GibbsDDRM: A Partially Collapsed Gibbs Sampler for Solving Blind Inverse Problems with Denoising Diffusion Restoration

Naoki Murata, Koichi Saito, Chieh-Hsin Lai, Yuhta Takida, Toshimitsu Uesaka, Yuki Mitsufuji, Stefano Ermon

Presentation by Marty Mukherjee

Department of Applied Mathematics, University of Waterloo

Table of contents

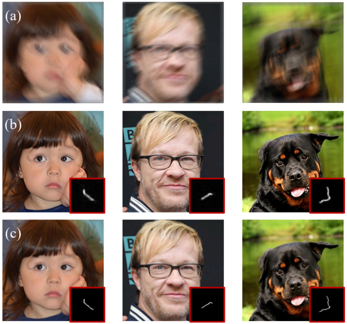


Figure 1. Blind image deblurring results obtained by GibbsDDRM: (a) measurement, (b) restored clean images with blur kernels (bottom right insets), and (c) ground truth images and blur kernels.

Introduction

Blind linear inverse problems

Estimation of both unknown clean data and the parameters of a linear operator from noisy measurements.

$$y = H_\varphi x_0 + z, \tag{1}$$

where $y \in \mathbb{R}^{d_y}$ is a vector of measurements, $H_\varphi \in \mathbb{R}^{d_y \times d_{x_0}}$ is a linear operator parameterized by $\varphi \in \mathbb{R}^{d_\varphi}$ and $x_0 \in \mathbb{R}^{d_{x_0}}$ is the original clean data. $z \sim N(0, \sigma_y^2 I)$ is a Gaussian measurement noise with known covariance.

Aim: Estimate x_0, φ

Bayesian framework: Sample from the posterior $p(x_0, \varphi | y)$

Assumption: x_0 is drawn from a generative model $p_\theta(x_0)$ (close to the true data distribution) and φ is drawn from a known prior $p(\varphi)$ (Laplace distribution) independently of the data.

Denoising Diffusion Restoration Models (DDRM)

Uses a pre-trained diffusion model as a prior for data in a non-blind linear inverse problem. It is defined as a Markov chain $x_T \to x_{T-1} \to \cdots \to x_1 \to x_0$ where

$$p(x_{0:T}|y) = p_\theta^{(T)}(x_T|y) \prod_{t=0}^{T-1} p_\theta^{(t)}(x_t|x_{t+1}, y) \tag{2}$$

Core idea: use the singular value decomposition (SVD) of a linear operator H to transform x_0 and y to a shared spectral space.

Gibbs sampler

A Gibbs sampler is a MCMC method for sampling from the joint distribution of a set of variables.

Suppose we want to obtain k samples of $X = (x_1, ..., x_n)$ from a joint distribution $p(x_1, ..., x_n)$.

Start with $X^{(0)} = (x_1^{(0)}, ..., x_n^{(0)})$.

Sample iteratively by $x_j^{(i+1)} \sim p(x_j^{(i+1)} | x_1^{(i+1)}, ..., x_{j-1}^{(i+1)}, x_{j+1}^{(i)}, ..., x_n^{(i)})$.

A blocked Gibbs sampler samples variables in a group or a ”block” of variables simultaneously while conditioned on all the other variables.

Partially collapsed Gibbs sampler (PCGS)

1. Sample multiple variables at once
2. If a variable is sampled multiple times, then only the last value is relevant. The remaining unused variables can be integrated out.
3. Choose an arbitrary sampling order.

e.g. Sample (W, X, Y, Z).

1. Sample Y from $p(Y, \cancel{W}|X, Z)$
2. Sample Z from $p(Z, \cancel{W}|X, Y)$
3. Sample W from $p(W|X, Y, Z)$
4. Sample X from $p(X|W, Y, Z)$

GibbsDDRM: Partially Collapsed Gibbs Sampler with DDRM

Methods that employ rich generative models as priors to solve inverse problems

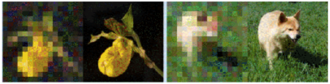
Denoising Diffusion Restoration Models (DDRM) [Kawar, NeurIPS2022]

Deblurring

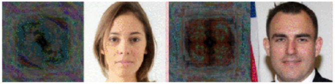
Colorization

Inpainting

Diffusion Posterior Sampling (DPS) [Chung, ICLR2023]



Super resolution



Phase retrieval

Non-uniform deblur

Both methods assume that corruption processes are known.

