APOD: Adaptive PDE-Observation Diffusion for Physics-Constrained Sampling

Ruichen Xu [1] **Haochun Wang** [1] **Georgios Kementzidis** [1] **Chenhao Si** [2] **Yuefan Deng** [1]

Abstract

In many parametric PDE problems, partial or noisy observations pose a serious challenge for building robust world models that also respect physical constraints. We introduce Adaptive PDE-Observation Diffusion (APOD), a novel framework that dynamically couples PDE constraints with measurement data during the reverse diffusion sampling process. At each denoising iteration, APOD balances a PDE-consistency term derived from governing equations and a data-fidelity term informed by partial observations, guiding the model to produce physically valid solutions. This balanced enforcement naturally handles sparse or noisy data, alleviates mismatches between PDE residuals and diffusion steps, and enhances solution diversity. Empirical results demonstrate APOD's ability to yield accurate, reliable solution even under uncertainty and limited measurements. Our approach paves a principled way to generate high-fidelity parametric PDE solutions in world-model-based reasoning for scientific and engineering domains.

1. Introduction

Recent advances in *world modeling* emphasize the need for frameworks that reliably capture complex systems with limited, often noisy, observations. Many of these systems in science and engineering are governed by partial differential equations (PDEs), which describe physical, biological, or chemical phenomena under diverse conditions. Traditional solvers (e.g., finite element or finite difference methods) provide accurate solutions for individual PDE instances but become computationally heavy when confronted with large parameter spaces or high-dimensional inverse problems. This challenge is exacerbated in real-world applications where data may be incomplete, making it difficult to learn a robust *world model* that aligns with both observed measurements and underlying physical laws.

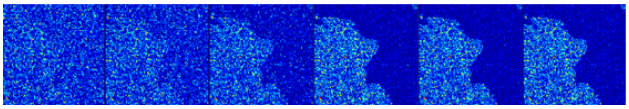

Figure 1. **Unstable PDE enforcement at initial diffusion steps.** Early samples are overwhelmed by noise, and the PDE residuals (seen as random speckles) do not carry meaningful physical structure. Consequently, applying strong PDE constraints at these stages can hinder convergence. This demonstrates the need to adjust PDE weighting as the diffusion process evolves.

Many recent operator-learning approaches (Li et al., 2020; 2021; Kovachki et al., 2021; Raissi et al., 2019) partly address this issue by training neural networks to map parameter fields (e.g., boundary conditions, coefficients) to PDE solutions. Such methods offer improved scalability in multi-query or partial-observation scenarios, yet they often assume deterministic mappings or demand extensive coverage of the parameter domain. As a result, they may struggle in highly uncertain settings where measurements are scarce or noisy, a scenario frequently encountered in real-world *world model* construction.

Diffusion-based generative models offer a complementary solution by naturally accommodating uncertainty and generating multiple valid realizations of a PDE state. Instead of producing a single deterministic outcome, diffusion models learn a forward process that adds noise to PDE-related fields and a reverse process that iteratively "denoises" samples, guiding them back toward plausible configurations (Song et al., 2020). This generative perspective readily extends to partial-observation tasks: a PDE constraint can be integrated as a *guidance mechanism*—ensuring solutions adhere to known physics—while a data-fidelity term enforces alignment with any sparse or noisy measurements. Crucially, these methods allow flexible enforcement of PDE constraints at different stages of denoising, making them well-suited to building data-aware *world models*.

However, balancing PDE constraints against observation-driven objectives remains a core challenge for diffusion-based PDE solvers. During the initial phases of the reverse diffusion process, the presence of strong Gaussian noise in

[1]Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY, USA [2]School of Data Science, The Chinese University of Hong Kong, Shenzhen, China. Correspondence to: Yuefan Deng <yuefan.deng@stonybrook.edu>.

ICML 2025 Workshop on Assessing World Models

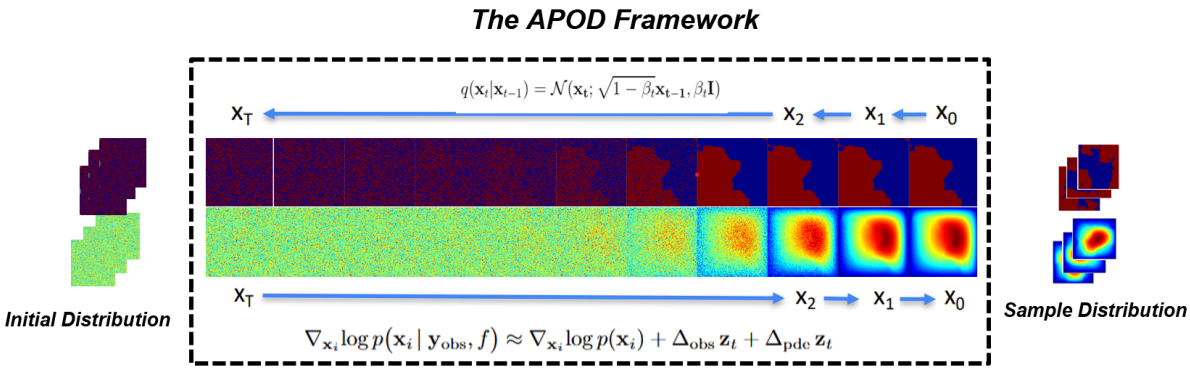

Figure 2. **The APOD Framework.** Illustration of the generative process from x_T (noisy initial distributions, left) to x_0 (sharpened samples, right). The top row shows silhouettes merging from noise to shape, while the bottom row shows probability mass focusing over iterations. The PDE-inspired gradients guide the diffusion steps, as indicated by $\nabla_{x_i} \log p(x_i \mid y_{\text{obs}}, f) \approx \nabla_{x_i} \log p(x_i) + \Delta_{\text{obs}}\, \mathbf{z}_t + \Delta_{\text{pde}}\, \mathbf{z}_t$.

the samples results in significant volatility in the PDE residuals. Overly strong PDE weighting can destabilize sampling, while weak weighting may never fully impose physical constraints. Fixed hyperparameters for PDE-to-observation loss ratios (Huang et al., 2024) can yield suboptimal performance across varying PDE types, geometries, and noise regimes. Consequently, adaptive strategies are essential for robust enforcement of both physical and measurement-derived constraints when building PDE-based *world models*.

