k-th diffusion update, $\lambda_{k,j}$ denotes the weights determined by the anisotropic decomposition, and U_j denotes j-th subnetworks specialized for different regions of the domain and each outer iteration k. We provide the overall training procedure of PINN-ffusion in Algorithm 5 of Appendix B.2.

Table 2: Quantitative Evaluation of PINN-ffusion and Baselines across PDE Benchmarks. The best-performing results are highlighted in bold.

Category	Equation	Component	Method				
			PINN	XPINN	APINN	PINN-ball	PINN-ffusion
Linear	Helmholtz	-	$1.01e{-00}_{\pm 1.92e-02}$	$1.82e{-01}_{\pm 1.70e-02}$	$1.35e{-03}_{\pm 1.74e-03}$	$1.51e{-09}_{\pm 1.72e-09}$	$\mathbf{1.28e{-09}}_{\pm 1.86e-09}$
Non-Linear	Burgers	-	$2.65e{-02}_{\pm 6.28e-04}$	$3.02e{-02}_{\pm 1.74e-03}$	$1.36e{-03}_{\pm 9.15e-04}$	$5.98e{-04}_{\pm 3.34e-04}$	$\mathbf{4.62e{-04}}_{\pm 2.97e-04}$
	Klein–Gordon	-	$4.18e{-03}_{\pm 8.65e-04}$	$5.98e{-01}_{\pm 7.57e-02}$	$5.07e{-03}_{\pm 1.57e-03}$	$8.42e{-05}_{\pm 4.07e-05}$	$\mathbf{1.11e{-05}}_{\pm 2.07e-05}$
Non-Linear System	Boussinesq –Burger	u	$1.60e{-02}_{\pm 7.15e-03}$	$3.05e{-02}_{\pm 5.97e-03}$	$2.31e{-02}_{\pm 4.86e-03}$	$9.09e{-03}_{\pm 3.52e-03}$	$\mathbf{6.71e{-03}}_{\pm 2.78e-03}$
		v	$1.32e{-01}_{\pm 5.06e-02}$	$1.14e{-01}_{\pm 4.88e-03}$	$9.76e{-02}_{\pm 3.14e-02}$	$1.15e{-01}_{\pm 7.32e-02}$	$\mathbf{8.99e{-02}}_{\pm 3.07e-02}$
	Navier –Stokes	u	$6.50e{-01}_{\pm 1.31e-01}$	$2.36e{-01}_{\pm 4.90e-02}$	$1.42e{-01}_{\pm 3.41e-03}$	$1.90e{-02}_{\pm 8.95e-03}$	$\mathbf{5.89e{-03}}_{\pm 4.11e-03}$
		v	$1.04e{-01}_{\pm 7.35e-02}$	$9.33e{-01}_{\pm 2.66e-02}$	$9.35e{-02}_{\pm 4.14e-03}$	$1.73e{-02}_{\pm 9.25e-03}$	$\mathbf{5.82e{-03}}_{\pm 4.24e-03}$
		p	$1.25e{+00}_{\pm 3.90e-01}$	$1.33e{+00}_{\pm 8.60e-02}$	$8.24e{-01}_{\pm 5.07e-02}$	$8.86e{-02}_{\pm 8.09e-03}$	$\mathbf{1.18e{-02}}_{\pm 8.74e-03}$

4 Experimental Results

4.1 Experimental Settings

We evaluate the proposed PINN-ffusion on five representative PDE benchmarks, including 1) Helmholtz (Kristensson, 1980; Moiola & Spence, 2014) as a linear equation, 2) Burgers (Hopf, 1950; Cole, 1951) and Klein–Gordon (Ginibre & Velo, 1989) as nonlinear equations, and 3) Boussinesq–Burger (two variables) (Chen & Li, 2006; Mhlanga & Khalique, 2012) and Navier–Stokes (three variables) (Leray, 1934; Kato, 1984) as nonlinear systems. The ground truth solutions are derived from the repository (Raissi, 2018). The details of each PDE are provided in Appendix E. We compare our method against 1) standard PINN (Raissi et al., 2019), 2) domain decomposition approaches such as XPINN (Jagtap & Karniadakis, 2020) and APINN (Hu et al., 2023), and 3) the dynamic decomposition approach of PINN-ball (Bonfanti et al., 2025). Baselines follow their original training settings, while PINN-ball and PINN-ffusion share the same second-order budget of 2,000 outer Truncated Newton (TN) iterations (Dembo & Steihaug, 1983), each with 64 inner iterations approximating the LM update. Performance is evaluated using the Relative L2 Error (Rell2) (Raissi et al., 2019), averaged over ten independent runs. Details are provided in Appendices F and G.

4.2 Main Result

As shown in Table 2, PINN-ffusion achieves the lowest error on every benchmark. Among the baselines, PINN-ball improves substantially over PINN, XPINN, and APINN by focusing on high-residual regions, e.g., reaching 1.51e-9 on Helmholtz, where the other baselines remain at or above 1.35e-3. However, its isotropic decomposition limits its ability to capture oscillatory structures, whereas the anisotropic decomposition of PINN-ffusion aligns with directional error patterns and further reduces the error of PINN-ball on every component. The improvement is particularly pronounced for the nonlinear equations and systems: PINN-ffusion reduces the error of PINN-ball by 86.8% on Klein–Gordon (8.42e-5 to 1.11e-5) and by 69.0%, 66.4%, and 86.7% on the u, v, and p components of Navier–Stokes, respectively. For further analysis, Appendices G.2 and G.3 assess statistical significance and robustness across initializations, respectively; and Appendix E presents predicted solutions and error maps.

4.3 Ablation Study

Table 3 isolates the contribution of each component on Navier–Stokes by toggling diffusion sampling and anisotropic decomposition within the same framework, with PINN-ball (adversarial sampling, isotropic decomposition) as the both-off baseline. All variants share the same optimizer and training budget.

Individual Effect Each component is effective on its own: anisotropic decomposition alone (w/o Diffusion) reduces the error of PINN-ball by 58.2%, 58.9%, and 45.0% on u, v, and p, and diffusion sampling alone (w/o Anisotropic) by 55.7%, 55.8%, and 52.9%.

Field-dependent Role Anisotropy alone performs better on the velocity components, whereas diffusion alone performs better on the pressure (4.17e-2 vs. 4.87e-2). Plausibly,

Table 3: Ablation study on Navier–Stokes (Rell2).

Model	Sampling	Decomposition	u	v	p
PINN-ball	Adversarial	Isotropic	1.90e-2	1.73e-2	8.86e-2
w/o Diffusion	Adversarial	Anisotropic	7.94e-3	7.11e-3	4.87e-2
w/o Anisotropic	Diffusion	Isotropic	8.42e-3	7.65e-3	4.17e-2
PINN-ffusion (Ours)	Diffusion	Anisotropic	**5.89e-3**	**5.82e-3**	**1.18e-2**

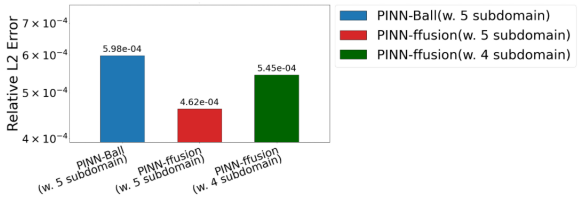
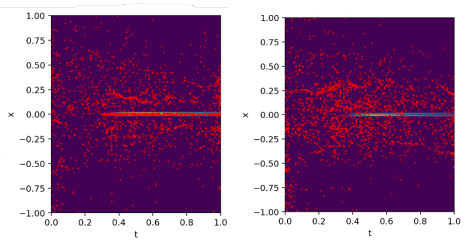


Figure 2: Decomposition efficiency of PINN-ball and PINN-ffusion on Burgers



(a) PINN-ffusion (b) PINN-ball

Figure 3: Collocation points (red) overlaid on the learned subdomains for Burgers in the (t, x) plane

ellipsoidal subdomains align with the localized, directional velocity structures in the wake, while the globally determined pressure benefits more from a stable sampling distribution.

Complementarity Combining both components further reduces the error on every field, by 18.1–30.0% on the velocity components and 71.7–75.8% on the pressure relative to the single-component variants. On p, the combined error (1.18e-2) is even lower than the value predicted by multiplying the two individual reduction ratios (2.29e-2), indicating a synergistic gain. Overall, PINN-ffusion reduces the error of PINN-ball by 69.0%, 66.4%, and 86.7% on u, v, and p. Comparisons with adaptive-sampling methods (e.g., QR-DEIM, QR-DEIM-R, and RAD) and the sensitivity analysis are provided in Appendix G.5 and G.6.

4.4 Partition Efficiency of Anisotropic Decomposition

Figure 2 compares PINN-ball and PINN-ffusion on Burgers from two perspectives: the accuracy under the same number of subdomains and the number of subdomains required for comparable accuracy. With the same five subdomains, PINN-ffusion reduces the error by 22.7% (5.98e-4 to 4.62e-4). Even with only four subdomains, PINN-ffusion (5.45e-4) still outperforms PINN-ball with five, indicating that anisotropic partitioning attains higher accuracy with fewer subdomains. The same trend holds on Helmholtz and Navier–Stokes. Detail of Appendix D. Figure 3 further shows where collocation points concentrate in the (t, x) plane, where the Burgers solution develops a sharp transition along $x = 0$ that steepens over time. Both methods move points toward this high-residual region, so the sampling mechanism is effective in either case; the difference lies in the geometry of the subdomains within which the points are allocated. The anisotropic subdomains of PINN-ffusion elongate along the transition, forming a thin band that follows $x = 0$ from roughly $t = 0.3$ onward, while the remaining points are spread over the smooth region. In contrast, the isotropic balls of PINN-ball cannot elongate in a preferred direction, yielding a wider, less directed region with more diffuse points in x. For more details, see Appendix G.7.

5 Optimization Discussion

5.1 Theoretical Analysis of LM Approximation

Since dynamic domain decomposition changes the residual landscape during training, PINN-ball adopts the second-order Levenberg–Marquardt (LM) algorithm (Marquardt, 1963) for stable updates, but solving its Jacobian linear system at every iteration is costly in computation and memory for large-scale problems. We therefore approximate the LM update

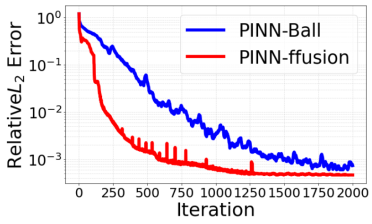


Figure 4: LM optimization behavior on Burgers

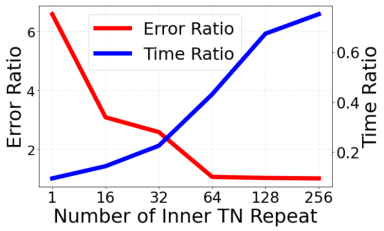


Figure 5: Error and time ratio of TN to LM on Burgers

with the Truncated Newton (TN) method (Dembo & Steihaug, 1983), which solves this system with a finite number of inner iterations. At outer iteration k, let $\theta_{k,i}$ be the parameter at the i-th inner iteration, ρ_k the frozen residual map (the residual r_{θ_k} in our setting), D the Jacobian operator, and $A_i := D\rho_k(\theta_{k,i})^\top D\rho_k(\theta_{k,i}) + \eta_k I$ with $\eta_k > 0$. Proposition 5.1 shows that the number of inner iterations controls the accuracy-cost trade-off with respect to the exact LM direction

$$d_k^{\mathrm{LM}} := -A_i^{-1} D\rho_k(\theta_{k,i})^\top \rho_k(\theta_{k,i}). \tag{18}$$

Proposition 5.1. (*Convergence of Truncated Newton to LM*) *Let k be fixed and assume A_i is symmetric positive definite. Then, as the number of inner iterations in the Truncated Newton method increases, the approximate solution converges to the exact LM direction d_k^{LM}.*

Proof. See Appendix C.3. □

5.2 Empirical Analysis of Optimization Efficiency

On Burgers under the exact LM optimizer (Figure 4), PINN-ffusion converges faster than PINN-ball with a consistently lower loss, but LM requires approximately 7 days for 2,000 iterations on an NVIDIA RTX A6000 Ada. Figure 5 shows the TN-to-LM ratios of error (red) and computation time (blue) over the number of inner iterations: with 64 inner iterations, TN reaches nearly the same loss as LM in less than 50% of its time. Under matched parameter budgets (∼5k, Table 14 in Appendix G.8), the cost of the second-order methods is dominated by the optimizer, and against PINN-ball with the same optimizer and budget, PINN-ffusion reduces the time per epoch from 45.2s to 36.5s (19%) and the peak memory from 38GB to 36GB, since its sampling density has no learnable parameters.

