

Posterior Sampling Overview

References:

1. Advait Parulekar and Litu Rout, Personal Communication, 2025.
2. F. Coeurdoux, N. Dobigeon and P. Chainais, Plug-and-play split Gibbs sampler: Embedding deep generative priors in Bayesian inference, arXiv 2304.11134, 2023.
3. X. Xu and Y. Chi, Provably robust score-based diffusion posterior sampling for plug-and-play image reconstruction, arXiv: 2403.17042, 2024.
4. S. Chatterjee and P. Diaconis, The sample size required in importance sampling, arXiv: 1511.01437, 2015.
5. M. Vono, N. Dobigeon and P. Chainais, Split-and-augmented Gibbs sampler - Application to large-scale inference problems, arXiv: 1804.05809, 2018.
6. M. Perera, Proximal Markov Chain Monte Carlo algorithms, Stat. computing, vol. 26, July 2016.
7. A. Wibisono, Sampling as optimization in the space of measures: The LD as a composite optimization problem, COLT 2018.

Posterior Sampling

So far, goal was to sample $x \sim p_*(x)$, the ground-truth dist. We now focus on conditional sampling, i.e., $x \sim p_*(x|y)$, where y is some side information / measurement about the specific 'x' that we are interested in.

Examples include: $y \equiv$ part of an image ;
 $y \equiv$ blurry image ; $y \equiv$ text string for editing the image ; etc. A few image examples are below:

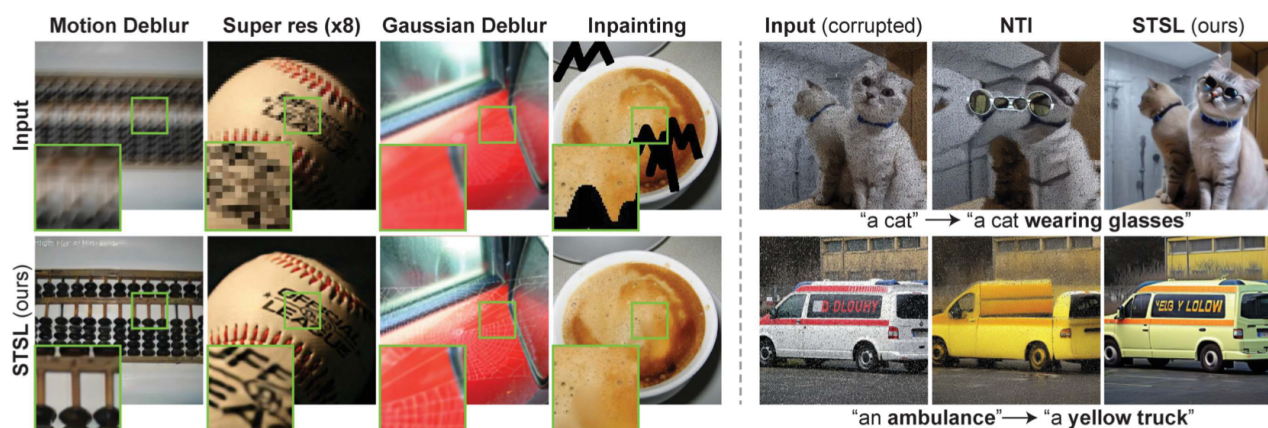


Figure 1. **Image inversion and editing with latent diffusion:** Our method, termed STSL, provides efficient inversion while enhancing quality of reconstructed images, even when addressing corruptions (e.g., blurriness, low resolution, noise). We show its versatility in various inversion tasks (**left figure**): motion deblurring, super-resolution, gaussian deblurring, and inpainting. In addition, STSL extends to text-guided image editing with corrupted images (**right**), surpassing the performance of NTI [34], a prominent method in this domain.

Image credit: L. Raut et al., Beyond First-order Tweedie: Solving inverse problems using latent diffusion, CVPR 2024.

