Inverse Problem

Choeun Kim

IDEA

February 10, 2026

Inverse Problem

Consider the following form

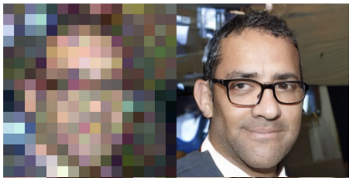
$$\mathbf{y} = \mathcal{A}(\mathbf{x}) + \mathbf{n}$$

- forward operator $\mathcal{A} : \mathbb{R}^n \to \mathbb{R}^m, m < n$
- what we want : $\mathbf{x} \in \mathbb{R}^n$

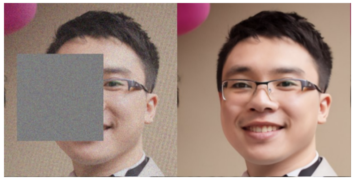
  what we have : $\mathbf{y} \in \mathbb{R}^m$
- noise $\mathbf{n} \in \mathbb{R}^m$, usually Gaussian (i.e., $\sim \mathcal{N}(0, \sigma_y^2 \mathbf{I})$)

Examples

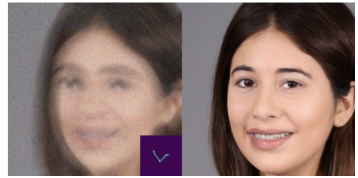
Super-Resolution



Inpainting

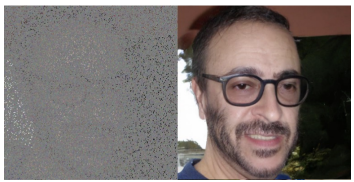


Motion Deblurring



Colorization

Inpainting(random)



Gaussian Deblurring

DMs for Inverse Problems

$$\mathbf{y} = \mathcal{A}(\mathbf{x}) + \mathbf{n}$$

- **Goal**: Estimate the posterior score $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y})$ to solve the reverse-time SDE. ($\mathbf{x}_0 = \mathbf{x}$)
- **Decomposition**:

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y}) = \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t)$$

  - **First Term**: Pushes the noisy state $\mathbf{x}_t$ towards the data manifold.
  - **Second Term**: Pulls $\mathbf{x}_t$ in a direction where the degraded version ($\mathcal{A}(\mathbf{x}_0)$) aligns with the measurement $\mathbf{y}$.

While the first term is accessible via unconditional diffusion models, the second term is intractable in general, thus it requires a proper approximation.

Part I

Projection-based

Score-SDE
Song et al., 2021

- **Focus**: Linear inverse problems ($\mathbf{y} = A\mathbf{x}$) with an orthogonal matrix A.
- **Approximation**:
$$\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) \approx -A^\top(\mathbf{y}_t - A\mathbf{x}_t)$$
where $\mathbf{y}_t$ denotes the perturbed measurement at time t via the forward process.
- **Interpretation**: Acts as a projection onto the constraint $\mathbf{y}_t = A\mathbf{x}_t$.

ILVR

Choi et al., 2021

- **Focus**: Super-Resolution (SR) task via *'condition encoding'*.
  Ensures the consistency constraint: $\phi_N(\mathbf{x}_0) = \phi_N(\mathbf{y}) \approx \mathbf{y}$.
- **Foundation**: Theoretically grounded in the general solution of linear systems using the Moore-Penrose pseudo-inverse:

$$\mathbf{x}_{new} = A^\dagger \mathbf{y} + (I - A^\dagger A)\mathbf{x}_{old}$$

- **Approximation**: Since the low-pass filter ϕ_N is an idempotent projection ($A^\dagger = A$), this update implies the following score approximation:

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) \approx -(A^\top A)^{-1} A^\top (\mathbf{y}_t - A\mathbf{x}_t)$$

Limitations

- when there is noise in $\mathbf{y}$, the noise will be amplified
- cannot handle the nonlinear operator

Part II

DPS

DPS

Chung, Kim, Mccann, et al., 2023

- **Theoretical Basis**:

$$p(\mathbf{y}|\mathbf{x}_t) = \int p(\mathbf{y}|\mathbf{x}_0)p(\mathbf{x}_0|\mathbf{x}_t)d\mathbf{x}_0 = \mathbb{E}_{\mathbf{x}_0 \sim p(\mathbf{x}_0|\mathbf{x}_t)} p(\mathbf{y}|\mathbf{x}_0),$$

$$p(y|\mathbf{x}_t) \approx p(\mathbf{y}|\hat{\mathbf{x}}_0(\mathbf{x}_t)), \text{ where } \hat{\mathbf{x}}_0(\mathbf{x}_t) = \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]$$

- **Approximation**:

