Improving Diffusion Inverse Problem Solving with Decoupled Noise Annealing

Junjie Shi

Wuhan University

April 29, 2025

Contents

Background

Target — Recover the true value x_0 from its noisy measurement y

Challenges of inverse problems

- Do not have a unique solution

Challenges of previous diffusion sampling methods

- Struggles to correct errors from earlier sampling steps, and thus incapable of tackling complicated nonlinear inverse problems

Bayesian inverse problem farmework

- Forward model: $y = \mathcal{A}(x_0) + n, \;\; n \sim N(0, \beta_y^2 I)$
- Controllable generation — want to sample from the posterior $p(x_0|y)$
- Challenge — how to incorporate information in measurement y

Contents

Solving inverse problem with DPS

- $y = \mathcal{A}(x_0) + n, \quad y, n \in \mathbb{R}^n, x_0 \in \mathbb{R}^d$ and $n \sim N(0, \sigma^2 I_n)$
- Choosing VP-SDE (continuous version of DDPM):

$$\begin{cases} \mathrm{d}x_t = -\frac{\beta(t)}{2} x_t \, \mathrm{d}t + \sqrt{\beta(t)} \, \mathrm{d}w_t \quad \text{(forward)} \\ \mathrm{d}x_t = \left[-\frac{\beta(t)}{2} x_t - \beta(t) \nabla_{x_t} \log p_t(x_t) \right] \mathrm{d}t + \sqrt{\beta(t)} \, \mathrm{d}\bar{w}_t \quad \text{(reverse)} \end{cases}$$

- Bayesian framework with

$$\mathrm{d}x_t = \left[-\frac{\beta(t)}{2} x_t - \beta(t) \nabla_{x_t} \log p_t(x_t | y) \right] \mathrm{d}t + \sqrt{\beta(t)} \, \mathrm{d}\bar{w}_t$$

$$\Rightarrow \mathrm{d}x_t = \left[-\frac{\beta(t)}{2} x_t - \beta(t) \left(\nabla_{x_t} \log p_t(x_t) + \nabla_{x_t} \log p_t(y | x_t) \right) \right] \mathrm{d}t + \sqrt{\beta(t)} \, \mathrm{d}\bar{w}_t$$

- $\nabla_{x_t} \log p_t(x_t) \simeq s_{\theta^*}(x_t, t)$: a pre-trained generative model; $\nabla_{x_t} \log p_t(y | x_t)$: term to be tackled with

Approximating $p_t(y|x_t)$ and then $\nabla_{x_t} \log p_t(y|x_t)$

- Incorporating information in x_0 gives

$$p(y|x_t) = \int p(y|x_0, x_t)p(x_0|x_t)\,\mathrm{d}x_0 = \int p(y|x_0)\underline{p(x_0|x_t)}\,\mathrm{d}x_0, \quad y|x_0 \sim N(\mathcal{A}(x_0), \sigma^2 I_n)$$

- VP-SDE (or DDPM) sampling gives the posterior mean representation

$$\hat{x}_0(x_t) := \mathbb{E}[x_0|x_t] = \frac{1}{\sqrt{\bar{\alpha}(t)}}\left(x_t + (1-\bar{\alpha}(t))\nabla_{x_t} \log p_t(x_t)\right), \quad \nabla_{x_t} \log p_t(x_t) \simeq s_{\theta^*}(x_t, t)$$

- Use approximation

$$p(y|x_t) = \mathbb{E}_{x_0 \sim p(x_0|x_t)}[p(y|x_0)] \simeq p(y|\mathbb{E}_{x_0 \sim p(x_0|x_t)}[x_0]) = p(y|\hat{x}_0)$$

(the Jensen gap between $p(y|x_t)$ and $p(y|\hat{x}_0)$ is upper bounded)

- $p(y|x_t) \simeq p(y|\hat{x}_0) \;\Rightarrow\; \nabla_{x_t} \log p_t(y|x_t) \simeq \nabla_{x_t} \log p_t(y|\hat{x}_0)$ No guarantee ??