We propose an *adaptive guided diffusion* framework that addresses these limitations through three key innovations:

1. **Adaptive coefficient updates**: We dynamically refine or clamp PDE parameters in real time based on partial observations, ensuring that each diffusion step reflects the latest measurement information.

2. **Domain-adaptive PDE loss**: We focus PDE enforcement where it matters most (e.g., near boundaries or in high-gradient regions), making better use of limited computational resources.

3. **Adaptive PDE vs. observation loss balancing**: We smoothly transition between enforcing PDE constraints and incorporating measurements, preventing noisy intermediate states from overwhelming the PDE gradient.

By integrating these strategies, our method converges to physically meaningful solutions under extremely sparse or noisy measurement conditions (see Appendix A for a broader discussion on AI4PDE). We demonstrate effectiveness on the PDE task—Darcy flow—and show improved accuracy and stability over existing diffusion-based and operator-learning baselines. Our results suggest that incorporating adaptive, physics-aware guidance into diffusion models can substantially enhance *world modeling* capabilities for complex, real-world systems.

2. Preliminaries

In this section, we first refer readers to Appendix **??** for a comprehensive description of the stationary and dynamic PDE problem settings considered (along with associated inverse problems). We then review the basics of diffusion models in Section 2.1 and show how to integrate PDE constraints into a generative sampling process in Section 2.2.

2.1. Diffusion Models and Score-Based Generative Processes

Diffusion models progressively add Gaussian noise to data and learn to reverse this process. Following (Song et al., 2020), one can describe reverse-time updates via

$$\frac{d\mathbf{x}}{dt} = -\dot{\sigma}(t)\,\sigma(t)\,\nabla_{\mathbf{x}} \log p\big(\mathbf{x}; \sigma(t)\big), \tag{1}$$

where $\sigma(t)$ is a noise schedule and $p(\mathbf{x}; \sigma)$ the distribution of noisy samples. A learned denoiser $D(\mathbf{x}; \sigma)$ approximates the score function via

$$\nabla_{\mathbf{x}} \log p\big(\mathbf{x}; \sigma(t)\big) \approx \frac{D(\mathbf{x}; \sigma(t)) - \mathbf{x}}{\sigma(t)^2}. \tag{2}$$

This framework underpins state-of-the-art generative performance in various domains (Ho et al., 2020; Yang et al., 2023).

2.2. Guided Diffusion for PDE-Constrained Inverse Problems

When partial or noisy observations $\mathbf{y}$ are available, they can guide the reverse diffusion via an additional gradient term. In *diffusion posterior sampling* (Mammadi et al., 2023), this

can be written as

$$d\mathbf{x} = -\dot{\sigma}(t)\,\sigma(t)\left[\nabla_{\mathbf{x}}\log p\big(\mathbf{x};\sigma(t)\big) + \nabla_{\mathbf{x}}\log p\big(\mathbf{y}\,|\,\mathbf{x}\big)\right]dt, \tag{3}$$

often approximated by penalizing measurement mismatch $\|\mathbf{y} - \mathcal{M}(\hat{\mathbf{x}}_N^i)\|_2^2$ at each denoising step i. To incorporate PDE constraints, one adds $\nabla_{\mathbf{x}}\mathcal{L}_{\text{pde}}$, where $\mathcal{L}_{\text{pde}}$ quantifies residuals $f(\cdot) = 0$. Concretely,

$$\nabla_{\mathbf{x}_i}\log p\big(\mathbf{x}_i\,|\,\mathbf{y}_{\text{obs}}, f\big) \approx \nabla_{\mathbf{x}_i}\log p(\mathbf{x}_i) - \zeta_{\text{obs}}\nabla_{\mathbf{x}_i}\mathcal{L}_{\text{obs}} - \zeta_{\text{pde}}\nabla_{\mathbf{x}_i}\mathcal{L}_{\text{pde}}, \tag{4}$$

enforcing both observation consistency and PDE feasibility (Huang et al., 2024; Cheng et al., 2025a).

3. Method

In this section, we detail our proposed *Adaptive PDE-Observation Diffusion* (APOD) strategy. The key idea is to iteratively update a latent variable z that represents the PDE solution (and possibly other parameters) while respecting both partial observations and PDE constraints.

Below, we first introduce the observation loss in Section 3.1, then define the region-focused PDE loss in Section 3.2, discuss how to update z using each loss, and finally combine everything in the APOD algorithm in Section 3.3.

3.1. Observation Loss

Suppose we have m partial observations $\{(x_j, y_j)\}_{j=1}^m$, where x_j denotes a point in the domain (or domain-time for time-dependent problems) and y_j is the measured solution value. A common way to measure how well z fits these observations is via a squared-error mismatch:

$$\mathcal{L}_{\text{obs}}(z) \;=\; \sum_{j=1}^{m}\big\|z(x_j) - y_j\big\|^2.$$

Minimizing this loss encourages z to match the known data at each observation point. This setup directly parallels partial-observation scenarios discussed in Section 2.2, where PDE constraints are incorporated into the diffusion framework via additional penalties and updates. For broader treatments of PDE-constrained inverse problems, see also (Raissi et al., 2019; Huang et al., 2024).