6 Limitations and Future Works

The proposed framework relies primarily on local geometric information when constructing subdomains, so its effectiveness may be limited when the error distribution is highly global, scattered, or multi-scale. In addition, diffusion-based sampling does not guarantee optimal sampling dynamics or globally optimal distributions, and iterative diffusion updates may still introduce computational overhead. Future work may incorporate global or hierarchical geometric information into the decomposition and exploit accelerated diffusion sampling or optimal probability-flow formulations to improve scalability to more complex PDE systems.

7 Conclusion

We propose PINN-ffusion, which reformulates adaptive sampling and domain decomposition in PINNs as a coupled problem of probability evolution and geometry learning. Diffusion-based sampling yields stable, collapse-free sampling consistent with the original PINN objective, and anisotropic decomposition combining diversity and uncertainty geometry captures directional and localized error structures with a smaller local approximation-error bound. Experiments show improved accuracy and stability across PDE benchmarks, particularly with shocks, boundary layers, and anisotropic residual patterns, suggesting that sampling and decomposition should be treated as jointly evolving processes driven by the residual structure.

References

Franz Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM computing surveys (CSUR)*, 23(3):345–405, 1991.

Andrea Bonfanti, Ismael Medina, Roman List, Björn Staeves, Roberto Santana, and Marco Ellero. PINN balls: Scaling second-order methods for PINNs with domain decomposition and adaptive sampling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=xhKsnxJ9FD.

HS Carslow, John Conrad Jaeger, and JE Morral. Conduction of heat in solids. *Journal of engineering materials and technology*, 108(4):378–378, 1986.

Adrian Celaya, David Fuentes, and Beatrice Riviere. Adaptive collocation point strategies for physics informed neural networks via the qr discrete empirical interpolation method. *arXiv preprint arXiv:2501.07700*, 2025.

Aihua Chen and Xuemei Li. Darboux transformation and soliton solutions for boussinesq–burgers equation. *Chaos, Solitons & Fractals*, 27(1):43–49, 2006.

Philippe G Ciarlet and Pierre-Arnaud Raviart. General lagrange and hermite interpolation in rn with applications to finite element methods. *Archive for Rational Mechanics and Analysis*, 46(3):177–199, 1972.

Julian D Cole. On a quasi-linear parabolic equation occurring in aerodynamics. *Quarterly of applied mathematics*, 9(3):225–236, 1951.

Richard Courant, Kurt Friedrichs, and Hans Lewy. Über die partiellen differenzengleichungen der mathematischen physik. *Mathematische annalen*, 100(1):32–74, 1928.

Ron S Dembo and Trond Steihaug. Truncated-newton algorithms for large-scale unconstrained optimization. *Mathematical Programming*, 26(2):190–212, 1983.

Qiang Du, Vance Faber, and Max Gunzburger. Centroidal voronoi tessellations: Applications and algorithms. *SIAM review*, 41(4):637–676, 1999.

Lawrence C Evans. *Partial differential equations*, volume 19. American mathematical society, 2022.

Luca Formaggia and Simona Perotto. New anisotropic a priori error estimates. *Numerische Mathematik*, 89(4):641–667, 2001.

Guido Fubini. *Sugli integrali multipli: nota*. Tipografia della R. Accademia dei Lincei, 1907.

Watson Fulks. On the boundary values of solutions of the heat equation. *Pacific Journal of Mathematics*, 2:141–145, 1952. URL https://api.semanticscholar.org/CorpusID:121022610.

Jean Ginibre and Giorgio Velo. The global cauchy problem for the non linear klein-gordon equation-ii. In *Annales de l'IHP Analyse non linéaire*, volume 6, pp. 15–35, 1989.

Jean-Bastien Grill, Michal Valko, and Remi Munos. Optimistic optimization of a brownian. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/b132ecc1609bfcf302615847c1caa69a-Paper.pdf.

Ulrich G Haussmann and Etienne Pardoux. Time reversal of diffusions. *The Annals of Probability*, pp. 1188–1205, 1986.

Magnus R Hestenes, Eduard Stiefel, et al. Methods of conjugate gradients for solving linear systems. *Journal of research of the National Bureau of Standards*, 49(6):409–436, 1952.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Eberhard Hopf. The partial differential equation. 1950.

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*, 126: 107183, 2023. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2023.107183. URL https://www.sciencedirect.com/science/article/pii/S0952197623013672.

Weizhang Huang. Metric tensors for anisotropic mesh generation. *Journal of computational physics*, 204(2):633–665, 2005.

Ameya D. Jagtap 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), 11 2020. ISSN ISSN 1815-2406. doi: 10.4208/cicp.oa-2020-0164. URL https://www.osti.gov/biblio/2282003.

Xiaowei Jin, Shengze Cai, Hui Li, and George Em Karniadakis. Nsfnets (navier-stokes flow nets): Physics-informed neural networks for the incompressible navier-stokes equations. *Journal of Computational Physics*, 426:109951, 2021. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2020.109951. URL https://www.sciencedirect.com/science/article/pii/S0021999120307257.

Tosio Kato. Strong l p-solutions of the navier-stokes equation in r m, with applications to weak solutions. *Mathematische Zeitschrift*, 187(4):471–480, 1984.

Vladimir Kostic, Karim Lounici, Pietro Novelli, and Massimiliano Pontil. Sharp spectral rates for koopman operator learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 32328–32339. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/663bce02a0050c4a11f1eb8a7f1429d3-Paper-Conference.pdf.

Gerhard Kristensson. A uniqueness theorem for helmholtz' equation: penetrable media with an infinite interface. *SIAM Journal on Mathematical Analysis*, 11(6):1104–1117, 1980.

Peter D Lax and Robert D Richtmyer. Survey of the stability of linear finite difference equations. *Communications on pure and applied mathematics*, 9(2):267–293, 1956.

Jean Leray. Sur le mouvement d'un liquide visqueux emplissant l'espace. *Acta mathematica*, 63(1):193–248, 1934.

Shuning Lin and Yong Chen. A two-stage physics-informed neural network method based on conserved quantities and applications in localized wave solutions. *Journal of Computational Physics*, 457:111053, 2022. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2022.111053. URL https://www.sciencedirect.com/science/article/pii/S0021999122001152.

Qiang Liu, Jason Lee, and Michael Jordan. A kernelized stein discrepancy for goodness-of-fit tests. In *International conference on machine learning*, pp. 276–284. PMLR, 2016.

Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, 1963. ISSN 03684245. URL http://www.jstor.org/stable/2098941.

Michael D McKay, Richard J Beckman, and William J Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61, 2000.

Goeffrey J McLachlan. Mahalanobis distance. *Resonance*, 4(6):20–26, 1999.

Isaiah Elvis Mhlanga and Chaudry Masood Khalique. Exact solutions of generalized boussinesq-burgers equations and (2+ 1)-dimensional davey-stewartson equations. *Journal of Applied Mathematics*, 2012(1):389017, 2012.

Andrea Moiola and Euan A Spence. Is the helmholtz equation really sign-indefinite? *Siam Review*, 56(2):274–312, 2014.

Ben Moseley, Andrew Markham, and Tarje Nissen-Meyer. Finite basis physics-informed neural networks (fbpinns): a scalable domain decomposition approach for solving differential equations: B. moseley et al. *Advances in Computational Mathematics*, 49(4):62, 2023.

Boaz Nadler, Stephane Lafon, Ioannis Kevrekidis, and Ronald Coifman. Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators. *Advances in neural information processing systems*, 18, 2005.

B.-T. Nguyen and D. S. Grebenkov. Localization of laplacian eigenfunctions in circular, spherical, and elliptical domains. *SIAM Journal on Applied Mathematics*, 73(2):780–803, 2013. doi: 10.1137/120869857. URL https://doi.org/10.1137/120869857.

Mingsheng Peng, Hesheng Tang, and Yingwei Kou. Adversarial and self-adaptive domain decomposition physics-informed neural networks for high-order differential equations with discontinuities. *Engineering Applications of Artificial Intelligence*, 145:110156, 2025.

Jiang Qian, Delin Chu, and Roger C. E. Tan. Analyticity of semisimple eigenvalues and corresponding eigenvectors of matrix-valued functions. *SIAM Journal on Matrix Analysis and Applications*, 36(4):1542–1566, 2015. doi: 10.1137/151003799. URL https://doi.org/10.1137/151003799.

M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2018.10.045. URL https://www.sciencedirect.com/science/article/pii/S0021999118307125.

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

Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=St1giarCHLP.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

Luke Travis and Kolyan Ray. Pointwise uncertainty quantification for sparse variational gaussian process regression with a brownian motion prior. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 7419–7442. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/176a579942089c4cdc70136c567932ab-Paper-Conference.pdf.

Georges Voronoi. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. deuxième mémoire. recherches sur les parallélloèdres primitifs. *Journal für die reine und angewandte Mathematik* (*Crelles Journal*), 1908(134):198–287, 1908.

Chenxi Wu, Min Zhu, Qinyang Tan, Yadhu Kartha, and Lu Lu. A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 403:115671, 2023.

Wei Xiong, Xiangyun Long, Stéphane PA Bordas, and Chao Jiang. The deep finite element method: A deep learning framework integrating the physics-informed neural networks with the finite element method. *Computer Methods in Applied Mechanics and Engineering*, 436:117681, 2025.

A Notation

We provide detailed definitions of the notations that are used throughout this study in Table 4.

