DiffusionPDE: Generative PDE-Solving Under Partial Observation

Jiahe Huang[1], Guandao Yang[2], Zichen Wang[1], Jeong Joon Park[1]

[1]University of Michigan [2]Stanford University

Motivation: PDE Solver

Predict future states of a system (forward process) and estimate underlying physical properties from state measurement (inverse process)

Darcy Flow (Static System)

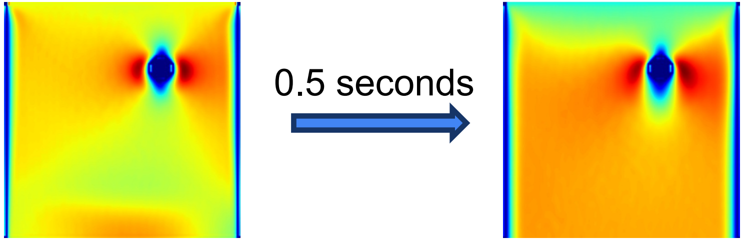
$$-\nabla \cdot (\boldsymbol{a}(\boldsymbol{c})\nabla \mathbf{u}(\boldsymbol{c})) = q(\boldsymbol{c}),$$
$$\mathbf{u}(\boldsymbol{c}) = 0,$$

In fluid flow through porous media, a is the material's permeability. u is the pressure field whose gradient drives the flow.

Navier-Stokes (Dynamic System)

$$\partial_t u(\boldsymbol{c},\tau) + u(\boldsymbol{c},\tau)\cdot\nabla u(\boldsymbol{c},\tau) + \frac{1}{\rho}\nabla p = \nu\nabla^2 u(\boldsymbol{c},\tau),$$
$$\nabla \cdot u(\boldsymbol{c},\tau) = 0,$$

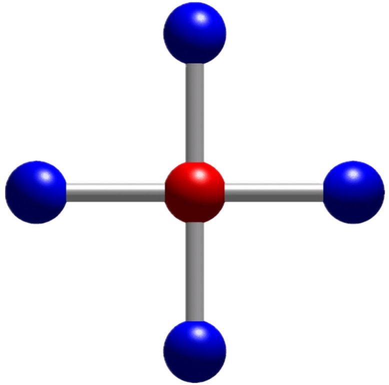
u is the velocity field of the flow.



Numerical Solver

Idea: discretize the space and solve the linear system.

e.g. Finite differential method (FDM): approximate the derivative with neighbors



5-point stencil for Laplacian:

$$\nabla^2 u(x,y) \approx \frac{u(x-h,y) + u(x+h,y) + u(x,y-h) + u(x,y+h) - 4u(x,y)}{h^2}$$

Numerical solvers are accurate but expensive.

Physics-Informed Neural Network (PINNs)

Idea: optimize a neural network using PDE constraints as self-supervised losses to output the PDE solution.

Given PDE function $f(x) = 0$,

MSE_u: The difference between prediction and ground truth

MSE_f: The difference between $f(x)$ and 0, ensuring physical property

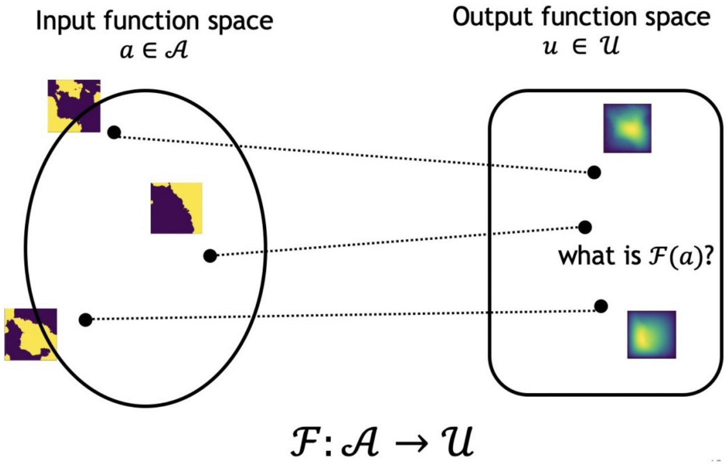
Total loss $MSE_{total} = MSE_u + MSE_f$

PINNs are simple and can be applied to various PDE families, but they are less accurate and hard to optimize.

Neural Operator

Idea: directly map between coefficient (or initial state) space and solution (or final state) space.

e.g. FNO (map on the Fourier space)



Li et al, 2021

Limitation of Prior Work

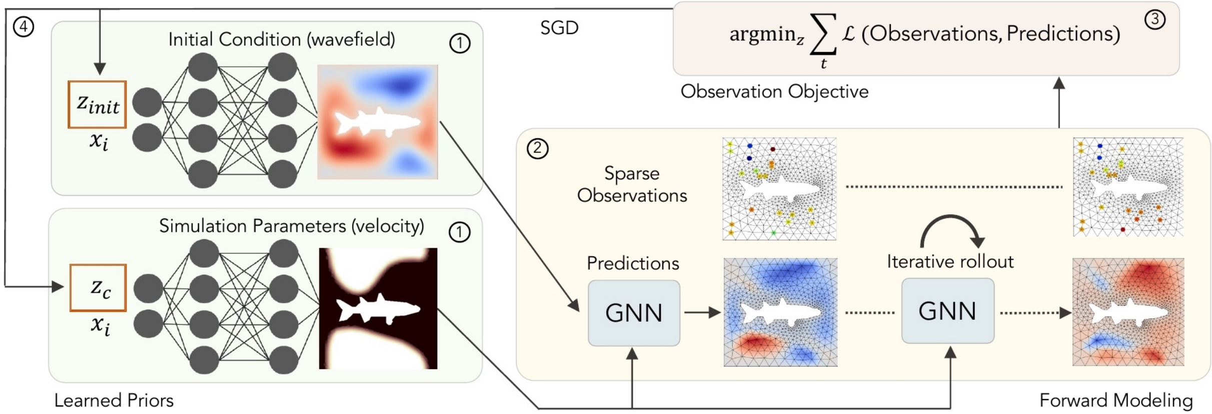
They assume full observation on the coefficient (or initial state) space for the forward process, and full observation on the solution (or final state) space for the inverse process.