☞**We consider a blind setting,
where the degradation process is unknown.**

Target joint distribution for blind linear inverse problems

Joint distribution:

$$p(x_0, y, \varphi) = p_\theta(x_0)p(\varphi)N(y|H_\varphi x_0, \sigma_y^2 I) \tag{3}$$

We model the data distribution using a pre-trained diffusion model

$$p(x_{0:T}, \varphi, y) = p_\theta^{(T)}(x_T)\prod_{t=0}^{T-1} p_\theta^{(t)}(x_t|x_{t+1})p(\varphi)N(y|H_\varphi x_0, \sigma_y^2 I) \tag{4}$$

Algorithm 1 Proposed PCGS for the posterior in Eq. (7)

Input: Measurement $\mathbf{y}$, initial values $\boldsymbol{\varphi}^{(0,0)}$.
Output: Restored data $\mathbf{x}_0^{(N,M_0)}$, linear operator's parameters $\boldsymbol{\varphi}^{(N,K)}$.
$K \leftarrow 0$ // K counts the number of updates for $\boldsymbol{\varphi}$ in a cycle.
for $n = 1$ **to** N **do**
  $\boldsymbol{\varphi}^{(n,0)} \leftarrow \boldsymbol{\varphi}^{(n-1,K)}$, $K \leftarrow 0$
  Sample $\mathbf{x}_T^{(n,0)} \sim p(\mathbf{x}_T|\boldsymbol{\varphi}^{(n,K)}, \mathbf{y})$
  // $\uparrow$ approximated by $p_\theta(\mathbf{x}_T|\boldsymbol{\varphi}, \mathbf{y})$.
  for $t = T-1$ **to** 0 **do**
    $\chi_t \leftarrow \{\mathbf{x}_{t+1}^{(n,M_{t+1})}, \mathbf{x}_{t+2}^{(n,M_{t+2})}, \cdots, \mathbf{x}_T^{(n,0)}\}$
    Sample $\mathbf{x}_t^{(n,0)} \sim p(\mathbf{x}_t|\boldsymbol{\varphi}^{(n,K)}, \chi_t, \mathbf{y})$
    // $\uparrow$ approximated by $p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1}, \boldsymbol{\varphi}, \mathbf{y})$.
    for $m = 1$ **to** M_t **do**
      Sample $\boldsymbol{\varphi}^{(n,K+1)} \sim p(\boldsymbol{\varphi}|\mathbf{x}_t^{(n,m-1)}, \chi_t, \mathbf{y})$
      // $\uparrow$ Langevin sampling with the approximated score $\nabla_{\boldsymbol{\varphi}} \log p(\mathbf{y}|\mathbf{x}_{\theta,t}, \boldsymbol{\varphi})$.
      $K \leftarrow K + 1$
      Sample $\mathbf{x}_t^{(n,m)} \sim p(\mathbf{x}_t|\boldsymbol{\varphi}^{(n,K)}, \chi_t, \mathbf{y})$
      // $\uparrow$ approximated by $p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1}, \boldsymbol{\varphi}, \mathbf{y})$.
    end for
  end for
end for

Sampling of x_T

Theoretically sample from $p(x_T|\varphi, y)$, but this is intractable.

Solution: Sample from the modified DDRM $p_\theta(x_T|\varphi, y)$.

We use the singular value decomposition (SVD), $H_\varphi = U_\varphi \Sigma_\varphi V_\varphi^T$. Let $\bar{x}_t^{(i)}$ be the i-th singular vector of $\bar{x}_t = V_\varphi^T x_t$, $\bar{y}^{(i)}$ be the i-th singular vector of $\bar{y} = \Sigma_\varphi^+ U_\varphi^T y$.

$$p_\theta^{(T)}\left(\bar{\mathrm{x}}_T^{(i)} \mid \boldsymbol{\varphi}, \mathrm{y}\right) = \begin{cases} \mathcal{N}(\bar{\mathrm{y}}^{(i)}, \sigma_T^2 - \sigma_y^2/s_i^2), & s_i > 0 \\ \mathcal{N}(0, \sigma_T^2), & s_i = 0 \end{cases} \tag{5}$$

