# Score-based + Annealing Langevi Dynamics.

* Introduction.

likelihood-based 와 GAN은 성공적이지만 본질적인 문제가 있다.

likelihood는 surrogate loss를 쓰거나. flow-based의 경우 reversable 한 archi가 필요. (or Autoregressive)

GAN은 unstable. model compare가 힘듦 (직관적인 data 분포의 NLL과 달리 낮은 Loss가 훈련 보장 x)

논문은 pdf의 gradient를 score ($\nabla \log p_\theta(x)$)로 정의. model은 score를 추정하고,

이를 바탕으로 $p_\theta(x)$가 높은 곳으로 이동.

Challenge.

1. low-dimension의 dataset이 몰려 있어서, 임의의 space에서 score 추정이 힘듦.

2. 따라서 noise부터 시작하는 sampling 시에, 정확하지 않음.

↳ puturbated noise를 제안

이 방식은 tractable 하고, objective function은 model 비교에 사용가능.

* Model

pdf : $P(x)$

score : $\nabla_x \log P(x)$

network: $S_\theta(x)$ → $P_{data}$에 대한 score 학습

* Score Matching

$\theta^* = \arg\min_\theta E_{P(x)}\left[\frac{1}{2}\|\nabla_x \log p(x) - S_\theta(x)\|_2^2\right]$

↓ $P(x)$계산 x, score matching

$\theta^* = \arg\min_\theta E_{P(x)}\left[\frac{1}{2}\|S_\theta(x)\|_2^2 + tr(\nabla_x S_\theta(x))\right]$

↳ Jacobian은 이미지나 커질수록 계산량↑

- Denoising Score Matching

perturbated된 score가 같다.

$\theta^* = \arg\min_\theta E_{P(x)}\left[\frac{1}{2}\|S_\theta(x)\|_2^2 + tr(\nabla_x S_\theta(x))\right]$

↓ Denoising Score Matching

$\frac{1}{2}E_{q_\sigma(\tilde{x}|x)P_{data}(x)}\left[\|S_\theta(\tilde{x}) - \nabla_{\tilde{x}} \log q_\sigma(\tilde{x}|x)\|_2^2\right]$

↳ Gaussian 가정 : $\frac{1}{\sigma^2}(\tilde{x}-x)^\top \Sigma^{-1}(\tilde{x}-x)$

- Sliced score Matching

projection 방법

$$\mathbb{E}_{p_{\mathbf{v}}}\mathbb{E}_{p_{\text{data}}}\left[\mathbf{v}^\intercal \nabla_{\mathbf{x}} \mathbf{s}_\theta(\mathbf{x})\mathbf{v} + \frac{1}{2}\|\mathbf{s}_\theta(\mathbf{x})\|_2^2\right],$$

↳ DSM보다 계산량 많아져 안씀.

* Sampling with Langevin dynamics

score 만으로 p(x)로부터 sample 하는 방법

$\tilde{x}_t = \tilde{x}_{t-1} + \frac{\epsilon}{2}\nabla_x \log p(\tilde{x}_{t-1}) + \sqrt{\epsilon} z_t$

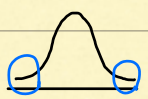
* Challenges of Score-based GM.

* Manifold hypothesis

실제 data는 high-dimension ambient space에 있다

문제점.

1. ambient 에 있다보니 low dimension에서는 score 계산이 힘듦.



2. score matching은 data가 전체 분포여야 가능.

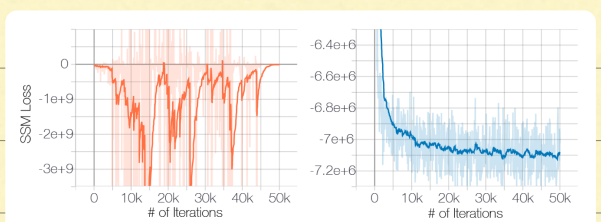



Figure 1: **Left**: Sliced score matching (SSM) loss w.r.t. iterations. No noise is added to data. **Right**: Same but data are perturbed with $\mathcal{N}(0, 0.0001)$.