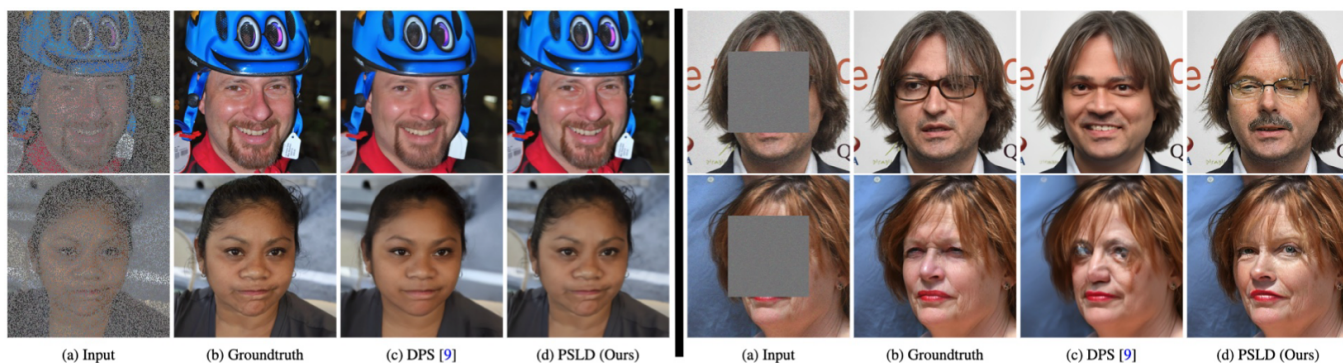


Figure 3: **Left panel:** Random Inpainting on images from FFHQ 256 [25] using PS LD with Stable Diffusion v-1.5. Notice the text in the top row and the facial expression in the bottom row. **Right panel:** Block (128 × 128) inpainting, using the LDM-VQ-4 model trained on FFHQ 256 [25]. Notice the glasses in the top row and eyes in the bottom row.

Image credit: L. Rout et. al., Solving linear inverse problems provably via posterior sampling with latent diffusion models, NeurIPS, 2023.

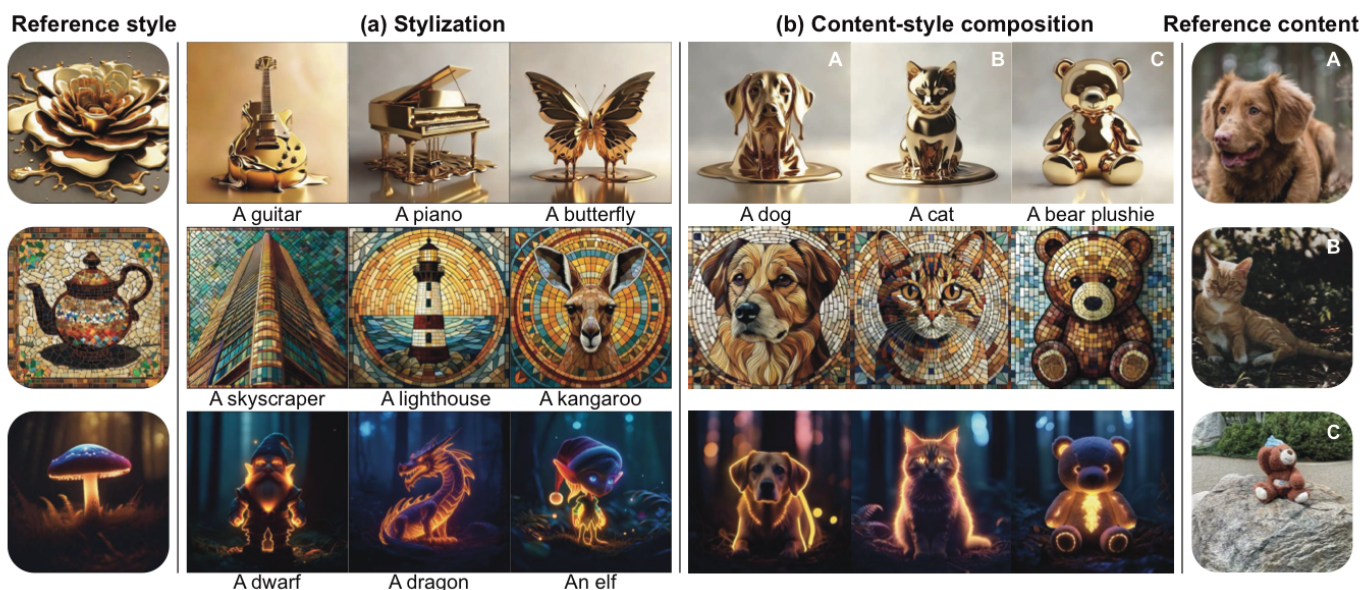


Figure 1: Given a single reference image (rounded rectangle), our method **RB-Modulation** offers a plug-and-play solution for (a) stylization, and (b) content-style composition with various prompts while maintaining sample diversity and prompt alignment. For instance, given a reference style image (e.g. “melting golden 3d rendering style”) and content image (e.g. (A) “dog”), our method adheres to the desired prompts without leaking contents from the reference style image and without being restricted to the pose of the reference content image.