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) \approx -\frac{1}{2\sigma_y^2} \nabla_{\mathbf{x}_t} \|\mathbf{y} - \mathcal{A}(\hat{\mathbf{x}}_0(\mathbf{x}_t))\|_2^2$$

ΠGDM

Song et al., 2023

- **Theoretical Basis**: While DPS used $p(\mathbf{x}_0|\mathbf{x}_t) \approx \delta(\mathbf{x}_0 - \hat{\mathbf{x}}_0(\mathbf{x}_t))$, the work instead used the Gaussian approximation, i.e., $p(\mathbf{x}_0|\mathbf{x}_t) \approx \mathcal{N}(\hat{\mathbf{x}}_0(\mathbf{x}_t), r_t^2\mathbf{I})$, where r_t is a hyperparameter.
- **Approximation**:

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) \approx -\nabla_{\mathbf{x}_t}\hat{\mathbf{x}}_0(\mathbf{x}_t)(r_t^2 AA^T + \sigma_y^2\mathbf{I})^{-1}A^T(\mathbf{y} - A\hat{\mathbf{x}}_0(\mathbf{x}_t))$$

Part III

Blind Inverse Problem

BlindDPS
Chung, Kim, Kim, and Ye, 2023

- Consider the following forward model
$$\mathbf{y} = \mathcal{A}_{\varphi}(\mathbf{x}) + \mathbf{n}$$
where φ is the parameter of the forward operator, $\mathbf{x}$ is the ground truth image, and $\mathbf{n}$ is some noise.
- Specifically, consider where the forward operator is a blur kernel. That is,
$$\mathbf{y} = k * \mathbf{x} + \mathbf{n}, \mathbf{n} \sim \mathcal{N}(0, \mathbf{I})$$
- Assume that the image $\mathbf{x}$ and the kernel k are statistically independent.

BlindDPS

Chung, Kim, Kim, and Ye, 2023

- **Training**
  - Image Prior : Leverage off-the-shelf pre-trained diffusion models
  - Operator Prior : Train a lightweight score model for the parameter k using DSM

$$\mathcal{L}_{DSM} = \mathbb{E}_{t,k_0,k_t} \| s_\theta^k(k_t, t) - \nabla_{k_t} \log p(k_t | k_0) \|^2$$

- **Inference**
  - Inputs : $\mathbf{y}$, random noise $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$, and $k_T \sim \mathcal{N}(0, \mathbf{I})$
  - Run two parallel reverse SDEs

$$d\mathbf{x}_t = [f(\mathbf{x}) - g^2(t)(s_{\theta^*}^{\mathbf{x}}(\mathbf{x}_t, t) + \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\hat{\mathbf{x}}_0(\mathbf{x}_t), \hat{k}_0(k_t)))]dt + g(t)d\bar{\mathbf{w}}_t$$
$$dk_t = [f(\mathbf{x}) - g^2(t)(s_{\theta^*}^{k}(k_t, t) + \nabla_{k_t} \log p(\mathbf{y}|\hat{\mathbf{x}}_0(\mathbf{x}_t), \hat{k}_0(k_t)))]dt + g(t)d\bar{\mathbf{w}}_t$$

  - The processes are coupled solely through the data consistency term $\nabla \log p(\mathbf{y}|\hat{\mathbf{x}}_0, \hat{k}_0)$.
  - Ouputs : $\hat{\mathbf{x}}_0$ and $\hat{k}_0$

GDiff

Heurtel-Depeiges et al., 2024

- Formalize the problem as follows:

$$\mathbf{y} = \mathbf{x} + \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_\phi),$$

  where $\phi \in \mathbb{R}^K$ is an unknown vector of parameters.

- **Idea**: Use a Gibbs sampler to sample $\mathbf{x}, \phi \sim p(\mathbf{x}, \phi|\mathbf{y})$.
  - $p(\mathbf{x}|\mathbf{y}, \phi)$: Use DM
  - $p(\phi|\mathbf{y}, \mathbf{x})$: Use HMC

GDiff
Heurtel-Depeiges et al., 2024

Consider noising Itô processes $(\mathbf{z}_t)_{t\in[0,1]}$ defined by:

$$d\mathbf{z}_t = -f(t)\mathbf{z}_t dt + g(t)\boldsymbol{\Sigma}_\varphi^{1/2} d\mathbf{w}_t, \mathbf{z}_0 = \mathbf{x}, \tag{1}$$

where $f, g : [0, 1] \to \mathbb{R}_+$ are two continuous functions, $(\mathbf{w}_t)_{t\in[0,1]}$ is a standard d-dimensional Wiener process, and $\boldsymbol{\Sigma}_\varphi$ corresponds to a noramlized version of the covariance matrix $\boldsymbol{\Sigma}_\phi$.

