VIDEOPDE: UNIFIED GENERATIVE PDE SOLVING VIA VIDEO INPAINTING DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce a unified generative framework for solving partial differential equations (PDEs) and quantifying predictive uncertainty across forward, inverse, and partial-observation tasks. In contrast to prior approaches that design separate strategies for each setting, we recast PDE solving as a generalized video inpainting problem, where future or missing spatiotemporal states are inferred from arbitrary patterns of observed data. Our method employs a pixel-space transformer diffusion model that directly operates on physical fields, avoiding the accuracy degradation observed with latent-space representations in scientific domains. To enhance efficiency, we incorporate a hierarchical transformer strategy that balances resolution, fidelity, and computational cost. This design enables fine-grained, high-quality reconstructions together with per-pixel uncertainty estimates that capture spatial and temporal variability. Extensive experiments on five representative synthetic PDE benchmarks and a real-world ERA5 dataset demonstrate that our framework consistently outperforms state-of-the-art baselines, offering a versatile and robust approach to scientific and engineering applications.

1 INTRODUCTION

Computational science communities have proposed numerous learning-based approaches for solving PDE-governed systems for simulation, optimization, and scientific discovery. These methods provide trade-offs across accuracy, applicability, and speed. Physics-informed neural networks (PINNs) (57; 58) flexibly handle forward or inverse predictions from sparse differential measurements, but often converge to poor local minima, sacrificing accuracy. Neural operators (40; 42; 45) offer fast approximate solvers, yet struggle under the partial observations typical in real-world scenarios. Generative methods (8; 65; 82) accommodate incomplete data, but are computationally expensive and have limited ability to model dense temporal evolution. These limitations hinder the practical use of learning-based PDE approaches for real-world state reconstruction and forecasting.

In this work, we introduce VideoPDE, a unified framework for PDE solving that is accurate, efficient, and broadly applicable. Our key insight is to cast diverse PDE tasks as a generative video inpainting problem, where missing spatiotemporal fields are inferred from arbitrary patterns of observed data. For example, forward simulation corresponds to predicting missing frames conditioned on initial states, while partial reconstructions map to inpainting under spatiotemporal sparse observations. This formulation naturally unifies different PDE tasks, supports uncertainty-aware predictions in chaotic systems, and offers a single network that adapts across sensor configurations.

Our framework builds on transformer-based inpainting models (30), but departs from prior work in three crucial ways. First, we employ generative diffusion models rather than regression-based methods, enabling calibrated uncertainty estimates and recovery of high-frequency details. Second, unlike most video diffusion models (VDMs) that operate in latent space (3; 23; 25; 81), we perform pixel-space denoising and conditioning, which is essential for scientific fields where fine-grained accuracy matters more than perceptual realism. Third, we introduce a hierarchical architecture with 3D neighborhood attention that preserves high-resolution structure while improving computational efficiency. Together, these innovations yield accurate, high-fidelity reconstructions and per-pixel uncertainty quantification through multiple generative samples.

Our contributions include:

- **Unified formulation:** We recast forward, inverse, and partial-observation PDE solving as a single generative video inpainting task, enabling one model to adapt seamlessly across diverse scenarios.

- **Architectural innovation:** We design a hierarchical pixel-space diffusion transformer with 3D neighborhood attention, showing that direct pixel-level modeling avoids the accuracy loss of latent approaches and delivers high-fidelity, efficient solutions.
- **Uncertainty and validation:** Our framework naturally provides spatiotemporal uncertainty and achieves state-of-the-art results on five synthetic PDE benchmarks and a real-world ERA5 case study, reducing error by up to an order of magnitude over prior methods.

2 Related Work

Neural PDEs Solving partial differential equations (PDEs) is a fundamental problem in physical sciences. Traditional numerical methods such as the Finite Element Method (56; 67) and Boundary Element Method (1; 32) have long been the backbone of PDE solving but are computationally expensive and inflexible for complex systems. Neural approaches offer data-driven alternatives: physics-informed neural networks (PINNs) (57; 58) enforce PDE constraints via the loss function and have been applied to a wide range of PDEs (5; 6; 20; 24; 47; 49; 54; 72; 73). While PINNs can work on sparse measurements, in practice they often face optimization instability and poor scalability. Neural operators, such as FNO (40), DeepONet (45), and PINO (42), learn mappings between function spaces to avoid expensive optimization and achieve resolution-invariance. These models have been extended to various forward (4; 10; 37; 38; 41; 53; 63; 76) and inverse (43; 50) PDE tasks, but remain limited in flexibility for handling arbitrary and sparse input patterns.

Solving PDEs Under Sparse Measurements Recently, neural methods have gained attention for solving PDEs under sparse measurements, reflecting the challenge of acquiring full spatiotemporal data. DiffusionPDE (31) addresses this by modeling the joint distribution over coefficients and solutions, allowing flexible inputs, but its DPS (12) backbone requires PDE-specific tuning and struggles with dynamic PDEs. Spatially-aware diffusion models (82) use cross-attention to handle partial observations but lack temporal modeling. Temporal PDEs are especially important for modeling nonlinear fluid and gas dynamics (17; 19; 34; 74; 83). Super-resolution frameworks (21; 22; 36) reconstruct full fields from coarse data. Recent methods (39; 64; 65) combine physics-informed losses with diffusion models or transformers for high-fidelity turbulent flow reconstruction. Despite past successes, existing methods often rely on strong assumptions about PDEs, boundary conditions, or sensor layouts. We propose a unified generative framework that requires no prior knowledge and generalizes well across forward, inverse, and partial observation problems.

Inpainting Diffusion Models Diffusion models (29; 66; 69; 71; 78) have emerged as particularly suited for image and video inpainting due to their capability to model complex, high-dimensional distributions effectively. Training-free methods guide the sampling trajectory to satisfy the conditions at inference time through noise inference (48), resampling (46), or latent alignment (11). It can also be studied as a linear inverse problem (12; 14; 35; 68; 77). However, these methods often struggle with extremely sparse or ambiguous observations. Another class of methods directly trains a conditional diffusion model. These methods typically modify the network architecture to inject conditioning information, such as channel concatenation (61; 62), cross-attention (2; 55; 59), or ControlNet (79). We adopt channel concatenation in this work since it is simple and effective. These conditioning techniques have been extended to video diffusion models (30) for video inpainting (44; 75; 80).

3 Methods

3.1 Preliminaries: Diffusion Models and Guided Sampling

Diffusion models learn data distributions by reversing a gradual noising process. Starting from a clean sample $\mathbf{x}_0$ from a data distribution $p(\mathbf{x})$, a forward stochastic process progressively adds noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to produce $\mathbf{x}_t = \mathbf{x}_0 + \sigma(t)\epsilon$ and hence a family of distributions $p(\mathbf{x}_t; \sigma(t))$, where $\sigma(t)$ denotes the standard deviation of the noise at diffusion time t, following the noise schedule of the Elucidating Diffusion Models (EDM) (33) framework we adopt in this work. The goal is to learn the *reverse* process to recover $\mathbf{x}_0$ from $\mathbf{x}_t$ by training a denoising neural network $D_\theta(\mathbf{x}_t, \sigma(t))$ with loss

$$\mathcal{L}_{\text{EDM}} = \mathbb{E}_{\mathbf{x}_0 \sim p(\mathbf{x})} \mathbb{E}_{\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} \left[\| D_\theta(\mathbf{x}_t, \sigma(t)) - \mathbf{x}_0 \|^2 \right] \tag{1}$$

This gives us an estimate of the *score function* (70), a vector field pointing to higher data density,

$$\nabla_{\mathbf{x}} \log p\big(\mathbf{x}; \sigma(t)\big) = (D(\mathbf{x}, \sigma(t)) - \mathbf{x}) / \sigma(t)^2, \tag{2}$$

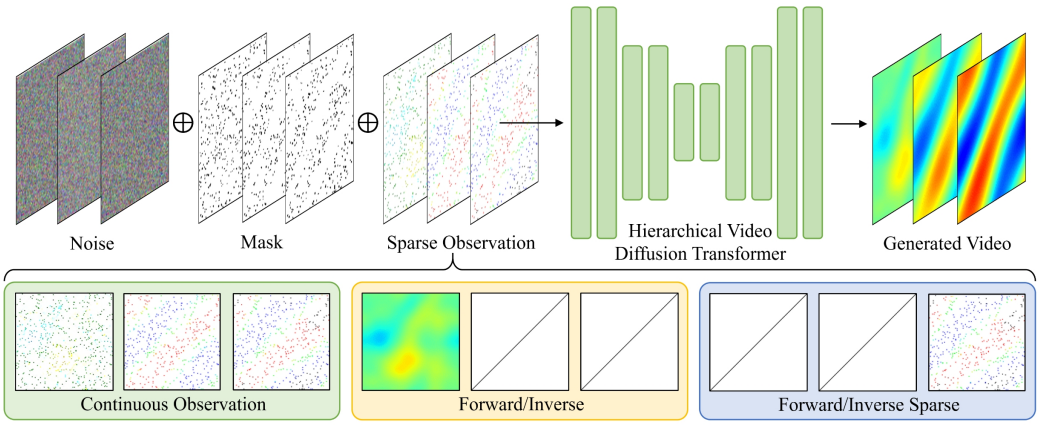


Figure 1: **VideoPDE pipeline.** We cast PDE solving as a video inpainting task. Our Hierarchical Video Diffusion Transformer (HV-DiT) denoises initial noise into a full video, conditioned on pixel-level sparse measurements. Its ability to handle arbitrary input patterns enables flexible application to diverse PDE scenarios, including forward, inverse, and continuous measurement tasks.

from which we can apply numerical ODE solvers to iteratively denoise from a complete noise $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ following

$$\mathrm{d}\mathbf{x} = -\dot{\sigma}(t)\sigma(t)\nabla_{\mathbf{x}} \log p\big(\mathbf{x}; \sigma(t)\big)\mathrm{d}t. \tag{3}$$

Conditional Sampling with Diffusion Models A central challenge in diffusion models is how to incorporate external conditions during sampling. Broadly, approaches fall into two categories. The first are *guidance-based methods*, such as classifier guidance (18) or Diffusion Posterior Sampling (DPS) (13), which steer the reverse process toward satisfying a condition by modifying the score estimate with an auxiliary gradient or correction terms. While powerful, these approaches require additional inference-time optimization or careful hyperparameter tuning, which can increase computational cost and sensitivity.

The second family are architectural conditioning methods, where the conditioning signal is directly provided to the network. Examples include Palette (60) for image-to-image translation, where conditioning images are concatenated with the noisy input at every denoising step. Our method follows this strategy: we concatenate the available spatiotemporal observations with the diffusion input, enabling the model to flexibly adapt to arbitrary observation masks (Fig. 1). This simple design avoids inference-time optimization, eliminates guidance hyperparameters, and naturally supports partial, sparse, or irregular observation patterns common in scientific PDE problems.