따라서 사람이 알아보기 힘든 작은 noise로 범위 확장해서 훈련.

* Low data density regions

Dataset 이 부족할 수 있다.

density가 낮은 곳은 학습이 잘 안됨.

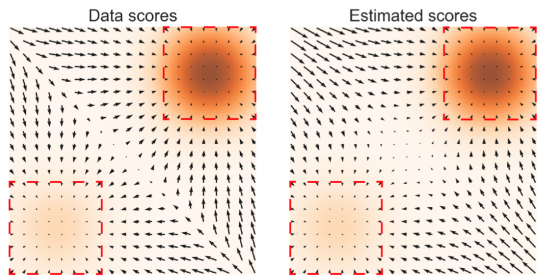



Figure 2: **Left**: $\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$; **Right**: $\mathbf{s}_\theta(\mathbf{x})$. The data density $p_{\text{data}}(\mathbf{x})$ is encoded using an orange colormap: darker color implies higher density. Red rectangles highlight regions where $\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) \approx \mathbf{s}_\theta(\mathbf{x})$.

- Slow Mixing of Langevin dynamics

따라서 두 분포 사이를 잘 modeling 못하고, 상대적 분포는 고려x.

$p_{data}(x) = \pi p_1(x) + (1-\pi) p_2(x)$ 일때, score는 $\pi$에 영향을 안받아서, 두 분포의 상대밀도 추정이 안됨.

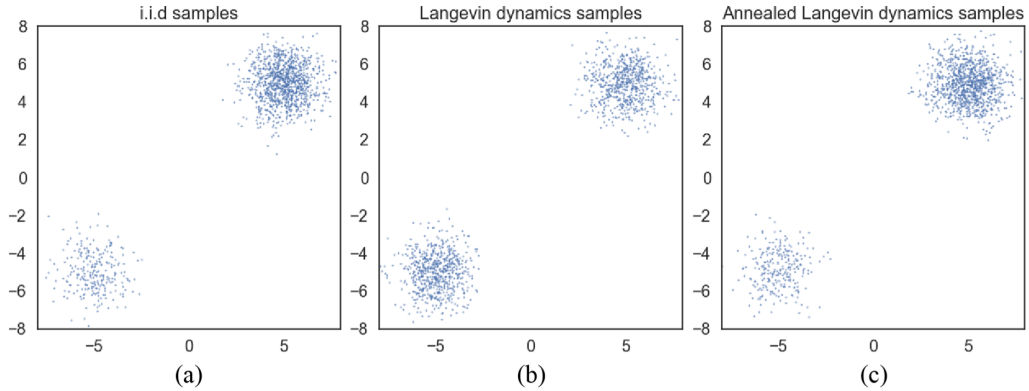



Figure 3: Samples from a mixture of Gaussian with different methods. (a) Exact sampling. (b) Sampling using Langevin dynamics with the exact scores. (c) Sampling using annealed Langevin dynamics with the exact scores. Clearly Langevin dynamics estimate the relative weights between the two modes incorrectly, while annealed Langevin dynamics recover the relative weights faithfully.

* Noise condition Score Network : learning & Infer.

Gaussian은 whole space 에서 데이터의 low-dimension manifold에 국한됨. → score 추정 ↑

" origin data에서 low-density 영역을 지나는 경향이 있다.

∴ 다양한 noise-level에 대해 modeling.

Inf 시 Langevin 시 . 초기에는 큰 noise level로.

* NCSN

Score Net에 noise를 condition으로 줌.

t가 지날수록 $\sigma_t$가 작아짐. $\frac{\sigma_t}{\sigma_{t+1}} > 1$

$s_\theta(x,\sigma) \approx \nabla_x \log q_\sigma(x)$

Network는 일단 segmentation에 유명한 UNet 사용

* Learning NCSNs

$q_\sigma(\tilde{x}|x) = \mathcal{N}(\tilde{x}|x, \sigma^2 I)$

$\nabla_{\tilde{x}} \log q_\sigma(\tilde{x}|x) = -\frac{1}{\sigma^2}(\tilde{x}-x)$

