DM-BASED SR

Choeun Kim

IDEA

June 4, 2026

Part I

BACKGROUND

Generative Modeling via Reverse SDE

- Consider a forward SDE that diffuses data into noise:

$$d\mathbf{x}_t = f(\mathbf{x}_t, t)dt + g(t)dW_t, \quad \mathbf{x}_0 \sim p_0(\mathbf{x})$$

  where W_t is the standard Brownian motion, and p_0 is our unknown data distribution.
- According to Anderson's theorem, the corresponding **reverse-time process** shares the same marginal distributions $p_t(\mathbf{x}_t)$ for $t \in [0, 1]$:

$$d\mathbf{x}_t = \left\{ f(\mathbf{x}_t, t) - g^2(t)\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) \right\} dt + g(t)d\bar{W}_t$$

  where dt flows backward from 1 to 0, and $d\bar{W}_t$ is a backward Brownian motion.
- **Key Insight:** If we can estimate the **score function** $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$, we can start from a known noise distribution p_1 and simulate backward to sample from p_0.

WHY OU PROCESS?

STATIONARY & LIMITING DISTRIBUTION

- To sample from the reverse SDE, we must know the exact distribution at the terminal time, $p_1(\mathbf{x}_1)$.
- Therefore, we need a forward process whose **limiting distribution** is easy to sample from (e.g., Standard Gaussian).
- The **Ornstein-Uhlenbeck (OU) Process** is a perfect candidate:

$$d\mathbf{x}_t = -\theta \mathbf{x}_t dt + \sigma dW_t$$

- **Property 1 Stationary:** If $\mathbf{x}_0 \sim \mathcal{N}\left(0, \frac{\sigma^2}{2\theta}\mathbf{I}\right)$, the distribution never changes over time ($\partial_t p_t = 0$).
- **Property 2 Limiting & Ergodic:** Crucially, **no matter what the initial distribution p_0 is** (even complex image data), as $t \to \infty$, p_t unconditionally converges to the stationary distribution:

$$\lim_{t\to\infty} p_t(\mathbf{x}_t) = \mathcal{N}\left(0, \frac{\sigma^2}{2\theta}\mathbf{I}\right)$$

VP-SDE AS A TIME-RESCALED OU

- In generative models, we scale noise over time using a schedule $\beta(t) > 0$. This leads to the **VP-SDE** (used in DDPM):

$$d\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t dt + \sqrt{\beta(t)}dW_t$$

- At first glance, the diffusion term depends on time t, unlike the standard OU process.

VP-SDE AS A TIME-RESCALED OU

- In generative models, we scale noise over time using a schedule $\beta(t) > 0$. This leads to the **VP-SDE** (used in DDPM):

$$d\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t dt + \sqrt{\beta(t)}dW_t$$

- At first glance, the diffusion term depends on time t, unlike the standard OU process.
- **The Mathematical Trick (Time Rescaling)**: By redefining the time scale based on cumulative noise, $\tau(t) = \int_0^t \beta(s)ds$, the VP-SDE transforms into:

$$d\mathbf{x}_\tau = -\frac{1}{2}\mathbf{x}_\tau d\tau + dW_\tau$$

  which is a standard, time-invariant OU process ($\theta = \frac{1}{2}, \sigma = 1$).
- **Conclusion:** Inheriting the ergodicity of the OU process, if T is large enough, the terminal distribution $p_T(\mathbf{x}_T)$ of the VP-SDE becomes **exactly the standard normal distribution** $\mathcal{N}(0, \mathbf{I})$, regardless of p_0.

How to Learn the Score? Denoising Score Matching

- **Goal:** Train a neural network $\mathbf{s}_\theta(\mathbf{x}_t, t)$ to estimate the true score $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$.
- **Problem:** The marginal distribution $p_t(\mathbf{x}_t)$ is highly complex and intractable because it requires integrating over all possible original data $\mathbf{x}_0$.