Sampling of x_t

Theoretically sample from $p(x_t|x_{t+1:T}, \varphi, y)$, but this is intractable. Approximate the conditional distribution using DDRM. Denote the prediction of x_0 at time step t by $x_{\theta,t}$.

$$p_\theta^{(t)}\left(\overline{\mathrm{x}}_t^{(i)} \mid \mathrm{x}_{t+1}, \boldsymbol{\varphi}, \mathbf{y}\right) =$$
$$\begin{cases} \mathcal{N}\left(\overline{\mathrm{x}}_{\theta,t}^{(i)} + \sqrt{1-\eta^2}\sigma_t \frac{\overline{\mathrm{x}}_{t+1}^{(i)} - \overline{\mathrm{x}}_{\theta,t}^{(i)}}{\sigma_{t+1}}, \eta^2\sigma_t^2\right) & \text{if } s_i = 0 \\ \mathcal{N}\left(\overline{\mathrm{x}}_{\theta,t}^{(i)} + \sqrt{1-\eta^2}\sigma_t \frac{\overline{\mathrm{y}}^{(i)} - \overline{\mathrm{x}}_{\theta,t}^{(i)}}{\sigma_{\mathrm{y}}/s_i}, \eta^2\sigma_t^2\right) & \text{if } \sigma_t < \frac{\sigma_{\mathrm{y}}}{s_i} \\ \mathcal{N}\left((1-\eta_b)\,\overline{\mathrm{x}}_{\theta,t}^{(i)} + \eta_b \overline{\mathbf{y}}^{(i)}, \sigma_t^2 - \frac{\sigma_{\mathrm{y}}^2}{s_i^2}\eta_b^2\right) & \text{if } \sigma_t \geq \frac{\sigma_{\mathrm{y}}}{s_i} \end{cases}$$

where $0 \leq \eta \leq 1$ and $0 \leq \eta_b \leq 1$ are hyperparameters, and $0 = \sigma_0 < \sigma_1 < \sigma_2 < \cdots < \sigma_T$ are noise levels that is the same as that defined with the pre-trained diffusion model.

Sampling of φ

Theoretically sample from $p(\varphi|x_{t:T}, y)$.

Perform sampling by Langevin dynamics as follows:

$$\boldsymbol{\varphi} \leftarrow \boldsymbol{\varphi} + (\xi/2)\nabla_{\varphi} \log p\left(\boldsymbol{\varphi} \mid \mathbf{x}_{t:T}, \mathbf{y}\right) + \sqrt{\xi}\boldsymbol{\epsilon},$$

where ξ is a step size and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$. By Bayes' rule, the score $\nabla_{\varphi} \log p\left(\varphi \mid \mathbf{x}_{t:T}, \mathbf{y}\right)$ can be decomposed into two terms:

$$\nabla_{\boldsymbol{\varphi}} \log p\left(\boldsymbol{\varphi} \mid \mathbf{x}_{t:T}, \mathbf{y}\right) = \nabla_{\boldsymbol{\varphi}} \log p\left(\mathbf{y} \mid \mathbf{x}_{t:T}, \boldsymbol{\varphi}\right) + \nabla_{\boldsymbol{\varphi}} \log p\left(\boldsymbol{\varphi} \mid \mathbf{x}_{t:T}\right)$$

Theorem 3.2. *(modified version of Theorem 1 in (Chung et al., 2023b)) For the measurement model in Eq. (1), we have*

$$p(\mathbf{y}|\mathbf{x}_{t:T}, \boldsymbol{\varphi}) \simeq p(\mathbf{y}|\mathbf{x}_{\theta,t}, \boldsymbol{\varphi}), \tag{13}$$

and the approximation error can be quantified with the Jensen gap (Gao et al., 2017), which is upper bounded by

$$\mathcal{J} \leq \frac{1}{\sigma_{\mathbf{y}} \left(\sqrt{2\pi\sigma_{\mathbf{y}}^2}\right)^{d_{\mathbf{y}}}} e^{-1/2} s_1 m_1, \tag{14}$$

where $m_1 := \int \|\mathbf{x}_0 - \mathbf{x}_{\theta,t}\| p(\mathbf{x}_0|\mathbf{x}_{t:T}) d\mathbf{x}_0$, and s_1 is the largest singular value of $\mathbf{H}_{\boldsymbol{\varphi}}$.