DSM : $$\ell(\theta;\sigma) \triangleq \frac{1}{2}\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\mathbb{E}_{\tilde{\mathbf{x}}\sim\mathcal{N}(\mathbf{x},\sigma^2 I)}\left[\left\|\mathbf{s}_\theta(\tilde{\mathbf{x}},\sigma)+\frac{\tilde{\mathbf{x}}-\mathbf{x}}{\sigma^2}\right\|_2^2\right].$$

$$\mathcal{L}(\theta;\{\sigma_i\}_{i=1}^L) \triangleq \frac{1}{L}\sum_{i=1}^{L}\lambda(\sigma_i)\ell(\theta;\sigma_i),$$

경험적으로, $\|s_\theta(x,\sigma)\| \propto 1/\sigma$ 일 때 성능 ↑. ∴ $\lambda(\sigma) = \sigma^2$

- NCSN inference via annealed Langevin Dynamics

**Algorithm 1** Annealed Langevin dynamics.

**Require:** $\{\sigma_i\}_{i=1}^{L}, \epsilon, T$.

Initialize $\tilde{\mathbf{x}}_0$
**for** $i \leftarrow 1$ to $L$ **do**
  $\alpha_i \leftarrow \epsilon \cdot \sigma_i^2/\sigma_L^2$ ▷ $\alpha_i$ is the step size.
  **for** $t \leftarrow 1$ to $T$ **do**
    Draw $\mathbf{z}_t \sim \mathcal{N}(0, I)$
    $\tilde{\mathbf{x}}_t \leftarrow \tilde{\mathbf{x}}_{t-1} + \frac{\alpha_i}{2}\mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}_{t-1}, \sigma_i) + \sqrt{\alpha_i}\,\mathbf{z}_t$
  **end for**
  $\tilde{\mathbf{x}}_0 \leftarrow \tilde{\mathbf{x}}_T$
**end for**
**return** $\tilde{\mathbf{x}}_T$

* Experiments

| Model | Inception | FID |
|---|---|---|
| **CIFAR-10 Unconditional** | | |
| PixelCNN [59] | 4.60 | 65.93 |
| PixelIQN [42] | 5.29 | 49.46 |
| EBM [12] | 6.02 | 40.58 |
| WGAN-GP [18] | $7.86 \pm .07$ | 36.4 |
| MoLM [45] | $7.90 \pm .10$ | **18.9** |
| SNGAN [36] | $8.22 \pm .05$ | 21.7 |
| ProgressiveGAN [25] | $8.80 \pm .05$ | - |
| **NCSN (Ours)** | $\mathbf{8.87} \pm .12$ | 25.32 |
| **CIFAR-10 Conditional** | | |
| EBM [12] | 8.30 | 37.9 |
| SNGAN [36] | $8.60 \pm .08$ | 25.5 |
| BigGAN [6] | **9.22** | **14.73** |

Table 1: Inception and FID scores for CIFAR-10

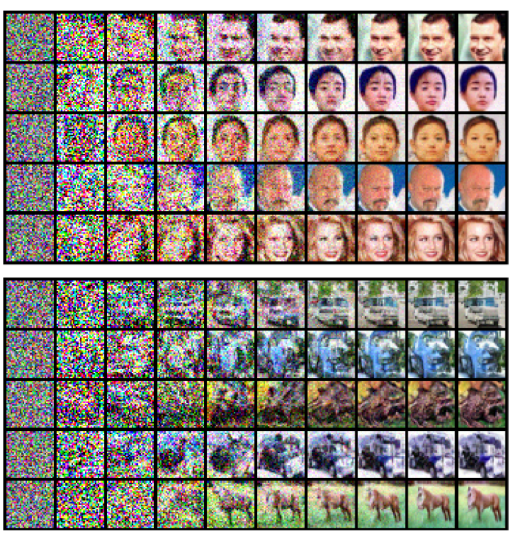


Figure 4: Intermediate samples of annealed Langevin dynamics.