How to Learn the Score? Denoising Score Matching

- **Goal:** Train a neural network $\mathbf{s}_\theta(\mathbf{x}_t, t)$ to estimate the true score $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$.
- **Problem:** The marginal distribution $p_t(\mathbf{x}_t)$ is highly complex and intractable because it requires integrating over all possible original data $\mathbf{x}_0$.
- **Solution (Denoising Score Matching):** Instead of the marginal score, we use the **conditional score** from the forward diffusion process.
- For a known forward process, the transition kernel is Gaussian: $p_{0t}(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \mu(\mathbf{x}_0, t), \sigma(t)^2\mathbf{I})$.
- Its score is perfectly tractable and simply points back to the clean data:

$$\nabla_{\mathbf{x}_t} \log p_{0t}(\mathbf{x}_t|\mathbf{x}_0) = -\frac{\mathbf{x}_t - \mu(\mathbf{x}_0, t)}{\sigma(t)^2} \propto -\boldsymbol{\epsilon}$$

How to Learn the Score? Denoising Score Matching

- **Goal:** Train a neural network $\mathbf{s}_\theta(\mathbf{x}_t, t)$ to estimate the true score $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$.
- **Problem:** The marginal distribution $p_t(\mathbf{x}_t)$ is highly complex and intractable because it requires integrating over all possible original data $\mathbf{x}_0$.
- **Solution (Denoising Score Matching):** Instead of the marginal score, we use the **conditional score** from the forward diffusion process.
- For a known forward process, the transition kernel is Gaussian: $p_{0t}(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \mu(\mathbf{x}_0, t), \sigma(t)^2\mathbf{I})$.
- Its score is perfectly tractable and simply points back to the clean data:

$$\nabla_{\mathbf{x}_t} \log p_{0t}(\mathbf{x}_t|\mathbf{x}_0) = -\frac{\mathbf{x}_t - \mu(\mathbf{x}_0, t)}{\sigma(t)^2} \propto -\boldsymbol{\epsilon}$$

- **Mathematical Equivalence:** Minimizing the expected distance to the conditional score is equivalent to matching the intractable marginal score!

$$\arg\min_\theta \mathbb{E}_{t,\mathbf{x}_0,\mathbf{x}_t} \left[\left\| \mathbf{s}_\theta(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log p_{0t}(\mathbf{x}_t|\mathbf{x}_0) \right\|^2 \right]$$

Intuition Behind Score Matching and Reverse SDE

- Note that the score is proportional to the noise ϵ —not just any random noise, but the specific noise used to perturb $\mathbf{x}_0$ into $\mathbf{x}_t$.
- Thus, in the diffusion model literature, some interpret the training process as "score matching", while others refer to it as "noise prediction" — both perspectives are mathematically equivalent.

Intuition Behind Score Matching and Reverse SDE

- Note that the score is proportional to the noise ϵ —not just any random noise, but the specific noise used to perturb $\mathbf{x}_0$ into $\mathbf{x}_t$.
- Thus, in the diffusion model literature, some interpret the training process as "score matching", while others refer to it as "noise prediction" — both perspectives are mathematically equivalent.
- DDPM introduced its method entirely based on "noise prediction," deriving the objective function via the ELBO.
- The seminal work by Song et al., 2021 established the formal connection between DDPM and the Reverse SDE via Score Matching, as described in the previous section.

Intuition Behind Score Matching and Reverse SDE

- Note that the score is proportional to the noise ϵ —not just any random noise, but the specific noise used to perturb $\mathbf{x}_0$ into $\mathbf{x}_t$.
- Thus, in the diffusion model literature, some interpret the training process as "score matching", while others refer to it as "noise prediction" — both perspectives are mathematically equivalent.
- DDPM introduced its method entirely based on "noise prediction," deriving the objective function via the ELBO.
- The seminal work by Song et al., 2021 established the formal connection between DDPM and the Reverse SDE via Score Matching, as described in the previous section.

- Intuitively, the score-based reverse process is closely related to Langevin Dynamics, which iteratively steers the state toward high-density (high-posterior) regions using the score function. A similar philosophy is also seen in gradient descent optimization.
- From a noise prediction perspective, the intuition aligns perfectly: since the noise vector has the opposite sign of the score function, following the reverse SDE is equivalent to iteratively subtracting noise from the current state. By repeating this process, we successfully recover a clean sample from the target data distribution.

