Diffusion Inverse Problem

DMQA Open Seminar (26. 04. 10)

Data Mining & Quality Analytics Lab.

송하영

발표자 소개

❖ **송하영 (Hayoung Song)**

- 고려대학교 산업경영공학과 석사과정 (2026.03 ~ Present)
- Data Mining & Quality Analytics Labs. (김성범 교수님)

❖ **Research Interest**

- Generative Models
- Diffusion Models

❖ **Contact**

- thdgk1245@korea.ac.kr

Introduction

Diffusion Inverse Problem

❖ **Diffusion models as powerful image generators**

Introduction

Diffusion Inverse Problem

- ❖ **Diffusion models as powerful image generators**

Introduction

Diffusion Inverse Problem

❖ Diffusion models as powerful image generators

Diffusion의 강력한 생성능력을 활용해서 실생활 문제에 어떻게 접근할 수 있을까?

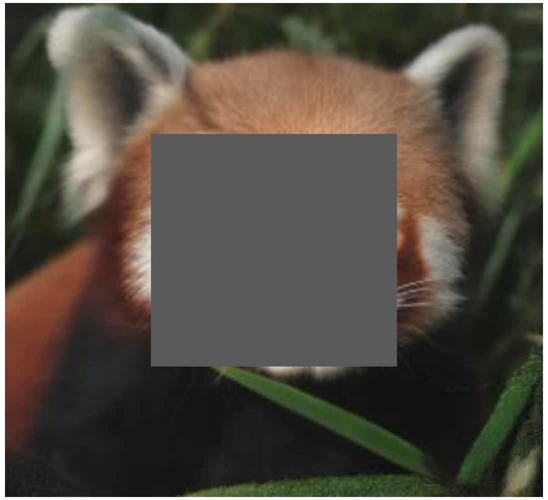
Introduction

Diffusion Inverse Problem

❖ **Inverse problems (Inpainting)**

Original Image

Missing pixels



Q. 어떻게 빈 픽셀을 채울 수 있을까?

→ 같은 빈 공간을 설명할 수 있는 값들은 무수히 존재한다 (ill-posed problem)

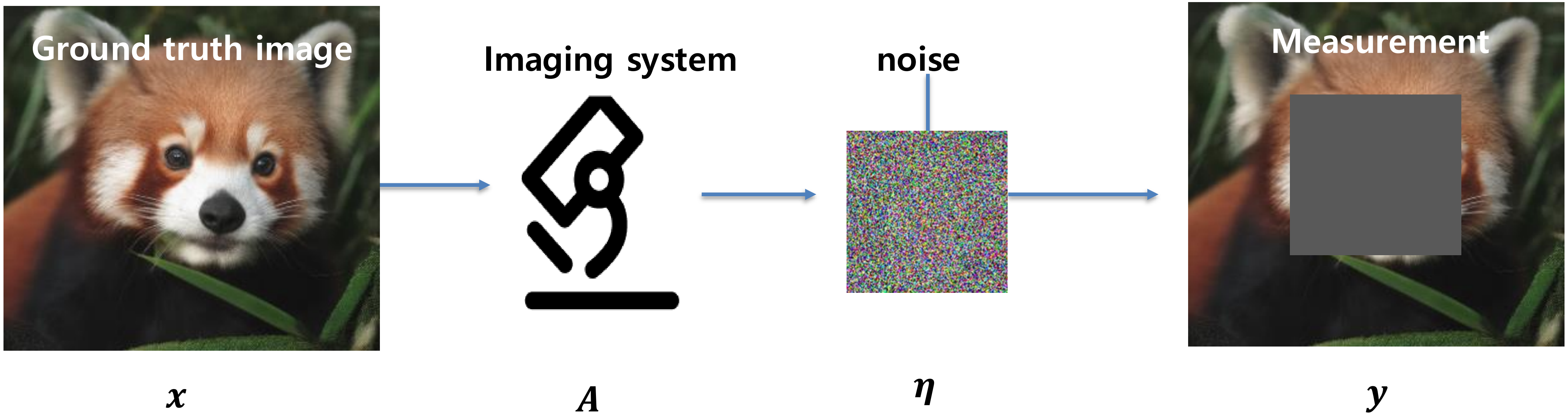
→ 그 중, 우리의 사전 지식을 이용해서 해당 빈 값을 적절히 채워야 한다.

따라서, **적절한 사전 지식과 관측된 정보**를 잘 활용해야 한다.

Introduction

Diffusion Inverse Problem

❖ Inverse problems



- Problem: noise가 섞인 관측값 y로부터 원본 이미지 x를 복원하는 문제 (y만 주어져 있을 때 x를 복원하는 문제)
- 같은 y를 설명하는 x가 무한히 존재

  → 따라서 그럴듯한 이미지란 무엇인가에 대한 사전 지식(prior knowledge)이 필요

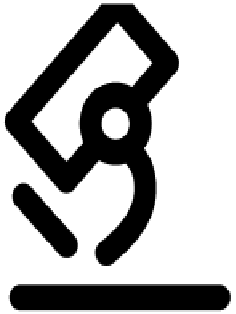
Introduction

Diffusion Inverse Problem

❖ Inverse problems

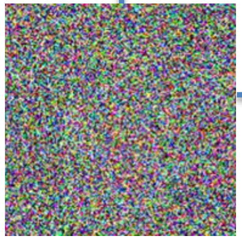
x

Imaging system

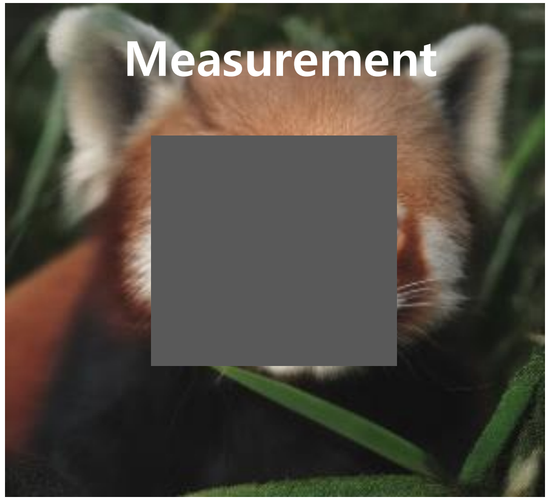


A

noise



η



y

- Problem: noise가 섞인 관측값 y로부터 원본 이미지 x를 복원하는 문제
- 같은 y를 설명하는 x가 무한히 존재

→ 따라서 그럴듯한 이미지란 무엇인가에 대한 사전 지식(prior knowledge)이 필요

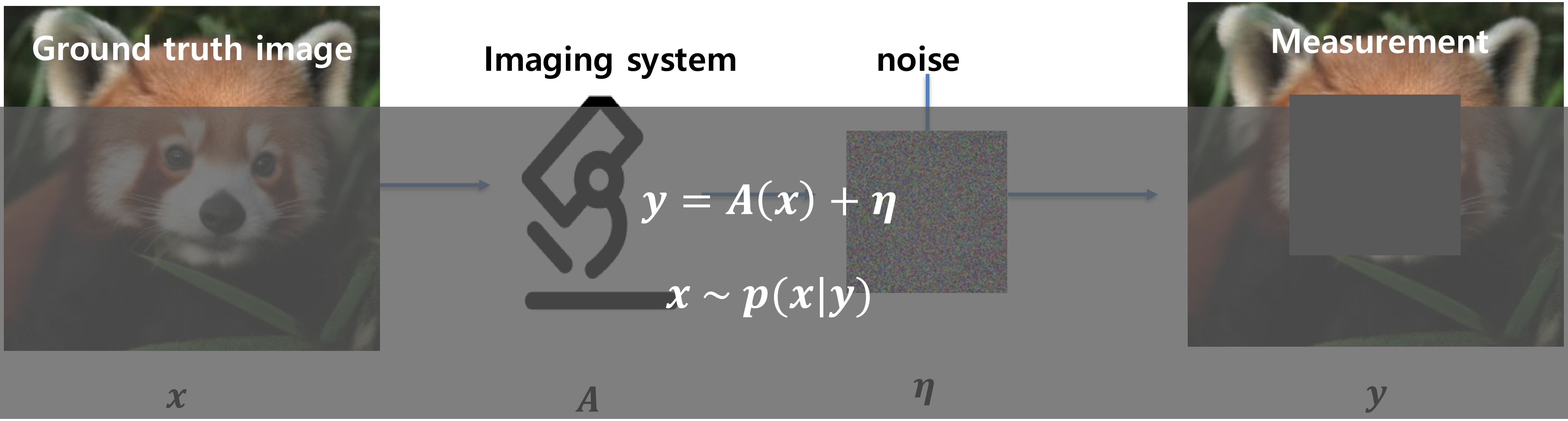
Diffusion 사전 학습 모델 사용

(e.g., Stable Diffusion)

Introduction

Diffusion Inverse Problem

❖ Inverse problems



$$y = A(x) + \eta$$

$$x \sim p(x|y)$$

- Problem: noise가 섞인 관측값 y로부터 원본 이미지 x를 복원하는 문제
- 같은 y를 설명하는 x가 무한히 존재

  → 따라서 그럴듯한 이미지란 무엇인가에 대한 **사전 지식(prior knowledge)**이 필요

Diffusion 사전 학습 모델 사용

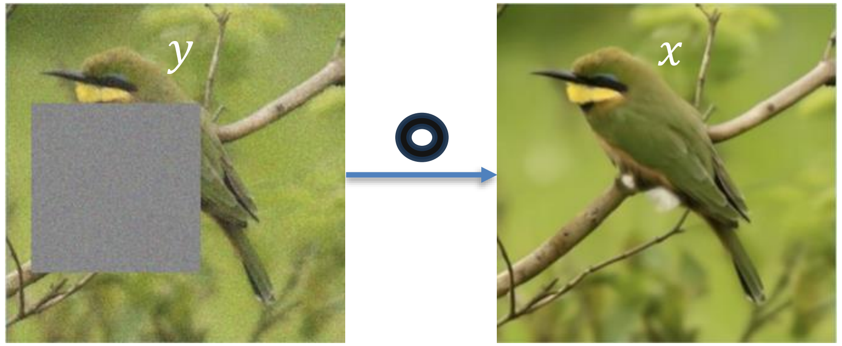
(e.g., Stable Diffusion)

Introduction

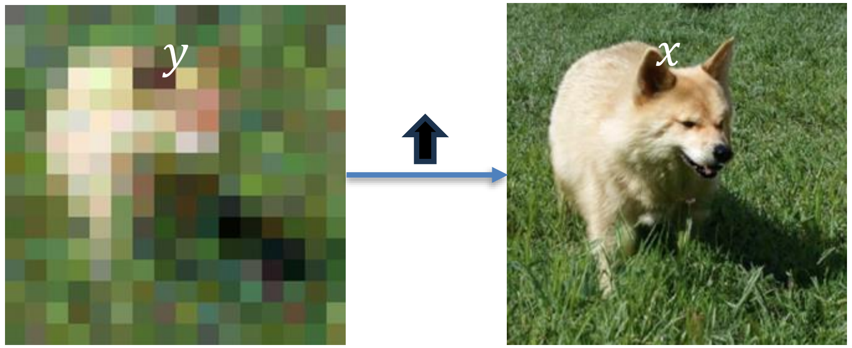
Diffusion Inverse Problem

❖ **Inverse problems: examples**

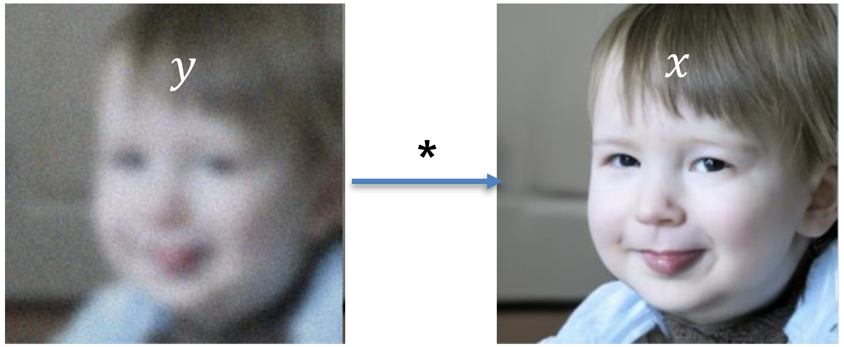
Inpainting



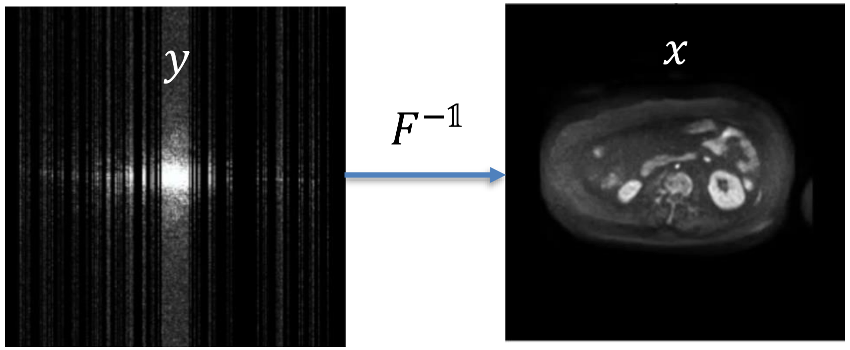
Super-resolution



Deblur



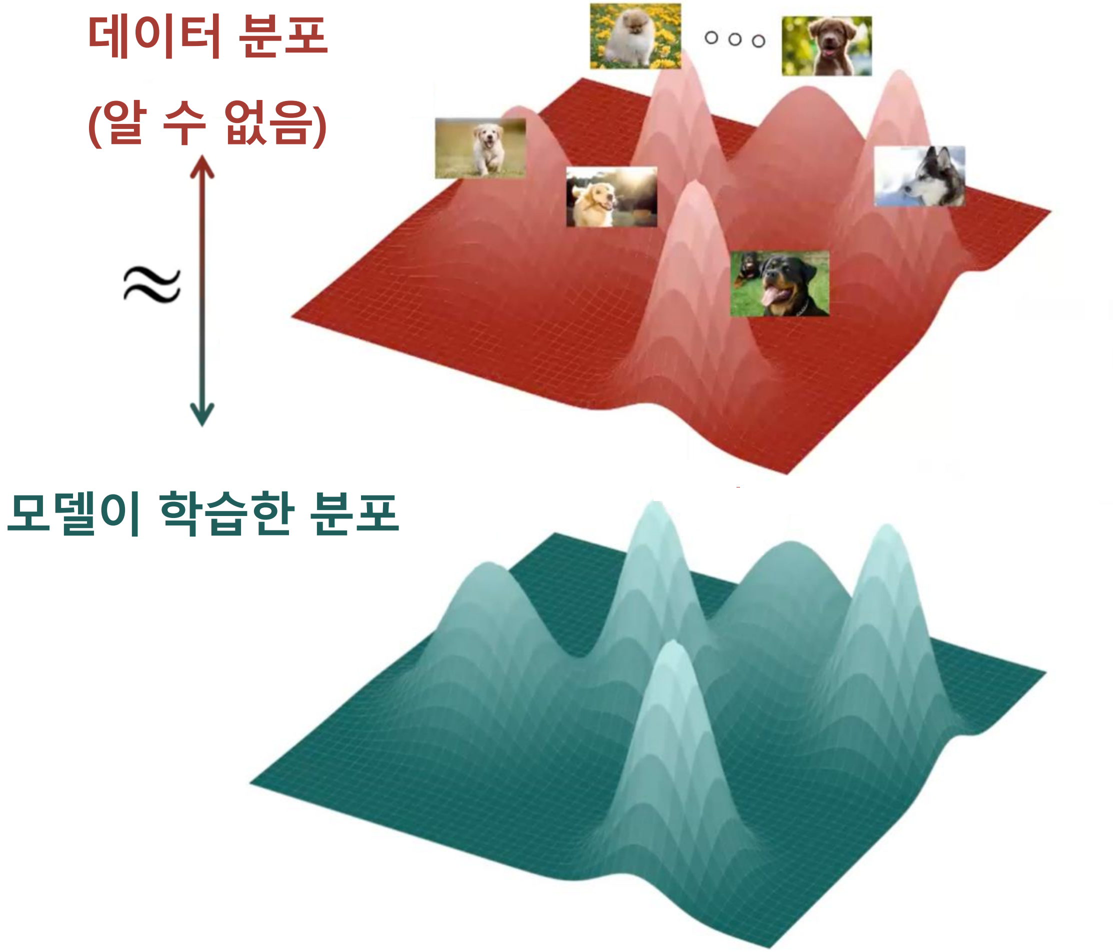
MRI



Introduction

Diffusion Inverse Problem

❖ Deep generative models can sample



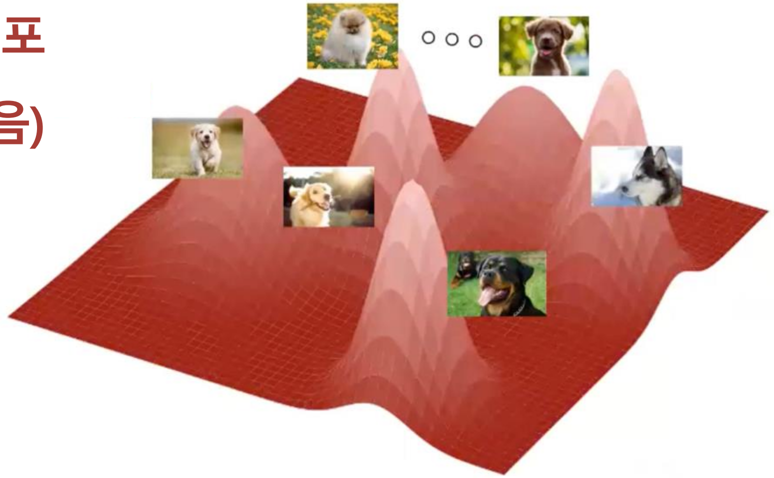
Introduction

Diffusion Inverse Problem

❖ Deep generative models can **sample**

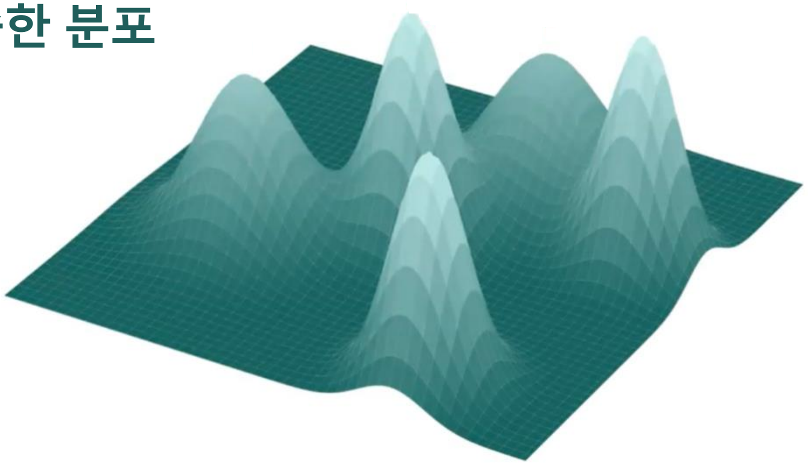
데이터 분포

(알 수 없음)

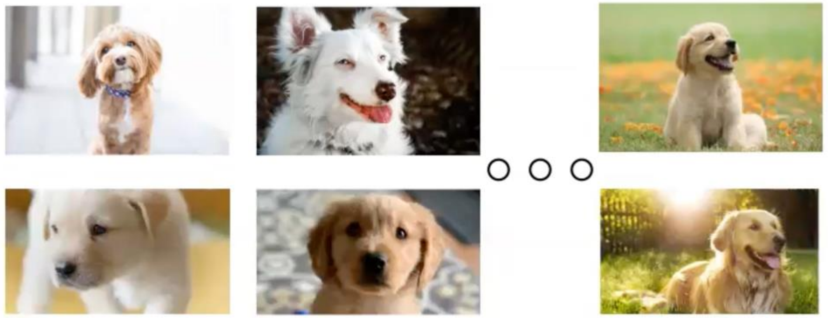


≈

모델이 학습한 분포



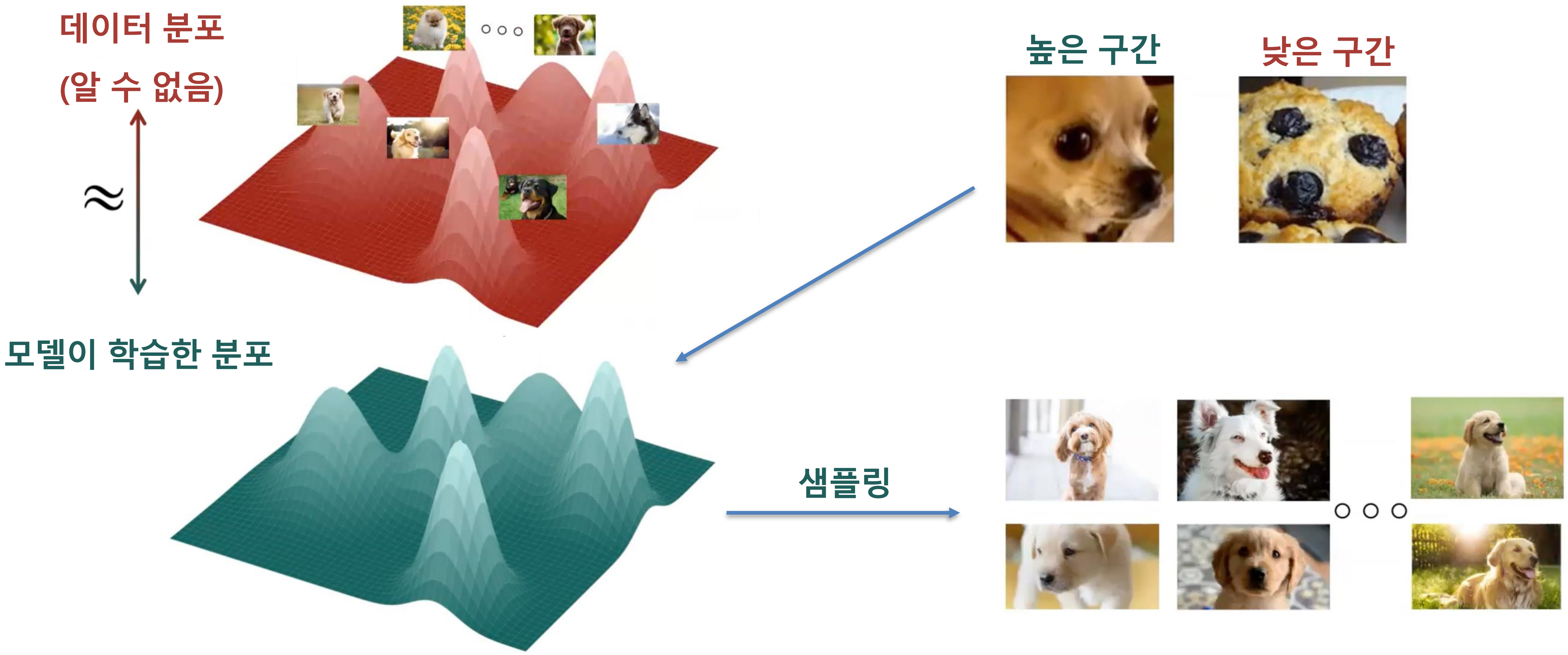
샘플링 →



Introduction

Diffusion Inverse Problem

❖ Deep generative models can **sample**



❖ Toy Example : Diffusion model and Stein score

정상에 가기 위해서는 산 전체를 알아야 함

→ **데이터 분포 전체**를 알아야 한다!

$$p_\theta(x) = \frac{exp\big(f_\theta(x)\big)}{\int exp\big(f_\theta(x)\big)dx}$$

Introduction

Diffusion Inverse Problem

❖ Toy Example : Diffusion model and Stein score

정상에 가기 위해서는 산 전체를 알아야 함

→ **데이터 분포 전체**를 알아야 한다!

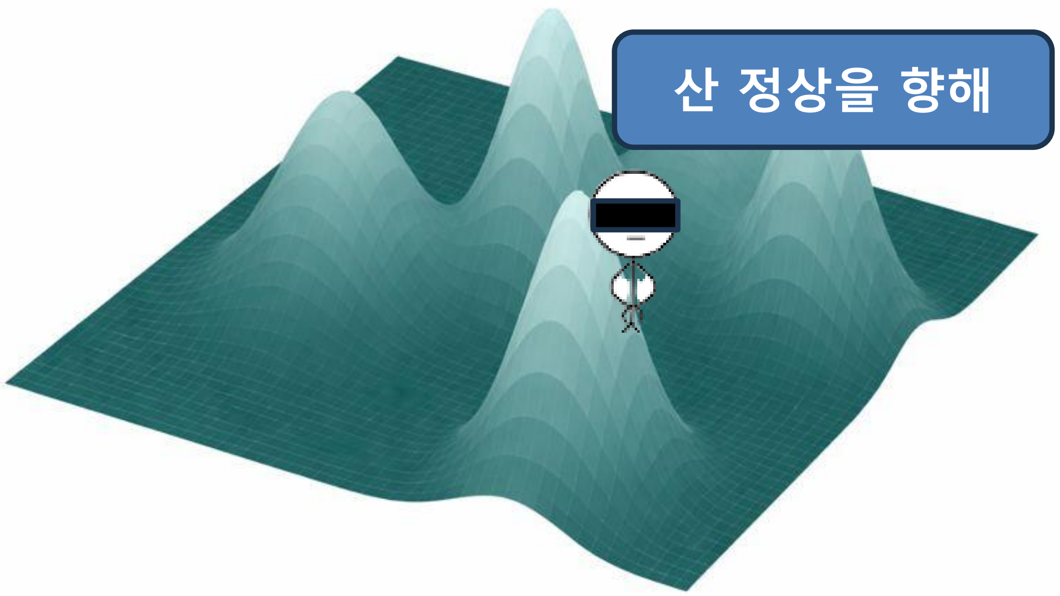
$$p_\theta(x) = \frac{exp\left(f_\theta(x)\right)}{\int exp\left(f_\theta(x)\right)dx}$$