In the real world, however, only partial data is likely observed.

Inverse PDE Solvers Under Partial Observation

Idea: leverage generative priors.

e.g. GraphPDE: learns a bounded forward GNN model and latent space model.

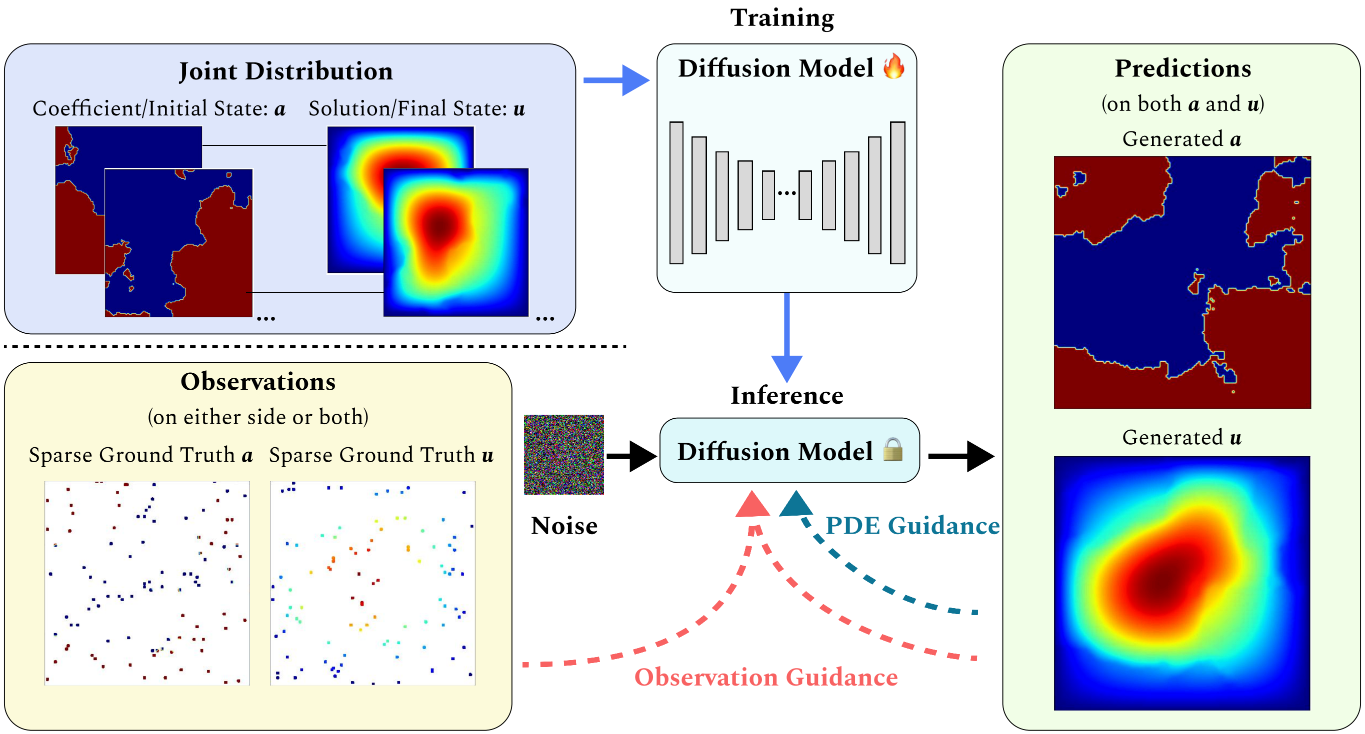


Zhao et al, 2022

The GNN prior faces challenges when solving high-resolution PDEs.

The autodecoder of the latent space model is also weaker than diffusion models.