3.2. Region-Focused PDE Loss

Enforcing the PDE strictly across the entire domain can be both expensive and detrimental if the data are very sparse. Instead, we introduce a *localized* PDE loss around each observation point x_j by defining a neighborhood of radius r_j:

$$B_{r_j}(x_j) \;=\; \{\,x \in \Omega : \|x - x_j\| \le r_j\}.$$

If $\mathcal{P}(z(x)) = f(z(x))$ characterizes our PDE, we measure its residual in these local neighborhoods:

$$\mathcal{L}_{\text{pde},j}(z(x)) \;=\; \int_{B_{r_j}(x_j)}\big|\mathcal{P}(z(x))\big|^2\,\mathrm{d}x.$$

Summing over j then yields

$$\mathcal{L}_{\text{pde}}(z(x)) \;=\; \sum_{j=1}^{m}\mathcal{L}_{\text{pde},j}(z).$$

For comparison, a *global* PDE loss such as in (Huang et al., 2024) might integrate the residual over the entire domain,

$$\widetilde{\mathcal{L}}_{\text{pde}}(z) \;=\; \int_{\Omega}\big|\mathcal{P}(z(x))\big|^2\,\mathrm{d}x.$$

Compared to this global approach—where the PDE residual is enforced everywhere—our localized formulation provides two key benefits. First, we only compute the PDE residual in neighborhoods of observed points, avoiding expensive calculations in unobserved (and possibly irrelevant) regions. Second, we naturally weight the PDE loss by data density: if a position is near multiple observations, it is included in multiple neighborhoods and thus exerts stronger influence on $\mathcal{L}_{\text{pde}}$. As a result, the solution is guided more aggressively toward satisfying the PDE near regions that contain more observed information.

3.3. Iterative Correction Steps

We perform two main correction steps at each diffusion iteration: one for the observation loss and one for the PDE constraint. First, we ensure agreement with the true values from the observations $\{(x_j, y_j)\}$. Specifically, we want our solution z_t at each observed point x_j to match the measured y_j. One way to achieve this is to solve

$$z_{t+1} \;=\; \min_{z'}\Big[\mathcal{L}_{\text{obs}}(z') \;+\; \lambda_{\text{obs}}\,\big\|z_t - z'\big\|\Big],$$

where $\mathcal{L}_{\text{obs}}(z')$ enforces $z'(x_j) = y_j$ at each observed location. In many practical cases, we can directly use the *ideal* field $\widetilde{y}$ that matches every measurement. According to **Theorem C.1**, this leads to the closed-form update

$$\Delta_{\text{obs}} z_t \;=\; z_{t+1} - z_t \;=\; \frac{\widetilde{y} \;-\; z_t}{\lambda_{\text{obs}} + 1},$$

which nudges z_t toward the true observations while balancing proximity to its previous state.

Next, we refine the solution to enforce the PDE locally. Let $\mathcal{F}_{\text{pde}}(z)$ measure the PDE residual (e.g., via finite differences in neighborhoods around each observation point). Suppose we solve

$$z_{t+1} \;=\; \min_{z'}\Big[\mathcal{F}_{\text{pde}}(z') \;+\; \lambda_{\text{pde}}\,\big\|z_t - z'\big\|\Big].$$

If the PDE discretization is *linear*, i.e. $\mathbf{Az} = \mathbf{b}$ with constant $\mathbf{b}$, we can set $\mathcal{F}_{\text{pde}}(z) = \frac{1}{2}\|\mathbf{A}\,z - \mathbf{b}\|^2$ and apply **Theorem C.2** to obtain the closed-form correction

$$\Delta_{\text{pde}} z_t \;=\; \big(\mathbf{A}^\top\mathbf{A} + \lambda_{\text{pde}}\,\mathbf{I}\big)^{-1}\Big(\mathbf{A}^\top\mathbf{b} + \lambda_{\text{pde}}\,z_t\Big) \;-\; z_t.$$

However, if the PDE is *nonlinear* or depends on z in a more complex way, we cannot write $\mathbf{Az} = \mathbf{b}$ explicitly. In that case, **Theorem C.3** describes how to apply an iterative solver (e.g., Newton's method) by forming a local linearization at z_t:

$$R(z_t) \;=\; \nabla_z \mathcal{F}_{\text{pde}}(z_t), \quad J(z_t) \;=\; \nabla_z^2 \mathcal{F}_{\text{pde}}(z_t),$$

and then updating via

$$\Delta_{\text{pde}} z_t \;=\; -\big[J(z_t) + \lambda_{\text{pde}}\,\mathbf{I}\big]^{-1} R(z_t).$$

In either case, once $\Delta_{\text{pde}} z_t$ is computed, we refine z_t to satisfy the PDE more accurately. By combining the *observation* update and *PDE* update at each iteration, we iteratively guide z_t to satisfy both data fidelity and physical constraints.

Comparing APOD (details at Appendix 1) with a standard *fixed-penalty* diffusion (Huang et al., 2024), one might update $\mathbf{z}_t$ by descending the combined loss gradient:

$$\mathbf{z}_{t+1} \;=\; \mathbf{z}_t \;-\; \eta\Big[\zeta_{\text{obs}}\,\nabla\mathcal{L}_{\text{obs}}(\mathbf{z}_t) \;+\; \zeta_{\text{pde}}\,\nabla\mathcal{L}_{\text{pde}}(\mathbf{z}_t)\Big],$$

where η is a learning rate, and $\zeta_{\text{obs}}, \zeta_{\text{pde}}$ are fixed penalty coefficients. By contrast, APOD *decouples* these losses into separate steps and can focus PDE enforcement selectively (e.g., via local neighborhoods). This leads to greater flexibility and often better performance under sparse or noisy observations.

4. Experiments

We evaluate our approach on the classic Darcy Flow problem, modeling fluid flow through a porous medium. Our specific setup uses the static form:

$$-\nabla\cdot\big(a(c)\,\nabla u(c)\big) \;=\; q(c), \quad u(c) \;=\; 0 \quad \text{on } \partial\Omega, \quad (5)$$

where $q(c) = 1$ is a constant forcing term and a takes on binary values. This represents different material properties across the domain. We focus on three tasks: *Forward* (recover u given a), *Backward* (recover a from partial or noisy observations of u), and *Both* (recover both a and u jointly).