→ **계산 불가능 !**

Introduction

Diffusion Inverse Problem

❖ Toy Example : Diffusion model and Stein score



더 높은 곳으로 가기 위해서는 자기 위치에서

기울기가 가장 높은 곳으로 향해야 함

Introduction

Diffusion Inverse Problem

❖ **Toy Example : Diffusion model and Stein score**

더 높은 곳으로 가기 위해서는 자기 위치에서

기울기가 가장 높은 곳으로 향해야 함

$$s_\theta(x) \simeq \nabla_x logp(x)$$

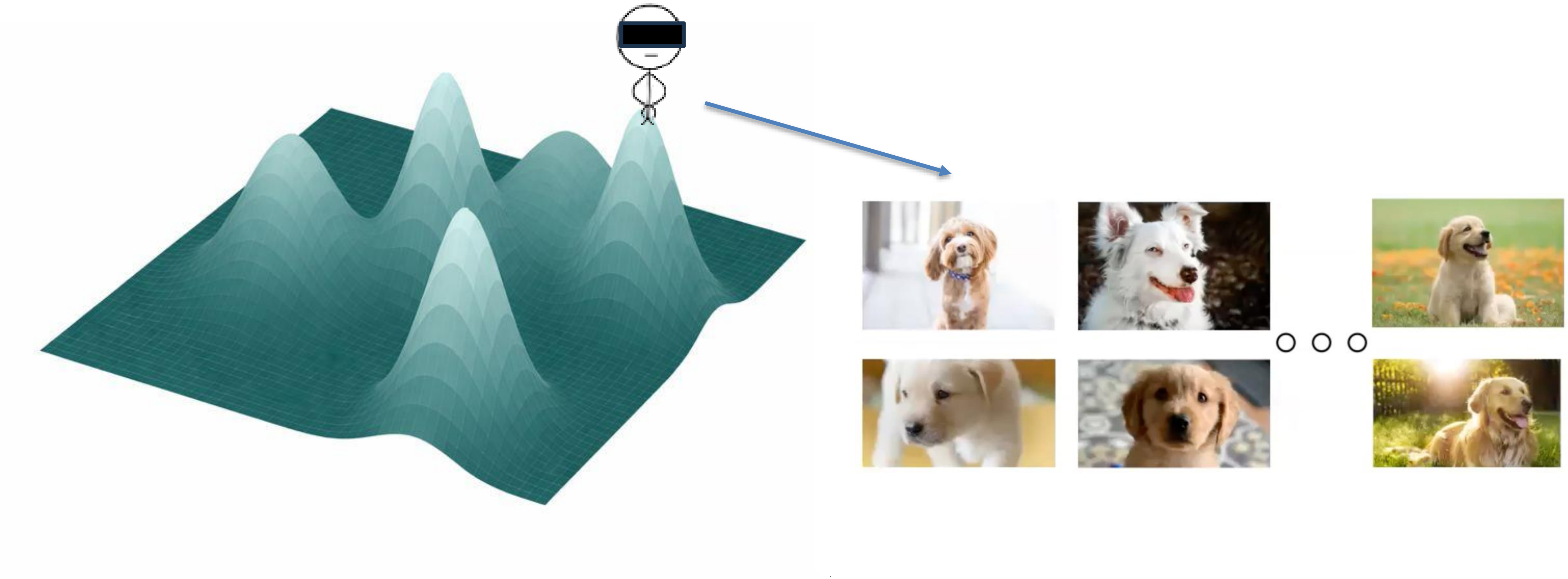
→ score

$$E\left[||\nabla_x logp(x) - \nabla_x logp_\theta(x)||_2^2\right]$$

Introduction

Diffusion Inverse Problem

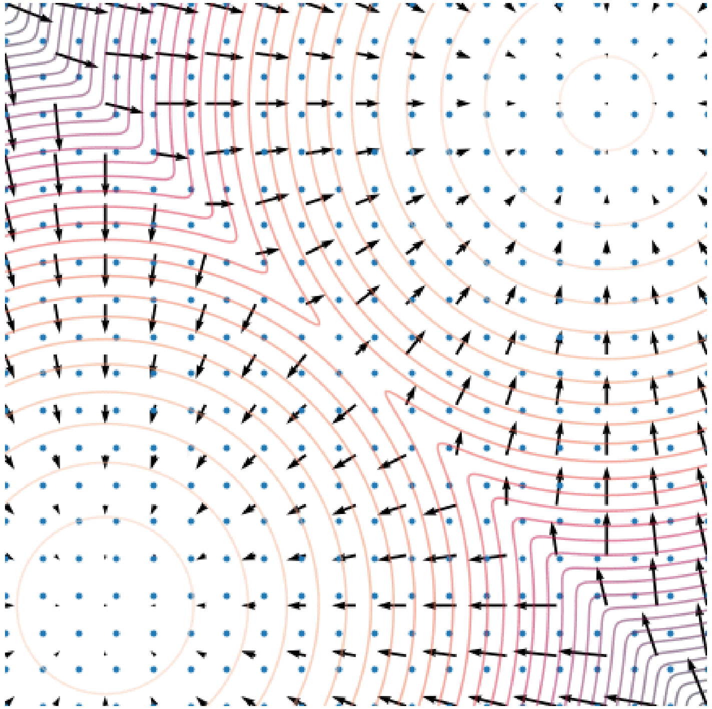
❖ **Toy Example : Diffusion model and Stein score**



Introduction

Diffusion Inverse Problem

❖ Diffusion model and Stein score



- Score 모델이 충분히 학습되면,
- 현재 샘플이 데이터 분포에서 어디로 가야 밀도가 증가하는지 알려줌

$$\text{i.e.}, s_\theta(x) \simeq \nabla_x \log p(x)$$

- Score를 향해 가되, <u>임의의 방향으로 노이즈를 주면서</u>
- 확률밀도가 높은 지역으로 올라감 (<u>Langevin Dynamics</u>)

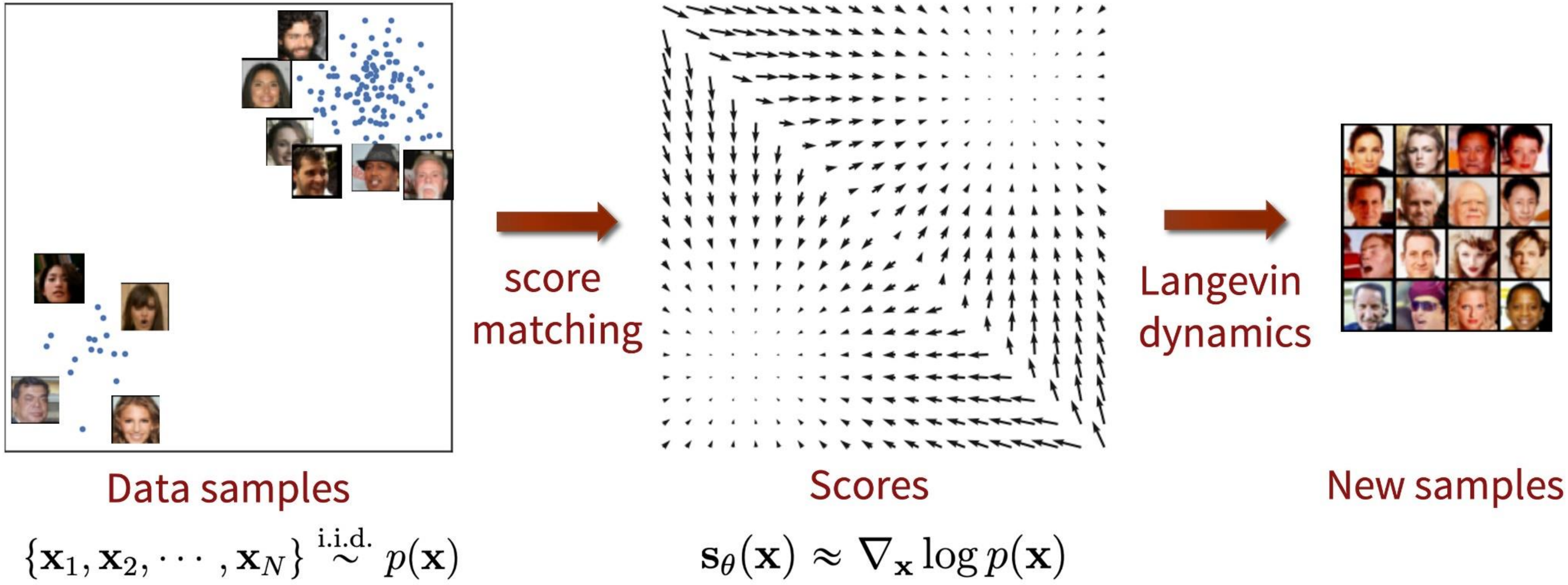
$$X_{i+1} \leftarrow X_i + \epsilon \nabla_x logp(x) + \sqrt{2\epsilon} z_i$$

$$i = 0, 1, \dots, K$$

Introduction

Diffusion Inverse Problem

❖ **Diffusion model and Stein score**



Introduction

Diffusion Inverse Problem

❖ **Diffusion model and Stein score**

Forward Process (data → noise) $0 < t < T$

x(0) → $dx = f(x,t)dt + g(t)dw$ → $x(T)$

score function

$x(T)$ ← $dx = [f(x,t) - g^2(t)\nabla_x \log p_t(x)]dt + g(t)d\bar{w}$ ← x(0)

Reverse Process (noise → data)

Introduction

Diffusion Inverse Problem

❖ **Posterior sampling for inverse problems**

Forward Process (data → noise) $0 < t < T$

$x(0)$ → $dx = f(x,t)dt + g(t)dw$ → $x(T)$

score function

$x(T)$ ← $dx = [f(x,t) - g^2(t)\nabla_x \log p_t(x)]dt + g(t)d\bar{w}$ ← $x(0)$

Reverse Process (noise → data)

$\nabla_x \log p(x|y)$ ← **다시 학습? <u>High cost!</u>**

Introduction

Diffusion Inverse Problem

❖ **Posterior sampling for inverse problems**

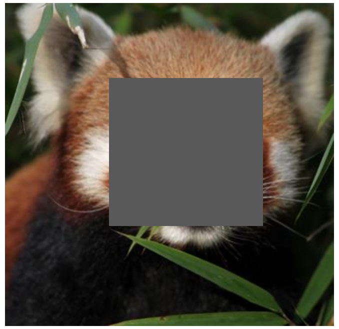
Bayes' Theorem

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)}$$

$$\nabla_x log\ p(x|y) = \nabla_x log\ p(y|x) + \nabla_x log\ p(x)$$

관측과정에 의해 얻어지는 값

우리가 알고 있는 사전지식



Diffusion 사전학습 모델 사용

(e.g., Stable Diffusion)

Introduction

Diffusion Inverse Problem

❖ **Posterior sampling for inverse problems**

Forward Process (data → noise) $0 < t < T$

$x(0)$ → $dx = f(x,t)dt + g(t)dw$ → $x(T)$

score function

$x(T)$ ← $dx = [f(x,t) - g^2(t)\nabla_x log p_t(x)]dt + g(t)d\bar{w}$ ← $x(0)$

Reverse Process (noise → data)

$\nabla_x log\ p(x|y) = \nabla_x log\ p(y|x) + \nabla_x log\ p(x)$ → **재학습 필요 X**

Introduction

Diffusion Inverse Problem

❖ **Posterior sampling for inverse problems**

Forward Process (data → noise)

$0 < t < T$

$x(0)$ $\longrightarrow$ $dx = f(x,t)dt + g(t)dw$ $\longrightarrow$ $x(T)$

$x(T)$ $\longleftarrow$ $dx = [f(x,t) - g^2(t)\nabla_x logp(y|x_t) + \nabla_x logp(x_t)]dt + g(t)d\bar{w}$ $\longleftarrow$ $x(0)$

Reverse Process (noise → data)

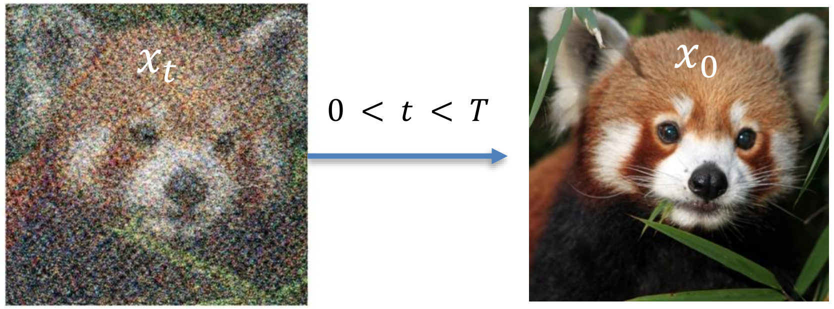
각 timestep별로 score를 학습하기 때문에 위와 같이 쓰임

Introduction

Diffusion Inverse Problem

❖ **Posterior sampling for inverse problems**

$$\nabla_x log p(y|x_t) + \nabla_x log p(x_t)$$



$$p(y|x_t) = \int p(y|x_0, x_t) p(x_0|x_t) dx_0$$

수학적으로 구할 수 없다

Diffusion Posterior Sampling for General Noisy Inverse Problems

ICLR 2023 Spotlight

Diffusion Posterior Sampling (DPS)

Diffusion Inverse Problem

❖ Diffusion Posterior Sampling for General Noisy Inverse Problems

- ICLR 2023 Spotlight, KAIST
- 2026년 4월 10일 기준 1587회 인용

Published as a conference paper at ICLR 2023

DIFFUSION POSTERIOR SAMPLING FOR GENERAL NOISY INVERSE PROBLEMS

Hyungjin Chung[*1,2], **Jeongsol Kim**[*1], **Michael T. Mccann**[2], **Marc L. Klasky**[2] **& Jong Chul Ye**[1]
[1]KAIST, [2] Los Alamos National Laboratory
{hj.chung, jeongsol, jong.ye}@kaist.ac.kr, {mccann, mklasky}@lanl.gov

ABSTRACT

Diffusion models have been recently studied as powerful generative inverse problem solvers, owing to their high quality reconstructions and the ease of combining existing iterative solvers. However, most works focus on solving simple linear inverse problems in noiseless settings, which significantly under-represents the complexity of real-world problems. In this work, we extend diffusion solvers to efficiently handle general noisy (non)linear inverse problems via approximation of the posterior sampling. Interestingly, the resulting posterior sampling scheme is a blended version of diffusion sampling with the manifold constrained gradient without a strict measurement consistency projection step, yielding a more desirable generative path in *noisy* settings compared to the previous studies. Our method demonstrates that diffusion models can incorporate various measurement noise statistics such as Gaussian and Poisson, and also efficiently handle noisy *nonlinear* inverse problems such as Fourier phase retrieval and non-uniform deblurring. Code is available at https://github.com/DPS2022/diffusion-posterior-sampling.

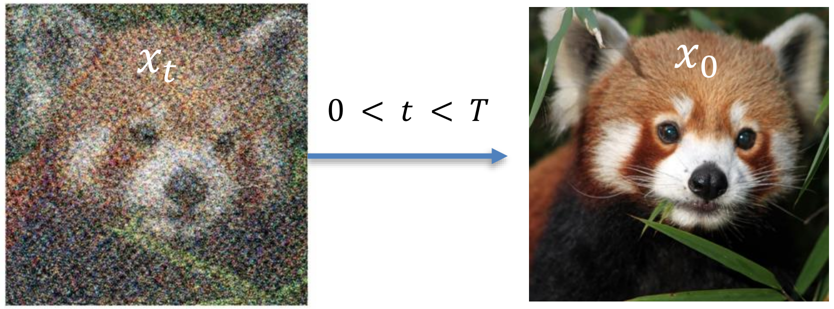
Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., & Ye, J. C. (2023). Diffusion Posterior Sampling for General Noisy Inverse Problems. In *The Eleventh International Conference on Learning Representations, ICLR 2023*. The International Conference on Learning Representations.

Diffusion Posterior Sampling (DPS)

Diffusion Inverse Problem

❖ **Posterior sampling for inverse problems**

$$\nabla_x log p(y|x_t) + \nabla_x log p(x_t)$$



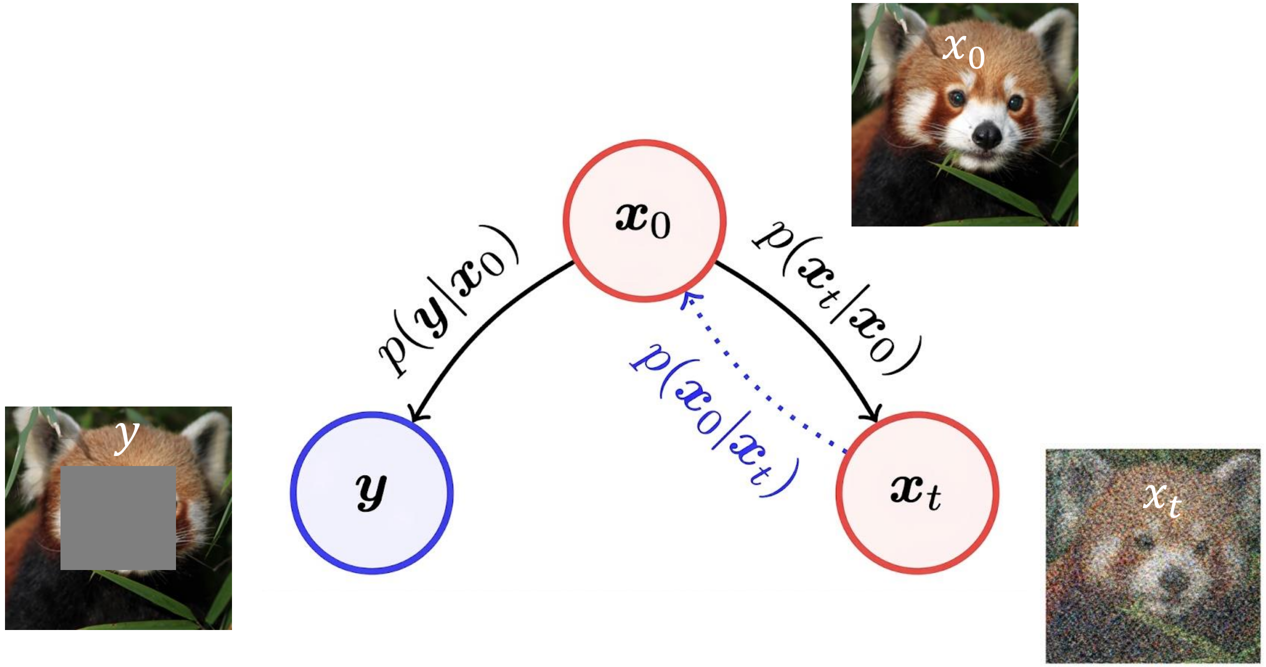
$$p(y|x_t) = \int p(y|x_0, x_t) p(x_0|x_t) dx_0$$

수학적으로 구할 수 없다

Diffusion Posterior Sampling (DPS)

Diffusion Inverse Problem

❖ **Posterior sampling for inverse problems**



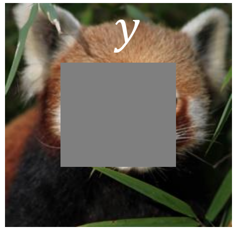
$$p(y|x_0) = N(y|Ax_0, \sigma^2 I)$$

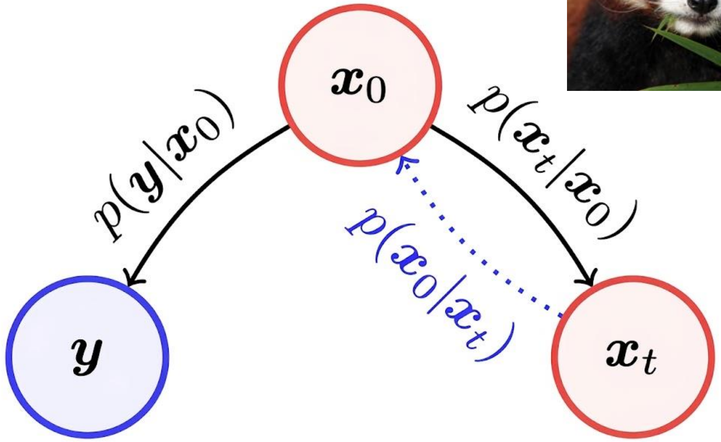
$$\nabla x_0 log p(y|x_0) = -||y - Ax_0||_2^2/\sigma^2$$

Diffusion Posterior Sampling (DPS)

Diffusion Inverse Problem

❖ **Posterior sampling for inverse problems**





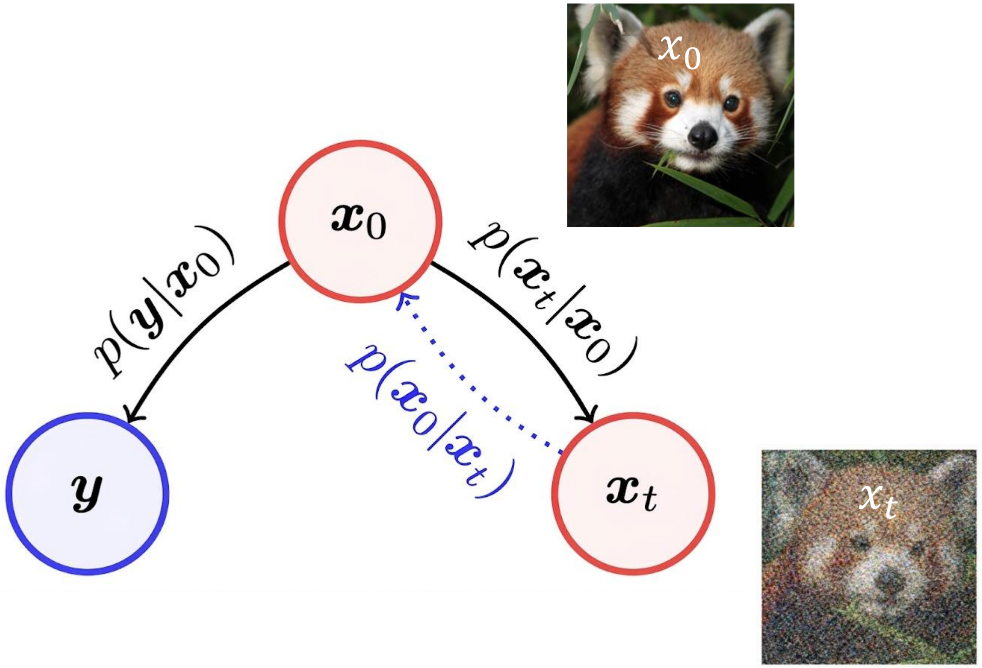
$$p(y|x_0) = N(y|Ax_0, \sigma^2 I)$$

$$\nabla x_0 \log p(y|x_0) = -||y - Ax_0||_2^2 / \sigma^2$$

Diffusion Posterior Sampling (DPS)

Diffusion Inverse Problem

❖ **Posterior sampling for inverse problems**



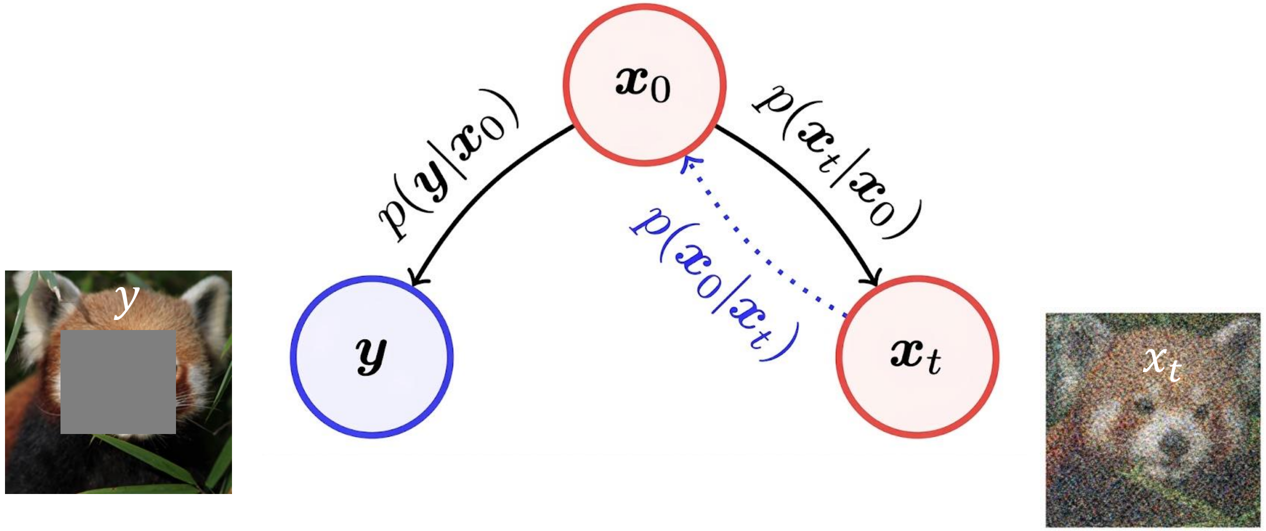
$$p(y|x_0) = N(y|Ax_0, \sigma^2 I)$$

$$\nabla x_0 log p(y|x_0) = -||y - Ax_0||_2^2/\sigma^2$$

Diffusion Posterior Sampling (DPS)

Diffusion Inverse Problem

❖ **Posterior sampling for inverse problems**



$$p(y|x_0) = N(y|Ax_0, \sigma^2 I)$$

$$\nabla x_0 log p(y|x_0) = -||y - Ax_0||_2^2/\sigma^2$$

$\nabla x_t log p(y|x_t)$

이 부분은 계산불가능!

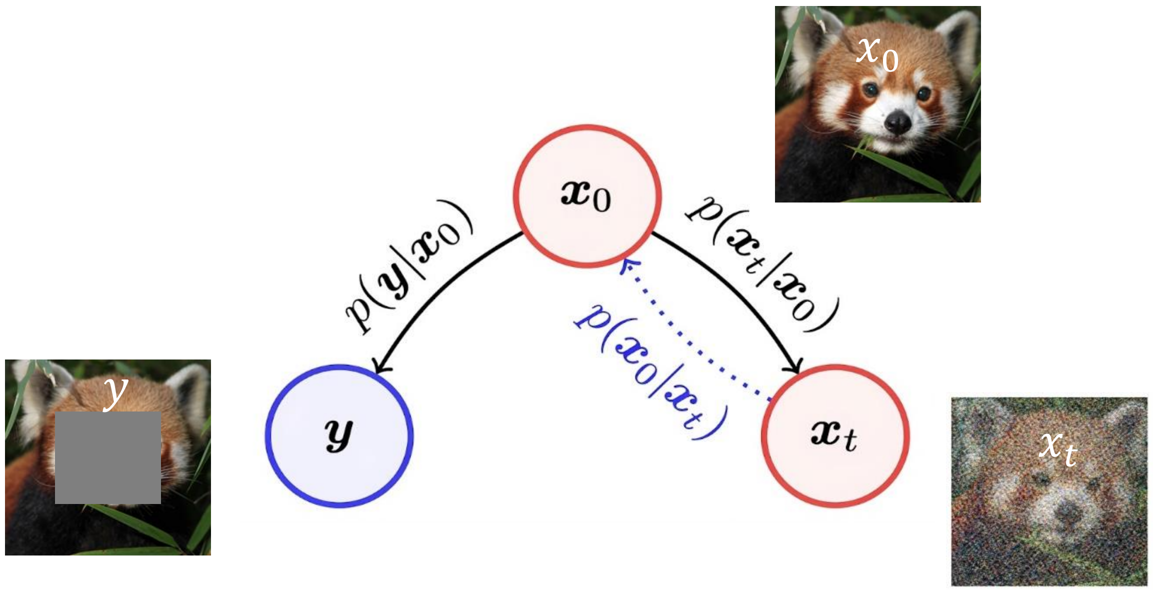
Diffusion Posterior Sampling (DPS)

Diffusion Inverse Problem

❖ **Posterior sampling for inverse problems**

$$p(y|x_t) = \int p(y|x_0, x_t)p(x_0|x_t)dx_0$$

$$p(y|x_t) = \int p(y|x_0)p(x_0|x_t)dx_0$$



Diffusion Posterior Sampling (DPS)

Diffusion Inverse Problem

❖ **Posterior sampling for inverse problems**

$$p(y|x_t) = \int p(y|x_0, x_t)p(x_0|x_t)dx_0$$

$$p(y|x_t) = \int p(y|x_0)p(x_0|x_t)dx_0$$

알고 있는 것 모델이 학습한 정보

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

Diffusion Posterior Sampling (DPS)

Diffusion Inverse Problem

❖ **Posterior sampling for inverse problems**

$$p(y|x_t) = \int p(y|x_0, x_t)p(x_0|x_t)dx_0$$

$$p(y|x_t) = \int p(y|x_0)p(x_0|x_t)dx_0$$

알고 있는 것 **모델이 학습한 정보**

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

Jensen's inequality

$$\simeq p(y|E[x_0|x_t])$$

Diffusion Posterior Sampling (DPS)

Diffusion Inverse Problem

❖ **Posterior sampling for inverse problems**

$$p(y|x_t) = \int p(y|x_0, x_t)p(x_0|x_t)dx_0$$

$$p(y|x_t) = \int {\color{blue}p(y|x_0)}{\color{red}p(x_0|x_t)}dx_0$$

알고 있는 것

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

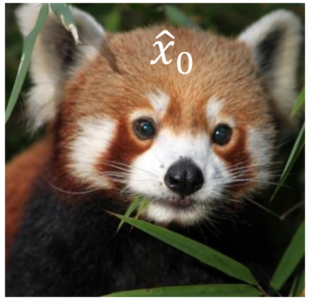
Jensen's inequality

$$\simeq p(y|E[x_0|x_t])$$

$$= p(y|\hat{x}_0)$$

Tweedie's formula

s_{θ^*}



Diffusion Posterior Sampling (DPS)

Diffusion Inverse Problem

❖ **Posterior sampling for inverse problems**

Algorithm 1 DPS - Gaussian

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

$\boldsymbol{x}_N \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

for $i = N - 1$ **to** 0 **do**

$\quad \hat{\boldsymbol{s}} \leftarrow \boldsymbol{s}_\theta(\boldsymbol{x}_i, i)$

$\quad \hat{\boldsymbol{x}}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_i}}(\boldsymbol{x}_i + (1 - \bar{\alpha}_i)\hat{\boldsymbol{s}})$

$\quad \boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

$\quad \boldsymbol{x}'_{i-1} \leftarrow \frac{\sqrt{\alpha_i}(1-\bar{\alpha}_{i-1})}{1-\bar{\alpha}_i}\boldsymbol{x}_i + \frac{\sqrt{\bar{\alpha}_{i-1}}\beta_i}{1-\bar{\alpha}_i}\hat{\boldsymbol{x}}_0 + \tilde{\sigma}_i \boldsymbol{z}$