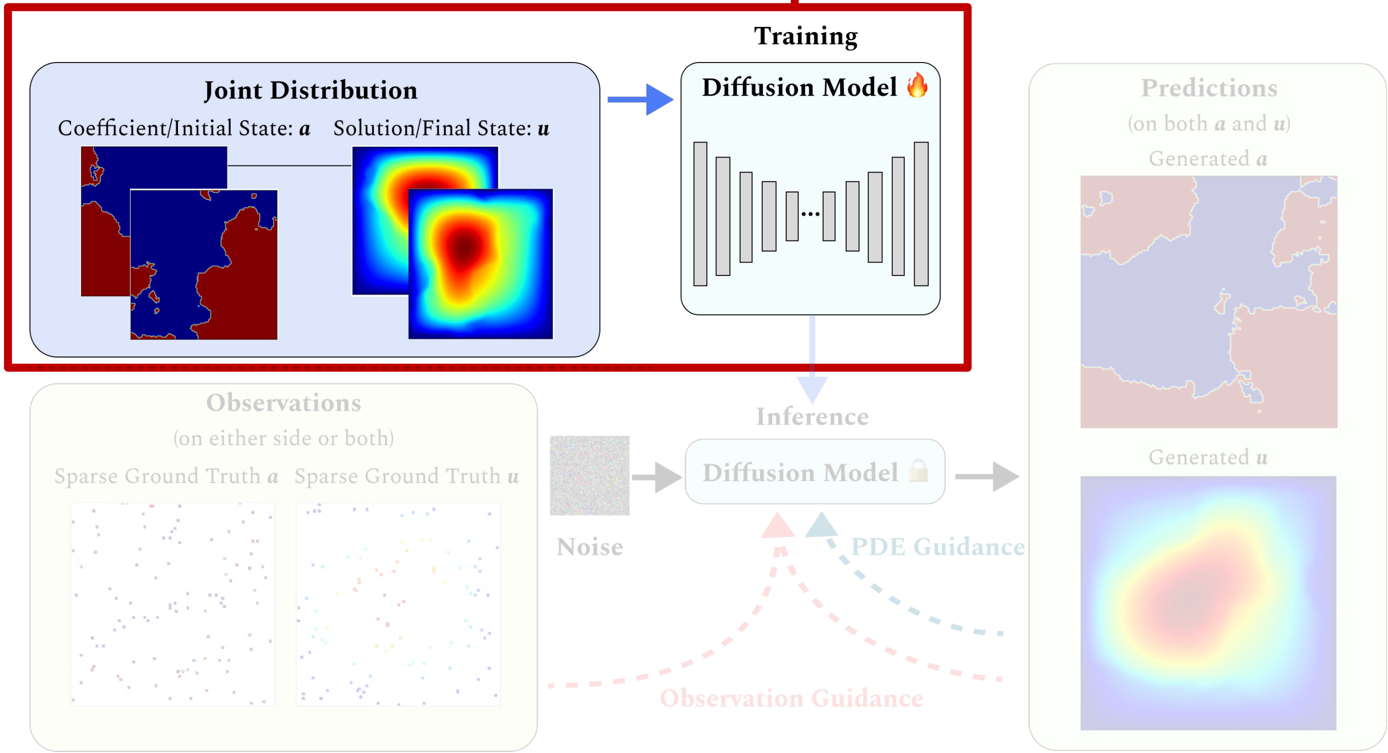
DiffusionPDE: Using the Diffusion Model as the Prior



DiffusionPDE Pipeline

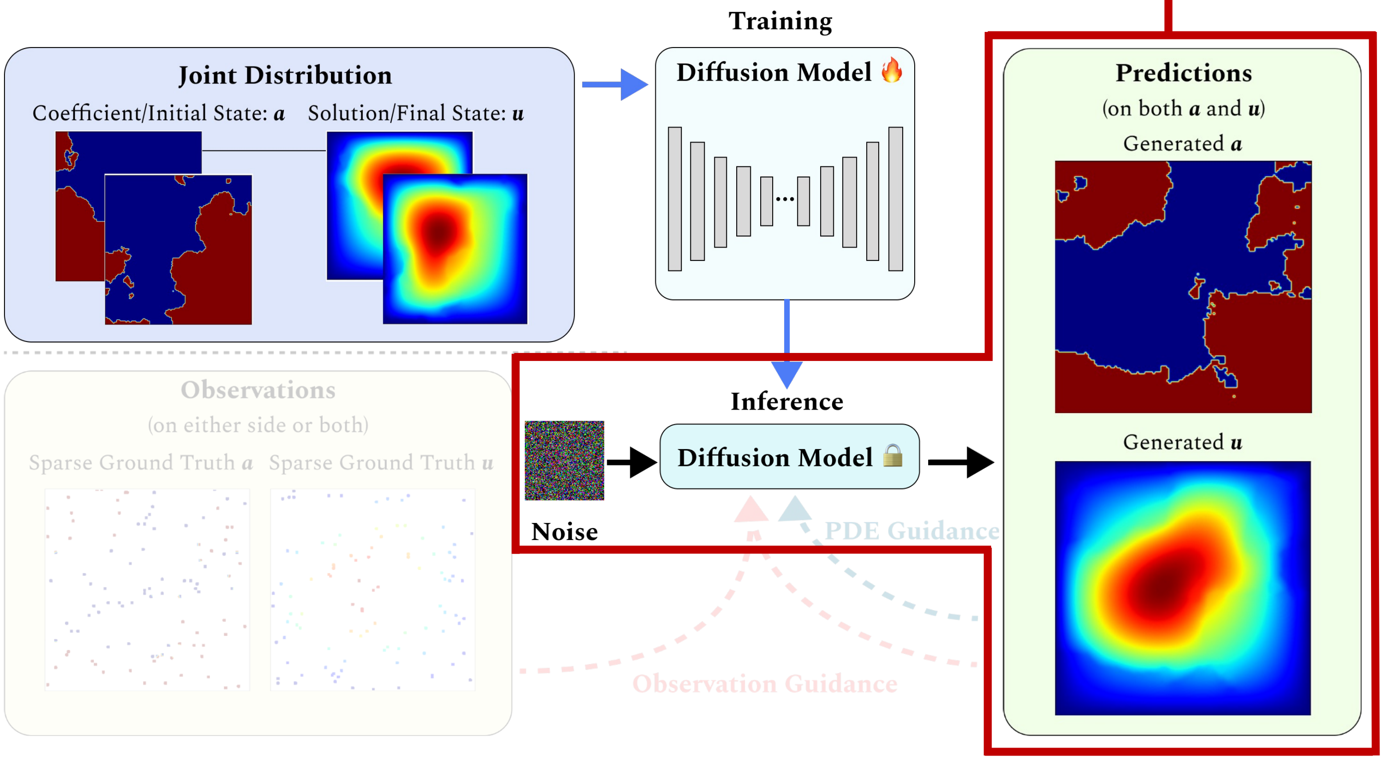
Train the diffusion model on the joint distribution of a and u (concatenated on the channel dimension)

Step 1.



