POD-FLOW: A POD-DRIVEN FLOW MATCHING FRAMEWORK FOR PHYSICAL FIELD GENERATION ON IRREGULAR GEOMETRIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative models such as diffusion and flow matching offer a promising probabilistic framework for solving PDE-based forward and inverse problems with sparse or noisy observations. However, most existing methods rely on U-Net or Fourier architectures built for regular grids, limiting their use on irregular meshes. We propose POD-Flow, a flow matching framework that uses a Proper Orthogonal Decomposition Neural Operator (PODNO) as its generative backbone. The data-derived POD modes provide a globally supported basis that captures dominant long-range correlations on irregular meshes, avoiding graph message passing or attention. POD-Flow predicts the final solution directly instead of the velocity field, aligning naturally with POD projection and yielding high-signal-to-noise estimates early in inference. Observation-guided sampling then steers generation toward available measurements, enabling a single trained model to perform forward prediction, field reconstruction, and parameter inversion under varying partial observations without retraining. Experiments on multiple PDE benchmarks show that POD-Flow achieves competitive or improved reconstruction accuracy with substantially fewer sampling steps than diffusion baselines, establishing it as an efficient generative framework for probabilistic PDE solving beyond regular grids. The code is available at `https://anonymous.4open.science/r/PodFlow-C6E5`.

1 INTRODUCTION

Partial differential equations (PDEs) govern physical phenomena ranging from fluid dynamics (Morton et al., 2018; Bar-Sinai et al., 2019; Lino et al., 2023) to solid mechanics (Haghighat et al., 2021; Tandale & Stoffel, 2024; Yang et al., 2026). Forward problems predict solution fields u from parameters a, while inverse problems infer a from observations of u. Classical finite-difference and finite element methods (Liu et al., 2022; Guo et al., 2025; Ahmmed, 2023; He et al., 2024) provide accurate solutions but can be computationally expensive. Physics-informed neural networks (Raissi et al., 2019) enforce PDE constraints during training, while neural operators such as FNO (Li et al., 2021) and DeepONet (Lu et al., 2021) learn mappings between function spaces. However, sparse observations and ill-posed inverse problems remain challenging.

Generative models offer a probabilistic approach by learning the joint distribution $p(u, a)$ of parameter and solution fields. DiffusionPDE (Huang et al., 2024) combines a learned generative prior with observation guidance at inference time, enabling forward prediction and inverse reconstruction under partial observations within a unified framework.

An important challenge is extending this approach beyond regular grids. Many generative PDE solvers use U-Net (Ronneberger et al., 2015) or FNO (Li et al., 2021) backbones, which rely on grid-based convolutions or Fourier transforms. Irregular meshes instead require representations that accommodate nonuniform discretizations while capturing long-range physical dependencies. Graph-based methods address irregular connectivity through message passing (Li et al., 2025; Gao et al., 2025; Valencia et al., 2025), whereas neural-field approaches learn compact latent representations for subsequent generation (Du et al., 2024; Liu et al., 2025; Feng et al., 2025). These

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.

developments motivate a complementary approach: using a data-derived global basis that directly represents the dominant structures of discretized physical fields.

We propose POD-Flow, a flow matching framework with a Proper Orthogonal Decomposition Neural Operator (PODNO) backbone (Cheng et al., 2026). Constructed from clean training snapshots, the POD basis captures dominant global correlations without requiring regular-grid connectivity, graph message passing, or attention. This data-derived representation also helps distinguish physical structure from noise: the dominant physical-field energy is concentrated in relatively few modes, whereas noise is spread across a broader range of directions. Retaining these dominant modes therefore reduces the noise burden while preserving the main physical structures. Since truncation can also discard useful details, a parallel branch preserves access to the full hidden representation. POD-Flow combines this architecture with direct clean-endpoint prediction, so that the model learns to recover the physical field itself and provides an explicit estimate for observation guidance. During inference, available measurements constrain this estimate and steer the generation, allowing a single trained model to support forward prediction, field reconstruction, and parameter inversion without retraining.

Our contributions are threefold:

POD-Flow framework. We introduce POD-Flow, which integrates a data-derived POD representation into an endpoint-prediction flow matching framework for physical fields on regular and irregular discretizations.

Energy-based analysis of POD truncation. We provide an energy-based explanation of POD truncation, quantifying its trade-off between physical-field preservation and noise rejection. Energy distributions and intermediate prediction SNR empirically support this mechanism and reveal its diminishing advantage as the input noise vanishes.

Unified prediction, reconstruction, and inversion. We demonstrate unified forward prediction, field reconstruction, and parameter inversion through observation-guided sampling. Across four PDE benchmarks, POD-Flow achieves competitive accuracy on regular grids and generally improves reconstruction on irregular meshes, with fewer parameters and sampling steps than the diffusion-based baselines.

2 Related Work

Neural Operators for PDEs Neural operators learn solution operators of parametric PDEs in a mesh-invariant manner. FNO(Li et al., 2021) is the most influential, parametrizing the integral kernel in the Fourier domain, with extensions using orthogonal polynomial bases for non-periodic boundaries (Liu et al., 2024), trigonometric bases for mixed boundary conditions (Fanaskov & Oseledets, 2023), and Laplace transforms for transient and steady-state responses (Cao et al., 2024); DeepONet(Lu et al., 2021) and its physics-informed variants (Taassob et al., 2024) instead adopt an encoder-decoder structure for greater architectural flexibility. Replacing the Fourier basis with a data-derived POD basis yields PODNO(Cheng et al., 2026), which builds the kernel via POD and enables end-to-end learning in function space; unlike PCA-Net (Bhattacharya et al., 2021) and POD-DeepONet(Eivazi et al., 2024), which use POD as pre- or post-processing, PODNO embeds it directly into kernel integration, capturing dominant energy modes adaptively and providing a natural backbone for generative modeling on irregular geometries.

Generative Models for PDE Solving Generative models offer an alternative paradigm for PDE solving, modeling the joint distribution of parameters and solutions rather than a deterministic mapping. Diffusion models (Huang et al., 2024; Cachay et al., 2023; Bastek et al., 2025; Oommen et al., 2025) generate high-quality samples via iterative denoising and naturally handle ill-posed inverse problems and sparse observations, while flow matching regresses a time-dependent vector field transporting noise to data along a prescribed probability path, enabling deterministic sampling with fewer evaluations (Lipman et al., 2023; Liu et al., 2023; Tong et al., 2024) and matching diffusion quality at faster inference (Esser et al., 2024).

Most existing generative PDE solvers, however, remain confined to regular grids due to CNN- or FNO-based backbones (Shu et al., 2023; Zhuang et al., 2025; Sawhney et al., 2025), and their integration with architectures for irregular geometries, whether graph-based or latent, is only begin-

ning to be explored and still lacks principled solutions, which motivates our work. For conditional guidance, DPS (Chung et al., 2023) incorporates arbitrary observation operators using only forward propagation and gradients; DiffusionPDE (Huang et al., 2024) learns the joint distribution of solutions and coefficients, using physical residual gradients to guide generation from sparse observations. A key challenge is that physical constraints are defined on clean fields while diffusion guidance operates on noisy intermediate states, causing the Jensen gap (Gao et al., 2019), which PIDDM (Tee et al., 2024) addresses via post-training distillation, and FunDPS (Yao et al., 2026) extends Tweedie's formula to infinite-dimensional Banach spaces with theoretical guarantees and resolution independence.

3 PRELIMINARIES

3.1 FORWARD AND INVERSE PROBLEMS FOR PDES

We consider a parametric PDE

$$\mathcal{F}(u, a) = 0 \quad \text{in } \Omega, \tag{1}$$

where a and u denote the parameter and solution fields, respectively. Forward prediction recovers u from known a, while inverse tasks recover complete fields (u, a) from partial observations. With noise-free observations $y_{\text{obs}} = \mathcal{A}(u, a)$, where $\mathcal{A}$ selects observed variables and locations, these tasks are unified as sampling from $p(u, a \mid y_{\text{obs}})$.

3.2 FLOW MATCHING

Flow matching (Lipman et al., 2023; Liu et al., 2023) learns an ODE that transports a prior $p_0 = \mathcal{N}(0, I)$ to the data distribution $p_1 = p(u, a)$:

$$\frac{\mathrm{d}X_t}{\mathrm{d}t} = v_\theta(X_t, t), \qquad X_0 \sim p_0, \quad t \in [0, 1]. \tag{2}$$

For independent samples $X_0 \sim p_0$ and $X_1 \sim p_1$, we use the linear path $X_t = (1-t)X_0 + tX_1$ and train the velocity field through

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t, X_0, X_1} \left\| v_\theta(X_t, t) - (X_1 - X_0) \right\|_2^2, \qquad t \sim \mathcal{U}[0, 1]. \tag{3}$$

Sampling integrates equation 2 from Gaussian noise. Conditional variants use $v_\theta(X_t, t, y)$ and paired data (X_1, y) to learn the conditional distribution $p_1(X_1 \mid y)$.

3.3 PROPER ORTHOGONAL DECOMPOSITION

POD constructs an orthonormal basis from clean snapshots $\{X_1^{(j)}\}_{j=1}^M \subset \mathbb{R}^D$ through the singular value decomposition

$$S = [X_1^{(1)}, \ldots, X_1^{(M)}] = \Phi \Sigma B^\top, \tag{4}$$

where Φ and B contain the left and right singular vectors, respectively. The average modal energies are λ_k, ordered in descending order. Retaining the first K modes yields $\Phi_K = [\phi_1, \ldots, \phi_K]$, with energy ratio $\rho(K) = \sum_{k=1}^K \lambda_k / \sum_{k=1}^r \lambda_k$, where $r = \operatorname{rank}(S)$.

4 METHODOLOGY

4.1 OVERVIEW OF POD-FLOW

POD-Flow combines flow matching with a PODNO backbone to model the joint distribution of parameter and solution fields on regular and irregular discretizations. As illustrated in Figure 1, a truncated POD basis Φ_K is constructed from clean training samples and kept fixed. The backbone learns to predict the clean endpoint $\widehat{X}_1$ from the intermediate state X_t. During inference, endpoint estimates define the flow velocity, while observation guidance steers generation toward the available measurements. The same trained model thus supports forward prediction, field reconstruction, and parameter inversion without retraining.