Taxonomy of Diffusion Research

1. **Score Network Architecture (From CNN to Transformer)**
   - Motivation: Overcoming the scalability limits of traditional CNN-based U-Nets.
   - Key Frameworks: **U-ViT**, **DiT** (Diffusion Transformers)
   - Impact: Enables stable scaling for massive datasets and high-resolution generation.

2. **Fast Sampling Path (Advanced SDE/ODE Solvers)**
   - Motivation: Reducing the notoriously slow sampling speed (typically 1000 steps).
   - Key Frameworks: **DDIM** (Deterministic ODE), **DPM-Solver** (High-order solver)
   - Impact: Drastically accelerates sampling down to 10–20 steps without retraining.

Taxonomy of Diffusion Research

3. **Guidance & Controllable Generation**
   - Motivation: Controlling the generative trajectory via external conditions or internal representation manipulation.
   - Key Frameworks:
     - **CFG** (Classifier-Free Guidance): Conditioning via text or labels.
     - **Latent Disentanglement**: Manipulating isolated semantic features (e.g., age, expression) for fine-grained editing (e.g., Asyrp, DiffAE).
     - **DPS** (Diffusion Posterior Sampling): Guidance via physical measurement likelihood.
   - Impact: Shifts diffusion from a random sampler to a highly controllable tool. **DPS serves as the foundational baseline for today's main topic (BlindDPS).**

Taxonomy of Diffusion Research

4. **Generalized Forward Process (Beyond Fixed Gaussian)**
   - Motivation: Breaking free from predefined, data-independent Gaussian noise to learn domain-specific trajectories.
   - Key Frameworks: **DSBM** (Schrödinger Bridge), **I2SB**, **NFDM**

5. **Large Language Diffusion Models (LLDM)**
   - Motivation: Overcoming the sequential, left-to-right bottleneck of Autoregressive LLMs (like GPT) by applying continuous diffusion to discrete text tokens.
   - Key Frameworks: **D3PM** (Discrete Diffusion), **LLaDA**, **Diffusion-LM** (also addressed controllable generation, i.e., persona guidance)
   - Impact: Enables non-autoregressive parallel decoding and highly controllable text generation (e.g., infilling, exact length constraints).

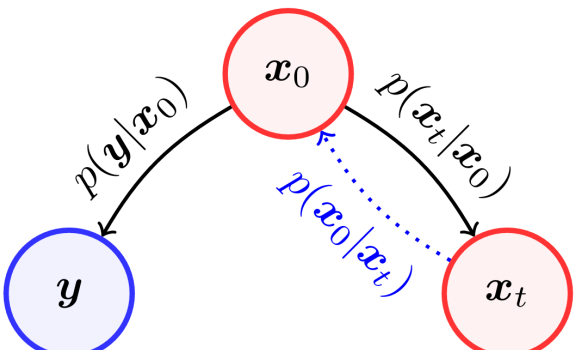
Taxonomy of Diffusion Research

6. **Next Paradigm: Optimal Transport (OT) & Flow Matching**
   - Motivation: Minimizing the transport cost between distributions by straightening the curved, inefficient SDE paths.
   - Key Frameworks: **Entropic OT**, **Rectified Flow**, **Flow Matching**
   - Impact: Unifies SDEs and ODEs into optimal, straight-line vector fields, dominating the latest generation of fundamental models.

Part II

DPS

DIFFUSION FOR INVERSE PROBLEMS

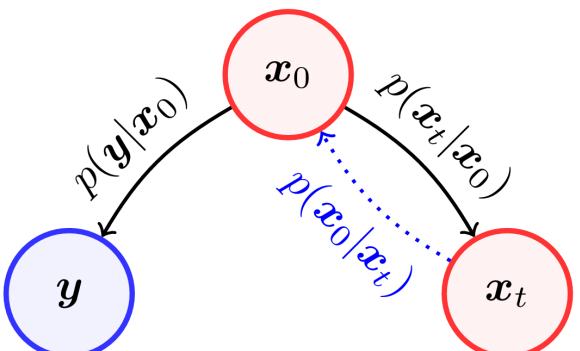


- **Inverse Problem:** We want to recover the clean target data $\mathbf{x}_0$ (at $t = 0$) from a degraded observation $\mathbf{y}$:

$$\mathbf{y} = \mathcal{A}(\mathbf{x}_0) + \mathbf{n}, \quad \mathbf{n} \sim \mathcal{N}(0, \sigma_y^2 \mathbf{I}) \tag{1}$$

  where $\mathcal{A}$ is a degradation operator (e.g., blurring, subsampling). When it is known, the inverse problem is non-blind; otherwise, it is blind.
- **Diffusion State $\mathbf{x}_t$:** Let $\mathbf{x}_t$ be the noisy intermediate state of our target $\mathbf{x}_0$ at diffusion time step t.