We generate random binary fields for a on a two-dimensional grid and solve (5) to obtain the corresponding solution u. Both **DiffusionPDE** (Huang et al., 2024) and our **APOD*** (detailed in Appendix E) are then used to reconstruct the unknown fields for each task. We measure performance using the relative L^2-error versus the ground truth.

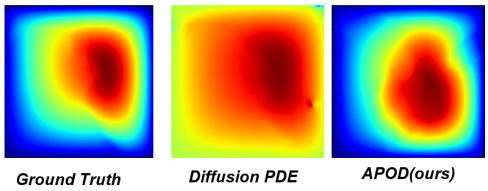

Figure 3. **Comparison of Darcy Flow Reconstructions.** (Left) Ground Truth, (Center) DiffusionPDE, (Right) APOD. Each panel shows color-coded solutions for the pressure u (top row) and the material property a (bottom row), where warmer colors indicate larger values. Visual inspection shows that APOD yields reconstructions more faithful to the ground truth, particularly near interface boundaries.

Table 1 reports the mean relative errors. For the forward task, **APOD*** obtains a lower error of 0.047 compared to 0.088 for DiffusionPDE, demonstrating improved solution recovery when a is known. In the backward setting, APOD* is less accurate, with an error of 0.305 versus 0.153 for DiffusionPDE. In the joint "both" scenario, the two methods trade advantages: DiffusionPDE recovers a slightly better (0.037 vs. 0.040), while APOD* yields a more accurate u (0.047 vs. 0.060).

Table 1. Relative errors for each PDE method, measured in forward, backward, and both-side settings (subdivided into fields a and u).

	Forward	Backward	Both	
Method			a	u
DiffusionPDE	0.088	0.153	0.037	0.060
APOD	0.047	0.305	0.040	0.047

5. Discussion and Future Work

Our experiments on Darcy Flow demonstrate the promise of incorporating adaptive, physics-aware guidance into diffusion-based PDE solvers. By dynamically balancing PDE constraints against partial observations, we achieve improved accuracy and stability compared to standard operator-learning and diffusion approaches. However, these advantages come at the cost of additional matrix inversion steps for PDE residual computation, which can become expensive in high-dimensional or complex geometries.

In future work, we plan to address this computational overhead, potentially by integrating efficient linear solvers (Luo et al., 2025) or other preconditioned iterative methods. We also intend to apply our framework to a broader range of PDE tasks, such as wave equations, Navier–Stokes flows, and advanced inverse problems, comparing against state-of-the-art models like Fourier Neural Operators (Li et al.,

2020; Gao et al., 2024), Physics-Informed Neural Networks (Raissi et al., 2019), and domain-decomposition PINN variants (Jin et al., 2021; Si & Yan, 2025).

References

Cheng, C., Han, B., Maddix, D. C., Ansari, A. F., Stuart, A., Mahoney, M. W., and Wang, Y. Hard constraint guided flow matching for gradient-free generation of PDE solutions. In *International Conference on Learning Representations (ICLR)*, 2025a.

Cheng, C., Han, B., Robinson, D., Ansari, A., and Stuart, A. Gradient-free generation for hard-constrained systems. amazon.science, 2025b.

Dasgupta, A., Ramaswamy, H., and Murugito-Esandi, J. Conditional score-based diffusion models for solving inverse elasticity problems. *Computer Methods in Applied Mechanics and Engineering*, 2025.

Gao, W. and Wang, C. Active learning based sampling for high-dimensional nonlinear partial differential equations. *Journal of Computational Physics*, 475:111848, 2023.

Gao, W., Xu, R., Wang, H., and Liu, Y. Coordinate transform fourier neural operators for symmetries in physical modelings. *Transactions on Machine Learning Research*, 2024.

Gao, W., Luo, J., Xu, R., and Liu, Y. Dynamic schwartz-fourier neural operator for enhanced expressive power. *Transactions on Machine Learning Research*, 2025a. URL https://openreview.net/forum?id=B0E2yjrNb8.

Gao, W., Xu, R., Deng, Y., and Liu, Y. Discretization-invariance? on the discretization mismatch errors in neural operators. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=J9FgrqOOni.

He, Z. and Reina, C. EVODMs: Variational learning of PDEs for stochastic systems via diffusion models with quantified epistemic uncertainty. *arXiv preprint arXiv:2502.10588*, 2025.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Huang, J., Yang, G., Wang, Z., and Park, J. J. DiffusionPDE: Generative PDE-solving under partial observation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Hughes, T. J. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Courier Corporation, 2012.

Jia, Y., Miao, S., Wu, J., Yang, M., Hu, C., and Li, X. Flow-Aware Navigation of Magnetic Micro-Robots in Complex Fluids via PINN-Based Prediction. *arXiv preprint arXiv:2503.11124*, 2025.

Jin, X., Cai, S., Li, H., and Karniadakis, G. E. NSFnets (Navier–Stokes Flow Nets): Physics-informed neural networks for the incompressible Navier–Stokes equations. *Journal of Computational Physics*, 426:109951, 2021.

Kovachki, N., Li, Z., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A., and Anandkumar, A. Neural operator: Learning maps between function spaces. *arXiv preprint arXiv:2108.08481*, 2021.

LeVeque, R. J. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002.

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. Physics-informed neural operator for learning partial differential equations. *arXiv preprint arXiv:2103.17115*, 2021.

Lino, M., Pfaff, T., and Thuerey, N. Learning Distributions of Complex Fluid Simulations with Diffusion Graph Networks. *arXiv preprint arXiv:2504.02843*, 2025.

Luo, J., Wang, J., Wang, H., Geng, Z., Chen, H., Kuang, Y., et al. Neural krylov iteration for accelerating linear system solving. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2025.