Image credit: L. Rout et. al., RB-Modulation: Training free stylization using reference based modulation, ICLR 2025.

Connection between inverse problems and posterior sampling

(see also [7])

Given measurement y , we could, as a base case, solve an optimization problem to recover ' x ', i.e.,

$$x^* = \arg \min_x L_y(x), \quad \text{why } L_y(\cdot) \text{ is the loss function.}$$

e.g.: Suppose $y = A(x) + n$

Then, $L_y(x) = \|y - A(x)\|^2$

noise
measurement operator (e.g., blurry camera)

If the problem is ill posed (i.e., no unique solution), we would regularize this as:

$$\min_x L_y(x) + g(x)$$

regularizer to bias toward a specific solution

In particular, we could use the prior distribution

$p_*(x)$ to regularize using the following reformulation of the above optimization: Sample $x \sim q_*(x)$, where $q_*(\cdot)$ satisfies:

$$q_* = \arg \min_q E_{x \sim q} [L_y(x)] + KL(q \| p_*)$$

↙
Sample from $q(\cdot)$
minimizes expected loss
↘
 $q(\cdot)$ is "close"
to $p_*(\cdot)$

$$= \arg \min_q E_{x \sim q} \left[-\ln e^{-L_y(x)} + \ln \left(\frac{q(x)}{p_*(x)} \right) \right]$$

$$= \arg \min_q E_{x \sim q} \left[\ln \left(\frac{q(x)}{p_*(x) \cdot e^{-L_y(x)}} \right) \right]$$

Let $Z = \int p_*(x) e^{-L_y(x)} dx$ (a constant).

↙

$$= \arg \min_q E_{x \sim q} \left[\ln \left(\frac{q(x)}{\frac{1}{Z} p_*(x) e^{-L_y(x)}} \right) \right] - \ln Z$$

$$= \underset{q}{\operatorname{argmin}} \quad KL \left(q(\cdot) \parallel \frac{1}{Z} P_*(\cdot) e^{-L_y(\cdot)} \right) - \ln Z.$$

This is minimized when $q_*(x) = \frac{1}{Z} P_*(x) e^{-L_y(x)}$.

In words, q_* is the posterior distribution that is given by the measurement loss-tilted ground-truth prior distribution.

Note 1: We could instead consider context 'y' and an associated reward $R_y(x)$. In this case, the above discussion can be restated as:

$$\max_q \quad E_{x \sim q} [R_y(x) - \alpha KL(q \parallel P_*)].$$

where P_* is the nominal base distribution. Then, identical to above, we have

$$q_*(x) \propto e^{\frac{1}{\alpha} R_y(x)} P_*(x)$$

As an aside, observe that the solution to RL PPO (Proximal Policy Optimization) closely resembles the exponentially tilted distribution above. Thus, posterior sampling can be used as a possible approach for addressing RL problems, which we will discuss in later classes.

In summary, we can view inverse problems or KL regularized reward optimization as posterior sampling problems. In this note and the subsequent classes, we discuss a few approaches to sampling from the posterior:

- a. Training-based "direct" approaches, where we train / fine tune a model for $q_{\star}(\cdot)$.
- b. Training-free approaches (aka test-time optimization) where we have access only to $p_{\star}(\cdot)$, and $L_y(\cdot)$; the goal is to sample from the tilted posterior (Importance sampling, Split Gibbs, approximations to posterior).

Direct approach: Given many examples of $z = (x, y)$, train a new ground-truth model $q_{x,y}(z) = P_*(x|y)$. Thus, we reduce the posterior sampling problem to the standard generation problem discussed so far.