DiffusionPDE Pipeline

Gradually denoise the random noise during the inference

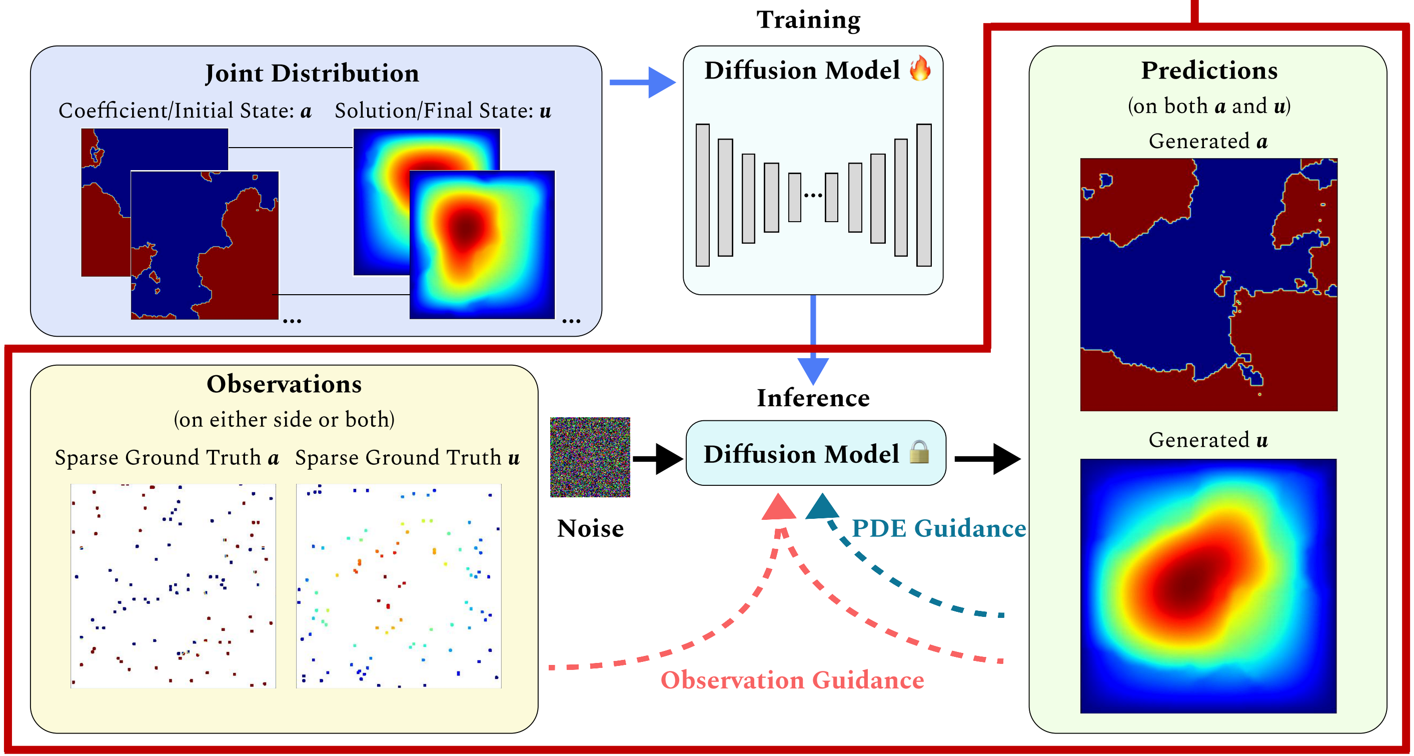


Step 2.