Approximate gradient: $\nabla_{\varphi} \log p(\mathrm{y} \mid \mathrm{x}_{t:T}, \varphi) \approx \nabla_{\varphi} \log p(\mathrm{y} \mid \mathrm{x}_{\theta,t}, \varphi)$

And from our model, $p(\mathrm{y} \mid \mathrm{x}_{t:T}, \varphi) = \mathcal{N}(\mathrm{y}|\mathrm{H}_{\varphi}\mathrm{x}_{\theta,t}, \sigma_y^2 I)$

$$\nabla_{\varphi} \log p(\mathrm{y} \mid \mathrm{x}_{\theta,t}, \varphi) = -\frac{1}{2\sigma_{\mathrm{y}}^2} \nabla_{\varphi} \|\mathrm{y} - \mathrm{H}_{\varphi}\mathrm{x}_{\theta,t}\|^2 \tag{6}$$

Also, $\log p(\varphi \mid \mathrm{x}_{t:T}) = \log p(\varphi)$

Results

Blind image deblurring

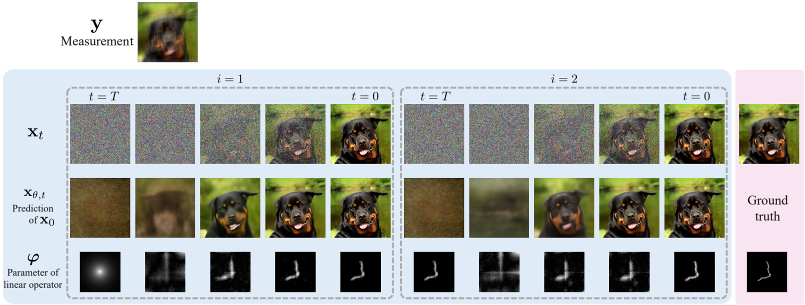


Figure 4. Visualization of GibbsDDRM for the blind image deblurring task on the AFHQ dataset.

Blind image deblurring

Quantitative results

Table 1. Blind image deblurring results on FFHQ and AFHQ (256 × 256). The blurred images have additive Gaussian noise with $\sigma_{\mathbf{y}} = 0.02$. (∗) The results for BlindDPS (Chung et al., 2023a), as reported in the original paper, are also listed, although that method uses a pre-trained score function for blur kernels. The results of DDRM (Kawar et al., 2022) with the ground truth kernels (i.e., non-blind setting) are also listed. **Bold**: Best. underscore: second best.

	FFHQ (256 × 256)			**AFHQ** (256 × 256)		
Method	FID↓	LPIPS ↓	PSNR ↑	FID ↓	LPIPS↓	PSNR↑
GibbsDDRM (ours)	38.71	**0.115**	25.80	48.00	**0.197**	22.01
MPRNet (Zamir et al., 2021)	62.92	0.211	**27.23**	50.43	0.278	**27.02**
DeblurGANv2 (Kupyn et al., 2019)	141.55	0.320	19.86	156.92	0.429	17.64
Pan-DCP (Pan et al., 2017)	239.69	0.653	14.20	185.40	0.632	14.48
SelfDeblur (Ren et al., 2020)	283.69	0.859	10.44	250.20	0.840	10.34
BlindDPS (Chung et al., 2023a)*	**29.49**	0.281	22.24	**23.89**	0.338	20.92
DDRM (Kawar et al., 2022) with GT kernel	33.97	0.062	30.64	24.60	0.078	29.37

GibbsDDRM achieved the lowest LPIPS (Learned Perceptual Image Patch Similarity) and the second lowest FID (Frechet Inception Distance)

Qualitative results

Vocal dereverberation

Table 2. Vocal dereverberation results. **Bold**: Best.

Method	FAD ↓	SI-SDR ↑ improvement	SRMR ↑
Wet (unprocessed)	5.74	–	7.11
Reverb Conversion (Koo et al., 2021)	5.69	0.02	7.23
Music Enhancement (Kandpal et al., 2022)	7.51	−23.9	7.92
Unsupervised Dereverberation(Saito et al., 2023)	4.99	0.37	7.94
GibbsDDRM	**4.21**	**0.59**	**8.40**

The task: Restore the original dry vocal from a noisy reverberant (wet) vocal.