This approach can be refined as follows: As in the diffusion case, we can learn conditional annealed scores:

$$S_t(x_t|y) = \nabla_{x_t} \ln P_t(x_t|y)$$

$$= \nabla_{x_t} \ln \left(\frac{P_t(y|x_t) \cdot P_t(x_t)}{P_t(y)} \right)$$

$$= \underbrace{\nabla \ln P_t(y|x_t)}_{\text{measurement based guidance}} + \underbrace{\nabla \ln P_t(x_t)}_{\text{the "standard" annealed score}} - \underbrace{\nabla \ln P_t(y)}_{= 0}$$

The vector field in the diffusion decomposes into the sum of two terms: the "standard" pre-trained annealed

score, along with a measurement based guidance term. Intuitively, the unconditional score points toward the manifold of images, whereas the guidance term points toward vectors that are consistent with the measurement y .

We can now train a smaller network using data of (x, y) pairs to learn only the guidance term, or finetune the score network using this data to directly incorporate the guidance term. For text conditioning, we can directly train the base model using text dropout [9]; see references below.

8. P. Dhariwal and A. Nichol, Diffusion models beat GANs on image synthesis, arXiv: 2105.05233, 2021.
9. J. Ho and T. Salimans, Classifier-free diffusion guidance, arXiv: 2207.12598, 2022.
10. Sander Dieleman, The geometry of diffusion guidance, (blog post), August 2023. sander.ai/2023/08/28/geometry.html.

Training-free approaches

These approaches assume access to only the unconditional sampler (i.e., $p_*(x)$, or some variant such as $\nabla \ln p_t(x)$), and the goal is to generate conditional samples during test time.

① Rejection Sampling

Setting: We have access to a black box that returns samples $x_i \sim p_*(x)$. Goal is to use this sample access, to instead sample from $q_*(x) \propto e^{R_y(x)} p_*(x)$ (the tilted distribution).

Suppose that the reward $R_y(x) \in [0, r_{\max}]$.

Then, observe that $q_*(x) = \frac{e^{R_y(x)}}{Z} p_*(x)$,

where $Z = \int e^{R_Y(x)} p_X(x) dx \geq 1.$
($\because R_Y(x) \geq 0$).

$$\therefore q_X(x) \leq e^{R_Y(x)} p_X(x) \leq e^{r_{\max}} \cdot p_X(x).$$

Let $M = e^{r_{\max}}.$

We are now in the setting of rejection sampling (see Note 1).

Algorithm:

- (a) Sample $X \sim p_X(x).$
- (b) Sample $U \sim \text{Uniform}[0, 1]$ indep.
↓ ($X \perp U$)

(c) Accept x if $U \leq \frac{q_X(x)}{M p_X(x)} = e^{(R_Y(x) - r_{\max})}.$

else reject the sample x .

Denote $A = \mathcal{X} \left\{ U \leq \frac{q_X(x)}{M p_X(x)} \right\}$ (acceptance
r.v.)

i.e., $A \sim \text{Bernoulli}(e^{R_y(x) - r_{\max}})$

Then, $q_*(x) = P_*(x | A=1)$

i.e., sample repeatedly: $x_i \sim P_*(\cdot)$, compute $e^{R_y(x_i) - r_{\max}}$, and generate $a_i \sim \text{Bernoulli}(e^{R_y(x_i) - r_{\max}})$.
If $a_i = 1$, then return x_i .

⑥ Split Gibbs Sampling:

Background: Recall Gibbs sampling (Note 1) to sample from $q(x, z)$. Suppose that we have some mechanism that allows us to generate samples from $q(x|z)$ and $q(z|x)$.

The prev. motivation in Note 1 was that we can sample from scalars, and needed to generalize to vectors (x, z) . Then, the Gibbs sampler specialized to this case leads to the following algorithm:

Initialize with x_0 .

Loop: $i = 0, 1, 2, \dots$

Generate $z_i \sim q(z|x_{i-1})$

$x_i \sim q(x|z_i)$

Then, $(x_i, z_i) \rightarrow (x, z) \sim q(x, z)$

Note that the above procedure directly also applies if $x \in \mathbb{R}^d$, $z \in \mathbb{R}^p$, and appropriate samplers $q(x|z)$ and $q(z|x)$ are available.