3.2 Spatiotemporal PDE Solving as Video Inpainting

We cast the problem of spatiotemporal PDE solving as a video inpainting task, enabling a unified and flexible framework for handling a wide range of prediction scenarios (Figure 2). Like prior data-driven PDE approaches, our goal is to learn a neural network that can infer unknown system states across a family of equations. However, unlike existing methods that typically design separate models for forward, inverse, or partially observed cases, our approach treats all such tasks as instances of conditional video inpainting.

In this formulation, we cast PDE solving as the task of filling in missing regions of a video representing the evolution of physical states over time and space. For example, forward prediction corresponds to inpainting future frames based on an initial condition; partially observed setups correspond to inpainting from sparse spatiotemporal sensor data. Our proposed architecture, described in detail in Section 3.3, is a transformer-based diffusion model explicitly designed to condition on arbitrary patterns of observed data and generate coherent, accurate completions.

PDE Formulation While our formulation accommodates both static (time-independent) and dynamic (time-dependent) PDEs, we focus on dynamic systems, e.g., Navier–Stokes:

$$\begin{aligned} f(\boldsymbol{c}, \tau; \mathbf{u}) &= 0, && \text{in } \Omega \times (0, \infty), \\ \mathbf{u}(\boldsymbol{c}, \tau) &= \boldsymbol{g}(\boldsymbol{c}, \tau), && \text{on } \partial\Omega \times (0, \infty), \\ \mathbf{u}(\boldsymbol{c}, \tau) &= \boldsymbol{o}(\boldsymbol{c}, \tau), && \text{on } \mathcal{O} \subset \Omega \times (0, \infty) \end{aligned} \tag{4}$$

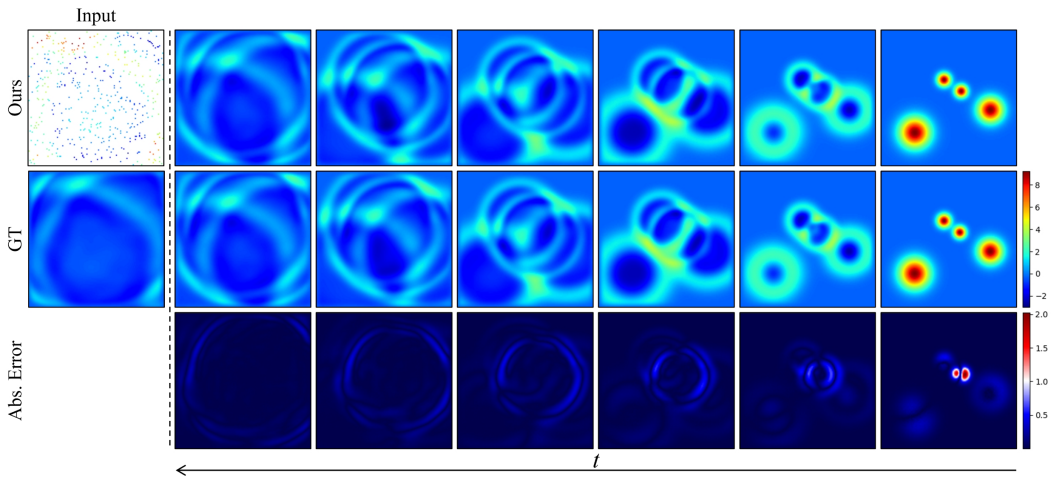


Figure 2: **Inverse simulation from partial observation.** VideoPDE formulates general PDE solving as a video inpainting problem, where unknown pixels are denoised conditioned on sparse inputs. Here, given 3% observation at time T, VideoPDE accurately recovers the whole trajectory $T-1 \rightarrow 1$.

Here, $\boldsymbol{c}$ and τ denote the spatial and temporal coordinates, respectively, and $\mathbf{u}(\boldsymbol{c}, \tau)$ is the solution field. The boundary condition is given by $\mathbf{u}|_{\partial\Omega\times(0,\infty)} = \boldsymbol{g}$. We aim to recover the full solution $\mathbf{u}_\tau$ at any time $\tau \in [0, T]$ from sparse spatiotemporal observations $\mathcal{O}$, where $\mathbf{u}|_{\mathcal{O}} = \boldsymbol{o}$. We make no assumptions about the structure of these observed locations.

Diffusion-based Video Inpainting We cast PDE-solving as a spatiotemporal inpainting task, where missing regions of the solution field $\mathbf{u}(\boldsymbol{c}, \tau)$ are inferred from sparse observations $\mathcal{O}$. To solve this inpainting problem, we leverage the powerful generative capabilities of diffusion models. Specifically, we train a conditional diffusion model to learn the distribution of physically consistent video-like solution trajectories, while conditioning on arbitrary known subsets of the spatiotemporal domain.

We represent each PDE solution as a video $\mathbf{x} \in \mathbb{R}^{H\times W\times T\times C}$, where $H \times W$ is the spatial grid, T is the number of time steps, and C the number of field channels. The conditioning signal is defined by a binary mask $\mathbf{m} \in \{0, 1\}^{H\times W\times T}$ and corresponding observed values $\boldsymbol{y} = \mathbf{x} \odot \mathbf{m}$. During training, we sample random spatiotemporal masks and supervise the model to reconstruct the full video from these partial views. The model learns the conditional score function:

$$\nabla_{\mathbf{x}} \log p(\mathbf{x}|\boldsymbol{y}; \sigma(t)) \approx (D_\theta(\mathbf{x}_t, \boldsymbol{y}, \mathbf{m}; \sigma(t)) - \mathbf{x}_t)/\sigma(t)^2, \tag{5}$$

where D_θ is a transformer-based denoising network conditioned on $\boldsymbol{y}$ and $\mathbf{m}$, and $\mathbf{x}_t$ is a noisy intermediate sample at diffusion time t. During inference, we take sparse observations $\boldsymbol{y}$ and $\mathbf{m}$ as inputs, initialize $\mathbf{x}_T$ with pure Gaussian noise, and denoise it using the learned score function.

By casting PDE-solving as conditional video generation, we unify a broad class of spatiotemporal problems under a generative modeling task. Importantly, our formulation enables conditioning the same model on different observation patterns, e.g. forward and inverse predictions, or interpolation from arbitrary observed subsets. Section 3.3 details the model design and training process.

3.3 Hierarchical Video Diffusion Transformer (HV-DiT)

While most recent state-of-the-art diffusion models (59) operate in a learned latent space to reduce computational cost, we design our architecture to perform diffusion directly in pixel space, as shown in Figure 1. This choice is motivated by our observation that pixel-space diffusion yields significantly more accurate and physically consistent reconstructions, which is particularly important in PDE settings where fine-grained field values matter more than perceptual qualities.

To manage the high dimensionality of pixel-space video data, we tokenize each input video $\mathbf{x} \in \mathbb{R}^{H\times W\times T\times C}$ by merging small spatiotemporal neighborhoods, for example, $N \times N \times N$ patches, into single tokens. This results in a structured token sequence over which we design an efficient variant of the Video DiT architecture (52), which we refer to as **HV-DiT**, inspired by the hierarchical

Method	Type	Partial obs.	Flexible inference	Dense temporal	Inf. Time (s)↓	Forward Error↓
PINN (57)	PDE loss	✓	✓	✓	66	27.3%
FNO (40)	Neural Operator			✓	2.0	<u>2.7%</u>
DeepONet (45)	Neural Operator			✓	1.7	11.3%
PINO (42)	Neural Operator+PDE			✓	2.2	4.9%
DiffusionPDE (31)	Generative+PDE	✓	△		4480	5.8%
Shu et al. (65)	Generative+PDE	✓			5760	X
Zhuang et al. (82)	Generative	✓			4860	X
VideoPDE (Ours)	Generative	✓	✓	✓	9.3	**0.45%**

Table 1: **Conceptual comparison of PDE-solving methods.** Neural operator methods struggle with partial inputs. Only PINN and VideoPDE handle forward, inverse, and continuous measurements flexibly. Generative baselines focus on reconstructing one or two frames (instead of dense temporal frames) and are often not designed for forward prediction, where VideoPDE excels.

image model HDiT (15). Unlike standard transformers with global self-attention, HV-DiT employs localized attention, restricting each token's receptive field to nearby spatiotemporal neighbors. This reduces computational complexity and allows the model to focus on local PDE dynamics.

Our transformer architecture is hierarchical (15; 51): tokens are progressively downsampled by merging neighboring tokens, creating a multi-scale representation. This downsampling path is paired with an upsampling path with skip connections in the style of U-Net, enabling both local detail preservation and global context integration. At each layer, we apply spatiotemporal neighborhood attention. At the coarsest resolution (bottleneck), we use global attention layers to capture long-range spatiotemporal dependencies.

A key architectural innovation is the way we condition the model on known observations. For each token, we concatenate its associated binary mask (indicating observed pixels) and the corresponding observed values. This allows our model to condition at the individual pixel level, enabling fine-grained, spatially varying guidance during the denoising process. Concatenating the binary mask resolves ambiguity between observed and unobserved pixels. This formulation supports flexible conditioning across a wide range of scenarios, including forward prediction, inverse recovery, and inpainting from arbitrary subsets of observations. The concatenated final input to $D_\theta(\mathbf{x}_t^{cond})$ is:

$$\mathbf{x}_t^{cond} \equiv \text{concat}(\mathbf{x}_t, \mathbf{m}, \boldsymbol{y}), \quad \#\text{ of tokens is } H/N \times W/N \times T/N \tag{6}$$

Note that only the solution field $\mathbf{x}$ part of the input token contains the diffusion noise.

Overall, our HV-DiT combines the expressiveness of pixel-space modeling with the efficiency of localized and hierarchical attention, forming a powerful and versatile backbone for generative PDE-solving through conditional video inpainting.

4 Experiments

We comprehensively evaluate VideoPDE's ability to solve a range of temporal PDEs across diverse inference scenarios. Specifically, we assess its performance in (i) reconstructing from continuous spatiotemporal sensor measurements (Table 2), (ii) predicting future or past system states (Table 3), (iii) handling partial observations during forward and inverse prediction (Table 4) , and (iv) generalizing across multiple inference tasks, including forward, inverse, and reconstruction.

