Diffusion-Based Inverse Solver on Function Spaces With Applications to PDEs

Abbas Mammadov[1] **Julius Berner**[2] **Kamyar Azizzadenesheli**[3]

Jong Chul Ye[1] **Anima Anandkumar**[2]

[1] KAIST [2] Caltech [3] NVIDIA

Abstract

We present a novel framework for solving inverse problems in function spaces using diffusion-based generative models. Unlike traditional methods, which often discretize the domain and operate on fixed grids, our approach is discretization-agnostic, allowing for flexibility during sampling and generalization across different resolutions. Built upon function space diffusion models with neural operator architectures, we adapt the denoising process of pre-trained diffusion models to efficiently sample from posterior distributions in function spaces. This framework can be applied to a variety of problems, such as recovery of initial conditions and coefficient functions in noisy or partially observed PDE-based inverse problems like Darcy flows and Navier–Stokes equations. To the best of our knowledge, this is the first diffusion-based plug-and-play solver for inverse problems that operates in a discretization-agnostic manner, providing a new perspective on inverse problems with functional data, as typically arising in the context of PDEs.

1 Introduction

Solving inverse problems is an essential task in applied mathematics and the computational sciences, dealing with the recovery of unknown signals from corrupted or incomplete measurements. The fundamental goal is to reconstruct an unknown signal $\boldsymbol{a}$ from an observed measurement $\boldsymbol{u} = \boldsymbol{A}(\boldsymbol{a}) + \epsilon$, where $\boldsymbol{A}$ is the so-called forward operator $\boldsymbol{A}$ and ϵ represents noise. Assuming a prior distribution $p(\boldsymbol{a})$ on the signal and a suitable distribution for the noise, we can take a Bayesian perspective on the inverse problem, where we seek to sample from the posterior distribution $p(\boldsymbol{a}|\boldsymbol{u})$ [1].

In recent years, diffusion models have emerged as a promising approach for learning the prior distribution $p(\boldsymbol{a})$, offering new possibilities for the solution of inverse problems [2, 3, 4, 5]. Various approaches have been proposed to leverage diffusion priors, ranging from guidance terms or resampling strategies within the generative process [4, 5, 6, 7] to integrations within variational frameworks [8, 9]. The former plug-and-play methods rely on adaptations of the generative process of diffusion models to push samples from the prior $p(\boldsymbol{a})$ to the posterior distribution $p(\boldsymbol{a}|\boldsymbol{u})$, for instance, by guiding the process towards the measurement subspace defined by $\{\boldsymbol{a}\colon \boldsymbol{A}(\boldsymbol{a}) = \boldsymbol{u}\}$ [5, 6, 7].

Plug-and-play conditional generation and the solution of inverse problems with diffusion models have been extensively studied in finite-dimensional spaces, such as fixed-resolution images. However, applying these techniques to inverse problems in the context of partial differential equations (PDEs) formulated in function spaces remains largely unexplored. A recent advancement in this direction is the DiffPDE method [10], which proposes learning the joint distribution between initial conditions or coefficient functions and the solutions of PDEs. While it demonstrated promising results in solving forward and inverse problems using sparse observations, DiffPDE is limited by its reliance on fixed

Machine Learning and the Physical Sciences Workshop, NeurIPS 2024.

2D discretizations. In particular, it employs image-domain diffusion models to solve inpainting problems on discretized spaces, although the underlying data is naturally given by functions [11].

In contrast, our approach operates directly on function spaces without relying on fixed resolutions. Discretization is employed solely for computational purposes, allowing for natural generalization across different resolutions. This is achieved by adapting existing diffusion models in function spaces [12, 13, 14] for plug-and-play conditional sampling. Diffusion models in function spaces differ in two crucial aspects from their finite-dimensional counterparts. First, these approaches replace the neural network used as a denoiser with a discretization-agnostic neural operator [15, 16, 17]. Second, the multivariate Gaussian noise used for training and generation is replaced by an appropriate concept of noise in function spaces, i.e., Gaussian random fields. We base our implementation on EDM-FS[1], which adapts successful techniques for diffusion models on finite-dimensional domains [18] to function spaces.