Mammadi, A., Berner, J., Azizzadenesheli, K., and Ye, J. C. Diffusion-Based Inverse Solver on Function Spaces With Applications to PDEs. ml4physicalsciences.github.io, 2023.

Mistrangelo, F. Investigation of 4DVarNet Algorithm for Image Reconstruction of Suspended Particulate Matter Dynamics Data. essay.utwente.nl, 2024.

Oommen, V., Bora, A., and Zhang, Z. Integrating neural operators with diffusion models improves spectral representation in turbulence modeling. *arXiv preprint arXiv:XXXX.XXXXX*, 2024.

Oommen, V., Bora, A., and Zhang, Z. Integrating neural operators with diffusion models improves spectral representation in turbulence modelling. *Proceedings of the Royal Society A*, 2025.

Plessix, R.-E. A review of the adjoint-state method for computing the gradient of a functional with geophysical

applications. *Geophysical Journal International*, 167(2): 495–503, 2006.

Raissi, M., Perdikaris, P., and Karniadakis, G. E. 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.

Saad, Y. *Iterative Methods for Sparse Linear Systems*. SIAM, 2003.

Si, C. and Yan, M. Initialization-enhanced physics-informed neural network with domain decomposition (idpinn). *Journal of Computational Physics*, 530:113914, 2025.

Si, C., Yan, M., Li, X., and Xia, Z. Complex physics-informed neural network. *arXiv preprint arXiv:2502.04917*, 2025.

Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. 2015.

Song, Y., Sohl-Dickstein, J., Kingma, D., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Stuart, A. M. Inverse problems: A bayesian perspective. *Acta Numerica*, 19:451–559, 2010.

Sun, L., Gao, H., Pan, S., Wang, M., Bai, Y., Luo, X., and Liang, L. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Computer Methods in Applied Mechanics and Engineering*, 361:112732, 2020.

Valencia, M. L., Pfaff, T., and Thuerey, N. Learning Distributions of Complex Fluid Simulations with Diffusion Graph Networks. In *The Thirteenth International Conference on ...*, 2024. openreview.net.

Xu, R., Wu, Z., Chen, L., Kementzidis, G., Wang, S., Wang, H., Shi, Y., and Deng, Y. Velocity-inferred hamiltonian neural networks: Learning energy-conserving dynamics from position-only data. *arXiv preprint arXiv:2505.02321*, 2025.

Yang, L., Zhang, Z., Song, Y., et al. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4):1–39, 2023.

Yue, X., Zhu, L., and Yang, Y. Deltaphi: Learning physical trajectory residual for pde solving, 2024. URL https://arxiv.org/abs/2406.09795.

Yue, X., Yang, Y., and Zhu, L. Holistic physics solver: Learning pdes in a unified spectral-physical space, 2025. URL https://arxiv.org/abs/2410.11382.

Zhuang, Y., Cheng, S., and Duraisamy, K. Spatially-aware diffusion models with cross-attention for global field reconstruction with sparse observations. *Computer Methods in Applied Mechanics and Engineering*, 2025.

Zirvi, R., Tolooshams, B., and Anandkumar, A. Projected Low-Rank Gradient in Diffusion-based Models for Inverse Problems. In *NeurIPS 2024 Workshop on Data-Driven PDEs*, 2024. openreview.net.

A. Related Work

Related Work. Classical numerical solvers (Hughes, 2012; LeVeque, 2002; Saad, 2003) achieve high accuracy in deterministic PDE tasks but can become infeasible for large-scale or parameter-rich problems. Data-centric methods such as Physics-Informed Neural Networks (PINNs) (Raissi et al., 2019) embed PDE constraints within neural networks and have shown promising results in fluid mechanics (Jin et al., 2021), elasticity (Sun et al., 2020), and related fields. However, training PINNs in complex or high-dimensional inverse settings often demands significant computational effort and meticulous tuning (Plessix, 2006; Stuart, 2010; Yue et al., 2025; 2024).

Neural operators offer a flexible alternative by learning solution operators directly, enabling discretization-invariant or symmetry-aware PDE solvers. Recent research advances include: Coordinate Transform Fourier Neural Operators for capturing PDE symmetries (Gao et al., 2024), active learning strategies to handle high-dimensional PDE sampling (Gao & Wang, 2023), dynamic domain decomposition techniques that enhance expressive power (Gao et al., 2025a), and discretization-invariance analyses that mitigate mismatch errors (Gao et al., 2025b). Other notable developments extend PDE-based modeling to Hamiltonian systems from position-only data (Xu et al., 2025) or introduce domain decomposition for initialization (Si & Yan, 2025) and complex-valued PDE frameworks (Si et al., 2025). Neural Krylov methods (Luo et al., 2025) further accelerate linear system solvers arising from PDEs.

In parallel, diffusion models (Sohl-Dickstein et al., 2015; Song et al., 2020) have gained prominence for PDE tasks by jointly encoding physical constraints and data fidelity while naturally accommodating uncertainty. Integration with neural operators has shown efficacy in turbulence modeling (Oommen et al., 2024; 2025), inverse elasticity (Dasgupta et al., 2025), and sparse-field reconstructions via wavelet and cross-attention modules (Zhuang et al., 2025). This diffusion-based paradigm has also been applied to high-dimensional fluid-flow distributions (Lino et al., 2025; Valencia et al., 2024) and PDE sampling without explicit gradient-based optimization (Cheng et al., 2025b). Recent approaches include EVODMs (He & Reina, 2025), physics-aware projections (Zirvi et al., 2024), and expansions to practical scientific applications such as microrobotics (Jia et al., 2025) or imaging-based PDE reconstructions (Mistrangelo, 2024). These advances underscore the growing versatility of diffusion-powered methods in scientific *world modeling* under partial or noisy observations.