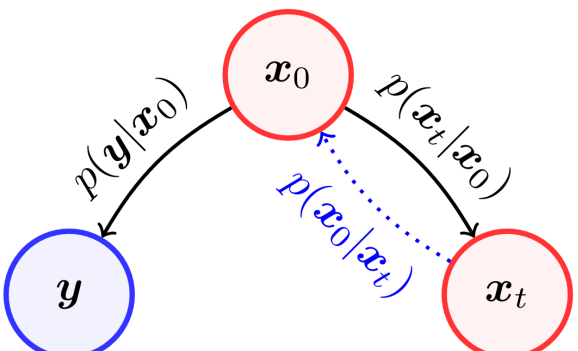
Diffusion for Inverse Problems



- **Goal:** Sample from the posterior distribution $p(\mathbf{x}_0|\mathbf{y})$. By Bayes' rule, the **posterior score** for the noisy state $\mathbf{x}_t$ is split into two terms:

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) = \underbrace{\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)}_{\text{Unconditional Score}} + \underbrace{\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)}_{\text{Likelihood Score}}$$

Diffusion for Inverse Problems



- For the prior score, we use the pre-trained unconditional diffusion model.
- The key is to estimate the likelihood score $\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$ since it is mathematically intractable because the mapping from a noisy latent $\mathbf{x}_t$ to $\mathbf{y}$ is unknown.
- **Intuition:** The likelihood score guides the current state toward better alignment with the observation.

Diffusion Posterior Sampling

Chung, Kim, Mccann, et al., 2023

- **Theoretical Basis:** To handle the intractable likelihood, we integrate over the clean data space $\mathbf{x}_0$:

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

- **Tweedie's Approximation:** DPS approximates this expectation by substituting $\mathbf{x}_0$ with its conditional expectation $\hat{\mathbf{x}}_0(\mathbf{x}_t) = \mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]$:[1]

$$p(\mathbf{y}|\mathbf{x}_t) \approx p(\mathbf{y}|\hat{\mathbf{x}}_0(\mathbf{x}_t)), \quad \text{where } \hat{\mathbf{x}}_0(\mathbf{x}_t) = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t + (1-\bar{\alpha}_t)\mathbf{s}_\theta(\mathbf{x}_t, t)\right)$$

- **Posterior Score:** Since $p(\mathbf{y}|\hat{\mathbf{x}}_0) \propto \exp\left(-\frac{1}{2\sigma_y^2}\|\mathbf{y} - \mathcal{A}(\hat{\mathbf{x}}_0)\|_2^2\right)$, the likelihood score becomes fully tractable:

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) \approx \mathbf{s}_\theta(\mathbf{x}_t, t) - \frac{\mathbf{1}}{\mathbf{2}\sigma_{\mathbf{y}}^{\mathbf{2}}}\nabla_{\mathbf{x}_t}\|\mathbf{y} - \mathcal{A}(\mathbf{x_0}(\mathbf{x_t}))\|_{\mathbf{2}}^{\mathbf{2}} \tag{2}$$

[1] The conditional expectation is derived by applying Tweedie's formula.