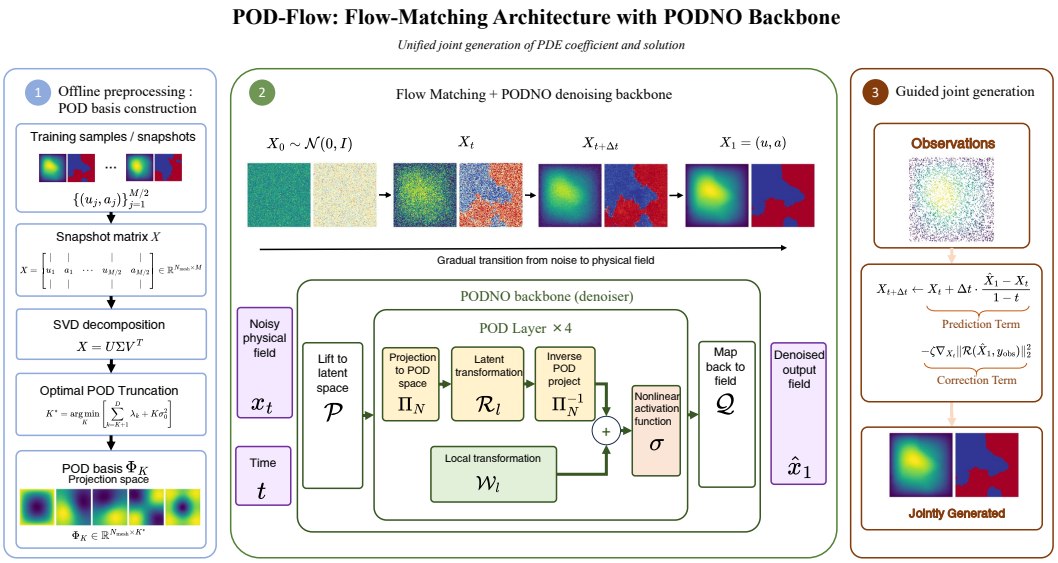


Figure 1: Overall architecture of POD-Flow. The pipeline consists of three stages: (1) POD basis construction; (2) Endpoint prediction training; (3) Observation-guided sampling

4.2 PODNO Backbone and Endpoint Prediction

Standard flow matching learns the velocity $X_1 - X_0$ along the probability path. POD-Flow instead directly predicts the clean endpoint as $\widehat{X}_1 = f_\theta(X_t, t)$ by minimizing

$$\mathcal{L}_{\mathrm{EP}}(\theta) = \mathbb{E}_{t,X_0,X_1}\left[\|f_\theta(X_t,t) - X_1\|_2^2\right]. \tag{5}$$

For the linear path in equation 2, the endpoint estimate defines the sampling velocity

$$v_\theta(X_t,t) = \frac{\widehat{X}_1 - X_t}{1-t}, \qquad t < 1. \tag{6}$$

We instantiate f_θ using PODNO with a fixed truncated POD basis $\Phi_K = \{\phi_k\}_{k=1}^K$, where K is the number of retained modes. The basis defines the POD transform $\Pi_K : \mathbb{R}^D \to \mathbb{R}^K$ and its pseudo-inverse $\Pi_K^{-1} : \mathbb{R}^K \to \mathbb{R}^D$:

$$\Pi_K[V] = (\langle V, \phi_1\rangle, \ldots, \langle V, \phi_K\rangle)^\top, \qquad \Pi_K^{-1}[\bar{V}] = \sum_{k=1}^K \bar{V}_k \phi_k. \tag{7}$$

The PODNO architecture comprises a lifting operator $\mathcal{P}$, a sequence of L POD layers, and a projection operator $\mathcal{Q}$. The lifting operator first maps the input state and time to a hidden representation:

$$V^{(0)} = \mathcal{P}([X_t, t]) \in \mathbb{R}^{D\times d_v}, \tag{8}$$

where D is the number of sample elements, d_v is the hidden feature dimension, and t is broadcast across all sample elements.

Each PODNO layer consists of a POD subspace transformation branch and a detail compensation branch. The first branch projects the hidden representation onto the retained POD modes, applies a learnable transformation in the POD coefficient space, and reconstructs the result in the original sample space:

$$G^{(\ell)} = \Pi_K^{-1}\left[\mathcal{R}_\ell\left(\Pi_K\left(V^{(\ell)}\right)\right)\right], \tag{9}$$

where $\mathcal{R}_\ell$ is a learnable transformation acting on the POD coefficients. Thus, $G^{(\ell)}$ lies in the retained POD subspace.

In parallel, the detail compensation branch applies a pointwise linear operator directly to the complete input $V^{(\ell)}$. The layer update is

$$V^{(\ell+1)} = \sigma\left(G^{(\ell)} + \mathcal{W}_\ell\left(V^{(\ell)}\right)\right), \qquad \ell = 0, \ldots, L-1, \tag{10}$$

where $\mathcal{W}_\ell$ is a pointwise linear mapping and σ is the activation function. The POD branch transforms the dominant structures represented by the retained modes, while the compensation branch provides access to details beyond the truncated representation. The complete layer output is therefore not restricted to the POD subspace.

Finally, a pointwise output mapping $\mathcal{Q}$ reduces the hidden feature dimension d_v to the original number of physical-variable channels:

$$\widehat{X}_1 = \mathcal{Q}\left(V^{(L)}\right) = f_\theta(X_t, t). \tag{11}$$

4.3 Observation-Guided Sampling

POD-Flow learns the unconditional joint distribution of the parameter and solution fields, with observations introduced only during inference. Following observation-guided generative methods (Chung et al., 2023; Huang et al., 2024), we evaluate observation consistency on the predicted clean endpoint $\widehat{X}_1 = f_\theta(X_t, t)$. For noise-free observations y_{obs}, the residual is

$$\mathcal{R}(\widehat{X}_1, y_{\text{obs}}) = \mathcal{A}(\widehat{X}_1) - y_{\text{obs}}, \tag{12}$$

where $\mathcal{A}$ selects the observed physical variables and locations. We augment the learned flow with an observation-guidance term:

$$\frac{\mathrm{d}X_t}{\mathrm{d}t} = \frac{\widehat{X}_1 - X_t}{1-t} - \zeta \nabla_{X_t} \left\| \mathcal{R}(\widehat{X}_1, y_{\text{obs}}) \right\|_2^2, \tag{13}$$

where $\zeta > 0$ controls the guidance strength. The gradient is obtained by backpropagating through the endpoint predictor, steering sampling toward observation consistency. Thus, one trained model supports forward prediction, field reconstruction, and parameter inversion without retraining.

4.4 Why POD-Flow Works: Optimal Signal Preservation and Noise Rejection

Let $X_1 \in \mathbb{R}^D$ be a centered random clean physical field with covariance matrix

$$C = \mathbb{E}[X_1 X_1^\top] = \Phi \Lambda \Phi^\top, \qquad \lambda_1 \geq \cdots \geq \lambda_D \geq 0, \tag{14}$$

where $\Phi = [\phi_1, \ldots, \phi_D]$ is an orthonormal eigenbasis and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_D)$.

For any rank-K orthogonal projector P, the projection onto its retained subspace can be written as

$$PX_1 = \sum_{k=1}^{K} \langle X_1, \psi_k \rangle \, \psi_k, \qquad P = \Psi_K \Psi_K^\top, \tag{15}$$

where $\{\psi_k\}_{k=1}^K$ is an orthonormal basis of the retained subspace. The retained energy is

$$\mathbb{E}\|PX_1\|_2^2 = \mathrm{tr}(PC). \tag{16}$$

The POD projector is obtained by minimizing the expected projection error:

$$\min_{\substack{\psi_1, \ldots, \psi_K \\ \psi_i^\top \psi_j = \delta_{ij}}} \mathbb{E}\left[\|X_1 - PX_1\|_2^2\right], \qquad P = \Psi_K \Psi_K^\top. \tag{17}$$

Its solution is given by the leading eigenvectors of the covariance operator, i.e.,

$$C\phi_k = \lambda_k \phi_k, \qquad \lambda_1 \geq \cdots \geq \lambda_K \geq \cdots \geq 0, \tag{18}$$

and the corresponding POD projector is

$$P_K = \Phi_K \Phi_K^\top, \qquad \Phi_K = [\phi_1, \ldots, \phi_K]. \tag{19}$$

Equivalently, P_K solves the energy-maximization problem

$$P_K = \underset{\substack{P=P^\top=P^2 \\ \mathrm{rank}(P)=K}}{\arg\max}\ \mathrm{tr}(PC), \tag{20}$$

and by orthogonality,

$$\mathbb{E}\|P_K X_1\|_2^2 = \sum_{k=1}^{K} \lambda_k, \qquad \mathbb{E}\|(I-P_K)X_1\|_2^2 = \sum_{k=K+1}^{D} \lambda_k. \tag{21}$$

Thus, minimizing the expected projection error is equivalent to maximizing the expected retained physical-field energy for a fixed K.

We now analyze the effect of POD projection on isotropic Gaussian noise. Let

$$X_0 \sim \mathcal{N}(0, \sigma_0^2 I_D) \tag{22}$$

be independent of X_1. For any rank-K orthogonal projector P, the expected projected noise energy is

$$\mathbb{E}\|PX_0\|_2^2 = \mathrm{tr}\left(P\,\mathbb{E}[X_0 X_0^\top]\right) = \mathrm{tr}\left(P\sigma_0^2 I_D\right) = K\sigma_0^2, \tag{23}$$

where we used $\mathrm{tr}(P) = K$ for a rank-K orthogonal projector. Similarly, the expected removed noise energy is

$$\mathbb{E}\|(I-P)X_0\|_2^2 = (D-K)\sigma_0^2. \tag{24}$$

Thus, for a fixed expected retained signal energy, a smaller K removes more noise in expectation. At the same time, the POD projector P_K is optimal among all rank-K orthogonal projections in the sense that it maximizes the expected retained physical-field energy $\mathrm{tr}(PC)$. Therefore, under POD projection, choosing K is exactly the trade-off between preserving enough expected physical-field energy and removing as much noise as possible.

To quantify this trade-off, we average over $t \sim \mathcal{U}[0,1]$ and define the projection-based denoising error as

$$\begin{aligned} G(K) &= \mathbb{E}\big\|(I-P_K)(tX_1)\big\|_2^2 + \mathbb{E}\big\|P_K((1-t)X_0)\big\|_2^2 \\ &= \mathbb{E}[t^2] \sum_{k=K+1}^{D} \lambda_k + \mathbb{E}[(1-t)^2] K\sigma_0^2 \\ &= \underbrace{\frac{1}{3}\sum_{k=K+1}^{D} \lambda_k}_{\text{discarded signal}} + \underbrace{\frac{1}{3}K\sigma_0^2}_{\text{retained noise}}. \end{aligned} \tag{25}$$

which measures the two undesirable effects of truncation: losing signal energy and keeping noise energy. The optimal number of POD modes is then selected by minimizing this loss,

$$K^* = \underset{K}{\arg\min}\ G(K). \tag{26}$$

5 Experiments

5.1 Experimental Setup

Datasets, tasks, and baselines. We evaluate POD-Flow on four PDE benchmarks. The steady-state Darcy equation and the time-dependent Burgers' equation follow Li et al. (2021). Both support forward prediction and reconstruction from partial observations. Baselines include Diffusion-PDE (Huang et al., 2024) and flow matching variants replacing the PODNO backbone with FNO (Li et al., 2021) or U-Net (Ronneberger et al., 2015), denoted FM-FNO and FM-U-Net.