B. PDE Problem Settings

B.1. Static (Time-Independent) PDEs

We consider a stationary PDE of the form

$$\begin{aligned} \mathcal{P}\big(x;\, a,\, u\big) &= f\big(x;\, a,\, u\big), \quad x \in \Omega, \\ u(x) &= g(x), \quad x \in \partial\Omega, \end{aligned} \tag{6}$$

where $\Omega \subset \mathbb{R}^d$ is the spatial domain, $a(x)$ is a parameter field, and $u(x)$ is the PDE solution subject to the boundary condition $g(x)$. Examples in this category include elliptic PDEs, Darcy's equation, and the steady-state heat equation. Under partial observations, the objective is to recover the entire parameter $a(x)$, the entire solution $u(x)$, or both, depending on available data.

B.2. Dynamic (Time-Dependent) PDEs

We now consider a time-dependent PDE of the form

$$\begin{aligned} \mathcal{R}\big(x,t;\, a,\, u\big) &= f\big(x,t;\, a,\, u\big), \quad (x,t) \in \Omega \times (0,T], \\ u(x,t) &= g(x,t), \quad x \in \partial\Omega,\ t \in (0,T], \\ u(x,0) &= a(x), \quad x \in \Omega. \end{aligned} \tag{7}$$

Examples in this class include the wave equation, heat equation, Navier–Stokes, and convection–diffusion. Under partial observations, the objective might be to recover the entire spatiotemporal solution $u(x,t)$, the initial condition $u_0(x)$, the final state $u_T(x)$, or any unknown parameter field, depending on what partial data are collected.

C. Theoretical Results

C.1. Closed-Form Observation Update

Theorem C.1 (Closed-Form Observation Update)**.** *Let $\mathcal{L}_{\text{obs}}(z')$ be a mismatch functional that satisfies $z'(x_j) = y_j$ for each observed data point (x_j, y_j). Suppose we approximate $\mathcal{L}_{\text{obs}}(z')$ by a squared discrepancy from the* ideal *function $\widetilde{y}$, which exactly matches these measurements. Then the solution to*

$$\min_{z'} \Big[\mathcal{L}_{\text{obs}}(z') \;+\; \lambda_{\text{obs}} \,\|\, z_t - z' \|^2 \Big]$$

is given by

$$z_{t+1} \;=\; \frac{\lambda_{\text{obs}}\, z_t \;+\; \widetilde{y}}{\lambda_{\text{obs}} + 1}.$$

Proof. In a Euclidean setting, interpret $\mathcal{L}_{\text{obs}}(z')$ as $\|\widetilde{y} - z'\|^2$, where $\widetilde{y}(x_j) = y_j$ for each observed x_j. The objective becomes

$$\|\widetilde{y} - z'\|^2 \;+\; \lambda_{\text{obs}} \,\|\, z_t - z' \|^2.$$

Denote $z' \in \mathbb{R}^n$ (or a discretized field). Setting the gradient to zero,

$$2(z' - \widetilde{y}) \;+\; 2\,\lambda_{\text{obs}}(z' - z_t) \;=\; 0 \quad\Longrightarrow\quad (1 + \lambda_{\text{obs}})\, z' \;=\; \widetilde{y} \;+\; \lambda_{\text{obs}}\, z_t.$$

Thus

$$z_{t+1} \;=\; z' \;=\; \frac{\widetilde{y} \;+\; \lambda_{\text{obs}}\, z_t}{\lambda_{\text{obs}} + 1}.$$

Hence $\Delta_{\text{obs}} z_t = z_{t+1} - z_t$ as desired. □

C.2. Linear PDE Correction

Theorem C.2 (Closed-Form PDE Update for Linear Systems)**.** *Assume a linear PDE discretization $\mathbf{A}\mathbf{z} = \mathbf{b}$ with constant* b*. Consider*

$$\min_{z'} \Big[\tfrac{1}{2} \,\|\mathbf{A}\, z' - \mathbf{b}\|^2 \;+\; \tfrac{\lambda_{\text{pde}}}{2} \,\|\, z_t - z' \|^2 \Big].$$

Its minimizer z_{t+1} is given by

$$\Delta_{\text{pde}} z_t \;=\; \big(\mathbf{A}^\top \mathbf{A} + \lambda_{\text{pde}}\, \mathbf{I}\big)^{-1} \Big(\mathbf{A}^\top \mathbf{b} + \lambda_{\text{pde}}\, z_t \Big) \;-\; z_t.$$

Proof. In vector form, the objective is

$$\tfrac{1}{2} \,\|\mathbf{A}\, z' - \mathbf{b}\|^2 \;+\; \tfrac{\lambda_{\text{pde}}}{2} \,\|\, z_t - z' \|^2.$$

Taking the gradient w.r.t. z' and setting it to zero yields $\mathbf{A}^\top \mathbf{A}\, z' + \lambda_{\text{pde}}\, z' = \mathbf{A}^\top \mathbf{b} + \lambda_{\text{pde}}\, z_t$. That is, $(\mathbf{A}^\top \mathbf{A} + \lambda_{\text{pde}}\, \mathbf{I})\, z' = \mathbf{A}^\top \mathbf{b} + \lambda_{\text{pde}}\, z_t$. Hence

$$z_{t+1} \;=\; \big(\mathbf{A}^\top \mathbf{A} + \lambda_{\text{pde}}\, \mathbf{I}\big)^{-1} \Big(\mathbf{A}^\top \mathbf{b} + \lambda_{\text{pde}}\, z_t \Big),$$

and $\Delta_{\text{pde}} z_t = z_{t+1} - z_t$. □

C.3. Nonlinear PDE Correction

Theorem C.3 (Newton's Method for Nonlinear PDE Updates)**.** *If $\mathcal{F}_{\text{pde}}(z)$ is not purely quadratic in z (i.e., the PDE is nonlinear), let*

$$R(z_t) \;=\; \nabla_z \mathcal{F}_{\text{pde}}(z_t), \quad J(z_t) \;=\; \nabla_z^2 \mathcal{F}_{\text{pde}}(z_t).$$

Then at each inner iteration, one solves

$$\big[J(z_t) + \lambda_{\text{pde}}\, \mathbf{I} \big]\, \Delta_{\text{pde}} z_t \;=\; -\, R(z_t),$$

and updates $z_{t+1} = z_t + \Delta_{\text{pde}} z_t$. Under typical smoothness and Lipschitz continuity assumptions, this converges to a local minimum of $\mathcal{F}_{\text{pde}}(z) + \frac{\lambda_{\text{pde}}}{2} \,\|\, z_t - z \|^2$.