Motivated by these advancements in operator learning, we propose a new framework for posterior sampling with function space diffusion models. Our contributions can be summarized as follows:

- We develop a plug-and-play framework for solving inverse problems naturally formulated in function spaces. Our approach requires no additional training and only relies on a pre-trained diffusion model specifying the prior distribution.
- We validate the efficacy of our approach through various experiments on inverse problems in the context of Darcy flows and Navier–Stokes equations.
- We empirically demonstrate that our approach is discretization-agnostic, allowing for generalizations to resolutions different from the training resolution. In particular, we show that the diffusion model can be efficiently trained by predominately using low-resolution data.

2 Diffusion Based Inverse Solver on Function Spaces

We focus on general inverse problems in function spaces, with the goal of retrieving an unknown function $\boldsymbol{a} \in \mathcal{A}$ from a measurement function $\boldsymbol{u} \in \mathcal{U}$ given by

$$\boldsymbol{u} = \boldsymbol{A}(\boldsymbol{a}) + \epsilon \quad \text{with} \quad \boldsymbol{A} : \mathcal{A} \to \mathcal{U}. \tag{1}$$

In the above, $\mathcal{A} = \mathcal{A}(D; \mathbb{R}^{d_a})$ and $\mathcal{U} = \mathcal{U}(D; \mathbb{R}^{d_u})$ are separable Banach spaces of $\mathbb{R}^{d_a}$-valued and $\mathbb{R}^{d_u}$-valued functions on a bounded open set $D \subset \mathbb{R}^d$ and ϵ is an $\mathcal{U}$-valued random variable representing the measurement noise. In particular, we consider the Bayesian viewpoint, where our goal is to sample from the posterior distribution $p(\boldsymbol{a}|\boldsymbol{u})$.

To this end, we assume that we have access to a pre-trained (function space) diffusion model that can sample from the prior $p(\boldsymbol{a})$. Diffusion (probabilistic) models have demonstrated high quality and stability in generation tasks [2, 19, 20, 18]. They are based on defining a sequence of distributions $p_t(\boldsymbol{a}_t)$ by gradually adding noise to the data $\boldsymbol{a}$ until approximately reaching a tractable and easy-to-sample distribution $\boldsymbol{\Gamma}$. For suitable noise distributions, one can reverse this noising process and sample from the distribution $p(\boldsymbol{a})$ when having access to the conditional expectations $\mathbb{E}[\boldsymbol{a}|\boldsymbol{a}_t]$ [21, 14]. Since these conditional expectations are typically intractable, one approximates them using a denoiser $\boldsymbol{D}_\theta(\boldsymbol{a}_t, t)$ trained using a denoising score matching objective [2, 18]. For data $\boldsymbol{a}$ from infinite-dimensional function spaces, we parametrize $\boldsymbol{D}_\theta$ as a neural operator [12, 22, 23] and choose the noise as Gaussian random fields (as compared to neural networks and multivariate Gaussian random variables for the finite-dimensional setting).

Gradient guidance. Our focus is to adapt the generative process of a pre-trained diffusion model via an extension of gradient guidances in order to sample from the posterior distribution $p(\boldsymbol{a}|\boldsymbol{u})$ instead of the prior $p(\boldsymbol{a})$. Leveraging the Bayesian perspective, we decompose the posterior distribution of the noisy data into a prior and likelihood term during the denoising steps, i.e.,

$$p_t(\boldsymbol{a}_t|\boldsymbol{u}) \propto p_t(\boldsymbol{a}_t) p_t(\boldsymbol{u}|\boldsymbol{a}_t). \tag{2}$$

Using the pre-trained diffusion model, one can approximately reverse the noising process $p_t(\boldsymbol{a}_t)$ of the prior. Motivated by the gradient-based inverse solver DPS [5], we can consider a generalization of Tweedie's formula [24] to account for the effect of the likelihood term $p_t(\boldsymbol{u}|\boldsymbol{a}_t)$. In particular, we

[1] https://github.com/neuraloperator/cond-diffusion-operators-edm

Algorithm 1 Diffusion-Based Inverse Solver on Function Spaces

Require: $\boldsymbol{u}, \boldsymbol{A}, \boldsymbol{D}_\theta, \{t_i\}_{i=0}^{N}, \{\lambda_i\}_{i=0}^{N-1}$