Algorithm in application

Algorithm 1 DPS - Gaussian

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

1: $\boldsymbol{x}_N \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
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: $\quad \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
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** $\hat{\mathbf{x}}_0$

- $\nabla_{x_t} \log p_t(y|x_t) \simeq \nabla_{x_t} \log p_t(y|\hat{x}_0) = -\frac{1}{\sigma^2}\nabla_{x_t}\|y - \mathcal{A}(\hat{x}_0(x_t))\|_2^2$
- SDE decomposition
- Recursively do: denoising step (following DDPM) — correction step

Contents

Motivation of DAPS

Existing diffusion sampling methods (e.g. DPS)

- Sample from $p(x_0|y)$ by reversing the SDE with conditional score $\nabla_{x_t} \log p_t(x_t|y)$
- Denoising steps approximately sample from $p(x_t|x_{t+\Delta t}, y)$
- x_t is close to $x_{t+\Delta t}$ with small step size Δt
- x_t can at most correct local errors in $x_{t+\Delta t}$ but struggles to correct global errors
- Facing challenges in complicated nonlinear inverse problems

Highlights of DAPS

- Do not repetitively sample from $p(x_t|x_{t+\Delta t}, y)$ following a specific SDE/ODE
- Factorizing and then sampling from $p(x_t|y)$ recursively to get $p(x_0|y)$
- Decoupling helps correct global errors

DAPS — **D**: decoupling x_t and $x_{t+\Delta t}$; **A**: noise-annealing sampling process

Setting of DAPS

Problem — $y = \mathcal{A}(x_0) + n,\ \ n \sim N(0, \beta_y^2 I)$. Give $y, \mathcal{A}(\cdot)$ and β_y^2 to sample $p(x_0|y)$

Diffusion process used (unconditional)

$$\begin{cases} \text{(Forward)} \quad \mathrm{d}x_t = \sqrt{2\dot{\sigma}_t\sigma_t}\,\mathrm{d}w_t = \sqrt{\dfrac{\mathrm{d}\sigma_t^2}{\mathrm{d}t}}\,\mathrm{d}w_t \quad \text{(VE-SDE)} \\ \text{(Reverse)} \quad \mathrm{d}x_t = -2\dot{\sigma}_t\sigma_t\nabla_{x_t}\log p_t(x_t)\,\mathrm{d}t + \sqrt{2\dot{\sigma}_t\sigma_t}\,\mathrm{d}w_t \\ \text{(probability flow ODE)} \quad \mathrm{d}x_t = -\dot{\sigma}_t\sigma_t\nabla_{x_t}\log p_t(x_t)\,\mathrm{d}t \end{cases}$$

Properties of the forward process

- σ_t is a predefined noise schedule with $\sigma_0 = 0, \sigma_T = \sigma_{\max}$
- $x_t|x_0 \sim N(x_0, \sigma_t^2 I) \Rightarrow x_T|x_0 \sim N(x_0, \sigma_{\max}^2 I) \simeq N(0, \sigma_{\max}^2 I)$

Bayesian inverse probelms with diffusion

Conditional diffusion process

$$\begin{cases} \text{(Reverse)} \quad \mathrm{d}x_t = -2\dot{\sigma}_t\sigma_t\nabla_{x_t}\log p_t(x_t|y)\,\mathrm{d}t + \sqrt{2\dot{\sigma}_t\sigma_t}\,\mathrm{d}w_t \\ \text{(probability flow ODE)} \quad \mathrm{d}x_t = -\dot{\sigma}_t\sigma_t\nabla_{x_t}\log p_t(x_t|y)\,\mathrm{d}t \end{cases}$$

Bayesian framework of previous methods (not DAPS)

- Bayes' s formula gives $p(x_t|y) \propto p(y|x_t)p(x_t)$
- Score decomposition $\nabla_{x_t}\log p(x_t|y) = \nabla_{x_t}\log p(x_t) + \nabla_{x_t}\log p(y|x_t)$
- Reverse process becomes

$$\begin{cases} \mathrm{d}x_t = -2\dot{\sigma}_t\sigma_t\nabla_{x_t}\log p_t(x_t)\,\mathrm{d}t - 2\dot{\sigma}_t\sigma_t\nabla_{x_t}\log p_t(y|x_t)\,\mathrm{d}t + \sqrt{2\dot{\sigma}_t\sigma_t}\,\mathrm{d}w_t \\ \mathrm{d}x_t = -\dot{\sigma}_t\sigma_t(\nabla_{x_t}\log p_t(x_t) + \nabla_{x_t}\log p_t(y|x_t))\,\mathrm{d}t \end{cases}$$