Diffusion Posterior Sampling

Chung, Kim, Mccann, et al., 2023

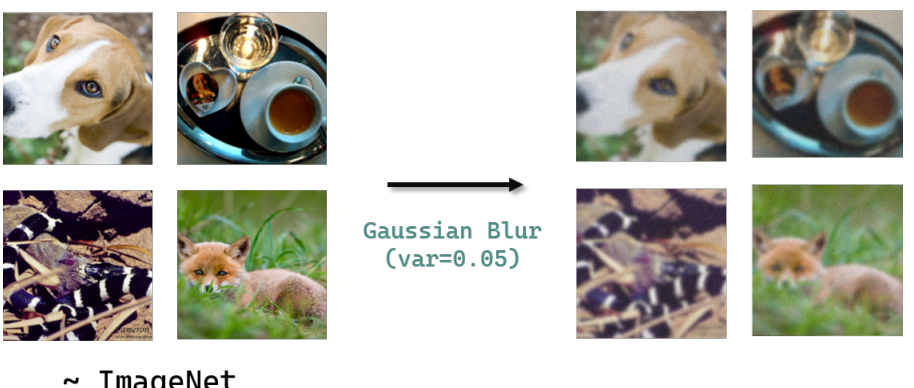


Figure. Training data preparation

Diffusion Posterior Sampling

Chung, Kim, Mccann, et al., 2023

Algorithm 1 DPS - Gaussian

Require: $N, \boldsymbol{y}, \{\zeta_i\}_{i=1}^N, \{\tilde{\sigma}_i\}_{i=1}^N$

1: $\boldsymbol{x}_N \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
2: **for** $i = N-1$ **to** 0 **do**
3: $\quad \hat{\boldsymbol{s}} \leftarrow \boldsymbol{s}_\theta(\boldsymbol{x}_i, i)$
4: $\quad \hat{\boldsymbol{x}}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_i}}(\boldsymbol{x}_i + (1-\bar{\alpha}_i)\hat{\boldsymbol{s}})$
5: $\quad \boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
6: $\quad \boldsymbol{x}'_{i-1} \leftarrow \frac{\sqrt{\alpha_i}(1-\bar{\alpha}_{i-1})}{1-\bar{\alpha}_i}\boldsymbol{x}_i + \frac{\sqrt{\bar{\alpha}_{i-1}}\beta_i}{1-\bar{\alpha}_i}\hat{\boldsymbol{x}}_0 + \tilde{\sigma}_i \boldsymbol{z}$
7: $\quad \boldsymbol{x}_{i-1} \leftarrow \boldsymbol{x}'_{i-1} - \zeta_i \nabla_{\boldsymbol{x}_i} \|\boldsymbol{y} - \mathcal{A}(\hat{\boldsymbol{x}}_0)\|_2^2$
8: **end for**
9: **return** $\hat{\mathbf{x}}_0$

- Line 1: Start from Gaussian noise.
- Lines 3–6: Unconditional DM reverse process (used DDPM ancestral sampling).
- **Line 7:** Alignment with the observation.

BLINDDPS

CHUNG, KIM, KIM, AND YE, 2023

- Now, consider the following model:

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

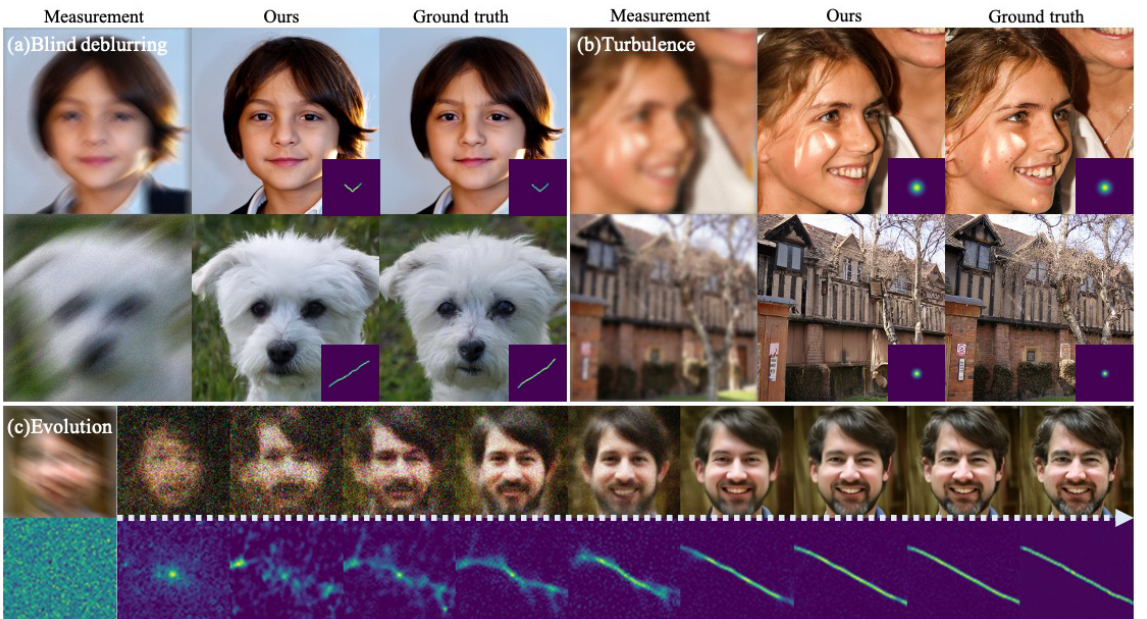
  where φ is the parameter of the forward operator. Unlike (1), the forward operator is unknown.
- In this setting, in order to utilize (2), we must simultaneously estimate φ.
- The paper considers a scenario where the forward operator is a blur kernel. That is,