$\quad \boldsymbol{x}_{i-1} \leftarrow \boldsymbol{x}'_{i-1} - \zeta_i \nabla_{\boldsymbol{x}_i} \|\boldsymbol{y} - \mathcal{A}(\hat{\boldsymbol{x}}_0)\|_2^2$

end for

return $\hat{\mathbf{x}}_0$

Diffusion Posterior Sampling (DPS)

❖ **Posterior sampling for inverse problems**

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** $\hat{\mathbf{x}}_0$

→ $\nabla_x log p(x_t)$

Diffusion Posterior Sampling (DPS)

Diffusion Inverse Problem

❖ **Posterior sampling for inverse problems**

Algorithm 1 DPS - Gaussian

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

$\boldsymbol{x}_N \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$

for $i = N - 1$ **to** 0 **do**

$\quad \hat{\boldsymbol{s}} \leftarrow \boldsymbol{s}_\theta(\boldsymbol{x}_i, i)$

$\quad \hat{\boldsymbol{x}}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_i}}(\boldsymbol{x}_i + (1 - \bar{\alpha}_i)\hat{\boldsymbol{s}})$

$\quad \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$

$\quad \boldsymbol{x}'_{i-1} \leftarrow \frac{\sqrt{\alpha_i}(1-\bar{\alpha}_{i-1})}{1-\bar{\alpha}_i}\boldsymbol{x}_i + \frac{\sqrt{\bar{\alpha}_{i-1}}\beta_i}{1-\bar{\alpha}_i}\hat{\boldsymbol{x}}_0 + \tilde{\sigma}_i \boldsymbol{z}$

$\quad \boldsymbol{x}_{i-1} \leftarrow \boldsymbol{x}'_{i-1} - \zeta_i \nabla_{\boldsymbol{x}_i} \|\boldsymbol{y} - \mathcal{A}(\hat{\boldsymbol{x}}_0)\|_2^2$

end for

return $\hat{\mathbf{x}}_0$

$\nabla_x logp(x_t)$

$\nabla_x logp(y|x_t) \simeq \nabla_x logp(y|\hat{x}_0)$

$$p(y|x_0) = N(y|Ax_0, \sigma^2 I)$$

$$\nabla x_0 logp(y|x_0) = -||y - Ax_0||_2^2/\sigma^2$$

Diffusion Posterior Sampling (DPS)

Diffusion Inverse Problem

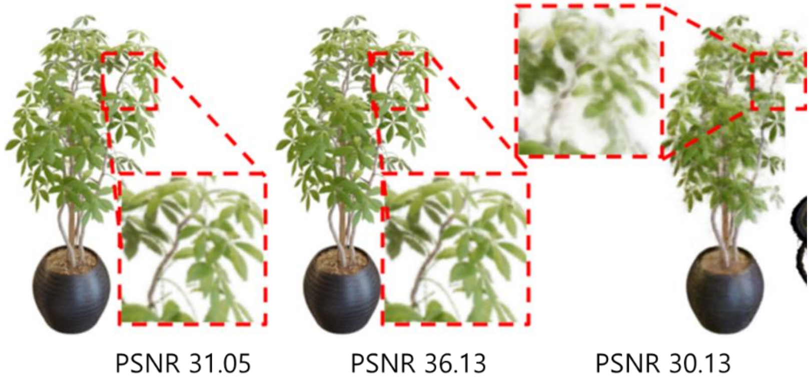
❖ Evaluation Metrics (PSNR, SSIM)

- 픽셀 기반 평가 및 높을수록 품질 우수

정답 이미지 최대 픽셀 값

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right)$$

정답, 예측 이미지 픽셀 차이



$$\text{SSIM}(x,y) = l(x,y)^{\alpha} \cdot c(x,y)^{\beta} \cdot s(x,y)^{\gamma}$$

빛의 밝기 $l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1}$

빛의 대비 $c(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}$

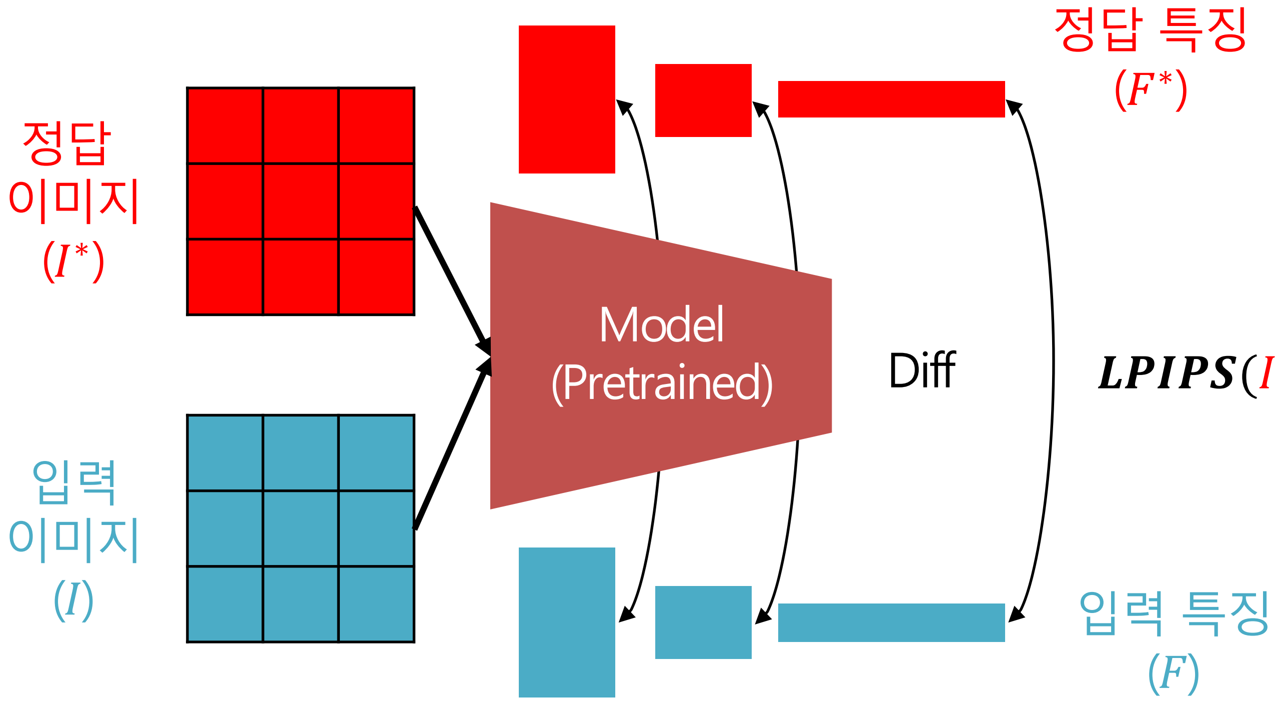
이미지 구조 $s(x,y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3}$

Diffusion Posterior Sampling (DPS)

Diffusion Inverse Problem

❖ Evaluation Metrics (LPIPS)

- 이미지 간의 지각적 유사성을 측정하기 위한 지표
- 딥러닝 모델이 추출한 특성의 유사도를 기반으로 계산



학습된 가중치

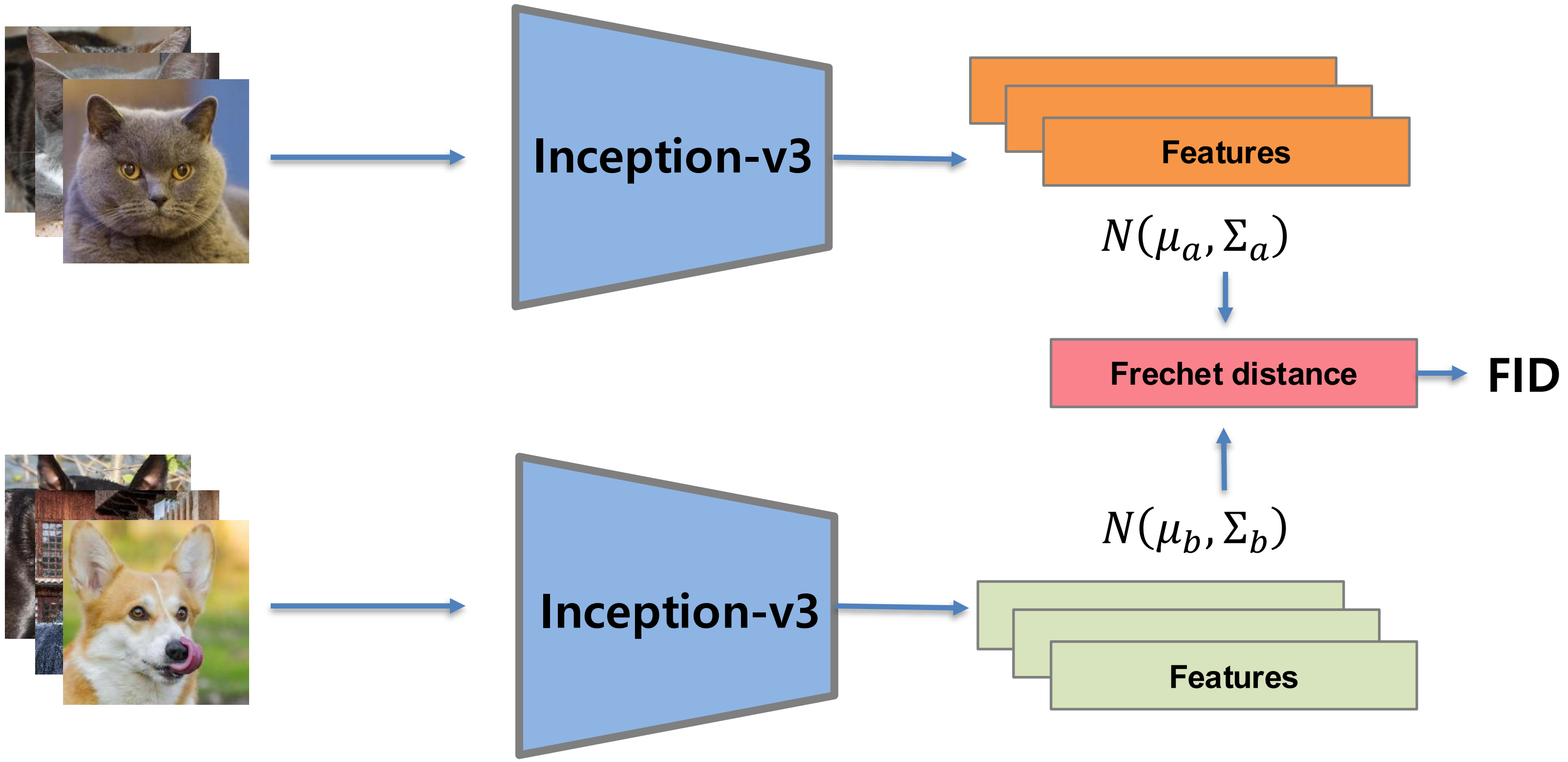
$$LPIPS(I^*, I) = \frac{1}{n}\sum_{i=1}^{n} w_i \cdot (F_i^* - F_{*})^2$$

Diffusion Posterior Sampling (DPS)

Diffusion Inverse Problem

❖ Evaluation Metrics (FID)

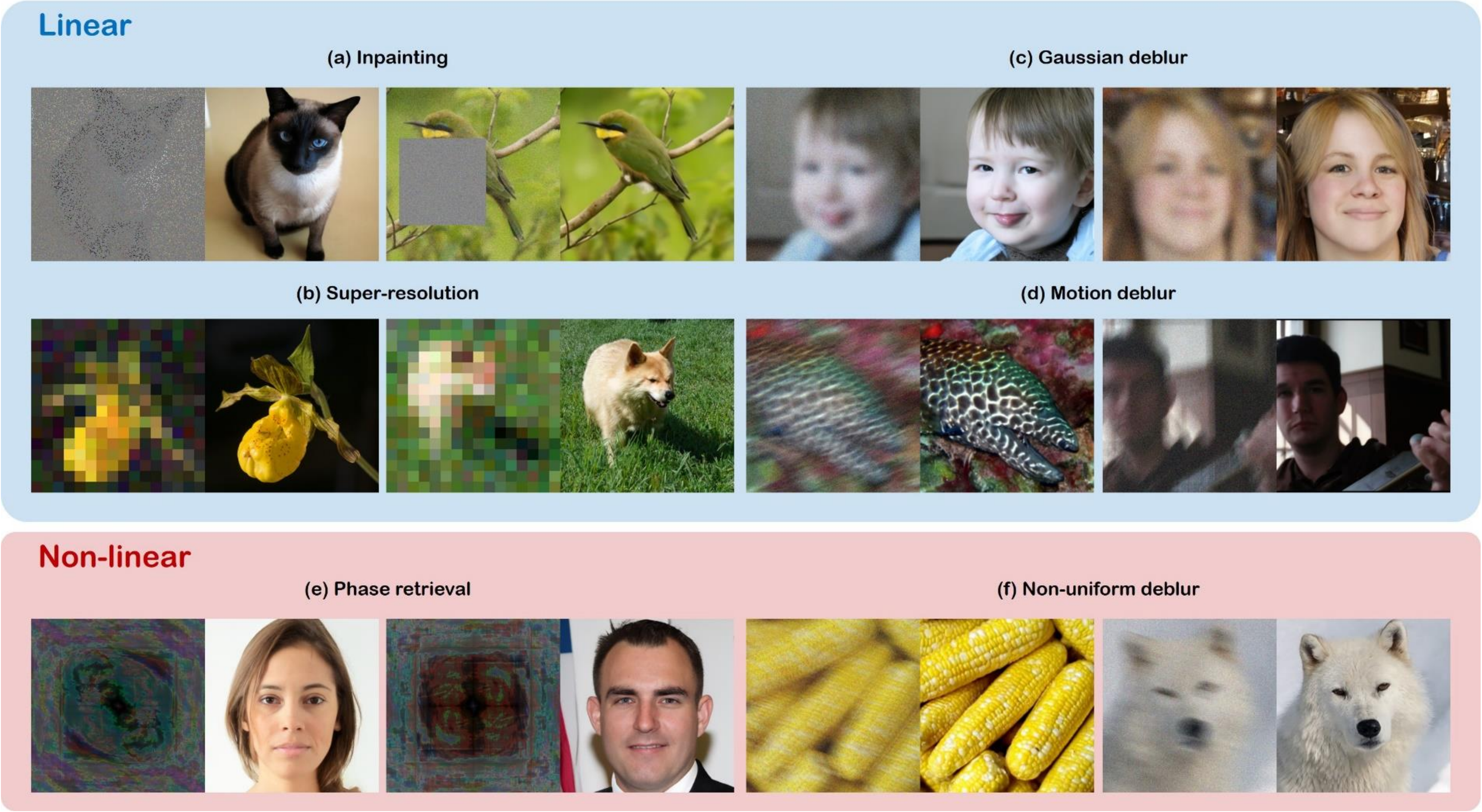
- Incenption v3 네트워크의 feature를 활용해 두 이미지 셋의 전체 분포 차이를 계산
- 단순히 좋아보이는 샘플을 평가하는 것이 아닌, 모델이 전체 데이터 분포를 얼마나 재현했는지를 수치화



Diffusion Posterior Sampling (DPS)

Diffusion Inverse Problem

❖ **Experiments**

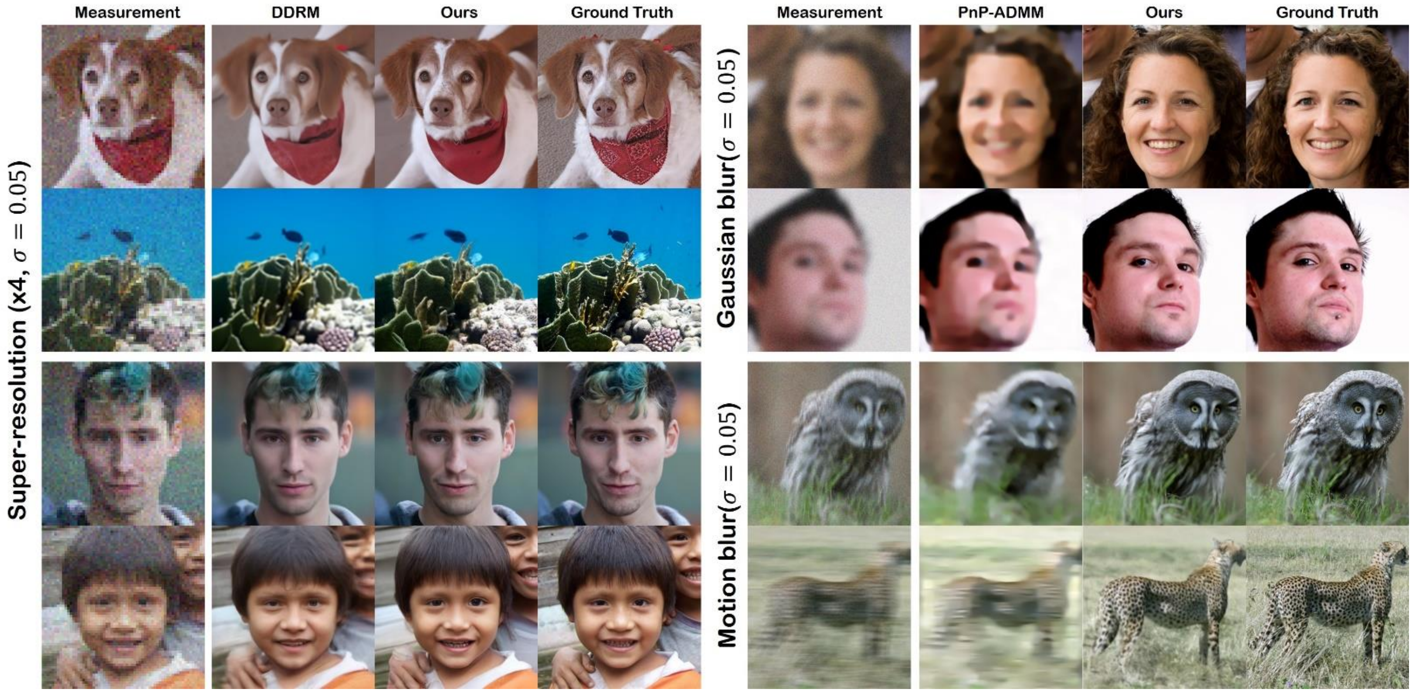


Diffusion Posterior Sampling (DPS)

Diffusion Inverse Problem

❖ Experiments – linear problem

- 정성적으로 비교방법론에 비해 우수한 품질을 보임



Diffusion Posterior Sampling (DPS)

Diffusion Inverse Problem

❖ Experiments – linear inverse problems

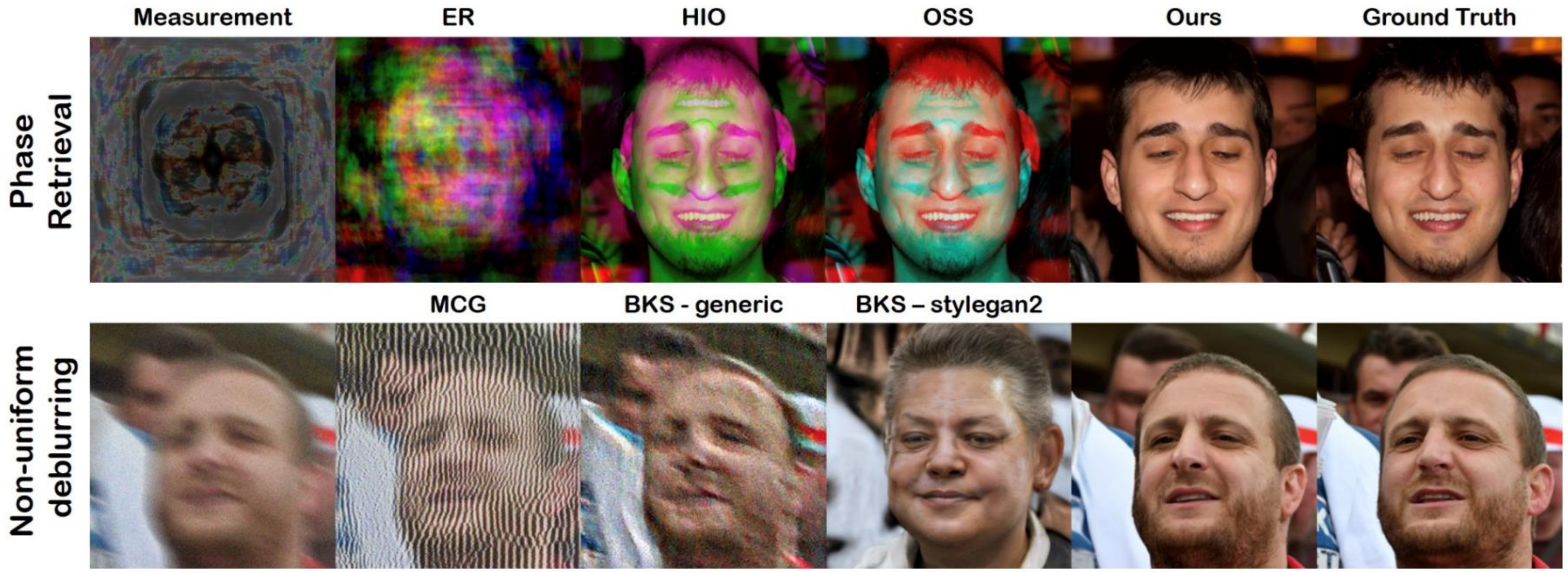
Method	SR (×4)		Inpaint (box)		Inpaint (random)		Deblur (gauss)		Deblur (motion)	
	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓	FID ↓	LPIPS ↓
DPS (ours)	**39.35**	**0.214**	**33.12**	**0.168**	**21.19**	**0.212**	**44.05**	**0.257**	**39.92**	**0.242**
DDRM (Kawar et al., 2022)	62.15	0.294	42.93	0.204	69.71	0.587	74.92	0.332	-	-
MCG (Chung et al., 2022a)	87.64	0.520	40.11	0.309	29.26	0.286	101.2	0.340	310.5	0.702
PnP-ADMM (Chan et al., 2016)	66.52	0.353	151.9	0.406	123.6	0.692	90.42	0.441	89.08	0.405
Score-SDE (Song et al., 2021b) (ILVR (Choi et al., 2021))	96.72	0.563	60.06	0.331	76.54	0.612	109.0	0.403	292.2	0.657
ADMM-TV	110.6	0.428	68.94	0.322	181.5	0.463	186.7	0.507	152.3	0.508

Table 1: Quantitative evaluation (FID, LPIPS) of solving linear inverse problems on FFHQ 256×256-1k validation dataset. **Bold**: best, underline: second best.

Diffusion Posterior Sampling (DPS)

Diffusion Inverse Problem

❖ **Experiments – nonlinear inverse problems**



Improving Diffusion Models for Inverse Problems using Manifold Constraints

NeurIPS 2022

Manifold Constraints Gradient (MCG)

Diffusion Inverse Problem

❖ Improving Diffusion Models for Inverse Problems using Manifold Constraints

- NeurIPS 2022, KAIST
- 2026년 4월 10일 기준 682회 인용

Improving Diffusion Models for Inverse Problems using Manifold Constraints

Hyungjin Chung*[,1] Byeongsu Sim*[,2] Dohoon Ryu[1] Jong Chul Ye[3,1,2]

[1] Dept. of Bio and Brain Engineering

[2] Dept. of Mathematical Sciences

[3]Kim Jaechul Graduate School of AI

*Equal contribution

Korea Advanced Institute of Science and Technology (KAIST)

{hj.chung, byeongsu.s, dh.ryu, jong.ye}@kaist.ac.kr

Abstract

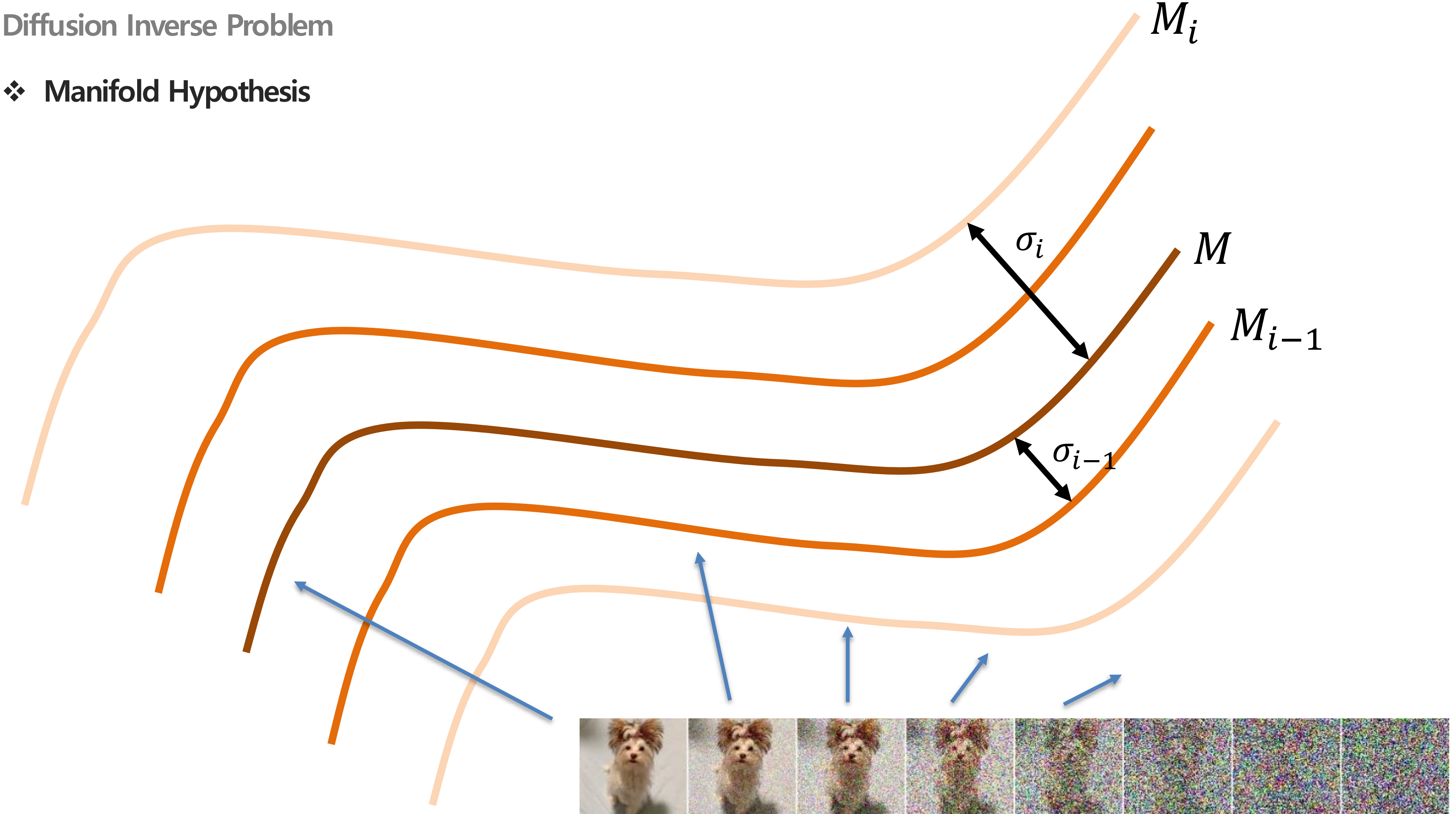
Recently, diffusion models have been used to solve various inverse problems in an unsupervised manner with appropriate modifications to the sampling process. However, the current solvers, which recursively apply a reverse diffusion step followed by a projection-based measurement consistency step, often produce sub-optimal results. By studying the generative sampling path, here we show that current solvers throw the sample path off the data manifold, and hence the error accumulates. To address this, we propose an additional correction term inspired by the manifold constraint, which can be used synergistically with the previous solvers to make the iterations close to the manifold. The proposed manifold constraint is straightforward to implement within a few lines of code, yet boosts the performance by a surprisingly large margin. With extensive experiments, we show that our method is superior to the previous methods both theoretically and empirically, producing promising results in many applications such as image inpainting, colorization, and sparse-view computed tomography. Code available here

Chung, H., Sim, B., Ryu, D., & Ye, J. C. (2022). Improving diffusion models for inverse problems using manifold constraints. *Advances in Neural Information Processing Systems, 35*, 25683-25696.