(a) MNIST (b) CelebA (c) CIFAR-10

Figure 5: Uncurated samples on MNIST, CelebA, and CIFAR-10 datasets.

Figure 6: Image inpainting on CelebA (**left**) and CIFAR-10 (**right**). The leftmost column of each figure shows the occluded images, while the rightmost column shows the original images.

**Algorithm 2** Inpainting with annealed Langevin dynamics.

**Require:** $\{\sigma_i\}_{i=1}^{L}, \epsilon, T$ ▷ $\epsilon$ is smallest step size; $T$ is the number of iteration for each noise level.
**Require:** $\mathbf{m}, \mathbf{x}$ ▷ $\mathbf{m}$ is a mask to indicate regions not occluded; $\mathbf{x}$ is the given image.

Initialize $\tilde{\mathbf{x}}_0$
**for** $i \leftarrow 1$ to $L$ **do**
  $\alpha_i \leftarrow \epsilon \cdot \sigma_i^2 / \sigma_L^2$ ▷ $\alpha_i$ is the step size.
  Draw $\tilde{\mathbf{z}} \sim \mathcal{N}(0, \sigma_i^2)$
  $\mathbf{y} \leftarrow \mathbf{x} + \tilde{\mathbf{z}}$
  **for** $t \leftarrow 1$ to $T$ **do**
    Draw $\mathbf{z}_t \sim \mathcal{N}(0, I)$
    $\tilde{\mathbf{x}}_t \leftarrow \tilde{\mathbf{x}}_{t-1} + \frac{\alpha_i}{2}\mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}_{t-1}, \sigma_i) + \sqrt{\alpha_i}\,\mathbf{z}_t$
    $\tilde{\mathbf{x}}_t \leftarrow \tilde{\mathbf{x}}_t \odot (1 - \mathbf{m}) + \mathbf{y} \odot \mathbf{m}$
  **end for**
  $\tilde{\mathbf{x}}_0 \leftarrow \tilde{\mathbf{x}}_T$
**end for**
**return** $\tilde{\mathbf{x}}_T$

# * NCSN ++

## * Introduction

NCSN은 score estim으로 Langevin 통해 noise를 제거한다.

DDPM은 noise의 역 process를 modeling한다.

Continous의 경우 DDPM의 loss는 각 noise scale에서 score를 implicit compute 한다.

Score-based의 확장을 위해, SDE를 통해, 위의 방법을 통합

유한한 data에서 perturbing 하는 대신,

Diffusion process에 따라, 시간이 지나면서 발전하는 continuum of distribution을 제안한다.

↳ data를 random noise로 diffusion

↳ 이 과정은 trainable 한 param이 없고 data에 의존하지 않는, 규정된 SDE에 의해 주어짐.

Reverse process에서, noise → sample 가능.

↳ Reverse-time SDE를 만족하고, 시간의 함수로, marginal probability density의 score가

주어지면, forward SDE에서 유도이 가능.

∴ Reverse-time SDE를 score에 대한 t조건의 NN을 훈련하여 근사가능.

↳ numerical SDE solver로 sample 생성

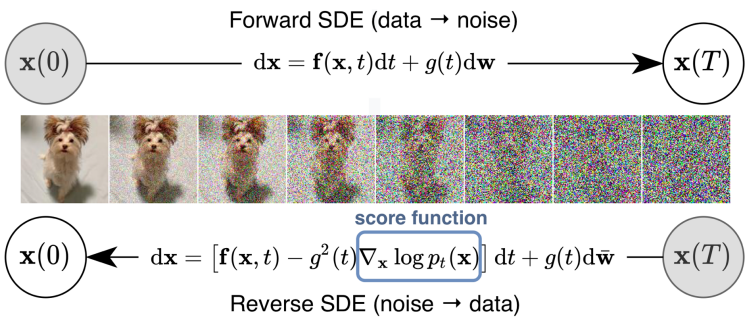



Figure 1: **Solving a reverse-time SDE yields a score-based generative model.** Transforming data to a simple noise distribution can be accomplished with a continuous-time SDE. This SDE can be reversed if we know the score of the distribution at each intermediate time step, $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$.