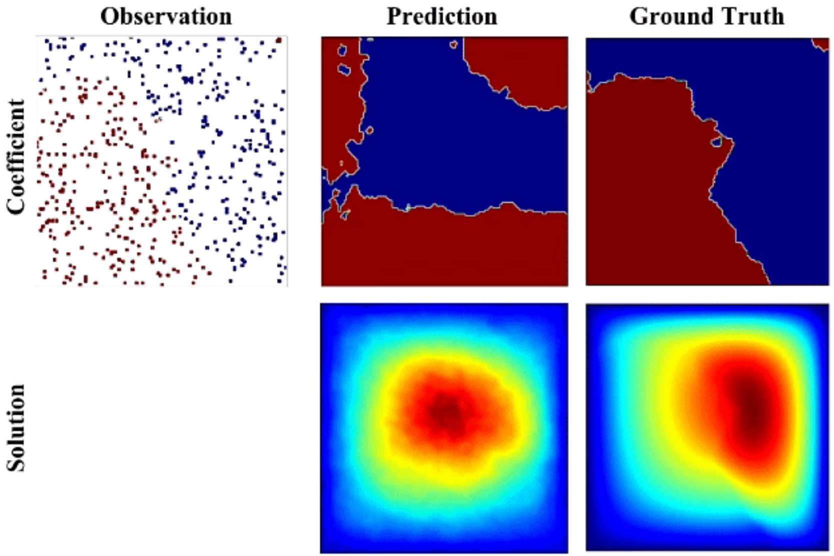
DiffusionPDE Pipeline

Guide the sampling with sparse observation and known PDE function

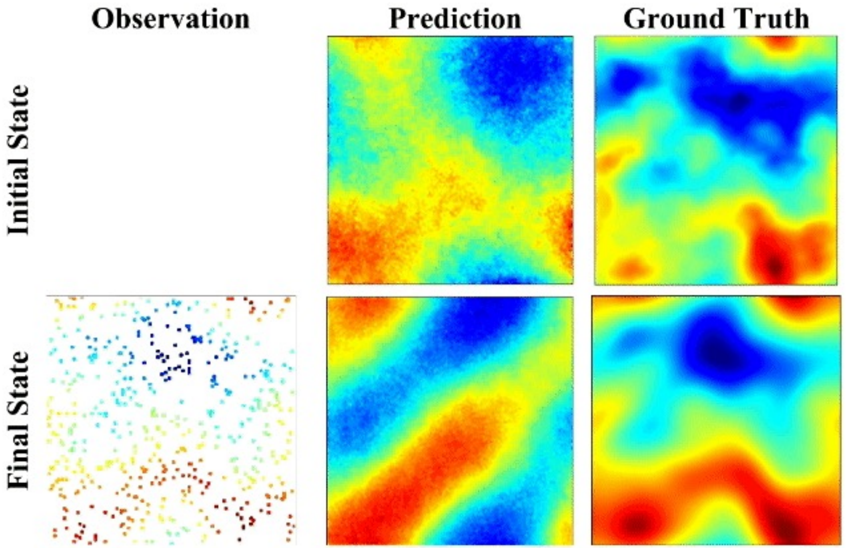
Step 3.



DiffusionPDE Denoising Process



Darcy Flow



Navier-Stokes Equation

Sparse Observation & PDE Guided Sampling

Idea: guide the sampling at each step i with corresponding distribution x_i

$$\nabla_{\boldsymbol{x}_i} \log p(\boldsymbol{x}_i | \boxed{\boldsymbol{y}_{obs}}, \boxed{f}) \approx \boxed{\nabla_{\boldsymbol{x}_i} \log p(\boldsymbol{x}_i)} - \zeta_{obs} \nabla_{\boldsymbol{x}_i} \boxed{\mathcal{L}_{obs}} - \zeta_{pde} \nabla_{\boldsymbol{x}_i} \boxed{\mathcal{L}_{pde}}$$

Observed Values

PDE Condition: $f(\cdot) = \mathbf{0}$

Pre-trained Diffusion Prior

Observation Loss

PDE Loss

Observation loss:

$\hat{\boldsymbol{x}}_N^i = D_\theta(\boldsymbol{x}_i)$: Clean image estimated at step i

$$\mathcal{L}_{obs}(\boldsymbol{x}_i, \boldsymbol{y}_{obs}; \boxed{D_\theta}) = \frac{1}{n} \|\boldsymbol{y}_{obs} - \boxed{\hat{\boldsymbol{x}}_N^i}\|_2^2$$

n: number of observation points

Pre-trained denoiser

PDE loss:

$$\mathcal{L}_{pde}(\boldsymbol{x}_i; D_\theta, f) = \frac{1}{m} \|\mathbf{0} - f(\hat{\boldsymbol{x}}_N^i)\|_2^2$$