Baselines We compare VideoPDE against a representative set of learning-based PDE solvers. For standard forward and inverse prediction under full initial or final conditions, we include FNO, PINO, DeepONet, and DiffusionPDE, each representing a distinct modeling paradigm (see Table 1). For partial observation settings, we compare only against DiffusionPDE, which has demonstrated superior performance and shown that prior baselines struggle with sparse conditioning. For the continuous measurement reconstruction task, we evaluate against state-of-the-art generative methods, including those proposed by Shu et al. (65), Zhuang et al. (82), and DiffusionPDE (31). We also extend DiffusionPDE for improved temporal message passing. See the supplementary for more details.

4.1 PDE Problem Settings

We evaluate VideoPDE on five representative PDEs plus one real-world dataset, spanning wave propagation, fluid dynamics, reaction–diffusion, and elliptic problems. This diversity demonstrates

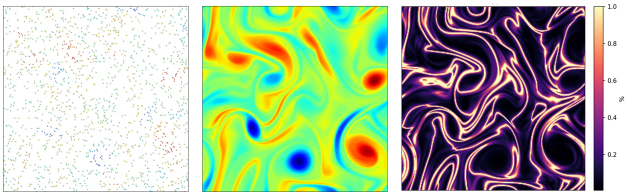


Figure 3: **Uncertainty visualization.** From a sparse first frame (a), VideoPDE predicts future states (b). By running diffusion inference with multiple random seeds, we obtain per-pixel uncertainty estimates (c).

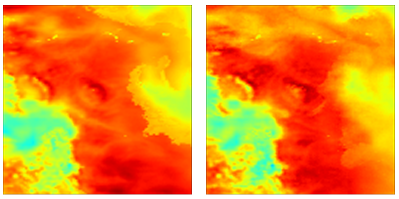


Figure 4: **ERA5 temperature data.** VideoPDE can predict accurate real-world atmospheric temeprature field from 3% sparse sensors.

the flexibility of our unified framework across forward, inverse, and partial-observation scenarios. Full dataset construction details are provided in the supplementary.

Wave-Layer A variable-speed wave equation with absorbing boundary conditions, following Poseidon (26), modeling seismic propagation through layered media: $\partial_t^2 u(\boldsymbol{c}, \tau) + q(\boldsymbol{c})^2 \Delta u(\boldsymbol{c}, \tau) = 0$.

Navier–Stokes Two-dimensional incompressible Navier–Stokes in vorticity form with weak, static forcing (31), producing diffusive decay: $\partial_t \omega + v \cdot \nabla \omega = \nu \Delta \omega + q(\boldsymbol{c})$.

Kolmogorov Flow A forced Navier–Stokes system with dynamic feedback, yielding sustained quasi-turbulent regimes (65): $\partial_t \omega + v \cdot \nabla \omega = \nu \Delta \omega - 4\cos(4c_2) - 0.1\omega$.

Allen–Cahn and Helmholtz We also test on the Allen–Cahn reaction–diffusion equation and the static Helmholtz equation to demonstrate adaptability to nonlinear and elliptic PDEs. We include experiment results for these eqution in the supplementary document.

ERA5 Reanalysis Data We evaluate on a real-world atmospheric dataset (27) providing full-field spatiotemporal measurements from which we use 2m temperature scalar field. We crop North American region and downsample into 256×256 video-like slices for training and evaluation.

4.2 Experiment Details

Training was conducted on 4 NVIDIA L40S GPUs (batch size 8/GPU) for about 24 hours per model, using Adam with a fixed learning rate of 5×10^{-4}. Our HV-DiT architecture operates in pixel space, tokenizing videos into $4 \times 4 \times 2$ patches. The model consists of 2 transformer layers with $7 \times 7 \times 2$ neighborhood attention and downsampling via patch merging (factor 2). Additional dataset, architecture, and hyperparameter details are provided in the supplementary.

4.3 Experiment Results

Continuous Partial Observation We evaluate the ability of VideoPDE to reconstruct full spatiotemporal PDE trajectories from sparse, fixed-point observations. Specifically, we randomly sample a very small percentage of spatial coordinates (1% or 3%) and provide the solution values across all time steps at those locations. This setting mimics real-world sensor deployments, where measurements are collected continuously at fixed spatial positions. Our model is conditioned on these sparse yet temporally continuous observations.

As shown in Table 2, we report the relative ℓ_2 error across 100 held-out trajectories for three PDEs: Wave-Layer, Navier–Stokes, and Komolgorov Flow. In Figure 5 we visualize the error map for the Navier–Stokes. Our method significantly outperforms existing generative baselines, including DiffusionPDE (31), Shu et al. (65), and Zhuang et al. (82), up to an order of magnitude, demonstrating robustness under extreme observation sparsity. We also demonstrate real-world temperature field reconstruction on the ERA5 dataset (27), using only 3% continuous sensor observations (Fig. 4).

Forward/Inverse Full Observation We evaluate VideoPDE on reconstructing full PDE trajectories given a single frame at either the start (forward prediction) or end (inverse inference) of the sequence. The full conditioning frame is provided while the remaining frames are masked. This setup reflects practical simulation scenarios where dense initial conditions are available and parallels image-to-video tasks in generative modeling.

	Wave-Layer		Navier–Stokes		Kolmogorov Flow	
	1%	3%	1%	3%	1%	3%
DiffusionPDE	48.3%	17.4%	4.7%	3.7%	20.3%	11.9%
DiffusionPDE (Ext.)	45.2%	15.6%	4.1%	3.4%	19.5%	10.3%
Shu et al.	49.7%	17.7%	8.6%	6.2%	19.7%	11.8%
Zhuang et al.	29.9%	10.3%	12.7%	4.8%	13.9%	6.1%
Ours	**2.62%**	**1.57%**	**0.80%**	**0.44%**	**6.48%**	2.71%
Ours (unified)	3.97%	2.05%	1.13%	0.48%	7.59%	**2.55%**

Table 2: **Continuous partial observation reconstruction.** We quantitatively measure the performance of different methods using average ℓ_2 relative errors on Wave-Layer, Navier–Stokes, and Kolmogorov Flow benchmarks with 1% and 3% observation points.

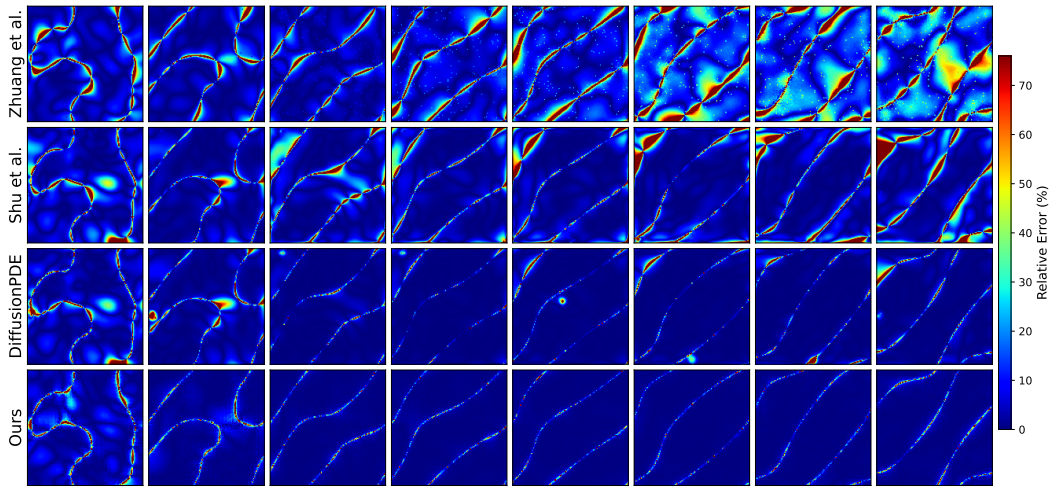


Figure 5: **Continuous measurement reconstruction comparison.** We compare relative error maps for reconstructing dense spatiotemporal fields from fixed sensors providing 1% continuous observations on Navier–Stokes. Our results are the most accurate with minimal error. In contrast, baseline methods are significantly slower and not suitable for forward prediction (Zhuang & Shu).

Figure 6 shows the final/initial frames of the fully observed forward/inverse processes on the Kolmogorov Flow dataset, demonstrating that VideoPDE consistently produces results that are closer to the ground truth. Table 3 reports the relative ℓ_2 error across 100 held-out trajectories for three PDEs. VideoPDE consistently outperforms baselines in both forward and inverse tasks, except for the low-frequency inverse setting. We attribute this to aleatoric uncertainty: in the NS dataset, diffusive dynamics lead to low-frequency end states that may originate from many high-frequency initial conditions. In such cases, pixel-wise ℓ_2 loss penalizes plausible reconstructions and favors blurry averages. We leave exploration of distribution-based evaluation metrics for future work.

In Fig. 7 we showcase long-horizon forward prediction, where we autoregressively run multiple 20-frame predictions conditioned on previously generated frames. Refer to supplementary for details.

Forward/Inverse Partial Observation We extend the forward and inverse prediction tasks to the partially observed setting by conditioning on a single frame—either at the start or end of the trajectory, with only 3% of spatial points revealed. The model must reconstruct the full trajectory from these observations, reflecting real-world scenarios where sensors provide limited data at a single timepoint.

In Figure 2, we present the inverse simulation of a Wave-Layer sample, where VideoPDE recovers all time steps in reverse given only 3% of observation points from the final frame. As shown in Table 4, VideoPDE outperforms DiffusionPDE, the current SOTA for this task, by a significant margin across all settings, except for inverse prediction on the Navier–Stokes case, where aleatoric uncertainty remains high due to the diffusive loss of high-frequency information. We note that VideoPDE performs similarly on this task to the forward/inverse *full* observation task, particularly for Wave-Layer forward prediction, and both Navier–Stokes forward and inverse prediction.

	Wave-Layer		Navier–Stokes		Kolmogorov Flow	
	Forward	Inverse	Forward	Inverse	Forward	Inverse
FNO	35.34%	65.43%	2.71%	7.62%	56.43%	59.42%
PINO	10.8%	19.7%	4.9%	**6.9%**	7.4%	7.6%
DeepONet	47.68%	53.32%	11.29%	12.63%	46.61%	46.92%
DiffusionPDE	6.7%	14.2%	6.1%	8.6%	9.1%	10.8%
Ours	**1.21%**	**5.24%**	**0.45%**	9.87%	**2.95%**	**4.90%**
Ours (unified)	1.53%	12.65%	1.52%	10.37%	4.60%	4.99%

Table 3: **Forward/inverse full observation.** Average ℓ_2 relative errors of baseline methods for forward and inverse subtasks across datasets.

	Wave-Layer		Navier–Stokes		Kolmogorov Flow	
	Forward	Inverse	Forward	Inverse	Forward	Inverse
DiffusionPDE	19.5%	24.3%	3.9%	**10.2%**	28.2%	32.6%
Ours	**1.40%**	**11.81%**	**0.71%**	10.41%	**11.66%**	**13.50%**
Ours (unified)	1.89%	18.56%	1.61%	11.45%	16.11%	25.75%

Table 4: **Forward/inverse 3% observation.** Average ℓ_2 relative errors of baseline methods for forward and inverse subtasks across datasets.