Table 4: Summary of Notations

Domain	Notation	Description	Notation	Description
PDE	Ω	PDE domain	$\partial\Omega$	Domain boundary
	x	Collocation or input point	$h(x)$	Boundary/initial data
	N_Ω	Interior collocation size	$N_{\partial\Omega}$	Boundary sample size
	$\mathcal{R}$	PDE differential operator	$r_\theta(x)$	PDE residual
	u_θ	PINN solution approximation	θ	Trainable parameters
PINN-ffusion	j	Subdomain index	M	Number of subdomains
	$U_j(\cdot;\theta_j)$	j-th subnetwork	θ_j	Parameters of U_j
	$p(x,t)$	Sampling density	t	Diffusion time
	p_k	Density at iteration k	p_0	Initial probability density
	$K(t,x,y)$	Heat kernel	κ	Heat diffusion strength
	$f(x_t,t)$	Drift term	$g(t)$	Diffusion coefficient
	w_t	Brownian motion	ρ_k	Frozen residual map
	$c_{k,j}$	Subdomain centroid	$s_{k,j}$	Subdomain radius
	$G_{k,j}$	Anisotropic metric matrix	$d^2_{k,j}(z)$	Mahalanobis distance
	$S_{k,j}$	Residual-gradient outer product	$\lambda_{k,j}(x)$	Subdomain partition weight
	$\Sigma_{k,j}$	Weighted covariance	α_k	Geometry mixing weight
	$G^{div}_{k,j}$	Diversity geometry	$G^{unc}_{k,j}$	Uncertainty geometry
	K_w	Warm-up parameter	ϵ, ξ	Stability and scaling constants
Optimization	k	Outer training index	i	Inner truncated Newton (TN) optimization index
	D	Jacobian operator	A_i	System matrix
	η_k	LM damping parameter	d^{LM}_k	Exact LM direction
	$d^{(i)}_k$	TN-approximated direction		

B Algorithm of PINN and PINN-ffusion

Appendix B summarizes the optimization and decomposition procedures of baseline PINN frameworks and the proposed PINN-ffusion. Compared to existing methods, PINN-ffusion jointly updates the sampling distribution and anisotropic domain geometry through diffusion-driven sampling and geometry-aware partition refinement.

B.1 Baseline PINN Framewsorks

Algorithm 1 Physics-Informed Neural Networks (PINN)

Initialize network parameters θ
for $k = 1, \ldots, N_{\text{iter}}$ **do**
  Sample interior points X_f and boundary/initial points X_u
  Compute network output u_θ
  Compute PDE residual loss $\mathcal{L}_f = \frac{1}{|X_f|}\sum_{x\in X_f}\left|\mathcal{N}[u_\theta](x) - f(x)\right|^2$
  Compute data/boundary loss $\mathcal{L}_u = \frac{1}{|X_u|}\sum_{x\in X_u}\left|u_\theta(x) - u(x)\right|^2$
  Form total loss $\mathcal{L} = \lambda_f\mathcal{L}_f + \lambda_u\mathcal{L}_u$
  Update θ by gradient-based optimizer
end for
return u_θ

Algorithm 2 eXtended Physics-Informed Neural Network (XPINN)

Decompose domain into subdomains $\{\Omega_i\}_{i=1}^{M}$
Initialize subnetwork parameters $\{\theta_i\}_{i=1}^{M}$
for $k = 1, \ldots, N_{\text{iter}}$ **do**
  for $i = 1, \ldots, M$ **do**
    Sample interior points $X_f^{(i)}$ and boundary points $X_u^{(i)}$ in Ω_i
    Compute local loss $\mathcal{L}^{(i)} = \lambda_f \mathcal{L}_f^{(i)} + \lambda_u \mathcal{L}_u^{(i)}$
  end for
  Sample interface points X_Γ
  Compute interface loss $\mathcal{L}_\Gamma = \sum_{\Gamma_{ij}} \frac{1}{|X_{\Gamma_{ij}}|} \sum_{x \in X_{\Gamma_{ij}}} \left(|u_{\theta_i}(x) - u_{\theta_j}(x)|^2 + |\nabla u_{\theta_i}(x) - \nabla u_{\theta_j}(x)|^2 \right)$
  Form total loss $\mathcal{L} = \sum_{i=1}^{M} \mathcal{L}^{(i)} + \lambda_\Gamma \mathcal{L}_\Gamma$
  Update $\{\theta_i\}_{i=1}^{M}$
end for
return $\{u_{\theta_i}\}_{i=1}^{M}$

Algorithm 3 Augmented Physics-Informed Neural Network (APINN)

Initialize expert networks $\{u_{\theta_i}\}_{i=1}^{M}$
Initialize gating network g_ϕ
for $k = 1, \ldots, N_{\text{iter}}$ **do**
  Sample interior points X_f and boundary/initial points X_u
  Compute gating weights $w_i(x) = g_\phi(x)$, $\sum_{i=1}^{M} w_i(x) = 1$
  Construct global solution $u(x) = \sum_{i=1}^{M} w_i(x)\, u_{\theta_i}(x)$
  Compute residual loss $\mathcal{L}_f = \frac{1}{|X_f|} \sum_{x \in X_f} \left| \mathcal{N}[u](x) - f(x) \right|^2$
  Compute data/boundary loss $\mathcal{L}_u = \frac{1}{|X_u|} \sum_{x \in X_u} \left| u(x) - u^*(x) \right|^2$
  Form total loss $\mathcal{L} = \lambda_f \mathcal{L}_f + \lambda_u \mathcal{L}_u$
  Update $\{\theta_i\}_{i=1}^{M}$ and ϕ
end for
return $u(x) = \sum_{i=1}^{M} w_i(x)\, u_{\theta_i}(x)$

Algorithm 4 PINN-ball

Initialize local regions $\{B_j\}_{j=1}^{M}$, subnetworks $\{U_j(\cdot; \theta_j)\}_{j=1}^{M}$, and weights $\{\lambda_j\}_{j=1}^{M}$
for $k = 1$ to N_{iter} **do**
  Sample collocation points from p_k
  Set $u_\theta(x) = \sum_{j=1}^{M} \lambda_j(x)\, U_j(x; \theta_j)$
  Compute $\mathcal{L} = \lambda_f \mathcal{L}_f + \lambda_u \mathcal{L}_u$
  Update $\{\theta_j\}_{j=1}^{M}$
  Update p_k by maximizing $\int_\Omega r_\theta(x)^2 p(x)\, dx + \tau \mathcal{M}(p)$
  Update local regions $\{B_j\}_{j=1}^{M}$
end for
return u_θ

B.2 Full Algorithm of Proposed PINN-ffusion

Algorithm 5 summarizes the training procedure of PINN-ffusion referenced in Section 3.3.

Algorithm 5 PINN-ffusion

Initialize $\{c_{0,j}, s_{0,j}, G_{0,j}\}_{j=1}^{M}$, $\{\theta_j\}_{j=1}^{M}$, and p_0
for $k = 1, \ldots, N_{\text{iter}}$ **do**
  Define $\phi_{k,j}(x) = \left(1 - \frac{(x-c_{k,j})^\top G_{k,j}(x-c_{k,j})}{s_{k,j}^2}\right)^2$, $\lambda_{k,j}(x) = \frac{\phi_{k,j}(x)}{\sum_{\ell=1}^{M}\phi_{k,\ell}(x)}$
  Set $u_\theta(x) = \sum_{j=1}^{M} \lambda_{k,j}(x)\, U_j(x;\theta_j)$
  Sample $\{x_i^k\}_{i=1}^{N_\Omega} \sim p_k$
  Compute $\widehat{\mathcal{L}}_k(\theta) = \frac{1}{N_\Omega}\sum_{i=1}^{N_\Omega} \|\mathcal{R}u_\theta(x_i^k) - f(x_i^k)\|_2^2 + \frac{1}{N_{\partial\Omega}}\sum_{m=1}^{N_{\partial\Omega}} \|u_\theta(\hat{x}_m) - g(\hat{x}_m)\|_2^2$
  for $i = 1, \ldots, N_{TN}$ **do**
    Update θ by Truncated Newton
  end for
  Update p_{k+1} by Eq. 7 and obtain $\{z_i^{k+1}\} \sim p_{k+1}$ by integrating Eq. 8 from samples of p_0^{k+1}
  Update $c_{k+1,j}$ by Lloyd steps and set $G_{k+1,j} = (1-\alpha_k)\frac{d\,(\Sigma_{k+1,j}+\epsilon I)^{-1}}{\mathrm{tr}\left((\Sigma_{k+1,j}+\epsilon I)^{-1}\right)} + \alpha_k\left(I + \xi\frac{S_{k+1,j}}{\mathrm{tr}(S_{k+1,j})}\right)$
  Update $s_{k+1,j}$
end for
return u_θ

C Theoretical Details and Proofs

C.1 Theoretical Analysis of Diffusion-based Adaptive Sampling

This section provides the theoretical details underlying the diffusion-based adaptive sampling strategy introduced in Section 3.1. We first derive the heat-equation form of the sampling density from the drift-free diffusion dynamics and show how the corresponding heat-kernel solution can be obtained under Neumann boundary conditions. We then prove that the resulting sampling-weighted residual loss remains consistent with the original PINN objective, which justifies the use of diffusion-based sampling as a stable alternative to adversarial adaptive sampling.

C.1.1 Derivation of Diffusion-based Sampling Density

Let us begin by recalling Eq. 4 of the main text:

$$dx_t = f(x_t, t)dt + g(t)dw_t. \tag{19}$$

As we discussed earlier, the marginal density p of forward SDE satisfies $\partial_t p(x_t, t) = -\nabla \cdot (f(x_t, t)p(x_t, t)) + \frac{1}{2}g(t)^2\Delta p(x_t, t)$. Then, applying the previously discussed assumptions $f = 0$ and $\frac{1}{2}g(t)^2 = \kappa = 1$, we obtain as below.

$$\partial_t p(x_t, t) = \Delta p(x_t, t) \tag{20}$$

Meanwhile, the probability-flow ODE (Song et al., 2021b) that shares the marginal densities $\{p_t\}_{t\in[0,T]}$ of Eq. 4 is given by

$$dx_t = \left[f(x_t, t) - \frac{1}{2}g(t)^2\nabla \log p(x_t, t)\right]dt. \tag{21}$$

With $f = 0$ and $\frac{1}{2}g(t)^2 = \kappa = 1$, Eq. 21 reduces to $dx_t = -\nabla \log p(x_t, t)\, dt$, which is Eq. 8. Substituting the induced velocity field $v(x, t) = -\nabla \log p(x, t)$ into the continuity equation $\partial_t p = -\nabla \cdot (pv)$ recovers Eq. 20:

$$\begin{aligned}
\partial_t p(x, t) &= -\nabla \cdot \big(p(x, t)v(x, t)\big) && (22)\\
&= -\nabla \cdot \big(p(x, t)(-\nabla \log p(x, t))\big) && (23)\\
&= \nabla \cdot \big(p(x, t)\nabla \log p(x, t)\big) && (24)\\
&= \nabla \cdot (\nabla p(x, t)) && (25)\\
&= \Delta p(x, t) && (26)
\end{aligned}$$

Therefore, the SDE in Eq. 4 and the ODE in Eq. 21 share the same family of marginal densities $\{p_t\}_{t\in[0,T]}$. By solving the heat equation, we can now explicitly express p.

