MANIFOLD-PROJECTED DIFFUSION MODEL FOR PHYSICAL FIELD GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models are expressive generative priors for physical fields, but standard formulations typically treat coupled field variables as free outputs even when some are deterministically determined by known physical operators. As a result, generated parameter–state pairs can be individually plausible yet mutually inconsistent under the governing physics, particularly when forcing or boundary conditions shift beyond those seen during training. We propose the Manifold-Projected Diffusion Model (MPDM), which factorizes physical-field generation according to this dependency structure. Instead of directly denoising the complete field stack, MPDM predicts a compact latent representation of the independent physical parameter field, decodes it into a feasible parameter map, and uses a differentiable numerical solver to render the dependent state fields under prescribed conditions. This construction restricts each clean-state prediction to a solver-rendered, operator-consistent manifold while allowing reconstruction losses to propagate through the solver for end-to-end training without an auxiliary physics-residual objective. Across multiple PDE systems under controlled distribution shifts, MPDM consistently reduces state reconstruction error relative to ambient full-state and residual-based diffusion, while achieving strong parameter recovery and inverse consistency. These results demonstrate the benefit of aligning generative parameterization with the dependency structure imposed by the governing physics.

1 INTRODUCTION

Diffusion models have become powerful generative priors for scientific and physical fields Dhariwal & Nichol (2021); Rombach et al. (2022); Ho & Salimans (2022); Nichol & Dhariwal (2021); Karras et al. (2022). Unlike free-form image generation, however, physical fields are governed by known structural relations among system variables. A typical sample may contain multiple solution fields under prescribed excitations or boundary conditions together with an underlying spatial parameter field Um et al. (2020); Sanchez-Gonzalez et al. (2020); Li et al. (2020b). These components are not independent: conservation laws, boundary conditions, and constitutive relations restrict valid field stacks to a highly structured subset of the ambient space Cho et al. (2023). Consequently, a generated field stack can appear visually plausible while remaining physically inconsistent if its parameter and state variables do not satisfy the governing operator Cao (2021).

Existing diffusion-based approaches do not fully exploit this dependency structure. Data-driven models, including pixel-space and latent diffusion models as well as PDE-oriented generative priors such as DiffusionPDE, typically learn distributions over complete field representations Dhariwal & Nichol (2021); Rombach et al. (2022); Huang et al. (2024). While expressive, they do not structurally guarantee that generated parameter maps and state fields correspond under the prescribed physical conditions. Physics-informed diffusion models instead introduce governing-equation residuals into the training objective Bastek et al. (2025), turning physical consistency into a trade-off controlled by penalty weights. Neural operators and learned PDE solvers provide efficient approximations to physical solution maps Li et al. (2020a); Um et al. (2020), but their predictions remain subject to surrogate approximation error when used to render dependent physical states.

To exploit the deterministic dependency among physical variables, we introduce the Manifold-Projected Diffusion Model (MPDM). Rather than directly denoising the complete field stack,

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.

MPDM predicts a compact representation of the independent physical parameter field and reconstructs the dependent states through a differentiable numerical solver. Specifically, a timestep-conditioned predictor estimates a latent code, which is decoded into a feasible parameter field. Given this parameter field and the prescribed physical conditions, the solver evaluates the governing operator to render the corresponding state fields. The resulting clean estimate is assembled from the decoded parameter field and the solver-rendered states, thereby restricting each prediction to an operator-consistent subset of the ambient field space.

This factorization changes the role of learning in the generative process. The neural network is responsible for inferring the independent physical structure, while deterministic parameter–state coupling is delegated to the known forward operator rather than learned implicitly or encouraged through an auxiliary residual penalty. Because the numerical renderer is differentiable, reconstruction losses on the generated state fields propagate through the solver and decoder to the latent predictor, allowing MPDM to be optimized end-to-end with an $\hat{x}_0$-prediction diffusion objective, without introducing an auxiliary physics-residual loss.

This structural factorization is particularly beneficial under distribution shifts in physical conditions. Whereas a full-state denoiser must extrapolate the learned joint relationship between parameter and state fields, MPDM renders the dependent states from the predicted parameter field using the prescribed forward operator. As a result, the generated parameter–state pair remains consistent with the embedded physical solver even when the forcing or boundary conditions differ from those encountered during training. We evaluate this property on multiple physical-field benchmarks under controlled shifts in forcing families and boundary configurations, assessing reconstruction accuracy, parameter recovery, and physical consistency while separately examining the effects of solver-based rendering and coordinate-space parameterization. Across these settings, MPDM consistently improves state reconstruction over ambient and residual-based diffusion, while achieving strong parameter recovery and inverse consistency.

Contributions. Our contributions are threefold: (1) We formulate physical-field generation by factorizing independent physical parameters from state variables deterministically induced by a known forward operator. (2) We propose MPDM, which performs compact coordinate-space denoising and differentiable solver-based rendering to construct operator-consistent clean predictions without an auxiliary physics-residual objective. (3) Through controlled OOD evaluations and targeted ablations, we show how solver-based rendering and coordinate-space parameterization jointly improve state reconstruction, parameter recovery, and physical consistency.

2 Related Work

Data-Driven Methods. DDPMs Ho et al. (2020) learn generative distributions through a reverse denoising process in the ambient data space, while latent diffusion models move generation to learned compressed representations Rombach et al. (2022). Recent PDE-oriented diffusion models, such as DiffusionPDE Huang et al. (2024), further model joint coefficient–solution distributions for forward and inverse PDE problems under partial observations. These approaches demonstrate the effectiveness of diffusion priors for physical systems, but generally retain the parameter and state fields as jointly learned generative variables. MPDM instead exploits their deterministic dependency by learning the independent physical representation and rendering the dependent states through the prescribed forward operator.

Physics-Constrained Generative Modeling. Physics-informed diffusion models incorporate governing equations through residual-based training objectives Bastek et al. (2025), while recent constrained generative methods enforce feasibility through projection or constrained sampling Christopher et al. (2024); Baldan et al. (2026); Blanke et al. (2026). These methods primarily focus on enforcing or restoring feasibility of generated states during training or sampling. This leaves open a complementary question: whether variables that are deterministically specified by known physics should be generated independently in the first place.

Differentiable Forward Models and Solvers. Differentiable forward models have also been integrated into diffusion and scientific-learning pipelines. Diffusion with Forward Models Tewari et al. (2023) incorporates a known differentiable forward process into diffusion-based inverse problems, while solver-in-the-loop methods Um et al. (2020) enable gradients to propagate through numeri-

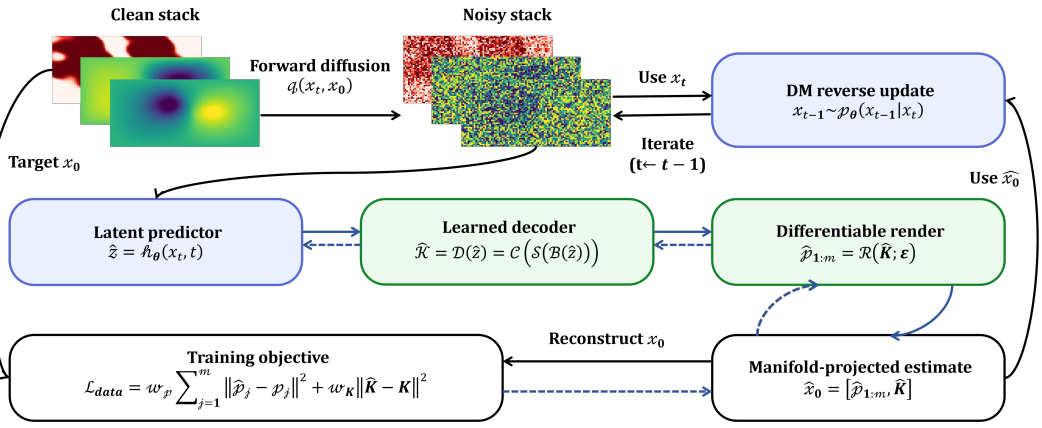


Figure 1: Overview of MPDM. Given a noisy field stack x_t, MPDM predicts a compact latent representation $\hat{z}$, decodes the independent parameter field $\hat{K}$, and renders the dependent states through the differentiable forward operator $\mathcal{R}_\mathcal{E}$. The resulting clean estimate $\hat{x}_0 = [\hat{p}_{1:m}, \hat{K}]$ is parameter–state consistent with the embedded physical solver by construction. Solid arrows denote forward computation and sampling, while dashed arrows indicate gradient propagation.

cal PDE solvers. Neural operators such as FNO Li et al. (2020a) instead learn efficient surrogate mappings between function spaces. In MPDM, the forward operator directly defines the dependent components of each clean-state prediction, while the denoiser learns only a compact representation of the independent physical parameter field.

3 METHOD

We propose the MPDM, a diffusion framework for physical field stacks whose components are coupled by a known forward operator. As illustrated in Figure 1, rather than directly denoising the complete field stack in the ambient space, MPDM factorizes the clean prediction into an independent physical parameter field and its dependent state fields. The denoiser predicts a compact latent representation of the parameter field, which is decoded into a feasible spatial map, while a differentiable numerical solver renders the corresponding state fields under the prescribed physical conditions. This construction restricts the clean prediction to an operator-induced rendered manifold and enforces parameter–state consistency with the embedded physical solver by construction. The geometric, solver-consistency, and coordinate-stability properties of this construction are formalized in Appendix C.

Physical Field as Rendered Manifolds. A clean sample is represented as an ambient multi-channel field stack containing (i) dependent state fields under multiple prescribed problem conditions and (ii) an independent spatial parameter field:

$$x_0 = [p_{1:m}, K] \in \mathbb{R}^{(m+1)\times H\times W}, \tag{1}$$

where $p_{1:m} = [p_1, \ldots, p_m]$ denotes the dependent state fields under m prescribed conditions, K denotes the independent spatial parameter field, and $\mathcal{K}$ denotes its admissible parameter space. Here p_j is used as generic notation for a dependent state variable; in the Navier–Stokes benchmark, it is instantiated as the vorticity field ω_j. The diffusion variables x_t remain in this ambient stack space. MPDM restricts the denoiser's clean estimate $\hat{x}_0$ through the structured path

$$\hat{z} \longrightarrow \hat{K} \longrightarrow \mathcal{R}(\hat{K}; \mathcal{E}). \tag{2}$$

We assume that the governing physics induces a deterministic forward map from the parameter field to the corresponding state fields. Let $\mathcal{R}$ denote this known forward operator, implemented through an implicit PDE solve:

$$[p_1, \ldots, p_m] = \mathcal{R}(K; \mathcal{E}) \equiv \mathcal{R}_\mathcal{E}(K), \tag{3}$$

where $\mathcal{E}$ denotes the prescribed problem conditions, including the forcing and boundary data. For a fixed condition set $\mathcal{E}$, the operator induces the solution manifold

$$\mathcal{M}_{\mathcal{E}} = \left\{ \left[\mathcal{R}_{\mathcal{E}}(K), K \right] : K \in \mathcal{K} \right\}. \tag{4}$$

Hence, although x_0 is represented in a high-dimensional ambient space, its dependent state fields are deterministically specified once K and $\mathcal{E}$ are fixed. This structure motivates learning a compact representation of the independent parameter field and recovering the remaining state variables through the known forward operator.