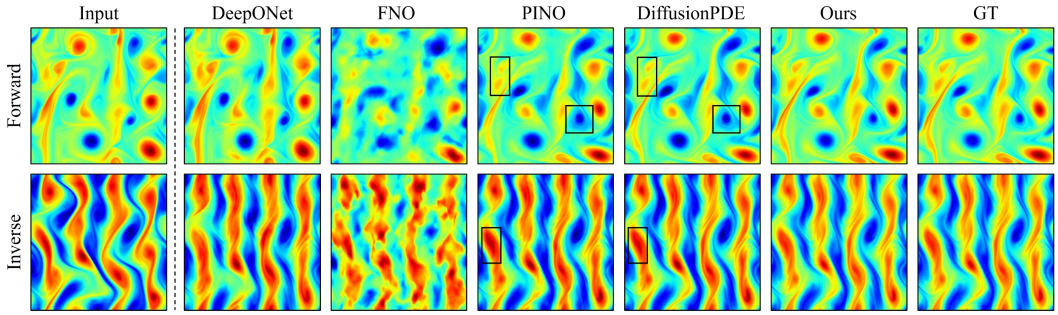


Figure 6: **Comparison of forward/inverse predictions on Kolmogorov Flow.** Our predictions are perceptually indistinguishable from ground truths, whereas other baseline results exhibit deviations. Significant errors of PINO and DiffusionPDE are squared in black.

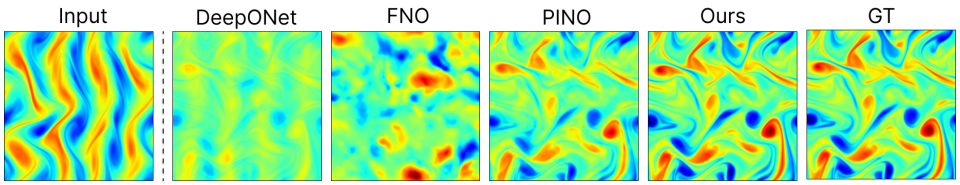


Figure 7: **Long-horizon prediction.** We autoregressively roll out 100 frames from the initial input using our and baseline methods, where VideoPDE produces the most accurate long-term predictions.

Unified Model We evaluate whether a single model can jointly learn multiple inference tasks within our video inpainting framework. For each dataset, we train one unified model on six tasks: continuous partial observation (3% and 1%), forward and inverse prediction under full/partial observation. As shown in Tables 2, 3, and 4, the unified model matches the performance of task-specific variants and outperforms prior baselines in most settings. In contrast, all baselines require separate models per task, highlighting VideoPDE's potential to be a unified framework for flexible PDE solving.

4.4 Uncertainty Quantification

A key advantage of our generative formulation is that it naturally provides uncertainty estimates alongside point predictions. Unlike regression-based approaches that output a single deterministic

	Navier–Stokes
DiT with mixed noise and conditioning tokens	50.77%
DiT with each token concatenated on sparse observations	1.46%
+ Latent diffusion	7.13%
+ 3D Neighborhood attention and downsampling	<u>0.73%</u>
+ concatenating binary mask (Ours, HV-DiT)	**0.44%**

Table 5: **Ablation study.** We ablate our design choices, beginning with a latent space DiT. We report average relative ℓ_2 errors for all configurations on Navier–Stokes with 3% observation rate.

trajectory, our diffusion model defines a full distribution over conditional spatiotemporal fields. By sampling ten trajectories with different random seeds, we estimate per-pixel predictive standard deviation (std), and normalize it by the mean absolute value to obtain a relative std map. This allows us to quantify and visualize spatiotemporal uncertainty (Fig. 3), particularly important for chaotic dynamics where small perturbations can yield diverging outcomes.

4.5 Ablation Study

We conduct an ablation study to assess the impact of key architectural choices, evaluated on the continuous partial observation task for low-frequency Navier–Stokes with a 3% observation rate. Relative ℓ_2 errors are reported in Table 5.

We begin with a video DiT architecture adapted from VDT (44), originally designed for natural video inpainting. The model input is $\mathbf{x} \odot (1 - \mathbf{m}) + \mathbf{y} \odot \mathbf{m}$, where $\mathbf{x}$ is Gaussian noise, $\mathbf{m}$ a binary mask, and $\mathbf{y}$ the ground truth. This model performed poorly, likely due to confusion between sparse observations and noise.

Replacing the conditioning method with channel-wise concatenation of noise and masked ground truth, i.e., $\mathrm{concat}(\mathbf{x}_t, \boldsymbol{y})$, significantly improves performance. Building on this, we train a latent diffusion version using a task-specific VAE. However, due to the precision requirements of PDEs, the latent model performs poorly, highlighting the need for pixel-space modeling in scientific applications. Next, we introduce a hierarchical variant of the DiT with 3D neighborhood attention and temporal downsampling inspired by HDiT (16), which further reduces error. Finally, conditioning on the binary mask itself yields the best performance in our setup, indicating that the binary mask resolves ambiguity between masked and unmasked pixels: the input is $\mathrm{concat}(\mathbf{x}_t, \boldsymbol{y}, \mathbf{m})$.

4.6 Supplementary

We provide extended experimental results and analysis, including *additional PDEs*, *long-horizon predictions*, and *solution videos* in the supplementary. Moreover, we provide further details on our training hyperparameters, datasets, and model architecture. We urge readers to review the additional images and videos in the supplementary.

5 Conclusion

In this work, we introduced VideoPDE, a unified framework for PDE solving by reframing forward, inverse, and partial observation problems as video inpainting with diffusion models. Our hierarchical, pixel-level architecture leverages flexible conditioning to adapt seamlessly to arbitrary observation patterns, and consistently outperforms state-of-the-art baselines. Notably, a single model trained jointly across tasks achieves higher accuracy and generalization than existing task-specific approaches. We also demonstrate that our framework can naturally quantify uncertainty and show cutting-edge performance across five representative PDEs and a real-world dataset.

Looking ahead, we plan to extend VideoPDE toward more challenging long-horizon prediction regimes, where compounding uncertainty poses a key challenge. We will also investigate evaluation metrics that better capture stochastic, multimodal generative behavior, and explore 3D extensions for volumetric spatiotemporal systems. A discussion of current limitations is provided in the supplementary material.

REFERENCES

[1] Ferri MH Aliabadi. Boundary element methods. In *Encyclopedia of continuum mechanics*, pages 182–193. Springer, 2020.

[2] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers, 2023.

[3] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[4] Boris Bonev, Thorsten Kurth, Christian Hundt, Jaideep Pathak, Maximilian Baust, Karthik Kashinath, and Anima Anandkumar. Spherical fourier neural operators: Learning stable dynamics on the sphere. In *International conference on machine learning*, pages 2806–2823. PMLR, 2023.

[5] Shengze Cai, Zhiping Mao, Zhicheng Wang, Minglang Yin, and George Em Karniadakis. Physics-informed neural networks (pinns) for fluid mechanics: A review. *Acta Mechanica Sinica*, 37(12):1727–1738, 2021.

[6] Shengze Cai, Zhicheng Wang, Sifan Wang, Paris Perdikaris, and George Em Karniadakis. Physics-informed neural networks for heat transfer problems. *Journal of Heat Transfer*, 143(6):060801, 2021.

[7] Gary J Chandler and Rich R Kerswell. Invariant recurrent solutions embedded in a turbulent two-dimensional kolmogorov flow. *Journal of Fluid Mechanics*, 722:554–595, 2013.

[8] Xinyuan Chen, Yaohui Wang, Lingjun Zhang, Shaobin Zhuang, Xin Ma, Jiashuo Yu, Yali Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Seine: Short-to-long video diffusion model for generative transition and prediction. In *ICLR*, 2023.

[9] Chaoran Cheng, Boran Han, Danielle C. Maddix, Abdul Fatir Ansari, Andrew Stuart, Michael W. Mahoney, and Bernie Wang. Gradient-free generation for hard-constrained systems. In *The Thirteenth International Conference on Learning Representations*, 2025.

[10] Aditya Chetan, Guandao Yang, Zichen Wang, Steve Marschner, and Bharath Hariharan. Accurate differential operators for hybrid neural fields, 2023.

[11] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models, 2021.

[12] Hyungjin Chung, Jeongsol Kim, Michael T. Mccann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems, 2024.

[13] 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.

[14] Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints, 2024.

[15] Katherine Crowson, Stefan Andreas Baumann, Alex Birch, Tanishq Mathew Abraham, Daniel Z Kaplan, and Enrico Shippole. Scalable high-resolution pixel-space image synthesis with hourglass diffusion transformers. In *Forty-first International Conference on Machine Learning*, 2024.

[16] Katherine Crowson, Stefan Andreas Baumann, Alex Birch, Tanishq Mathew Abraham, Daniel Z Kaplan, and Enrico Shippole. Scalable high-resolution pixel-space image synthesis with hourglass diffusion transformers. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of*

the 41st International Conference on Machine Learning, volume 235 of *Proceedings of Machine Learning Research*, pages 9550–9575. PMLR, 21–27 Jul 2024.

[17] Moein Darman, Pedram Hassanzadeh, Laure Zanna, and Ashesh Chattopadhyay. Fourier analysis of the physics of transfer learning for data-driven subgrid-scale models of ocean turbulence, 2025.

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

[19] Surya Dheeshjith, Adam Subel, Alistair Adcroft, Julius Busecke, Carlos Fernandez-Granda, Shubham Gupta, and Laure Zanna. Samudra: An ai global ocean emulator for climate, 2024.

[20] Hamidreza Eivazi, Mojtaba Tahani, Philipp Schlatter, and Ricardo Vinuesa. Physics-informed neural networks for solving reynolds-averaged navier–stokes equations. *Physics of Fluids*, 34(7), 2022.

[21] Kai Fukami, Koji Fukagata, and Kunihiko Taira. Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows. *Journal of Fluid Mechanics*, 909, December 2020.

[22] Kai Fukami, Koji Fukagata, and Kunihiko Taira. Super-resolution analysis via machine learning: a survey for fluid flows. *Theoretical and Computational Fluid Dynamics*, 37(4):421–444, 2023.

[23] Agrim Gupta, Lijun Yu, Kihyuk Sohn, Xiuye Gu, Meera Hahn, Li Fei-Fei, Irfan Essa, Lu Jiang, and José Lezama. Photorealistic video generation with diffusion models, 2023.