We then solve the heat equation on the one-dimensional domain $\Omega = [0, L]$. Let $\partial_x p(0,t) = 0$, $\partial_x p(L,t) = 0$, $p(x,0) = p_0(x)$ and $p(x_t,t) = T(t)X(x)$. Then, we can see that $\partial_t p(x_t,t) = T'(t)X(x)$, $\partial_{xx} p(x_t,t) = T(t)X''(x)$ and $T'(t)X(x) = T(t)X''(x)$ is also true. Additionally, we reconstruct initial and boundary condition as $\partial_t p(0,t) = T(t)X'(0) = 0$, $X'(0) = 0$ and $\partial_x p(L,0) = T(t)X'(L) = 0$, $X'(L) = 0$. Using the method of separation of variable following yields the below.

$$\frac{T'(t)}{T(t)} = \frac{X''(x)}{X(x)} = -\Lambda. \tag{27}$$

According to Eq. 27, we obtain two ODEs.

$$T'(t) + \Lambda T(t) = 0 \tag{28}$$

$$X''(x) + \Lambda X(x) = 0 \tag{29}$$

The spatial eigenvalue problem admits three cases depending on the sign of Λ. For $\Lambda = 0$, we obtain $X''(x) = 0$ and $X(x) = ax + b$ where $a, b \in R$. Since $X'(0) = 0$, we can conclude that $X(x) = b$. Therefore, we acquire the eigenvalue $\Lambda_0 = 0$ (Kostic et al., 2023; Qian et al., 2015) and the eigenfunction $X_0 = 1$ (Nadler et al., 2005; Nguyen & Grebenkov, 2013). For $\Lambda = \sigma^2 > 0$, we obtain the general solution $X(x) = A\cos\sigma x + B\sin\sigma x$ and $X'(x) = -A\sin\sigma x + B\sigma\cos\sigma x$. Since $X'(0) = X'(L) = 0$, we can obtain $B = 0$, $\sin\sigma L = 0$ and $\sigma = \frac{n\pi}{L}$. Therefore, we acquire the eigenvalue $\Lambda_n = (\frac{n\pi}{L})^2$ and the eigenfunction $X_n = \cos\frac{n\pi}{L}$. Finally, for $\Lambda = -\sigma^2 < 0$, we obtain the general solution $X(x) = A\cosh\sigma x + B\sinh\sigma x$ and $X'(x) = A\sigma\sinh\sigma x + B\sigma\cosh\sigma x$. Since $X'(0) = X'(L) = 0$, we obtain $A = B = 0$. However, since it has only trivial solution, there is no solution under the condition $\Lambda = -\sigma^2 < 0$. Also, solution of $T'(t) + \Lambda T(t) = 0$ is given by $T(t) = Ce^{-\Lambda t}$.

To sum up, the final solution $p(x,t)$ is presented by linear combination of eigenfunction as followings.

$$p(x,t) = T(t)X(t) = a_0 + \sum_{n=1}^{\infty} a_n \cos\frac{n\pi}{L} x e^{-\frac{n^2\pi^2}{L^2}t}. \tag{30}$$

Applying the initial condition $p(x,0) = p_0$ determines the expansion coefficients, yielding the heat-kernel representation used in Eq. 7 of the main text.

C.1.2 Proof of Theorem 3.1

This section provides a proof for Theorem 3.1, which establishes the consistency of the diffusion-based sampling loss with the original PINN residual objective. Before proving the theorem, we first verify two basic properties of the Newmann heat kernel used in Eq.7

Proposition C.1 (Integration of Kernel Function K_k). *Define the Neumann heat kernel* $K_k(x,y,t) = \frac{1}{L} + \frac{2}{L}\sum_{n=1}^{\infty} e^{-\frac{n^2\pi^2}{L^2}t}\cos\frac{n\pi}{L}x\cos\frac{n\pi}{L}y$. *Then, for any fixed* $t > 0$ *and* $y \in [0, L]$, *it holds that*

$$\int_{\Omega} K_k(x,y,t)dx = 1. \tag{31}$$

Proof. Since $\Omega = [0, L]$, we can rewrite Eq. 31 as:

$$\int_0^L \frac{1}{L} + \frac{2}{L}\sum_{n=1}^{\infty} e^{-\frac{n^2\pi^2}{L^2}t}dx, \tag{32}$$

where the integration for the first term satisfies:

$$\int_0^L \frac{1}{L}dx = 1, \tag{33}$$

and the integration for the second term satisfies:

$$\int_0^L \frac{2}{L}\sum_{n=1}^{\infty} e^{-\frac{n^2\pi^2}{L^2}t}\cos\frac{n\pi}{L}x\cos\frac{n\pi}{L}ydx = 0. \tag{34}$$

Hence, we obtain

$$\int_0^L K_k(x, y, t)dx = 1. \tag{35}$$

This proves the claim. □

Proposition C.2 (Validity of $p(x,t)$ as Probability Distribution)**.** *Assume that the initial density satisfies* $\int_\Omega p_0(y)dy = 1$*. Then, the density* $p(x,t) = \int_\Omega K_k(x, y, t)p_0(y)dy$ *satisfies*

$$\int_\Omega p(x,t)dydx = 1. \tag{36}$$

Therefore, $p(x,t)$ *is a probability distribution function (PDF).*

Proof. Let us express Eq. 36 as:

$$\int_0^L \int_0^L K_k(x, y, t)p_0(y)dydx. \tag{37}$$

Applying Fubini's theorem (Fubini, 1907; Liu et al., 2016), we can transform Eq. 37 as:

$$\int_0^L p_0(y) \int_0^L K_k(x, y, t)dxdy \tag{38}$$

From the result in proposition C.1, we obtain the following,

$$\int_0^L p_0(y) \int_0^L K_k(x, y, t)dxdy = \int_0^L p_0(y)dy = 1, \tag{39}$$

which completes the proof. □

Finally, we prove Theorem 3.1 and conclude this section. Recall that the sampling-weighted residual loss is defined as:

$$J(\theta_k, p_{t,k}) = \int_\Omega r^2_{\theta_k}(x)p_{t,k}(x)dx, \tag{40}$$

where $r_{\theta_k}(x) = \mathcal{R}[u_{\theta_k}](x) - f(x)$ and $J(\theta_k, p_{t,k}) \geq 0$.

Since $\Omega = [0, L]$ is compact and, for every fixed $t > 0$, the Neumann heat kernel $K_k(\cdot, \cdot, t)$ is continuous and strictly positive on $\Omega \times \Omega$, it attains a positive minimum $m(t) := \min_{x,y\in\Omega} K_k(x, y, t) > 0$. Because p_0^k is a probability density (Proposition C.2), for every $x \in \Omega$ and every k,

$$p_{t,k}(x) = \int_\Omega K_k(x, y, t)\, p_0^k(y)\, dy \geq m(t) \int_\Omega p_0^k(y)\, dy = m(t). \tag{41}$$

Hence

$$J(\theta_k, p_{t,k}) = \int_\Omega r^2_{\theta_k}(x)\, p_{t,k}(x)\, dx \geq m(t)\, \|r_{\theta_k}\|^2_{L^2(\Omega)}. \tag{42}$$

Since $m(t)$ does not depend on k, $J(\theta_k, p_{t,k}) \to 0$ implies $\|r_{\theta_k}\|_{L^2(\Omega)} \to 0$.

For the uniform statement, suppose $\{r_{\theta_k}\}_k$ is uniformly Lipschitz with constant Λ and let $\delta_k := \|r_{\theta_k}\|_{L^\infty(\Omega)}$, attained at $x_k \in \Omega$. Then $|r_{\theta_k}(x)| \geq \delta_k/2$ on $B_k := \{x \in \Omega : |x - x_k| \leq \delta_k/(2\Lambda)\}$, whose length is at least $\min\{\delta_k/(2\Lambda), L\}$. Hence $\|r_{\theta_k}\|^2_{L^2(\Omega)} \geq (\delta_k/2)^2 \min\{\delta_k/(2\Lambda), L\}$, so $\|r_{\theta_k}\|_{L^2(\Omega)} \to 0$ forces $\delta_k \to 0$, which completes the proof.

C.1.3 Multi-dimensional Extension and Irregular Domains

The derivation in Appendix C.1.1 is carried out on the one-dimensional domain $\Omega = [0, L]$. Here we state the multi-dimensional form of the sampling density, describe how non-rectangular geometries are handled, and verify that the guarantee of Theorem 3.1 is preserved in both cases.

Tensor-product structure. Writing the bounding box as a product of per-coordinate intervals,

$$\Omega_{\text{box}} = \prod_{i=1}^{d} [0, L_i], \tag{43}$$

the Neumann heat kernel separates across coordinates, so that

$$K_k(\mathbf{x}, \mathbf{y}, t) = \prod_{i=1}^{d} K_{k,i}(x_i, y_i, t), \tag{44}$$

where each one-dimensional factor $K_{k,i}$ is the Neumann heat kernel obtained in Eq. 7. This factorization is a standard consequence of separation of variables on rectangular domains, and each factor admits the classical method-of-images representation (Carslow et al., 1986). Consequently the density

$$p(\mathbf{x}, t) = \int_{\Omega_{\text{box}}} K_k(\mathbf{x}, \mathbf{y}, t)\, p_0(\mathbf{y})\, d\mathbf{y} \tag{45}$$

and its score $\nabla \log p$ remain available in closed form: no d-dimensional heat equation is discretized and no parametric density estimator is required. Evaluating Eq. 45 costs $O(dN)$ for N collocation points, since the product structure reduces a d-dimensional evaluation to d one-dimensional evaluations. Propositions C.1 and C.2 extend verbatim, as each factor integrates to one and the product of the factors therefore also integrates to one by Fubini's theorem.

Masking and renormalization for irregular domains. The closed-form kernel in Eq. 44 is exact only on a rectangular box. For a non-rectangular geometry we therefore diffuse on the bounding box and restrict the result to the effective domain $\Omega_{\text{eff}} \subseteq \Omega_{\text{box}}$ on which the PDE is posed, masking by an indicator and renormalizing,

$$\tilde{p}_{t,k}(\mathbf{x}) = \frac{p_{t,k}(\mathbf{x})\, \mathbf{1}_{\Omega_{\text{eff}}}(\mathbf{x})}{Z}, \qquad Z := \int_{\Omega_{\text{eff}}} p_{t,k}(\mathbf{x}')\, d\mathbf{x}'. \tag{46}$$

In the Navier–Stokes benchmark this corresponds to $\Omega_{\text{eff}} = \Omega_{\text{box}} \setminus \bar{B}_\rho$, which excludes the cylinder of radius ρ: collocation points proposed inside the cylinder are rejected and the density is renormalized over the fluid region.

The guarantee is preserved under masking. Masking discards part of the mass, so one may ask whether the positive lower bound used in the proof of Theorem 3.1 survives. It does, in a strengthened form.

Lemma C.3 (Masking sharpens the lower bound)**.** *Let $m(t) := \prod_{i=1}^{d} m_i(t) > 0$, where $m_i(t) := \min_{x_i, y_i \in [0, L_i]} K_{k,i}(x_i, y_i, t)$, be the k-independent lower bound of $p_{t,k}$ on Ω_{box} obtained as in the proof of Theorem 3.1. Then*

$$\tilde{p}_{t,k}(\mathbf{x}) \;\geq\; \frac{m(t)}{Z} \;\geq\; m(t) \qquad \text{for all } \mathbf{x} \in \Omega_{\text{eff}}. \tag{47}$$

Proof. On Ω_{eff} the indicator equals one, so $\tilde{p}_{t,k}(\mathbf{x}) = p_{t,k}(\mathbf{x})/Z \geq m(t)/Z$. Since $\Omega_{\text{eff}} \subseteq \Omega_{\text{box}}$ and $p_{t,k} \geq 0$,

$$Z = \int_{\Omega_{\text{eff}}} p_{t,k} \;\leq\; \int_{\Omega_{\text{box}}} p_{t,k} = 1, \tag{48}$$

where the last equality is Proposition C.2. Hence $1/Z \geq 1$ and the claim follows. □

Corollary C.4 (Convergence on the effective domain)**.** *Define $J(\theta_k, \tilde{p}_{t,k}) = \int_{\Omega_{\text{eff}}} r_{\theta_k}^2(\mathbf{x})\, \tilde{p}_{t,k}(\mathbf{x})\, d\mathbf{x}$. Then for every $t > 0$,*

$$\|r_{\theta_k}(\mathbf{x})\|^2_{L^2(\Omega_{\text{eff}})} \;\leq\; \frac{J(\theta_k, \tilde{p}_{t,k})}{m(t)}, \tag{49}$$

and consequently $J(\theta_k, \tilde{p}_{t,k}) \to 0$ implies $\|r_{\theta_k}(\mathbf{x})\|_{L^2(\Omega_{\text{eff}})} \to 0$.