Manifold Constraints Gradient (MCG)

Diffusion Inverse Problem

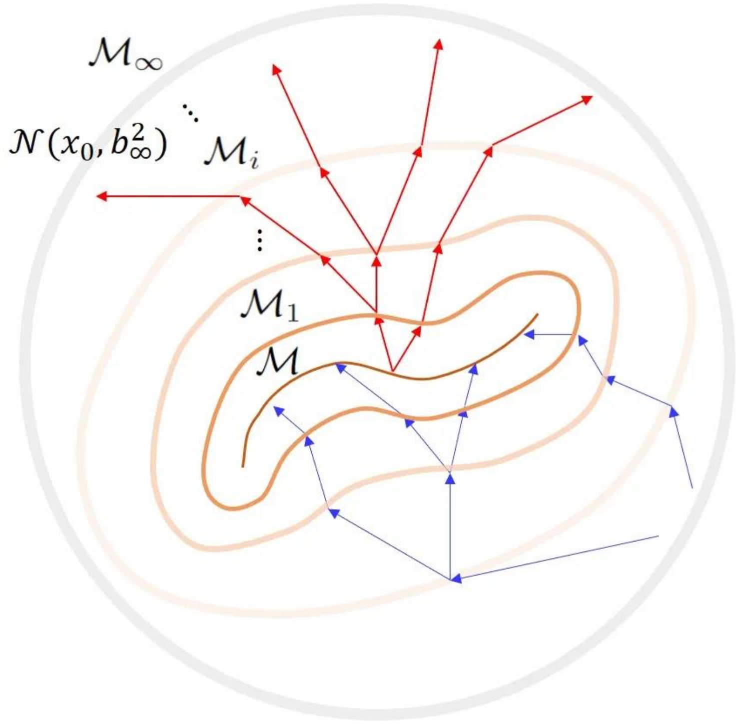
❖ **Manifold Hypothesis**



Manifold Constraints Gradient (MCG)

Diffusion Inverse Problem

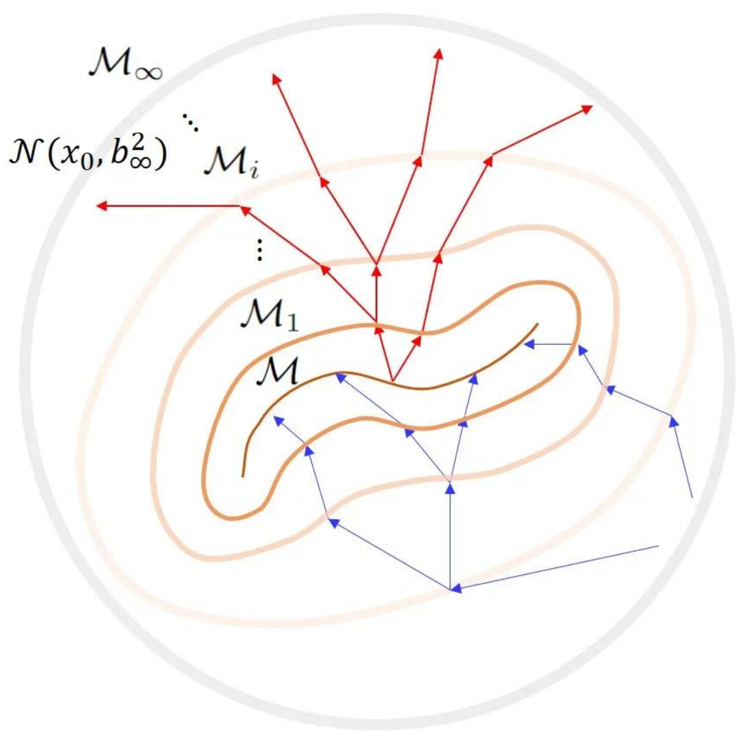
❖ **Manifold Hypothesis**

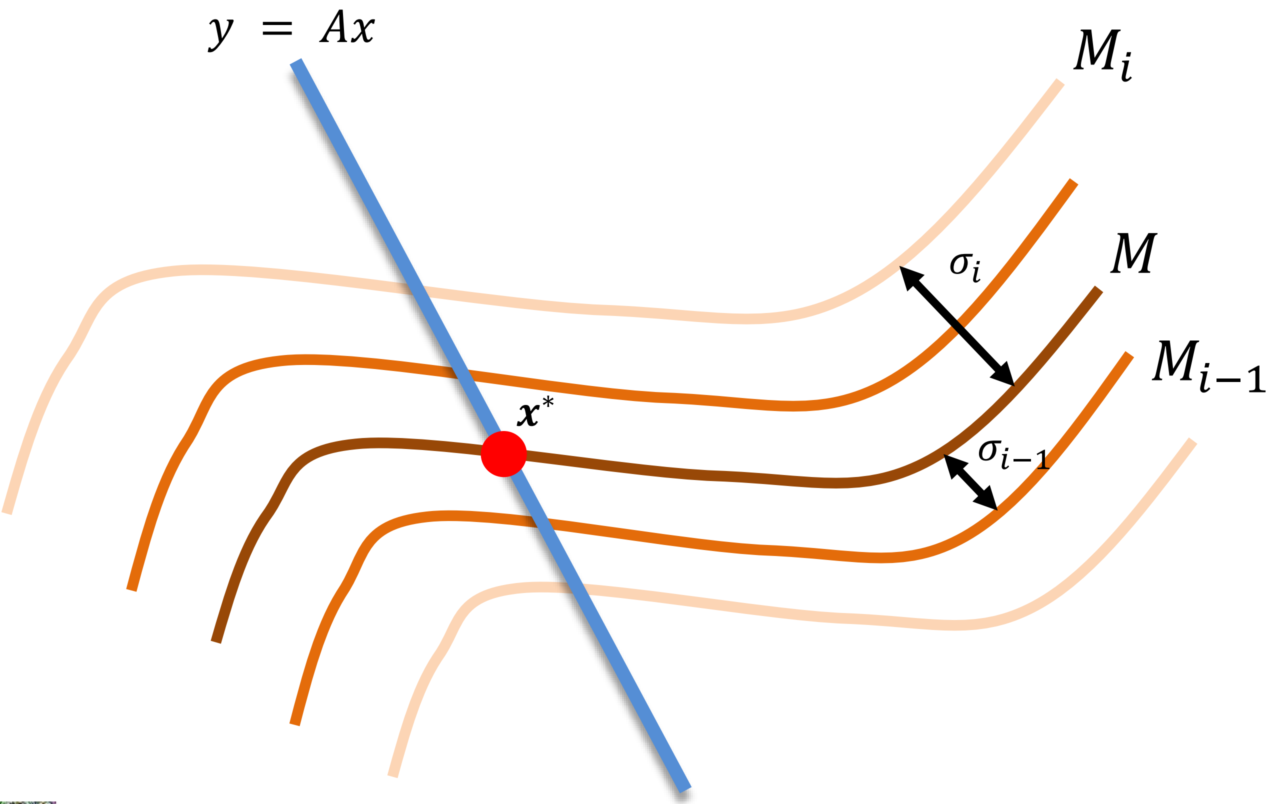


Manifold Constraints Gradient (MCG)

Diffusion Inverse Problem

❖ **Manifold Hypothesis**

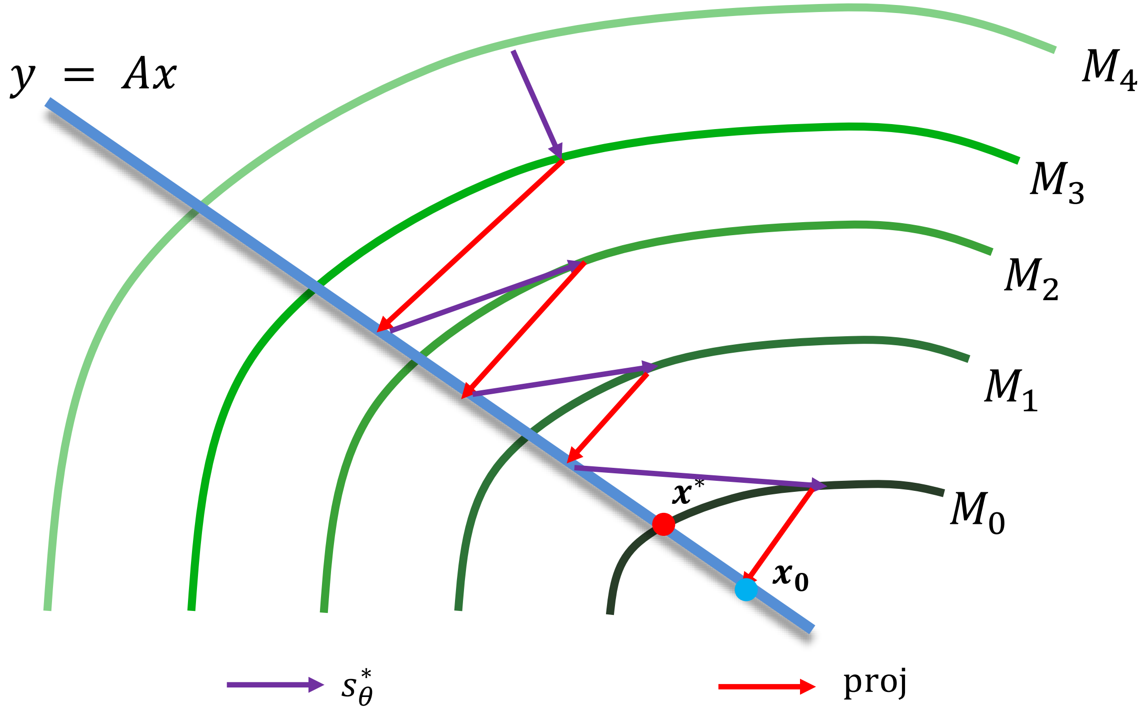




Manifold Constraints Gradient (MCG)

Diffusion Inverse Problem

❖ Score-SDE inverse problem



Geometry of Score-SDE

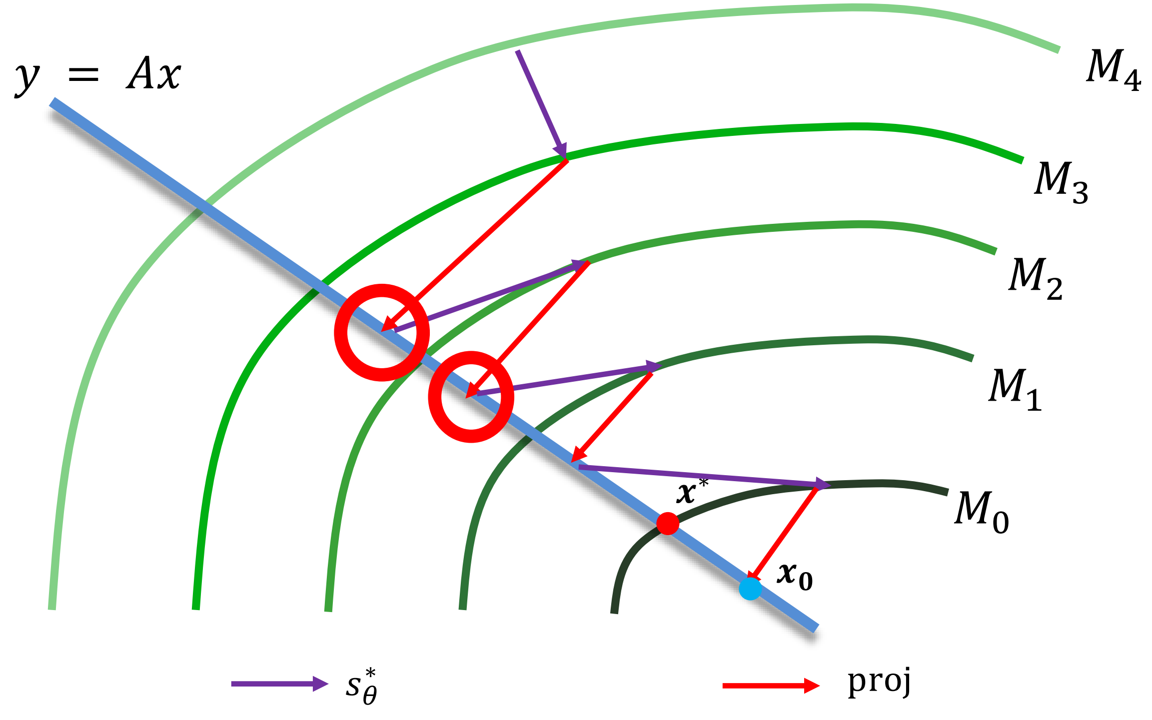
1. $x_i' \leftarrow s_\theta^*(x_i, i)$
2. $Proj_y(x_i')$

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*.

Manifold Constraints Gradient (MCG)

Diffusion Inverse Problem

❖ Score-SDE inverse problem



Geometry of Score-SDE

1. $x_i^{'} \leftarrow s_\theta^*(x_i, i)$
2. $Proj_y(x_i^{'})$

Manifold 영역 밖으로 벗어남

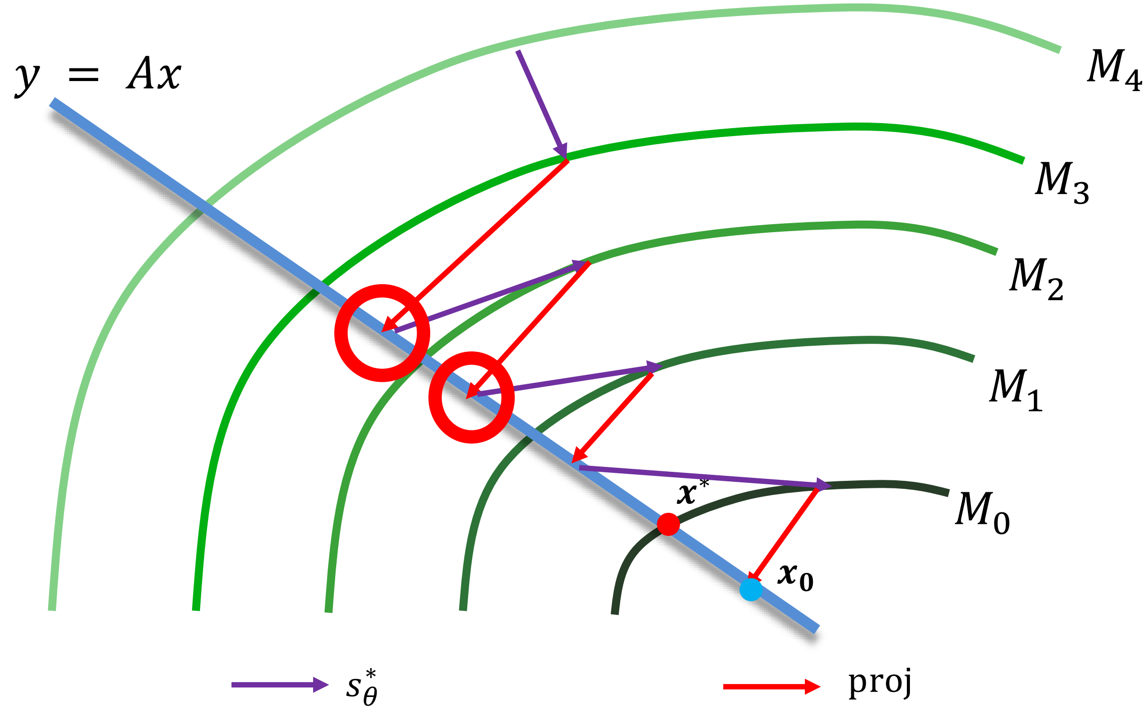
→ 엉뚱한 생성 결과물

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*.

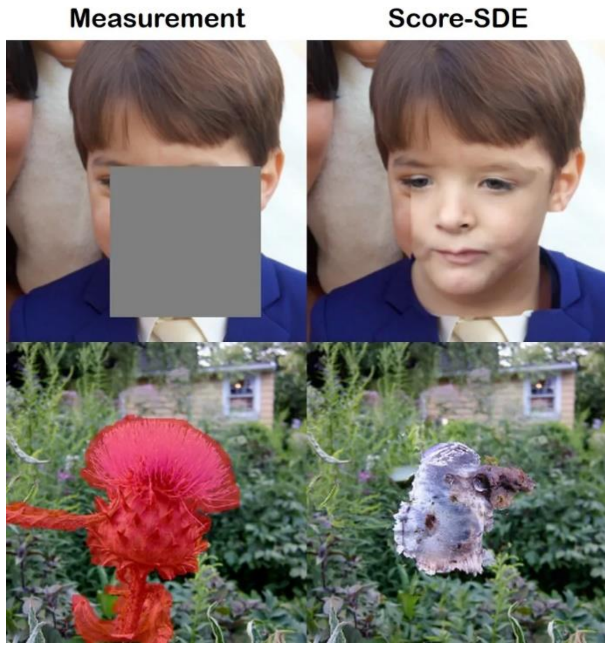
Manifold Constraints Gradient (MCG)

Diffusion Inverse Problem

❖ Score-SDE inverse problem



Geometry of Score-SDE



Manifold 영역 밖으로 벗어남

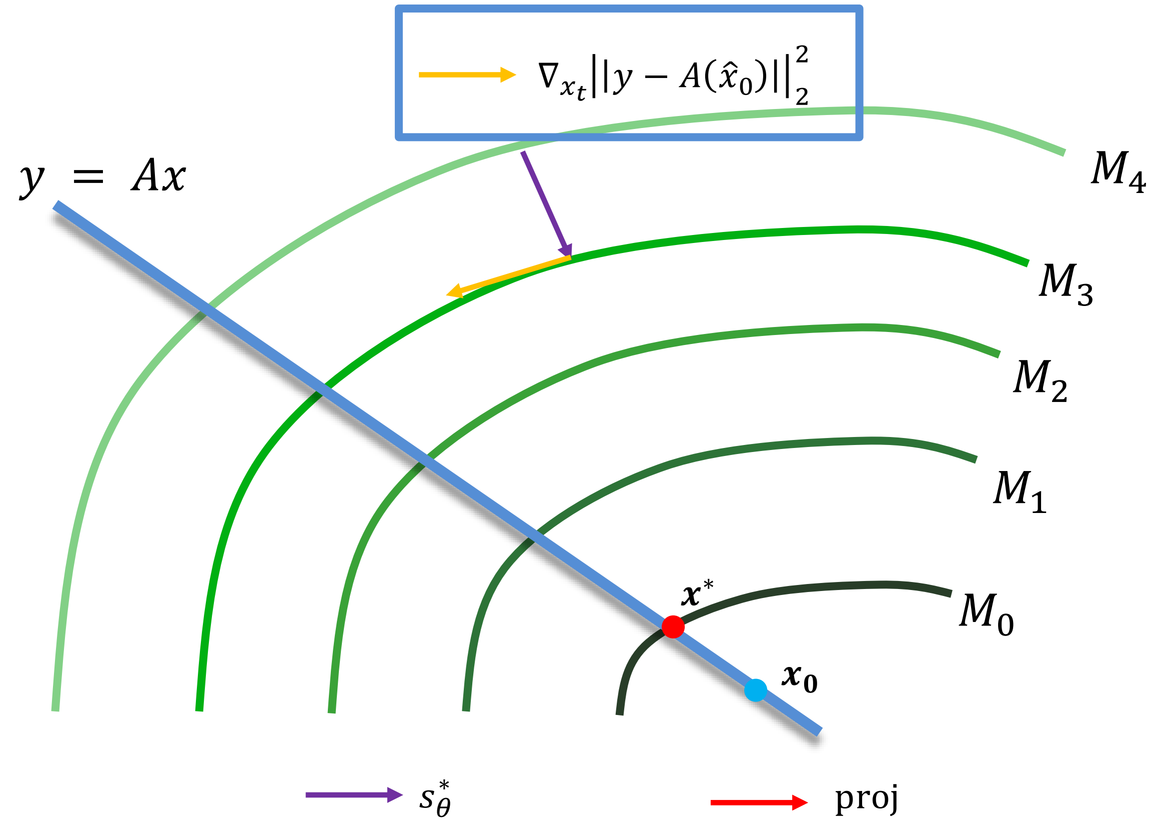
→ 엉뚱한 생성 결과물

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*.

Manifold Constraints Gradient (MCG)

Diffusion Inverse Problem

❖ **Theorem1 : MCG는 현재 manifold에서 y로 향하는 접공간에 해당하는 방향으로 스텝을 취한다.**

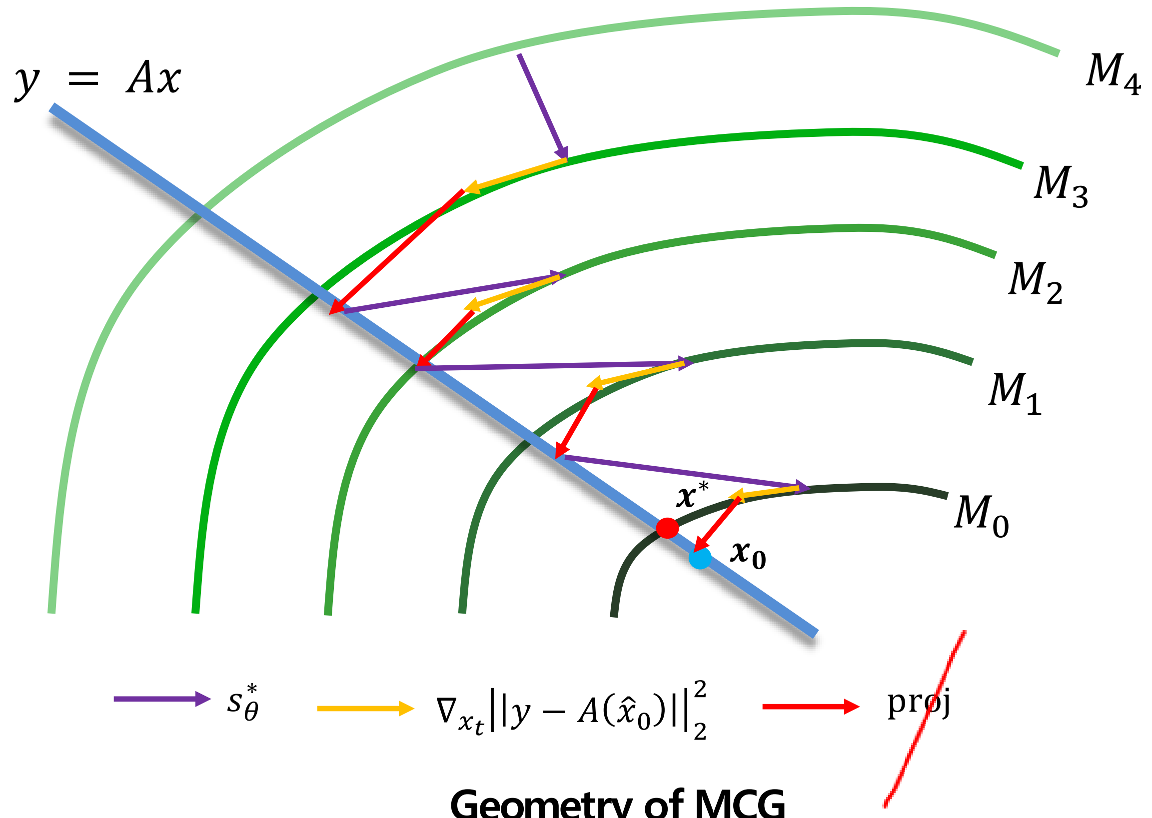


Geometry of MCG

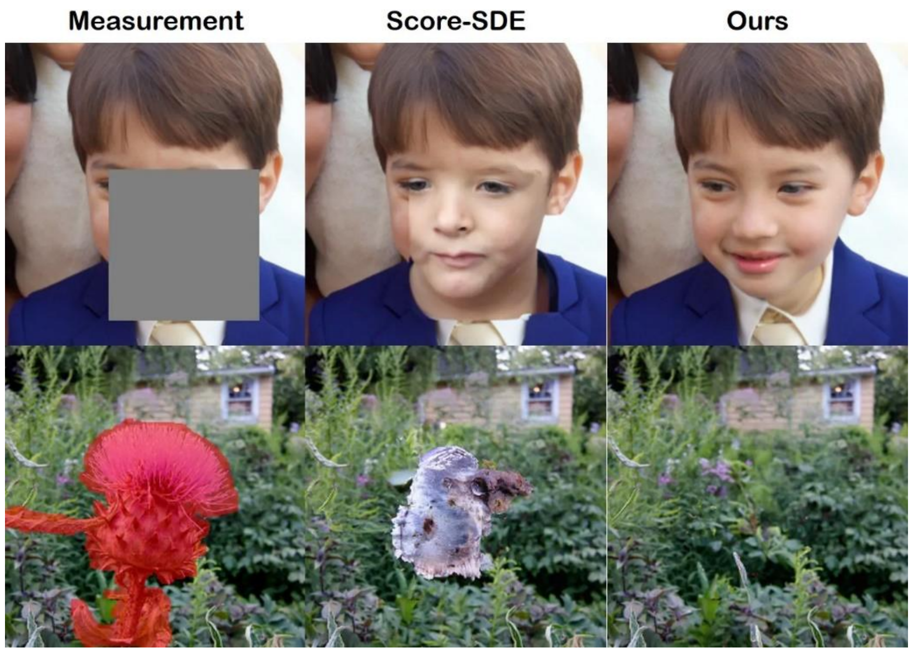
Manifold Constraints Gradient (MCG)

Diffusion Inverse Problem

❖ **Theorem1 : MCG는 현재 manifold에서 y로 향하는 접공간에 해당하는 방향으로 스텝을 취한다.**



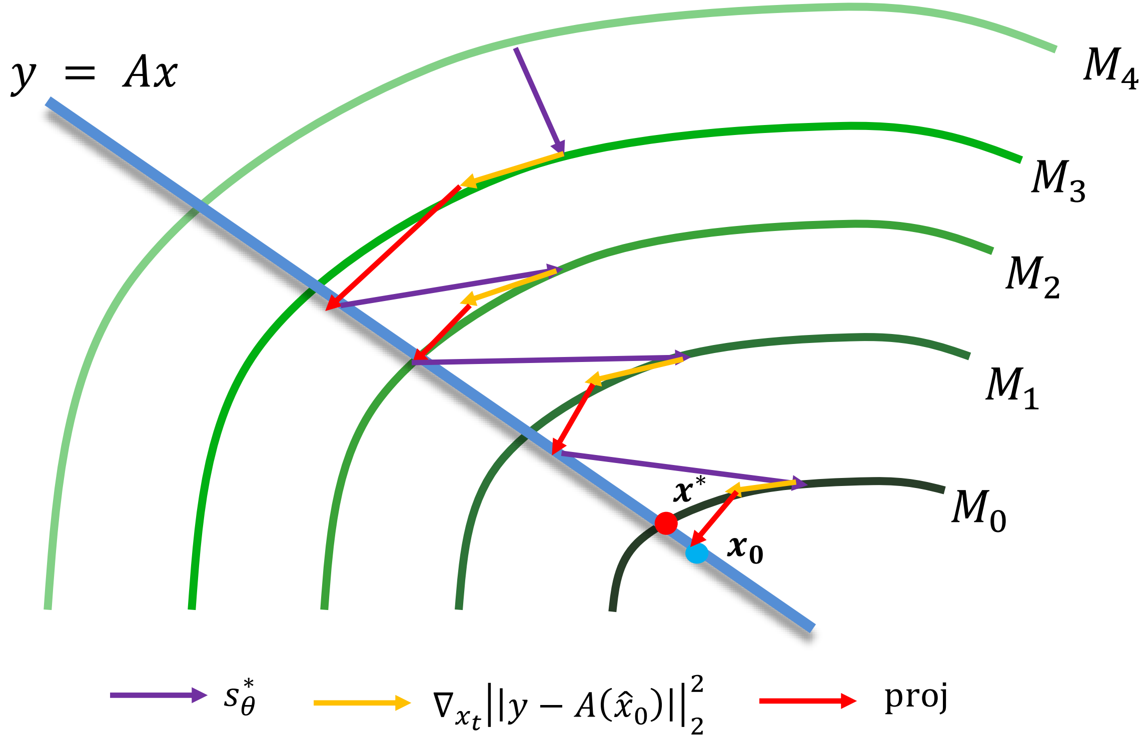
Geometry of MCG



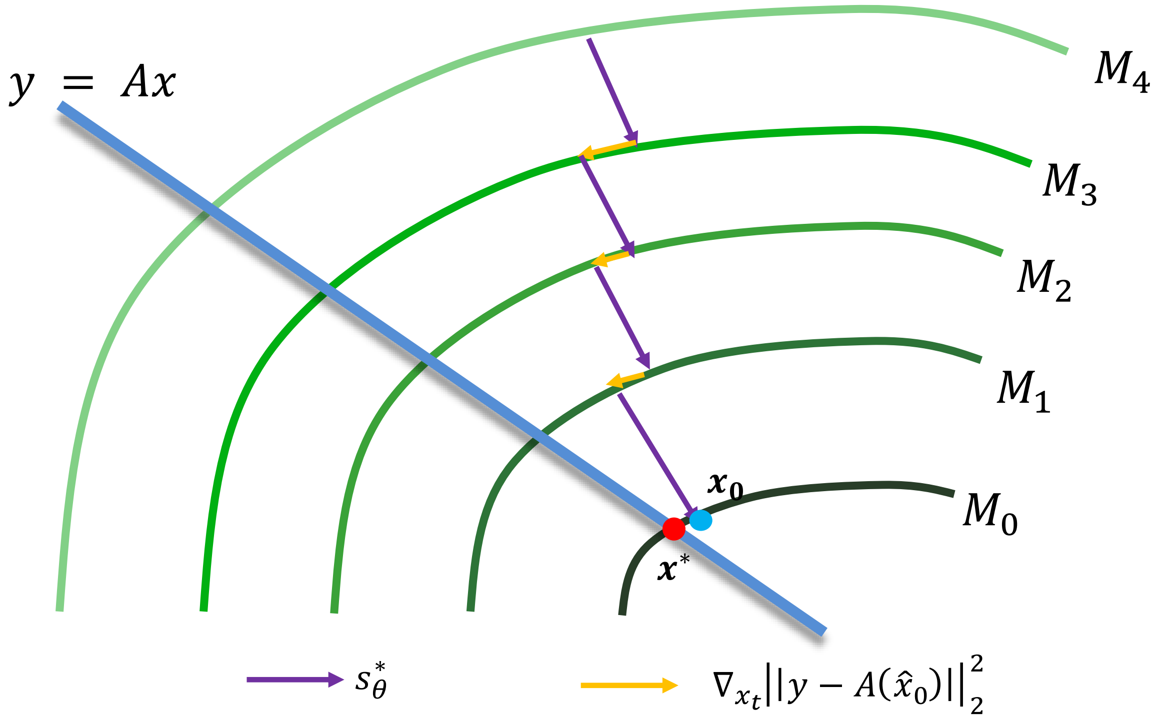
MCG, DPS geometric interpretation

Diffusion Inverse Problem

❖ MCG, DPS geometric interpretation



Geometry of MCG



Geometry of DPS

Improving Diffusion Inverse Problem Solving with Decoupled Noise Annealing

CVPR 2025 ORAL

Decoupled Annealing Posterior Sampling (DAPS)