[24] Zhongkai Hao, Jiachen Yao, Chang Su, Hang Su, Ziao Wang, Fanzhi Lu, Zeyu Xia, Yichi Zhang, Songming Liu, Lu Lu, et al. Pinnacle: A comprehensive benchmark of physics-informed neural networks for solving pdes. *arXiv preprint arXiv:2306.08827*, 2023.

[25] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation. 2022.

[26] Maximilian Herde, Bogdan Raonic, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bezenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for PDEs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[27] Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, et al. The era5 global reanalysis. *Quarterly journal of the royal meteorological society*, 146(730):1999–2049, 2020.

[28] Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, Adrian Simmons, Cornel Soci, Saleh Abdalla, Xavier Abellan, Gianpaolo Balsamo, Peter Bechtold, Gionata Biavati, Jean Bidlot, Massimo Bonavita, Giovanna De Chiara, Per Dahlgren, Dick Dee, Michail Diamantakis, Rossana Dragani, Johannes Flemming, Richard Forbes, Manuel Fuentes, Alan Geer, Leo Haimberger, Sean Healy, Robin J. Hogan, Elías Hólm, Marta Janisková, Sarah Keeley, Patrick Laloyaux, Philippe Lopez, Cristina Lupu, Gabor Radnoti, Patricia de Rosnay, Iryna Rozum, Freja Vamborg, Sebastien Villaume, and Jean-Noël Thépaut. The era5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020.

[29] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020.

[30] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models, 2022.

[31] Jiahe Huang, Guandao Yang, Zichen Wang, and Jeong Joon Park. Diffusionpde: Generative pde-solving under partial observation, 2024.

[32] Sergio R Idelsohn, Eugenio Onate, Nestor Calvo, and Facundo Del Pin. The meshless finite element method. *International Journal for Numerical Methods in Engineering*, 58(6):893–912, 2003.

[33] 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.

[34] Karthik Kashinath, M Mustafa, Adrian Albert, JL Wu, C Jiang, Soheil Esmaeilzadeh, Kamyar Azizzadenesheli, R Wang, Ashesh Chattopadhyay, A Singh, et al. Physics-informed machine learning: case studies for weather and climate modelling. *Philosophical Transactions of the Royal Society A*, 379(2194):20200093, 2021.

[35] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In *Advances in Neural Information Processing Systems*, 2022.

[36] Hyojin Kim, Junhyuk Kim, Sungjin Won, and Changhoon Lee. Unsupervised deep learning for super-resolution reconstruction of turbulence. *Journal of Fluid Mechanics*, 910, January 2021.

[37] Bian Li, Hanchen Wang, Shihang Feng, Xiu Yang, and Youzuo Lin. Solving seismic wave equations on variable velocity models with fourier neural operator. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–18, 2023.

[38] Zhijie Li, Wenhui Peng, Zelong Yuan, and Jianchun Wang. Fourier neural operator approach to large eddy simulation of three-dimensional turbulence. *Theoretical and Applied Mechanics Letters*, 12(6):100389, 2022.

[39] Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for partial differential equations' operator learning. *Transactions on Machine Learning Research*, 2023.

[40] 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*, 2020.

[41] Zongyi Li, Nikola Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, et al. Geometry-informed neural operator for large-scale 3d pdes. *Advances in Neural Information Processing Systems*, 36, 2024.

[42] Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *ACM/JMS Journal of Data Science*, 2021.

[43] Da Long and Shandian Zhe. Invertible fourier neural operators for tackling both forward and inverse problems. *arXiv preprint arXiv:2402.11722*, 2024.

[44] Haoyu Lu, Guoxing Yang, Nanyi Fei, Yuqi Huo, Zhiwu Lu, Ping Luo, and Mingyu Ding. VDT: General-purpose video diffusion transformers via mask modeling. In *The Twelfth International Conference on Learning Representations*, 2024.

[45] 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.

[46] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models, 2022.

[47] Zhiping Mao, Ameya D Jagtap, and George Em Karniadakis. Physics-informed neural networks for high-speed flows. *Computer Methods in Applied Mechanics and Engineering*, 360:112789, 2020.

[48] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

[49] George S Misyris, Andreas Venzke, and Spyros Chatzivasileiadis. Physics-informed neural networks for power systems. In *2020 IEEE power & energy society general meeting (PESGM)*, pages 1–5. IEEE, 2020.

[50] Roberto Molinaro, Yunan Yang, Björn Engquist, and Siddhartha Mishra. Neural inverse operators for solving pde inverse problems. *arXiv preprint arXiv:2301.11167*, 2023.

[51] Piotr Nawrot, Szymon Tworkowski, Michał Tyrolski, Łukasz Kaiser, Yuhuai Wu, Christian Szegedy, and Henryk Michalewski. Hierarchical transformers are more efficient language models. *arXiv preprint arXiv:2110.13711*, 2021.

[52] OpenAI. Introducing sora: A text-to-video model, February 2024. Accessed: 2025-05-13.

[53] Wenhui Peng, Zelong Yuan, Zhijie Li, and Jianchun Wang. Linear attention coupled fourier neural operator for simulation of three-dimensional turbulence. *Physics of Fluids*, 35(1), 2023.

[54] Michael Penwarden, Shandian Zhe, Akil Narayan, and Robert M. Kirby. A metalearning approach for physics-informed neural networks (pinns): Application to parameterized pdes. *Journal of Computational Physics*, 477:111912, 2023.

[55] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer, 2017.

[56] Alfio Quarteroni and Alberto Valli. *Numerical approximation of partial differential equations*, volume 23. Springer Science & Business Media, 2008.

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

[58] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, 2017.

[59] 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*, pages 10684–10695, 2022.

[60] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022.

[61] Chitwan Saharia, William Chan, Huiwen Chang, Chris A. Lee, Jonathan Ho, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models, 2022.

[62] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement, 2021.

[63] Louis Serrano, Lise Le Boudec, Armand Kassaï Koupaï, Thomas X Wang, Yuan Yin, Jean-Noël Vittaut, and Patrick Gallinari. Operator learning with neural fields: Tackling pdes on general geometries. *Advances in Neural Information Processing Systems*, 36, 2024.

[64] Siming Shan, Pengkai Wang, Song Chen, Jiaxu Liu, Chao Xu, and Shengze Cai. Pird: Physics-informed residual diffusion for flow field reconstruction, 2024.

[65] 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, April 2023.

[66] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015.

[67] Pavel Ŝolín. *Partial differential equations and the finite element method*. John Wiley & Sons, 2005.

[68] Bowen Song, Soo Min Kwon, Zecheng Zhang, Xinyu Hu, Qing Qu, and Liyue Shen. Solving inverse problems with latent diffusion models via hard data consistency, 2024.

[69] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution, 2020.

[70] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[71] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021.

[72] Juan Diego Toscano, Vivek Oommen, Alan John Varghese, Zongren Zou, Nazanin Ahmadi Daryakenari, Chenxi Wu, and George Em Karniadakis. From pinns to pikans: Recent advances in physics-informed machine learning, 2024.

[73] Umair bin Waheed, Ehsan Haghighat, Tariq Alkhalifah, Chao Song, and Qi Hao. Pineik: Eikonal solution using physics-informed neural networks. *Computers & Geosciences*, 155:104833, October 2021.

[74] Rui Wang, Karthik Kashinath, Mustafa Mustafa, Adrian Albert, and Rose Yu. Towards physics-informed deep learning for turbulent flow prediction. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1457–1466, 2020.

[75] Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. Videocomposer: Compositional video synthesis with motion controllability, 2023.

[76] Gege Wen, Zongyi Li, Kamyar Azizzadenesheli, Anima Anandkumar, and Sally M Benson. U-fno—an enhanced fourier neural operator-based deep-learning model for multiphase flow. *Advances in Water Resources*, 163:104180, 2022.

[77] Sihan Xu, Yidong Huang, Jiayi Pan, Ziqiao Ma, and Joyce Chai. Inversion-free image editing with natural language. 2024.

[78] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Comput. Surv.*, 56(4), November 2023.

[79] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023.

[80] Zhixing Zhang, Bichen Wu, Xiaoyan Wang, Yaqiao Luo, Luxin Zhang, Yinan Zhao, Peter Vajda, Dimitris Metaxas, and Licheng Yu. Avid: Any-length video inpainting with diffusion model, 2024.

[81] Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models, 2023.

[82] Yilin Zhuang, Sibo Cheng, and Karthik Duraisamy. Spatially-aware diffusion models with cross-attention for global field reconstruction with sparse observations, 2024.

[83] Zeshun Zong, Xuan Li, Minchen Li, Maurizio M Chiaramonte, Wojciech Matusik, Eitan Grinspun, Kevin Carlberg, Chenfanfu Jiang, and Peter Yichen Chen. Neural stress fields for reduced-order elastoplasticity and fracture. *arXiv preprint arXiv:2310.17790*, 2023.

A Overview

This supplementary material provides additional experiments and analyses to support the findings presented in the main paper. In Section B, we evaluate VideoPDE on additional datasets involving both static and dynamic PDEs to highlight its generalization capability. Section D presents an extension of DiffusionPDE for spatio-temporal PDEs. Section E assesses the model's robustness to random noise and its inherent stochasticity. In Section F, we analyze the relationship between overall error rates and the observation ratio. Section G investigates the long-range generalization ability of models trained on short-range data. Section H explores the performance of VideoPDE under various video settings. Finally, Section I includes VideoPDE's architectural and training details.

A.1 HTML Webpage

We provide a project page in HTML format with embedded video resources for additional video results showcasing a variety of tasks and PDE families. These visualizations further illustrate the versatility and effectiveness of VideoPDE across diverse problem settings. We strongly encourage readers to view the supplemental webpage to fully appreciate the quality of our predictions visually.

B Additional PDE Experiments and Experiment Settings

In this section, we further evaluate VideoPDE on additional datasets, including one dynamic (Allen-Cahn) and another static equations (Helmholtz). We also provide detailed explanations of the settings for the three PDEs addressed in the main paper (Wave-Layer, Navier-Stokes, Kolmogorov Flow).

VideoPDE can be trivially extended to handle static PDEs by interpreting the coefficient field and the corresponding solution as two distinct time steps within a temporal framework. This perspective enables the application of our dynamic modeling approach to inherently time-independent problems, such as the Helmholtz Equation described below.

Inhomogeneous Helmholtz Equation Further, we evaluate the static Helmholtz equation as described in DiffusionPDE (31).

$$\begin{aligned} \nabla^2 \mathbf{u}(\mathbf{c}) + \alpha^2 \mathbf{u}(\mathbf{c}) &= \mathbf{a}(\mathbf{c}), && \mathbf{c} \in \Omega \\ \mathbf{u}(\mathbf{c}) &= 0, && \mathbf{c} \in \partial\Omega, \end{aligned}$$

where $\alpha = 1$.