Coordinate-Space Parameterization and Learned Decoding. Our instantiation represents the independent parameter field K using a compact latent code $z \in \mathbb{R}^{d_z}$. A learned decoder $\mathcal{D}_\phi$ maps z into the feasible parameter space:

$$K = (\mathcal{C} \circ \mathcal{S} \circ \mathcal{B}_\phi)(z) \equiv \mathcal{D}_\phi(z) \in \mathcal{K}, \tag{5}$$

where $\mathcal{B}_\phi$ is a learned latent-to-field lift, $\mathcal{S}$ is an optional stabilizing transform, and $\mathcal{C}$ is a differentiable feasibility map into $\mathcal{K}$. When no additional stabilization is required, $\mathcal{S}$ reduces to the identity map.

Combining the learned decoder with the prescribed forward operator defines the rendered mapping

$$\Phi_{\mathcal{E}}(z) = \left[\mathcal{R}_{\mathcal{E}}(\mathcal{D}_\phi(z)), \mathcal{D}_\phi(z) \right]. \tag{6}$$

Accordingly, the latent representation parameterizes the learned rendered subset

$$\widetilde{\mathcal{M}}_{\mathcal{E}} = \left\{ \Phi_{\mathcal{E}}(z) : z \in \mathbb{R}^{d_z} \right\} \subseteq \mathcal{M}_{\mathcal{E}}. \tag{7}$$

Under a smooth-embedding assumption on $\mathcal{D}_\phi$, this construction defines a d_z-dimensional embedded manifold; the corresponding geometric result is given in Appendix C.1. Thus, z provides a compact coordinate-space representation of the parameter field, while the dependent state fields are deterministically rendered from the decoded parameter through the known forward operator.

Field-Space vs. Coordinate-Space Parameterization. Within the same solver-rendered framework, the independent parameter field can be represented either directly in field space or through a compact coordinate space. A field-space variant predicts a full-resolution unconstrained parameter representation from the noisy stack,

$$K_{\text{logits}} = \mathcal{G}_\theta(x_t, t), \qquad \hat{K} = \mathcal{C}(K_{\text{logits}}) \in \mathcal{K}, \tag{8}$$

followed by rendering of the dependent states,

$$\hat{p}_{1:m} = \mathcal{R}_{\mathcal{E}}(\hat{K}), \qquad \hat{x}_0 = [\hat{p}_{1:m}, \hat{K}]. \tag{9}$$

For bounded parameter fields, the feasibility map can be implemented as

$$\hat{K} = K_{\min} + (K_{\max} - K_{\min})\, \sigma(K_{\text{logits}}), \tag{10}$$

which explicitly constrains the prediction to the prescribed parameter range.

MPDM instead performs denoising in coordinate space:

$$\hat{z} = h_\theta(x_t, t), \qquad \hat{K} = \mathcal{D}_\phi(\hat{z}). \tag{11}$$

The field-space variant therefore estimates an $H \times W$ parameter field directly, whereas MPDM predicts a d_z-dimensional latent representation before decoding it into $\mathcal{K}$. Both variants use the same physical renderer $\mathcal{R}_{\mathcal{E}}$, so their difference isolates the effect of coordinate-space parameterization while preserving the same parameter–state coupling mechanism. In MPDM, the decoder constrains the predicted parameter field to the learned feasible subset, and the forward operator then deterministically renders the associated state fields.

Differentiable Physical Renderer. The physical renderer $\mathcal{R}_{\mathcal{E}}$ must be differentiable with respect to the predicted parameter field $\hat{K}$ so that supervision on the rendered states can propagate to the learned parameter representation. Given $\hat{K}$ and the prescribed condition set $\mathcal{E}$, $\mathcal{R}_{\mathcal{E}}$ solves the discretized governing equations to obtain the dependent state fields $\hat{p}_{1:m} = \mathcal{R}_{\mathcal{E}}(\hat{K})$. In our implementation, the numerical renderer is benchmark specific. Darcy uses a CG-based variable-coefficient elliptic solve with implicit-adjoint differentiation; Helmholtz differentiates through a CG solve of the

reaction–diffusion system; and Navier–Stokes uses a nonlinear periodic pseudo-spectral fixed-point solver with implicit-adjoint differentiation. Multiple prescribed conditions are evaluated within the corresponding batched rendering call. Because each renderer provides derivatives with respect to $\hat{K}$, reconstruction gradients from $\hat{p}_{1:m}$ propagate through $\mathcal{R}_{\mathcal{E}}$ to $\hat{K}$ and further through the decoder to the latent predictor h_θ. The coefficient decoder $\mathcal{D}_\phi$ is pretrained on the training coefficient fields and held fixed during the canonical MPDM diffusion stage; gradients nevertheless propagate through it with respect to its latent input, so the latent-predictor parameters θ are optimized using both parameter- and state-reconstruction supervision. This enables end-to-end optimization of the diffusion predictor without introducing an auxiliary PDE-residual objective, while maintaining parameter–state consistency with the embedded discrete solver up to numerical tolerance. This solver-consistency property is formalized in Appendix C.2.

4 Model Formulation

We now formulate MPDM within the standard x_0-prediction DDPM framework. Recall that the clean physical-field stack is $x_0 = [p_{1:m}, K]$, while the predicted clean state in MPDM is constrained to the learned rendered manifold $\widetilde{\mathcal{M}}_{\mathcal{E}}$ defined in the previous section.

Ambient Diffusion over Physical Field Stacks. The forward diffusion process progressively perturbs the clean stack x_0 with Gaussian noise:

$$\begin{aligned} q(x_t \mid x_0) &= \mathcal{N}\big(x_t; \sqrt{\bar{\alpha}_t}\, x_0, (1-\bar{\alpha}_t) I\big)\,, \\ x_t &= \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \end{aligned} \tag{12}$$

where $\bar{\alpha}_t$ is determined by the diffusion noise schedule. We use the cosine schedule of Improved DDPM Nichol & Dhariwal (2021). The noisy variables x_t remain in the ambient field-stack space throughout the diffusion process.

Manifold-Projected Denoising Diffusion. A standard x_0-prediction denoiser directly estimates the complete clean stack in the ambient space as $\hat{x}_0 = g_\theta(x_t, t)$, without an architectural guarantee that the predicted parameter and state fields satisfy the prescribed forward operator. MPDM instead predicts a compact coordinate representation $\hat{z} = h_\theta(x_t, t)$ and constructs the clean estimate through the rendered mapping introduced previously:

$$\hat{x}_0 = \Phi_{\mathcal{E}}\big(h_\theta(x_t, t)\big) = \left[\mathcal{R}_{\mathcal{E}}(\mathcal{D}_\phi(h_\theta(x_t, t)))\,, \mathcal{D}_\phi(h_\theta(x_t, t))\right]. \tag{13}$$

Therefore, $\hat{x}_0 \in \widetilde{\mathcal{M}}_{\mathcal{E}} \subseteq \mathcal{M}_{\mathcal{E}}$, so the predicted parameter and state fields are coupled through the embedded forward solver rather than predicted independently. Appendix C.3 further characterizes how errors in the coordinate prediction propagate through the decoder and physical renderer to the complete field stack.

For both MPDM and PIDM, we define the channel-weighted clean-state reconstruction loss as

$$\ell_{\text{data}} = w_p \sum_{j=1}^{m} \|\hat{p}_j - p_j\|_2^2 + w_K \left\|\hat{K} - K\right\|_2^2. \tag{14}$$

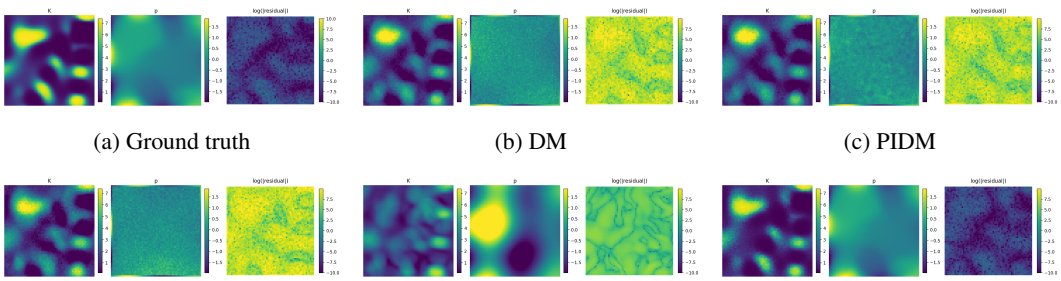
For comparison, PIDM retains the ambient clean-state predictor $\hat{x}_0 = g_\theta(x_t, t)$ and incorporates the governing equations through an auxiliary residual term:

$$\mathcal{L}_{\text{PIDM}} = \mathbb{E}_{x_0, t, \epsilon}\left[\ell_{\text{data}} + \lambda \left\|\mathcal{F}(\hat{x}_0; \mathcal{E})\right\|_2^2\right], \tag{15}$$

where $\mathcal{F}$ denotes the benchmark-specific discrete physics residual. The residual term encourages physical feasibility but introduces an additional penalty weight λ, whereas MPDM imposes parameter–state coupling directly through the clean-state parameterization. A controlled experiment isolating the effect of this hard manifold projection from coordinate-space parameterization is provided in Appendix **??**.

Training Objective and Sampling. MPDM is trained using the same channel-weighted x_0-prediction objective, without an auxiliary PDE-residual term:

$$\mathcal{L}_{\text{MPDM}} = \mathbb{E}_{x_0, t, \epsilon}\left[\ell_{\text{data}}\right]. \tag{16}$$



(a) Ground truth (b) DM (c) PIDM

(d) DiffusionPDE (e) PISD (f) MPDM

Figure 2: Darcy flow.

Here, the expectation is taken over training pairs $(x_0, \mathcal{E}) \sim p_{\text{data}}$, $t \sim \text{Unif}\{1, \dots, T\}$, and $\epsilon \sim \mathcal{N}(0, I)$, with x_t constructed according to Equation (12). The weights w_p and w_K balance the reconstruction terms for the state and parameter fields. Because $\hat{p}_{1:m}$ is obtained through the differentiable renderer, state-reconstruction gradients propagate through $\mathcal{R}_{\mathcal{E}}$ and $\mathcal{D}_\phi$ to the latent predictor.

During sampling, the predicted clean estimate $\hat{x}_0$ from Equation (13) is used in the standard DDPM reverse transition to obtain x_{t-1}. The noisy states x_t remain in the ambient field space, while only the clean prediction at each reverse step is solver-rendered. Consequently, the final output at $t = 0$ is parameter–state consistent with the embedded discrete solver up to its numerical solve accuracy.

5 Experiments

All experiments use an RTX 4090 GPU, Adam with learning rate 2×10^{-4}, batch size 32, and resolution 128. We compare diffusion-based generative priors under a common benchmark setting. DiffusionPDE and PISD are matched-backbone adaptations that retain their residual-guided and spectral mechanisms; thus, the results evaluate these benchmark implementations rather than exact reproductions of the original training recipes.