For irregular meshes, we follow CoNFiLD (Du et al., 2024) to solve the time-dependent incompressible Navier–Stokes equations in a pipe. Using 32-frame windows, we reconstruct velocity and pressure from sparse observations. We also build a 3D geological dataset based on linear elasticity, reconstructing $(d_x, d_y, d_z, \nu, E, \rho)$ from partial displacement observations. CoNFiLD is the baseline for both benchmarks. Further details are in Appendix A.

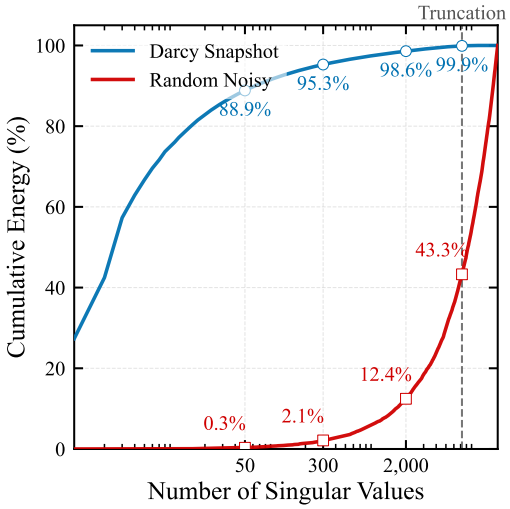


Figure 2: Cumulative POD energy of Darcy fields and Gaussian noise.

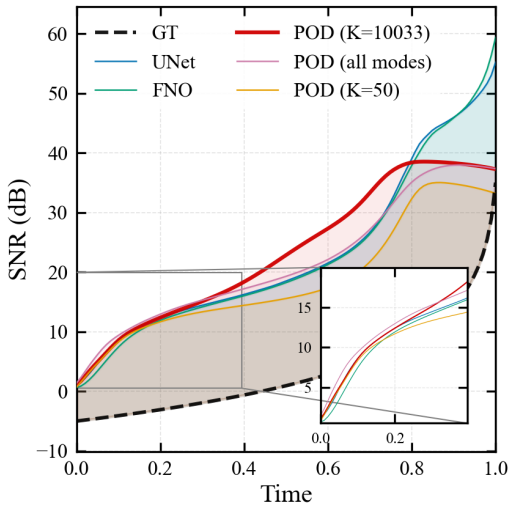


Figure 3: Denoising SNR across backbones and POD truncations on Darcy flow.

Evaluation and implementation. We report relative L_2 errors for continuous fields and the node-wise misclassification rate for the binary Darcy permeability field. Denoising performance is additionally evaluated using SNR at different sampling times. All observations are noise-free. Physical variables are normalized or standardized before training, and the POD basis is constructed exclusively from clean training snapshots and fixed thereafter. Dataset-specific model and training configurations are summarized in Appendix B.

5.2 Results

Reconstruction accuracy and efficiency. Table 1 compares reconstruction errors, parameter counts, and sampling steps across the four benchmarks. For inverse tasks' setting, results are averaged over cases with different numbers of observation points. On regular grids, POD-Flow achieves accuracy comparable to DiffusionPDE, FM-FNO, and FM-U-Net for forward and inverse tasks. On irregular meshes, it generally improves reconstruction over CoNFiLD, although the advantage varies across physical quantities. Together with its substantially smaller trainable parameter count and fewer sampling steps than the diffusion-based baselines, these results demonstrate the effectiveness and efficiency of POD-Flow across different discretizations.

Physical-field and noise energy separation. Figure 2 compares physical-field and Gaussian noise energies across POD modes. Physical-field energy concentrates in leading modes, while noise energy is distributed more evenly. Retaining 99.9% of the training-field energy removes 56.7% of the noise energy, supporting the analysis in Section 4.4: POD truncation preserves dominant physical structures while suppressing substantial noise.

Denoising performance during sampling. Figure 3 compares the SNR of clean-endpoint estimates from different backbones. For v-prediction models, the endpoint is recovered as $\widehat{X}_1 = X_t + (1-t)v_\theta(X_t, t)$, so all methods are evaluated against the same clean target. POD-Flow clearly outperforms U-Net and FNO over $t \in [0.4, 0.8]$, indicating more accurate intermediate endpoint estimates. For $t > 0.8$, U-Net and FNO attain higher SNR: as input noise diminishes, noise rejection matters less, while fine physical details matter more. This intermediate-time advantage helps explain the competitive reconstruction achieved with fewer sampling steps in Table 1.

The same figure also compares different numbers of POD modes: the first 50 modes, the optimal number from equation 25, and the full basis. Aggressive truncation removes more noise but discards physical information, limiting the representation capacity of the subspace transformation branch. The full basis introduces no projection loss and thus no noise rejection, retaining more detail and achieving higher SNR late in sampling at the cost of a larger modal transformation. The selected

Table 1: Performance comparison across four PDE benchmarks. Trainable params are in millions; errors are relative L_2 errors. "For"/"Inv" denote errors of the predicted solution field. "Obs" denotes the number of observations used for inversion. Each entry is averaged over cases. Best and second-best errors are **bold** and underlined.

	Regular meshes									
	Darcy						Burgers			
Method	Params	Steps	For	Inv (5%)	Inv (10%)	Inv (20%)	Params	Steps	Inv (20 Obs)	Inv (30 Obs)
DiffusionPDE	54.41	2000	0.0379	0.0913	0.0553	0.0519	54.41	2000	0.1853	**0.0619**
FM-U-Net			0.0384	**0.0442**	**0.0390**	0.0424			0.1927	0.1185
FM-FNO	134.24	1000	0.1047	0.0816	0.0801	0.0635	134.24	500	0.2180	0.1034
POD-Flow	**41.10**	**500**	**0.0306**	0.0531	0.0399	**0.0417**	**0.80**	**100**	**0.1571**	0.0958

	Irregular meshes: inverse tasks											
		Navier–Stokes					Linear elasticity					
Method	Obs.	Params	Steps	u	v	p	Params	Steps	d	E	ν	ρ
CoNFiLD	30	92.17	1000	0.5563	0.3983	0.7210	44.02	1000	0.2053	**0.2247**	0.0712	0.0555
	45			0.5055	0.3772	**0.6593**			0.1503	**0.2228**	0.0706	0.0624
	90			0.4813	0.3487	**0.6090**			0.1315	**0.2107**	0.0922	0.0492
	200			0.4453	0.3093	0.6192			**0.0639**	**0.1876**	0.0584	**0.0397**
	368			0.4221	0.2901	0.6034			0.1401	**0.2006**	0.0570	0.0366
POD-Flow	30	**60.27**	**100**	**0.3275**	**0.2029**	**0.6948**	**4.52**	**100**	**0.1390**	0.2363	**0.0642**	**0.0544**
	45			**0.2913**	**0.1959**	0.6609			**0.1174**	0.2268	**0.0608**	**0.0529**
	90			**0.2778**	**0.1726**	0.6324			**0.0988**	0.2165	**0.0551**	**0.0479**
	200			**0.2159**	**0.1309**	**0.5572**			0.0931	0.2102	**0.0470**	0.0404
	368			**0.2090**	**0.1311**	**0.5655**			**0.0928**	0.2054	**0.0316**	**0.0251**

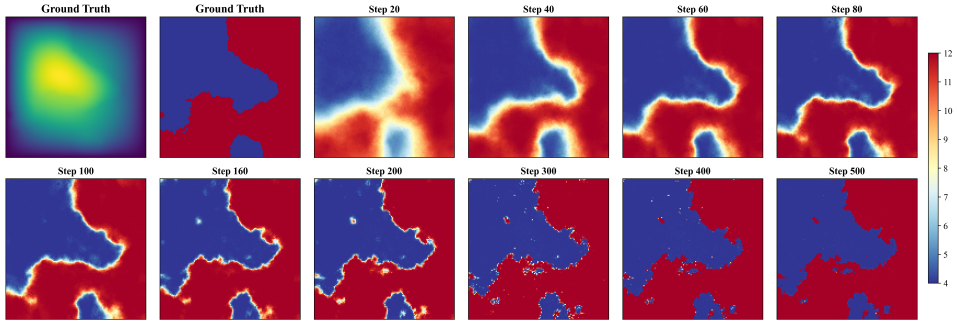


Figure 4: Intermediate predictions during the denoising process.

mode count achieves higher SNR at intermediate flow times while matching the full basis near the endpoint, balancing noise suppression and physical-detail preservation. Figure 4 further visualizes intermediate endpoint predictions during observation-guided sampling: POD-Flow recovers the main field contours early, then progressively refines local details and boundaries.

5.3 Ablation Studies

Effect of the Number of POD Modes We evaluate models with different POD modes on the Darcy benchmark under the same partial-observation conditioning. The reconstruction results in Figure 5 show that with too few modes, the model struggles to recover the permeability structure and its fine details. Reconstructions improve substantially with 500 or more modes, while further

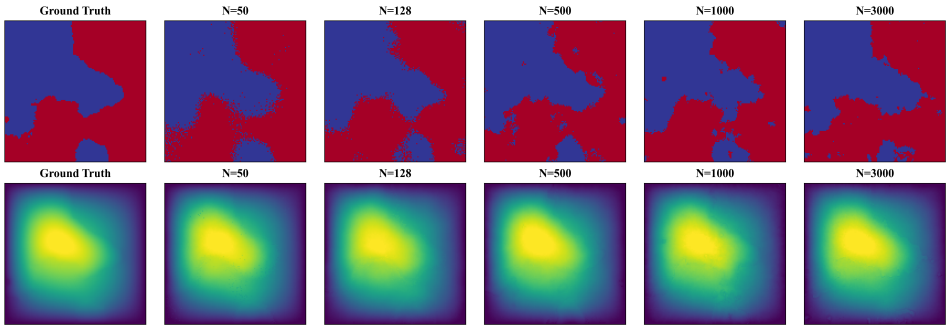


Figure 5: Darcy reconstructions from models trained with different numbers of retained POD modes.

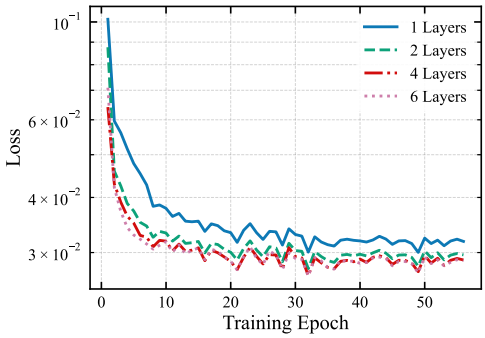


Figure 6: Training loss convergence with different numbers of POD layers.

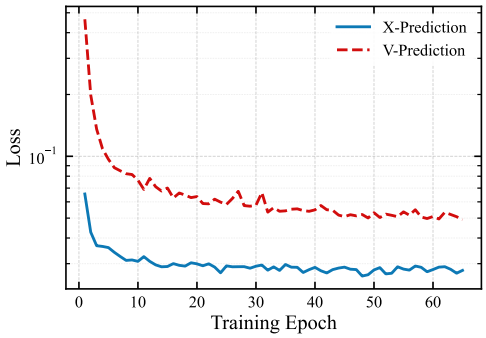


Figure 7: Comparison of training convergence between x-prediction and v-prediction.

increases yield limited visual improvements. These results highlight the need to retain sufficient physical information while avoiding an unnecessarily large modal representation.