Moreover, to compensate the dynamic PDEs in the main paper, which mostly feature highly dynamic propagations, we test VideoPDE on another PDE (Allen-Cahn) that models reaction-diffusion described below. As shown in Fig. 8, the slow-moving nature of the solution fields makes them challenging to capture from fixed sensors, resulting in higher overall errors compared to more dynamic datasets in the main paper.

Allen–Cahn Equation We study the time-dependent Allen-Cahn equation (ACE) task using the dataset prepared by Poseidon (26).

$$\begin{aligned} \partial_t \mathbf{u}(\mathbf{c}, \tau) &= \Delta \mathbf{u}(\mathbf{c}, \tau) - \gamma^2 \mathbf{u}(\mathbf{c}, \tau)(\mathbf{u}(\mathbf{c}, \tau)^2 - 1), && \mathbf{c} \in \Omega, \tau \in (0, T] \\ \mathbf{u}(\mathbf{c}, 0) &= \mathbf{u}_0(\mathbf{c}), && \mathbf{c} \in \Omega \end{aligned}$$

where $\gamma = 220$ is the reaction rate.

ERA5 Additionally, we study the ERA5 dataset of global climate and weather (28). Current climate observations are combined with previous forecasts to estimate the state of the atmosphere. This dataset contains hourly data for many variables for over 80 years. For evaluation, we selected the 2m temperature variable, which measures the temperature 2 meters above the surface of the planet. We used the observations from January 2024, and cropped a fixed 128×128 subset of the global data.

Wave-Layer We evaluate our method on the Wave-Layer task following Poseidon (26). This task is based on the wave equation with spatially varying propagation speed and absorbing boundary:

$$\partial_t^2 \mathbf{u}(\boldsymbol{c}, \tau) + (\mathbf{q}(\boldsymbol{c}))^2 \Delta \mathbf{u}(\boldsymbol{c}, \tau) = 0, \quad (\boldsymbol{c}, \tau) \in \Omega \times (0, T).$$

Here, $\mathbf{u} \colon \Omega \times (0, T) \to \mathbb{R}$ is a scalar field representing displacement, and $\mathbf{q} \colon \Omega \to \mathbb{R}$ represents propagation speed. The initial condition is the sum of 2-6 Gaussians with random location and scale: The propagation speed coefficient $\boldsymbol{c}$ is generated by creating 3-6 layers with piecewise constant propagation speeds. The layers are separated by reandomly generated frontiers. The dataset contains 10,512 trajectories, each with 21 time steps at 128×128 resolution. The final 100 trajectories are used for validation and the rest for training. This task arises from propagation of seismic waves through a layered medium. See the supplementary for more details on this problem.

Navier–Stokes Equation We study the two-dimensional incompressible Navier–Stokes equations in vorticity form, following the setup introduced in DiffusionPDE (31):

$$\begin{aligned} \partial_t \mathbf{w}(\boldsymbol{c}, \tau) + \mathbf{v}(\boldsymbol{c}, \tau) \cdot \nabla \mathbf{w}(\boldsymbol{c}, \tau) &= \nu \Delta \mathbf{w}(\boldsymbol{c}, \tau) + \mathbf{q}(\boldsymbol{c}), && \boldsymbol{c} \in \Omega,\ \tau \in (0, T], \\ \nabla \cdot \mathbf{v}(\boldsymbol{c}, \tau) &= 0, && \boldsymbol{c} \in \Omega,\ \tau \in [0, T], \\ \mathbf{w}(\boldsymbol{c}, 0) &= \mathbf{w}_0(\boldsymbol{c}), && \boldsymbol{c} \in \Omega. \end{aligned} \tag{7}$$

Here, $\mathbf{w} = \nabla \times \mathbf{v}$ denotes the vorticity field, and $\mathbf{v}(\boldsymbol{c}, \tau)$ is the velocity field at spatial location $\boldsymbol{c}$ and time τ. We fix the viscosity coefficient to $\nu = 10^{-3}$, corresponding to a Reynolds number of $Re = 1/\nu = 1000$. Initial conditions $\mathbf{w}_0$ are sampled from a Gaussian random field as in DiffusionPDE. Each datapoint is composed of 20 frames of a 128×128 vorticity field $\mathbf{w}$.

The external forcing $\mathbf{q}(\boldsymbol{c})$ determines the long-term behavior of the system. In this setting, we adopt a static, time-independent forcing term:

$$\mathbf{q}(\boldsymbol{c}) = 0.1 \left(\sin(2\pi(c_1 + c_2)) + \cos(2\pi(c_1 + c_2)) \right),$$

which introduces smooth, low-frequency energy into the system without any feedback from the flow itself. Due to the weak magnitude of this forcing and the absence of dynamic coupling, the system exhibits diffusion-like decay: initial high-frequency vorticity structures dissipate over time as the system evolves under viscous damping.

Kolmogorov Flow To study more complex and persistent flow dynamics, we also evaluate our method on the Kolmogorov flow (KF) (7), a classical setup used in (65) to simulate forced, quasi-turbulent regimes in 2D fluid dynamics. The same Navier–Stokes formulation applies, but with a different forcing term of the KF form:

$$\mathbf{q}(\boldsymbol{c}, \tau) = -4 \cos(4c_2) - 0.1 \mathbf{w}(\boldsymbol{c}, \tau).$$

This forcing is composed of a strong, anisotropic spatial component ($\cos(4c_2)$) that continuously injects energy into the system, and a linear drag term ($-0.1\mathbf{w}$) that stabilizes the flow by removing energy at small scales. Crucially, the forcing depends on the evolving state $\mathbf{w}(\boldsymbol{c}, \tau)$, introducing dynamic feedback that enables sustained motion.

Unlike the decaying dynamics of the previous setup, Kolmogorov flow exhibits persistent, swirling structures and high-frequency vorticity patterns over time. This makes it a challenging and realistic benchmark for generative PDE modeling, particularly in capturing long-term, high-fidelity spatiotemporal behavior. Finally, each datapoint is a 20-frame 256×256 vorticity field.

Similar to the main paper, we assess the performance of various models across three distinct tasks.

Continuous Partial Observation Based on the settings from the main experiments, we reconstruct complete spatiotemporal and static PDE trajectories using sparse, fixed-point observations. Specifically, we evaluate the average error of the coefficient and solution for static PDEs. Table 6 demonstrates that VideoPDE consistently produces more accurate reconstructions than all baseline methods on the dynamic PDE task, as shown in Figure 8 as well, reinforcing the findings reported in the main paper. Furthermore, VideoPDE achieves performance comparable to state-of-the-art methods on static PDE reconstruction when there are observations on both coefficients and solutions.

	Allen–Cahn		Helmholtz		ERA5	
	1%	3%	1%	3%	3%	10%
DiffusionPDE	30.65%	7.45%	**9.75%**	**6.30%**	X	X
DiffusionPDE (Ext.)	29.05%	6.93%	X	X	X	X
Shu et al.	30.71%	8.10%	X	X	X	X
Zhuang et al.	27.43%	7.07%	X	X	X	X
Ours	**23.95%**	**6.15%**	10.14%	6.41%		
Ours (Unified)	27.11%	7.20%	10.72%	6.92%	X	X

Table 6: **Additional continuous partial observation reconstruction.** Quantitative ℓ_2 relative errors of various methods on the dynamic Allen–Cahn equation and the static Helmholtz equation.

	Allen–Cahn		Helmholtz	
	Forward	Inverse	Forward	Inverse
DiffusionPDE	1.43%	3.41%	2.3%	4.0%
PINO	1.39%	2.70%	4.9%	4.9%
Ours	**0.55%**	**1.18%**	**0.47%**	**3.87%**
Ours (Unified)	0.90%	1.42%	1.07%	4.84%

Table 7: **Additional forward/inverse full observation.** Average ℓ_2 relative errors of baseline methods for forward and inverse subtasks on ACE and Helmholtz datasets.

Forward/Inverse Full Observation Similarly, we evaluate the results of both forward and inverse problems for these additional datasets. In the context of static PDEs, the forward problem involves predicting the solution space given the coefficient space, whereas the inverse problem entails inferring the coefficient space from the observed solution space. Table 7 shows that VideoPDE outperforms all baselines on fully observed forward and inverse problems across all PDE families, including both dynamic and static cases.

Forward/Inverse Partial Observation We further investigate the forward and inverse problems using 3% observation points on the respective side. In Table 8, we show that VideoPDE has a lower error regarding both forward and inverse processes compared with DiffusionPDE. VideoPDE tends to generate more accurate results, as present in Figure 9.

Experiment Details We provide additional details on the datasets and their processing in the supplementary. Training is performed on 4 NVIDIA L40S GPUs with a batch size of 8 per GPU, taking approximately 24 hours per model. All models are trained until convergence using the Adam optimizer with a constant learning rate schedule (initial LR 5×10^{-4}). Our HV-DiT architecture operates directly in pixel space. Videos are tokenized into $4 \times 4 \times 2$ patches, forming a $32 \times 32 \times 10$ token grid with embedding dimension 384 for WL and NS. The model uses 2 transformer layers with neighborhood attention (window size $7 \times 7 \times 2$) and a downsampling operation via patch merging with factor 2. We provide more details on the model architecture and hyperparameters in the supplementary.

C Uncertainty Quantification

We illustrate uncertainty estimation on a single scene. Starting from a partial first frame, we ran diffusion inference ten times with different random seeds. Per-pixel uncertainty was computed as the standard deviation across samples, normalized by the absolute mean (relative std) and clipped for stability. Figure 10 shows relative std maps at $t = 5, 10, 15, 20$, where uncertainty grows over time and concentrates in dynamically complex regions.

	Allen–Cahn		Helmholtz	
	Forward	Inverse	Forward	Inverse
DiffusionPDE	15.29%	15.83%	8.8%	22.6%
Ours	<u>14.68%</u>	<u>13.48%</u>	**2.32%**	<u>11.02%</u>
Ours (Unified)	**13.27%**	**12.64%**	<u>2.45%</u>	**8.97%**

Table 8: **Additional forward/inverse 3% observation.** Average ℓ_2 relative errors of baseline methods for forward and inverse subtasks on ACE and Helmholtz datasets.