GibbsDDRM achieved the highest SRMR (speech-to-reverberation modulation energy ratio) and the lowest FAD (Frechet Audio Distance).

Summary

Problem: posterior sampling from blind linear inverse problems: $p(x_0, \varphi | y)$ where $y = H_\varphi x_0 + z, z \sim N(0, \sigma_y^2)$

Approach: Efficient posterior sampling using PCGS.

Advantage:

- A pretrained diffusion model can be applied to different inverse problems without requiring fine-tuning
- Success even when generic priors (e.g. Laplacian prior) are used for the linear operators

Extension: Solving PDEs with DDRM

Denoising Diffusion Restoration Tackles Forward and Inverse Challenges in the Laplace Operator

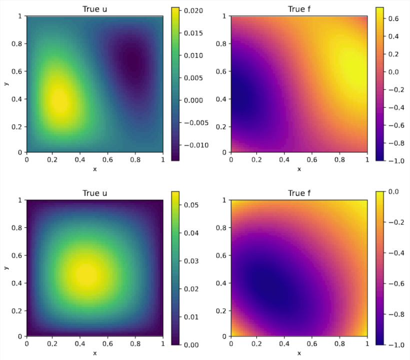
2D Poisson Equation: $\langle \nabla, \nabla u(x,y) \rangle = f(x,y)$ in the domain $\Omega = [0,1]^2$.
Boundary conditions: $u(x,y) = 0$ in $\partial\Omega$.

We trained a denoising diffusion implicit model (DDIM) to generate pairs of u and f in the form of a $2 \times 64 \times 64$ image, where the first channel corresponds to u and the second channel corresponds to f.

1. Neural network pairs. We sampled $u(x, y)$ randomly as

$$u(x, y) = g_{NN}(x, y)x(1 - x)y(1 - y), \tag{7}$$

where g_{NN} is a randomly initialized neural network with three hidden layers and *tanh* activation function. We then compute $f = \Delta u$ using auto-differentiation by using the ***`autograd.grad`*** function from ***`Pytorch`***.



Dataset generation

2. Sine pairs

$$u(x, y) = \sin(n\pi x)\sin(k\pi y)$$

where n, k are positive integers. We can solve u analytically to get

$$\nabla^2 u = f = -\pi^2(n^2 + k^2)\sin(n\pi x)\sin(k\pi y).$$

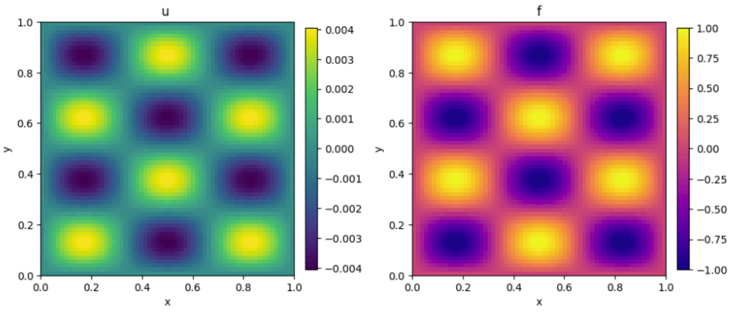


Figure 4: Example type 1 analytical solution with $n = 3$ and $k = 4$.

Generated pairs

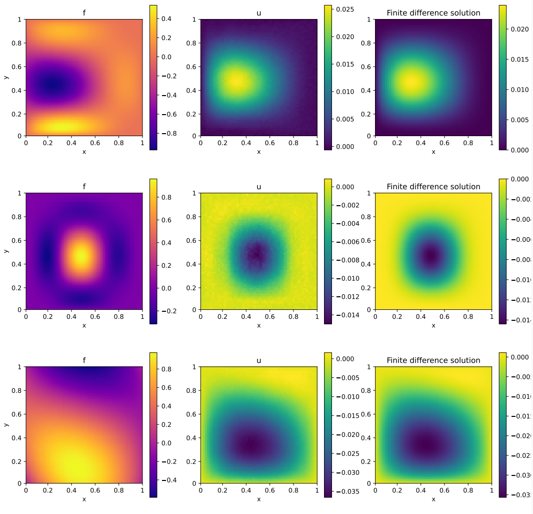


Figure 1: The MAE between the generated solution $u(x, y)$ and the finite difference solution is 4.373×10^{-4}.