Proof. Consider the Taylor expansion of $\mathcal{F}_{\mathrm{pde}}(z)$ around z_t:

$$\mathcal{F}_{\mathrm{pde}}(z_t + \delta) \;\approx\; \mathcal{F}_{\mathrm{pde}}(z_t) \;+\; R(z_t)^\top \delta \;+\; \tfrac{1}{2}\,\delta^\top J(z_t)\,\delta.$$

Adding $\frac{\lambda_{\mathrm{pde}}}{2} \,\|\, z_t - (z_t + \delta)\|^2$ gives

$$R(z_t)^\top \delta \;+\; \tfrac{1}{2}\,\delta^\top \big[J(z_t) + \lambda_{\mathrm{pde}}\,\mathbf{I}\big]\delta,$$

and setting the gradient w.r.t. δ to zero yields $(J(z_t) + \lambda_{\mathrm{pde}}\,\mathbf{I})\,\delta = -\,R(z_t)$. Thus $\Delta_{\mathrm{pde}} z_t = \delta$, and by standard Newton convergence theory (assuming smoothness, invertibility, etc.), successive iterations converge to a local minimizer. □

D. Additional Algorithm Details

Explanation: Directly computing $\big(\mathbf{A}^\top\mathbf{A} + \lambda_{\mathrm{pde}}\,\mathbf{I}\big)^{-1}$ can be very time-consuming, especially for high-dimensional problems. One practical workaround is to replace the matrix-inverse update $\Delta_{\mathrm{pde}}\,\mathbf{z}_t$ with a gradient-based step that enforces the PDE on localized neighborhoods only, similar to the original APOD strategy. In that setup, we do not require the explicit inverse but instead apply iterative corrections. A difference is that APOD fully captures the coefficient adjustments from the inverse matrix, providing a more accurate PDE constraint, whereas the gradient-based approach uses a fixed coefficient and avoids large matrix factorizations. Striking the right balance between these two methods (accuracy vs. computational cost) remains a key point for future enhancements of the algorithm.

Algorithm 1 APOD with a Linear Discretization $\mathbf{Az} = \mathbf{b}$

Input: Sampler $D_\theta(\mathbf{z}; \sigma(t))$, schedule $\{\sigma(t)\}_{t=0}^{N}$, observation $\widetilde{y}$, discretization $\mathbf{Az} = \mathbf{b}$ (linear system), weights λ_{obs}, λ_{pde}.
Draw initial sample $\mathbf{z}_0 \sim \mathcal{N}\big(\mathbf{0}, \sigma(t_0)^2\mathbf{I}\big)$
for $t \leftarrow 0$ to $N-1$ **do**
  $\widehat{\mathbf{z}}_N^t \leftarrow D_\theta\big(\mathbf{z}_t;\, \sigma(t)\big)$
  $\mathbf{d}_t \leftarrow \big(\mathbf{z}_t - \widehat{\mathbf{z}}_N^t\big)/\sigma(t)$
  $\mathbf{z}_{t+1} \leftarrow \mathbf{z}_t \;+\; \big(\sigma(t+1) - \sigma(t)\big)\,\mathbf{d}_t$
  if $\sigma(t+1) \neq 0$ **then**
    $\widehat{\mathbf{z}}_N^t \leftarrow D_\theta\big(\mathbf{z}_{t+1};\, \sigma(t+1)\big)$
    $\mathbf{d}_t' \leftarrow \big(\mathbf{z}_{t+1} - \widehat{\mathbf{z}}_N^t\big)/\sigma(t+1)$
    $\mathbf{z}_t \leftarrow \mathbf{z}_t \;+\; \big(\sigma(t+1) - \sigma(t)\big)\,\big(\tfrac{1}{2}\,\mathbf{d}_t + \tfrac{1}{2}\,\mathbf{d}_t'\big)$
  end if
  $\Delta_{\mathrm{obs}}\,\mathbf{z}_t \leftarrow \dfrac{\lambda_{\mathrm{obs}}\big(\widetilde{y} - \mathbf{z}_t\big)}{\lambda_{\mathrm{obs}} + 1},$
  $\Delta_{\mathrm{pde}}\,\mathbf{z}_t \leftarrow \big(\mathbf{A}^\top\mathbf{A} + \lambda_{\mathrm{pde}}\,\mathbf{I}\big)^{-1}\Big(\mathbf{A}^\top\mathbf{b} + \lambda_{\mathrm{pde}}\,\mathbf{z}_t\Big) - \mathbf{z}_t,$
  $\mathbf{z}_{t+1} \leftarrow \mathbf{z}_t + \Delta_{\mathrm{obs}}\,\mathbf{z}_t + \Delta_{\mathrm{pde}}\,\mathbf{z}_t.$
end for
Return: $\mathbf{z}_N$

E. Parameter Settings for Darcy Flow Experiments

We provide the detailed configuration used in all Darcy flow experiments. All models are initialized from the same pre-trained score network, use 100 denoising iterations, and share the same noise schedule and batch settings. The PDE and observation constraints vary depending on the experimental condition.

Table 2. Shared parameter settings across all experiments.

Parameter	**Value**
Datapath	`data/testing/darcy.mat`
Offset	1001
Pre-trained model	`pretrained-models/pretrained-darcy.pkl`
Iterations	100
Batch size	1
Device	`cuda`
Seed	0
$\sigma_{\min}$	0.002
$\sigma_{\max}$	80
ρ	7

The following table outlines the differences in PDE/observation weights and partial observation configurations across all experimental variants.