Proof. By Lemma C.3, $\tilde{p}_{t,k} \geq m(t)$ on Ω_{eff}. Since $r_{\theta_k}^2 \geq 0$,

$$J(\theta_k, \tilde{p}_{t,k}) = \int_{\Omega_{\text{eff}}} r_{\theta_k}^2 \, \tilde{p}_{t,k} \;\geq\; m(t) \int_{\Omega_{\text{eff}}} r_{\theta_k}^2 = m(t) \, \|r_{\theta_k}\|_{L^2(\Omega_{\text{eff}})}^2. \tag{50}$$

Dividing by $m(t) > 0$ gives the bound, and the limit statement follows. □

Remark C.5. The guarantee holds on Ω_{eff}, which is precisely the region on which the PDE is posed. The masking approximation used for irregular geometries therefore does not weaken the theoretical guarantee; because $Z \leq 1$, it in fact yields a slightly sharper lower bound than the one available on the bounding box.

C.2 Proof of Theorem 3.3

Minimize the objective function $\Phi(h_{j,1}, ..., h_{j,d}) = \sum_{i=1}^{d} h_{j,i}^2 \mathbf{g}_{j,i}^2$ subject to volume constraint $V_j = \prod_{i=1}^{d} h_{j,i}$.

Therefore, The Lagrangian is given by the following.

$$\mathbf{L} = \sum_{i=1}^{d} h_{j,i}^2 \mathbf{g}_{j,i}^2 - \mu(\prod_{i=1}^{d} h_{j,i} - V_j). \tag{51}$$

Taking the partial derivative with respect to $h_{j,i}$ yields $\frac{\partial \mathbf{L}}{\partial h_{j,i}} = 2h_{j,i}\mathbf{g}_{j,i}^2 - \mu \prod_{l \neq i} h_{j,i}$, and multiplying both sides by $h_{j,i}$ yields the following.

$$2h_{j,i}^2 \mathbf{g}_{j,i}^2 = \mu V_j. \tag{52}$$

Since V_j is independent of i, it is $h_{j,i}\mathbf{g}_{j,i} = const = m \implies h_{j,i} = \frac{m}{\mathbf{g}_{j,i}} \propto \mathbf{g}_{j,i}^{-1}$, where $m = (V_j \prod_{i=1}^{d} \mathbf{g}_{j,i})^{\frac{2}{d}}$.

Therefore, the optimal solution for an anisotropic structure is as follows.

$$\Phi_{aniso}^2 = \sum_{i=1}^{d} h_{j,i}^2 \mathbf{g}_{j,i}^2 = dm^2 = dV_j^{\frac{2}{d}} (\prod_{i=1}^{d} \mathbf{g}_{j,i})^{\frac{2}{d}}. \tag{53}$$

Similarly, we can obtain the following.

$$\Phi_{iso}^2 = \sum_{i=1}^{d} h^2 \mathbf{g}_{j,i}^2 = V_j^{\frac{2}{d}} \sum_{i=1}^{d} \mathbf{g}_{j,i}^2 \tag{54}$$

Finally, the arithmetic-geometric mean inequality for d positive intergers $\mathbf{g}_{j,i}^2, ..., \mathbf{g}_{j,d}^2$ is $\frac{1}{d} \sum_{i=1}^{d} \mathbf{g}_{j,i}^2 \geq (\prod_{i=1}^{d} \mathbf{g}_{j,i}^2)^{\frac{1}{d}}$, and multiplying both sides by $(V_j)^{\frac{2}{d}}$ completes the proof.

$$\Phi_{aniso} = dV_j^{\frac{2}{d}} (\prod_{i=1}^{d} \mathbf{g}_{j,i})^{\frac{2}{d}} \leq V_j^{\frac{2}{d}} \sum_{i=1}^{d} \mathbf{g}_{j,i}^2 = \Phi_{iso}. \tag{55}$$

C.3 Proof of Proposition 5.1

This section provides the proof of Proposition 5.1, which shows that the Truncated Newton (TN) approximation converges to the exact Levenberg–Marquardt (LM) direction as the number of inner iterations increases. The result follows from the fact that the LM update is obtained by solving a symmetric positive definite linear system, and the Truncated Newton inner solve can be implemented using conjugate gradients.

Fix an outer training iteration k, and let $\theta_{k,i}$ denote the current parameter at the i-th innter TN iteration. Define

$$b_i := -D\rho_k(\theta_{k,i})^\top \rho_k(\theta_{k,i}). \tag{56}$$

The exact Lagenberg–Marquardt direction d_k^{LM} is the solution of the linear system

$$A_i d = b_i, \tag{57}$$

where $A_i = D\rho_k(\theta_{k,i})^\top D\rho_k(\theta_{k,i}) + \eta_i I$ for $\eta_i > 0$.

We first show that A_i is symmetric postivie definite. For any nonzero vector v, we have

$$v^\top A_i v = v^\top D\rho_k(\theta_{k,i})^\top D\rho_k(\theta_{k,i})v + \lambda_i v^\top v \tag{58}$$
$$= \|D\rho_k(\theta_{k,i})v\|^2 + \eta_i\|v\|^2. \tag{59}$$

Since $\eta_i > 0$, we have

$$v^\top A_i v > 0 \quad \text{for all } v \neq 0. \tag{60}$$

Hence, A_i is symmetric positive definite. Therefore the system $A_i d = b_i$ has a unique solution, which is precisely the exact LM direction d_k^{LM}.

Now, let $d_k^{(i)}$ be the TN-approximated direction produced after i inner iteration. The standard convergence theory of conjugate gradients for symmetric positive definite linear systems implies that the iterates converge to the unique solution of $A_i d = b_i$ (Hestenes et al., 1952; Saad, 2003), or

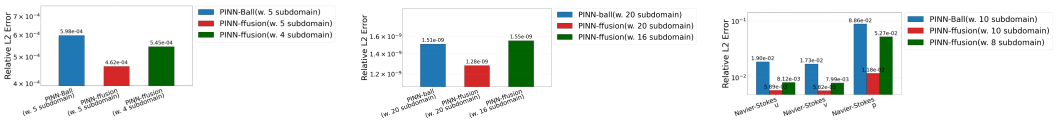
$$\lim_{i\to\infty} \|d_k^{(i)} - d_k^{\mathrm{LM}}\| = 0. \tag{61}$$

This completes the proof.

D Additional Experiments on Decomposition Efficiency

Appendix D provides additional experiments evaluating the efficiency of the proposed anisotropic domain decomposition on PDE benchmarks beyond the Burgers equation discussed in the main text. Similar to the analysis in Section 4.3, we compare PINN-ball and PINN-ffusion from two complementary perspectives: (1) accuracy under the same number of subdomains, and (2) the number of required subdomains to achieve comparable accuracy.

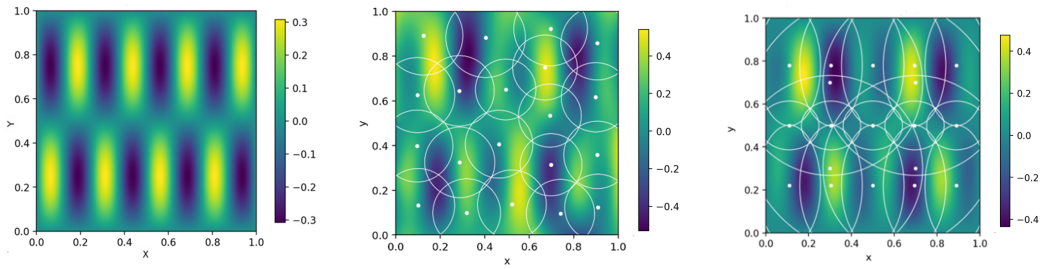
As shown in Figure 6, the proposed PINN-ffusion consistently achieves lower error under the same decomposition budget, while also requiring fewer subdomains to reach similar performance levels across multiple PDE benchmarks. These results further support the effectiveness of anisotropic decomposition for capturing directional and localized residual structures.



(a) Burgers Setting (b) Helmholtz Setting (c) Navier–Stokes Setting

Figure 6: Comparison of Different Ball Setting

In addition, Figure 7 qualitatively compares the solutions of isotropic and anisotropic decomposition on the Helmholtz equation. The anisotropic subdomains of PINN-ffusion better align with elongated solution structures than the isotropic subdomains of PINN-ball.



(a) Ground Truth (b) Solution of PINN-ball (c) Solution of PINN-ffusion

Figure 7: Comparison of Isotropic and Anisotropic Domain Decomposition

E PDE Benchmark Details and PINN-ffusion Solution Visualizations

Appendix E provides detailed descriptions of the PDE benchmark problems used in our experiments, including their governing equations, boundary and initial conditions, and solution visualizations by PINN-ffusion. These benchmarks cover a range of challenging behaviors, such as oscillatory structures, shock formation, and nonlinear spatiotemporal dynamics, allowing comprehensive evaluation of the proposed PINN-ffusion framework under diverse PDE settings.

E.1 Helmholtz Equation

The Helmholtz equation is a second-order linear PDE that frequently appears in wave propagation, acoustics, and frequency-domain physics. Although the equation is linear, it remains challenging for neural PDE solvers because its solution can contain high-frequency structures. This makes the Helmholtz benchmark useful for evaluating whether a method can accurately capture spatially oscillatory patterns. In this study, we consider the two-dimensional Helmholtz equation defined on the domain $\Omega = [0, 1]^2$ (Raissi, 2018):

$$\Delta u + k_x k_y u = f, \qquad x, y \in \Omega, \tag{62}$$

with Dirichlet boundary condition:

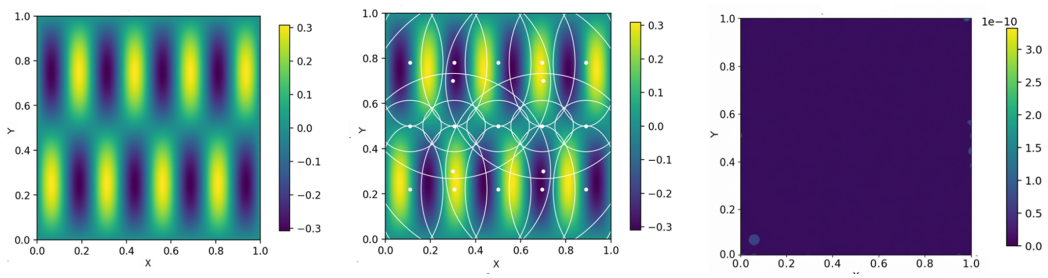
$$u(x, y) = 0, \qquad x, y \in \partial\Omega. \tag{63}$$

Here, the forcing term is given by

$$f(x, y) = -k_x k_y \sin(k_x x) \sin(k_y y). \tag{64}$$

Following the experimental setting used in the baselines (Hu et al., 2023), we set $k_x = 4$ and $k_y = 1$.