* contribution.

1. Flexible sampling & likelihood compute.

reverse-time SDE의 general-purpose SDE solver를 다 사용가능.

추가로

1. Predictor-Corrector(PC) : SDE-solver 와 score-based MCMC 결합.

↳ 기존 방법 통합.

2. ODE-based deterministic sampler

↳ black-box ODE solver를 통한 매우 adaptive sampling.

↳ latent를 통한 flexible manipulation.

↳ likelihood compute

↳ unique identifiable encoding

2. controllable generation

Conditional reverse-time SDE는 unconditional score에서 estim 가능하기 때문에

훈련 중에 가능하지 않았던 condition information으로 generation process 조정 가능.

↳ inpainting, class condition 등은 모두 재훈련 없이 생성가능.

3. Unified framework

여러 score-based model을 unify

* Background

* Denoising Score Matching with Langevin Dynamics (SMLD)

  = NCSN

* DDPM

* Score-based generative modeling with SDEs

Multi-level noise의 perturbating이 핵심이다.

논문에서는 noise scale을 infinite로 generalize 하여,

perturbated data의 distribution이 SDE에 따라 update 되도록 한다.

* Perturbing data with SDEs

$x(0) \sim p_0$, $x(T) \sim p_T$ 이 $t \in [0,1]$ 인 diffusion process $\{x(t)\}_{t=0}^{T}$

- $x(0)$에서 tractable.
- continuous

Diffusion process는 SDE로 modeling 가능.

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

- ↳ $w$: standard Wiener process (유리 돌에서 다음 입자가 움직이는 모양
- ↳ $f(\cdot,t): \mathbb{R}^d \to \mathbb{R}^d$, drift coefficient of $x(t)$
- ↳ $g(\cdot): \mathbb{R} \to \mathbb{R}$, diffusion coefficient of $x(t)$

coefficient가 state와 time에 대해 Lipschitz 가면, SDE는 unique strong solution.

$x(t)$의 확률은 $p_t(x)$. $p_{st}(x(t)|x(s))$, $0 \le s < t \le 1$

Distribution에서 고정된 사전 분포로 확산 $x(0) \to x(T)$

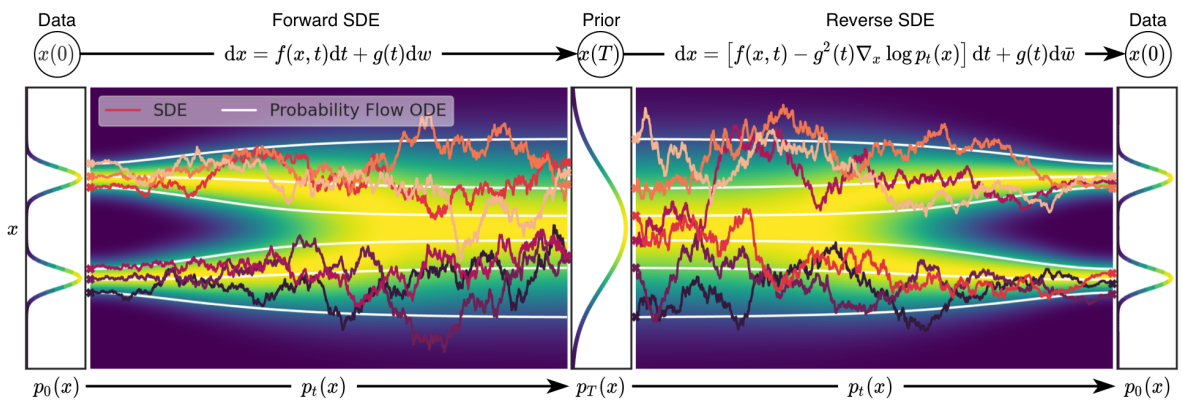



Figure 2: **Overview of score-based generative modeling through SDEs**. We can map data to a noise distribution (the prior) with an SDE (Section 3.1), and reverse this SDE for generative modeling (Section 3.2). We can also reverse the associated probability flow ODE (Section 4.3), which yields a deterministic process that samples from the same distribution as the SDE. Both the reverse-time SDE and probability flow ODE can be obtained by estimating the score $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ (Section 3.3).