$$\mathbf{y} = \mathbf{k} * \mathbf{x} + \mathbf{n}, \quad \mathbf{n} \sim \mathcal{N}(0, \sigma_y^2 \mathbf{I})$$

- Assume that the clean image $\mathbf{x}$ and the kernel $\mathbf{k}$ are statistically independent.

BLINDDPS

CHUNG, KIM, KIM, AND YE, 2023



PARALLEL REVERSE SDES FOR IMAGE AND KERNEL

1. This time, we must estimate both the kernel $\mathbf{k}$ and the clean image $\mathbf{x}$ based on the single observation $\mathbf{y}$.
2. That is, we need to sample $(\mathbf{k}, \mathbf{x})$ from the joint posterior $p(\mathbf{k}, \mathbf{x}|\mathbf{y})$. Thus, we formulate the following reverse SDE:

$$d\mathbf{e}_t = \left[-\frac{\beta(t)}{2}\mathbf{e}_t - \beta(t)\nabla_{\mathbf{e}_t} \log p(\mathbf{e}_t|\mathbf{y}) \right] dt + \sqrt{\beta(t)} d\bar{\mathbf{w}}_t,$$

   where $\mathbf{e} := (\mathbf{k}, \mathbf{x})$.
3. This naturally decomposes into the following parallel reverse SDEs:

$$\begin{aligned}
d\mathbf{x}_t &= \left[-\frac{\beta(t)}{2}\mathbf{x}_t - \beta(t)\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t, \mathbf{k}_t|\mathbf{y}) \right] dt + \sqrt{\beta(t)} d\bar{\mathbf{w}}_t \\
&= \left[-\frac{\beta(t)}{2}\mathbf{x}_t - \beta(t) \left(\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t, \mathbf{k}_t) \right) \right] dt + \sqrt{\beta(t)} d\bar{\mathbf{w}}_t, \\
d\mathbf{k}_t &= \left[-\frac{\beta(t)}{2}\mathbf{k}_t - \beta(t)\nabla_{\mathbf{k}_t} \log p(\mathbf{x}_t, \mathbf{k}_t|\mathbf{y}) \right] dt + \sqrt{\beta(t)} d\bar{\mathbf{w}}_t \\
&= \left[-\frac{\beta(t)}{2}\mathbf{k}_t - \beta(t) \left(\nabla_{\mathbf{k}_t} \log p(\mathbf{k}_t) + \nabla_{\mathbf{k}_t} \log p(\mathbf{y}|\mathbf{x}_t, \mathbf{k}_t) \right) \right] dt + \sqrt{\beta(t)} d\bar{\mathbf{w}}_t.
\end{aligned}$$

Core Philosophy of BlindDPS: Parallel Reverse Process

1. **Why use a Diffusion Prior for the Kernel?**
   - In nature, physical corruptions (e.g., motion blur, turbulence) are not entirely random. There exists an implicit, natural "distribution of kernels."
   - Instead of using hand-crafted heuristics (like sparsity alone), we can train a diffusion model to learn the true prior distribution of these kernels $p(\mathbf{k})$.