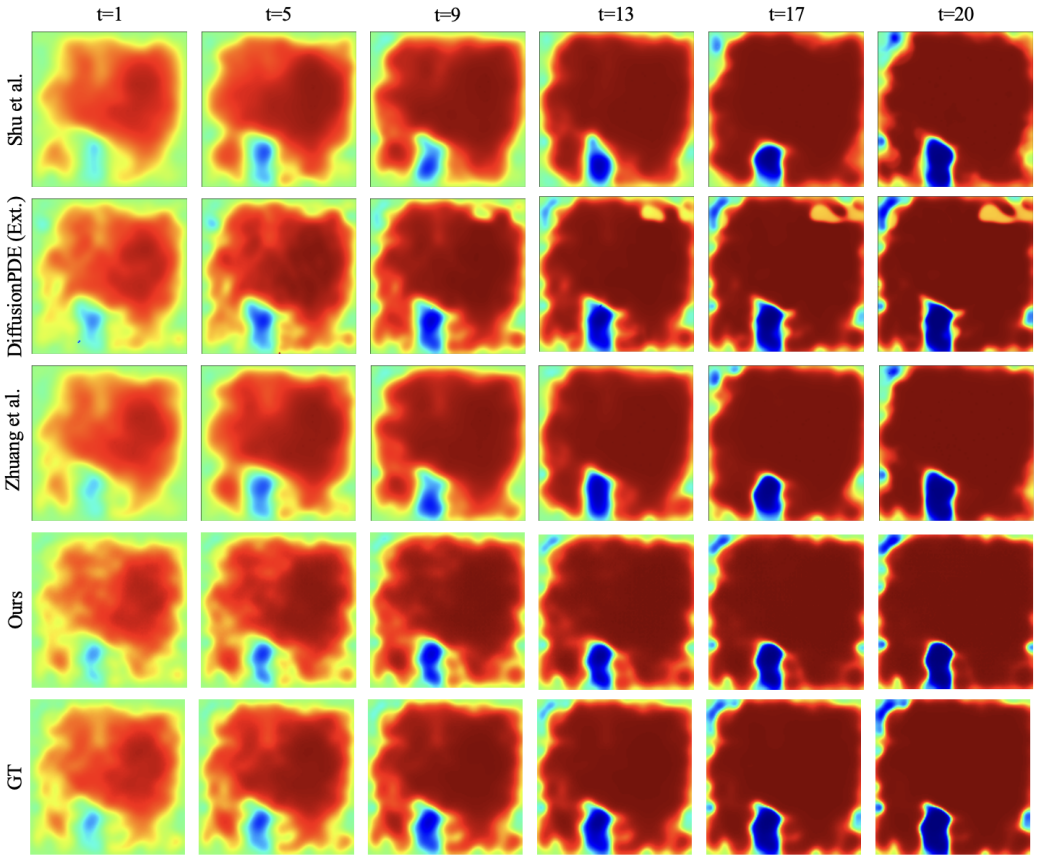


Figure 8: **Reconstruction results of ACE dataset with 1% observation points.** Visualizations of all methods for the continuous partial observation task.

D DiffusionPDE Extension

In this section, we explain how we extend the DiffusionPDE framework (31) to address dense temporal predictions by introducing a two-model architecture tailored for the continuous partial observation setting, as illustrated in Figure 11. Specifically, we employ a step model and a leap model to learn the joint distributions over adjacent timesteps, $P(t-1,t)$, and between the initial state and an arbitrary timestep, $P(1,t)$, respectively, where $t = 2, 3, \ldots, T$ is the index of the timestep. Both models are explicitly conditioned on the timestep t. The final score for $P(t)$ is computed by averaging the denoised estimates from the two pre-trained models, as demonstrated in Algorithm 1, thereby integrating both local and long-range temporal information.

This method can also be extended to both forward and inverse problems where the only observation is on the initial or final frame. However, rather than denoising all frames simultaneously, these tasks required an autoregressive approach to operate reasonably, which significantly increases computational cost. As a result, we do not include forward/inverse predictions with this extended model.

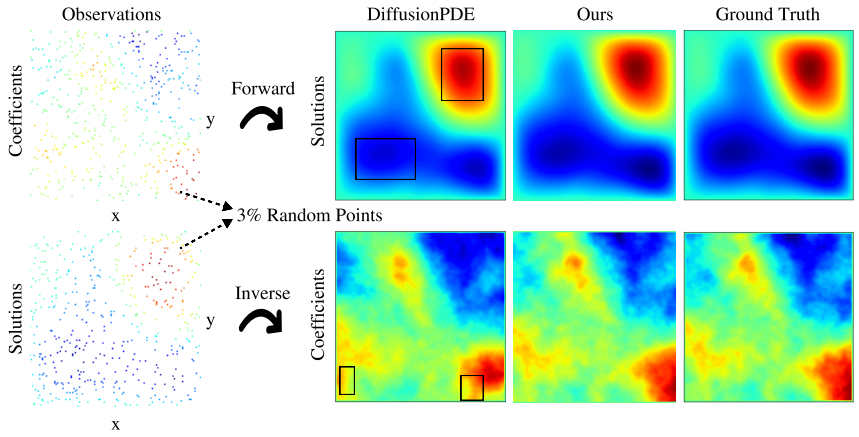


Figure 9: **Comparison of partially-observed forward/inverse predictions on Helmholtz equation.** Results of different methods for forward/inverse from partial observations (3%) are compared along with the ground truth fields. Notable errors produced by DiffusionPDE are highlighted with black squares for emphasis.

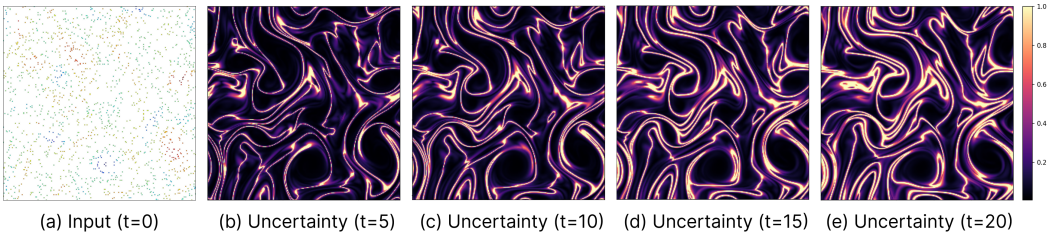


Figure 10: **Relative uncertainty visualization.** Starting from a sparse first frame (a), we run 10 stochastic diffusion predictions and compute per-pixel relative standard deviation (std/|mean|). Panels (b–e) show clipped uncertainty maps at $t = 5, 10, 15, 20$, highlighting growing uncertainty in dynamically complex regions.

E Multi-modality and Robustness Evaluation

To study the stochasticity and robustness of VideoPDE, we present reconstructions generated using different initial noise realizations of the diffusion model in Figure 12. Given a fixed observation mask, VideoPDE produces diverse reconstructions, particularly in regions lacking observations, which aligns with the physics property. Despite this variability, the model demonstrates strong robustness, as the reconstructed solutions and corresponding error metrics remain consistent across different noise seeds.

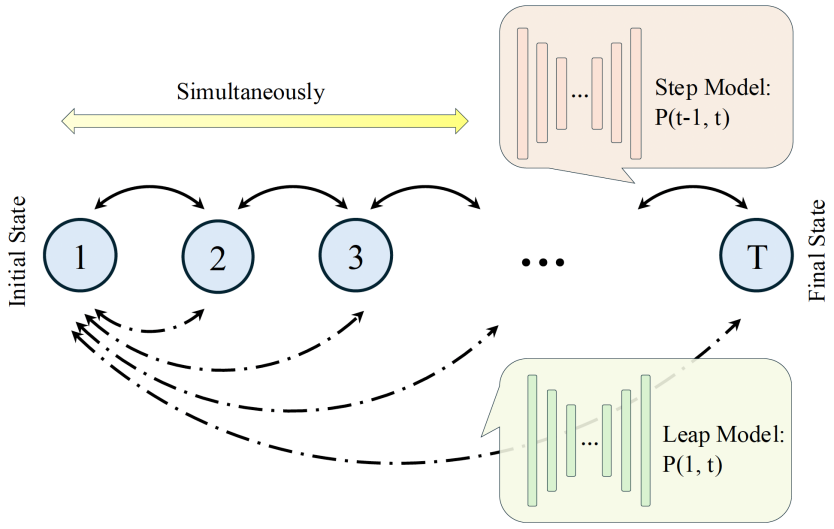


Figure 11: **Extension framework of DiffusionPDE.** The step model for distributing two adjacent frames and the leap model for distributing the initial state alongside one random timestep jointly denoise all time steps in a simultaneous manner.

Algorithm 1 DiffusionPDE (Ext.) Joint Denoising Algorithm.

input StepDenoiser $D_S(t)$, LeapDenoiser $D_L(t)$, VideoLength T, Timestep $t = 1, 2, \ldots, T$, TotalIterationCount N
for $i \in \{0, \ldots, N-1\}$ **do**
  for $j \in \{2, \ldots, T\}$ **do**
    $x_S^{j-1}, x_S^j \leftarrow D_S(j)$ ▷ Denoise the step model at timestep j
    $x_L^1, x_L^j \leftarrow D_L(j)$ ▷ Denoise the leap model at timestep j
    $x_j \leftarrow (x_S^j + x_L^j)/2$ ▷ Average the scores
    $x_{j-1} \leftarrow (x_S^{j-1} + x_L^{j-1})/2$
    {...Further guided sampling steps...}
  end for
end for
return $x_1, \ldots, x_T$

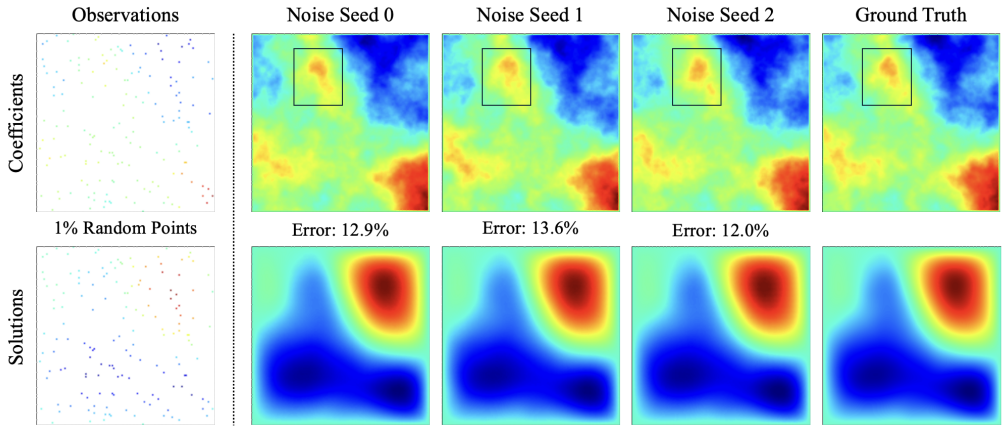


Figure 12: **Helmholtz reconstructions with different initial noise seeds.** Results and corresponding ℓ_2 relative errors obtained under identical 1% observation on both the coefficient and solution domains. Significant differences are emphasized with black squares.