* Generating samples by reversing the SDE

Forward가 diffusion이면, reverse도 diffusion.

$$\mathrm{d}\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}},$$

, $t = T \to 0$

↳ 모든 $t$에 대해 알면, $p_0$ 가능.

* Estimating score for SDE

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_t\Big\{\lambda(t)\mathbb{E}_{\mathbf{x}(0)}\mathbb{E}_{\mathbf{x}(t)|\mathbf{x}(0)}\big[\,\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}(t), t) - \nabla_{\mathbf{x}(t)} \log p_{0t}(\mathbf{x}(t) \mid \mathbf{x}(0))\|_2^2\,\big]\Big\}.$$

↳ 여기에 DSM 적용 (Sliced SM도 가능)

↳ 효율적으로 풀기 위해 transition kernel $p_{0t}(x(t) \mid x(0))$를 알아야 한다.

↳ $f(\cdot, t)$가 affine 이라면, transition kernel은 항상 Gaussian distribution이다. (mean, var는 closed form)

* Example VE, VP SDEs and beyond

SMLD와 DDPM은 SDE의 관점 다른 이산화.

NJuel noise scale의 SMLD 이해

$x_i = x_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2}\, z_{i-1}, \quad i = 1, \cdots, N$

$N \to \infty$ 이면, $\sigma_i \Rightarrow \sigma(t)$, $z_i \to z(t)$가 된다. (continuous, $t = [0,1]$)

$\{x(t)\}_{t=0}^{1}$ 의 SDE : $dx = \sqrt{\frac{d[\sigma^2(t)]}{dt}}\, dw$ ····· ①

DDPM 이해는

$x_i = \sqrt{1-\beta_i}\, x_{i-1} + \sqrt{\beta_i}\, z_{i-1}, \quad i = 1, \cdots, N$

$N \to \infty$ 이면,

$\{x(t)\}_{t=0}^{1}$ 의 SDE : $dx = -\frac{1}{2}\beta(t)x\,dt + \sqrt{\beta(t)}\, dw$ ···· ②

$t \to \infty$ 인두를 ① 은 variance가 exploding 된다. ⇒ VE SDE

② 는 fixed variance. ⇒ VP SDE

VP로부터 likelihood에 잘 맞는 SDE제시.

$$\mathrm{dx} = -\frac{1}{2}\beta(t)\mathrm{x}\,\mathrm{d}t + \sqrt{\beta(t)(1 - e^{-2\int_0^t \beta(s)\mathrm{d}s})}\mathrm{dw}.$$

↳ const $\beta(t)$를 쓰면, 항상 중간 time step에서 VP SDE에 의해 bound 된다.

↳ ∴ sub-VP SDE

VE, VP, sub-VP 모두 affine drift coefficient 를 가지고 있고. $P_{0t}(x(t)|x(0))$ 은 가우시안 이며 closed form이다.

* Solving the Reverse SDE

time dependent score-based $S_\theta$ 를 훈련하고 난 후,

reverse-time SDE를 구성하고, numerical approach로 $P_0$에서 sample을 생성할 수 있다.

* General-purpose numerical SDE solver

Numerical solver SDE는 trajectory를 제공한다.

Euler-maruyama 나 runge-kutta 같은 general-purpose numerical method는

학률에 따른 stochastic dynamics의 discretization이며, 이로 sample 생성가능.

새 SDE에 대한 reverse diffusion sampler를 제안한다.

↳ DDPM과 SMLD의 ancestral sampling보다 성능이 좋다.