- $\nabla_{x_t}\log p_t(x_t) \approx s_{\theta^*}(x_t, t)$ is modeled by a pre-trained generative model
- $\nabla_{x_t}\log p_t(y|x_t)$ is what we should tackle with

Process of DAPS

DAPS factorizes $p(x_t|y)$ into three distributions and sample from them in turn

Sample from unconditional $p_0(x_0)$**:** — denoising but without correction with y

- Sample $x_T \sim N(0, \sigma_T^2 I)$ and run ODE $\mathrm{d}x_t = -\dot{\sigma}_t \sigma_t s_{\theta^*}(x_t, t)\,\mathrm{d}t$ to get $\hat{x}_0(x_T)$

Sample from $p(x_0|x_T, y)$**:** — information of y is introduced

- Factorization $p(x_0|x_t, y) = \dfrac{p(x_0|x_t)p(y|x_0, x_t)}{p(y|x_t)} \propto p(x_0|x_t)\overbrace{p(y|x_0)}^{\text{known}}$
- Approximate $p(x_0|x_t)$ by Gaussian $N(x_0; \hat{x}_0(x_t), r_t^2 I)$ or use

$$\nabla_{x_0} \log p(x_0|x_t) = \nabla_{x_0} \log p(x_t|x_0) + \underbrace{\nabla_{x_0} \log p(x_0)}_{\simeq\, s_{\theta^*}(x_0, t_{\min})} \quad \text{(time-consuming)}$$

- Run with Langevin dynamics to sample

Sample from $x_{T-\Delta t} \sim p(x_{T-\Delta t}|y)$**:** — x_T and $x_{T-\Delta t}$ are decoupled

- Prop1 gives $x_{T-\Delta t} \sim \mathbb{E}_{x_{0|T} \sim p(x_0|x_T, y)}[N(x_{0|T}, \sigma_{T-\Delta t}^2 I)]$

Recursivly do the sampling with $\sigma_T > \sigma_{T-\Delta t} > \cdots > \sigma_0 = 0$