F Error Rate versus Observation Rate

We present a chart illustrating the relationship between the error rate and observation rate for the continuous partial observation task, as shown in Figure 13. The results demonstrate that our methods are can achieve relative ℓ_2 errors below 10% using only 3% of observation points across all PDE families. For PDEs with high spatial information, e.g., KF, or diffusive behavior, e.g., Allen-Cahn, the performance degrades relatively steeply as the observation ratio decreases.

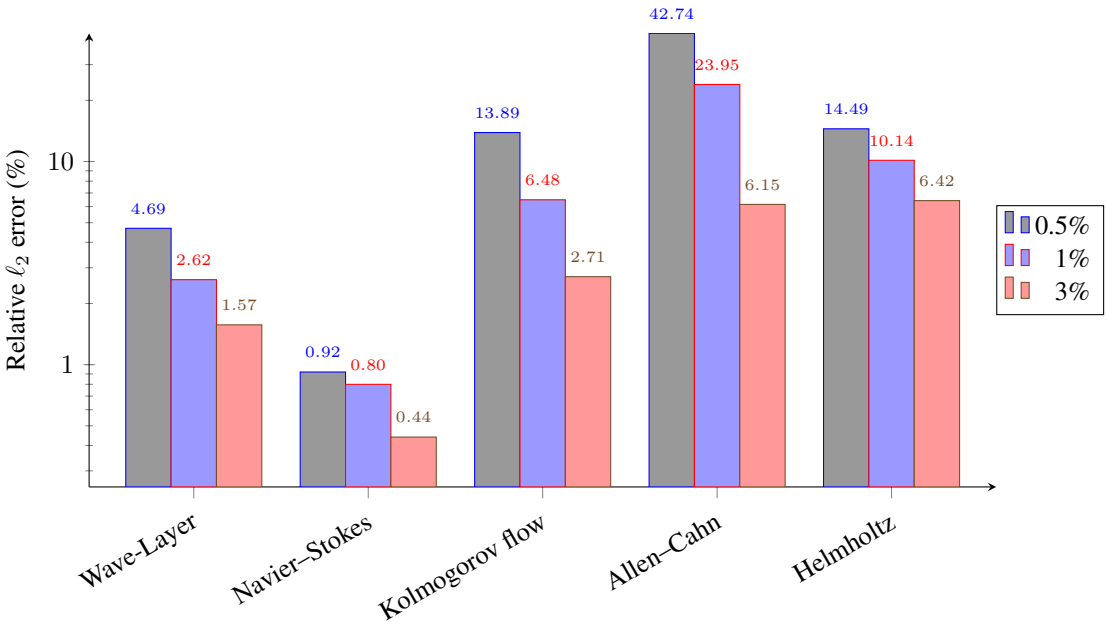


Figure 13: **Continuous partial observation comparison.** We report relative ℓ_2 errors for VideoPDE at 0.5%, 1%, and 3% observation rates.

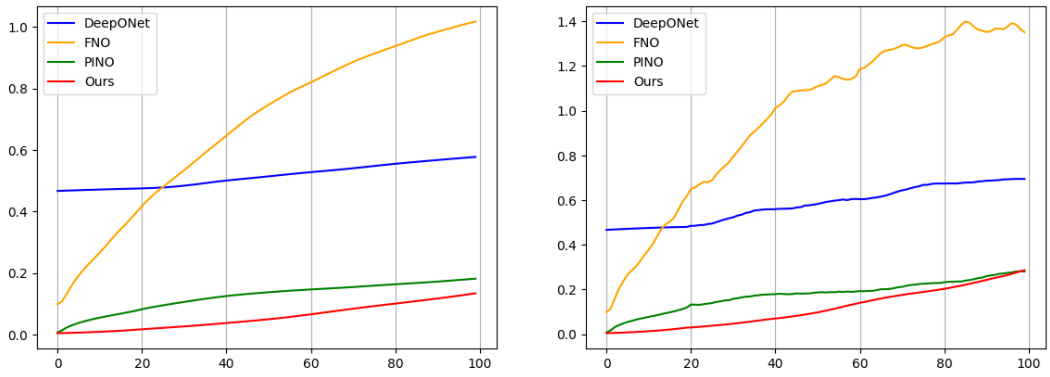
G Long Range Prediction

We further assess the performance of all models over an extended time horizon. Specifically, we employ autoregressive prediction to generate 100 future frames of the KF dataset using a 20-frame forward model. The sole input provided is the complete observation of the initial frame. Each 20-frame window is conditioned on the final frame of the previous window. The windows are concatenated to produce the final long trajectory.

In Fig. 14, we report the overall average errors as well as the single-frame errors for all evaluated models. The results indicate that VideoPDE achieves better predictive performance compared to baseline approaches. Nonetheless, a notable accumulation of error is observed for all methods, which may be attributed to the inherent uncertainty and difficulty of predicting long-range future and the absence of explicit PDE constraints for our method. Addressing this limitation remains an important direction for future research. We refer readers to the attached HTML file to view the videos of the 100-frame predictions.

H Further Ablations

We revisit the Wave-Layer dataset to conduct detailed ablations that study the sensitivity and impact of individual parameter choices on the model's overall performance. Specifically, we examine the influence of varying the number of frames, timestep sizes, and frames per second (FPS) within fixed physics intervals.



(a) Average errors of the first n frames.

(b) Errors of each single frame.

Figure 14: **Long-range prediction results.** The ℓ_2 relative errors of all methods plotted against the total number of frames.

Number of frames We first investigate how the number of frames affects our model's performance. From the available 20 frames in our raw data, we select subsets consisting of 5, 10, and 20 consecutive frames. All other experimental settings, including model configuration, remain consistent. The results summarized in Table 9 demonstrate that increasing the number of frames leads to improved performance, highlighting the effectiveness of our method in capturing temporal dependencies and longer temporal PDE dynamics. Additionally, since the error is averaged over all time frames, it suggests that utilizing longer temporal sequences improves the model's capability to accurately inpaint initial frames by providing richer temporal context.

	5 frames	10 frames	20 frames
Ours	0.62%	0.52%	0.49%

Table 9: **Number of frames**.

Timestep size Next, we explore how variations in the timestep size between frames impact the model's performance. Specifically, we fix the total number of frames to 5 and alter the interval between frames to step sizes of 1, 2, and 4, corresponding respectively to physical simulation intervals of 0.05s, 0.1s, and 0.2s within the total 1-second simulation duration. Table 10 summarizes the observed model errors across these different timestep sizes: smaller timesteps lead to reduced errors as expected.

	step=1	step=2	step=4
Ours	0.49%	0.57%	0.94%

Table 10: **Timestep size**.

Frames per second (FPS) Finally, we study the effect of FPS by varying both the number of frames and the timestep sizes to represent the same total physics duration of 1 second. Specifically, we compare scenarios using 5 frames with a timestep of 4, 10 frames with a timestep of 2, and 20 frames with a timestep of 1. As shown in Table 11, increasing the FPS improves the model's accuracy, showing the importance of temporal granularity in accurately modeling PDE dynamics.

	FPS = 5	FPS = 10	FPS = 20
Ours	0.94%	0.60%	0.49%

Table 11: **Frames per second**.

I Architecture and Training Details

We report our architectural and training hyperparameters in Table 12.

Hyperparameter	Ours	Ours (Unified)
Parameters	118M	74M
Training steps	50k	100k
Batch size	48	64
GPUs	2 × L40S	4 × L40S
Mixed Precision	bfloat16	bfloat16
Patch Size ($T \times H \times W$)	[2, 4, 4]	[2, 4, 4]
Neighborhood Attention Levels	1	1
Global Attention Level	1	1
Neighborhood Attention Depth	2	2
Global Attention Depth	11	6
Feature Dimensions	[384, 768]	[384, 768]
Attention Head Dimension	64	64
Neighborhood Kernel Size ($T \times H \times W$)	[2, 7, 7]	[2, 4, 4]
Mapping Depth	1	1
Mapping Width	768	768
Dropout	0	0
Optimizer	AdamW	AdamW
Learning Rate	5×10^{-4}	5×10^{-4}
$[\beta_1, \beta_2]$	$[0.9, 0.95]$	$[0.9, 0.95]$
Epsilon	1×10^{-8}	1×10^{-8}
Weight Decay	1×10^{-2}	1×10^{-2}

Table 12: **Training and inference hyperparameters.**

J Additional Baseline Results

We further evaluate the performance of the ECI sampling framework (9), which is an effective method for handling hard-constrained systems, particularly under various boundary conditions. However, as illustrated in Figure 15, the model encounters difficulties when working with sparse observations. This issue may arise from inconsistent guidance within the domain, leading to challenges in global reconstruction, especially with extremely sparse input points. Furthermore, since the data does not have strong boundary conditions, this method has significant limitations for the tasks we discuss.

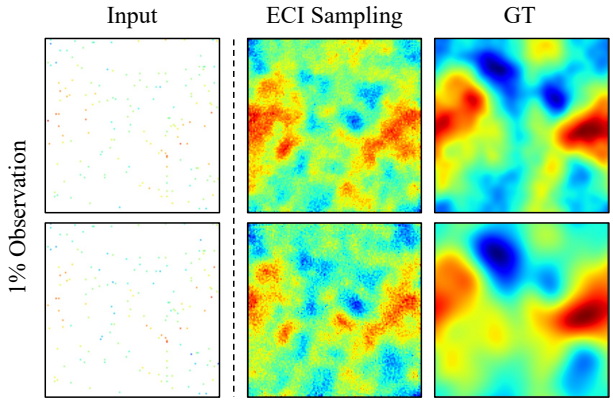


Figure 15: **ECI sampling results.** Reconstruction results with 1% observation points of the Navier-Stokes equation.

K LIMITATIONS AND FUTURE WORK

Our current evaluation relies on pixel-wise reconstruction loss, which may not fully reflect the physical accuracy of PDE solutions. Investigating more appropriate, PDE-specific evaluation metrics is an important direction for future work.

The present VideoPDE framework operates on 2D videos. Extending to 3D spatiotemporal domains would significantly broaden applicability, but will require more efficient architectural designs to handle the increased computational demands.

We also observe error accumulation in long-time-horizon predictions, which is common across all methods. This is likely due to the inherent uncertainty in extrapolating chaotic dynamics and the absence of explicit PDE constraints in our model. Mitigating this remains a key challenge.

Finally, our model currently operates on discrete, single-scale grid structures. While widely used in practice, enabling multi-scale or irregular discretization support, such as for adaptive or unstructured meshes, would further enhance the flexibility of our approach.