Diffusion Inverse Problem

❖ Improving Diffusion Inverse Problem Solving with Decoupled Noise Annealing

- CVPR 2025 Oral, California Institute of Technology
- 2026년 4월 10일 기준 91회 인용

Improving Diffusion Inverse Problem Solving with Decoupled Noise Annealing

Bingliang Zhang*[,1] Wenda Chu*[,1] Julius Berner[2,†]
Chenlin Meng[3] Anima Anandkumar[1] Yang Song[4]
[1]California Institute of Technology [2]NVIDIA [3]Stanford University [4]OpenAI

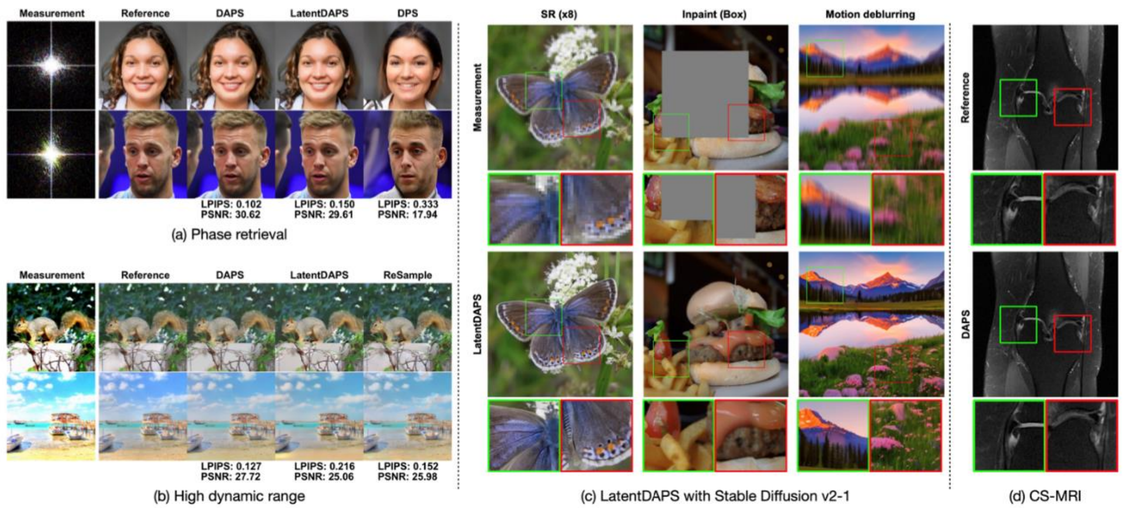


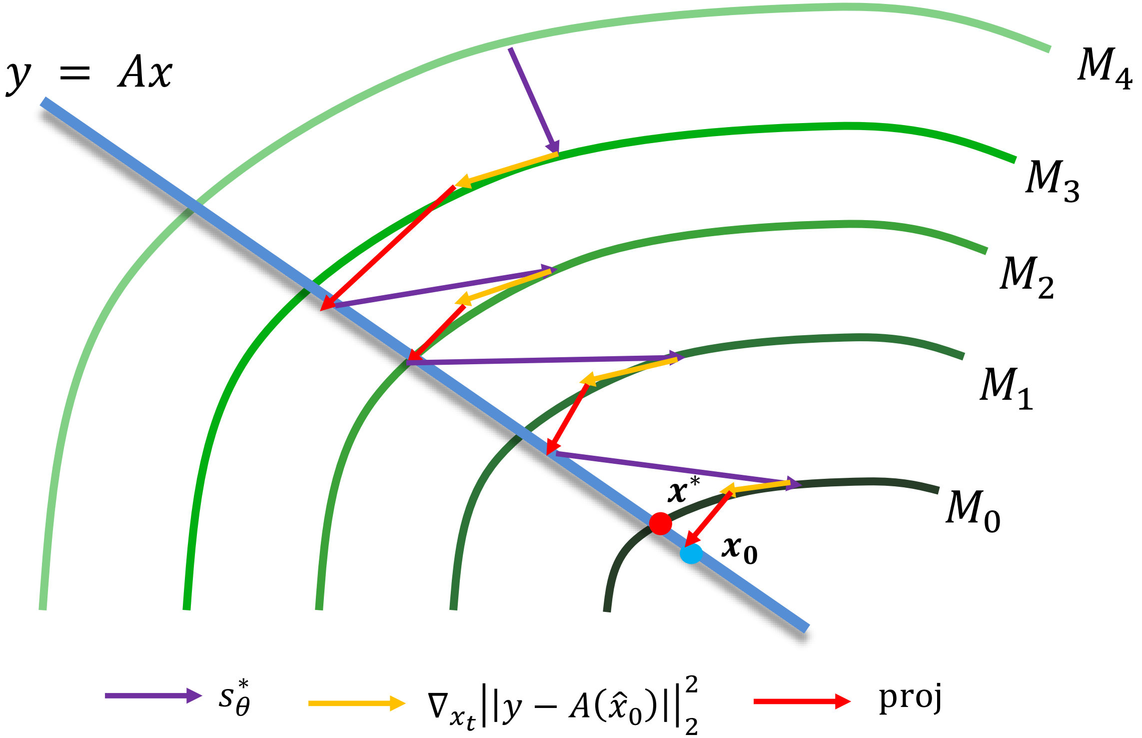
Figure 1. **Overview of Decoupled Annealing Posterior Sampling (DAPS).** Our method provides a flexible and effective framework for solving inverse problems through a decoupled posterior sampling process. In (a)(b), we present DAPS visual results on FFHQ and ImageNet at a resolution of 256, and in (c), on natural images at a resolution of 768. In (d), we display DAPS results on compressed sensing multi-coil MRI (CS-MRI). DAPS effectively addresses nonlinear inverse problems as well as medical imaging MRI challenges. Additionally, DAPS can be enhanced using large-scale latent diffusion models (LDMs) [40], as shown in (c).

Zhang, B., Chu, W., Berner, J., Meng, C., Anandkumar, A., & Song, Y. (2025). Improving diffusion inverse problem solving with decoupled noise annealing. In *Proceedings of the Computer Vision and Pattern Recognition Conference* (pp. 20895-20905).

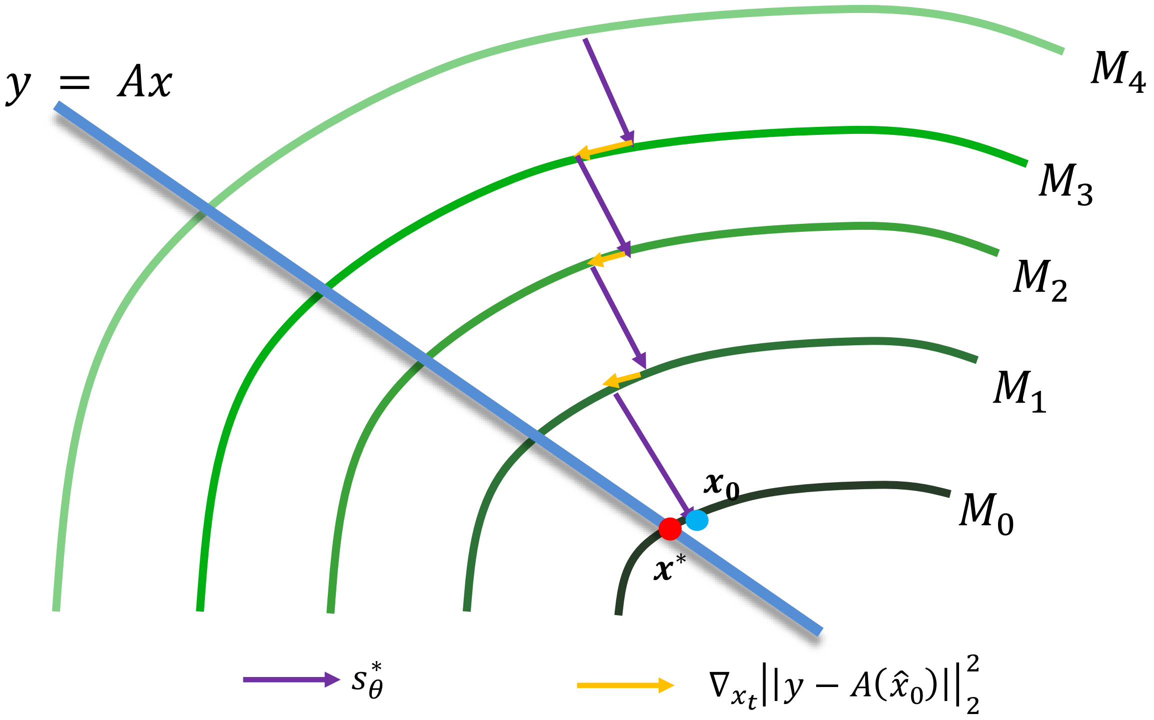
Decoupled Annealing Posterior Sampling (DAPS)

Diffusion Inverse Problem

❖ MCG DPS geometric interpretation



Geometry of MCG

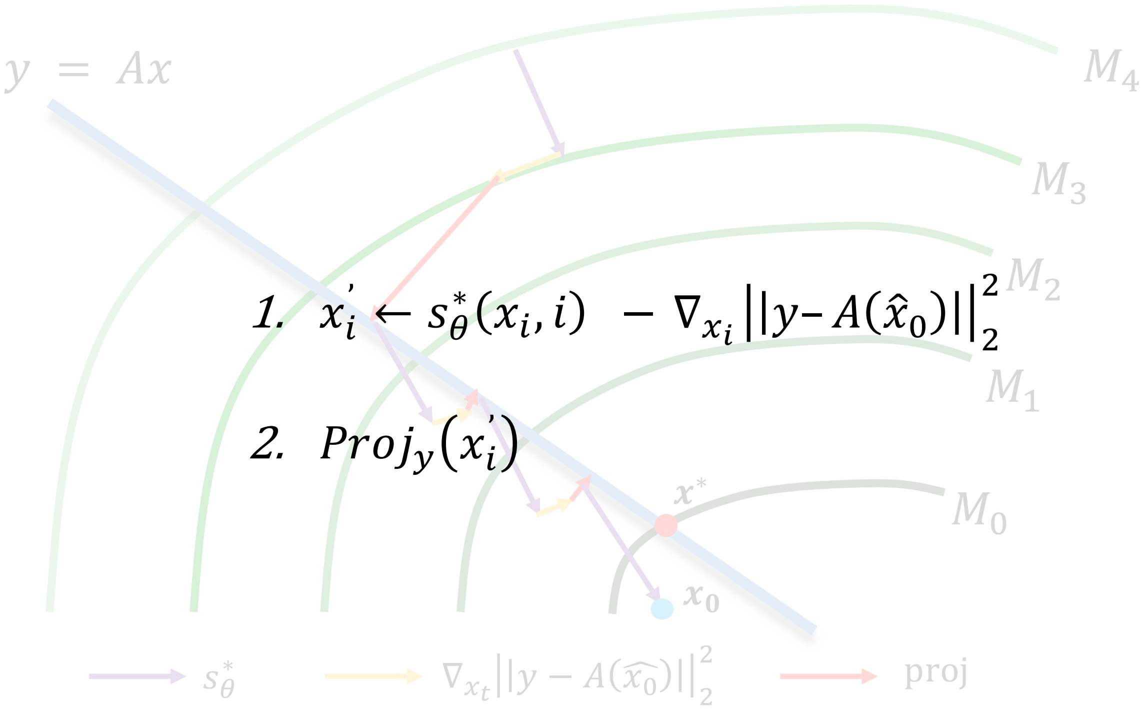


Geometry of DPS

Decoupled Annealing Posterior Sampling (DAPS)

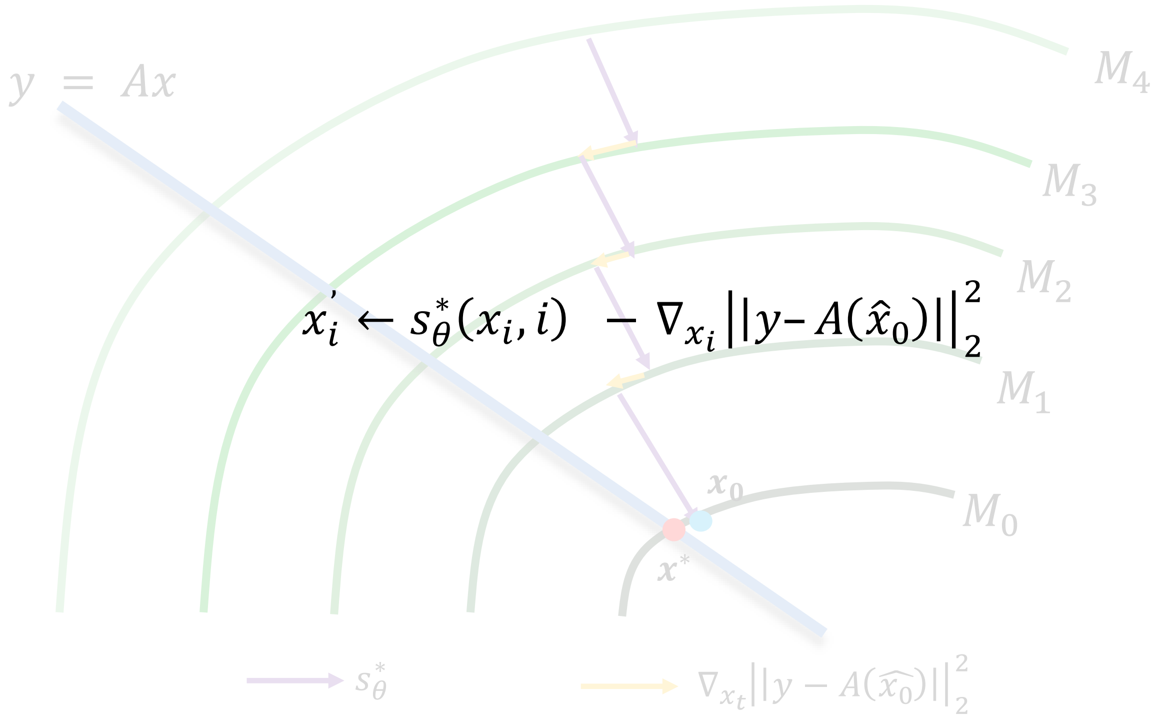
Diffusion Inverse Problem

❖ MCG, DPS geometric interpretation



1. $x_i' \leftarrow s_\theta^*(x_i, i) \;-\; \nabla_{x_i}||y - A(\hat{x}_0)||_2^2$
2. $Proj_y(x_i')$

Geometry of MCG



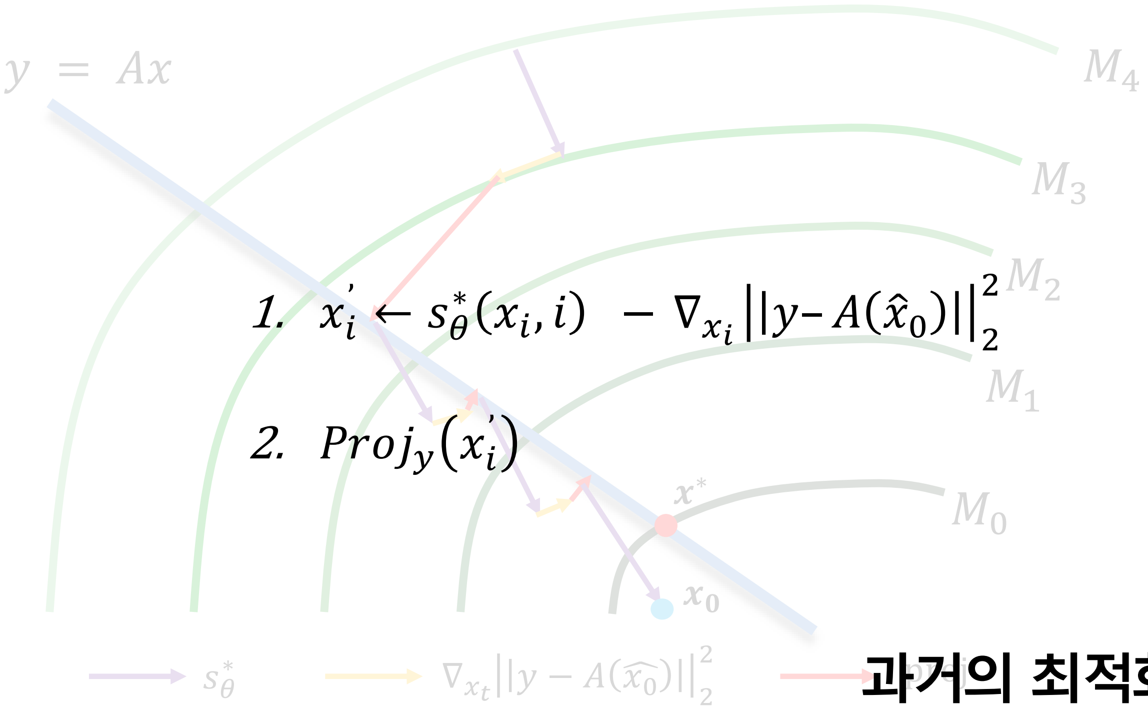
$$x_i' \leftarrow s_\theta^*(x_i, i) \;-\; \nabla_{x_i}||y - A(\hat{x}_0)||_2^2$$

Geometry of DPS

Decoupled Annealing Posterior Sampling (DAPS)

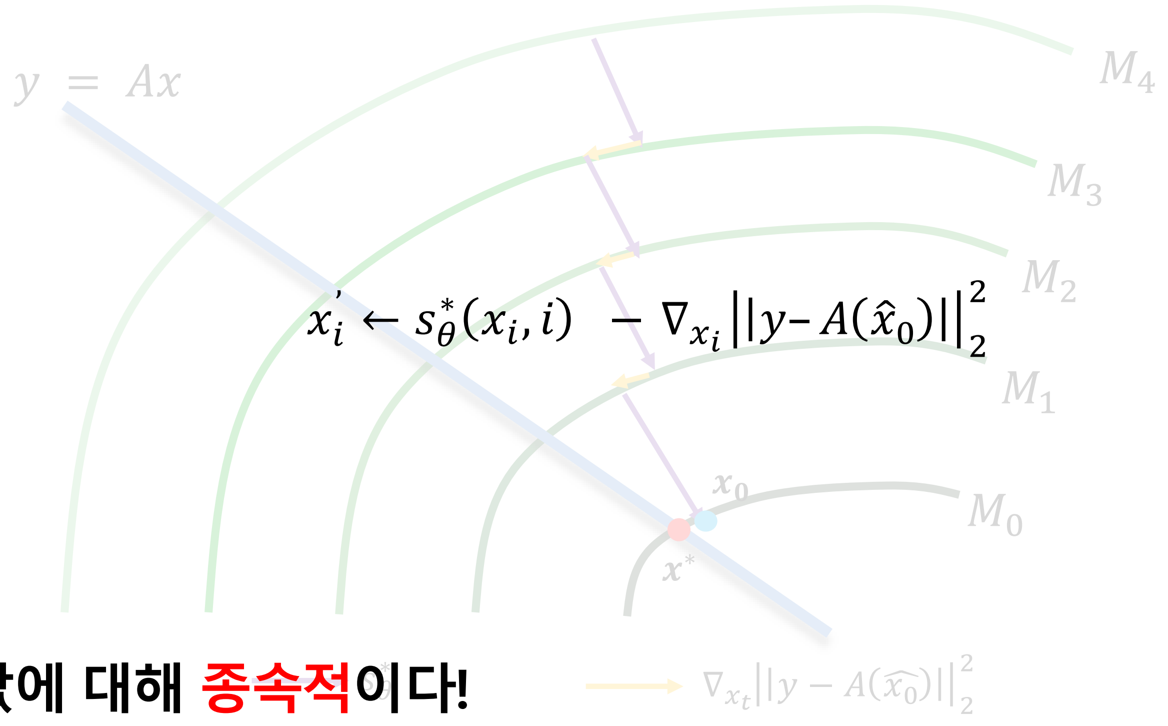
Diffusion Inverse Problem

❖ MCG, DPS geometric interpretation



1. $x_i' \leftarrow s_\theta^*(x_i, i) \; - \nabla_{x_i} ||y - A(\hat{x}_0)||_2^2$
2. $Proj_y(x_i')$

Geometry of MCG



$x_i' \leftarrow s_\theta^*(x_i, i) \; - \nabla_{x_i} ||y - A(\hat{x}_0)||_2^2$

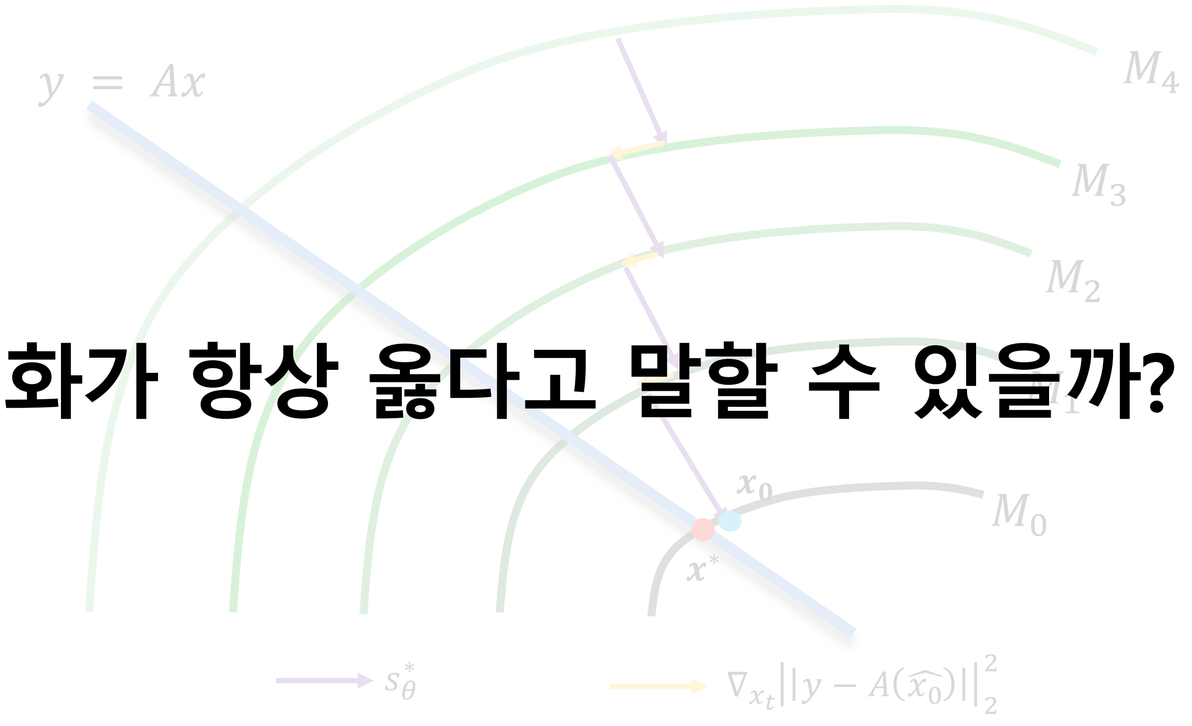
Geometry of DPS

과거의 최적화 값에 대해 종속적이다!

$x_{i-1}' = f(x_i, s_\theta) - \alpha + \sigma_i z$

❖ MCG, DPS geometric interpretation

Geometry of MCG

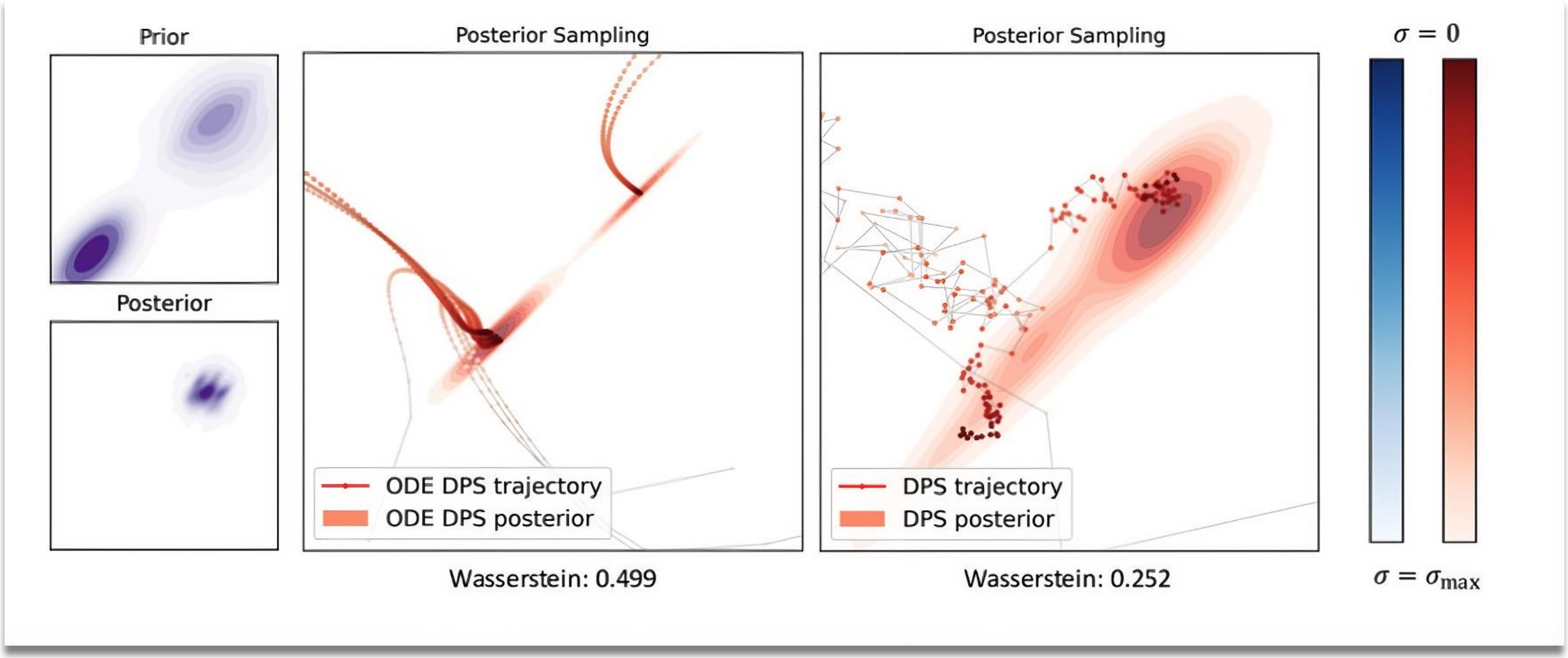


Geometry of DPS

이전 step에서의 최적화가 항상 옳다고 말할 수 있을까?

