

# SCORE-BASED GENERATIVE MODELING THROUGH STOCHASTIC DIFFERENTIAL EQUATIONS

Song, Y., & Jo, J. (2021). International Conference on Learning Representations, 1-15.

B4 西尾舞紘

# Abstract

目的: 確率微分方程式 (SDE) を用いたスコアベース生成モデルの新しいフレームワークを提案する。

Propose a novel framework for score-based generative modeling using stochastic differential equations (SDEs).

## Novelty

確率微分方程式 (SDE) および逆時間SDEに基づく新しい生成モデリングフレームワークの導入

The introduction of a new generative modeling framework based on Stochastic Differential Equations (SDEs) and reverse-time SDEs.

## Usefulness

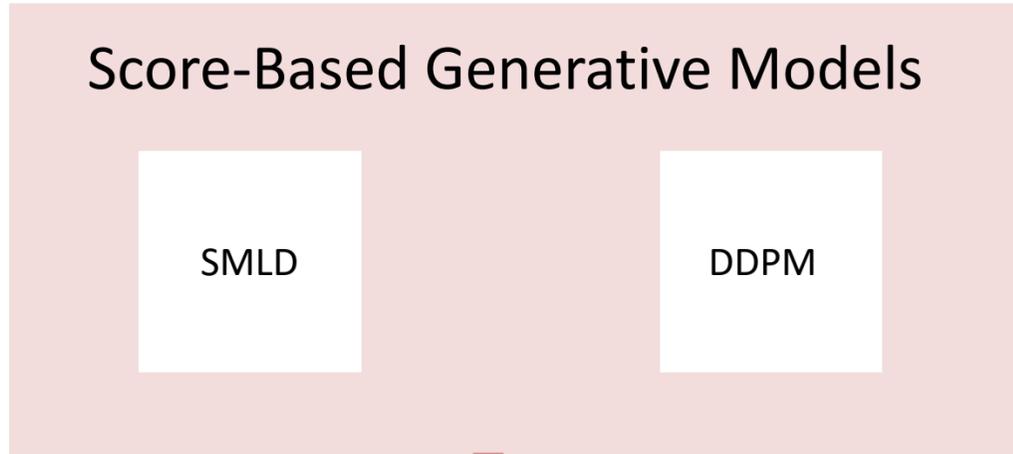
- 精度が高い効率的なサンプリング手法を提案  
Propose efficient sampling methods that enhance accuracy.
- 逆問題に対する新しい解決策を提供し、モデルの能力を拡張  
Provide new solutions for inverse problems, extending the capabilities of generative models.
- 画像生成タスクにおいて記録的な性能を達成し、高忠実度画像生成を実現  
Achieve record-breaking performance in image generation tasks, realizing high-fidelity image generation.