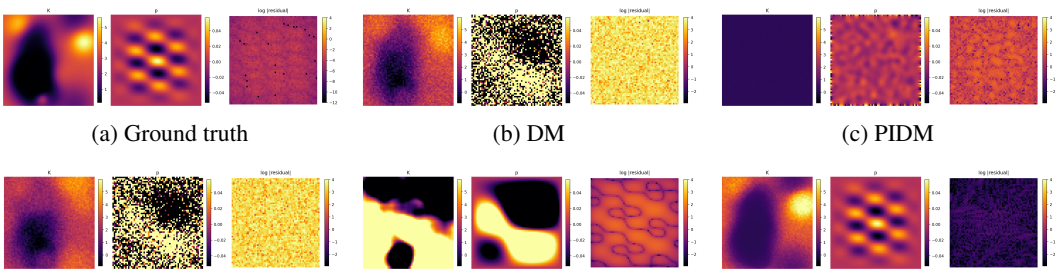
5.1 Compared Methods

DM Ho et al. (2020) predicts the full clean stack in ambient space. PIDM Bastek et al. (2025) uses the same parameterization with an additional discrete physics-residual loss. DiffusionPDE Huang et al. (2024) combines a joint state–coefficient diffusion prior with residual-gradient guidance during reverse sampling. PISD performs diffusion in normalized joint spectral coordinates with frequency-aware residual guidance. MPDM instead predicts a low-dimensional coefficient code, decodes $\hat{K}$, and renders the dependent states with the differentiable numerical solver, so parameter–state coupling is imposed by the clean-state parameterization rather than by a penalty or sampling-time correction.

5.2 OOD evaluation

Darcy, Helmholtz, and Navier–Stokes use 500, 2000, and 1000 training/test samples, respectively, with $m = 3$ state channels sharing each coefficient realization $K(x, y)$. The coefficient distribution is fixed between ID and OOD sets; only the prescribed condition $\mathcal{E}$ changes. Darcy shifts Dirichlet boundary modes from $\{1, 2, 3\}$ to $\{4, 5, 6\}$ with zero body forcing; Helmholtz shifts forcing pairs from $(1, 1), (2, 1), (1, 2)$ to $(3, 3), (4, 3), (3, 4)$; Navier–Stokes keeps forcing frequencies $(1, 1), (1, 2), (2, 1)$ fixed and shifts the forcing phase from 0 to $\pi/2$. For Darcy and Helmholtz, $x_0 = [p_{1:m}, K]$; for Navier–Stokes, $x_0 = [\omega_{1:m}, K]$.

We report state MSE (MSE_p, or $\text{MSE}_{\text{vorticity}}$), coefficient MSE ($\text{MSE}_K$), mean absolute physical residual (MAR), inverse consistency ICS_K computed from the ground-truth state paired with $\hat{K}$, and relative training/inference memory. MAR measures self-consistency of the predicted pair, whereas ICS_K tests whether $\hat{K}$ explains the true state. In the repository protocol, DM, PIDM, MPDM, and



(a) Ground truth (b) DM (c) PIDM

(d) DiffusionPDE (e) PISD (f) MPDM

Figure 3: Helmholtz.

the displayed DiffusionPDE row use ambient noisy-stack denoising; the displayed PISD row uses native spectral corruption. DiffusionPDE and PISD apply their defining guidance during reverse sampling, so these paired rows diagnose their unguided priors and PISD is not a strictly identical-corruption comparison.

Darcy flow Darcy (1856); Bear (2013); Efendiev & Hou (2009). The benchmark solves

$$\begin{aligned} -\nabla \cdot \big(K(x,y)\nabla p_j(x,y)\big) &= 0, \\ p_j|_{\partial\Omega} = g_j, \quad j = 1,\dots,m, &\qquad \Omega = [0,1]^2. \end{aligned} \tag{17}$$

Table 1: OOD evaluation results in Darcy flow.

model	$\mathrm{MSE}_p \downarrow$	$\mathrm{MSE}_K \downarrow$	MAR $\downarrow$	$\mathrm{ICS}_K \downarrow$	$\mathrm{RM}_{train} \downarrow$	$\mathrm{RM}_{infer} \downarrow$
DM	0.288780	1.04826	4564.54	57.7571	1	1
PIDM	0.273736	1.08376	3135.36	55.4784	0.991472	0.992329
DiffusionPDE	0.316776	1.04184	4518.06	58.0596	1	1.07974
PISD	1.06669	3.21320	152.632	68.9827	**0.546181**	**0.671437**
MPDM	**0.00665130**	**0.557039**	**0.00355846**	**18.3720**	0.594175	0.713418

The OOD boundary shift separates residual reduction from coefficient identification. PIDM reduces the governing residual but does not improve K recovery, consistent with the multiplicative structure $K\nabla p$: coordinated errors in K and ∇p can lower self-residual without recovering the true permeability. The unguided DiffusionPDE prior remains close to DM. PISD likewise attains a much lower MAR but worse paired errors and ICS_K; spectral truncation suppresses derivative-sensitive high-frequency content, while its different corruption space precludes a direct denoising ranking. MPDM instead infers $\hat{K}$ and re-renders pressure using the OOD boundary data, producing very small state error and MAR together with substantially lower MSE_K and ICS_K. The near-zero MAR is expected from solver-based rendering; the coefficient and inverse-consistency gains provide the stronger evidence of improved inverse recovery. PISD has the lowest memory ratios, while MPDM also remains below DM because compact coefficient coordinates and implicit solver differentiation offset much of the rendering cost.

Helmholtz Evans (2022); Gilbarg et al. (1998); Colton et al. (1998). We consider

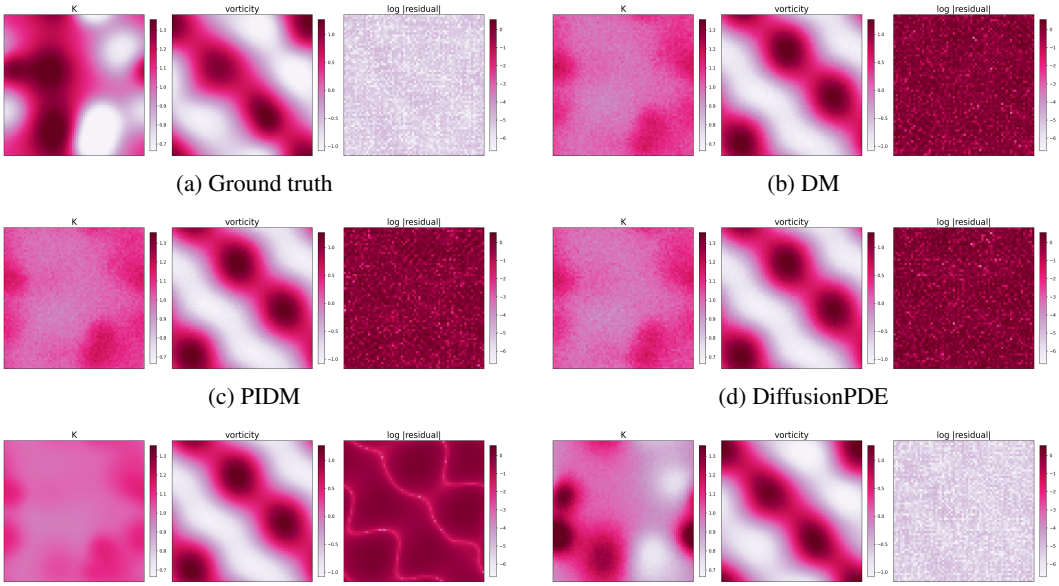
$$\begin{aligned} -\Delta p_j(x,y) + K(x,y)p_j(x,y) &= f_j(x,y), \\ p_j|_{\partial\Omega} = 0, \quad j = 1,\dots,m, &\quad \Omega = [0,1]^2. \end{aligned} \tag{18}$$

For a perturbation δK, $(-\Delta + K)\delta p_j \approx -(\delta K)p_j$. Hence, under the higher-frequency OOD forcing, the Laplacian becomes relatively dominant and state error becomes less sensitive to coefficient error.

This conditioning explains the apparent metric separation. PIDM strongly lowers state error and MAR while worsening MSE_K and ICS_K, showing that residual satisfaction need not resolve a weakly identified reaction coefficient. DiffusionPDE improves coefficient and inverse-consistency

Table 2: OOD evaluation results in Helmholtz.

model	$\text{MSE}_p \downarrow$	$\text{MSE}_K \downarrow$	MAR$\downarrow$	$\text{ICS}_K \downarrow$	$\text{RM}_{train} \downarrow$	$\text{RM}_{infer} \downarrow$
DM	0.07904	3.188	3239	0.01137	1.000000	1.000000
PIDM	0.0002729	7.658	24.83	0.01541	1.000000	1.019000
DiffusionPDE	0.0617863	1.93169	2886.91	0.00858957	0.978281	0.970177
PISD	0.00828834	73.541	9.15286	0.0468167	**0.534925**	**0.591466**
MPDM	$\mathbf{7.704 \times 10^{-9}}$	**1.092**	**0.008916**	**0.005532**	1.438000	0.658100



(a) Ground truth (b) DM

(c) PIDM (d) DiffusionPDE

(e) PISD (f) MPDM

Figure 4: Steady variable-viscosity Navier–Stokes.

error relative to DM but retains a large state residual in its unguided prior diagnostic. PISD attains low MAR and memory use but very poor coefficient recovery; spectral truncation suppresses derivative-sensitive components, while K enters only through Kp, so large coefficient errors can have limited residual effect when p is small or the Laplacian dominates. MPDM obtains the lowest coefficient and inverse-consistency errors and an effectively negligible rendered-state error because it explicitly solves the shifted equation for $\hat{K}$. Its higher training but lower inference memory further shows that the cost of hard rendering is solver dependent rather than uniformly smaller.

Navier–Stokes Li et al. (2020a). The third benchmark is steady two-dimensional incompressible flow on the periodic square $[0, 2\pi]^2$, with spatial viscosity $\nu(x, y) = \nu_{\text{ref}} K(x, y)$ and $\nu_{\text{ref}} = 0.3$. The implementation uses a streamfunction $\mathbf{u}_j = (\partial_y \psi_j, -\partial_x \psi_j)$, $-\Delta \psi_j = \omega_j$, and solves the steady vorticity equation

$$\mathbf{u}_j \cdot \nabla \omega_j - \nabla \times \nabla \cdot \left[\nu_{\text{ref}} K \left(\nabla \mathbf{u}_j + \nabla \mathbf{u}_j^{\mathsf{T}} \right) \right] = g_j^{(\phi)}. \tag{19}$$

Table 3: OOD evaluation results in Navier-Stokes.

model	$\text{MSE}_{\text{vorticity}} \downarrow$	$\text{MSE}_K \downarrow$	MAR$\downarrow$	$\text{ICS}_K \downarrow$	$\text{RM}_{train} \downarrow$	$\text{RM}_{infer} \downarrow$
DM	0.441832	0.0643350	1.49034	0.439155	1	1
PIDM	0.436652	0.0640831	1.11499	0.355210	1	1
DiffusionPDE	0.441832	0.0643350	1.49034	0.439155	1	1.070196
PISD	0.432114	0.0626337	0.575231	**0.141427**	**0.635349**	**0.740348**
MPDM	**0.00326694**	**0.0375528**	**3.13536e-06**	0.188276	0.815517	0.846425

Table 4: Controlled ablation. Values are means over three optimization seeds. $\mathcal{E}_{\mathrm{den}}$ uses common ambient OOD corruption, whereas generation MAR and Joint SW_1 use common unpaired OOD generation. Dashes denote non-comparable native denoising protocols. For DiffusionPDE, $\mathcal{E}_{\mathrm{den}}$ is the unguided-prior diagnostic; generation metrics use full guidance.