During training, we use $N_r = 10^4$ collocation points to evaluate the PDE residual, where the initial collocation points are generated using Latin hypercube sampling (McKay et al., 2000). For the boundary condition, we use $N_b = 3 \cdot 10^3$ boundary points sampled on $\partial\Omega$. Figure 8 shows the ground-truth solution, the predicted solution obtained by PINN-ffusion, and the corresponding pointwise squared error. The visualization confirms that PINN-ffusion accurately captures the oscillatory solution structure.



(a) Ground-truth Solution (b) Predicted Solution (c) Pointwise Squared Error

Figure 8: PINN-ffusion Results on Helmholtz Equation

E.2 Burgers' Equation

Burgers' equation can be interpreted as a one-dimensional form of the Navier–Stokes equations without the pressure term. In this benchmark, the solution develops a sharp transition over time, making it challenging for neural networks due to spectral bias and localized residual concentration. We consider the same Burgers' equation setting as in (Raissi, 2018), and we use the exact reference data provided by the authors. Specifically, on the spatitemporal domain $(x, \tau) \in \Omega = [-1, 1] \times [0, 1]$, the equation is given by:

$$\partial_\tau u + u\partial_x u - \nu \partial_{xx} u = 0, \qquad (x, \tau) \in \Omega. \tag{65}$$

with the initial condition given by:

$$u(x, 0) = -\sin(\pi x), \qquad x \in [-1, 1], \tag{66}$$

and the boundary condition given by:

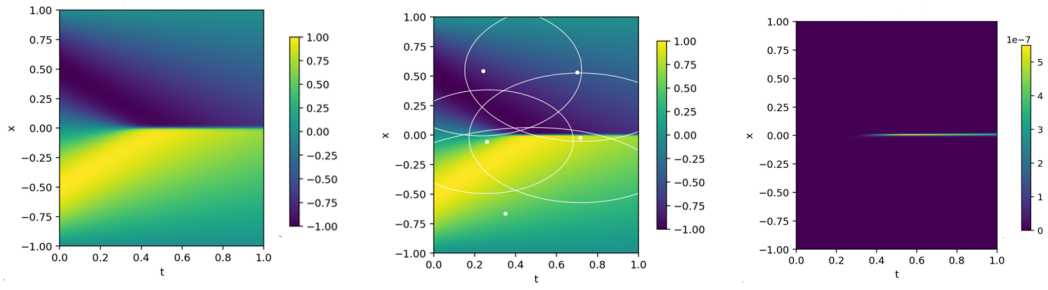
$$u(-1, \tau) = u(1, \tau) = 0, \qquad \tau \in [0, 1]. \tag{67}$$

The diffusion coefficient is set to

$$\nu = \frac{0.01}{\pi}. \tag{68}$$

The exact solution used for evaluation is taken from the publicly available data provided in (Raissi, 2018).

During training, we use $N_r = 10^4$ residual collocation points that are initially generated by Latin hypercube sampling. In addition, $N_b = 3 \cdot 10^3$ points are sampled on $\partial\Omega$ to enforce the initial and boundary conditions. Figure 9 presents the result of the best-performing PINN-ffusion model for Burgers' equation. The error map highlights whether the model accurately resolves the sharp transition region, which is the most difficult part of this benchmark.



(a) Ground-truth Solution (b) Predicted Solution (c) Pointwise Squared Error

Figure 9: PINN-ffusion Results on Burgers' Equation

E.3 Klein–Gordon Equation

The Klein–Gordon equation is a second-order hyperbolic PDE in which a mass term or reaction term has been added to the wave equation. In this study, we consider the nonlinear Klein–Gordon equation (Hu et al., 2023) on $(x,t) \in [0,1] \times [0,1]$, given by:

$$u_{tt} - u_{xx} + u^3 = f(x,t), \tag{69}$$

with initial conditions:

$$u(x,0) = u_t(x,0) = 0 \tag{70}$$

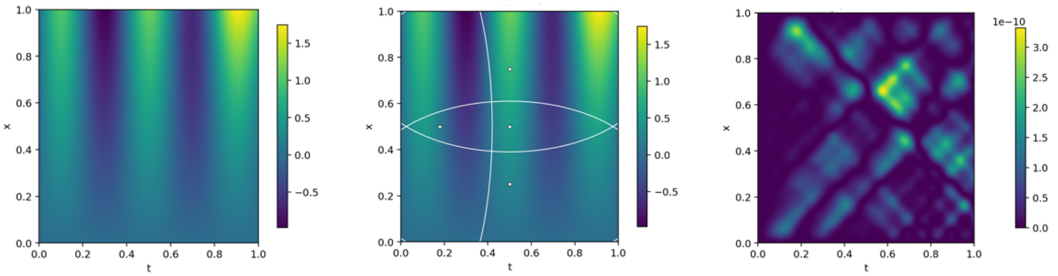
and boundary conditions:

$$u(x,t) = h(x,t), \qquad x \in \{0,1\},\ t \in [0,1]. \tag{71}$$

The nonlinearity of the equation arises from the cubic reaction term u^3. The boundary and source terms are determined from the reference solution:

$$u(x,t) = x\cos(5\pi t) + (xt)^3. \tag{72}$$

During training, we use $N_r = 10^4$ residual collocation points that are initially generated by Latin hypercube sampling. In addition, $N_b = 3 \cdot 10^3$ points are sampled on $\partial\Omega$ to enforce the initial and boundary conditions. Figure 10 presents the result of the best-performing PINN-ffusion model for Klein–Gordon equation. The visualization demonstrates how accurately our model captures the nonlinear oscillatory dynamics of the solution.



(a) Ground-truth Solution (b) Predicted Solution (c) Pointwise Squared Error

Figure 10: PINN-ffusion Results on Klein–Gordon Equation

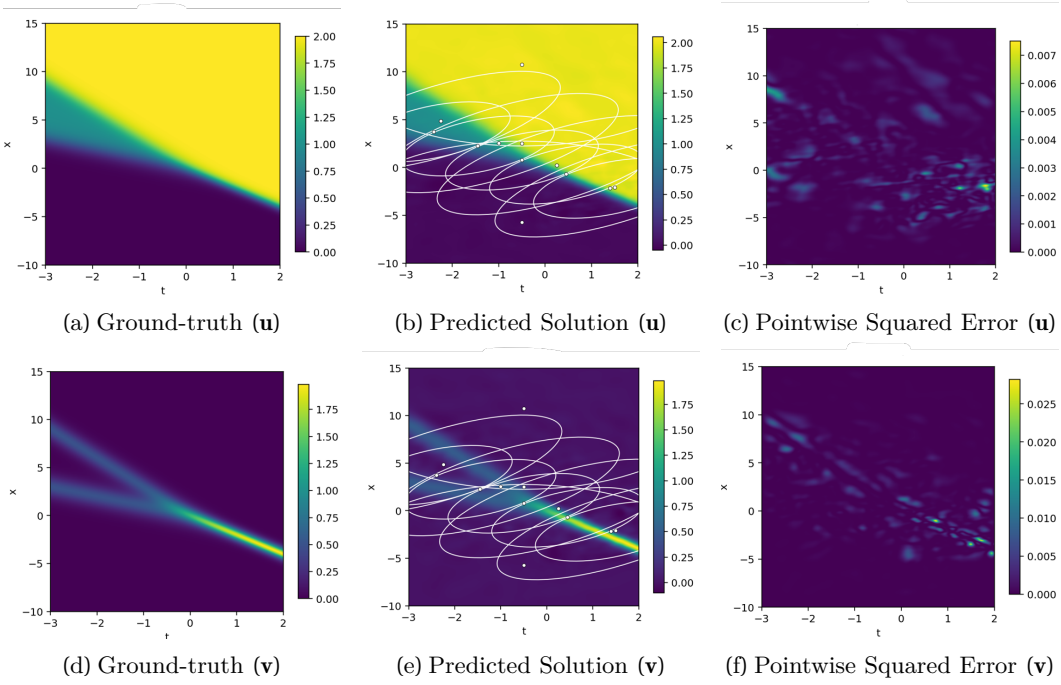
E.4 Boussinesq–Burger Equation

The Boussinesq–Burger equation is a coupled nonlinear PDE system that models nonlinear wave propagation. A thorough understanding of such a solution is important in order to apply it to harbor and coastal designs. Unlike scalar PDE benchmarks, this problem involves two interacting solution components, making it suitable for evaluating whether the proposed framework can handle coupled dynamics and component-wise residual structures. Following the prior works (Hu et al., 2023), we consider the Boussinesq–Burger system defined on $x \in [-10, 15]$, $t \in [-3, 2]$, with two unknown functions $\mathbf{u}(x,t)$ and $\mathbf{v}(x,t)$. The governing equations are given by:

$$\begin{aligned} \mathbf{u}_t &= 2\mathbf{u}\mathbf{u}_x + \frac{1}{2}\mathbf{v}_x, \\ \mathbf{v}_t &= \frac{1}{2}\mathbf{v}_{xxx} + 2(\mathbf{u}\mathbf{v})_x, \end{aligned} \tag{73}$$

where the Dirichlet boundary condition and the ground truth solution are obtained from the benchmark setting used in prior work (Lin & Chen, 2022).

During training, we use $N_r = 10^4$ residual collocation points that are initially generated by Latin hypercube sampling. In addition, $N_b = 3 \cdot 10^3$ points are sampled on $\partial\Omega$ to enforce the initial and boundary conditions. Figure 11 presents the result of the best-performing PINN-ffusion model for Boussinesq–Burger equation. These visualizations allow component-wise evaluation of the proposed method on a coupled nonlinear PDE system.



(a) Ground-truth (**u**) (b) Predicted Solution (**u**) (c) Pointwise Squared Error (**u**)

(d) Ground-truth (**v**) (e) Predicted Solution (**v**) (f) Pointwise Squared Error (**v**)

Figure 11: PINN-ffusion Results on Boussinesq–Burger System

E.5 Navier–Stokes Equation

The final PDE benchmark considered in this study is the incompressible Navier–Stokes equation, one of the most important PDE systems in fluid dynamics. Compared with the previous benchmarks, this problem is substantially more challenging because it involves nonlinear advection, viscosity, incompressibility constraints, and multiple coupled solution components. In particular, we consider the two-dimensional flow around a cylinder introduced in (Jin et al., 2021). Following the same experimental setting, the computational domain is $\Omega = [-2.5, 7.5] \times [-2.5, 2.5] \times [0, 16]$, and the objective is to approximate the vector-valued function $u : \Omega \to \mathbb{R}^3$, represented as:

$$u(x, y, t) = [\mathbf{u}(x, y, t), \mathbf{v}(x, y, t), \mathbf{p}(x, y, t)]^T, \tag{74}$$

where $\mathbf{u}$ and $\mathbf{v}$ denote the horizontal and vertical velocity components, respectively, and $\mathbf{p}$ denotes the pressure. The incompressible Navier–Stokes equations are given by:

$$\partial_t \mathbf{u} + \mathbf{u}\partial_x \mathbf{u} + \mathbf{v}\partial_y \mathbf{u} - \frac{1}{Re}(\partial_{xx}\mathbf{u} + \partial_{yy}\mathbf{u}) + \partial_x \mathbf{p} = 0, \qquad (x, y, t) \in \Omega, \tag{75}$$

$$\partial_t \mathbf{v} + \mathbf{u}\partial_x \mathbf{v} + \mathbf{v}\partial_y \mathbf{v} - \frac{1}{Re}(\partial_{xx}\mathbf{v} + \partial_{yy}\mathbf{v}) + \partial_y \mathbf{p} = 0, \qquad (x, y, t) \in \Omega, \tag{76}$$

$$\partial_x \mathbf{u} + \partial_y \mathbf{v} = 0, \qquad (x, y, t) \in \Omega. \tag{77}$$