m: number of total pixels on the image

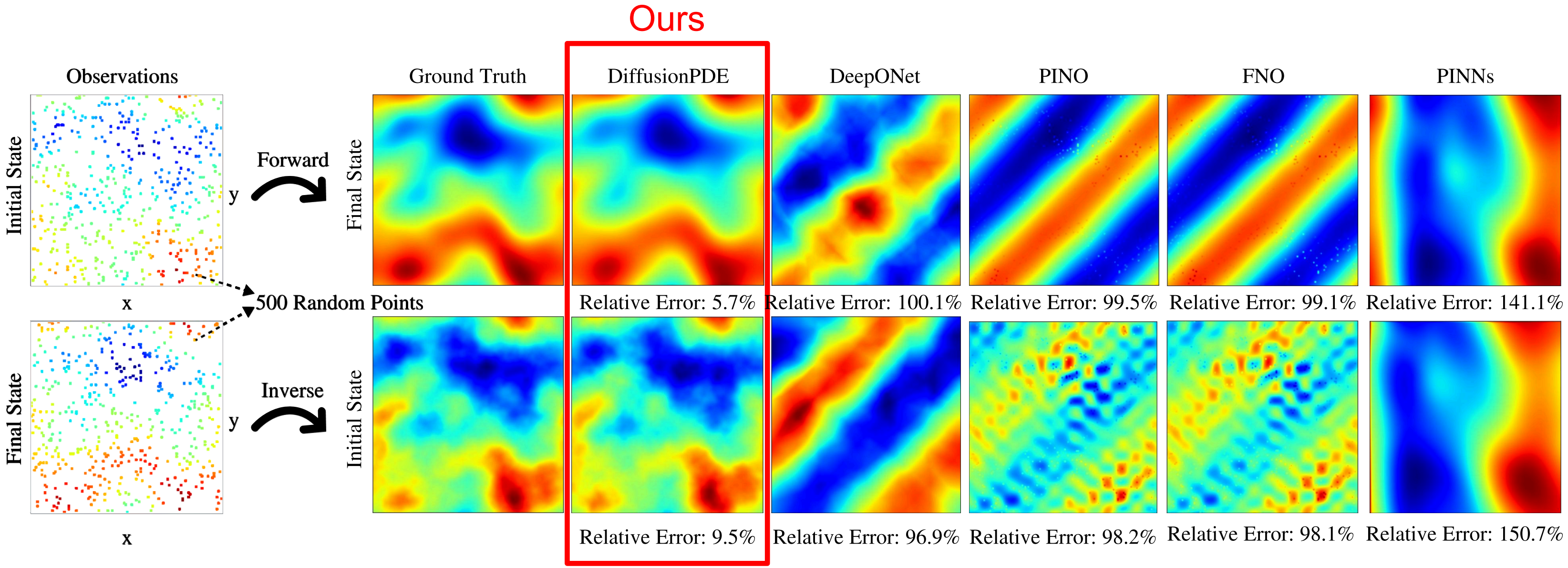
Main Evaluation Results

- Solve both forward problems and inverse problems under sparse observation.

		DiffusionPDE	PINO	DeepONet	PINNs	FNO
Darcy Flow	Forward	**2.5%**	35.2%	38.3%	48.8%	28.2%
	Inverse	**3.2%**	49.2%	41.1%	59.7%	49.3%
Poisson	Forward	**4.5%**	107.1%	155.5%	128.1%	100.9%
	Inverse	**20.0%**	231.9%	105.8%	130.0%	232.7%
Helmholtz	Forward	**8.8%**	106.5%	123.1%	142.3%	98.2%
	Inverse	**22.6%**	216.9%	132.8%	160.0%	218.2%
Non-bounded Navier-Stokes	Forward	**6.9%**	101.4%	103.2%	142.7%	101.4%
	Inverse	**10.4%**	96.0%	97.2%	146.8%	96.0%
Bounded Navier-Stokes	Forward	**3.9%**	81.1%	97.7%	100.1%	82.8%
	Inverse	**2.7%**	69.5%	91.9%	105.5%	69.6%

Main Evaluation Results

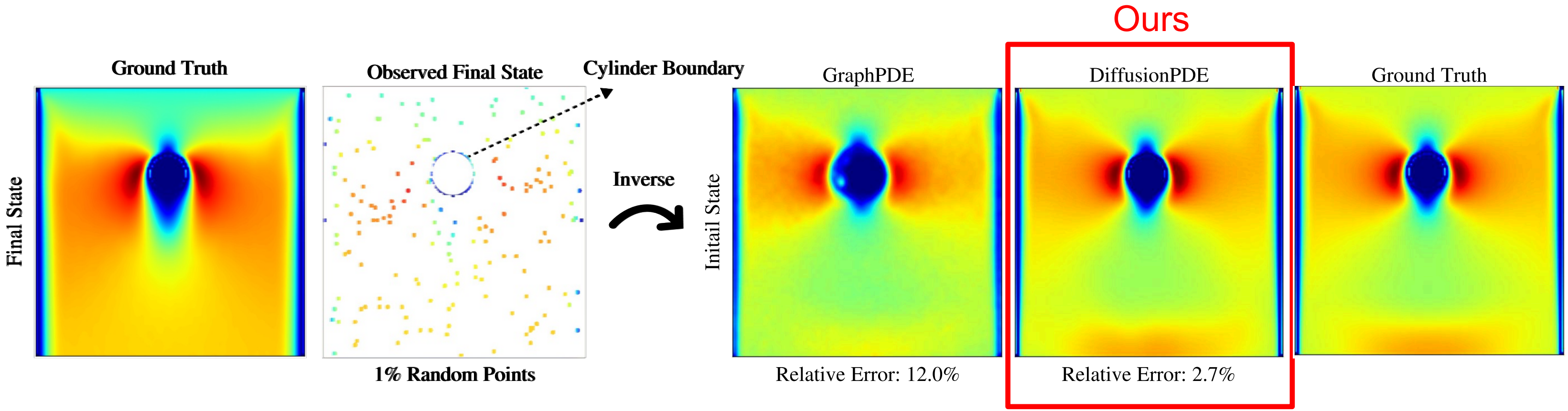
- Solve both forward problems and inverse problems under sparse observation.



Non-bounded Navier-Stokes equation on the vorticity field.