$\boldsymbol{a}_0 \sim \boldsymbol{\Gamma}$ ▷ Initialize
$\boldsymbol{a} \leftarrow \boldsymbol{D}_\theta(\boldsymbol{a}_0, t_0)$
for $i = 0$ to $N-1$ **do** ▷ Generative process
  $\boldsymbol{d}_i \leftarrow (\boldsymbol{a}_i - \boldsymbol{a})/t_i$
  $\boldsymbol{a}_{i+1} \leftarrow \boldsymbol{a}_i + (t_{i+1} - t_i)\boldsymbol{d}_i$ ▷ Take Euler step from t_i to t_{i+1}
  if $t_{i+1} \neq t_N$ **then** ▷ Apply 2^{nd}-order correction
    $\boldsymbol{a}' \leftarrow \boldsymbol{D}_\theta(\boldsymbol{a}_{i+1}, t_{i+1})$
    $\boldsymbol{d}_i' \leftarrow (\boldsymbol{a}_{i+1} - \boldsymbol{a}')/t_{i+1}$
    $\boldsymbol{a}_{i+1} = \boldsymbol{a}_i + (t_{i+1} - t_i)(\frac{1}{2}\boldsymbol{d}_i + \frac{1}{2}\boldsymbol{d}_i')$
  end if
  $\boldsymbol{a}_{i+1} \leftarrow \boldsymbol{a}_{i+1} - \lambda_i \partial_{\boldsymbol{a}'} \|\boldsymbol{u} - \boldsymbol{A}(\boldsymbol{a}')\|_U^2$ ▷ Invoke the gradient guidance
  $\boldsymbol{a} \leftarrow \boldsymbol{a}'$ ▷ Cache denoised sample
end for
return $\boldsymbol{a}_N$

iteratively update the samples $\boldsymbol{a}_t$ using the consistency between the given measurement $\boldsymbol{u}$ and the one obtained from the denoiser, i.e., $\boldsymbol{A}(D_\theta(\boldsymbol{a}_t, t))$, leading to the update rule

$$\boldsymbol{a}_t \leftarrow \boldsymbol{a}_t - \lambda \partial_{\boldsymbol{a}_t} \|\boldsymbol{u} - \boldsymbol{A}(\boldsymbol{D}_\theta(\boldsymbol{a}_t, t))\|_U^2, \tag{3}$$

where λ is a predefined guidance strength and $\partial_{\boldsymbol{a}_t}$ corresponds to the Gateaux derivative. The value of λ is typically tuned depending on the given task and decreased towards the end of generation. Algorithm 1 summarizes the generative process, leveraging deterministic sampling using Heun's 2^{nd}-order method for the diffusion model [18].

A key aspect of our approach is that it is discretization-agnostic, naturally generalizing across different resolutions of $\boldsymbol{a}_t$. Using the guidance in (3) in combination with a function space diffusion model enables us to seamlessly handle inverse problems on function spaces without being constrained to a fixed grid or mesh. We note that one could even gradually increase the resolution during the generative process to trade off fidelity with computational costs.

Inverse problems on function spaces. We tackle three types of tasks in the context of inverse problems on function spaces. For the first type, we recover functions from corrupted or noised versions using forward operators similar to classical inverse problems in finite dimensions, except that we replace the Gaussian noise with a Gaussian random field. For the second type, we tackle PDE-based inverse problems, where our goal is to recover coefficient functions $\boldsymbol{a}$ from (partially observed or noisy) PDE solutions $\boldsymbol{u}$. For this task, we use a PDE solver based on a (differentiable) finite difference method (FDM) as forward operator $\boldsymbol{A}$ in (1). The third task is inspired by DiffPDE [10]. Specifically, we learn the joint distribution of initial and terminal states of time-dependent PDEs using a function space diffusion model. Using our proposed method, we leverage this diffusion prior to solve combinations of forward and inverse PDE problems from sparse observations of the initial and terminal states. In this case, the forward operator is described by the given observations and the consistency with the PDE, where the latter is formulated as a loss in physics-informed neural operators [25]. In particular, this allows us to remove the dependency on the costly PDE solver in the forward operator.

Multi-resolution pre-training. Inspired by other works on operator learning [26, 27], we introduce a new training technique to learn the prior distribution with reduced computational costs. We first train the diffusion model on low-resolution data for a majority of the epochs and only train on higher resolutions for the final epochs. This curriculum learning approach guides the model to efficiently learn coarser information at the earlier stages of training and finer details in the later stages. Due to the discretization invariance of neural operators, the resulting model exhibits similar performance as training only on high resolutions, almost at the cost of low-resolution training.