Split Gibbs: Sample from: $p_*(x|y) \propto p_*(x) e^{-L_y(x)}$.

or more generally, $p_*(x|y) \propto p_*(x) \cdot p_*(y|x)$

Here, $p_*(y|x) = \frac{e^{-L_y(x)}}{Z'(x)}$, for

$\downarrow$
ground truth cond.
measurement dist.

$Z'(x) = \int e^{-L_y(x)} dy$. Below, we will stick to $e^{-L_y(x)}$ for concreteness.

First, introduce a new variable $z = x$, and rewrite as:

$$(x, z) \sim p_*(x, z | y) = \frac{1}{Z_y} p_*(x) e^{-L_y(x)} \cdot \delta_x(z)$$

Sample a pair (x, y)

dirac delta at $z = x$

$$= \frac{1}{Z_y} p_*(x) e^{-L_y(z)} \delta_x(z)$$

Now, relax the hard constraint $z = x$:

$$\delta_x(z) \rightarrow e^{-\frac{1}{\rho} \|x - z\|^2}, \text{ for } \rho > 0.$$

Note that as $\rho \rightarrow 0$, $e^{-\frac{1}{\rho} \|x - z\|^2} \approx \delta_x(z)$.

$\therefore$ Relaxed sampling problem is:

$$(x, z) \sim q_\rho(x, z | y)$$
$$= \frac{1}{Z_y} p_*(x) e^{-L_y(z)} \cdot e^{-\frac{1}{\rho} \|x - z\|^2}$$

$$\text{Let } q_{\rho, y}(x|z) = \frac{1}{Z_1} p_*(x) e^{-\frac{1}{\rho} \|x-z\|^2} \rightarrow \textcircled{1}$$

$$q_{\rho, y}(z|x) = \frac{1}{Z_2} e^{-L_y(z)} \cdot e^{-\frac{1}{\rho} \|z-x\|^2} \rightarrow \textcircled{2}$$

Suppose we can sample from each of the above two conditional dist. Then, we can use Gibbs sampling to return a sample (x, z) that is consistent with $q_{\rho}(x, z|y)$. Further, by letting $\rho \rightarrow 0$, we can get close to the true posterior (see [2, 3, 5, 6]). (Below corresponds to [3])

Sampling from $\textcircled{1}$: Observe that $\textcircled{1}$ corresponds to the posterior when: $z = y = x^* + n$ (noisy observation). But this is also the same as the sample from the forward process at a specific noise level (depends on ρ in $\textcircled{1}$). This connection is used in [3] to show that we can use the standard (reverse) DDPM/DDIM initialized at a specific time, and $IC = z$ to get a sample from $\textcircled{1}$.

Sampling from ②: $q_{p,y}(z|x) \propto e^{-\left(L_y(z) + \frac{1}{p} \|z-x\|^2\right)}$

Given (x, p, z) , we can sample from this using LMC or some variant (e.g. MALA $\rightarrow$ Metropolis Adjusted Langevin Algorithm, which uses LMC to propose the next step, and accepts/rejects the next step using MH Algorithm).

The Split-Gibbs sampler will asymptotically converge if p is scheduled appropriately, but there are limited guarantees on the number of steps (in general, would be exponential in dimension d)

③ Importance Sampling / Test time Monte Carlo

Recall that for DDPM, the key function that is needed is the annealed score:

$$S_t(x_t|y) = \nabla_{x_t} \ln p_t(x_t|y)$$

We now show that $S_t(x_t|y)$ can be represented as :

$$P(y|x_0) \propto e^{-L_y(x_0)}$$

$$\nabla_{x_t} \ln p_t(x_t|y) = \frac{E_{x_0|x_t=x_t} \left[P(y|x_0) \left(\frac{x_0 - x_t}{\sigma_t^2} \right) \right]}{E_{x_0|x_t=x_t} [P(y|x_0)]}$$

$$\frac{\int P(y|x_0) \left(\frac{x_0 - x_t}{\sigma_t^2} \right) p_t(x_0|x_t) dx_0}{\int P(y|x_0) p_t(x_0|x_t) dx_0} \rightarrow \textcircled{3}$$

(Thanks to Advait Parulekar for deriving this relation.)

We will prove this soon, but first remark on the implication of this identity.

Suppose that during test time, we are evolving the reverse process such that the current state is x_t , and the measurement is y .