Table 1: Comparing different reverse-time SDE solvers on CIFAR-10. Shaded regions are obtained with the same computation (number of score function evaluations). Mean and standard deviation are reported over five sampling runs. "P1000" or "P2000": predictor-only samplers using 1000 or 2000 steps. "C2000": corrector-only samplers using 2000 steps. "PC1000": Predictor-Corrector (PC) samplers using 1000 predictor and 1000 corrector steps.

| | Variance Exploding SDE (SMLD) | | | | Variance Preserving SDE (DDPM) | | | |
|---|---|---|---|---|---|---|---|---|
| FID↓ Sampler / Predictor | P1000 | P2000 | C2000 | PC1000 | P1000 | P2000 | C2000 | PC1000 |
| ancestral sampling | 4.98 ± .06 | 4.88 ± .06 | | **3.62 ± .03** | 3.24 ± .02 | 3.24 ± .02 | | **3.21 ± .02** |
| reverse diffusion | 4.79 ± .07 | 4.74 ± .08 | 20.43 ± .07 | **3.60 ± .02** | 3.21 ± .02 | 3.19 ± .02 | 19.06 ± .06 | **3.18 ± .01** |
| probability flow | 15.41 ± .15 | 10.54 ± .08 | | **3.51 ± .04** | 3.59 ± .04 | 3.23 ± .03 | | **3.06 ± .03** |

## Predictor-Corrector Samplers

MCMC로 sampling 특, numerical SDE solver로 만들 수 있다.

- time step에서 SDE solver는 다음 step의 sample을 estimate하는 predictor다.
- score-based MCMC는 corrector로, estimate된 sample을 marginal distribution에 맞게 수정
- Predictor & Corrector (PC)
- SMLD와 DDPM을 일반화

SMLD : identity function을 predictor, LD를 corrector.

DDPM : ancestral sampling을 predictor, identity function을 corrector

PC가 다른 것보다 성능↑

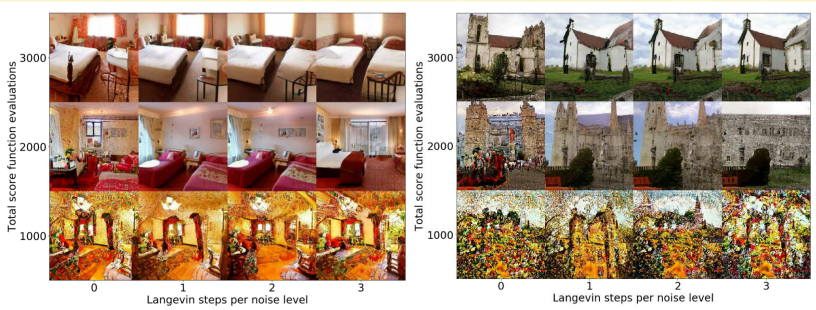



Figure 9: PC sampling for LSUN bedroom and church. The vertical axis corresponds to the total computation, and the horizontal axis represents the amount of computation allocated to the corrector. Samples are the best when computation is split between the predictor and corrector.

* Probability flow and connection to neural ODE.

Score-based 는 reverse-time SDE 의 numerical method 를 제공한다

모든 diffusion process 는 trajectory 가 동일한 marginal probability distribution $\{p_t(x)\}_{t=0}^{T}$ 을 SDE 로써

공유하는 deterministic process 가 있다. ⇒ SDE 푸는 방식만 다르지 결과는 같다고 말하는 듯?

이 deterministic process 는 다음 만족

$$\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - \frac{1}{2} g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] \mathrm{d}t, \quad \Rightarrow \text{probability flow ODE}$$

- Exact likelihood computation.

기존 DDPM 의 네트워 성능이 올라갔다.

- Manipulating latent representation.

$x_0$ 를 $x_T$ 로 encoding 이 가능.

latent $x_T$ 에서 manipulating 도 가능

- Uniquely identifiable encoding

unique 하게 invertible 하다.

충분한 data, capability, optim acc 가 있으면, $x_0$ 에 mm $x_T$ 가 unique 하게 결정된다

$dx = f(x,t)dt + g(t)dw$ 가 trainable param 이 없고.

ODE 인 normalize flow $dx = \left[f(x,t) - \frac{1}{2} g(t)^2 \nabla_x \log p_t(x)\right] dt$ 도 똑같은 score 가 주어지면

같은 궤적을 그린다

- Efficient sampling.

* Architecture Improvement.

* Controllable generation.