Model	$\mathcal{E}_{\mathrm{den}} \downarrow$	Gen. MAR$\downarrow$	Joint $\mathrm{SW}_1 \downarrow$
DM	0.6030	0.7148	0.5402
PIDM	0.6027	0.7028	0.5377
DM-Cond	0.6091	0.7733	0.4813
PIDM-Cond	0.6091	0.7733	0.4813
$\mathrm{MPDM}_{\mathrm{field}}$	0.5598	2.50×10^{-7}	0.3094
MPDM	**0.5576**	2.51×10^{-7}	0.3389
$\mathrm{MPDM}_{\mathrm{FNO}}$	0.5603	0.2449	0.4049
K-DM + Solver	–	2.63×10^{-7}	0.3865
K-LDM + Solver	–	2.53×10^{-7}	0.3222
VAE-K + Solver	–	2.40×10^{-7}	**0.0783**
DiffusionPDE	0.6030	0.7111	0.5395
PISD	–	0.4385	0.4115

The nonlinear advection is evaluated spectrally with strict $2/3$ truncation and the full variable-viscosity stress is retained; OOD evaluation changes only the forcing phase.

PIDM reduces MAR and ICS_K only modestly while leaving paired vorticity error near DM, because nonlinear residual terms permit compensating errors among velocity, vorticity gradients, and viscosity. DiffusionPDE matches DM in the reported paired metrics because this row evaluates the shared unguided ambient prior. PISD yields the lowest ICS_K and a lower MAR, consistent with the alignment between its Fourier representation and the periodic spectral discretization, although its pixelwise state and coefficient errors improve only slightly. MPDM shows the complementary behavior: solving the nonlinear equation with the OOD phase yields the lowest vorticity error, coefficient MSE, and MAR. Its near-zero MAR again follows from solver construction, while PISD's lower ICS_K reflects that this operator-weighted metric emphasizes dynamically active coefficient components rather than uniform pixelwise error. PISD has the lowest memory cost; MPDM also remains below DM. Thus, the nonlinear benchmark supports the same mechanism as the elliptic cases: residual guidance improves operator consistency, whereas solver-based rendering directly recomputes the dependent state under the shifted physical condition.

5.3 Ablation Study

A controlled benchmark isolates explicit conditioning, coefficient-space parameterization, and numerical rendering; the complete construction and training protocol are given in Appendix A. Table 4 shows that MPDM achieves the lowest comparable OOD denoising score (0.5576), improving over DM-Cond (0.6091) and slightly over $\mathrm{MPDM}_{\mathrm{field}}$ (0.5598). Replacing the numerical renderer with FNO preserves similar coefficient accuracy but increases the generation residual from 2.51×10^{-7} to 0.2449, highlighting the consistency benefit of solver-based rendering. The advantage is task-specific rather than universal: in unconditional generation, VAE-K-Solver attains the lowest joint sliced-Wasserstein distance (0.0783), whereas MPDM obtains 0.3389. Thus, the ablation primarily supports MPDM for physically constrained noisy-stack inference and parameter–state consistency, not for unconditional generation in general.

6 Discussion

Notably, MPDM is a methodology rather than a bespoke architecture, and is largely orthogonal to recent diffusion approaches that achieve state-of-the-art performance primarily through network-layer design. Consequently, stronger diffusion backbones can in principle be reformulated under the MPDM paradigm when a differentiable physical renderer is available. Our matched-backbone comparisons are designed to isolate the effects of output parameterization, physics-guided objectives, and numerical versus learned rendering while limiting confounding differences in network architecture. A limitation of MPDM is its reliance on a low-dimensional manifold. While this assumption is often reasonable, it may become restrictive for highly multiscale systems.

7 AI USE STATEMENT

In this work, we used generative AI tools, including OpenAI ChatGPT, to polish and restructure the writing of the manuscript, assist with table formatting, inspect author-provided source code, check consistency between the manuscript and implementation, and refine the presentation and interpretation of experimental results. We reviewed all AI-assisted work: technical statements and interpretations were checked against the source code, experimental records, and cited literature, and reported numerical values were verified against the corresponding experimental results. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

8 ETHICS STATEMENT

Not applicable. This work develops generative models for numerical physical fields governed by partial differential equations. All training and evaluation data are synthetic samples generated by numerical solvers; the study involves no human subjects, personal or sensitive information, or data collected from individuals. MPDM guarantees consistency only with the embedded discretized physical model, and any use in real-world or safety-critical applications would require independent validation of the underlying physical model and numerical solver.

9 REPRODUCIBILITY STATEMENT

We have made the following efforts to ensure reproducibility. The MPDM parameterization, differentiable physical rendering, diffusion formulation, and training objective are specified in the Method and Model Formulation sections. The Experiments section describes the Darcy, Helmholtz, and Navier–Stokes benchmarks, the controlled OOD shifts, compared methods, evaluation metrics, hardware, and common optimization settings. The appendix provides the complete controlled-ablation setup, metric definitions, theoretical assumptions and proofs, and additional experiments isolating manifold projection and coordinate-space parameterization.

The accompanying supplementary code contains the implementations of all three PDE benchmarks, their numerical solvers and data generators, the MPDM and baseline implementations, shared training and evaluation routines, explicit normal and smoke-test configurations, software dependencies, and regression and numerical-verification tests. The normal experimental protocol supports three independent model seeds with a fixed data split, while all methods within each benchmark use the same generated data partitions and evaluation operators. Hyperparameter and guidance selection is performed using independent validation data before test evaluation. The code additionally records resolved configurations, data statistics, selected hyperparameters, per-trial logs, aggregate multi-seed statistics, configuration fingerprints, and split-provenance hashes. These materials provide the information required to reconstruct the synthetic datasets and reproduce the training and evaluation procedures reported in the paper.

REFERENCES

Giacomo Baldan, Qiang Liu, Alberto Guardone, and Nils Thuerey. Physics vs distributions: Pareto optimal flow matching with physics constraints. In *International Conference on Learning Representations*, volume 2026, pp. 101215–101243, 2026.

Jan-Hendrik Bastek, WaiChing Sun, and Dennis Kochmann. Physics-informed diffusion models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 3360–3385, 2025.

Jacob Bear. *Dynamics of fluids in porous media*. Courier Corporation, 2013.

Matthieu Blanke, Yongquan Qu, Sara Shamekh, and Pierre Gentine. Strictly constrained generative modeling via split augmented langevin sampling. In *International Conference on Learning Representations*, volume 2026, pp. 145927–145965, 2026.

Shuhao Cao. Choose a transformer: Fourier or galerkin. *Advances in neural information processing systems*, 34:24924–24940, 2021.

Junwoo Cho, Seungtae Nam, Hyunmo Yang, Seok-Bae Yun, Youngjoon Hong, and Eunbyung Park. Separable physics-informed neural networks. *Advances in Neural Information Processing Systems*, 36:23761–23788, 2023.

Jacob K Christopher, Stephen Baek, and Ferdinando Fioretto. Constrained synthesis with projected diffusion models. *Advances in Neural Information Processing Systems*, 37:89307–89333, 2024.

David L Colton, Rainer Kress, and Rainer Kress. *Inverse acoustic and electromagnetic scattering theory*, volume 93. Springer, 1998.

Henry Darcy. *Les fontaines publiques de la ville de Dijon: Exposition et application des principes à suivre et des formules à employer dans les questions de distribution d'eau: Ouvrage terminé par un appendice relatif aux fournitures d'eau de plusieurs villes, au filtrage des eaux et à la fabrication des tuyaux de fonte, de plomb, de tôle et de bitume*, volume 2. Victor Dalmont, éditeur, 1856.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Yalchin Efendiev and Thomas Y Hou. *Multiscale finite element methods: theory and applications*. Springer Science & Business Media, 2009.

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

David Gilbarg, Neil S Trudinger, David Gilbarg, and NS Trudinger. *Elliptic partial differential equations of second order*, volume 2. Springer, 1998.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jiahe Huang, Guandao Yang, Zichen Wang, and Jeong Joon Park. Diffusionpde: Generative pde-solving under partial observation. *Advances in Neural Information Processing Systems*, 37: 130291–130323, 2024.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020a.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Andrew Stuart, Kaushik Bhattacharya, and Anima Anandkumar. Multipole graph neural operator for parametric partial differential equations. *Advances in Neural Information Processing Systems*, 33:6755–6766, 2020b.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International conference on machine learning*, pp. 8459–8468. PMLR, 2020.

Ayush Tewari, Tianwei Yin, George Cazenavette, Semon Rezchikov, Josh Tenenbaum, Frédo Durand, Bill Freeman, and Vincent Sitzmann. Diffusion with forward models: Solving stochastic inverse problems without direct supervision. *Advances in Neural Information Processing Systems*, 36:12349–12362, 2023.

Kiwon Um, Robert Brand, Yun Raymond Fei, Philipp Holl, and Nils Thuerey. Solver-in-the-loop: Learning from differentiable physics to interact with iterative pde-solvers. *Advances in neural information processing systems*, 33:6111–6122, 2020.

A Controlled Ablation Details

Problem construction. The ablation uses a periodic reaction–diffusion problem on the unit square,

$$-a\Delta u_j(\boldsymbol{x}) + K(\boldsymbol{x})u_j(\boldsymbol{x}) = f_j(\boldsymbol{x}), \qquad a = 0.02, \quad j = 1, 2, 3, \tag{20}$$

discretized with a complete real Fourier collocation basis. Writing L for the discrete approximation of $-a\Delta$, the renderer solves $U = (L + \operatorname{diag} K)^{-1}F$ in double precision. Data generation, MPDM rendering, residual penalties, reported residuals, and spectral guidance use this same discrete operator.

The coefficient family is deliberately low-dimensional. For $z \sim \mathcal{N}(0, I_6)$,

$$K(\boldsymbol{x}) = \exp\left[c \tanh\left(\frac{0.8}{c}\sum_{\ell=1}^{6} z_\ell b_\ell(\boldsymbol{x})\right)\right], \qquad c = 1.5, \tag{21}$$

where $\sqrt{6}\, b_\ell$ are 1, $\sqrt{2}\cos(2\pi x)$, $\sqrt{2}\sin(2\pi x)$, $\sqrt{2}\cos(2\pi y)$, $\sqrt{2}\sin(2\pi y)$, and $\sqrt{2}\cos(2\pi(x+y))$. The generator coordinates and basis are not supplied to MPDM; its chart autoencoder is trained only from training coefficient images.