Effect of the Number of POD Layers Comparing $L \in \{1, 2, 4, 6\}$ PODNO layers, Figure 6 shows that one layer converges to a relatively high loss, two and four layers progressively reduce it, and six layers bring little further improvement. Four layers thus offer a practical balance between fit and complexity.

Effect of Prediction Target Using the same PODNO backbone on Darcy, Figure 7 shows that x-prediction reaches a stable loss earlier (absolute values are not directly comparable across objectives), and it reduces the forward reconstruction error from 3.19% to 1.80% and the inverse error from 5.67% to 4.49%. Further results are in Appendix C.5.

6 Conclusion

We presented POD-Flow, a generative PDE framework combining a PODNO backbone, clean-endpoint prediction, and observation-guided sampling. Experiments demonstrate competitive accuracy on regular grids and improved reconstruction on irregular meshes, with fewer parameters and sampling steps than diffusion-based baselines. Energy analysis and denoising experiments support the role of POD truncation in preserving dominant physical structures while suppressing noise.

Several limitations motivate future work. (1) The fixed POD basis depends on the training data and discretization, motivating adaptive representations for transfer across geometries. (2) The denoising benefit diminishes late in inference, calling for better fine-detail recovery. (3) Treating each time instant as a separate feature channel restricts the model to fixed-length windows and hinders long-horizon prediction. Explicit temporal modeling is thus an important next step toward generative PDE solving over longer horizons.

AI USE STATEMENT

We used large language models (LLMs) to refine the writing and presentation of this paper, assist in discussions of theoretical derivations and experimental interpretations, and generate code for data processing and visualization. The authors retain responsibility for the problem formulation, method design, implementation, and validation, and take full responsibility for all technical claims and the final content of this work.

ETHICS STATEMENT

This work focuses on methods for solving scientific inverse problems governed by PDEs. The primary applications we target are in physics, engineering, and biomedical domains. Beyond the ethical concerns common to scientific machine learning, such as the potential misuse of improved modeling techniques in sensitive domains (e.g., defense applications). We are not aware of any direct ethical issues. We emphasize that our benchmarks are publicly available, non-sensitive, and purely synthetic or simulation-based.

REPRODUCIBILITY STATEMENT

Every dataset employed in this study is publicly accessible and appropriately cited. Comprehensive details regarding the benchmarks, model architectures, training procedures, and hyperparameters can be found in the Appendix. The code has also been submitted to the repository.

AUTHOR CONTRIBUTIONS

This section has been omitted for anonymous review and will be included in the camera-ready version.

ACKNOWLEDGMENTS

This section has been omitted for anonymous review and will be included in the camera-ready version.

REFERENCES

Mohammad Shakil Ahmmed. A finite volume numerical approach for predicting heat transfer in presence of thermal contact resistance. *Thermal Science and Engineering Progress*, 37:101618, 2023.

Yohai Bar-Sinai, Stephan Hoyer, Jason Hickey, and Michael P. Brenner. Learning data-driven discretizations for partial differential equations. *Proceedings of the National Academy of Sciences*, 116(31):15344–15349, 2019.

Jan-Hendrik Bastek, WaiChing Sun, and Dennis Kochmann. Physics-informed diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Kaushik Bhattacharya, Bamdad Hosseini, Nikola B. Kovachki, and Andrew M. Stuart. Model Reduction And Neural Networks For Parametric PDEs. *The SMAI Journal of computational mathematics*, 7:121–157, 2021.

Salva Rühling Cachay, Bo Zhao, Hailey Joren, and Rose Yu. DYffusion: A dynamics-informed diffusion model for spatiotemporal forecasting. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Qianying Cao, Somdatta Goswami, and George Em Karniadakis. Laplace neural operator for solving differential equations. *Nature Machine Intelligence*, 6(6):631–640, 06 2024.

Zilan Cheng, Zhongjian Wang, Li-Lian Wang, and Mejdi Azaiez. Podno: Proper orthogonal decomposition neural operators. *SIAM Journal on Scientific Computing*, 48(3):C479–C504, 2026.

Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023.

Pan Du, Meet Hemant Parikh, Xiantao Fan, Xin-Yang Liu, and Jian-Xun Wang. Conditional neural field latent diffusion model for generating spatiotemporal turbulence. *Nature Communications*, 15(1):10416, 2024.

Hamidreza Eivazi, Stefan Wittek, and Andreas Rausch. Nonlinear model reduction for operator learning. In *The Second Tiny Papers Track at ICLR 2024*, 2024.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

V. S. Fanaskov and I. V. Oseledets. Spectral neural operators. *Doklady Mathematics*, 108(2):S226–S232, 12 2023.

Zhao Feng, Xin-Yang Liu, Meet Hemant Parikh, Junyi Guo, Pan Du, Bicheng Yan, and Jian-Xun Wang. Generative latent diffusion model for inverse modeling and uncertainty analysis in geological carbon sequestration. *arXiv preprint arXiv:2508.16640*, 2025.

Han Gao, Sebastian Kaltenbach, and Petros Koumoutsakos. Generative learning of the solution of parametric partial differential equations using guided diffusion models and virtual observations. *Computer Methods in Applied Mechanics and Engineering*, 435:117654, 2025.

Xiang Gao, Meera Sitharam, and Adrian E. Roitberg. Bounds on the Jensen gap, and implications for mean-concentrated distributions. *The Australian Journal of Mathematical Analysis and Applications*, 16(2):1–18, 2019.

Hao Guo, Boxing Hu, Peixue Jiang, Xiaofeng Ma, and Yinhai Zhu. Order-lifted data inversion/retrieval method of neighbor cells to implement general high-order schemes in unstructured-mesh-based finite-volume solution framework. *Journal of Computational Physics*, 524:113735, 2025.

Ehsan Haghighat, Maziar Raissi, Adrian Moure, Hector Gomez, and Ruben Juanes. A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 379:113741, 2021.

Lishen He, Albert J. Valocchi, and C.A. Duarte. An adaptive global–local generalized fem for multiscale advection–diffusion problems. *Computer Methods in Applied Mechanics and Engineering*, 418:116548, 2024.

Jiahe Huang, Guandao Yang, Zichen Wang, and Jeong Joon Park. DiffusionPDE: Generative PDE-solving under partial observation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Zijie Li, Anthony Zhou, and Amir Barati Farimani. Generative latent neural pde solver using flow matching. *arXiv preprint arXiv:2503.22600*, 2025.

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

Mario Lino, Stathi Fotiadis, Anil A. Bharath, and Chris D. Cantwell. Current and emerging deep-learning methods for the simulation of fluid dynamics. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 479(2275):20230058, 2023.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Wing Kam Liu, Shaofan Li, and Harold S. Park. Eighty years of the finite element method: Birth, evolution, and future. *Archives of Computational Methods in Engineering*, 29(6):4431–4453, 10 2022.

Xin-Yang Liu, Meet Hemant Parikh, Xiantao Fan, Pan Du, Qing Wang, Yi-Fan Chen, and Jian-Xun Wang. Confild-inlet: Synthetic turbulence inflow using generative latent diffusion models with neural fields. *Physical Review Fluids*, 10:054901, 2025.

Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023.

Ziyuan Liu, Haifeng Wang, Hong Zhang, Kaijun Bao, Xu Qian, and Songhe Song. Render unto numerics: Orthogonal polynomial neural operator for pdes with nonperiodic boundary conditions. *SIAM Journal on Scientific Computing*, 46(4):C323–C348, 2024.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021.

Jeremy Morton, Freddie D. Witherden, Antony Jameson, and Mykel J. Kochenderfer. Deep dynamical modeling and control of unsteady fluid flows. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 9278–9288, Red Hook, NY, USA, 2018. Curran Associates Inc.

Vivek Oommen, Aniruddha Bora, Zhen Zhang, and George Em Karniadakis. Integrating neural operators with diffusion models improves spectral representation in turbulence modelling. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 481(2309): 20240819, 03 2025.

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.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, 2015.

Medha Sawhney, Abhilash Neog, Mridul Khurana, and Anuj Karpatne. Beyond loss guidance: Using pde residuals as spectral attention in diffusion neural operators. *arXiv preprint arXiv:2512.01370*, 2025.

Dule Shu, Zijie Li, and Amir Barati Farimani. A physics-informed diffusion model for high-fidelity flow field reconstruction. *Journal of Computational Physics*, 478:111972, 2023.

Arsalan Taassob, Anuj Kumar, Kevin M. Gitushi, Rishikesh Ranade, and Tarek Echekki. A pinn-deeponet framework for extracting turbulent combustion closure from multiscalar measurements. *Computer Methods in Applied Mechanics and Engineering*, 429:117163, 2024.

Saurabh Balkrishna Tandale and Marcus Stoffel. Physics-based self-learning spiking neural network enhanced time-integration scheme for computing viscoplastic structural finite element response. *Computer Methods in Applied Mechanics and Engineering*, 422:116847, 2024.

Joshua Tian Jin Tee, Kang Zhang, Hee Suk Yoon, Dhananjaya Nagaraja Gowda, Chanwoo Kim, and Chang D. Yoo. Physics informed distillation for diffusion models. *arXiv preprint arXiv:2411.08378*, 2024.

Alexander Tong, Kilian FATRAS, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024.

Mario Lino Valencia, Tobias Pfaff, and Nils Thuerey. Learning distributions of complex fluid simulations with diffusion graph networks. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yanyan Yang, Lili Wang, Xiaoya Zhai, Kai Chen, Wenming Wu, Yunkai Zhao, Falai Chen, Ligang Liu, and Xiao-Ming Fu. Guided diffusion for fast inverse design of voxel-based mechanical metamaterials. *Smart Materials in Manufacturing*, 4:100129, 2026.

Jiachen Yao, Abbas Mammadov, Julius Berner, Gavin Kerrigan, Jong Chul Ye, Kamyar Azizzadenesheli, and Anima Anandkumar. Guided diffusion sampling on function spaces with applications to PDEs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

Yilin Zhuang, Sibo Cheng, and Karthik Duraisamy. Spatially-aware diffusion models with cross-attention for global field reconstruction with sparse observations. *Computer Methods in Applied Mechanics and Engineering*, 435:117623, 2025.

A Dataset Details and Problem Configurations

A.1 Darcy Flow

We consider steady-state Darcy flow with homogeneous Dirichlet boundary conditions:

$$-\nabla \cdot (a(c)\nabla u(c)) = 1, \quad c \in \Omega, \qquad u(c) = 0, \quad c \in \partial\Omega, \tag{27}$$

where a is the permeability field and u is the pressure head. Following the benchmark configuration in (Li et al., 2021), permeability fields are generated by thresholding a Gaussian random field:

$$g \sim \mathcal{N}\left(0, (-\Delta + 9I)^{-2}\right), \qquad a(c) = \begin{cases} 12, & g(c) \geq 0, \\ 3, & g(c) < 0. \end{cases} \tag{28}$$

The dataset contains 50,000 samples, each comprising permeability and pressure fields on a 128×128 regular grid. Forward prediction recovers the pressure field from the known permeability, while inverse reconstruction recovers the complete fields from sparse pressure observations.