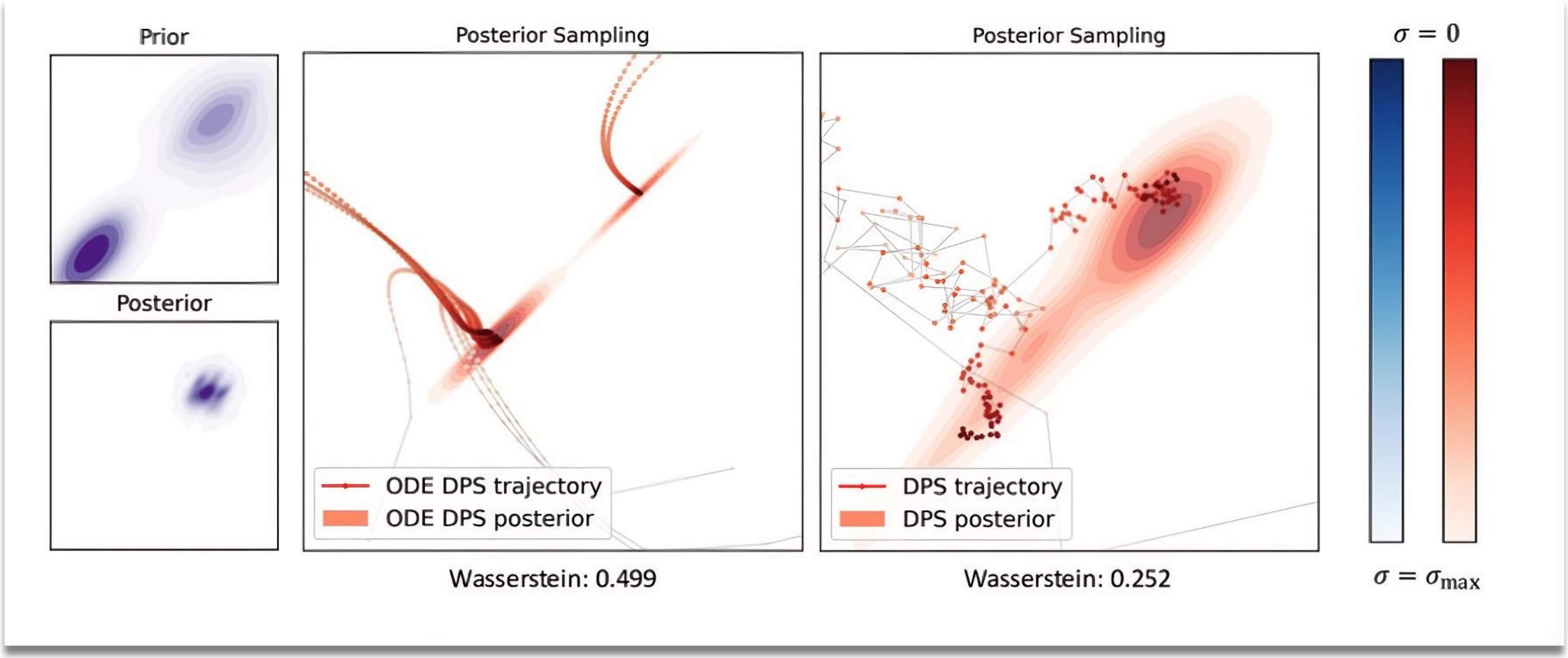
Decoupled Annealing Posterior Sampling (DAPS)

Diffusion Inverse Problem



Decoupled Annealing Posterior Sampling (DAPS)

Diffusion Inverse Problem



$p(\boldsymbol{x}_t|\boldsymbol{x}_{t+\Delta t}, \boldsymbol{y}),\ \boldsymbol{t} \leftarrow \boldsymbol{t} + \boldsymbol{\Delta t}$ 에 대한 업데이트

Local error가 누적되면서 global error에 대해 업데이트 불가

Decoupled Annealing Posterior Sampling (DAPS)

Diffusion Inverse Problem

$p(\boldsymbol{x}_t|\boldsymbol{x}_{t+\Delta t}, \boldsymbol{y})$, $\boldsymbol{t} \leftarrow \boldsymbol{t} + \boldsymbol{\Delta t}$ 에 대한 업데이트

Local error가 누적되면서 global error에 대해 업데이트 불가

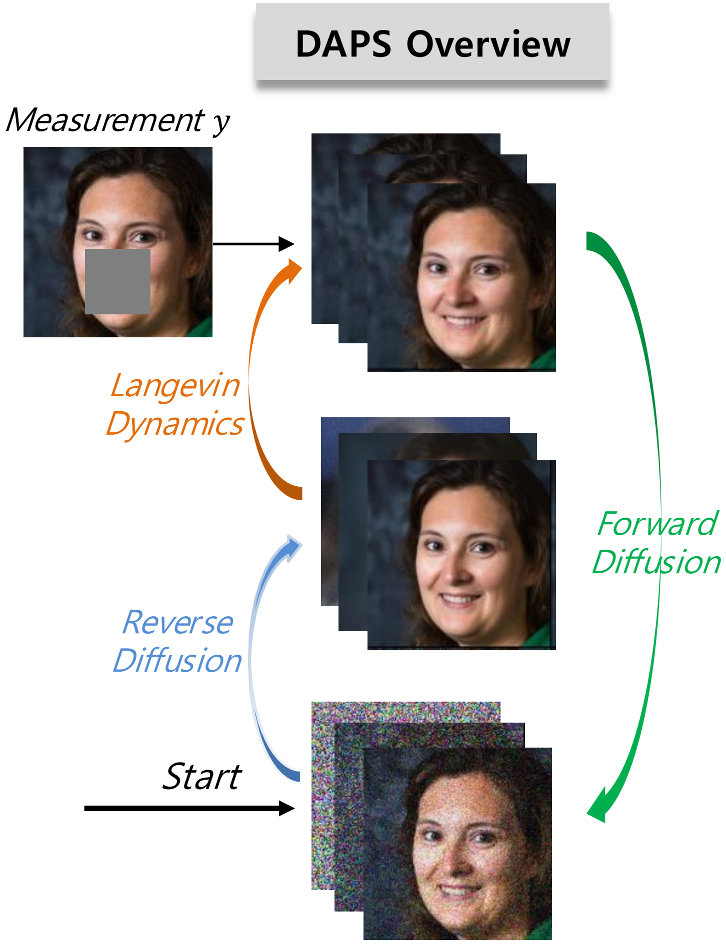
$\boldsymbol{t} \leftarrow \boldsymbol{t} + \boldsymbol{\Delta t}$ 에 대한 종속성을 제거하자 !

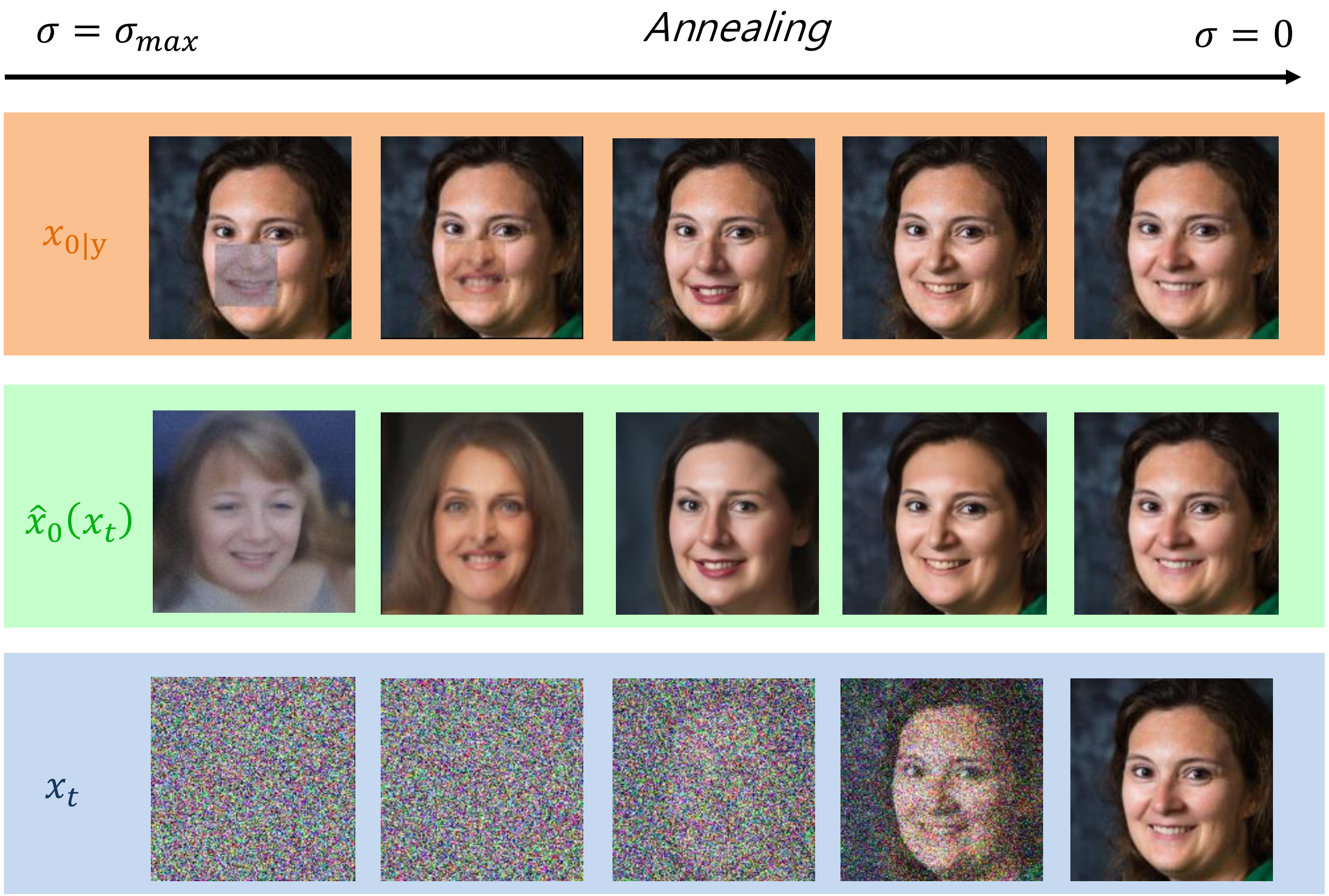
→ Decoupling!

Decoupled Annealing Posterior Sampling (DAPS)

Diffusion Inverse Problem

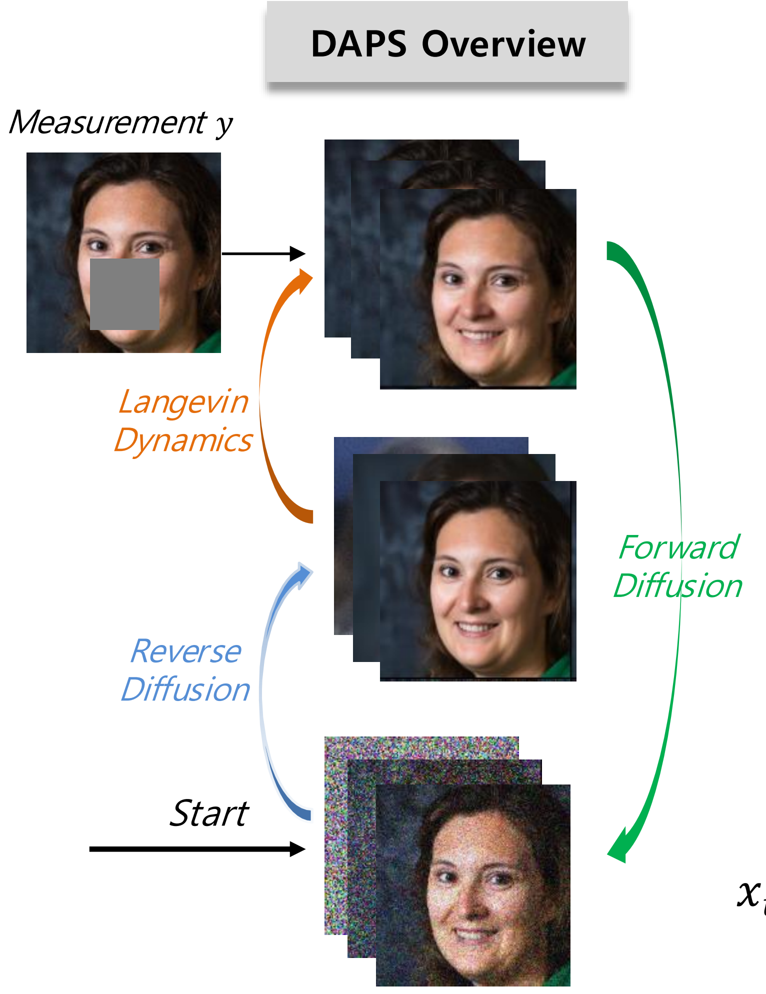
❖ **Proposed Method**

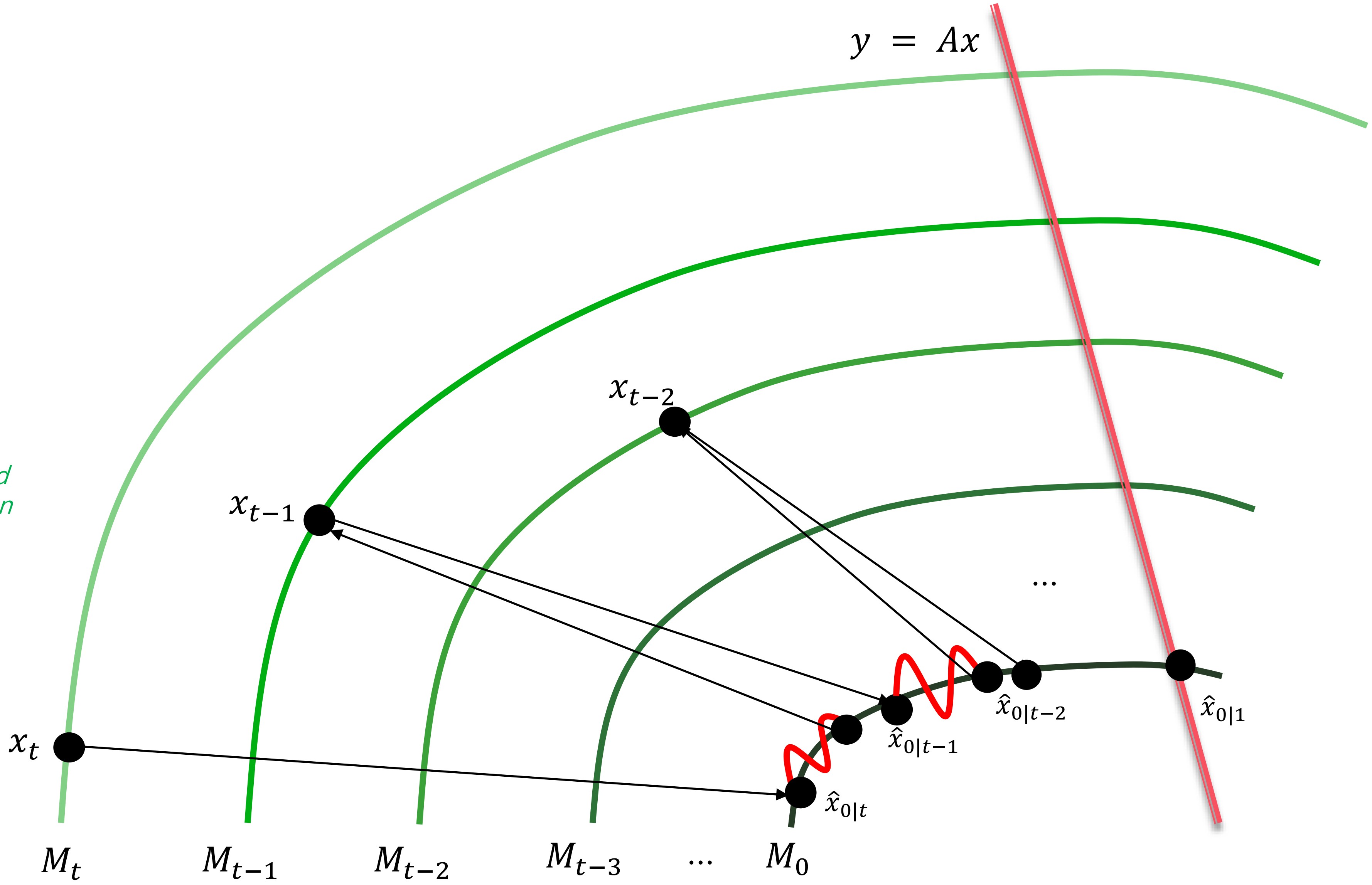




Decoupled Annealing Posterior Sampling (DAPS)

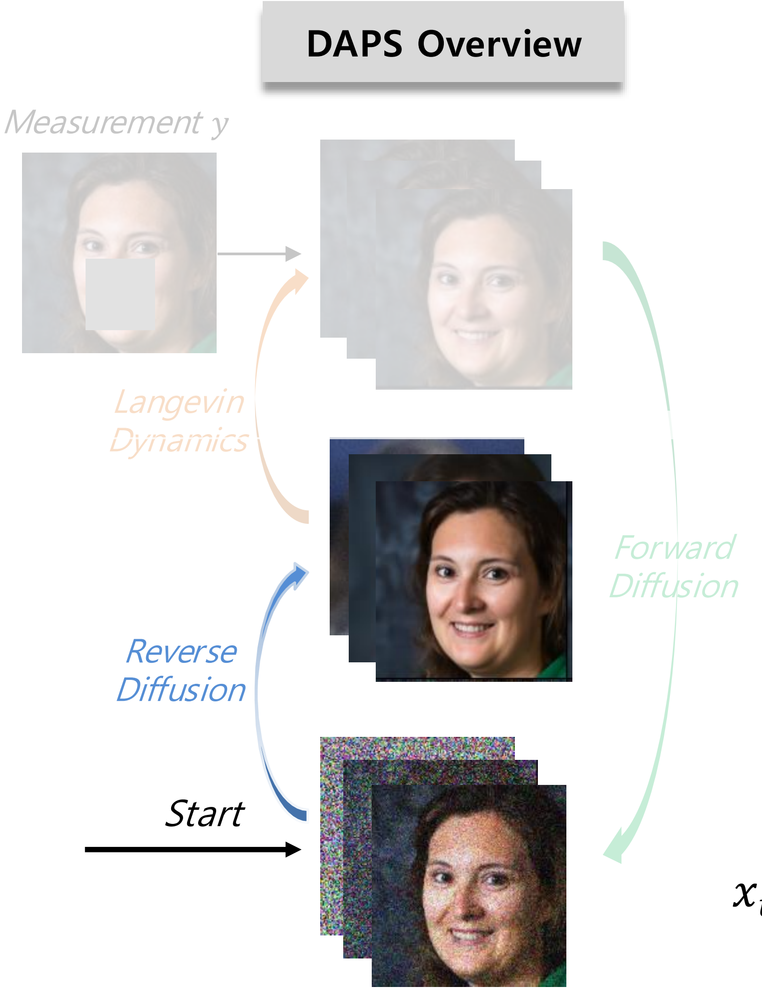
Diffusion Inverse Problem

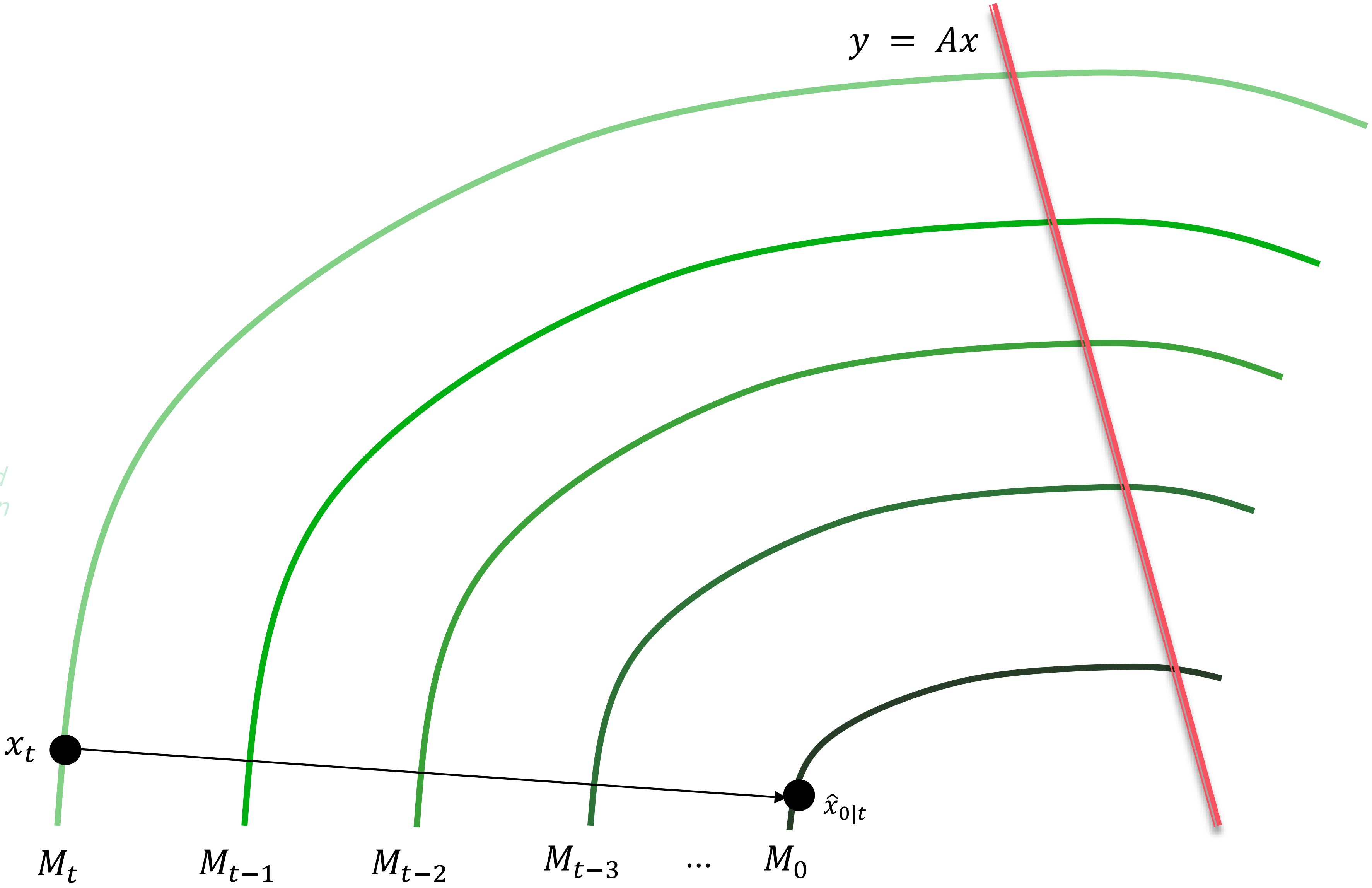




Decoupled Annealing Posterior Sampling (DAPS)

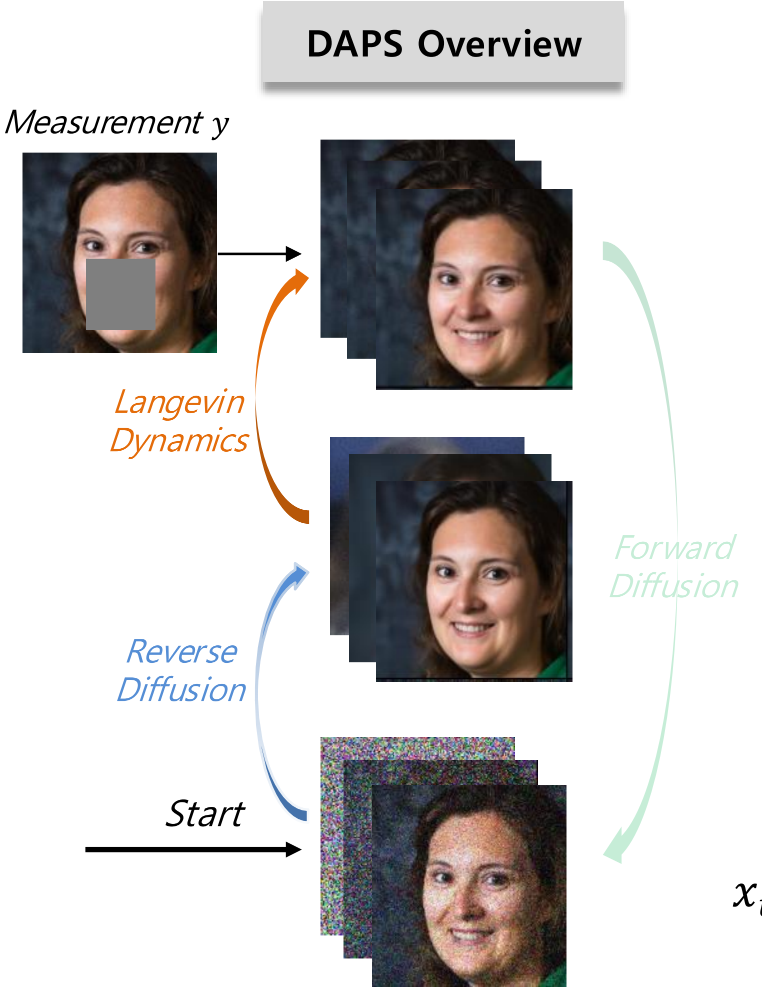
Diffusion Inverse Problem

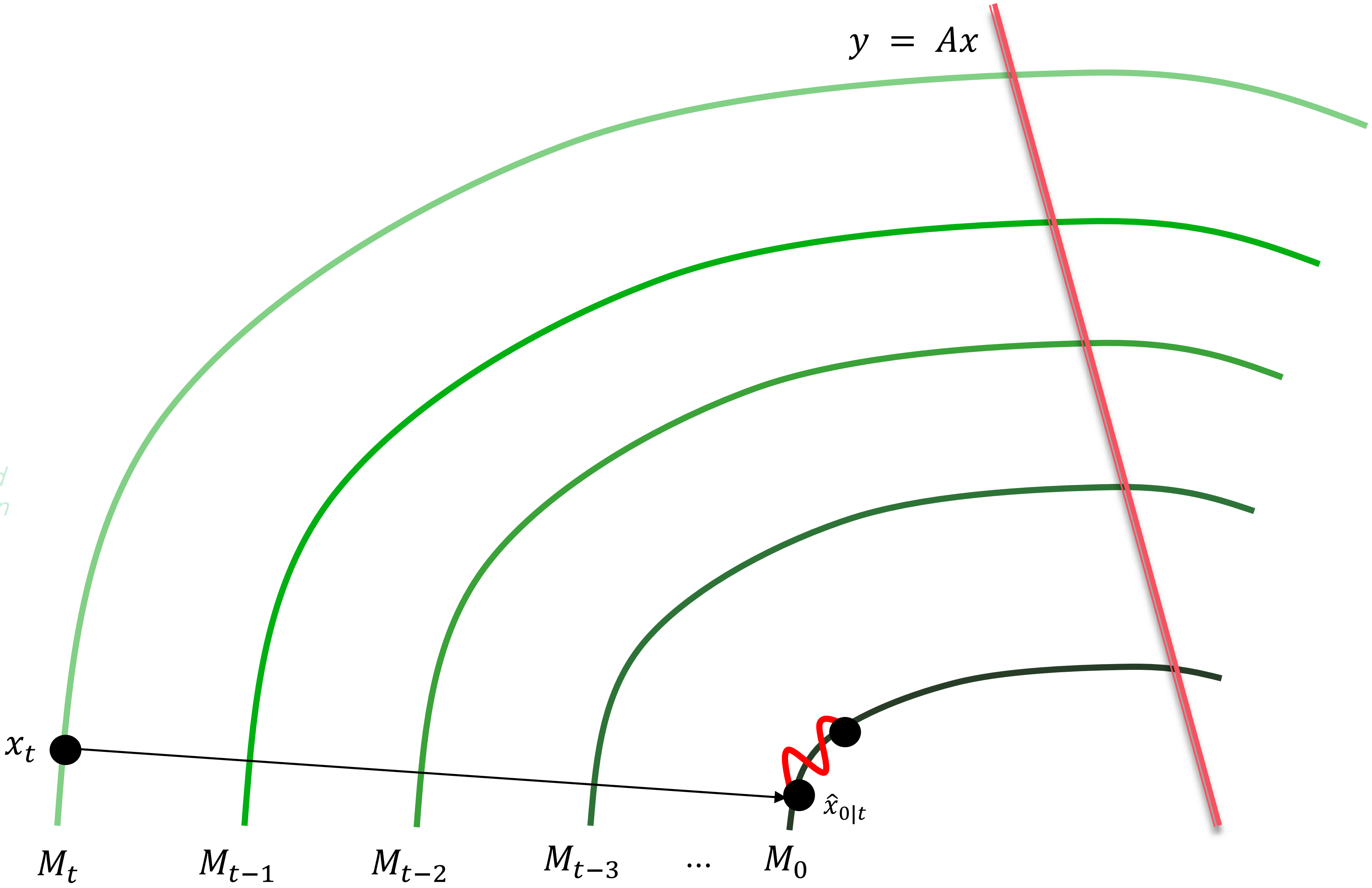




Decoupled Annealing Posterior Sampling (DAPS)