Visualization of DAPS

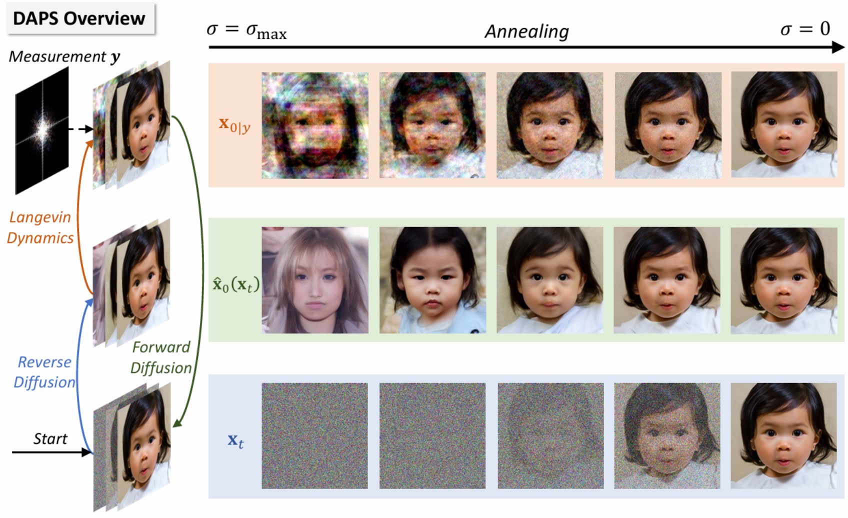

Figure 1: Overview of DAPS

DAPS summary

Pocess of DAPS — summary

- Sample $x_T \sim p(x_T|y) \approx p(x_T;\sigma_T) \approx N(0,\sigma_T^2 I)$ for σ_T large enough
- Solve the unconditional probability flow ODE $\mathrm{d}x_t = -\dot{\sigma}_t\sigma_t s_{\theta^*}(x_t,t)\,\mathrm{d}t$ starting at x_T to get $\hat{x}_0(x_T)$ — denoising but without correction with y
- Sample $x_{0|T} \sim p(x_0|x_T,y)$ — information of y is introduced here
  - y is conditionally independent from x_t given x_0
  - $p(x_0|x_t,y) = \frac{p(x_0|x_t)p(y|x_0,x_t)}{p(y|x_t)} \propto p(x_0|x_t)p(y|x_0)$
  - $p(x_0|x_t) \approx N(x_0;\hat{x}_0(x_t),r_t^2 I)$ (Gaussian approximation), $p(y|x_0) = N(y;\mathcal{A}(x_0),\beta_y^2 I)$
  - Use MCMC method like Langevin dynamics to sample from $p(x_0|x_t,y)$
- Sample $x_{T-\Delta t} \sim p(x_{T-\Delta t}|y)$ by $x_{T-\Delta t} \overset{\text{Prop1}}{\sim} \mathbb{E}_{x_{0|T}\sim p(x_0|x_T,y)}[N(x_{0|T},\sigma_{T-\Delta t}^2 I)]$
- Recursively do the process above till σ_t annealed from σ_T to 0
- Finally we' ve sampled $x_0 \sim p(x_0|y)$

Algorithm of DAPS

Algorithm 1 Decoupled Annealing Posterior Sampling (DAPS)

Require: Score model s_θ, measurement $\mathbf{y}$, noise schedule σ_t, $(t_i)_{i\in\{0,\ldots,N_A\}}$.

Sample $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \sigma_T^2 \boldsymbol{I})$.

for $i = N_A, N_A - 1, \ldots, 1$ **do**

Initial $\mathbf{p}^{(0)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ for HMC only

Compute $\hat{\mathbf{x}}_0^{(0)} = \hat{\mathbf{x}}_0(\mathbf{x}_{t_i})$ by solving the probability flow ODE in Eq. (48) with s_θ

for $j = 0, \ldots, N-1$ **do**

Langevin dynamics:

$$\hat{\mathbf{x}}_0^{(j+1)} \leftarrow \hat{\mathbf{x}}_0^{(j)} + \eta_t \left(\nabla_{\hat{\mathbf{x}}_0} \log p(\hat{\mathbf{x}}_0^{(j)} | \mathbf{x}_{t_i}) + \nabla_{\hat{\mathbf{x}}_0} \log p(\mathbf{y} | \hat{\mathbf{x}}_0^{(j)})\right) + \sqrt{2\eta_t}\boldsymbol{\epsilon}_j,\ \boldsymbol{\epsilon}_j \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}).$$

or HMC:

$$(\hat{\mathbf{x}}_0^{(j+1)}, \mathbf{p}^{(j+1)}) \leftarrow \text{Hamiltonian-Dynamics}(\hat{\mathbf{x}}_0^{(j)}, \mathbf{p}^{(j)}),$$

or Metropolis Hasting:

$$\hat{\mathbf{x}}_0^{(j+1)} \leftarrow \text{Metropolis-Hasting}(\hat{\mathbf{x}}_0^{(j)})$$

end for

Sample $\mathbf{x}_{t_{i-1}} \sim \mathcal{N}(\hat{\mathbf{x}}_0^{(N)}, \sigma_{t_{i-1}}^2 \boldsymbol{I})$.

end for

Return $\mathbf{x}_0$

More details about DAPS

- $p(x_0|x_t)$ approximation — Gaussian v.s. approximation with s_{θ^*}
- The starting point of Langevin — $\hat{x}_0(x_t)$
- Implementation of sampling $x_{t-\Delta t} \sim \mathbb{E}_{x_{0|t} \sim p(x_0|x_t,y)}[N(x_{0|t}, \sigma^2_{t-\Delta t} I)]$
- The geometric intuition of the algorithm — why is DAPS good (An 2d example)
- Efficiency of DAPS — different settings of NFE (Appendix E)
- Efficiency — DAPS v.s. DPS (how can DAPS be faster)

Problems and future extension

- No guarantee for gradient approximation in DPS ?

  To be explored

- x_t and $x_{t+\Delta t}$ are always conditionally independent given x_0 ?

  Yes. This is ensured by the SDE used (VP and VE-SDE)

- Why not sample from $p(x_0|y) \propto p(y|x_0)p(x_0)$ directly using Langevin ?

  When $p(x_0)$ is not good enough, using Langevin directly without diffusion can hardly yield good results

- In reality, is measurement operator $\mathcal{A}(\cdot)$ known or not ? The derivative of $\mathcal{A}$?

  Some cases yes, many others no. Physical models may be used to model $\mathcal{A}(\cdot)$. Numerical methods can be used to calculate the derivative but with no guarantee

Thank you!