Then in principle, we can use the standard unconditional DDPM (or DDIM) sample to get

$$\{x_0^{(1)}, x_0^{(2)}, \dots, x_0^{(n)}\} \sim P_t(x_0|x_t)$$

(repeat for i times: run DDPM starting from initial time t and initial state x_t , and evolve to $t=0$, $x_0^{(i)}$; $i=1, 2, \dots, n$).

We can use these samples and approximate:

$$\int P(y|x_0) \left(\frac{x_0 - x_t}{\sigma_t^2} \right) P_t(x_0|x_t) dx_0$$

$$\approx \frac{1}{n} \sum_{i=1}^n P(y|x_0^{(i)}) \left(\frac{x_0^{(i)} - x_t}{\sigma_t^2} \right)$$

with concentration results quantifying the (vanishing) approximation error as a function of n , the number of samples. (Similarly for denominator of ③)

Lemma 1

$$\nabla_{x_t} \ln p_t(x_t|y) =$$

$$\frac{\int p(y|x_0) \left(\frac{x_0 - x_t}{\sigma_t^2} \right) p_t(x_0|x_t) dx_0}{\int p(y|x_0) p_t(x_0|x_t) dx_0}$$

Proof: We have:

$$\nabla_{x_t} \int p(y|x_0) p_t(x_0|x_t) dx_0$$

$$= \int p(y|x_0) \nabla_{x_t} p_t(x_0|x_t) dx_0$$

$$= \int p(y|x_0) \frac{\nabla_{x_t} p_t(x_0|x_t)}{p_t(x_0|x_t)} p_t(x_0|x_t) dx_0$$

$$= \int p(y|x_0) \nabla_{x_t} (\ln p_t(x_0|x_t)) p_t(x_0|x_t) dx_0$$

$$= \int P(y|x_0) \nabla_{x_t} \left(\ln \frac{P(x_t|x_0)}{P_t(x_t)} \right) P_t(x_0|x_t) dx_0.$$

$$P_t(x_0|x_t) = \frac{P(x_t|x_0) P_*(x_0)}{P_t(x_t)} \quad (\text{Bayes rule}).$$

where $p(x_t|x_0) \propto e^{-\frac{1}{2\sigma_t^2} \|x_t - x_0\|^2}$
because of the forward Gaussian noising process

Also,

$$\nabla_{x_t} \left(\ln \frac{P(x_t|x_0)}{P_t(x_t)} \right) = -\frac{(x_t - x_0)}{\sigma_t^2} - \nabla_{x_t} \ln P_t(x_t)$$

$$\Rightarrow = \int P(y|x_0) \left(-\frac{(x_t - x_0)}{\sigma_t^2} - \nabla_{x_t} \ln P_t(x_t) \right) P_t(x_0|x_t) dx_0$$

i.e., $\nabla_{x_t} \int P(y|x_0) P_t(x_0|x_t) dx_0$

$$= \int P(y|x_0) \left(-\frac{(x_t - x_0)}{\sigma_t^2} - \nabla_{x_t} \ln P_t(x_t) \right) P_t(x_0|x_t) dx_0$$

→ (4)

Next, we have:

$$\nabla_{x_t} \ln p_t(y|x_t) = \nabla_{x_t} \ln \int p(y|x_0, x_t) p_t(x_0|x_t) dx_0$$

$$= \nabla_{x_t} \ln \left(\int p(y|x_0) p_t(x_0|x_t) dx_0 \right)$$

$$= \frac{\nabla_{x_t} \left(\int p(y|x_0) p_t(x_0|x_t) dx_0 \right)}{\int p(y|x_0) p_t(x_0|x_t) dx_0}$$

$$\stackrel{(4)}{=} \frac{\int p(y|x_0) \left(-\frac{(x_t - x_0)}{\sigma_t^2} - \nabla_{x_t} \ln p_t(x_t) \right) p_t(x_0|x_t) dx_0}{\int p(y|x_0) p_t(x_0|x_t) dx_0}$$

no dependence
on x_0

$$= \frac{\int p(y|x_0) \left(-\frac{(x_t - x_0)}{\sigma_t^2} \right) p_t(x_0|x_t) dx_0}{\int p(y|x_0) p_t(x_0|x_t) dx_0} - \nabla_{x_t} \ln p_t(x_t)$$

$$\therefore \nabla_{x_t} p_t(y|x_t) + \nabla_{x_t} \ln p_t(x_t)$$

$$= \frac{\int p(y|x_0) \left(-\frac{(x_t - x_0)}{\sigma_t^2} \right) p_t(x_0|x_t) dx_0}{\int p(y|x_0) p_t(x_0|x_t) dx_0} \rightarrow \textcircled{5}$$