Dry forward process

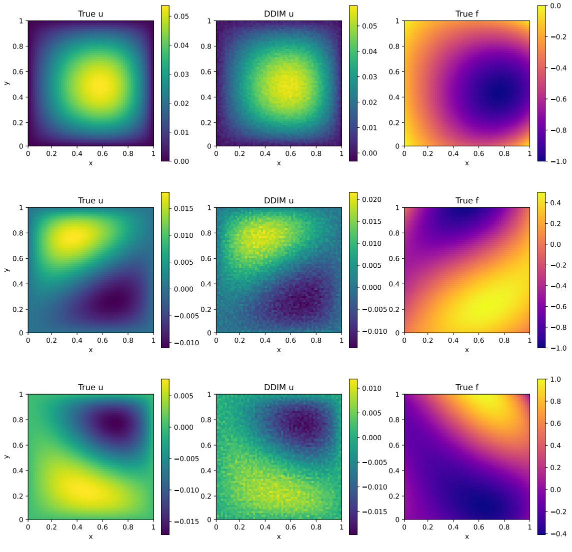


Figure 2: The average MAE is 1.123×10^{-3}.

Dry inverse process

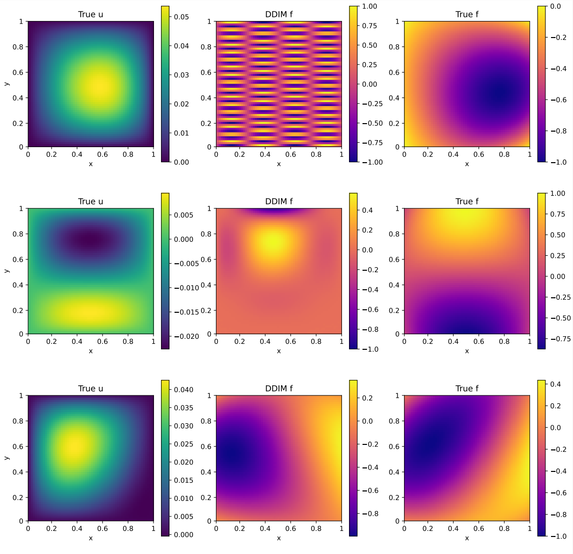


Figure 3: The average MAE is 0.5515.

Eigenvalues and eigenfunctions of the Laplace operator

Proposition 1 *The eigenfunction and eigenvalue pairs of the Laplacian operator Δ in a domain $\Omega = [0,1]^2$ subject to the boundary conditions $u = 0$ on $\partial\Omega$ are of the form*

$$u_{n,m}(x,y) = \sin(n\pi x)\sin(m\pi y), \tag{5}$$

$$\lambda_{n,m} = -(n\pi)^2 - (m\pi)^2. \tag{6}$$

Although this problem does not have a notion of SVD, we can exploit its eigenvalue decomposition.

Problem formulation: Inverse problem

Consider the following inverse problem defined on a domain $\Omega = [0, 1]^2$:

$$\begin{aligned} f_0 &= \Delta u_0, \\ u &= u_0 + z_u. \end{aligned} \tag{8}$$

We have the boundary conditions $u_0 = 0$ on $\partial\Omega$. $z_u(x, y)$ is a Brownian bridge satisfying the same boundary conditions as u_0. We can express z_u as a double sum of sinusoidal functions (satisfying the boundary conditions)

$$z_u = \sum_{n=0}^{N} \sum_{m=0}^{N} w_{n,m} \sin(n\pi x) \sin(m\pi y), \tag{9}$$

where $w_{n,m} \sim \mathcal{N}(0, \sigma_{n,m}^2)$ are random coefficients with known variance $\sigma_{n,m}^2$.