Condition shift and data. Each forcing is

$$f_j(\boldsymbol{x}) = 1 + 0.8\alpha_j \cos(2\pi \boldsymbol{k}_j \cdot \boldsymbol{x} + \varphi_j), \tag{22}$$

with $\alpha_j \sim \mathcal{U}[0.5, 1]$ and $\varphi_j \sim \mathcal{U}[0, 2\pi]$ during both training and testing. Training frequencies are $(1,0), (0,1), (1,1)$ and OOD frequencies are $(2,0), (0,2), (2,2)$. Paired ID/OOD test samples share coefficients, amplitudes, and phases; only frequency support changes. We use 1024 training examples, 256 validation examples, and 1024 examples for each ID/OOD test split. Results are means and sample standard deviations over three optimization seeds sharing the fixed data split.

Models. All methods use the manuscript's x_0-prediction formulation and common timestep-conditioned backbone wherever their definitions permit. DM-Cond and PIDM-Cond concatenate prescribed forcing images with the noisy stack. $\text{MPDM}_{\text{field}}$ predicts the full coefficient image and renders states with the numerical solver; MPDM predicts a learned six-dimensional chart coordinate and uses the same numerical renderer. MPDM_{FNO} retains the learned chart but replaces the numerical renderer with a pretrained FNO. K-DM-Solver diffuses only K, K-LDM-Solver diffuses its learned chart code, and VAE-K-Solver samples a variational coefficient prior; each calls the numerical solver only after generating K. DiffusionPDE and PISD use the matched-backbone adaptations described in the codebase, retaining their inference-time guidance mechanisms; PISD diffuses train-scaled real Fourier coordinates of the joint state–coefficient stack.

Optimization and selection. Diffusion backbones contain three hidden layers of width 128 and a 32-dimensional sinusoidal timestep embedding. We use Adam with learning rate 2×10^{-4}, batch size 32, 3000 diffusion updates, and 50 cosine-schedule DDPM steps. The coefficient chart and FNO are pretrained for 3000 updates; the VAE is trained for 6000 updates. The chart decoder is frozen during diffusion training.

For PIDM and PIDM-Cond, the common grid is

$$\lambda \in \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}. \tag{23}$$

Table 5: Detailed noisy-stack OOD denoising results. Mean ± sample standard deviation over three optimization seeds. DiffusionPDE is its unguided-prior diagnostic in this table.

Model	$\mathrm{MSE}_u \downarrow$	$\mathrm{MSE}_K \downarrow$	MAR↓	$\mathrm{ICS}_K \downarrow$	$\mathcal{E}_{\mathrm{den}} \downarrow$
DM	0.0580 ± 0.0006	0.1817 ± 0.0021	0.5774 ± 0.0060	0.2771 ± 0.0024	0.6030 ± 0.0066
PIDM	0.0579 ± 0.0004	0.1816 ± 0.0021	0.5703 ± 0.0062	0.2768 ± 0.0022	0.6027 ± 0.0065
DM-Cond	0.0590 ± 0.0011	0.1834 ± 0.0014	0.5941 ± 0.0061	0.2762 ± 0.0024	0.6091 ± 0.0051
PIDM-Cond	0.0590 ± 0.0011	0.1834 ± 0.0014	0.5941 ± 0.0061	0.2762 ± 0.0024	0.6091 ± 0.0051
$\mathrm{MPDM}_{\mathrm{field}}$	0.0478 ± 0.0003	0.1707 ± 0.0010	2.32×10^{-7}	0.2386 ± 0.0006	0.5598 ± 0.0026
MPDM	0.0480 ± 0.0006	0.1699 ± 0.0013	2.31×10^{-7}	0.2362 ± 0.0014	$\mathbf{0.5576 \pm 0.0036}$
$\mathrm{MPDM}_{\mathrm{FNO}}$	0.0507 ± 0.0015	0.1699 ± 0.0013	0.2520 ± 0.0632	0.2362 ± 0.0013	0.5603 ± 0.0027
DiffusionPDE (prior)	0.0580 ± 0.0006	0.1817 ± 0.0021	0.5774 ± 0.0060	0.2771 ± 0.0024	0.6030 ± 0.0066

Table 6: Detailed common unpaired OOD generation results. Mean ± sample standard deviation over three optimization seeds. DiffusionPDE and PISD use their full guidance mechanisms.

Model	Joint $\mathrm{SW}_1 \downarrow$	K $\mathrm{SW}_1 \downarrow$	MAR↓
DM	0.5402 ± 0.1413	0.3865 ± 0.0675	0.7148 ± 0.0268
PIDM	0.5377 ± 0.1439	0.3850 ± 0.0681	0.7028 ± 0.0093
DM-Cond	0.4813 ± 0.1496	0.4724 ± 0.0772	0.7733 ± 0.0666
PIDM-Cond	0.4813 ± 0.1496	0.4724 ± 0.0772	0.7733 ± 0.0665
$\mathrm{MPDM}_{\mathrm{field}}$	0.3094 ± 0.0988	0.3181 ± 0.0827	2.50×10^{-7}
MPDM	0.3389 ± 0.1283	0.3267 ± 0.0807	2.51×10^{-7}
$\mathrm{MPDM}_{\mathrm{FNO}}$	0.4049 ± 0.1210	0.3376 ± 0.0683	0.2449 ± 0.0621
K-DM + Solver	0.3865 ± 0.1189	0.3462 ± 0.0804	2.63×10^{-7}
K-LDM + Solver	0.3222 ± 0.0506	0.3507 ± 0.0653	2.53×10^{-7}
VAE-K + Solver	$\mathbf{0.0783 \pm 0.0101}$	$\mathbf{0.0900 \pm 0.0089}$	2.40×10^{-7}
DiffusionPDE	0.5395 ± 0.1419	0.3857 ± 0.0679	0.7111 ± 0.0242
PISD	0.4115 ± 0.0751	0.4076 ± 0.0783	0.4385 ± 0.0469

A complete trial is selected using independent ID validation under

$$\mathcal{E}_{\mathrm{den}} = \mathrm{MSE}_u + 3\,\mathrm{MSE}_K\,. \tag{24}$$

No test metric is used for model or guidance selection. The selected PIDM weights across the three seeds are $10^{-3}, 10^{-6}, 10^{-6}$, and the selected PIDM-Cond weights are $10^{-6}, 10^{-6}, 10^{-6}$. Because five of these six selections lie at the smallest tested value, the sweep does not establish that the optimal positive weight has been bracketed. Guidance settings are likewise selected on ID validation and then frozen.

Evaluation. For the common noisy-stack diagnostic, ambient denoisers share the same corrupted joint stacks and timestep/noise draws. We report MSE_u, MSE_K, mean absolute residual

$$\mathrm{MAR} = \mathbb{E}\Big[\mathrm{mean}\,\Big|(L + \mathrm{diag}\,\hat{K})\hat{U} - F\Big|\Big]\,, \tag{25}$$

and inverse consistency

$$\mathrm{ICS}_K = \mathbb{E}\Big[\mathrm{mean}\,\Big|(L + \mathrm{diag}\,\hat{K})U^{\mathrm{gt}} - F\Big|\Big]\,. \tag{26}$$

Native coefficient-only, latent, VAE, and PISD denoising diagnostics use different information or corruption distributions and are therefore not pooled with this ambient MSE comparison.

For common unpaired OOD generation, all twelve models generate 512 samples under the same OOD forcing cohort without clean test measurements. Joint and coefficient sliced Wasserstein-1 distances use 256 common random unit projections and training-only channel scales. Independently generated samples are not assigned paired reconstruction MSE.

Interpretation. The controlled comparison isolates three conclusions. First, solver-rendered MPDM variants substantially reduce ambient OOD denoising error relative to full-stack DM/PIDM and explicitly conditioned counterparts. Second, MPDM and $\mathrm{MPDM}_{\mathrm{FNO}}$ have nearly identical coefficient error, yet their residuals differ by roughly six orders of magnitude, directly isolating the

benefit of numerical rendering for parameter–state consistency. Third, the result is task-specific: VAE-K-Solver is the strongest unconditional generator under the chosen toy coefficient law, demonstrating that final solver consistency alone does not imply that MPDM must dominate a sufficiently accurate coefficient prior followed by one numerical solve.

B Definitions of Evaluation Metrics

B.1 Inverse Consistency Score

We define

$$r_K = \mathcal{Q}\big(p^{\mathrm{gt}}(, u^{\mathrm{gt}}), \hat{K}\big), \qquad \mathrm{ICS}_K = \mathbb{E}\big[\mathrm{mean}(|r_K|)\big], \tag{27}$$

where $\mathcal{Q}$ is the discrete residual operator, $\mathrm{mean}(\cdot)$ averages over all entries of r_K, and $\mathbb{E}[\cdot]$ averages over samples, $(\cdot)^{\mathrm{gt}}$ denotes the ground-truth quantity.

B.2 Relative Memory Overhead

We define the relative memory overhead during training as

$$\mathrm{RM}_{\mathrm{train}} = \frac{\mathrm{Mem}^{\mathrm{peak}}_{\mathrm{train}}(M)}{\mathrm{Mem}^{\mathrm{peak}}_{\mathrm{train}}(\mathrm{DM})}, \tag{28}$$

and the relative memory overhead during inference as

$$\mathrm{RM}_{\mathrm{infer}} = \frac{\mathrm{Mem}^{\mathrm{peak}}_{\mathrm{infer}}(M)}{\mathrm{Mem}^{\mathrm{peak}}_{\mathrm{infer}}(\mathrm{DM})}. \tag{29}$$

Here, M denotes the evaluated model, DM denotes the baseline DM model, and $\mathrm{Mem}^{\mathrm{peak}}_{\mathrm{train}}(\cdot)$ and $\mathrm{Mem}^{\mathrm{peak}}_{\mathrm{infer}}(\cdot)$ denote the peak GPU memory consumption, measured in GB, during training and inference, respectively.

C Theoretical Properties of MPDM

This section formalizes three structural properties of MPDM: the geometry of the solver-rendered parameterization, parameter–state consistency induced by the embedded numerical solver, and the propagation of coordinate-space prediction errors to the complete physical-field stack. These results characterize the architecture under explicit regularity assumptions; in particular, they do not assume that an arbitrary learned decoder is automatically an injective manifold parameterization.