The initial conditions are given by

$$\mathbf{u}(x, y, 0) = g_{\mathbf{u}0}(x, y), \qquad (x, y) \in [-2.5, 7.5] \times [-2.5, 2.5], \tag{78}$$

$$\mathbf{v}(x, y, 0) = g_{\mathbf{v}0}(x, y), \qquad (x, y) \in [-2.5, 7.5] \times [-2.5, 2.5]. \tag{79}$$

The boundary condition is given by

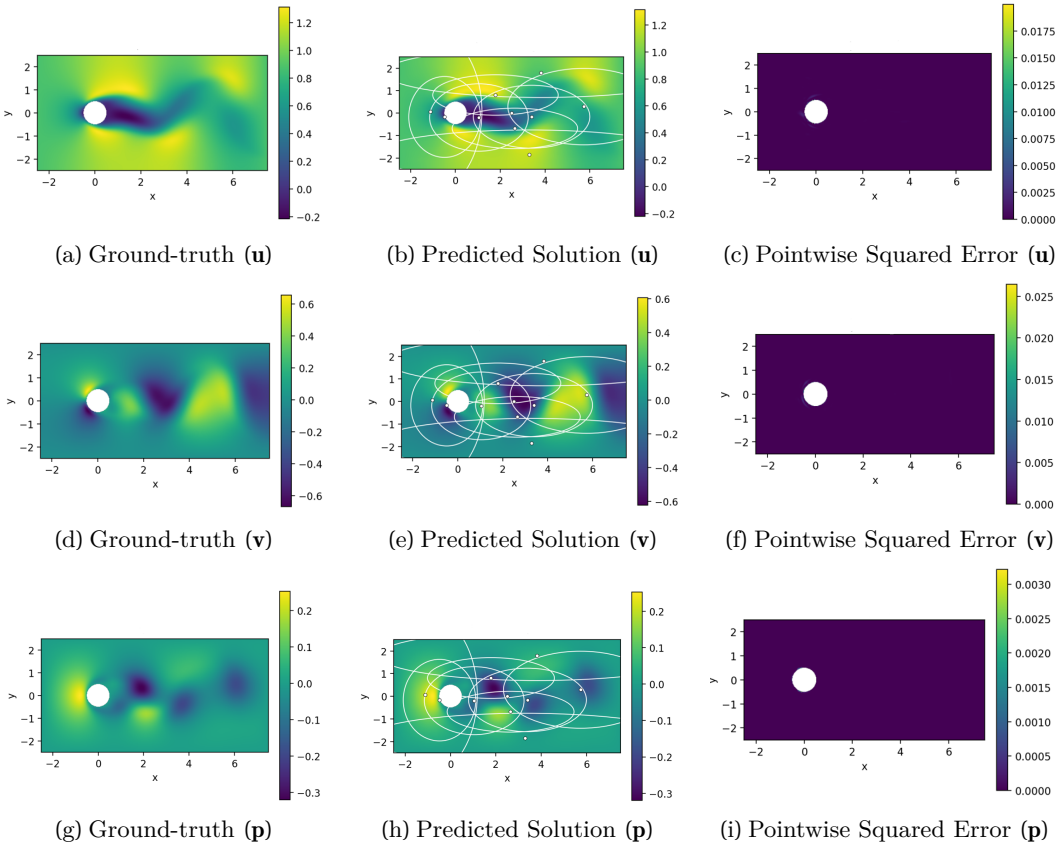
$$\mathbf{u}(2.5, y, t) = 1, \qquad (y, t) \in [-2.5, 2.5] \times [0, 16], \tag{80}$$

$$\mathbf{v}(2.5, y, t) = 0, \qquad (y, t) \in [-2.5, 2.5] \times [0, 16]. \tag{81}$$

We set $Re = 100$, which denotes the Reynolds number, a dimensionless quantity determined by the problem setting. The initial condition data $(g_{\mathbf{u}0}, g_{\mathbf{v}0})$ are obtained from the repository

of (Raissi, 2018), and the corresponding high-fidelity solution from the same source is used as the ground truth for model evaluation. In this problem, the inlet velocity is imposed at $x = 2.5$. In addition, a circular cylinder with center $(x, y) = (0, 0)$ and radius 0.25 is placed inside the computational domain, which introduces an additional boundary condition on the cylinder surface. Moreover, on the upper and lower boundaries, namely $y = \pm 2.5$, one may impose either the no-slip condition $\mathbf{u} = \mathbf{v} = 0$ or the exact solution as the boundary condition. Since the simulation data provided in (Raissi, 2018) correspond to a free-flow setting, we use the exact solution on these boundaries.

For training, we use $N_r = 5 \cdot 10^5$ residual collocation points that are generated by Latin hypercube sampling. To enforce the initial and boundary conditions, we use $N_b = 2 \cdot 10^4$ points sampled on $\partial\Omega$. At each iteration, the loss is minimized using random mini-batches from the training data, with 10^4 points for the residual loss and $5 \cdot 10^3$ points for the initial and boundary condition loss. Figure 12 shows the prediction results of the best-performing PINN-ffusion model for the Navier–Stokes equations, together with the learned domain decomposition. Since pressure is not directly normalized by the model, the pressure error is computed after comparing normalized pressure fields.



(a) Ground-truth (**u**) (b) Predicted Solution (**u**) (c) Pointwise Squared Error (**u**)

(d) Ground-truth (**v**) (e) Predicted Solution (**v**) (f) Pointwise Squared Error (**v**)

(g) Ground-truth (**p**) (h) Predicted Solution (**p**) (i) Pointwise Squared Error (**p**)

Figure 12: PINN-ffusion Results on Navier–Stokes System

F Details of Experimental Settings

Appendix F provides detailed explanations on the implementation of PINN-ffusion, with hyperparameter settings summarized in Table 5.

F.1 Training Configuration

We compare PINN-ffusion against PINN, XPINN, APINN, and PINN-ball under comparable training settings. PINN and XPINN are trained using the Adam optimizer for 100,000 iterations, while APINN combines Adam (100,000 iterations) and L-BFGS (20,000 iterations) optimization following the original implementation. PINN-ball and PINN-ffusion are trained using the Truncated Newton (TN) method for 2,000 outer iterations. At each outer iteration, 64 inner TN iterations are performed to approximate the Levenberg–Marquardt update.

To ensure fair comparison with baseline methods, the neural network architectures in PINN-ffusion were designed to maintain a similar number of trainable parameters across all PDE benchmarks. Since the spatial and temporal resolutions differ depending on the PDE, we adjusted both the network depth (i.e., number of layer) and width (i.e., number of neurons per each layer) to keep the total parameter count approximately around 5,000 while accounting for the complexity of each problem.

F.2 Domain Decomposition Settings

For anisotropic domain decomposition, the geometry matrix controls the shape and orientation of each ellipsoidal subdomain. During training, excessively elongated ellipsoids may extend far beyond the effective PDE region due to severe imbalance between the major and minor axes. In such cases, the corresponding partition weights can become overly diluted, leading to unstable optimization and degraded convergence behavior. To mitigate this issue, the anisotropic scaling of each subdomain is carefully constrained during training. Having said that, we set the number of subdomains as 20 for Helmholtz, 5 for Burgers, 4 for Klein–Gordon, 16 for Boussinesq–Burger, and 10 for Navier–Stokes.

Table 5: Detailed experimental configurations for each PDE benchmark

Hyperparameter	PDE Equation				
	Helmholtz	Burgers	Klein–Gordon	Boussinesq–Burger	Navier–Stokes
Number of layers	7	8	2	6	5
Width	15	15	28	16	10
Number of subdomains	20	5	4	12	10
N_b	3,000	3,000	3,000	3,000	20,000
N_r	10,000	10,000	10,000	10,000	500,000
Outer training iterations	2,000	2,000	2,000	2,000	2,000
Inner TN iterations	64	64	64	64	64

G Additional Experiments: Ablation, Sensitivity, and Computational Cost

G.1 Domain-Decomposition Comparison

We provide a formal comparison among domain-decomposition strategies along three axes: whether the subdomains are learned during training, whether the collocation points are adaptively sampled, and what geometry the decomposition admits. The comparison is summarized in Table 1 of the main text.

XPINN and FBPINN (Moseley et al., 2023) keep the decomposition fixed and sample non-adaptively, differing only in the admissible geometry, which is arbitrary for the former and restricted to rectangles for the latter. APINN makes the partition data-dependent through implicit gating, but the gating is not trained jointly and the sampling remains non-adaptive. AS-PINNs(Peng et al., 2025) is the first to couple a dynamic decomposition with sampling, though its sampling is non-adaptive and its interfaces are polygonal. PINN-ball couples a dynamic domain with adaptive sampling, but its subdomain geometry is restricted to isotropic balls.

Our method is the only one that is dynamic and adaptive while learning an anisotropic elliptical geometry, which can align with the directional structure of the error field. It is

also the only sampling-coupled method that replaces the adversarial minimax objective with a non-adversarial diffusion process, allowing the two components to be combined stably without collapse or additional regularization.

G.2 Statistical Significance

Table 6: One-sided Wilcoxon rank-sum test of PINN-ball versus PINN-ffusion. Significance levels: $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$.

Equation	U	p-value
Helmholtz	37	0.455
Burgers	27	0.044^{*}
Klein-Gordon	8	$<0.001^{***}$
Boussinesq-Burger(**u**)	25	0.031^{*}
Boussinesq-Burger(**v**)	46	0.422
Navier-Stokes(**u**)	8	$<0.001^{***}$
Navier-Stokes(**v**)	11	0.002^{**}
Navier-Stokes(**p**)	0	$<0.001^{***}$

A symmetric two-tailed treatment is inappropriate here, since the error is non-negative and right-skewed, so that a symmetric mean±sd interval can produce a negative lower bound. We therefore replace the parametric t-test with a non-parametric one-sided Wilcoxon rank-sum test, testing whether the error of PINN-ball is larger than that of PINN-ffusion. The U statistics and p-values are reported in Table 6.

The improvement is statistically significant on six of the eight components. The effect is strongest on Klein-Gordon and on the Navier–Stokes components, where U drops to 8, 8, 11, and 0 with $p < 0.001$, $p < 0.001$, $p = 0.002$, and $p < 0.001$, respectively, and remains significant at the 5% level on Burgers ($U = 27$, $p = 0.044$) and on the u component of Boussinesq–Burger ($U = 25$, $p = 0.031$).

The two non-significant cases are Helmholtz ($U = 37$, $p = 0.455$) and the v component of Boussinesq–Burger ($U = 46$, $p = 0.422$), for which a mean improvement over PINN-ball is still observed. Helmholtz in particular is a case where both methods are near saturation (error $\approx 10^{-9}$), so the residual difference is negligible rather than absent. The gains are thus strongly significant precisely on the harder nonlinear systems, which is where a directional, anisotropic error structure is present.

G.3 Effect of Initialization

Table 7: Median and IQR of the error over 10 independent initializations.