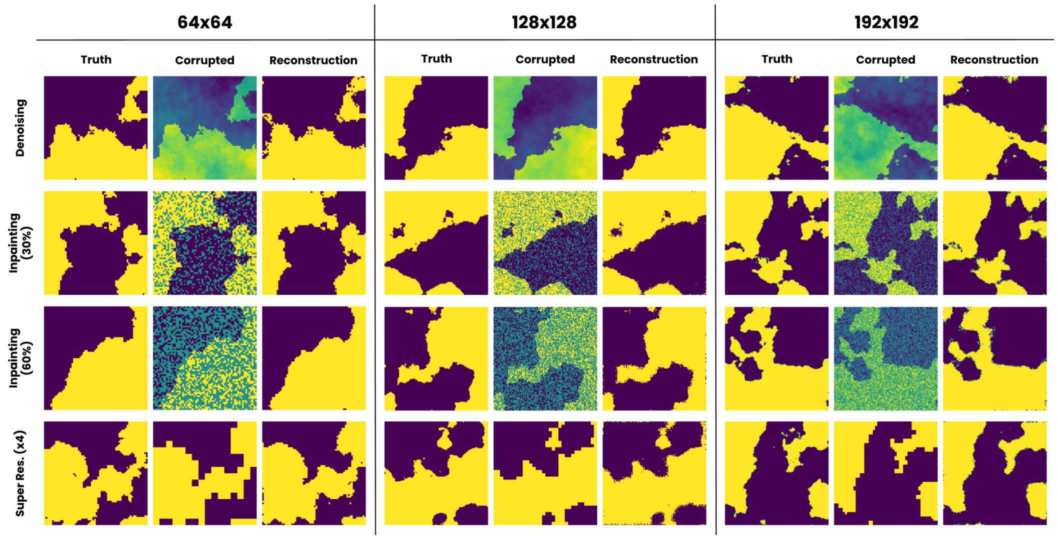

Figure 1: Qualitative results for various inverse problems on different resolutions of Darcy flow coefficient functions. While the model is solely trained on the lowest resolution, it can successfully reconstruct on higher resolutions even in challenging cases.

3 Results and Discussion

In this section, we describe our considered PDEs, the training of our diffusion priors, and the results of our experiments.

Darcy flow. To showcase the applicability of our approach, we perform several experiments on a 2D Darcy Flow, which is given by the PDE

$$-\nabla \cdot (\boldsymbol{a}(x)\nabla \boldsymbol{u}(x)) = f(x), \qquad x \in (0,1)^2, \tag{4}$$

with constant forcing $f(x) = 1$ and zero boundary conditions. We follow the strategies in [17] for the generation of coefficient functions $\boldsymbol{a} \sim h_{\#}\mathcal{N}(0, (-\Delta + 9I)^{-2})$, where $h\colon \mathbb{R} \to \mathbb{R}$ is defined to be 12 on positive numbers and 3 otherwise. Such PDEs are crucial for many physical applications, such as permeability in subsurface flows.

Navier-Stokes equations. We further evaluate the performance of our approach on Navier-Stokes equations by generating initial and terminal states of PDE as in [17]. In particular, we consider the evolution of a vorticity field $\boldsymbol{u}(x,t)$ over time given by the PDE

$$\partial_t \boldsymbol{u}(x,t) + \boldsymbol{w}(x,t) \cdot \nabla \boldsymbol{u}(x,t) = \nu \Delta \boldsymbol{u}(x,t) + f(x), \quad x \in (0,1)^2,\, t \in (0,T], \tag{5}$$

$$\nabla \cdot \boldsymbol{w}(x,t) = 0, \quad x \in (0,1)^2,\, t \in [0,T], \tag{6}$$

$$\boldsymbol{u}(x,0) = \boldsymbol{a}(x), \quad x \in (0,1)^2, \tag{7}$$

where $\boldsymbol{w}(x,t)$ denotes the velocity field, ν is the viscosity, and $f(x)$ represents a fixed forcing term. The initial condition $\boldsymbol{a}(x)$ is sampled from $\mathcal{N}(0, 7^{3/2}(-\Delta + 49I)^{-2.5})$ under periodic boundary conditions. The forcing term is chosen as $f(x) = 0.1\,(\sin(2\pi(x_1+x_2)) + \cos(2\pi(x_1+x_2)))$. We simulate the PDE for $T = 1$ second with 10 timesteps using a pseudo-spectral method.