2. **The Role of Kernel Prior Score ($s_\theta^{\mathbf{k}}$)**
   - Ensures the estimated kernel stays within the valid support of natural blur kernels.
   - It unconditionally pulls the noisy kernel toward a "realistic shape," preventing the generation of out-of-distribution or nonsensical kernels.

3. **The Role of Likelihood Score ($\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t, \mathbf{k}_t)$)**
   - While the prior ensures the kernel is "realistic," the likelihood score aligns both the kernel and the image with our specific observation $\mathbf{y}$.
   - **Conclusion:** Joint sampling finds the optimal intersection between the natural distribution (Prior) and the observation (Likelihood).

Mathematical Formulation: SDEs & Approximation

1. **Ideal Reverse SDEs for Joint Sampling**

$$d\mathbf{x}_t = \left(-\frac{\beta(t)}{2}\mathbf{x}_t - \beta(t)\left[\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t, \mathbf{k}_t) + s^{\mathbf{x}}_{\theta^*}(\mathbf{x}_t, t)\right] \right) dt + \sqrt{\beta(t)} d\bar{\mathbf{w}}_t,$$

   where $s^{\mathbf{x}}_{\theta^*}$ is a pre-trained score network for the distribution of $\mathbf{x}$.

   - **Bottleneck:** The time-conditional likelihood $\log p(\mathbf{y}|\mathbf{x}_t, \mathbf{k}_t)$ is mathematically intractable.

2. **Tweedie's Formula & Tractable Approximation**
   - We use Tweedie's formula to estimate the fully denoised states $\hat{\mathbf{x}}_0$ and $\hat{\mathbf{k}}_0$ using only the prior scores:

$$\hat{\mathbf{x}}_0(\mathbf{x}_t) := \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t + (1-\bar{\alpha}_t) s^{\mathbf{x}}_{\theta^*}(\mathbf{x}_t, t)\right)$$

   - **Approximation:** Replace the intractable likelihood with the likelihood evaluated at the estimated clean states:

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t, \mathbf{k}_t) \simeq \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\hat{\mathbf{x}}_0(\mathbf{x}_t), \hat{\mathbf{k}}_0(\mathbf{k}_t))$$

BLIND DPS SAMPLING ALGORITHM

Algorithm 1 BlindDPS — Blind Deblurring

Require: $N, \boldsymbol{y}, \alpha, \{\tilde{\sigma}_i\}_{i=1}^{N}, \lambda, R_{\boldsymbol{k}}(\cdot)$

1: $\boldsymbol{x}_N, \boldsymbol{k}_N \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
2: **for** $i = N-1$ **to** 0 **do**
3: $\quad \hat{\boldsymbol{s}}^i \leftarrow \boldsymbol{s}^i_{\theta^*}(\boldsymbol{x}_i, i)$
4: $\quad \hat{\boldsymbol{s}}^k \leftarrow \boldsymbol{s}^k_{\theta^*}(\boldsymbol{k}_i, i)$
5: $\quad \hat{\boldsymbol{x}}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_i}}(\boldsymbol{x}_i + \sqrt{1-\bar{\alpha}_i}\hat{\boldsymbol{s}}^i)$
6: $\quad \hat{\boldsymbol{k}}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_i}}(\boldsymbol{k}_i + \sqrt{1-\bar{\alpha}_i}\hat{\boldsymbol{s}}^k)$
7: $\quad \hat{\boldsymbol{k}}_0 \leftarrow \mathcal{P}_C(\hat{\boldsymbol{k}}_0)$
8: $\quad \boldsymbol{z}_i, \boldsymbol{z}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
9: $\quad \boldsymbol{x}'_{i-1} \leftarrow \frac{\sqrt{\alpha_i}(1-\bar{\alpha}_{i-1})}{1-\bar{\alpha}_i}\boldsymbol{x}_i + \frac{\sqrt{\bar{\alpha}_{i-1}}\beta_i}{1-\bar{\alpha}_i}\hat{\boldsymbol{x}}_0 + \tilde{\sigma}_i \boldsymbol{z}_i$
10: $\quad \boldsymbol{k}'_{i-1} \leftarrow \frac{\sqrt{\alpha_i}(1-\bar{\alpha}_{i-1})}{1-\bar{\alpha}_i}\boldsymbol{k}_i + \frac{\sqrt{\bar{\alpha}_{i-1}}\beta_i}{1-\bar{\alpha}_i}\hat{\boldsymbol{k}}_0 + \tilde{\sigma}_i \boldsymbol{z}_k$
11: $\quad \boldsymbol{x}_{i-1} \leftarrow \boldsymbol{x}'_{i-1} - \alpha \nabla_{\boldsymbol{x}_i} \|\boldsymbol{y} - \hat{\boldsymbol{k}}_0 * \hat{\boldsymbol{x}}_0\|_2$
12: $\quad \mathcal{L}_{\boldsymbol{k}} \leftarrow \|\boldsymbol{y} - \hat{\boldsymbol{k}}_0 * \hat{\boldsymbol{x}}_0\|_2 + \lambda R_{\boldsymbol{k}}(\hat{\boldsymbol{k}}_0)$
13: $\quad \boldsymbol{k}_{i-1} \leftarrow \boldsymbol{k}'_{i-1} - \alpha \nabla_{\boldsymbol{k}_i} \mathcal{L}_{\boldsymbol{k}}$
14: **end for**
15: **return** $\boldsymbol{x}_0, \boldsymbol{k}_0$