DDRM for solving PDEs: Inverse Process

Define $\overline{\mathrm{u}}^{(n,m)}$ and $\bar{\mathrm{f}}_t^{(n,m)}$ as follows:

$$\overline{\mathrm{u}}^{(n,m)} = \lambda_{n,m}\langle \mathrm{u}, \sin(n\pi x)\sin(m\pi y)\rangle, \tag{10}$$

$$\bar{\mathrm{f}}_t^{(n,m)} = \langle \mathrm{f}_t, \sin(n\pi x)\sin(m\pi y)\rangle, \tag{11}$$

Sampling of f_T:

$$p_\theta^{(T)}(\bar{\mathrm{f}}_T^{(n,m)} \mid \mathrm{u}) = \mathcal{N}(\overline{\mathrm{u}}^{(n,m)}, \sigma_T^2 - \sigma_{n,m}^2\lambda_{n,m}^2). \tag{12}$$

Sampling of f_t:

$$p_\theta^{(t)}\left(\bar{\mathrm{f}}_t^{(n,m)} \mid \mathrm{f}_{t+1}, \mathrm{u}\right) =$$
$$\begin{cases} \mathcal{N}\left(\bar{\mathrm{f}}_{\theta,t}^{(n,m)} + \sqrt{1-\eta^2}\sigma_t \frac{\overline{\mathrm{u}}^{(n,m)} - \bar{\mathrm{f}}_{\theta,t}^{(n,m)}}{\sigma_{n,m}\lambda_{n,m}}, \eta^2\sigma_t^2\right), & \text{if } \sigma_t < \sigma_{n,m}\lambda_{n,m}, \\ \mathcal{N}\left((1-\eta_b)\,\bar{\mathrm{f}}_{\theta,t}^{(n,m)} + \eta_b\overline{\mathrm{u}}^{(n,m)}, \sigma_t^2 - \sigma_{n,m}^2\lambda_{m,n}^2\eta_b^2\right), & \text{if } \sigma_t \geq \sigma_{n,m}\lambda_{n,m}, \end{cases} \tag{13}$$

Results

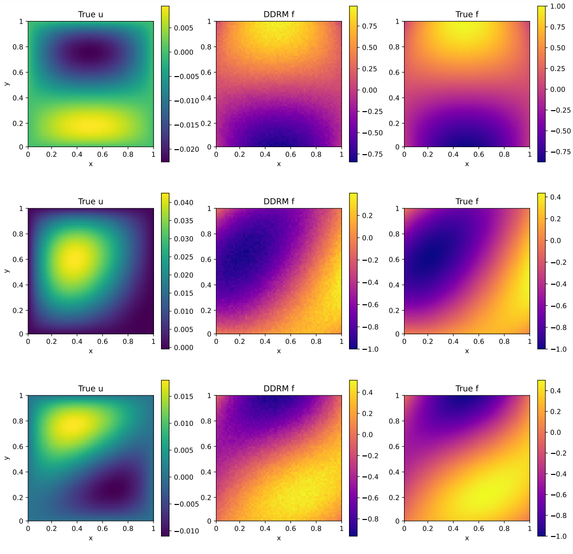


Figure 4: The average MAE is 3.215×10^{-2}.

Problem formulation: Forward problem

Consider the following forward problem defined on a domain $\Omega = [0,1]^2$:

$$f = \Delta u_0 + z_f, \tag{14}$$

where $z_f \sim \mathcal{N}(0, \sigma_u^2)$ is measurement noise with known covariance σ_u^2.

Theorem 3 *Let $f = \Delta u + z_f$ where $z_f \sim \mathcal{N}(0, \sigma_f^2)$. Consider the discrete sine transform (DST) of $u(x, y)$*

$$\overline{\mathbf{u}}^{(n,m)} = \langle \mathbf{u}, \sin(n\pi x)\sin(m\pi y)\rangle. \tag{10}$$

We can place an upper bound on the variance of $\overline{\mathbf{u}}^{(n,m)}$:

$$Var[\overline{\mathbf{u}}^{(n,m)} | f_0] \leq \left(\frac{1}{\pi^2(n^2+m^2)} + \ln 2 \max(n,m)\right)^2 \sigma_f^2. \tag{11}$$