For a prescribed physical condition set $\mathcal{E}$, let $\mathcal{Z} \subseteq \mathbb{R}^{d_z}$ denote the latent coordinate space, $\mathcal{K}$ the admissible parameter space, and $\mathcal{P}$ the space of dependent state fields. Recall the learned decoder $\mathcal{D}_\phi : \mathcal{Z} \to \mathcal{K}$ and the forward solution operator $\mathcal{R}_\mathcal{E} : \mathcal{K} \to \mathcal{P}$. Define the operator graph map

$$\mathcal{G}_\mathcal{E}(K) = \big[\mathcal{R}_\mathcal{E}(K), K\big], \tag{30}$$

and the MPDM rendered mapping

$$\Phi_\mathcal{E}(z) = \mathcal{G}_\mathcal{E}(\mathcal{D}_\phi(z)) = \left[\mathcal{R}_\mathcal{E}(\mathcal{D}_\phi(z)), \mathcal{D}_\phi(z)\right]. \tag{31}$$

The corresponding learned rendered set is

$$\widetilde{\mathcal{M}}_\mathcal{E} = \{\Phi_\mathcal{E}(z) : z \in \mathcal{Z}\}. \tag{32}$$

Unless stated otherwise, all norms below are Euclidean norms after vectorizing the spatial fields, and the joint parameter–state space is equipped with the corresponding product norm.

C.1 Geometry of the Rendered Parameterization

The first result formalizes when the latent representation defines a genuine low-dimensional manifold in the complete parameter–state space.

Proposition 1 (Geometry of the rendered parameterization). Assume that $\mathcal{Z}$ is a d_z-dimensional smooth manifold, $\mathcal{D}_\phi : \mathcal{Z} \to \mathcal{K}$ is a continuously differentiable embedding, and $\mathcal{R}_\mathcal{E} : \mathcal{K} \to \mathcal{P}$ is continuously differentiable on $\mathcal{D}_\phi(\mathcal{Z})$. Then $\Phi_\mathcal{E}$ is a continuously differentiable embedding. Consequently, $\widetilde{\mathcal{M}}_\mathcal{E}$ is a d_z-dimensional embedded manifold contained in the operator-induced set $\mathcal{M}_\mathcal{E}$.

Proof. Consider the graph map $\mathcal{G}_\mathcal{E} : K \mapsto [\mathcal{R}_\mathcal{E}(K), K]$. Because $\mathcal{R}_\mathcal{E}$ is continuously differentiable, $\mathcal{G}_\mathcal{E}$ is continuously differentiable. Moreover, $\mathcal{G}_\mathcal{E}$ is injective because its second component is exactly K. Its inverse on its image is the projection onto the parameter component,

$$\pi_K\left([\mathcal{R}_\mathcal{E}(K), K]\right) = K, \tag{33}$$

which is continuous. Hence $\mathcal{G}_\mathcal{E}$ is an embedding on the relevant parameter set.

Since $\mathcal{D}_\phi$ is also an embedding, the composition

$$\Phi_\mathcal{E} = \mathcal{G}_\mathcal{E} \circ \mathcal{D}_\phi \tag{34}$$

is an embedding. Therefore its image $\widetilde{\mathcal{M}}_\mathcal{E}$ is a d_z-dimensional embedded manifold.

Finally, $\mathcal{D}_\phi(z) \in \mathcal{K}$ for every $z \in \mathcal{Z}$, and therefore

$$\Phi_\mathcal{E}(z) = [\mathcal{R}_\mathcal{E}(\mathcal{D}_\phi(z)), \mathcal{D}_\phi(z)] \in \mathcal{M}_\mathcal{E}. \tag{35}$$

Thus $\widetilde{\mathcal{M}}_\mathcal{E} \subseteq \mathcal{M}_\mathcal{E}$. □

Remark. The embedding assumption is a sufficient regularity condition and is not automatically guaranteed by an unconstrained neural decoder. Proposition 1 should therefore be interpreted as characterizing the geometry induced by MPDM when the learned decoder is locally non-degenerate and globally injective on the latent region of interest.

C.2 Solver-Consistency by Construction

We next distinguish the abstract forward solution operator from its numerical realization. Let $\widetilde{\mathcal{R}}_\mathcal{E}$ denote the numerical renderer used in MPDM, and let $\mathcal{F}^h_\mathcal{E}(p, K)$ denote the residual of the discretized governing equations, where h represents the underlying spatial discretization.

Proposition 2 (Discrete solver consistency). Assume that the numerical renderer satisfies

$$\left\| \mathcal{F}^h_\mathcal{E}\left(\widetilde{\mathcal{R}}_\mathcal{E}(K), K\right) \right\| \le \tau_\mathcal{E} \qquad \text{for all } K \in \mathcal{K}, \tag{36}$$

where $\tau_\mathcal{E} \ge 0$ characterizes the numerical solve accuracy. Then every MPDM clean prediction

$$\hat{x}_0 = \left[\hat{p}_{1:m}, \hat{K}\right] = \left[\widetilde{\mathcal{R}}_\mathcal{E}(\hat{K}), \hat{K}\right] \tag{37}$$

satisfies

$$\left\| \mathcal{F}^h_\mathcal{E}\left(\hat{p}_{1:m}, \hat{K}\right) \right\| \le \tau_\mathcal{E}. \tag{38}$$

Proof. By construction of the MPDM clean prediction,

$$\hat{p}_{1:m} = \widetilde{\mathcal{R}}_\mathcal{E}(\hat{K}). \tag{39}$$

Substituting this identity into Equation (36) with $K = \hat{K}$ directly gives Equation (38). □

Remark. For an exact solution of the discrete governing equations, $\tau_\mathcal{E} = 0$. With an iterative numerical renderer such as the unrolled CG solver used in MPDM, $\tau_\mathcal{E}$ reflects the achieved numerical solve accuracy. This result concerns consistency with the embedded *discretized* physical model and does not imply zero discretization error with respect to the underlying continuous PDE.

C.3 Stability of Coordinate-Space Rendering

The final result characterizes how perturbations in the compact coordinate representation propagate through the decoder and physical renderer.

Proposition 3 (Stability of the rendered mapping). Assume that the decoder is L_D-Lipschitz,

$$\|\mathcal{D}_\phi(z_1) - \mathcal{D}_\phi(z_2)\| \le L_D \|z_1 - z_2\|, \tag{40}$$

and that, for a fixed condition set $\mathcal{E}$, the forward operator is $L_R(\mathcal{E})$-Lipschitz on the relevant parameter set:

$$\|\mathcal{R}_\mathcal{E}(K_1) - \mathcal{R}_\mathcal{E}(K_2)\| \le L_R(\mathcal{E}) \|K_1 - K_2\|. \tag{41}$$

Then the rendered mapping $\Phi_\mathcal{E}$ satisfies

$$\|\Phi_\mathcal{E}(z_1) - \Phi_\mathcal{E}(z_2)\| \le L_D \sqrt{1 + L_R(\mathcal{E})^2}\, \|z_1 - z_2\|. \tag{42}$$

Proof. Using the product-space norm and Equation (31),

$$\begin{aligned}
&\|\Phi_\mathcal{E}(z_1) - \Phi_\mathcal{E}(z_2)\|^2 \\
&= \|\mathcal{R}_\mathcal{E}(\mathcal{D}_\phi(z_1)) - \mathcal{R}_\mathcal{E}(\mathcal{D}_\phi(z_2))\|^2 \\
&\quad + \|\mathcal{D}_\phi(z_1) - \mathcal{D}_\phi(z_2)\|^2 \\
&\le \left(1 + L_R(\mathcal{E})^2\right) \|\mathcal{D}_\phi(z_1) - \mathcal{D}_\phi(z_2)\|^2 \\
&\le L_D^2 \left(1 + L_R(\mathcal{E})^2\right) \|z_1 - z_2\|^2.
\end{aligned} \tag{43}$$

Taking square roots proves Equation (42). □

Proposition 3 characterizes deterministic error propagation through the rendered parameterization. It does not imply that coordinate-space estimation is statistically superior to direct field-space estimation; rather, it shows that coordinate prediction errors are controlled by the sensitivities of the decoder and physical renderer.

Corollary 1 (Full-stack error decomposition). Let

$$x_0^\star = \mathcal{G}_\mathcal{E}(K^\star) = [\mathcal{R}_\mathcal{E}(K^\star), K^\star] \tag{44}$$

be a target physical-field stack. For any reference coordinate $z^\star \in \mathcal{Z}$ and an MPDM prediction $\hat{z}$, we have

$$\begin{aligned}
\|\Phi_\mathcal{E}(\hat{z}) - x_0^\star\| \le \sqrt{1 + L_R(\mathcal{E})^2}\Big(& L_D \|\hat{z} - z^\star\| \\
& + \|\mathcal{D}_\phi(z^\star) - K^\star\| \Big).
\end{aligned} \tag{45}$$

Proof. By the triangle inequality,

$$\begin{aligned}
\|\Phi_\mathcal{E}(\hat{z}) - \mathcal{G}_\mathcal{E}(K^\star)\| \le & \|\Phi_\mathcal{E}(\hat{z}) - \Phi_\mathcal{E}(z^\star)\| \\
& + \|\mathcal{G}_\mathcal{E}(\mathcal{D}_\phi(z^\star)) - \mathcal{G}_\mathcal{E}(K^\star)\|.
\end{aligned} \tag{46}$$

The first term is bounded by Proposition 3. For the second term, Equation (41) gives

$$\|\mathcal{G}_\mathcal{E}(K_1) - \mathcal{G}_\mathcal{E}(K_2)\| \le \sqrt{1 + L_R(\mathcal{E})^2}\, \|K_1 - K_2\|. \tag{47}$$

Substituting $K_1 = \mathcal{D}_\phi(z^\star)$ and $K_2 = K^\star$ yields Equation (45). □

The two terms in Equation (45) have distinct meanings. The first measures error in estimating the latent coordinate, while the second measures the representation error of the learned decoder relative to the target parameter field. If the target is exactly representable by the decoder, i.e., $K^\star = \mathcal{D}_\phi(z^\star)$, the representation term vanishes and

$$\|\Phi_\mathcal{E}(\hat{z}) - x_0^\star\| \le L_D \sqrt{1 + L_R(\mathcal{E})^2}\, \|\hat{z} - z^\star\|. \tag{48}$$

To additionally account for numerical rendering, suppose that $\widetilde{\mathcal{R}}_\mathcal{E}$ satisfies

$$\sup_{K \in \mathcal{K}} \left\| \widetilde{\mathcal{R}}_\mathcal{E}(K) - \mathcal{R}_\mathcal{E}(K) \right\| \le \delta_\mathcal{E}. \tag{49}$$

Defining

$$\widetilde{\Phi}_\mathcal{E}(z) = \left[\widetilde{\mathcal{R}}_\mathcal{E}(\mathcal{D}_\phi(z)), \mathcal{D}_\phi(z) \right], \tag{50}$$

we obtain

$$\left\|\widetilde{\Phi}_{\mathcal{E}}(\hat{z}) - x_0^\star\right\| \le \delta_{\mathcal{E}} + \sqrt{1 + L_R(\mathcal{E})^2}\Big(L_D \left\|\hat{z} - z^\star\right\| + \left\|\mathcal{D}_\phi(z^\star) - K^\star\right\|\Big). \tag{51}$$

Thus, the total full-stack error can be decomposed into coordinate-estimation error, decoder representation error, and numerical rendering error.