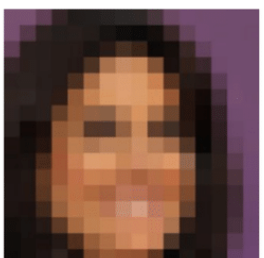
- Lines 3–10: DDPM ancestral sampling.
- **Lines 11–13**: Alignment with the observation.

Part III

SUPER-RESOLUTION

SRDIFF

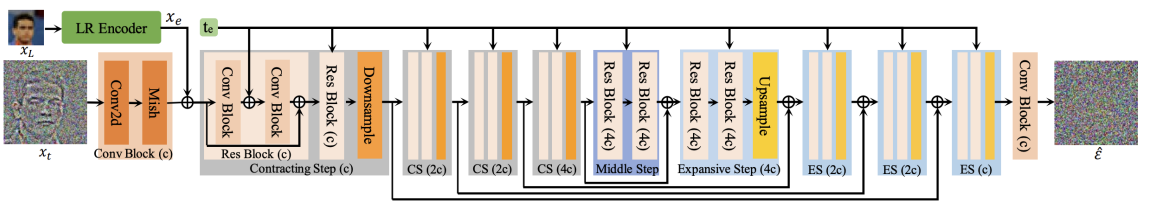

ground truth


low resolution


super-resolution (64x)

- The Super-Resolution (SR) task aims to estimate a high-resolution (HR) image from an observed low-resolution (LR) image, ensuring the output aligns perfectly with the observation.
- **Methodology:**
  1. Upscale the LR image $\mathbf{x}^L$ using an interpolation method (e.g., bicubic). This yields an upscaled LR image $up(\mathbf{x}^L)$ with the same spatial dimensions as the HR image.
  2. Compute the residual $\mathbf{x}^R = \mathbf{x}^H - up(\mathbf{x}^L)$.
  3. Define the forward diffusion on the residual: $\mathbf{x}_t^R = \sqrt{\bar{\alpha}_t}\mathbf{x}^R + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$. Train the noise prediction model $\epsilon_\theta(\mathbf{x}_t^R, \mathbf{x}^L, t)$. The objective function is $\|\epsilon - \epsilon_\theta(\mathbf{x}_t^R, \mathbf{x}^L, t)\|$.

SRDIFF



- Recall that sampling from the conditional distribution $p(\mathbf{x}|\mathbf{y})$ requires access to the exact posterior score $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y})$.
- While DPS analytically splits the score using Bayes' rule, SRDiff forces the model to learn the conditional score implicitly.
- However, the proposed objective naturally induces the network to converge toward the unconditional score, rather than the true conditional score.
- Consequently, practical performance degrades unless initialization relies on the LR image. To achieve competitive results, the reverse process must start from the perturbed LR image $\sqrt{\bar{\alpha}_T}\mathbf{x}^L + \sqrt{1-\bar{\alpha}_T}\boldsymbol{\epsilon}$.
- This procedure closely resembles the methodology of diffusion purification models.

REFERENCES I


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*.


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*.