Diffusion model. For the function space diffusion model, we use a U-shaped neural operator architecture [22] as denoiser and modify the noising process according to the discussion above. We train the denoiser using $50,000$ training samples at a 64×64 discretization (for multi-resolution training, we train the last 20% of the epochs on resolution 128×128). We then use the pre-trained denoiser within a deterministic sampler with $N = 50$ steps for the Darcy flow tasks and $N = 1000$ steps for the Navier-Stokes equations (as shown in Algorithm 1).

Recovering corrupted functions. We demonstrate the efficacy of our approach in solving inverse problems on function spaces by performing denoising, inpainting, and super-resolution tasks to retrieve coefficient functions $\boldsymbol{a}$ of the Darcy flow problem from corrupted measurements. For the

Table 1: Recovery of corrupted coefficient functions, where the model is trained on the lowest resolution.

Resolutions	Denoising	Inpainting (30%)	Inpainting (60%)	SR×4
64 × 64 (trained)	2.67%	0.31%	1.84%	1.68%
128 × 128	1.72%	0.46%	1.33%	1.53%
192 × 192	1.94%	0.33%	1.58%	1.81%

Table 2: Darcy flow inverse problem.

Pure	Noisy	Masked
4.28%	4.67%	4.47%

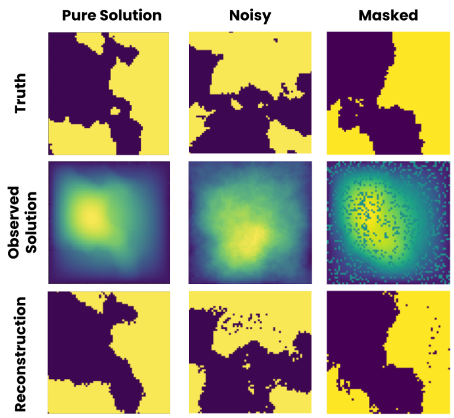

Figure 2: Reconstruction of coefficient functions from partially observed solutions of Darcy flow problems.

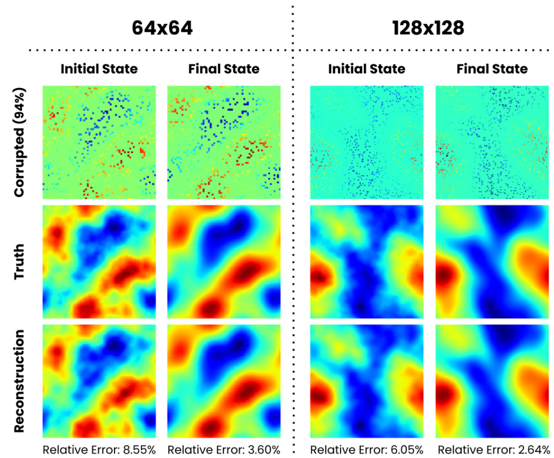

Figure 3: Results of combined forward and inverse problems on the Navier-Stokes equation. We also report relative L^2-errors averaged over the test set.

denoising task, we add noise from a Gaussian random field to the functions. In the inpainting task, we randomly mask 30% and 60% of the coordinates. For super-resolution, we attempt to reconstruct the function at 4-times higher resolution than the measurements. Figure 1 presents qualitative results at the training and higher resolutions. Given the binary nature of the data, we evaluate the error rate between true and reconstructed functions for each discretization. Table 1 provides quantitative results based on this metric, consistently demonstrating low error rates across different resolutions.

PDE-based inverse problems. We further consider common inverse problems in the context of the Darcy flow problem, seeking to determine the corresponding coefficient function $\boldsymbol{a}$ from the (sparsely observed) solution function $\boldsymbol{u}$. In this context, we define the forward operator $\boldsymbol{A}$ as an FDM-based numerical solver. To highlight practical applicability and generalizability, we recover coefficients from noisy or masked solutions. We present qualitative examples in Figure 2 and show in Table 2 that our method can achieve small relative L^2-errors even in the more challenging cases.