If $L_R^{\max} = \sup_{\mathcal{E}\in\mathfrak{E}} L_R(\mathcal{E}) < \infty$ over a family of physical conditions $\mathfrak{E}$, the same bounds hold uniformly over $\mathfrak{E}$ after replacing $L_R(\mathcal{E})$ by $L_R^{\max}$. This provides a stability characterization across condition changes, but does not by itself imply statistical OOD generalization of the learned latent predictor.

Scope of the analysis. The results above characterize the geometry of the MPDM parameterization, consistency with the embedded discrete solver, and deterministic propagation of latent and numerical errors. They do not constitute a convergence guarantee for the DDPM reverse process, nor do they imply that coordinate-space prediction is universally more statistically efficient than direct field-space prediction.

D Manifold Projection

We present a diffusion experiment designed to isolate the effect of manifold projection in a controlled abstract setting. The purpose of this study is not to reproduce the full MPDM framework, but to compare three architectures that differ only in how the clean prediction is parameterized: a standard ambient-space DM, a penalty-based PIDM, and a manifold-projected model using $\mathrm{MPDM}_{\mathrm{field}}$. We intentionally adopt the field-space variant rather than the coordinate-space variant so that the experiment isolates the contribution of manifold projection itself, without conflating it with the separate benefit of low-dimensional parameterization.

D.1 Problem Formulation

Let $s \in [0, 1]$ denote the spatial coordinate on a one-dimensional grid of N sample points. Each sample is a multi-channel stack

$$x = [p_{\mathrm{obs}}, p_1, \ldots, p_h, K] \in \mathbb{R}^{(h+2)\times N}, \tag{52}$$

where p_{obs} is an observed state channel, $p_j{}_{j=1}^{h}$ are hidden state channels, and K is a latent field. The stack is generated through a multiplicative renderer

$$p_j(s) = E_j(s)\, K(s), \tag{53}$$

where $E_j(s)$ denotes the excitation associated with channel j.

The observed channel p_{obs} is generated by an excitation that is held fixed between training and testing. By contrast, the hidden channels are rendered using one excitation family during training and a higher-frequency version of the same smooth family during testing. This creates a pure OOD hidden rendering shift: the observed channel remains in-distribution, while the hidden channels are governed by a shifted test-time operator. In the default implementation, the full stack is noised and provided to the denoiser, and the model predicts the clean stack from the noisy input and the timestep; the earlier observed-only masking protocol is retained only as an optional ablation. Consequently, successful recovery of the hidden stack must proceed under the structural assumptions encoded by the model rather than through any architectural advantage specific to observed-only masking.

For notational clarity, let $\hat{K}$ denote the predicted shared field, let $\hat{p}_j$ denote the predicted state channels, let

$$\hat{p} = [\hat{p}_{\mathrm{obs}}, \hat{p}_1, \ldots, \hat{p}_h], \tag{54}$$

and let

$$\hat{x}_0 = [\hat{p}_{\mathrm{obs}}, \hat{p}_1, \ldots, \hat{p}_h, \hat{K}] \tag{55}$$

denote the final clean estimate. We further define the deterministic rendering operator

$$\mathcal{R}(K; \mathcal{E}) = [E_0 K, E_1 K, \ldots, E_h K], \tag{56}$$

where E_0 corresponds to the observed channel and $\mathcal{E} = \{E_0, E_1, \ldots, E_h\}$ is the excitation family.

D.2 Compared Models

The three methods differ only in how the denoiser produces $\hat{x}_0$.

DM predicts the full clean stack directly in ambient space,

$$\hat{x}_0 = g_\theta(x_t, t), \tag{57}$$

where x_t is the noisy input at diffusion step t and g_θ is a timestep-conditioned denoiser. In this formulation, all state channels and the field are free outputs of the network. PIDM retains the same ambient-space parameterization, but augments the training objective with a residual penalty that discourages violation of the rendering relation. Writing

$$r_j(s) = \hat{p}_j(s) - E_j(s)\hat{K}(s), \tag{58}$$

the PIDM objective adds a penalty proportional to $\sum_{j,s} |r_j(s)|^2$. The rendering law is therefore encouraged during training, but not enforced structurally.

By contrast, $\mathrm{MPDM}_{\mathrm{field}}$ changes the architecture itself. Rather than predicting the full stack, the denoiser predicts only the field,

$$\hat{K} = h_\theta(x_t, t), \tag{59}$$

and all state channels are then generated deterministically by the renderer,

$$\hat{p}_j(s) = E_j(s)\hat{K}(s), \qquad j = 0, \ldots, h. \tag{60}$$

The final estimate is assembled as equation 55. In implementation, the backbone outputs logits for the field, which are mapped into the admissible interval $[K_{\min}, K_{\max}]$ by a sigmoid transform, where $K_{\min}$ and $K_{\max}$ denote the prescribed lower and upper bounds of the field. The key point is that the hidden channels are no longer predicted independently; they are rendered from $\hat{K}$ by construction. This is the manifold projection mechanism that the experiment is intended to isolate.

D.3 Conditioning Setup and Evaluation Metrics

We adopt a diffusion-style conditioning protocol in which the full stack is noised and the model predicts the clean stack $\hat{x}_0$ from the noisy input and the timestep. This default setting is intentionally closer to the ambient x_0-prediction formulation used in the Darcy experiments than the earlier observed-only masking protocol. It therefore provides a cleaner architectural comparison between direct ambient prediction, soft residual regularization, and manifold projection under an OOD hidden rendering shift.

We evaluate the models using the mean squared error on the hidden state channels,

$$\mathrm{MSE}_{p,\mathrm{hidden}} = \frac{1}{hN} \sum_{j=1}^{h} \left\| \hat{p}_j - p_j^{\mathrm{gt}} \right\|_2^2, \tag{61}$$

where p_j^{gt} denotes the ground-truth hidden channel. To further assess whether the predicted hidden channels satisfy the rendering structure at test time, we additionally report the rendering consistency error (RCE),

$$\mathrm{RCE} = \frac{1}{hN} \sum_{j=1}^{h} \left\| \hat{p}_j - E_j^{\mathrm{test}} \hat{K} \right\|_1, \tag{62}$$

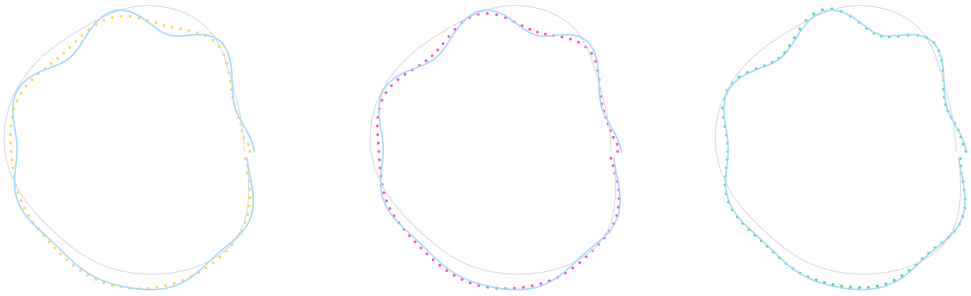
where E_j^{test} denotes the test-time excitation associated with hidden channel j. This quantity measures whether the predicted hidden stack is consistent with the test-time renderer implied by the predicted field. In this setting, RCE is a structural consistency diagnostic rather than a PDE residual, and a value of zero indicates exact agreement with the rendered manifold.

D.4 Quantitative Results under OOD Shift

As shown in Table 7, the quantitative results exhibit a clear ordering among ambient prediction, soft regularization, and manifold projection under the OOD hidden rendering shift. DM attains the largest hidden-state error and the largest renderer-consistency deviation. PIDM improves both

Table 7: Quantitative results for the projection experiment.

Method	$\mathrm{MSE}_{p,\mathrm{hidden}} \downarrow$	$\mathrm{RCE} \downarrow$
DM	0.0112	0.0775
PIDM	0.0056	0.0530
$\mathrm{MPDM}_{\mathrm{field}}$	**0.0022**	**0.0000**



(a) DM (b) PIDM (c) $\mathrm{MPDM}_{\mathrm{field}}$

Figure 5: OOD hidden-state predictions. Yellow, magenta, and green points show $\hat{p}_j$ for DM, PIDM, and $\mathrm{MPDM}_{\mathrm{field}}$; the gray curve is the observed input channel, and the blue curve is the OOD hidden-state ground truth.

quantities, which indicates that residual-based regularization is beneficial under distribution shift. However, the remaining mismatch shows that penalization alone does not reliably enforce the hidden rendering relation under the shifted test-time excitation family. By contrast, $\mathrm{MPDM}_{\mathrm{field}}$ achieves the lowest hidden-state error, while attaining $\mathrm{RCE} = 0$. Thus, the strongest improvement is observed not merely in denoising accuracy, but in reconstruction of the hidden OOD channels together with exact structural consistency with the renderer.

This pattern supports the intended architectural interpretation of the experiment. DM and PIDM both predict the hidden channels directly in ambient space, so consistency with the test-time renderer is only implicit in the learned mapping; for PIDM, the residual term encourages such consistency during training, but the hidden channels remain unconstrained outputs of the denoiser. In $\mathrm{MPDM}_{\mathrm{field}}$, by contrast, the model predicts only $\hat{K}$, and the state channels are generated deterministically as

$$\hat{p} = \mathcal{R}(\hat{K}; \mathcal{E}_{\mathrm{test}}). \tag{63}$$

The resulting performance gain is therefore most naturally attributed to projection onto the rendered manifold rather than to improved direct regression of the hidden state stack.

D.5 Visualization-Based Analysis

The overlay visualizations provide a geometric complement to the quantitative results by showing how each model reconstructs an OOD hidden state relative to both the observed input and the target hidden response. This visualization is not a test of ordinary reconstruction quality alone; rather, it probes whether the denoiser can recover a hidden channel governed by a shifted rendering law while remaining consistent with the latent field structure.

Under this interpretation, the three methods exhibit a clear qualitative ordering. The DM prediction remains displaced from the blue target over extended portions of the contour, indicating that unconstrained ambient prediction does not fully recover the hidden OOD geometry. PIDM reduces these deviations and tracks the target more closely than DM, which is consistent with its lower hidden-state error and improved consistency score. However, noticeable discrepancies remain in regions where the hidden target departs most from the observed geometry, showing that residual penalization improves alignment without fully enforcing the rendered-manifold structure. This qualitative

behavior matches the reported ordering in $\mathrm{MSE}_{p,\mathrm{hidden}}$ and RCE, where PIDM improves over DM but remains inferior to manifold projection.

By contrast, $\mathrm{MPDM}_{\mathrm{field}}$ exhibits the closest visual agreement with the blue curve across the full contour. This behavior is a direct consequence of the architecture: the model first predicts $\hat{K}$ and then renders the hidden channels deterministically from that field, rather than predicting the hidden channels as independent ambient outputs. As a result, once the field estimate is fixed, the hidden-state prediction is constrained to lie on the test-time rendered manifold. The visual alignment of the green points with the blue target is therefore fully consistent with the zero renderer-consistency error reported in the quantitative evaluation.