A.2 Burgers' Equation

We consider the one-dimensional viscous Burgers' equation with periodic boundary conditions (Li et al., 2021):

$$\frac{\partial u}{\partial \tau} + \frac{\partial}{\partial c}\left(\frac{u^2}{2}\right) = \nu \frac{\partial^2 u}{\partial c^2}, \qquad c \in (0,1), \quad \tau \in (0,T], \tag{29}$$

where $\nu = 0.01$, $T = 1$, and τ denotes physical time. The initial condition is sampled as

$$u_0 \sim \mathcal{N}\left(0, 625(-\Delta + 25I)^{-2}\right). \tag{30}$$

A spectral solver generates 1,000 solution trajectories, each discretized at 128 spatial points and 128 time instants, including the initial state. Each trajectory is treated as a two-dimensional spatiotemporal field, with space and time forming its two axes. We evaluate sparse reconstruction of the complete field from partial observations.

A.3 Incompressible Pipe Flow

We consider the two-dimensional pipe-flow benchmark with stochastic forcing used in CoNFiLD (Du et al., 2024). The governing incompressible Navier–Stokes equations are

$$\frac{\partial \boldsymbol{u}}{\partial \tau} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\nabla p + \nu \Delta \boldsymbol{u} + \boldsymbol{f}, \qquad \nabla \cdot \boldsymbol{u} = 0, \tag{31}$$

where $\boldsymbol{u} = (u, v)^\top$ is the velocity, p is the pressure, and $\boldsymbol{f}$ is a stochastic forcing field satisfying

$$\frac{\partial \boldsymbol{f}}{\partial \tau} = \nu_f \Delta \boldsymbol{f} + \boldsymbol{\delta}, \qquad \nu_f = 2. \tag{32}$$

Each component of the stochastic source $\boldsymbol{\delta}$ is sampled from $\mathcal{N}(0,1)$.

The data are generated by direct numerical simulation on an irregular mesh with 918 nodes. We extract 32-frame windows from the trajectories, with each window containing both velocity components and pressure. The task is to reconstruct these three fields throughout the window from sparse observations at a subset of mesh nodes.

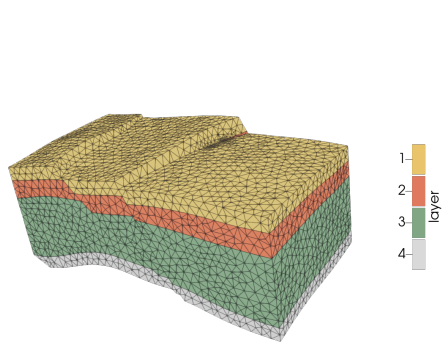
A.4 Geological Linear Elasticity

We construct a three-dimensional geological dataset governed by quasi-static linear elasticity:

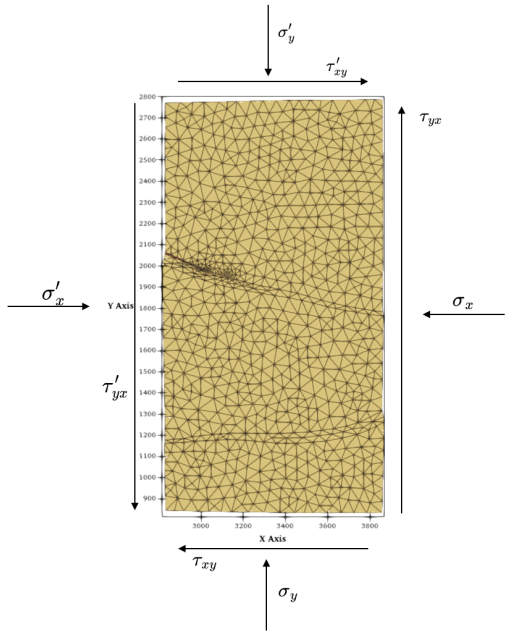
$$-\nabla \cdot \boldsymbol{\sigma}(c) = \rho(c)\boldsymbol{g}, \qquad c \in \Omega, \tag{33}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor, ρ is the density, and $\boldsymbol{g}$ is gravitational acceleration. For the displacement field $\boldsymbol{d} = (d_x, d_y, d_z)^\top$, the constitutive relations are

$$\boldsymbol{\sigma} = \lambda_{\mathrm{L}} \operatorname{tr}(\boldsymbol{\varepsilon}) I + 2\mu\boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} = \frac{1}{2}\left(\nabla \boldsymbol{d} + (\nabla \boldsymbol{d})^\top\right), \tag{34}$$



(a) Geometry and material layers



(b) Mechanical boundary conditions

Figure 8: Geological model geometry, material layers, and applied boundary conditions.

Table 2: Material-field statistics for the four geological layers. Values are reported as mean $\pm$ standard deviation.

Layer	E (MPa)	ν	Bounds of ν	ρ (kg/m^3)
1	15 ± 4.5	0.40 ± 0.020	$[0.30, 0.49]$	1700 ± 80
2	40 ± 10	0.33 ± 0.020	$[0.22, 0.45]$	1850 ± 80
3	100 ± 20	0.28 ± 0.015	$[0.18, 0.40]$	2050 ± 100
4	200 ± 40	0.25 ± 0.015	$[0.15, 0.38]$	2250 ± 110

with Lamé parameters

$$\lambda_{\mathrm{L}} = \frac{E\nu}{(1+\nu)(1-2\nu)}, \qquad \mu = \frac{E}{2(1+\nu)}, \tag{35}$$

where E and ν denote Young's modulus and Poisson's ratio.

The domain comprises four heterogeneous geological layers, as illustrated in Figure 8. Young's modulus and density are modeled as lognormal random fields, while Poisson's ratio follows a bounded Beta distribution. The fields are generated independently across properties and layers using a Gaussian covariance model with correlation lengths $(400, 400, 50)$ m. The layer-specific statistics are provided in Table 2.

The bottom boundary is fixed, with $\boldsymbol{d} = 0$ on Γ_D. Normal and shear stresses are applied to the four lateral boundaries, with eight loading components sampled independently within $[-10^7, 10^7]$ Pa. Gravity acts throughout the domain, and no contact interaction is included.

The dataset contains 5,000 samples on an irregular mesh with 13,481 nodes. Each sample includes six nodal quantities: $(d_x, d_y, d_z, \nu, E, \rho)$. Given sparse observations of the three displacement components, the model reconstructs the complete displacement and material-property fields.

A.5 Data Preprocessing and Dataset Splits

Each dataset is divided into training, validation, and test sets with an 8:1:1 ratio. For pipe flow, the original time-series data are partitioned before extracting 32-frame sliding windows within each

Table 3: Architecture configurations and initial learning rates for POD-Flow and baseline models.

Dataset	Model	Architecture configuration	Initial LR
Darcy	FM-FNO	$(d_v, L, K_x, K_y) = (64, 4, 64, 64)$	10^{-3}
	FM-U-Net	$(C, \boldsymbol{m}, R) = (128, [2, 2, 2], 4)$	10^{-4}
	DiffusionPDE	$(C, \boldsymbol{m}, R) = (128, [2, 2, 2], 4)$	10^{-4}
	POD-Flow	$(d_v, L, K) = (32, 4, 10033)$	10^{-3}
Burgers	FM-FNO	$(d_v, L, K_x, K_\tau) = (64, 4, 64, 64)$	10^{-3}
	FM-U-Net	$(C, \boldsymbol{m}, R) = (128, [2, 2, 2], 4)$	10^{-4}
	DiffusionPDE	$(C, \boldsymbol{m}, R) = (128, [2, 2, 2], 4)$	10^{-4}
	POD-Flow	$(d_v, L, K) = (32, 4, 193)$	10^{-3}
Pipe	CoNFiLD (CNF)	$(H, F) = (30, 128)$	10^{-4}
	CoNFiLD (U-Net)	$(C, \boldsymbol{m}, R) = (32, [1, 1, 2, 3, 4], 2)$	5×10^{-5}
	POD-Flow	$(d_v, L, K) = (128, 4, 916)$	10^{-3}
Geological elasticity	CoNFiLD (CNF)	$(H, F) = (30, 128)$	10^{-4}
	CoNFiLD (U-Net)	$(C, \boldsymbol{m}, R) = (1, [1, 1, 2, 3, 4], 2)$	5×10^{-5}
	POD-Flow	$(d_v, L, K) = (32, 4, 1101)$	10^{-3}

For PODNO and FNO, d_v and L denote the hidden width and number of operator layers. K denotes the retained POD modes, while K_x, K_y, and K_τ denote Fourier modes along the corresponding dimensions. For U-Net, C is the base channel width, $\boldsymbol{m}$ contains the channel multipliers, and R is the number of residual blocks per resolution. For the conditional neural field (CNF), H and F denote the number of hidden layers and features per layer.

subset, ensuring that no frames are shared across subsets. Physical variables are normalized or standardized before training. The POD basis is constructed exclusively from clean training samples and remains fixed during training and inference. All reconstruction tasks use noise-free observations.

B Implementation Details

B.1 Training Configurations

All models are implemented in PyTorch 2.6.0 and optimized using Adam. POD-Flow and FM-FNO are trained on two NVIDIA GeForce RTX 4090 GPUs with 24 GB of memory each, while U-Net-based models use two NVIDIA RTX PRO 6000 GPUs due to their higher memory requirements. Architecture configurations and initial learning rates are summarized in Table 3.

B.2 Evaluation Metrics and SNR Computation

For continuous physical quantities, reconstruction accuracy is measured by the relative L_2 error:

$$e_{\mathrm{rel}} = \frac{\|\widehat{q} - q\|_2}{\|q\|_2}, \tag{36}$$

where q and $\widehat{q}$ denote the reference and reconstructed fields. Errors are computed separately for each physical variable and averaged over evaluation samples and, where applicable, physical time frames.

For binary Darcy permeability, we report the proportion of incorrectly classified nodes:

$$e_{\mathrm{binary}} = \frac{1}{N_{\mathrm{mesh}}} \sum_{i=1}^{N_{\mathrm{mesh}}} \mathbb{I}[\widehat{a}_{\mathrm{cls}}(c_i) \neq a(c_i)], \tag{37}$$

Denoising performance is evaluated using the SNR of clean-endpoint estimates:

$$\mathrm{SNR}(t) = 10 \log_{10} \frac{\|X_1\|_2^2}{\|\widehat{X}_1(X_t, t) - X_1\|_2^2}. \tag{38}$$

C RECONSTRUCTION RESULTS

C.1 FORWARD PREDICTION ON REGULAR GRIDS

Figures 9 present representative forward predictions for the Darcy benchmarks. For Darcy flow, the relative performance varies across samples, with POD-Flow achieving reconstruction quality comparable to the regular-grid baselines.