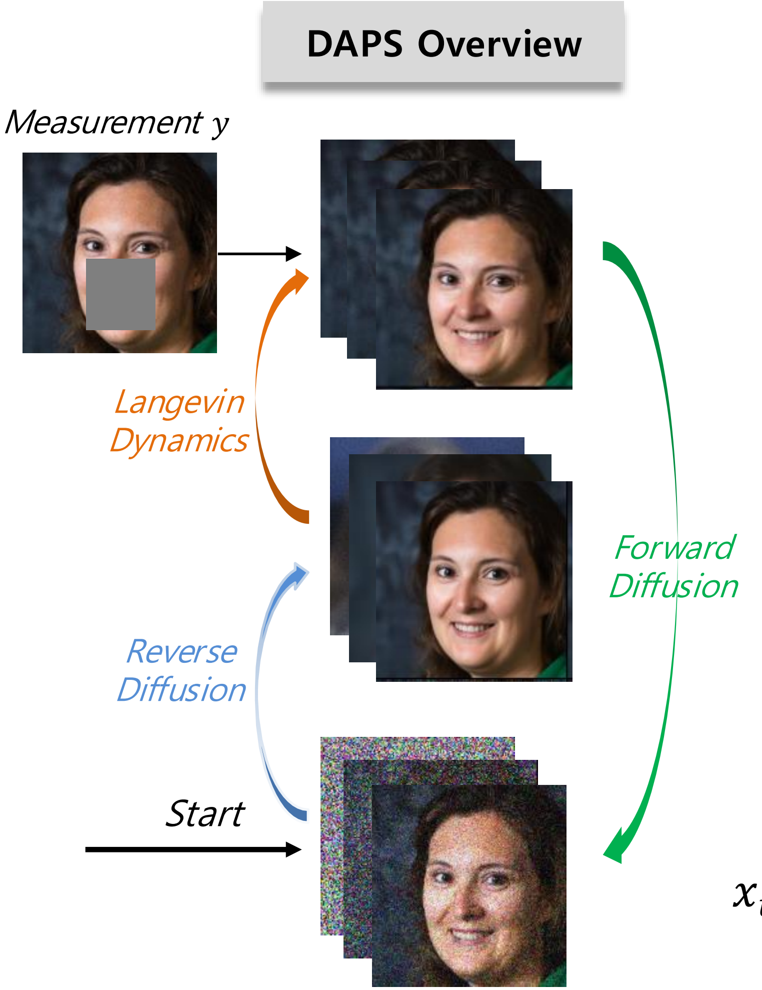
Diffusion Inverse Problem

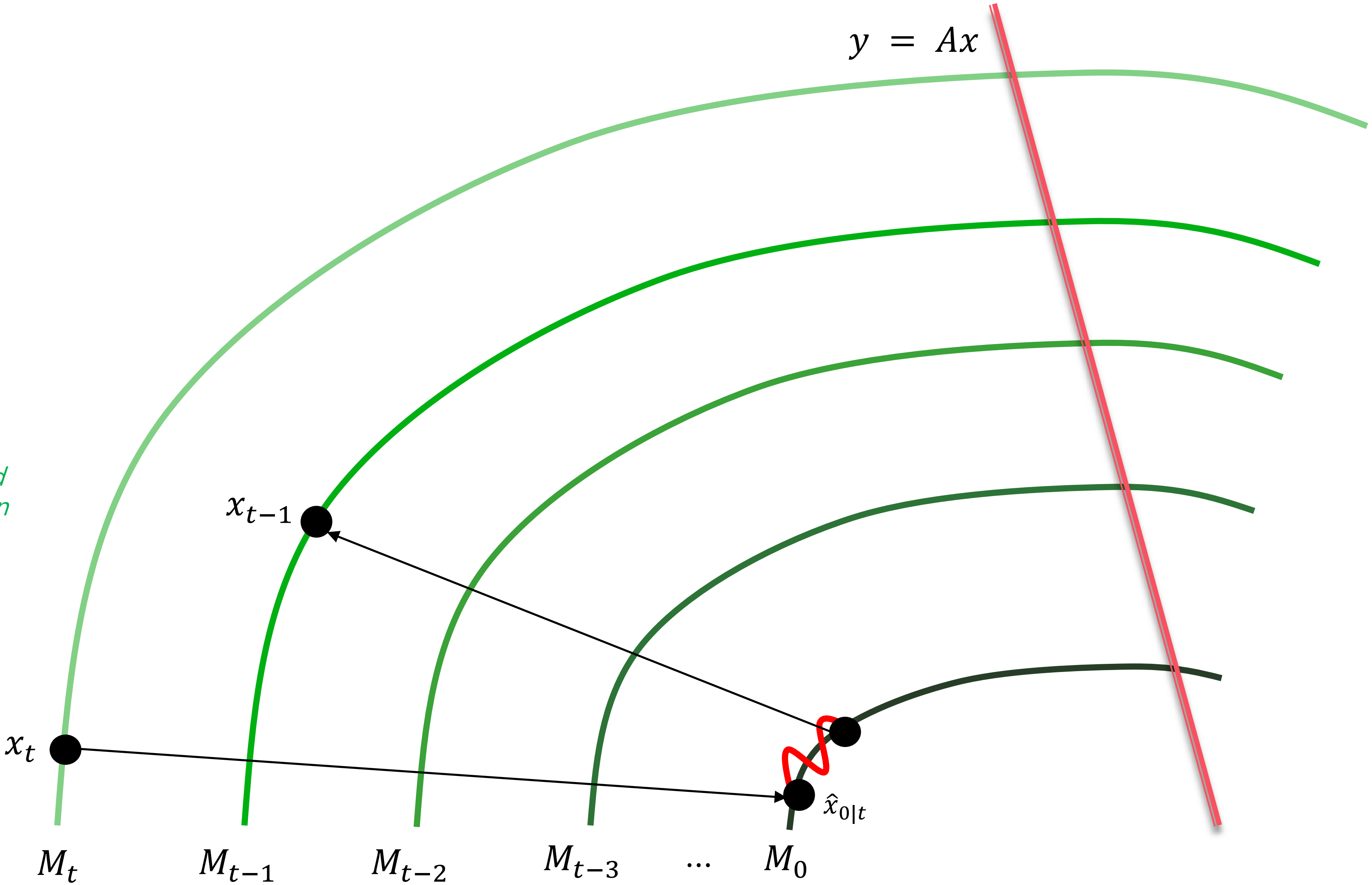




Decoupled Annealing Posterior Sampling (DAPS)

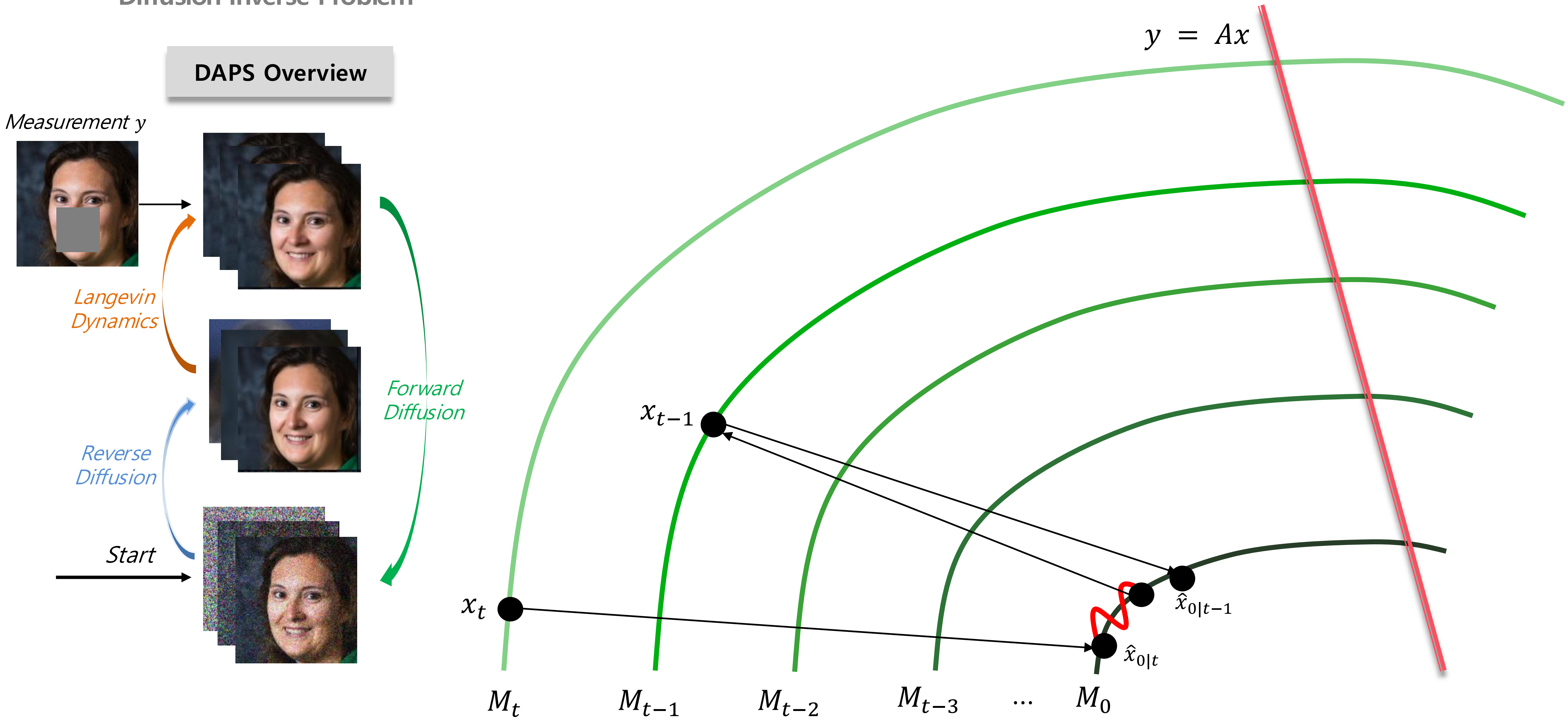
Diffusion Inverse Problem





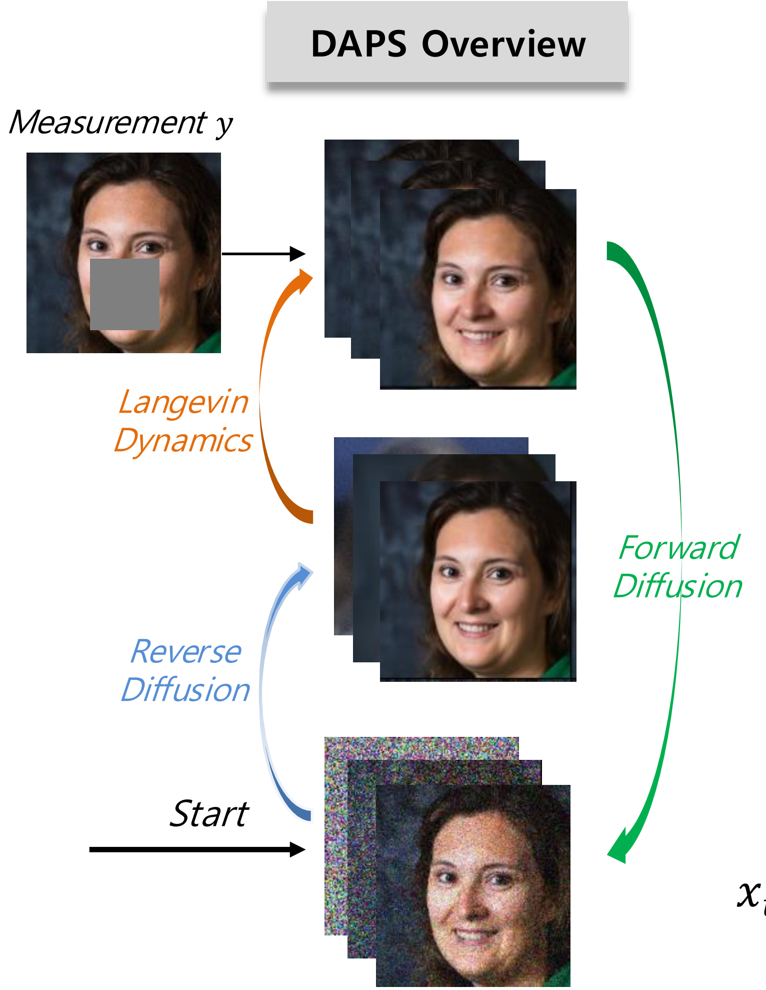
Decoupled Annealing Posterior Sampling (DAPS)

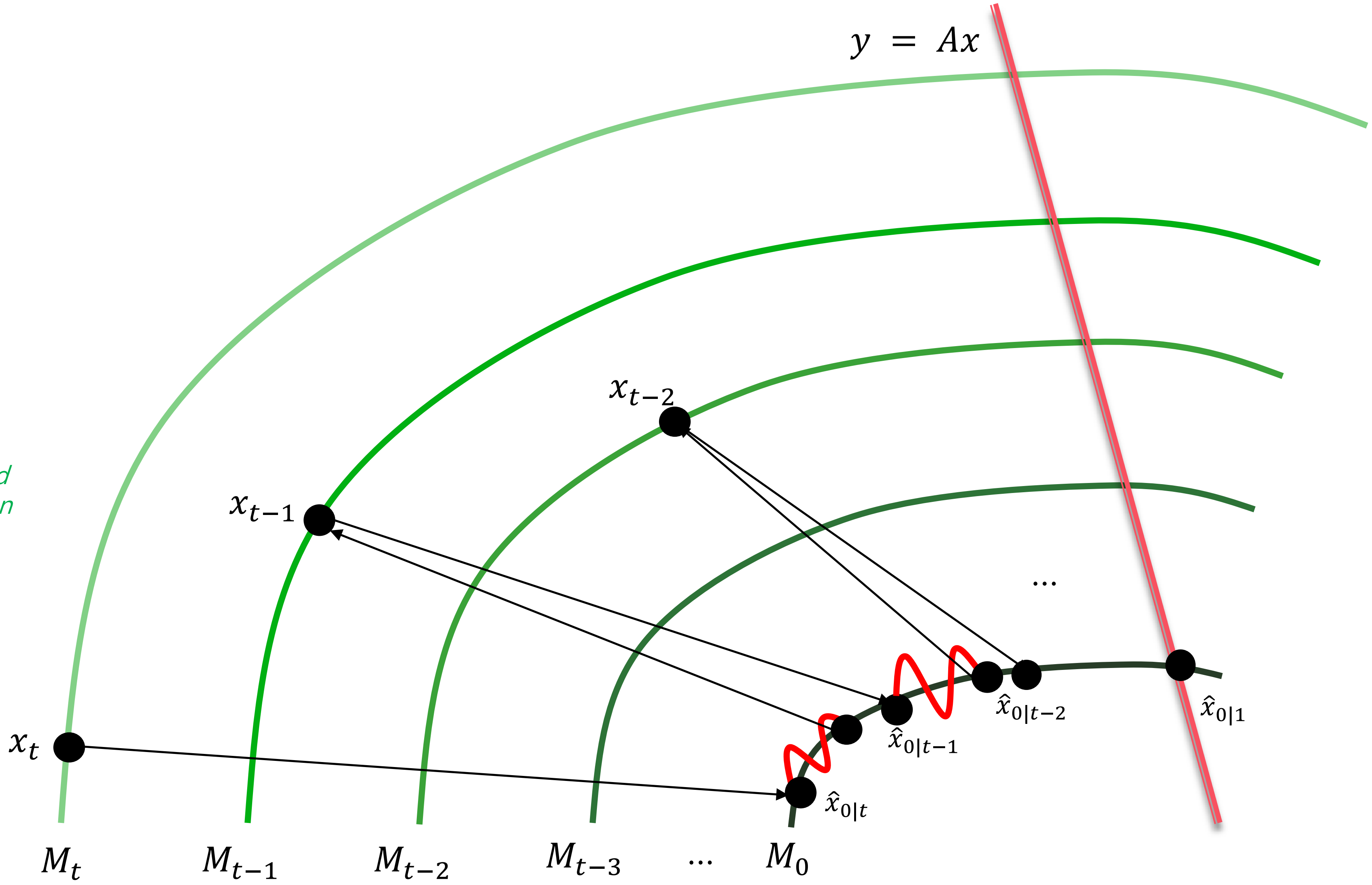
Diffusion Inverse Problem



Decoupled Annealing Posterior Sampling (DAPS)

Diffusion Inverse Problem

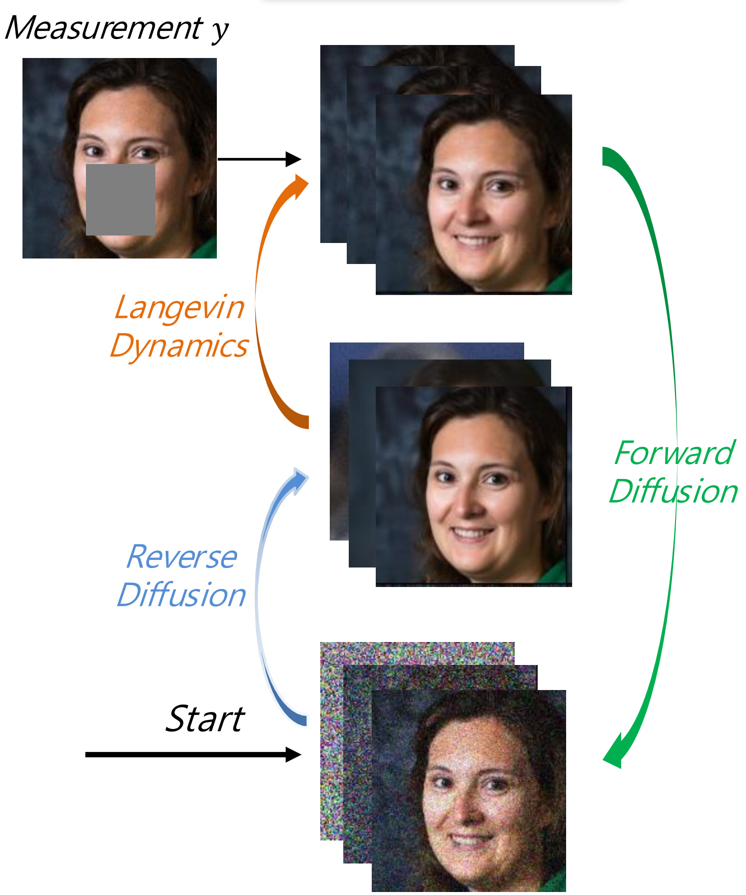




Decoupled Annealing Posterior Sampling (DAPS)

Diffusion Inverse Problem

DAPS Overview



Algorithm 1 Decoupled Annealing Posterior Sampling (DAPS)

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

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

for $i = N_A, N_A - 1, \dots, 1$ **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 $j = 0, \dots, N-1$ **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 $\mathbf{x}_{t_{i-1}} \sim \mathcal{N}(\hat{\mathbf{x}}_0^{(N)}, \sigma_{t_{i-1}}^2 \boldsymbol{I})$.

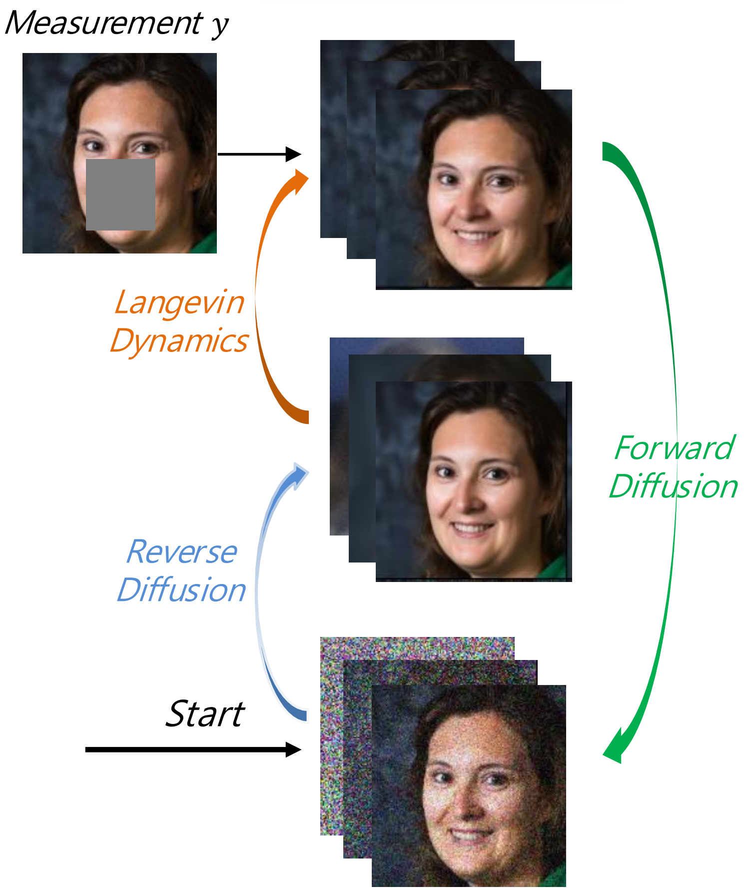
end for

Return $\mathbf{x}_0$

Decoupled Annealing Posterior Sampling (DAPS)

Diffusion Inverse Problem

DAPS Overview



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

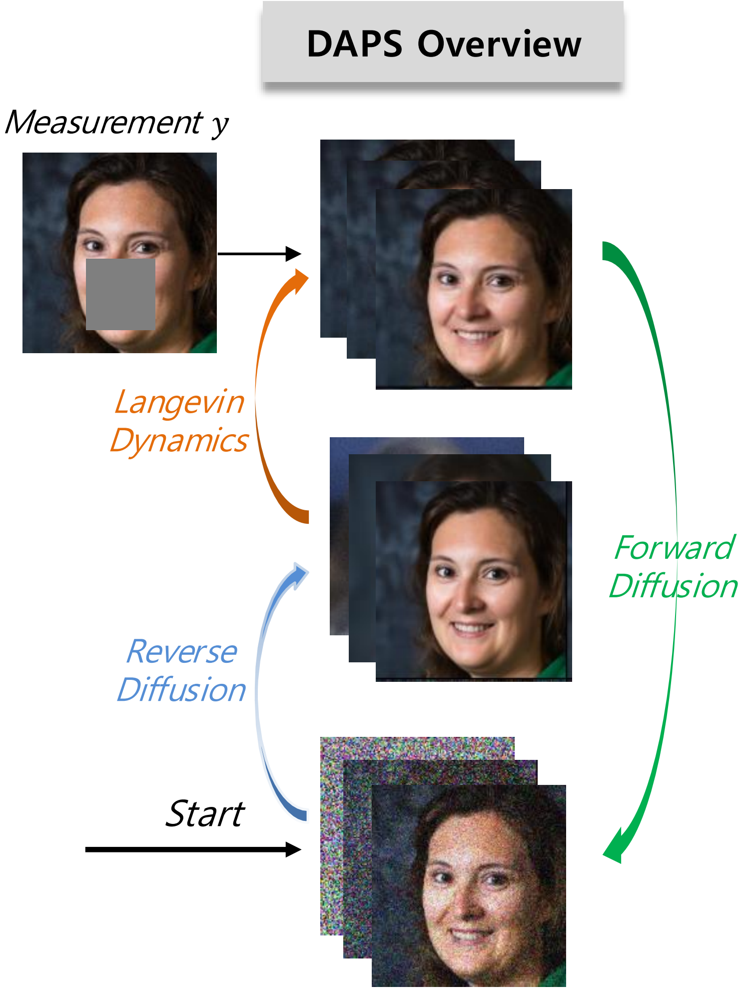
    Sample $\mathbf{x}_{t_{i-1}} \sim \mathcal{N}(\hat{\mathbf{x}}_0^{(N)}, \sigma_{t_{i-1}}^2 \boldsymbol{I})$.

  end for

  Return $\mathbf{x}_0$

Decoupled Annealing Posterior Sampling (DAPS)

Diffusion Inverse Problem



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

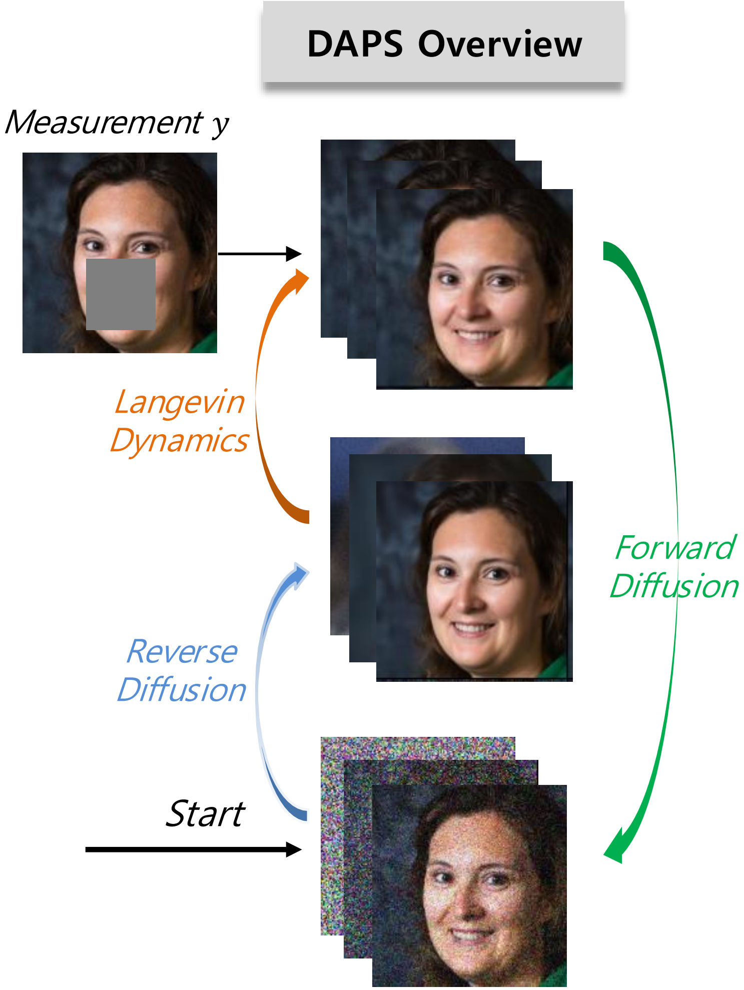
Sample $\mathbf{x}_{t_{i-1}} \sim \mathcal{N}(\hat{\mathbf{x}}_0^{(N)}, \sigma_{t_{i-1}}^2 \boldsymbol{I})$.

end for

Return $\mathbf{x}_0$

Decoupled Annealing Posterior Sampling (DAPS)

Diffusion Inverse Problem



Algorithm 1 Decoupled Annealing Posterior Sampling (DAPS)

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

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

for $i = N_A, N_A - 1, \dots, 1$ **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 $j = 0, \dots, N-1$ **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 $\mathbf{x}_{t_{i-1}} \sim \mathcal{N}(\hat{\mathbf{x}}_0^{(N)}, \sigma_{t_{i-1}}^2 \boldsymbol{I})$.