GDiff
Heurtel-Depeiges et al., 2024

Proposition 1

For a forward SDE $\mathbf{z}_{t\in[0,1]}$ *defined by (1), for all* $t \in [0,1], \mathbf{z}_t$ *reads*

$$\mathbf{z}_t = a(t)\mathbf{x} + b(t)\boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\varphi}),$$

with $a : [0,1] \to \mathbb{R}_+$ *a decreasing function with* $a(0) = 1$*, and* $b : [0,1] \to \mathbb{R}_+$ *an increasing function with* $b(0) = 0$*. Moreover, for* f *and* g *suited so that* $\sigma \leq b(1)/a(1)$*, there exists* $t^* \in [0,1]$ *such that*

$$\tilde{\mathbf{y}} := a(t^*)\mathbf{y} \overset{d}{=} \mathbf{z}_{t^*}. \tag{2}$$

Now, sampling $p(\mathbf{z}_0|\mathbf{z}_{t^*})$ can be achieved by solving the reverse SDE starting from time $t = t^*$ and initialization $\bar{\mathbf{y}}$. This naturally yields an (approximate) sample of $p(\mathbf{x}|\mathbf{y}, \phi)$.

GDiff

Heurtel-Depeiges et al., 2024

Algorithm 1 GDiff: Gibbs Diffusion for Blind Denoising

Input: observation $\boldsymbol{y}$, $p_{\text{init}}(\phi)$, M

Initialize $\phi_0 \sim p_{\text{init}}(\phi)$.

Initialize $\boldsymbol{x}_0 \sim q(\boldsymbol{x} \mid \boldsymbol{y}, \phi_0)$ (Diffusion step)

for $k = 1$ **to** M **do**

$\boldsymbol{\varepsilon}_{k-1} \leftarrow \boldsymbol{y} - \boldsymbol{x}_{k-1}$

$\phi_k \sim q(\phi \mid \boldsymbol{\varepsilon}_{k-1})$ (HMC step)

$\boldsymbol{x}_k \sim q(\boldsymbol{x} \mid \boldsymbol{y}, \phi_k)$ (Diffusion step)

end for

Output: samples $(\boldsymbol{x}_k, \phi_k)_{1 \leq k \leq M}$.

CL-DPS

Ye and Ferrao, 2026

- The first DM-based method that can solve *blind nonlinear* inverse problems
- **Objective:** Learn a Compatibility Critic that evaluates if a noisy latent state x_t and a measurement y are consistent.
  - **Standard MoCo:**
    - Pair: (Crop A, Crop B) from the same image.
    - Goal: Learn semantic invariance.
    - minimizes the InfoNCE loss
  - **CL-DPS (Cross-Modal MoCo):**
    - Pair: (**Noisy Image $\mathbf{x}_t$**, **Measurement $\mathbf{y}$**) from the same clean image $\mathbf{x}_0$.
    - Goal: Learn Physical Consistency between the latent space and the measurement space.
- **Key Benefit:** Solving *blind* inverse problems by following the learned gradients, without explicit knowledge of the degradation operator.

CL-DPS

Ye and Ferrao, 2026

To train the critic without fixing a specific operator, CL-DPS uses **Domain Randomization**.

Given a clean image $\mathbf{x}_0 \sim \mathcal{D}_{data}$:

Path A: Diffusion Process

- Sample time step $t \sim \mathcal{U}(0, T)$.
- Add Gaussian noise ϵ:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$$

- **Input for Query:** $(\mathbf{x}_t, t)$

Path B: Physics Simulation

- Sample an operator $\mathcal{A}$ from a pool (e.g., Gaussian blur, motion blur, masking).
- Generate measurement:

$$\mathbf{y} = \mathcal{A}(\mathbf{x}_0) + \mathbf{n}$$

- **Input for Key:** $\mathbf{y}$

CL-DPS
Ye and Ferrao, 2026

CL-DPS employs dual encoders with a **Momentum Update** mechanism.

1. **Query Encoder (E_Q):**
   - Inputs: Noisy state $\mathbf{x}_t$ and time embedding t.
   - Output: Query vector $q = E_Q(\mathbf{x}_t, t)$.
   - Trainable via backpropagation.
2. **Key Encoder (E_K):**
   - Input: Measurement $\mathbf{y}$.
   - Output: Key vector $k = E_K(\mathbf{y})$.
   - No Gradient: Updated via **Momentum (EMA)**. ($m \in [0, 1)$)

$$\theta_K \leftarrow m \cdot \theta_K + (1 - m) \cdot \theta_Q$$

CL-DPS

Ye and Ferrao, 2026

The model learns to pull positive pairs together and push negative pairs apart.