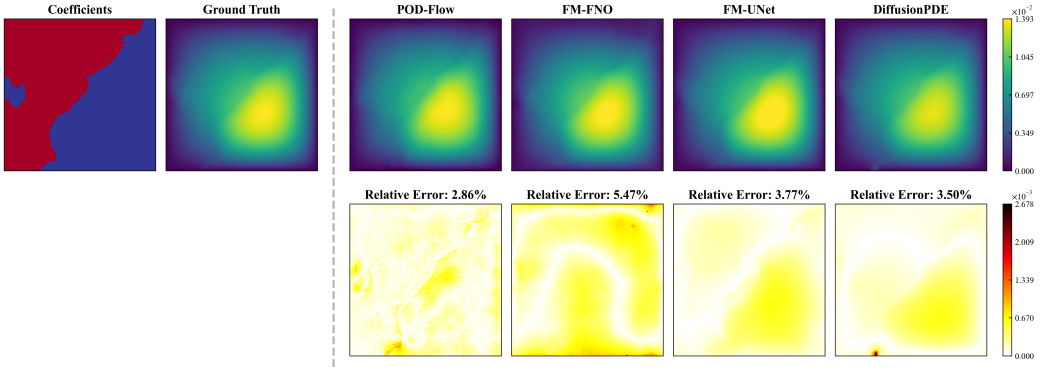


Figure 9: Darcy pressure predictions from complete permeability fields.

C.2 Inverse Reconstruction on Regular Grids

Figure 10 compares permeability reconstructions using pressure observations at 5%, 10%, and 20% of the grid points. POD-Flow recovers the main connectivity and internal structures of the permeability fields across these observation settings. Figure 11 presents spatiotemporal reconstruction from observations at 20 and 30 of the grid points. The reconstructed fields capture the principal solution dynamics despite the sparse spatial observations.

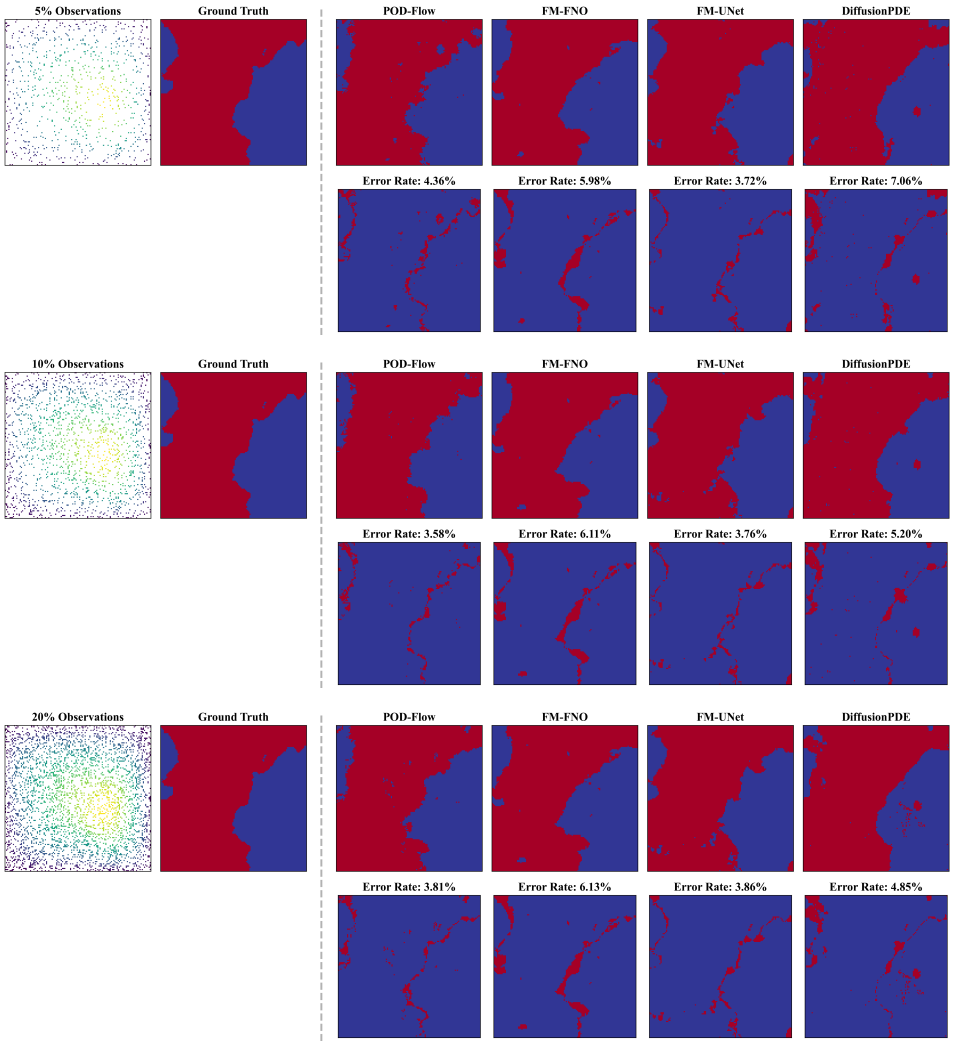


Figure 10: Darcy permeability reconstructions from sparse pressure observations. Observation ratios are 5%, 10%, and 20% from top to bottom.

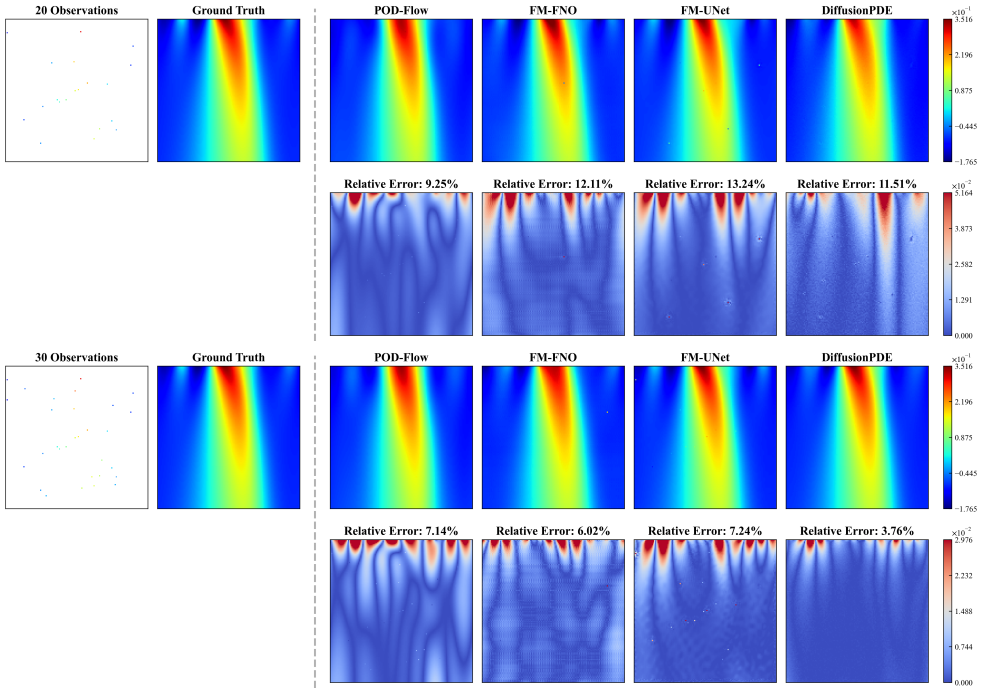


Figure 11: Burgers' equation inverse reconstruction: complete spatiotemporal fields recovered from 20 and 30 observations.

C.3 Pipe Flow Reconstruction

Figures 12–14 compare POD-Flow and CoNFiLD (Du et al., 2024) using observations at 200 of the 918 mesh nodes. The figures show the reference fields, reconstructions, and pointwise relative errors at selected physical time instants. In these examples, POD-Flow more closely reproduces the velocity and pressure distributions, with smaller and more localized reconstruction errors.

Figure 15 further compares the relative L_2 errors of POD-Flow and CoNFiLD for the two velocity components and pressure using 30, 45, 90, 200, and 368 observation nodes. This comparison evaluates how reconstruction accuracy changes with observation density for each physical quantity.

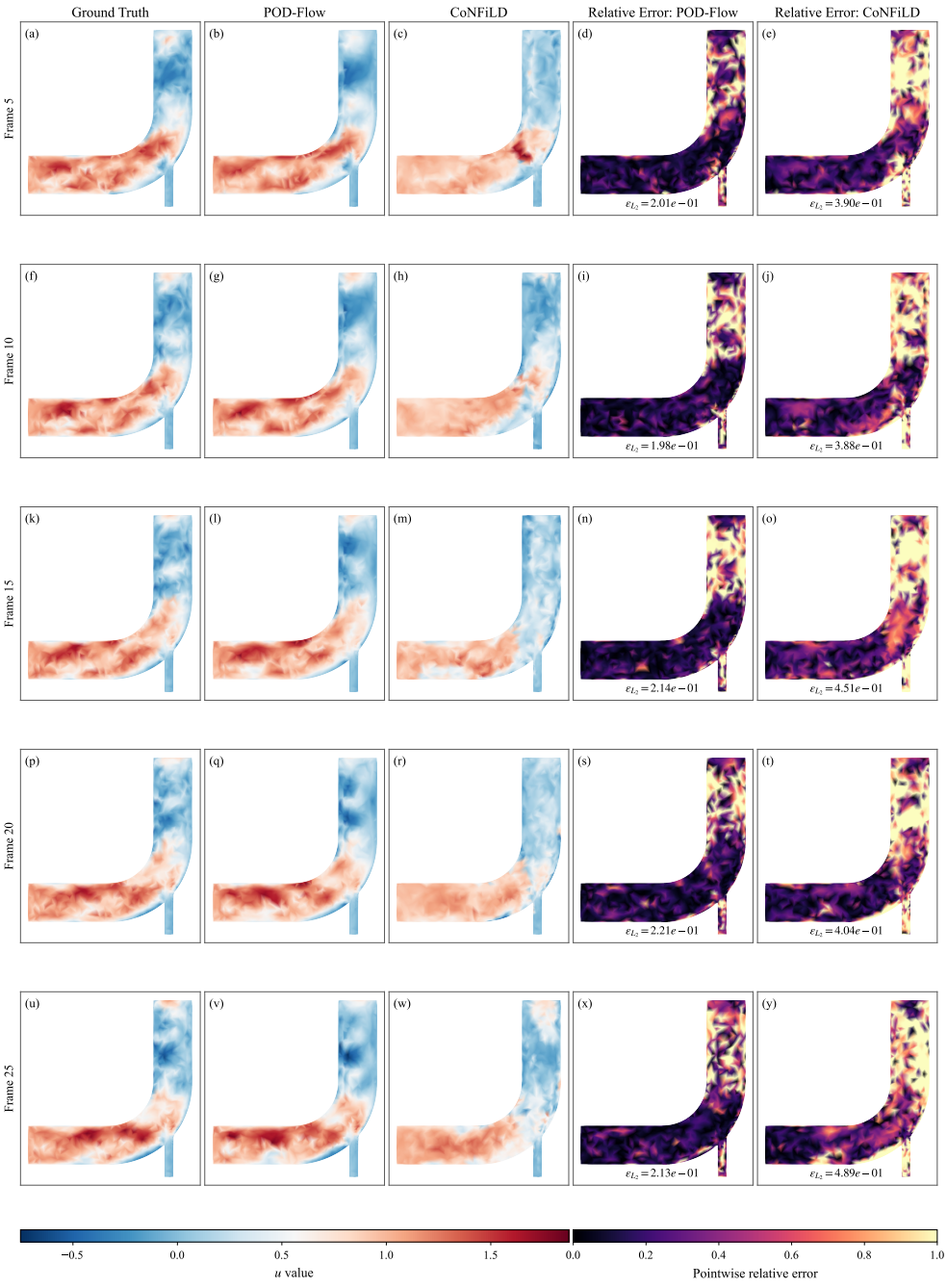


Figure 12: Pipe-flow u-velocity reconstructions by POD-Flow and CoNFiLD at selected physical time instants, using 368 observation nodes.