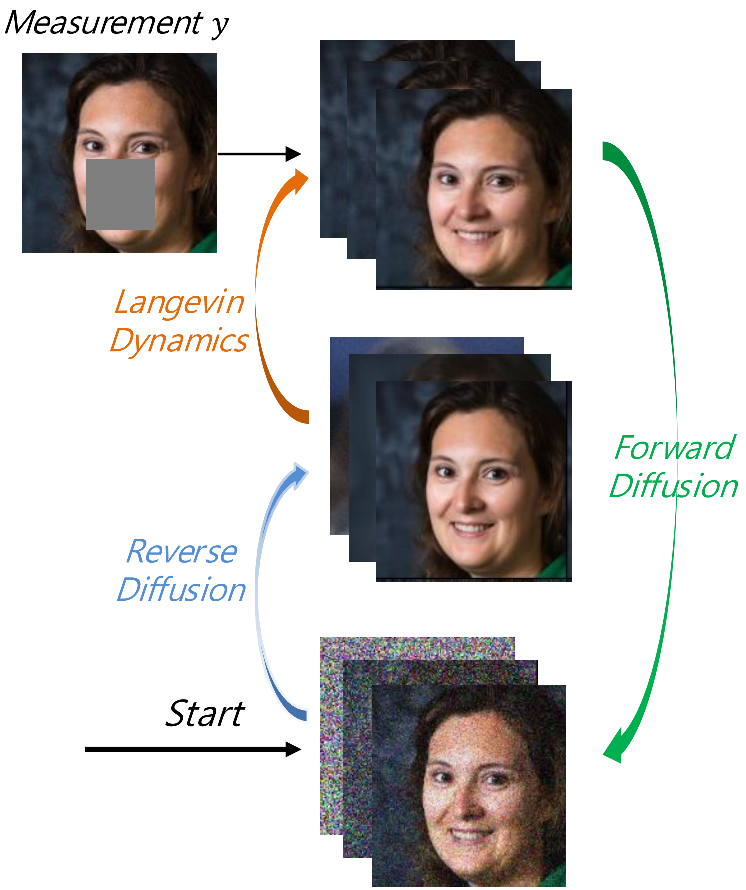
end for

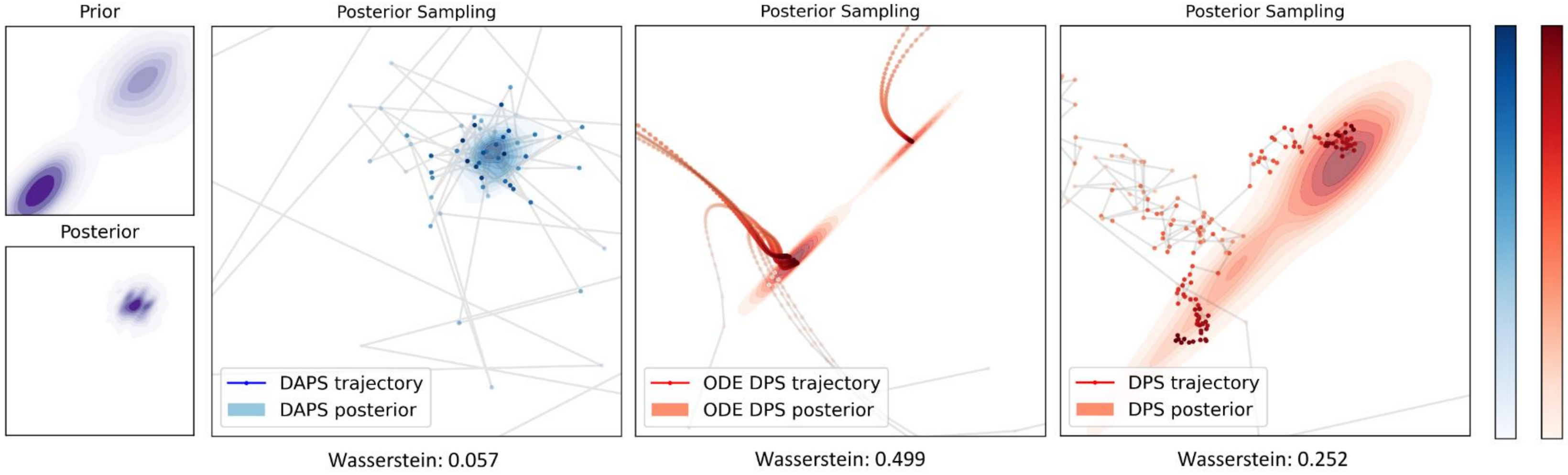
Return $\mathbf{x}_0$

Decoupled Annealing Posterior Sampling (DAPS)

Diffusion Inverse Problem

DAPS Overview





Decoupled Annealing Posterior Sampling (DAPS)

Diffusion Inverse Problem

❖ Experiments

- linear inverse problems (5 tasks) : super-resolution, gaussian deblurring, motion deblurring, inpainting (box), inpainting (random mask)
- nonlinear inverse problems (3 tasks) : phase retrieval, high dynamic range reconstruction, nonlinear deblurring

Decoupled Annealing Posterior Sampling (DAPS)

Diffusion Inverse Problem

❖ Experiments (linear inverse problem)

Task	Type	Method	FFHQ PSNR (↑)	FFHQ SSIM (↑)	FFHQ LPIPS (↓)	FFHQ FID (↓)	ImageNet PSNR (↑)	ImageNet SSIM (↑)	ImageNet LPIPS (↓)	ImageNet FID (↓)
Super resolution 4×	*Pixel*	DAPS (ours)	**29.07**	**0.818**	**0.177**	51.44	**25.89**	0.694	**0.276**	**83.57**
		DPS	25.86	0.753	0.269	81.07	21.13	0.489	0.361	106.32
		DDRM	26.58	0.782	0.282	79.25	22.62	0.521	0.324	103.85
		DDNM	28.03	0.795	0.197	64.62	23.96	0.604	0.475	98.62
		DCDP	28.66	0.807	0.178	53.81	-	-	-	-
		FPS-SMC	28.42	0.813	0.204	**49.25**	24.82	**0.703**	0.313	97.51
		DiffPIR	26.64	-	0.260	65.77	23.18	-	0.371	106.32
	Latent	LatentDAPS(ours)	**27.48**	**0.801**	**0.182**	59.62	25.06	0.673	**0.276**	**84.37**
		PSLD	24.35	0.649	0.287	74.36	**25.42**	**0.694**	0.360	97.45
		ReSample	23.29	0.594	0.392	93.18	22.61	0.576	0.370	113.42
Inpaint (box)	*Pixel*	DAPS(ours)	24.07	0.814	**0.133**	**43.10**	21.43	0.725	0.214	109.85
		DPS	22.51	0.792	0.209	61.27	18.94	0.722	0.257	126.52
		DDRM	22.26	0.801	0.207	78.62	18.63	0.733	0.254	116.37
		DDNM	24.47	**0.837**	0.235	46.59	21.64	**0.748**	0.319	**103.97**
		DCDP	23.89	0.760	0.163	45.23	-	-	-	-
		FPS-SMC	**24.86**	0.823	0.146	48.34	**22.16**	0.726	**0.208**	111.58
	Latent	LatentDAPS(ours)	23.99	0.802	0.194	46.52	17.19	0.624	0.340	145.63
		PSLD	**24.22**	**0.813**	**0.158**	**43.02**	**20.10**	**0.694**	0.465	146.53
		ReSample	20.06	0.749	0.184	53.21	18.29	0.631	**0.262**	**127.84**
Inpaint (random)	*Pixel*	DAPS(ours)	**31.12**	**0.844**	**0.098**	**32.17**	28.44	0.775	**0.135**	**54.25**
		DPS	25.46	0.823	0.203	69.20	23.52	0.745	0.297	87.53
		DDNM	29.91	0.817	0.121	44.37	**31.16**	**0.841**	0.191	63.84
		DCDP	30.69	0.842	0.142	52.51	-	-	-	-
		FPS-SMC	28.21	0.823	0.261	61.23	24.52	0.701	0.316	79.12
	Latent	LatentDAPS(ours)	**30.71**	**0.813**	0.141	**36.41**	27.59	0.772	0.164	61.62
		PSLD	30.31	0.809	0.221	47.21	**31.30**	**0.783**	0.337	83.21
		ReSample	29.61	0.746	**0.140**	39.85	27.50	0.756	**0.143**	**59.87**

Decoupled Annealing Posterior Sampling (DAPS)

Diffusion Inverse Problem

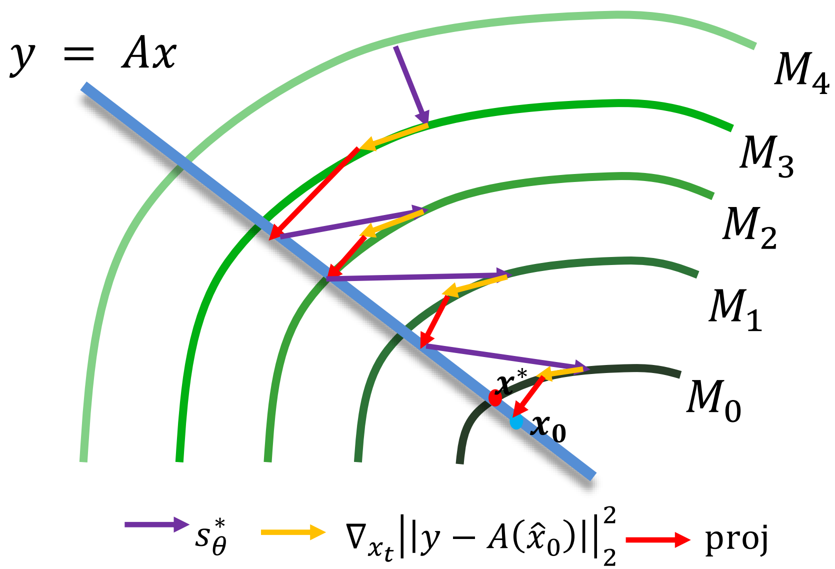
❖ **Experiments (nonlinear inverse problem)**

Task	Type	Method	FFHQ PSNR (↑)	FFHQ SSIM (↑)	FFHQ LPIPS (↓)	FFHQ FID (↓)	ImageNet PSNR (↑)	ImageNet SSIM (↑)	ImageNet LPIPS (↓)	ImageNet FID (↓)
Phase retrieval	*Pixel*	DAPS(ours)	**30.63**$_{\pm 3.13}$	**0.851**$_{\pm 0.072}$	**0.139**$_{\pm 0.060}$	**42.71**	**25.78**$_{\pm 6.92}$	**0.743**$_{\pm 0.084}$	**0.254**$_{\pm 0.125}$	**82.67**
		DPS	17.64$_{\pm 2.97}$	0.441$_{\pm 0.129}$	0.410$_{\pm 0.090}$	104.52	16.81$_{\pm 3.61}$	0.427$_{\pm 0.143}$	0.447$_{\pm 0.099}$	197.54
		RED-diff	15.60$_{\pm 4.48}$	0.398$_{\pm 0.195}$	0.596$_{\pm 0.092}$	167.43	14.98$_{\pm 3.75}$	0.386$_{\pm 0.057}$	0.536$_{\pm 0.129}$	212.24
		DCDP	28.65$_{\pm 8.09}$	0.781$_{\pm 0.217}$	0.203$_{\pm 0.196}$	68.13	-	-	-	-
	Latent	LatentDAPS(ours)	**29.16**$_{\pm 3.55}$	**0.796**$_{\pm 0.089}$	**0.199**$_{\pm 0.078}$	**54.26**	**20.54**$_{\pm 6.41}$	**0.612**$_{\pm 0.114}$	**0.361**$_{\pm 0.150}$	**129.54**
		ReSample	21.60$_{\pm 8.10}$	0.648$_{\pm 0.154}$	0.406$_{\pm 0.224}$	84.32	19.24$_{\pm 4.21}$	0.618$_{\pm 0.146}$	0.403$_{\pm 0.174}$	130.47
	Classical	HIO	13.53$_{\pm 2.50}$	0.359$_{\pm 0.093}$	0.726$_{\pm 0.068}$	268.09	-	-	-	-
Nonlinear deblur	*Pixel*	DAPS(ours)	28.29$_{\pm 1.77}$	0.783$_{\pm 0.036}$	**0.155**$_{\pm 0.032}$	49.38	27.73$_{\pm 3.23}$	0.724$_{\pm 0.048}$	**0.169**$_{\pm 0.056}$	59.87
		DPS	23.39$_{\pm 2.01}$	0.623$_{\pm 0.082}$	0.278$_{\pm 0.060}$	91.31	22.49$_{\pm 3.20}$	0.591$_{\pm 0.101}$	0.306$_{\pm 0.081}$	101.41
		RED-diff	**30.86**$_{\pm 0.51}$	**0.795**$_{\pm 0.028}$	0.160$_{\pm 0.034}$	**43.84**	**30.07**$_{\pm 1.41}$	**0.754**$_{\pm 0.023}$	0.211$_{\pm 0.083}$	**51.22**
		DCDP	27.92$_{\pm 2.64}$	0.779$_{\pm 0.067}$	0.183$_{\pm 0.051}$	51.96	-	-	-	-
	Latent	LatentDAPS(ours)	28.11$_{\pm 1.75}$	0.713$_{\pm 0.041}$	0.235$_{\pm 0.049}$	53.63	25.34$_{\pm 3.44}$	0.615$_{\pm 0.057}$	0.314$_{\pm 0.080}$	76.73
		ReSample	**28.24**$_{\pm 1.69}$	0.742$_{\pm 0.039}$	**0.185**$_{\pm 0.039}$	**51.62**	**26.20**$_{\pm 3.71}$	**0.653**$_{\pm 0.064}$	**0.206**$_{\pm 0.057}$	**61.16**
High dynamic range	*Pixel*	DAPS(ours)	**27.12**$_{\pm 3.53}$	**0.752**$_{\pm 0.041}$	**0.162**$_{\pm 0.072}$	**42.97**	**26.30**$_{\pm 4.10}$	**0.717**$_{\pm 0.067}$	**0.175**$_{\pm 0.107}$	**64.19**
		DPS	22.73$_{\pm 6.07}$	0.591$_{\pm 0.141}$	0.264$_{\pm 0.156}$	112.82	19.23$_{\pm 2.52}$	0.582$_{\pm 0.082}$	0.503$_{\pm 0.106}$	146.23
		RED-diff	22.16$_{\pm 3.41}$	0.512$_{\pm 0.083}$	0.258$_{\pm 0.089}$	108.32	22.03$_{\pm 5.90}$	0.601$_{\pm 0.094}$	0.274$_{\pm 0.198}$	113.48
	Latent	LatentDAPS(ours)	**25.94**$_{\pm 2.87}$	**0.751**$_{\pm 0.056}$	0.223$_{\pm 0.080}$	74.83	23.64$_{\pm 4.10}$	0.609$_{\pm 0.053}$	0.269$_{\pm 0.099}$	93.51
		ReSample	25.65$_{\pm 3.57}$	0.732$_{\pm 0.059}$	**0.182**$_{\pm 0.085}$	**67.22**	**25.11**$_{\pm 4.21}$	**0.633**$_{\pm 0.049}$	**0.198**$_{\pm 0.089}$	**87.66**

Conclusion

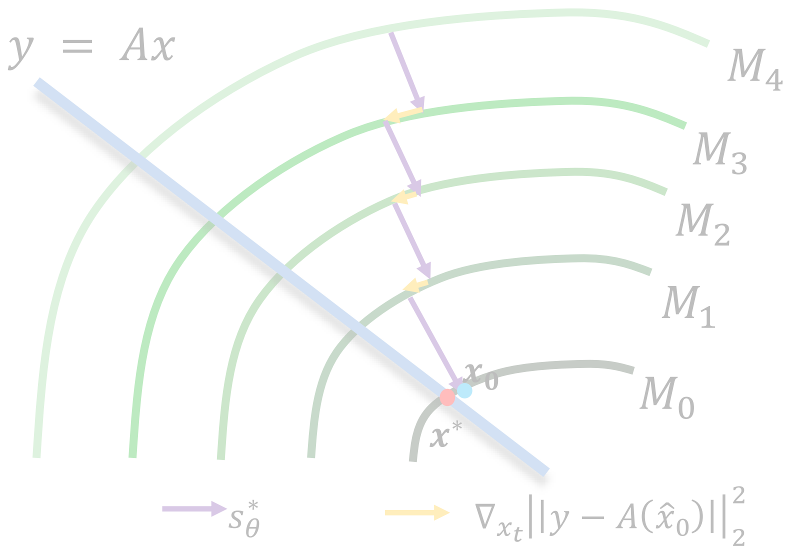
Diffusion Inverse Problem

MCG

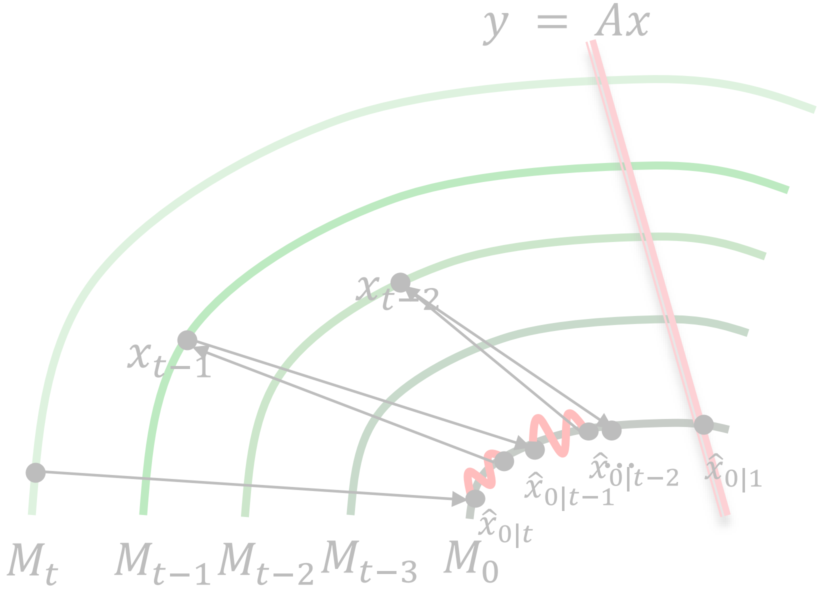


Manifold의 접평면 방향으로 gradient update를 하자!

DPS



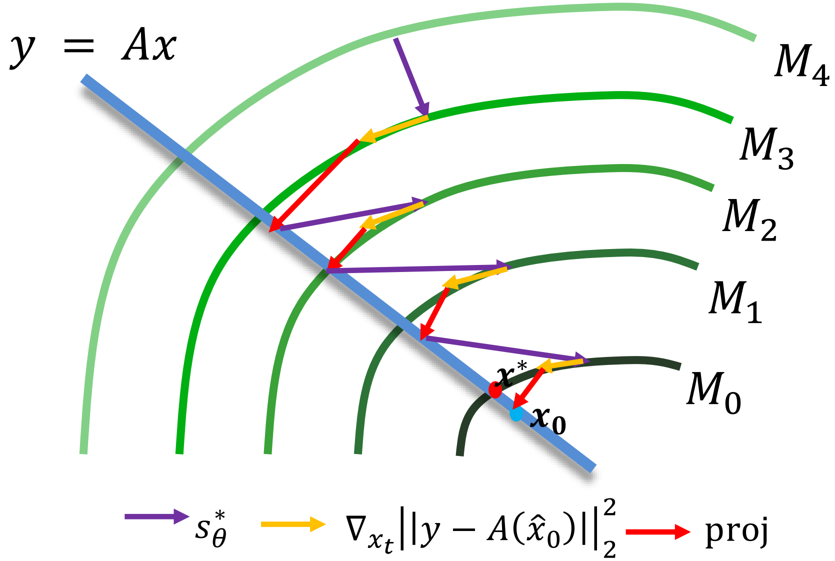
DAPS



Conclusion

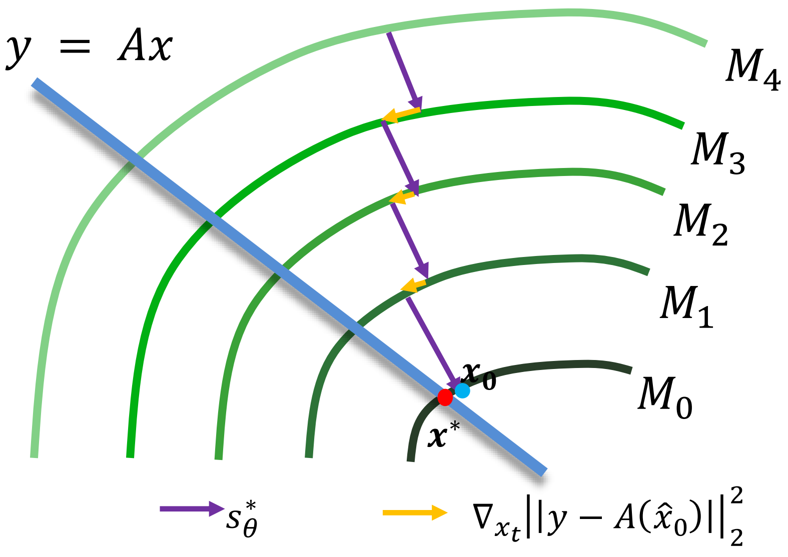
Diffusion Inverse Problem

MCG



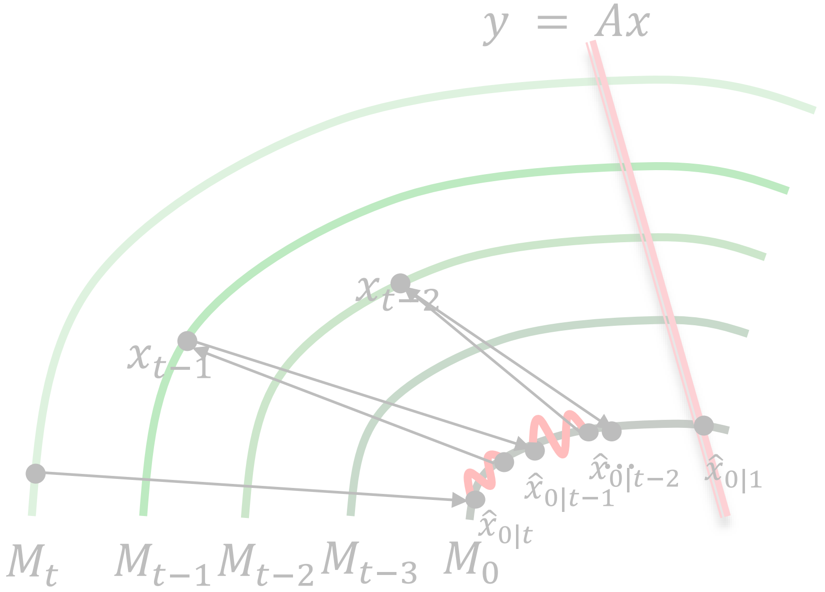
Manifold의 접평면 방향으로
gradient update를 하자!

DPS



Ax로의 projection을 없애서
manifold 바깥으로 가지 않게 하자!

DAPS

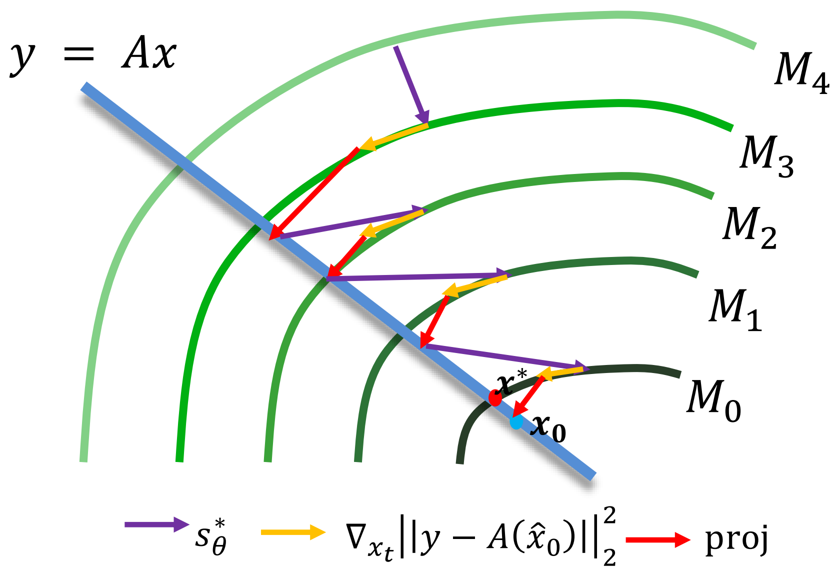


이전 상태의 종속성을 끊어내 global correction
을 유도하자!

Conclusion

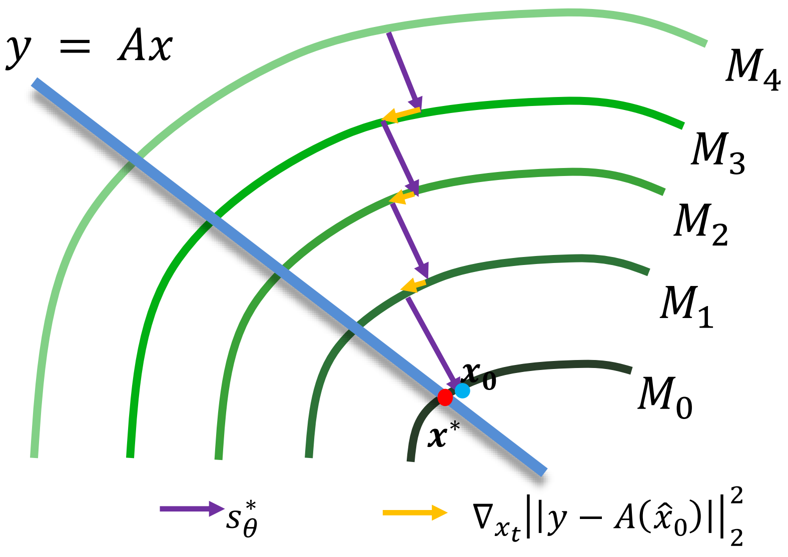
Diffusion Inverse Problem

MCG



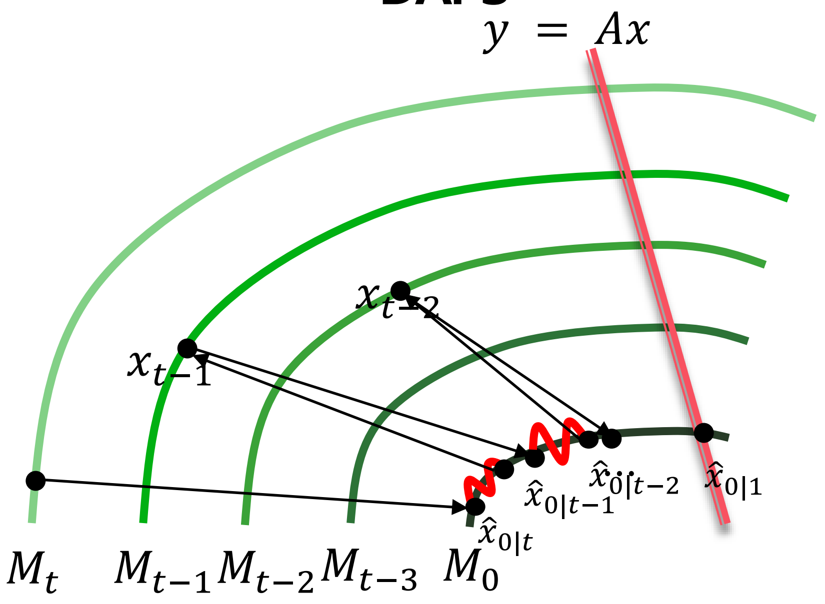
Manifold의 접평면 방향으로
gradient update를 하자!

DPS



Ax로의 projection을 없애서
manifold 바깥으로 가지 않게 하자!

DAPS



이전 상태의 종속성을 끊어내 global correction
을 유도하자!

고맙습니다