- **Positive Pair (q, k_+): $\mathbf{x}_t$ and $\mathbf{y}$** generated from the *same* $\mathbf{x}_0$.
- **Negative Pairs (q, k_-): $\mathbf{y}_{neg}$** stored in the **MoCo Queue** (measurements from *different* images).

InfoNCE Loss:

$$\mathcal{L} = -\log \frac{\exp(\mathrm{sim}(q, k_+)/\tau)}{\exp(\mathrm{sim}(q, k_+)/\tau) + \sum_{k_- \in \mathrm{Queue}} \exp(\mathrm{sim}(q, k_-)/\tau)}$$

* The **Queue** allows comparing against thousands of negative samples without large batch sizes, stabilizing the training.

CL-DPS

YE AND FERRAO, 2026

1. Sample a batch of clean images $\{\mathbf{x}_0\}$.
2. **Augment:** Generate pairs $(\mathbf{x}_t, \mathbf{y})$ using random t and random operators $\mathcal{A}$.
3. **Encode:**
   - $q \leftarrow E_Q(\mathbf{x}_t, t)$ (Backprop enabled)
   - $k_+ \leftarrow E_K(\mathbf{y})$ (Momentum encoder)
4. **Compute Loss:** Calculate InfoNCE using k_+ and the negative Queue.
5. **Update:**
   - Update θ_Q using gradients.
   - Update θ_K using momentum: $\theta_K \leftarrow m\theta_K + (1 - m)\theta_Q$.
   - Enqueue current keys k_+, Dequeue oldest keys.

CL-DPS

Ye and Ferrao, 2026

Intuition: Steering the Diffusion

The Sampling Procedure (Step-by-Step):

1. **Target Encoding (Pre-computation):** Encode the measurement $\mathbf{y}$ once to get the fixed reference vector.

$$k = E_K(\mathbf{y})$$

2. **Reverse Diffusion Loop (for $t = T \dots 0$):** Calculate the Guidance by comparing the current noisy state to the target.

$$\text{Energy} = E_Q(\mathbf{x}_t, t) \cdot k \quad (\approx \log p(\mathbf{y}|\mathbf{x}_t))$$

3. **Update Rule (Score Correction):** Add the gradient of the energy to the standard diffusion score.

$$\hat{\epsilon}_t \leftarrow \underbrace{\epsilon_\theta(\mathbf{x}_t, t)}_{\text{Prior}} - \underbrace{\zeta \cdot \nabla_{x_t}(E_Q(\mathbf{x}_t, t) \cdot k)}_{\text{Guidance (Consistency)}}$$

Other works...

- Improving Diffusion Models for Inverse Problems using Manifold Constraints (NeurIPS 2022)
- GibbsDDRM: A Partially Collapsed Gibbs Sampler for Solving Blind Inverse Problems with Denoising Diffusion Restoration (ICML 2023)
- FastDiffusionEM: A Diffusion Model for Blind Inverse Problems with Application to Deconvolution (WACV 2024)
- Monte Carlo Guided Denoising Diffusion Models for Bayesian Linear Inverse Problems (ICLR 2024, Oral)
- DiracDiffusion: Denoising and Incremental Reconstruction with Assured Data-Consistency (ICML 2024)
- FlowDPS: Flow-Driven Posterior Sampling for Inverse Problems (ICCV 2025)
- Noise-Adaptive Diffusion Sampling for Inverse Problems without Task Specific Tuning (ICLR 2026)

References I

- Choi, J., Kim, S., Jeong, Y., Gwon, Y., & Yoon, S. (2021).**Ilvr: Conditioning method for denoising diffusion probabilistic models.** *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 14367–14376.
- Chung, H., Kim, J., Kim, S., & Ye, J. C. (2023).**Parallel diffusion models of operator and image for blind inverse problems.** *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6059–6069.
- Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., & Ye, J. C. (2023).**Diffusion posterior sampling for general noisy inverse problems.** *The Eleventh International Conference on Learning Representations*.
- Heurtel-Depeiges, D., Margossian, C. C., Ohana, R., & Blancard, B. R.-S. (2024).**Listening to the noise: Blind denoising with gibbs diffusion.** *Proceedings of the 41st International Conference on Machine Learning*, 18284–18304.
- Song, J., Vahdat, A., Mardani, M., & Kautz, J. (2023).**Pseudoinverse-guided diffusion models for inverse problems.** *The Eleventh International Conference on Learning Representations*.
- Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2021).**Score-based generative modeling through stochastic differential equations.** *International Conference on Learning Representations*.

References II

Ye, L., & Ferrao, P. (2026). **Cl-dps: A contrastive learning approach to blind inverse problem solving via diffusion posterior sampling.** *The Fourteenth International Conference on Learning Representations*.