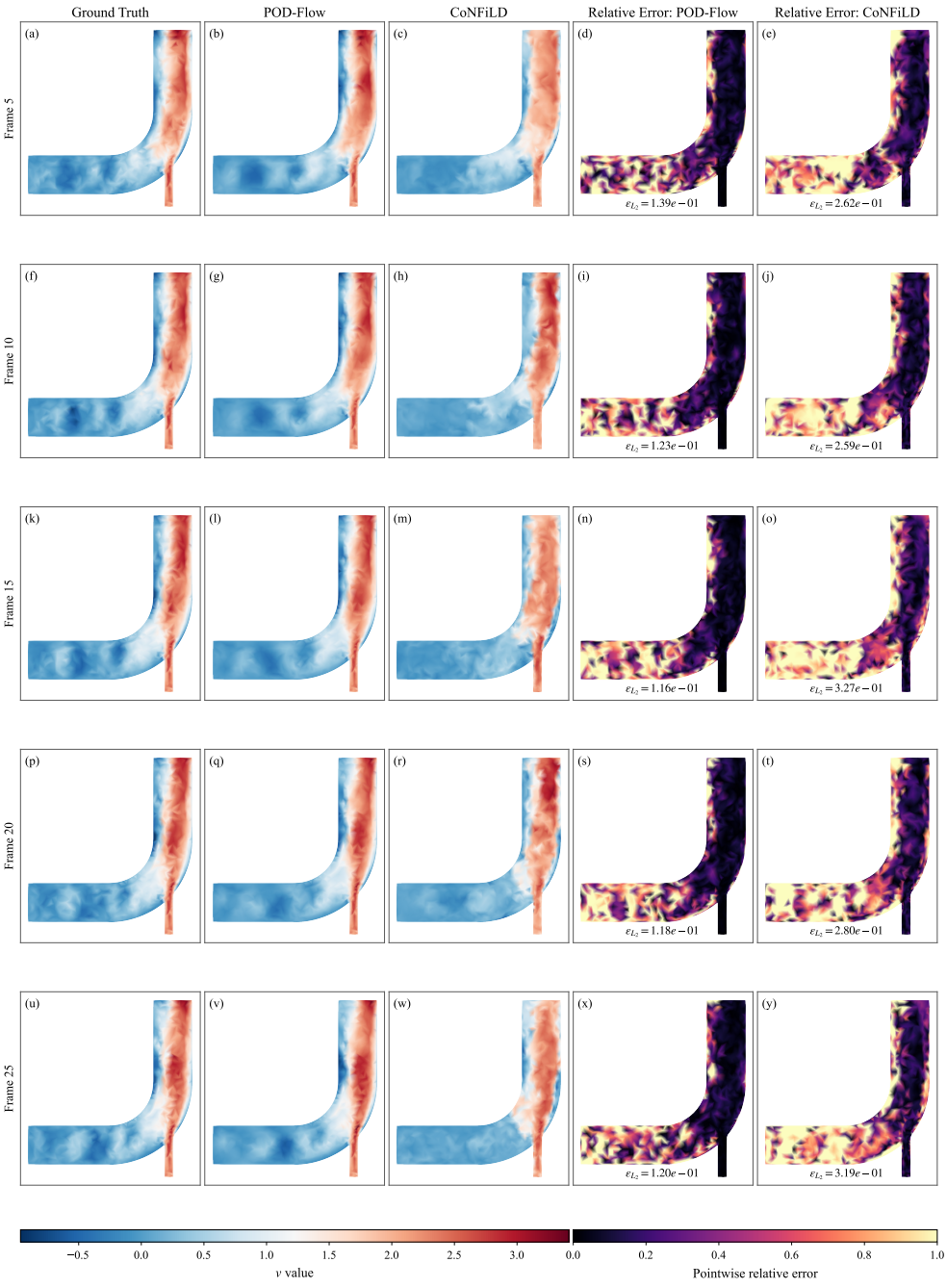


Figure 13: Pipe-flow v-velocity reconstructions by POD-Flow and CoNFiLD at selected physical time instants, using 368 observation nodes.

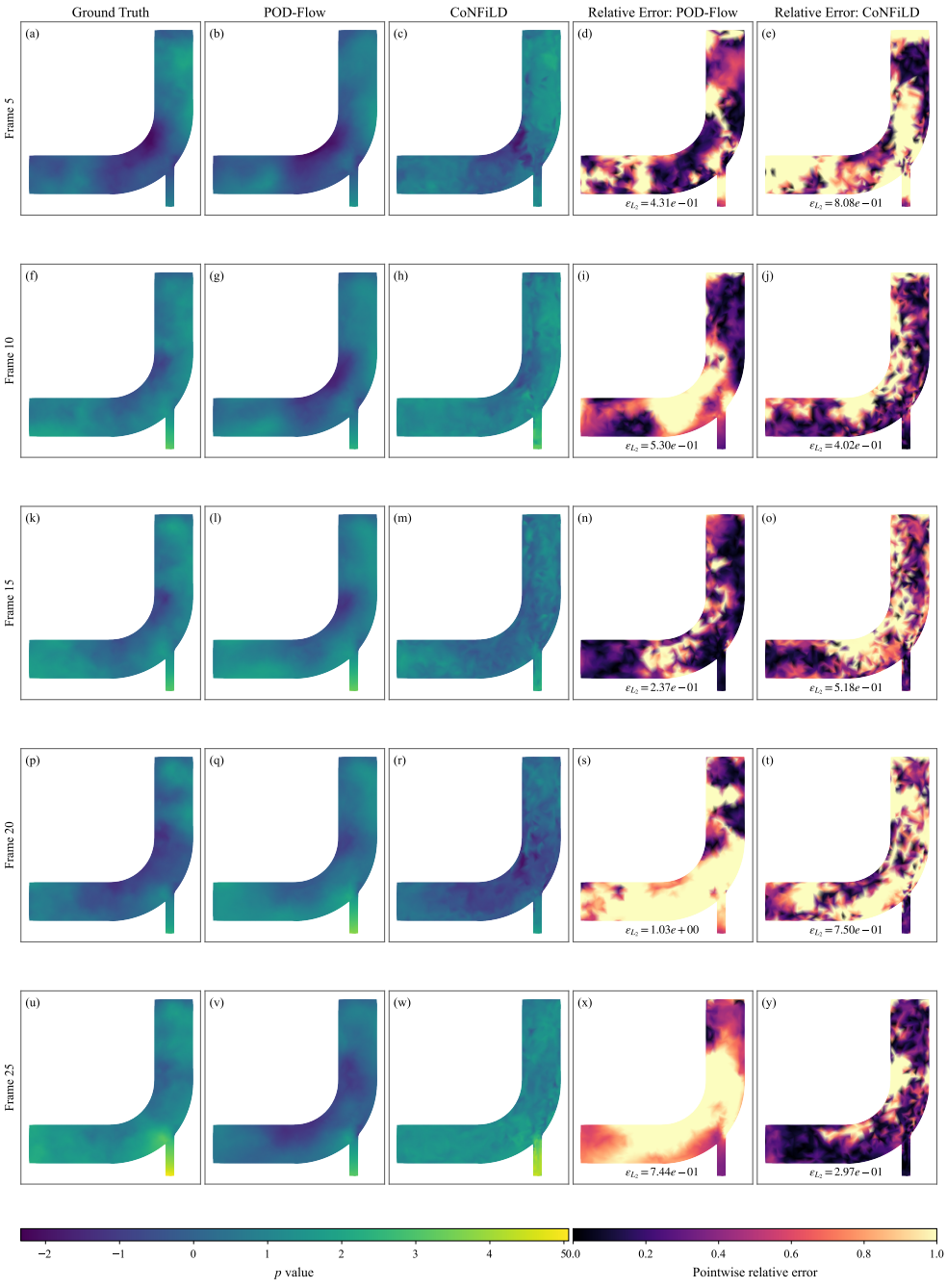


Figure 14: Pipe-flow pressure reconstructions by POD-Flow and CoNFiLD at selected physical time instants, using 368 observation nodes.

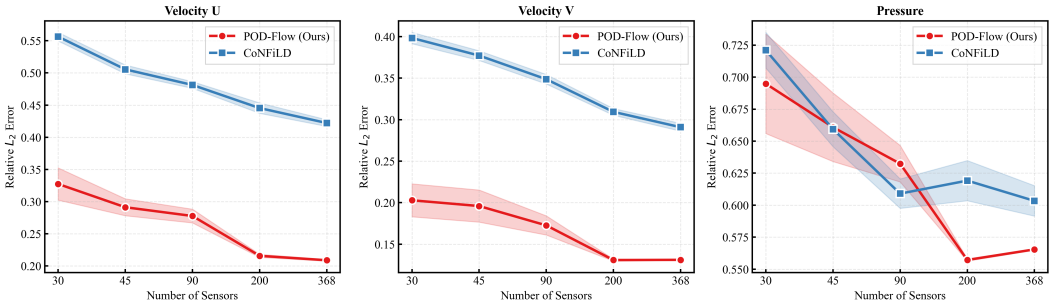


Figure 15: Relative L_2 errors of pipe-flow velocity components (u, v) and pressure p for POD-Flow and CoNFiLD under different numbers of observation nodes.

C.4 Displacement and Material-Property Reconstruction

Figures 16–21 present reconstructions of the six geological fields from displacement observations at 30 of the 13,481 mesh nodes. POD-Flow more accurately recovers the three displacement components than CoNFiLD. Performance on material properties is more mixed: POD-Flow achieves a slightly lower error for Young's modulus, whereas CoNFiLD obtains lower errors for Poisson's ratio and density. These comparisons complement the variable-wise errors reported in Section 5.2.