Taken together, the figures sharpen the interpretation of the experiment. All methods are given the same input format, but they differ in whether hidden-channel recovery is treated as direct ambient prediction, soft regularization, or deterministic rendering from a predicted field. The overlays show that DM is least successful at matching the hidden OOD target, PIDM provides a partial correction, and $\mathrm{MPDM}_{\mathrm{field}}$ most faithfully recovers the target hidden geometry. In this sense, the visual results provide a geometric manifestation of manifold projection: the main benefit is not merely reduced pointwise error, but a stronger tendency to recover the correct hidden structure under distribution shift.

D.6 Discussion

Taken together, the quantitative and visual results provide a clean ablation of the hard-projection principle. The setting is intentionally designed so that the observed channel remains in-distribution, while the hidden channels undergo an OOD rendering shift. Under these conditions, success depends on whether the model can recover a field and reconstruct hidden channels that remain consistent with the shifted rendering law. Ambient full-stack prediction performs worst, and residual penalization provides a clear but incomplete improvement. By contrast, $\mathrm{MPDM}_{\mathrm{field}}$ achieves the lowest hidden-state reconstruction error and exactly zero renderer-consistency error by construction.

The conclusion of this study is therefore clear. When valid samples lie on a known rendered manifold and the main challenge is to reconstruct hidden channels under distribution shift, it is advantageous to predict the independent field and generate the dependent channels through a deterministic projection layer, rather than to predict the full stack directly in ambient space and regularize it afterward. In this sense, the experiment isolates the essential architectural advantage of manifold projection that motivates the broader MPDM framework.

E Field–Coordinate Parameterization

We present an experiment designed to compare different spatial parameterizations for the regression task: direct prediction of the full field K in ambient space versus prediction of a low-dimensional intrinsic coordinate z followed by deterministic decoding. The experiment is constructed so that the comparison is as controlled as possible: both estimators use the same model class, namely ridge regression, and differ only in the variable they are asked to predict. As a result, any performance difference can be attributed to the choice of parameterization rather than to model capacity or optimization details.

E.1 Problem Formulation

Let $z \in \mathbb{R}^{d_z}$ denote the intrinsic coordinate vector, where $d_z = 6$, and let $K \in \mathbb{R}^N$ denote the one-dimensional field to be reconstructed, where $N = 240$ is the number of spatial sample points. The field is not generated from a simple linear family. Instead, it is produced through a nonlinear decoder

$$K = D(z), \tag{64}$$

where D combines several latent-dependent Gaussian-like bumps, an oscillatory component with latent-dependent frequency and phase, and a low-order trend term, followed by a bounded sigmoid transform that maps the field into the admissible interval $[0.10,\ 1.50]$. Thus, admissible fields lie on a nonlinear low-dimensional manifold embedded in the ambient field space.

The model does not observe K directly. Instead, it receives a low-dimensional sensor code

$$y = Mz + \varepsilon, \tag{65}$$

where $y \in \mathbb{R}^{d_z}$ is the observation, $M \in \mathbb{R}^{d_z \times d_z}$ is a fixed mixing matrix, and ε is additive Gaussian noise. This construction makes the central modeling question especially transparent, namely whether one should learn the map $y \mapsto K$ directly or first estimate the intrinsic coordinate z and then recover K through the decoder D.

To evaluate generalization, the train and test distributions differ in latent space. During training, the latent variable z is sampled from a relatively narrow Gaussian distribution. During testing, z is drawn from a broader shifted distribution with offsets applied to selected latent dimensions. This produces an extrapolation regime in which the alignment between the estimator and the intrinsic data-generating structure becomes especially important.

E.2 Compared Estimators

Two estimators are compared.

The field-space baseline directly regresses the ambient field from the observation,

$$\hat{K}_{\text{field}} = f_{\text{field}}(y), \tag{66}$$

where $\hat{K}_{\text{field}}$ denotes the predicted field. In the implementation, f_{field} is a ridge regressor fit from y to K.

The coordinate-space estimator instead regresses the intrinsic coordinate,

$$\hat{z} = f_{\text{coord}}(y), \qquad \hat{K}_{\text{coord}} = D(\hat{z}), \tag{67}$$

where $\hat{z}$ denotes the predicted latent coordinate and $\hat{K}_{\text{coord}}$ the decoded field. This estimator is also implemented using ridge regression, with the same regularization strength as the field-space baseline. Hence, the only substantive difference between the two methods is whether the regression target is the ambient field K or the intrinsic coordinate z.

This comparison isolates the essential parameterization question. In the field-space model, the regressor must implicitly learn both the inverse mapping from y to the latent structure and the nonlinear field geometry induced by D. In the coordinate-space model, these two roles are separated cleanly: the learned model recovers $\hat{z}$, and the decoder D supplies the nonlinear map into field space.

E.3 Evaluation Metrics and Notation

Performance is reported using two complementary metrics. The first is the field reconstruction error MSE_K. The second is a curvature-sensitive error based on discrete second differences. Writing

$$\Delta^2 K_i = K_{i+1} - 2K_i + K_{i-1}, \tag{68}$$

the curvature metric is defined as

$$\text{MSE}_{\text{curv}} = \frac{1}{N-2} \sum_{i=2}^{N-1} \left(\Delta^2 \hat{K}_i - \Delta^2 K_i \right)^2. \tag{69}$$

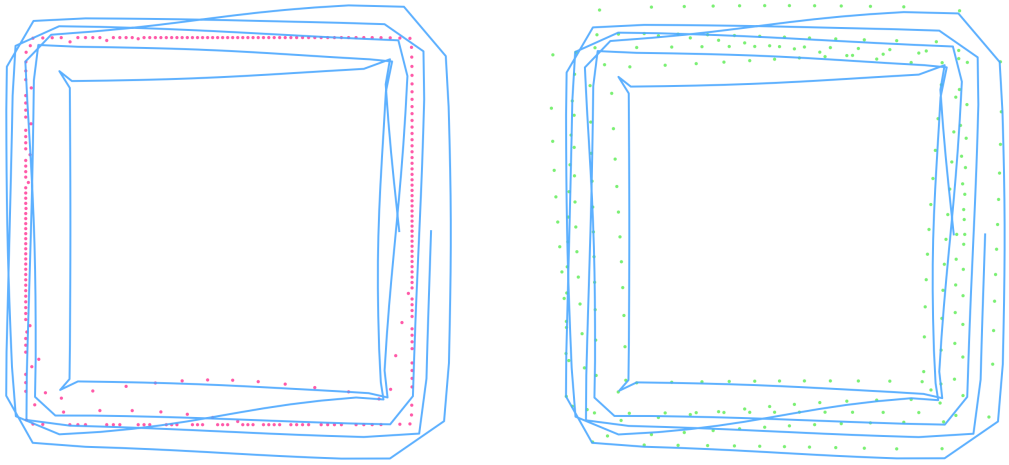
This quantity measures whether the predicted field reproduces the local geometric structure of the target signal rather than merely its pointwise magnitude. It is particularly useful here because nonlinear decoder errors often appear first as shape distortions.

E.4 Quantitative Results

The quantitative results show a strong advantage for coordinate-space prediction. These improvements support the intended interpretation of the toy problem. Because the observation y is generated directly from the intrinsic coordinate z, the inverse problem is fundamentally low-dimensional. Directly regressing the full field K therefore misaligns the estimator with the structure of the data-generating process: the field-space model must approximate, in one step, both latent-coordinate recovery and nonlinear field synthesis. By contrast, the coordinate-space estimator only learns the simpler map $y \mapsto z$, after which the decoder reconstructs the nonlinear geometry of admissible fields. The resulting gain is especially pronounced under the shifted test distribution, where extrapolation in latent space becomes more difficult for an ambient-space regressor.

Table 8: Quantitative results for the parameterization experiment.

Method	$\mathrm{MSE}_K \downarrow$	$\mathrm{MSE}_{\mathrm{curv}} \downarrow$
Field-space	0.5129	9.7197e-05
Coordinate-space	**0.0154**	**3.6059e-06**



(a) Field-space (b) Coordinate-space

Figure 6: Comparison of field-space and coordinate-space field predictions. Magenta and green points show $\hat{K}_{\mathrm{field}}$ and $\hat{K}_{\mathrm{coord}} = D(\hat{z})$, respectively; the blue curve is the ground-truth field K in square-coil coordinates.

E.5 Visualization-Based Analysis

The overlay visualizations provide a geometric interpretation of this difference. Importantly, the blue reference curve does not represent multiple distinct ground-truth signals. Rather, the code maps a single one-dimensional field K into a five-turn square-coil embedding for visualization. Specifically, the field is first placed along an outward spiral, then squared off so that the spiral becomes a sequence of nested square-like turns, and finally perturbed in the local normal direction according to the signal amplitude. Consequently, the blue figure appears as many box-like shapes of varying size, but it is in fact one continuous visual embedding of a single field.

This visualization clarifies the observed qualitative difference. The green points are distributed near essentially all square-like turns of the blue reference curve because the coordinate-space estimator recovers the latent coordinate $\hat{z}$ and then applies the exact decoder D. As a result, it reconstructs the overall field geometry across the full signal trajectory, not merely a local portion of it. The close alignment of the green points with both the outer and inner turns is therefore a visual manifestation of the low values of MSE_K and $\mathrm{MSE}_{\mathrm{curv}}$.

By contrast, the pink points appear concentrated near only part of the square-coil structure. This occurs because the field-space baseline is asked to learn the more difficult direct mapping $y \mapsto K$ using a linear ridge regressor, even though K is a nonlinear function of z. Under the shifted test distribution, this direct mapping is unable to recover the full nonlinear field geometry reliably. Instead, it tends to approximate only a partial or averaged portion of the target structure, which in the visualization appears as points clustering around one particular square-like turn rather than tracing the full blue trajectory. For this reason, the concentration should be understood not as an arbitrary artifact of plotting, but as a representative failure mode of direct field-space regression under the selected extrapolation setting.

E.6 DISCUSSION

Taken together, the quantitative and visual results show that the benefit of coordinate-space parameterization can be isolated even in a non-diffusion setting. When the underlying generative structure is

$$z \mapsto D(z), \tag{70}$$

and when the observation is naturally tied to z rather than directly to the full field K, it is inefficient to ask a model to regress K directly. The field-space baseline must approximate the latent inversion and the nonlinear decoding simultaneously in ambient space. The coordinate-space estimator, by contrast, separates these two roles cleanly: the learned model recovers $\hat{z}$, and the exact decoder produces $\hat{K}$.

The order-of-magnitude gains in both field error and curvature error therefore indicate that coordinate-space prediction is not merely a cosmetic reformulation. It is a materially better alignment between the estimator and the intrinsic structure of the problem. In this sense, the experiment serves as a direct conceptual analogue of the transition from $\mathrm{MPDM}_{\mathrm{field}}$ to MPDM in the full method: it demonstrates, independently of diffusion and independently of manifold projection, why predicting intrinsic coordinates can be preferable to predicting full parameter fields in ambient space whenever valid fields lie on a nonlinear low-dimensional manifold.