Combined forward and inverse problems. We also test our method on the Navier-Stokes equations by learning the joint distribution of initial and terminal states using our multi-resolution training. We verify our approach on a challenging case, where 94% of the coordinates are occluded in both the initial and terminal states. As shown in Figure 3, our approach can effectively retrieve both functions with relatively small errors on different discretizations.

4 Conclusion

In conclusion, we have introduced a novel discretization-agnostic generative framework for solving inverse problems in function spaces. Our framework can sample from posterior distributions without retraining across different resolutions or forward operators by leveraging pre-trained function space diffusion priors with neural operator architectures. We verified our approach in various settings, including PDE-based inverse problems on Darcy Flows and the Navier–Stokes equations.

References

[1] Masoumeh Dashti and Andrew M Stuart. The Bayesian approach to inverse problems. *Handbook of Uncertainty Quantification*, pages 311–428, 2017.

[2] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[3] 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.

[4] Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. *arXiv preprint arXiv:2201.11793*, 2022.

[5] Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022.

[6] Hyungjin Chung, Byeongsu Sim, Dohoon Ryu, and Jong Chul Ye. Improving diffusion models for inverse problems using manifold constraints. *Advances in Neural Information Processing Systems*, 35:25683–25696, 2022.

[7] Bingliang Zhang, Wenda Chu, Julius Berner, Chenlin Meng, Anima Anandkumar, and Yang Song. Improving diffusion inverse problem solving with decoupled noise annealing. *arXiv preprint arXiv:2407.01521*, 2024.

[8] Abbas Mammadov, Hyungjin Chung, and Jong Chul Ye. Amortized posterior sampling with diffusion prior distillation. *arXiv preprint arXiv:2407.17907*, 2024.

[9] Morteza Mardani, Jiaming Song, Jan Kautz, and Arash Vahdat. A variational perspective on solving inverse problems with diffusion models. *arXiv preprint arXiv:2305.04391*, 2023.

[10] Jiahe Huang, Guandao Yang, Zichen Wang, and Jeong Joon Park. Diffusionpde: Generative pde-solving under partial observation. *arXiv preprint arXiv:2406.17763*, 2024.

[11] Kamyar Azizzadenesheli, Nikola Kovachki, Zongyi Li, Miguel Liu-Schiaffini, Jean Kossaifi, and Anima Anandkumar. Neural operators for accelerating scientific simulations and design. *Nature Reviews Physics*, pages 1–9, 2024.

[12] Jae Hyun Lim, Nikola B Kovachki, Ricardo Baptista, Christopher Beckham, Kamyar Azizzadenesheli, Jean Kossaifi, Vikram Voleti, Jiaming Song, Karsten Kreis, Jan Kautz, et al. Score-based diffusion models in function space. *arXiv preprint arXiv:2302.07400*, 2023.

[13] Gavin Kerrigan, Justin Ley, and Padhraic Smyth. Diffusion generative models in infinite dimensions. *arXiv preprint arXiv:2212.00886*, 2022.

[14] Jakiw Pidstrigach, Youssef Marzouk, Sebastian Reich, and Sven Wang. Infinite-dimensional diffusion models. *arXiv preprint arXiv:2302.10130*, 2023.

[15] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

[16] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*, 2020.

[17] 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.

[18] 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.

[19] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34, 2021.

[20] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

[21] Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

[22] Md Ashiqur Rahman, Zachary E Ross, and Kamyar Azizzadenesheli. U-no: U-shaped neural operators. *arXiv preprint arXiv:2204.11127*, 2022.

[23] Md Ashiqur Rahman, Manuel A Florez, Anima Anandkumar, Zachary E Ross, and Kamyar Azizzadenesheli. Generative adversarial neural operators. *arXiv preprint arXiv:2205.03017*, 2022.

[24] Herbert E Robbins. An empirical bayes approach to statistics. In *Breakthroughs in Statistics: Foundations and basic theory*, pages 388–394. Springer, 1992.

[25] 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*, 1(3):1–27, 2024.

[26] Jiawei Zhao, Robert Joseph George, Yifei Zhang, Zongyi Li, and Anima Anandkumar. Incremental Fourier neural operator. *arXiv preprint arXiv:2211.15188*, 2022.

[27] Samuel Lanthaler, Andrew M Stuart, and Margaret Trautner. Discretization error of Fourier neural operators. *arXiv preprint arXiv:2405.02221*, 2024.