Equation	ffusion median	ffusion IQR	ball median	ball IQR
Helmholtz	4.34e-10	5.89e-10	1.05e-9	1.27e-9
Burgers	4.12e-4	2.98e-4	6.41e-4	3.43e-4
Klein-Gordon	2.30e-7	3.99e-7	8.99e-5	4.38e-5
Boussinesq-Burger(**u**)	6.17e-3	2.07e-3	8.41e-3	5.22e-3
Boussinesq-Burger(**v**)	9.22e-2	3.52e-2	9.56e-2	9.37e-2
Navier-Stokes(**u**)	4.00e-3	5.89e-3	1.82e-2	1.26e-2
Navier-Stokes(**v**)	4.76e-3	7.48e-3	1.59e-2	1.09e-2
Navier-Stokes(**p**)	9.06e-3	1.52e-2	8.48e-2	1.14e-2

To assess sensitivity to initialization, we analyze 10 independent runs with different initializations. Since a non-significant t-test does not establish equivalence, we instead report the median and the IQR of the non-negative, right-skewed errors and compare the dispersion against PINN-ball, as shown in Table 7.

The median error of our method is lower on all eight components, with the margin spanning more than two orders of magnitude on Klein-Gordon (2.30e-7 versus 8.99e-5) and nearly an order of magnitude on the pressure component of Navier–Stokes (9.06e-3 versus 8.48e-2), while the smallest margin occurs on the v component of Boussinesq–Burger (9.22e-2 versus 9.56e-2).

Our method also shows a smaller IQR on seven of the eight components, reducing the dispersion from 4.38e-5 to 3.99e-7 on Klein-Gordon, from 9.37e-2 to 3.52e-2 on the v component of Boussinesq–Burger, and from 1.26e-2 to 5.89e-3 on the u component of Navier–Stokes. The only exception is the pressure component of Navier–Stokes, where the IQR is 1.52e-2 against 1.14e-2, although the median error there remains almost an order of magnitude lower. A one-sided Wilcoxon signed-rank test on the paired IQRs confirms that PINN-ball has significantly larger dispersion with $p < 0.05$. These results indicate that diffusion sampling reduces adversarial-update instability and improves robustness to initialization.

G.4 Ablation Study

We ablate the two components of PINN-ffusion, diffusion sampling and anisotropic decomposition, by toggling each within the same framework, so that PINN-ball (adversarial sampling with isotropic decomposition) serves as the both-off baseline and PINN-ffusion as the both-on model. All four combinations are evaluated on Navier–Stokes for the u, v, and p components, and the results are reported in Table 3 of the main text.

Each component is effective on its own: enabling either one alone already reduces the error far below the both-off baseline, from 1.90e-02 to 7.94e-03 (diffusion off, anisotropic on) and 8.42e-03 (diffusion on, anisotropic off) on u, with the same pattern on v (1.73e-02 to 7.11e-03 and 7.65e-03) and p (8.86e-02 to 4.87e-02 and 4.17e-02). The two are complementary rather than redundant: the full model attains the lowest error on every component (5.89e-03 on u, 5.82e-03 on v, 1.18e-02 on p), and the joint gain exceeds either single-component gain.

The dominant component differs by field. For the velocity components, anisotropic decomposition alone outperforms diffusion alone (7.94e-03 vs. 8.42e-03 on u; 7.11e-03 vs. 7.65e-03 on v), whereas for the pressure component the ordering reverses (4.17e-02 for diffusion alone vs. 4.87e-02 for anisotropy alone). Combining both yields the largest improvement on the pressure component, reducing the error from 8.86e-02 to 1.18e-02.

G.5 Comparison with Contemporary Adaptive-Sampling Baselines

Table 8: Comparison against residual-based adaptive-sampling baselines on Navier–Stokes.

Method	u	v	p
QR-DEIM (Celaya et al., 2025)	8.70e-3	8.65e-3	3.39e-2
QR-DEIM-R (Celaya et al., 2025)	8.16e-3	8.30e-3	3.27e-2
RAD (Wu et al., 2023)	9.02e-3	8.85e-3	3.51e-2
PINN-ffusion(Ours)	5.89e-3	5.82e-3	1.18e-2

We compare against contemporary residual-based adaptive-sampling methods by swapping only the sampling mechanism within our framework, which isolates the contribution of the diffusion component. The results on Navier–Stokes are reported in Table 8.

Our method outperforms all three baselines on every component. On u, the error drops from 8.16e-3–9.02e-3 for the baselines to 5.89e-3, and on v from 8.30e-3–8.85e-3 to 5.82e-3. The margin is largest on p, where the baselines range from 3.27e-2 to 3.51e-2 while our method reaches 1.18e-2, roughly a threefold reduction over the strongest baseline. Among the baselines themselves, QR-DEIM-R is the most accurate and RAD the least, but the gap between them is small relative to the gap separating all of them from our method, indicating that the improvement stems from the diffusion-based sampling rather than from residual-based refinement in general.

G.6 Sensitivity Analysis

Table 9: Sensitivity to ξ on Navier–Stokes.

ξ	u	v	p
1	5.94e-03	5.88e-03	1.19e-02
3	5.89e-03	5.82e-03	1.18e-02
5	5.91e-03	5.87e-03	1.26e-02
7	5.87e-03	5.98e-03	1.29e-02

Table 10: Sensitivity to ϵ on Navier–Stokes.

ϵ	u	v	p
1e-2	5.91e-3	5.88e-03	1.19e-02
5e-2	5.90e-3	5.82e-03	1.18e-02
1e-3	5.89e-3	5.87e-03	1.26e-02
5e-3	5.95e-3	5.98e-03	1.29e-02

Table 11: Sensitivity to the diffusion step t on Navier–Stokes.

t	10	20	50	80	100
u	6.08e-03	5.89e-03	7.73e-03	1.97e-02	2.73e-02
v	6.53e-03	5.82e-03	7.12e-03	2.22e-02	3.07e-02
p	1.35e-02	1.18e-02	1.43e-02	4.21e-02	5.36e-02

Table 12: Sensitivity to K_w on Navier–Stokes.

K_w	u	v	p
400	6.02e-03	5.98e-03	1.37e-02
600	5.59e-03	5.53e-03	1.26e-02
800	5.89e-03	5.82e-03	1.18e-02
1000	6.01e-03	5.96e-03	1.66e-02

We conduct a sensitivity analysis on Navier–Stokes by varying one hyperparameter at a time while fixing the others at their default values ($\xi = 3$, $\epsilon = 10^{-3}$, $K_w = 800$, 10 subdomains, and $t = 20$). All tables report the u, v, and p components.

Robustness to ξ, ϵ, and K_w PINN-ffusion is robust to ξ and ϵ. Across $\xi \in \{1, 3, 5, 7\}$, the error varies only within 5.87e-03–5.94e-03 on u, 5.82e-03–5.98e-03 on v, and 1.18e-02–1.29e-02 on p. Similarly, across $\epsilon \in \{10^{-3}, 5 \times 10^{-3}, 10^{-2}, 5 \times 10^{-2}\}$, the error remains within 5.89e-03–5.95e-03 on u, 5.82e-03–5.98e-03 on v, and 1.18e-02–1.29e-02 on p. For the warm-up parameter $K_w \in \{400, 600, 800, 1000\}$, the velocity errors stay within 5.59e-03–6.02e-03 on u and 5.53e-03–5.98e-03 on v, with $K_w = 600$ yielding the lowest velocity errors. The pressure error is more sensitive to K_w, ranging from 1.18e-02 at $K_w = 800$ to 1.66e-02 at $K_w = 1000$, but it remains far below that of PINN-ball (8.86e-02 in Table 2).

Number of Subdomains Increasing the number of subdomains from 6 to 8 and 10 monotonically reduces the error on every component, from 9.07e-03 to 8.12e-03 and 5.89e-03 on u, from 8.99e-03 to 7.99e-03 and 5.82e-03 on v, and from 8.33e-02 to 5.27e-02 and 1.18e-02 on p. The pressure is the most affected, with a roughly sevenfold reduction from 6 to 10 subdomains, since a finer decomposition allocates more subnetworks to localized structures.

Diffusion Step On the velocity components, the diffusion step t has the largest effect among the hyperparameters. The error is lowest at $t = 20$ on every component (5.89e-03 on u, 5.82e-03 on v, and 1.18e-02 on p) and slightly higher at $t = 10$ (6.08e-03, 6.53e-03, and 1.35e-02). For larger values, it increases monotonically, reaching 2.73e-02 on u, 3.07e-02 on v, and 5.36e-02 on p at $t = 100$. A large t over-smooths the sampling density toward the uniform distribution, weakening its concentration on high-residual regions. Therefore, a small t around 20 is both cheaper and more accurate.

G.7 Boundary Layers and Collocation-Point Visualization

Using the Burgers equation, we examine where the collocation points concentrate during training. Figure 3 in the main text overlays the sampled points on the learned subdomain

Table 13: Sensitivity to the number of subdomains on Navier–Stokes.

Subdomains	u	v	p
6	9.07e-03	8.99e-03	8.33e-02
8	8.12e-03	7.99e-03	5.27e-02
10	5.89e-03	5.82e-03	1.18e-02

structure in the (t, x) plane, where the solution develops a sharp transition along $x = 0$ that steepens as t increases.

Both methods move points toward the high-residual sharp-transition region, so the sampling mechanism itself is effective in either case. The difference lies in the geometry of the subdomains that the points are allocated within. Our method aligns its anisotropic subdomains with the elongated error structure, producing a thin, horizontally extended band that follows the transition along $x = 0$ from roughly $t = 0.3$ onward, with the remaining points spread over the smooth region. The isotropic balls of PINN-ball cannot elongate along the transition, so the corresponding structure is a wider and less directed region around $x = 0$, and the sampled points are correspondingly more diffuse in x.

This confirms that the anisotropic decomposition allocates points more effectively along thin residual features, which is consistent with the component-wise ablation in Table 3, where anisotropy contributes most on the directional velocity components.

G.8 Network Size, Training Time, and Memory

Table 14: Parameter count, wall-clock time per epoch, and peak memory under matched budgets.

Method	Params	Time/epoch	Memory
PINN	5080	0.2s	878MB
XPINN	5140	3.5s	946MB
APINN	5376	6.6s	1GB
PINN-ball	4980	45.2s	38GB
PINN-ffusion	4910	36.5s	36GB

Under matched parameter budgets (∼5k) on identical hardware (RTX A6000 Ada), we compare the computational cost as well as the number of parameters across methods, as reported in Table 14. All methods use a comparable parameter count, ranging from 4910 to 5376, so the comparison reflects the method rather than model capacity.

The methods separate into two groups by cost. The first-order methods are inexpensive, requiring 0.2s to 6.6s per epoch and under 1GB of memory. The second-order methods are substantially heavier, at 45.2s and 36.5s per epoch with 38GB and 36GB of memory, since the cost is dominated by the second-order optimizer rather than by the parameter count.

The controlled pair is therefore PINN-ffusion versus PINN-ball, which share the same second-order optimizer and iteration budget, so that the remaining difference is attributable to the sampling mechanism. Our method is both faster and lighter, reducing the time per epoch from 45.2s to 36.5s (a 19% reduction) and the peak memory from 38GB to 36GB, while also using the fewest parameters (4910). The gain arises because our sampling distribution has no learnable parameters, which removes the per-iteration Adam updates required by the adversarial sampler of PINN-ball.

H Code Implementation

We provide our implementation codes as 1) a zip file in the Supplementary material and 2) the anonymous GitHub link of https://github.com/AnonumousAccount651321648/PINN-ffusion.