Also, we have from Bayes' rule:

$$\begin{aligned} \nabla_{x_t} \ln p_t(x_t|y) &= \nabla_{x_t} \ln p_t(y|x_t) + \nabla_{x_t} \ln p_t(x_t) \\ &\quad - \cancel{\nabla_{x_t} \ln p(y)} \\ &= 0 \end{aligned}$$

$\therefore$ From above and $\textcircled{5}$, the result follows $\square$

While these methods above, in principle, resolve the

test-time posterior sampling problem, they are computationally challenging. Specifically, they require many calls to the denoising score network, which can be expensive. (e.g., with importance sampling as above, we need $(2nt)$ calls for estimating ③; and if the overall diffusion process is over T steps, then the total number of calls $\approx nT^2$).

Hence in practice, approximations are used, which we will explore in subsequent notes. Here, we only discuss a popular approximation called DPS $\rightarrow$ Diffusion Posterior Sampling [10].

① Diffusion Posterior Sampling [11]

Recall that: $\nabla_{x_t} \ln p_t(x_t|y) = \underbrace{\nabla_{x_t} \ln p_t(y|x_t)}_{\text{need to estimate at test-time}} + \underbrace{\nabla_{x_t} \ln p_t(x_t)}_{\text{known}}$

Now,

$$\nabla_{x_t} \ln p_t(y|x_t) = \nabla_{x_t} \ln \int p_t(y|x_0, x_t) p_t(x_0|x_t) dx_0$$

$$= \nabla_{x_t} \ln \int p(y|x_0) p_t(x_0|x_t) dx_0$$

$$= \nabla_{x_t} \ln E_{x_0|x_t=x_t} [p(y|x_0)]$$

$$\text{Let } g(x_0) = p(y|x_0)$$

$$\text{Then } g(x_0) = g(\bar{x}) + \langle \nabla g(\bar{x}), (x_0 - \bar{x}) \rangle$$

$$+ \text{H.O.T}$$

DPS Approximation: $g(x_0) \approx g(\bar{x}) + \langle \nabla g(\bar{x}), (x_0 - \bar{x}) \rangle$
(first-order approx)

$$\text{where } \bar{x} = E_{x_0|x_t=x_t} [x_0|x_t=x_t]$$

$$\text{i.e., } \bar{x} = \text{MMSE}(x_t; \sigma_t)$$

In words, $\bar{x}$ is the minimum mean-square estimate of x_0 given x_t . Recall from Notes 9, 9b, that:

$$\text{MMSE}(x_0; \sigma) = x_0 + \sigma^2 S_\sigma(x_0).$$

Tweedie's Formula.

unconditional score is known

Thus, using Tweedie's formula (in the "reverse" sense from our previous use), the $\text{MMSE}(x_t; \sigma_t)$ can be computed using the available unconditional score.

$$\nabla_{x_t} \ln p_t(y|x_t) = \nabla_{x_t} \ln E_{x_0|x_t=x_t} [p_t(y|x_0)]$$

$$\approx \nabla_{x_t} \ln \left(E_{x_0|x_t=x_t} [g(\bar{x}) + \langle \nabla g(\bar{x}), (x_0 - \bar{x}) \rangle] \right)$$

$$= \nabla_{x_t} \ln \left(g(\bar{x}) + \langle \nabla g(\bar{x}), E_{x_0|x_t=x_t} [x_0 - \bar{x}] \rangle \right)$$

= 0

$$= \nabla_{x_t} \ln p(y) E_{x_0|x_t=x_t} [x_0]$$

→ Computed using uncond. score

Note that this is computationally much cheaper, as we require only one additional call to the denoising network in each denoising step of the diffusion.

Refs:

11. H. Chung, J. Kim, M. McCann, M. Klasky and J. Ye, Diffusion posterior sampling for general noisy inverse problems, arXiv: 2209.14687, 2022.