Main Evaluation Results

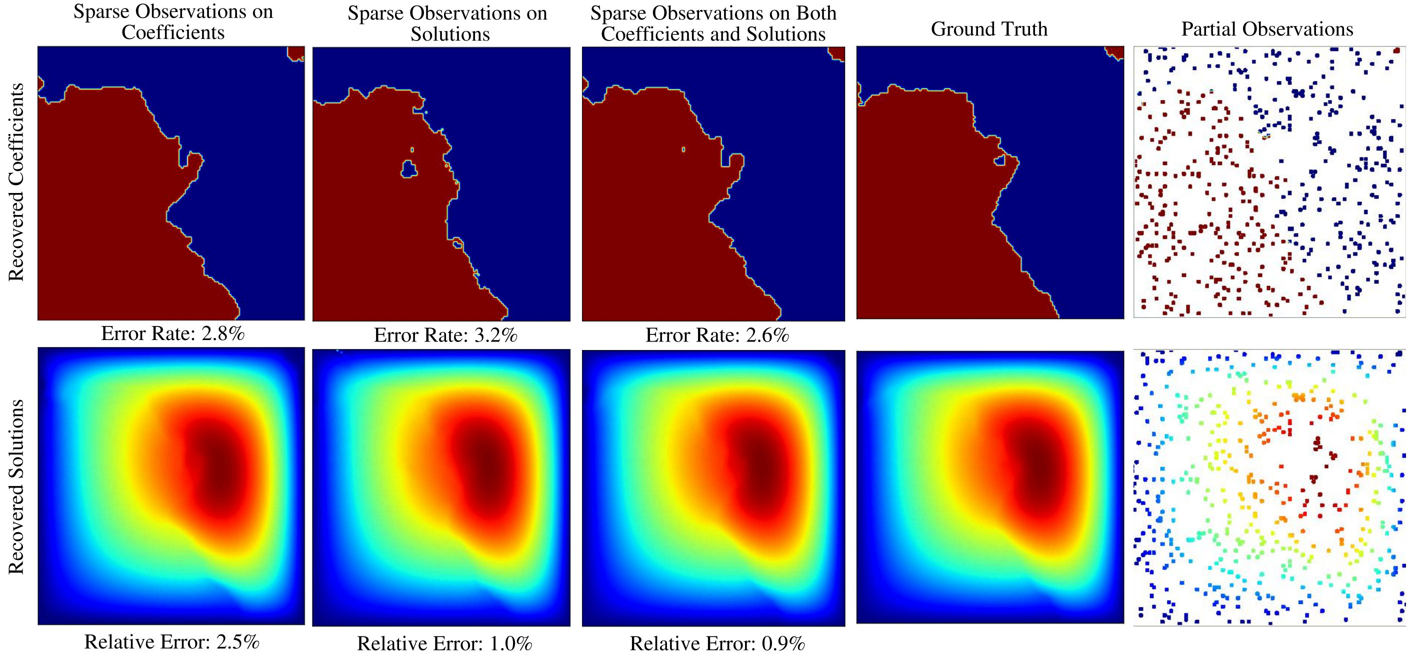
- Solve higher-resolution (128×128) inverse problems under sparse observation using different priors.



Bounded Navier-Stokes equation on the velocity field with 2D cylinder obstacle of random radius at random location.

Main Evaluation Results

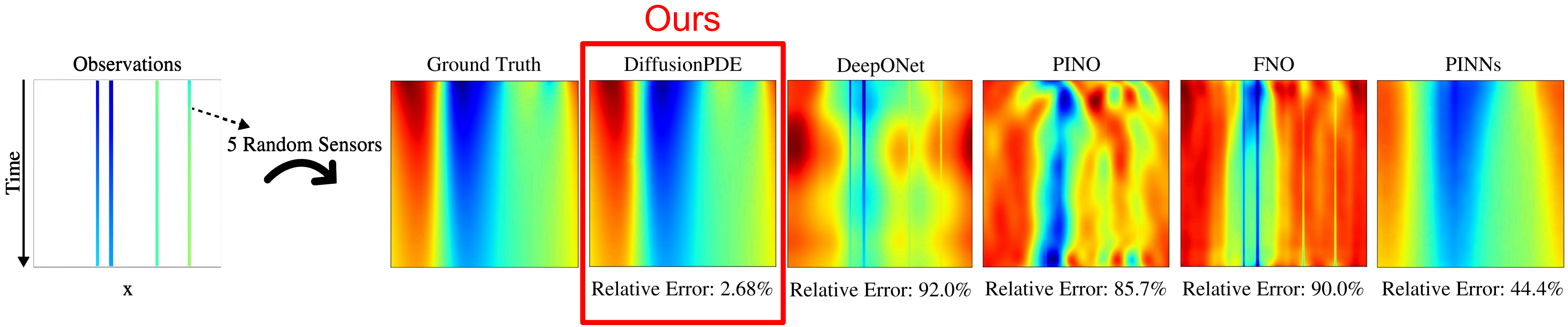
- Recover a and u simultaneously with sparse observation at any side.



2D Darcy Flow equation with no-slip boundary.

Recovering Solutions Throughout a Time Interval

- Recover $u_{0:T}$ from continuous observations on sparse sensors.