DDRM for solving PDEs: Forward Process

Define $\overline{\mathrm{u}}_t^{(n,m)}$, $\bar{\mathrm{f}}^{(n,m)}$, and $\overline{\mathrm{K}}^{(n,m)}$ as follows:

$$\overline{\mathrm{u}}_t^{(n,m)} = \langle \mathrm{u}_t, \sin(n\pi x)\sin(m\pi y)\rangle, \tag{15}$$

$$\bar{\mathrm{f}}^{(n,m)} = \langle \mathrm{f}, \sin(n\pi x)\sin(m\pi y)\rangle / \lambda_{n,m}, \tag{16}$$

$$\overline{\mathrm{K}}^{(n,m)} = \left(\frac{1}{|\lambda_{n,m}|} + \ln 2 \max(n,m)\right)^2. \tag{17}$$

Sampling of u_T

$$p_\theta^{(T)}(\overline{\mathrm{u}}_T^{(n,m)} \mid \mathrm{f}) = \mathcal{N}(\bar{\mathrm{f}}^{(n,m)}, \sigma_T^2 - \sigma_f^2 \overline{\mathrm{K}}^{(n,m)} / \lambda_{n,m}^2). \tag{18}$$

Sampling of u_t

$$p_\theta^{(t)}\left(\overline{\mathrm{u}}_t^{(n,m)} \mid \mathrm{u}_{t+1}, \mathrm{f}\right) = \begin{cases} \mathcal{N}\left(\overline{\mathrm{u}}_{\theta,t}^{(n,m)} + \sqrt{1-\eta^2}\sigma_t \frac{\bar{\mathrm{f}}^{(n,m)} - \overline{\mathrm{u}}_{\theta,t}^{(n,m)}}{\sigma_f \sqrt{\overline{\mathrm{K}}^{(n,m)}}/\lambda_{n,m}}, \eta^2 \sigma_t^2\right), & \sigma_t < \frac{\sigma_f \sqrt{\overline{\mathrm{K}}^{(n,m)}}}{\lambda_{n,m}}, \\ \mathcal{N}\left((1-\eta_b)\,\overline{\mathrm{u}}_{\theta,t}^{(n,m)} + \eta_b \bar{\mathrm{f}}^{(n,m)}, \sigma_t^2 - \frac{\sigma_f^2 \overline{\mathrm{K}}^{(n,m)}}{\lambda_{n,m}^2}\eta_b^2\right), & \sigma_t \geq \frac{\sigma_f \sqrt{\overline{\mathrm{K}}^{(n,m)}}}{\lambda_{n,m}}. \end{cases} \tag{19}$$

Results

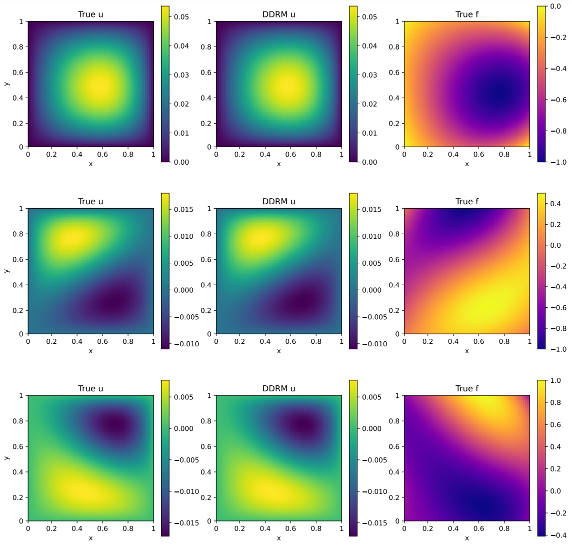


Figure 5: The average MAE over is 1.175×10^{-6}.

Quantitative results

Inverse problem:

PINNs	PI-DeepONet	DDIM	DDRM	FD
3.704e-01	4.224e-01	5.515e-01	3.215e-02	1.163e-02

Table 1: MAE in predicting f conditioned on u, averaged along 1024 samples

Forward problem:

PINNs	PI-DeepONet	DDIM	DDRM	FD
2.156e-03	3.183e-04	1.123e-03	1.175e-06	6.672e-07

Table 2: MAE in predicting u conditioned on f, averaged along 1024 samples

Future research

- Solving more complicated PDEs, like the Navier-Stokes equation
- Including physics in the training of diffusion models
- Outperforming finite difference methods