Table 3. Experiment-specific parameters for different Darcy flow configurations. AOD = Adaptive Observation Diffusion, APOD = Adaptive PDE Observation Diffusion.

Experiment	ζ_{pde}	$\zeta_{\text{obs},a}$	$\zeta_{\text{obs},u}$	$\lambda_{\text{obs},a}$	$\lambda_{\text{obs},u}$	`samples_a/_u`
Forward (DiffusionPDE)	100	5	0	1	1	500 / 0
Backward (DiffusionPDE)	100	0	5000	0	1	0 / 500
Both (DiffusionPDE)	100	5	5000	1	1	250 / 250
Forward (APOD)	0	1	0.002	1	0	500 / 0
Backward (APOD)	100	0	1	0	1	0 / 500
Both (APO)	0	1	1	1	1	250 / 250

Each experiment saves its output to a distinct results path (e.g., `darcy_forward_DiffusionPDE.mat`, `darcy_back_APOD.mat`) as shown in the corresponding YAML files.

F. Discretizing PDEs into the Linear System $\mathbf{Az} = \mathbf{b}$

In this appendix, we outline how the PDEs discussed in Section 4 are discretized into a linear system of the form $\mathbf{Az} = \mathbf{b}$. While each PDE may be solved by finite differences, finite elements, or other schemes, the end result in each case is that unknown field variables (e.g., $\{u(c,\tau),\, v(c,\tau)\}$) are collected into a vector $\mathbf{z}$, and boundary/forcing terms produce a right-hand side $\mathbf{b}$. Below, we sketch the derivation for three representative PDEs.

F.1. Darcy Flow

To illustrate how one obtains the matrices $\mathbf{M}$ and $\mathbf{C}$ when both $u(\cdot)$ and $a(\cdot)$ are unknown, consider the 1D steady Darcy equation

$$-\frac{\mathrm{d}}{\mathrm{d}x}\big(a(x)\,\tfrac{\mathrm{d}u}{\mathrm{d}x}\big) \;=\; q(x), \quad x \in (0,1), \quad u(0) = u(1) = 0.$$

We discretize the interval $[0,1]$ into nodes $\{x_i\}$ with uniform spacing $h = \frac{1}{N}$. Let $u_i \approx u(x_i)$ and $a_i \approx a(x_i)$ for $i = 0, \ldots, N$. For interior nodes $i = 1, \ldots, N-1$, we approximate

$$-\frac{\mathrm{d}}{\mathrm{d}x}\big(a\,\tfrac{\mathrm{d}u}{\mathrm{d}x}\big) \;\approx\; -\frac{1}{h}\Big[\, a_{\,\underline{i+1/2}}\;\tfrac{u_{i+1}-u_i}{h} \;-\; a_{\,\underline{i-1/2}}\;\tfrac{u_i-u_{i-1}}{h}\Big],$$

where $a_{(i+\frac{1}{2})} = \frac{a_i + a_{i+1}}{2}$. This yields, for each i,

$$\underbrace{\left[\frac{a_{(i+1/2)}}{h^2}\,(u_{i+1} - u_i) \;-\; \frac{a_{(i-1/2)}}{h^2}\,(u_i - u_{i-1})\right]}_{\text{depends on both } u_i \text{ and } a_i} \;=\; q_i.$$

We now collect $\mathbf{u} = (u_1, \ldots, u_{N-1})^\top$ and $\mathbf{a} = (a_1, \ldots, a_{N-1})^\top$ (omitting boundary nodes, which are fixed by $u(0) = u(1) = 0$). The unknown vector is

$$\mathbf{z} \;=\; \begin{pmatrix} \mathbf{u} \\ \mathbf{a} \end{pmatrix}.$$

Linearizing each interior equation with respect to $\mathbf{u}$ and $\mathbf{a}$ produces

$$\underbrace{\mathbf{M}}_{\substack{\text{derivative} \\ \text{w.r.t. } \mathbf{u}}}\;\mathbf{u} \;+\; \underbrace{\mathbf{C}}_{\substack{\text{derivative} \\ \text{w.r.t. } \mathbf{a}}}\;\mathbf{a} \;=\; \mathbf{b},$$

where:

- $\mathbf{M}$ is the usual (sparse) stiffness matrix for a 1D Laplacian *but* with coefficients set by the current estimate of $\mathbf{a}$. Concretely, for interior node i,

  $$M_{i,i+1} = -\,\frac{a_{(i+\frac{1}{2})}}{h^2}, \quad M_{i,i-1} = -\,\frac{a_{(i-\frac{1}{2})}}{h^2}, \quad M_{i,i} = \frac{a_{(i+\frac{1}{2})} + a_{(i-\frac{1}{2})}}{h^2}.$$

  All other entries of $\mathbf{M}$ are zero.

- $\mathbf{C}$ encodes the partial derivatives of each nodal equation w.r.t. $\mathbf{a}$. In particular, $\partial(a_{(i+\frac{1}{2})}(u_{i+1} - u_i))/\partial a_k \neq 0$ if $k = i$ or $k = i + 1$, etc. Hence each row i of $\mathbf{C}$ has up to two nonzero entries that reflect how $a_{(i\pm\frac{1}{2})}$ multiplies $(u_{i\pm1} - u_i)$.

- $\mathbf{b}$ collects the source terms q_i and any linearized boundary conditions.

In this way, solving the block system

$$\begin{pmatrix} \mathbf{M} & \mathbf{C} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{a} \end{pmatrix} \;=\; \mathbf{b}$$

delivers updates to both $\mathbf{u}$ and $\mathbf{a}$. Higher-dimensional variants (2D or 3D) or more general meshes follow the same principle, with the matrix entries defined by local finite-difference (or finite-element) stencils for $-\nabla \cdot (a\,\nabla u)$.