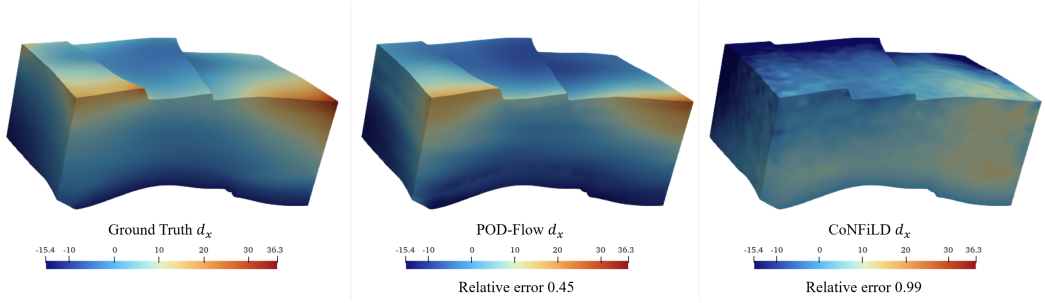


Figure 16: Reference and reconstructed x-displacement fields d_x for POD-Flow and CoNFiLD.

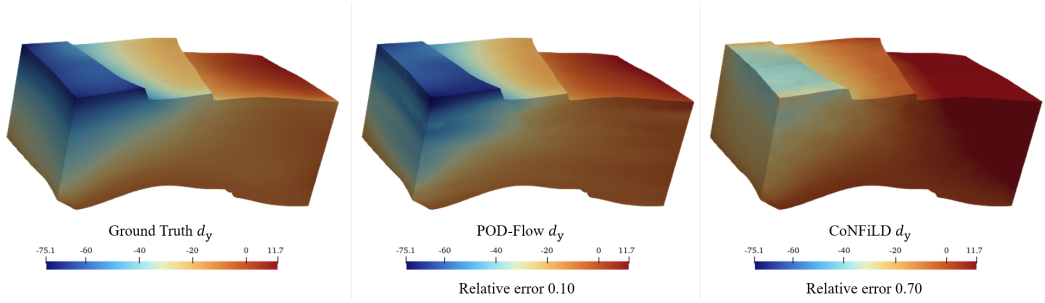


Figure 17: Reference and reconstructed y-displacement fields d_y for POD-Flow and CoNFiLD.

Figure 22 further compares the relative L_2 errors of POD-Flow and CoNFiLD for the displacement components and material parameters using 30, 45, 90, 200, and 368 observation nodes. This comparison evaluates how reconstruction accuracy changes with observation density for each physical quantity in the linear elasticity inverse task.

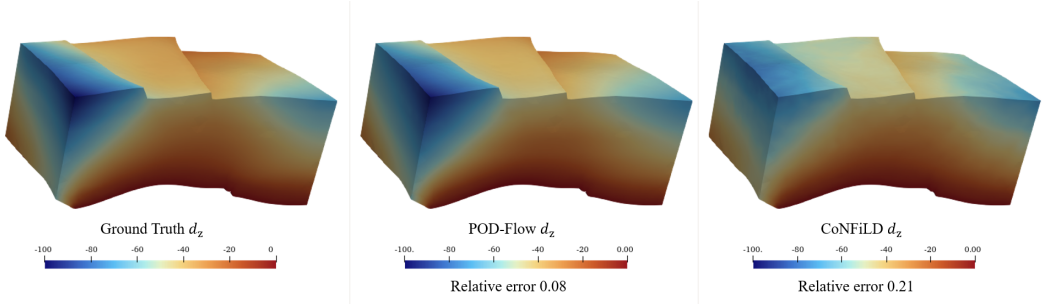


Figure 18: Reference and reconstructed z-displacement fields d_z for POD-Flow and CoNFiLD.

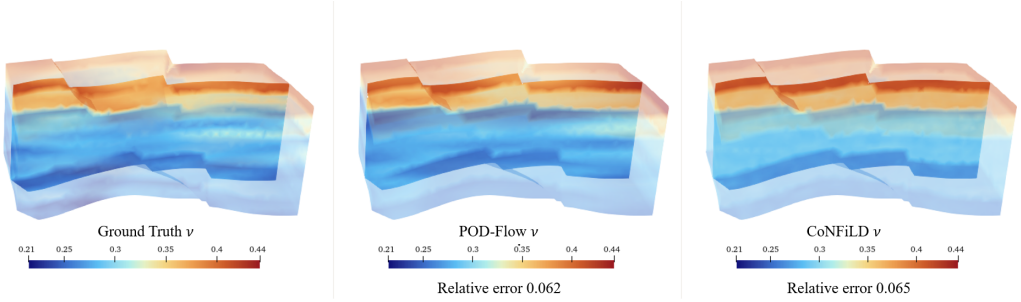


Figure 19: Reference and reconstructed Poisson's ratio fields ν for POD-Flow and CoNFiLD.

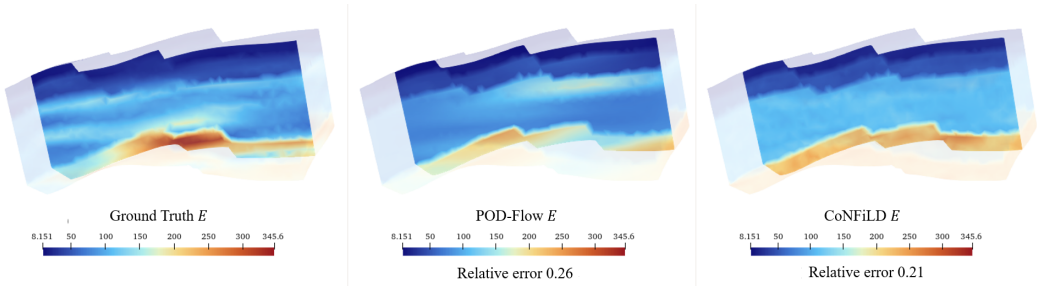


Figure 20: Reference and reconstructed Young's modulus fields E for POD-Flow and CoNFiLD.

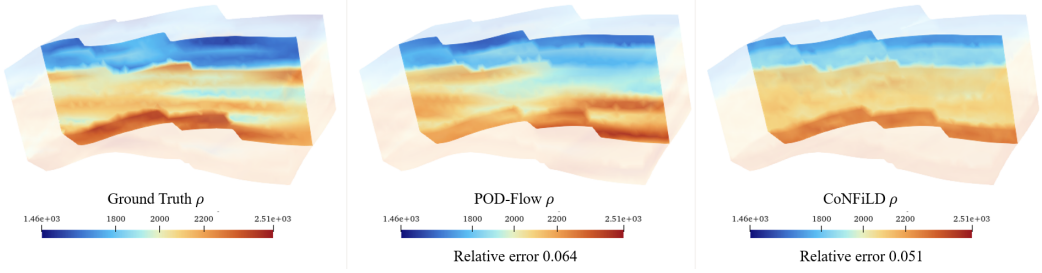


Figure 21: Reference and reconstructed density fields ρ for POD-Flow and CoNFiLD.

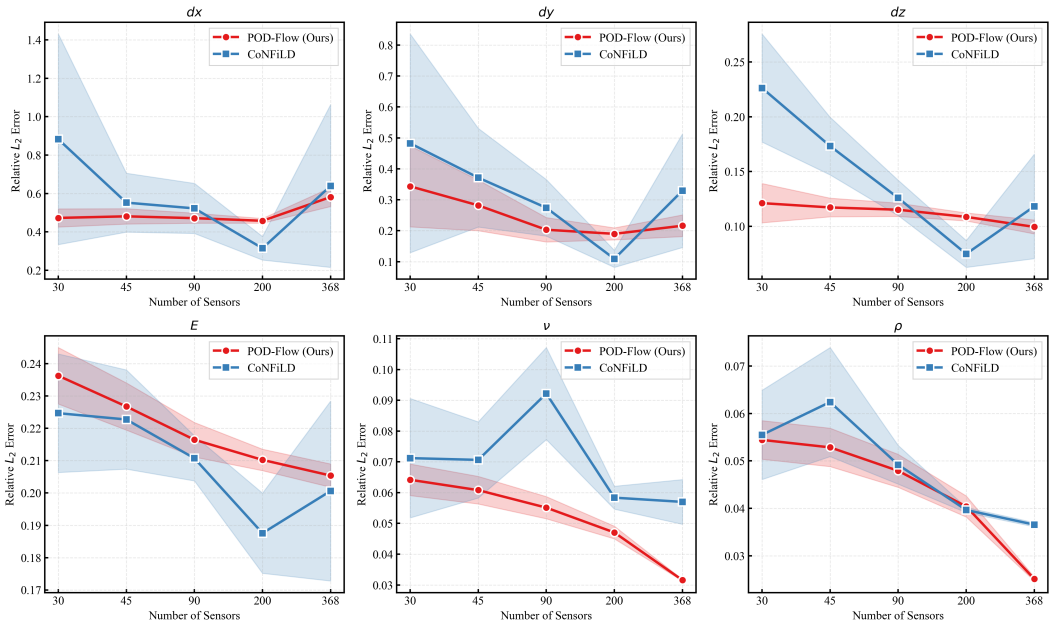
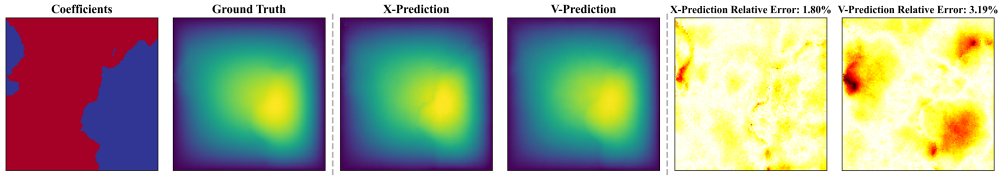


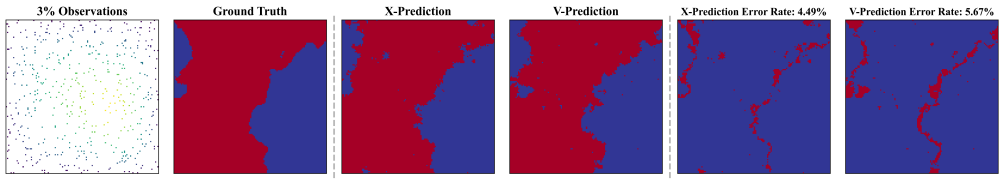
Figure 22: Relative L_2 errors of linear elasticity displacement d and material parameters E, ν, and ρ for POD-Flow and CoNFiLD under different numbers of observation nodes.

C.5 Additional Results on Prediction Target

Figure 23 provides the full comparison of guided-generation performance between x-prediction and v-prediction on Darcy. The top row shows the forward problems, and the bottom row shows the inverse problems. Consistent with the main text, x-prediction achieves lower reconstruction and inversion errors than v-prediction under the same PODNO backbone.



(a) Forward problems.



(b) Inverse problems.

Figure 23: Comparison of guided-generation performance between x-prediction and v-prediction on the forward and inverse problems.