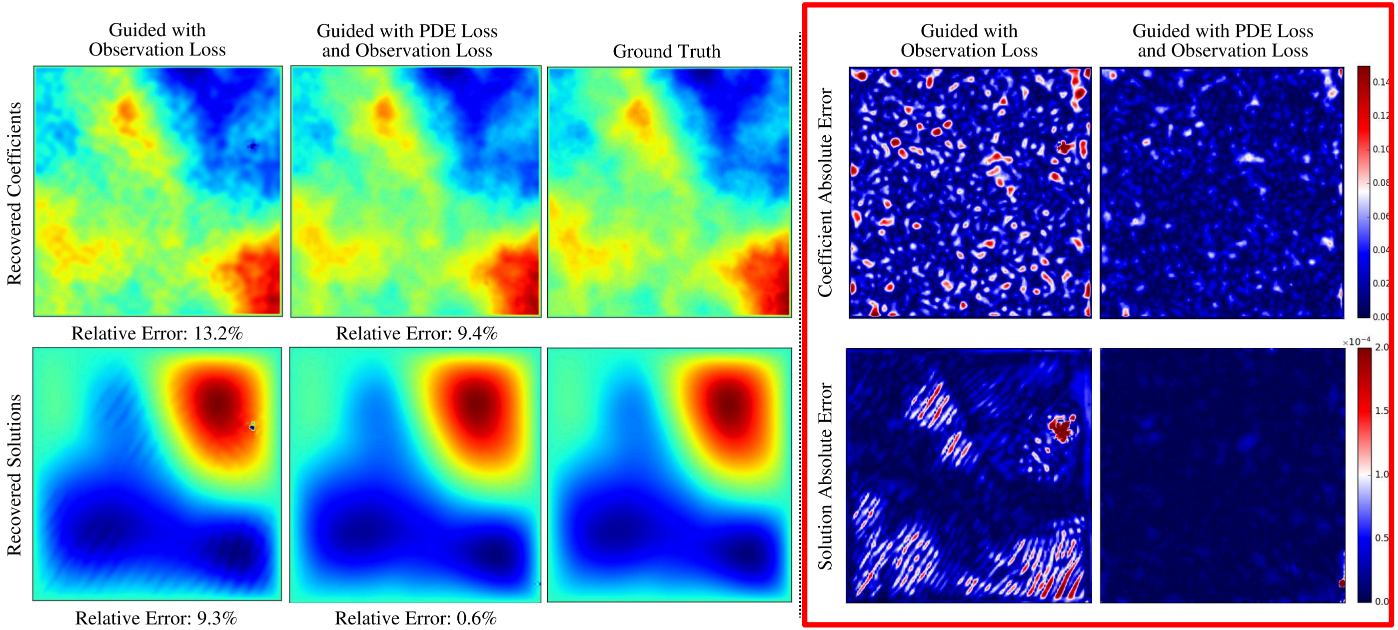
1D Burgers' equation with periodic boundary conditions.

Additional Analysis

$$\mathcal{L}_{obs}(\boldsymbol{x}_i, \boldsymbol{y}_{obs}; D_\theta) = \frac{1}{n}\|\boldsymbol{y}_{obs} - \hat{\boldsymbol{x}}_N^i\|_2^2$$

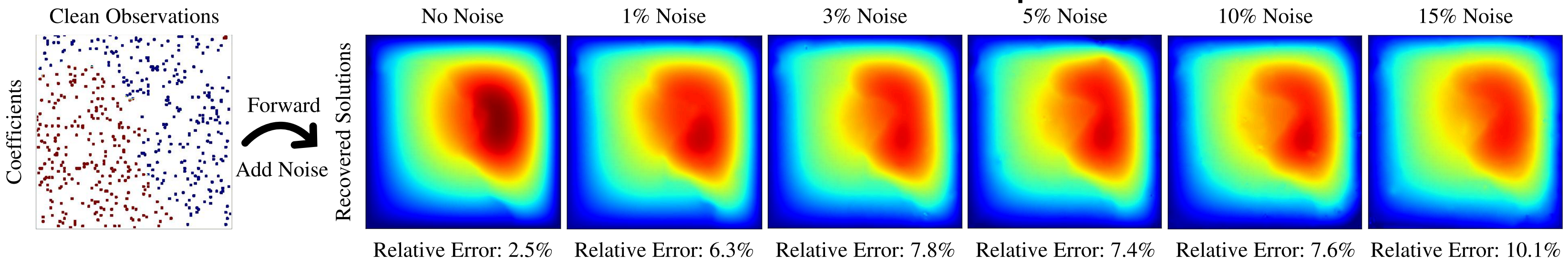
$$\mathcal{L}_{pde}(\boldsymbol{x}_i; D_\theta, f) = \frac{1}{m}\|\mathbf{0} - f(\hat{\boldsymbol{x}}_N^i)\|_2^2$$

- DiffusionPDE improves its performance with both PDE and observation guidance rather than only with observation guidance.

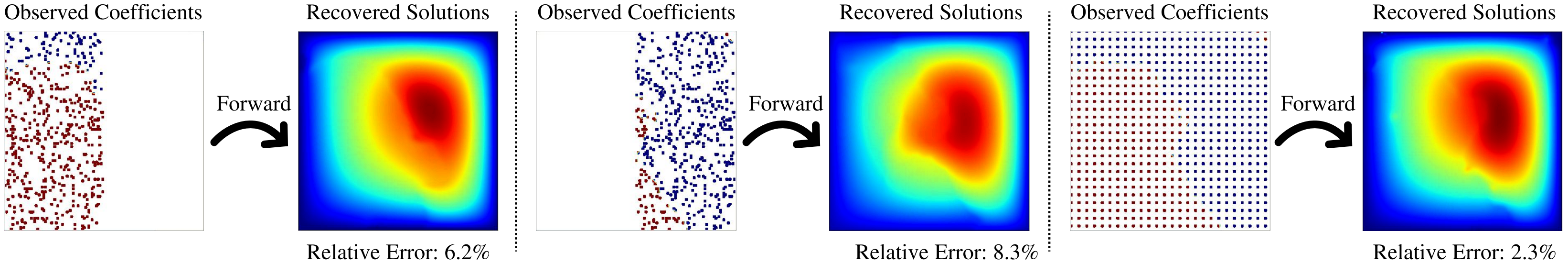


Additional Analysis

- DiffusionPDE is robust to random noise on the sparse observation.



- DiffusionPDE is robust to different sampling patterns of the sparse observation.



Conclusion

- DiffusionPDE can solve forward and inverse processes simultaneously.
- DiffusionPDE can recover coefficient and solution spaces with very sparse observation, outperforming SOTA methods.
- DiffusionPDE highlights the opportunity of solving PDEs with one single generative model.

Thank you!