## Reliability

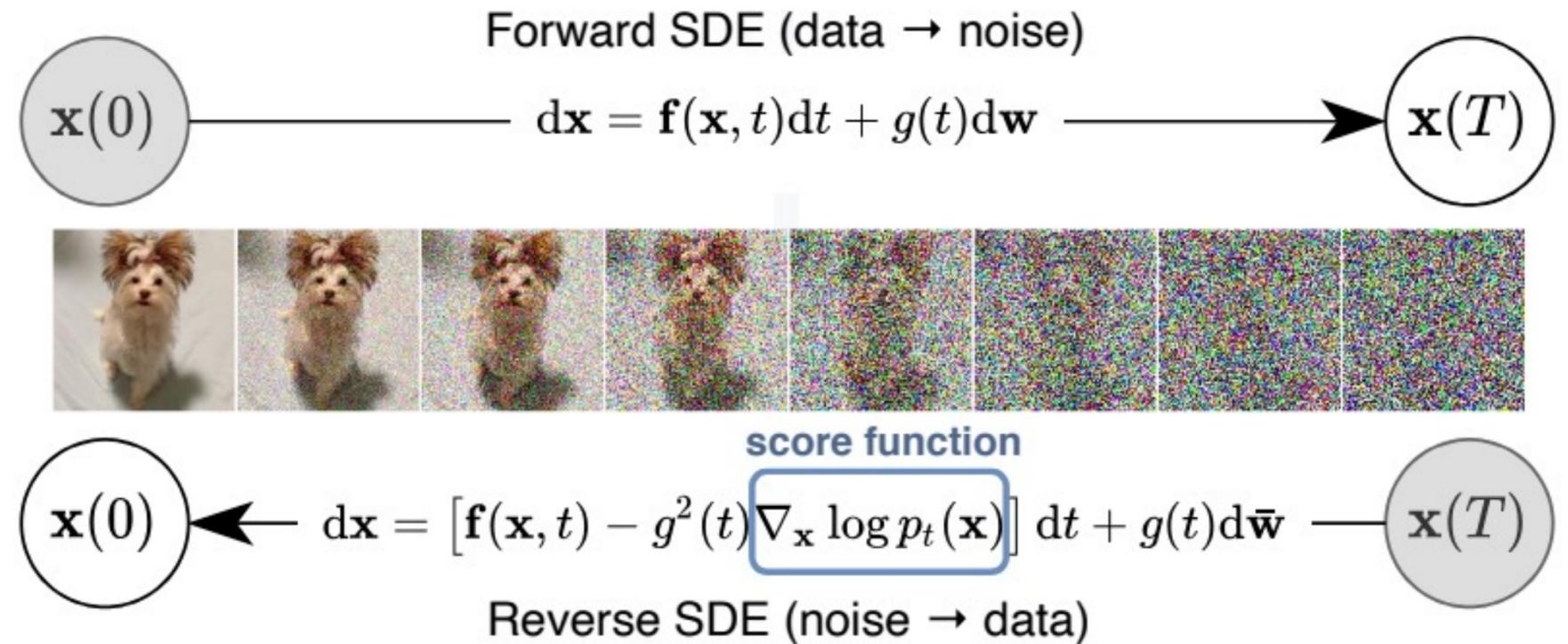
- フレームワークが、確立されたSDEおよび逆時間SDEの数学的原理に基づいている。  
The framework is based on well-established mathematical principles of SDEs and reverse-time SDEs.
- 定量的な指標 (Inceptionスコア、FID、尤度) を用いた広範な実験を行っている。  
Extensive experiments were conducted using quantitative metrics (Inception score, FID, and likelihood).

# 1. Introduction

# 1.Introduction



画像生成 (Song & Ermon, 2019, 2020; Ho et al., 2020)  
音声生成 (Chen et al., 2020; Kong et al., 2020)  
グラフ生成 (Niu et al., 2020)  
形状生成 (Cai et al., 2020) で効果的



拡散プロセスの連続進化と対応する逆時間ダイナミクスを活用することで、スコアベース生成モデルの能力を向上させ、拡張することを目指す。

This framework aims to enhance and expand the capabilities of score-based generative models by leveraging the continuous evolution of data under a diffusion process and its corresponding reverse-time dynamics.

# 1. Introduction

## ■ Some theoretical and practical contributions in this framework

### Flexible sampling and likelihood computation

#### 1. general-purpose SDE solver

To integrate the reverse-time SDE (stochastic differential equation), general-purpose SDE solvers can be used.

#### 2. Special sampling methods

Combining a numerical SDE solver with a score-based MCMC approach.

### Controllable generation

#### 1. Controllable generation

It is possible to modulate the generation process by conditioning on information that was not available during training.

#### 2. Efficient estimation of conditional reverse-time SDE

The conditional reverse-time stochastic differential equation (SDE) can be efficiently estimated from unconditional scores.

### Unified framework

#### 1. Integration of SMLD and DDPM

Providing a unified method for exploring and adjusting various stochastic differential equations (SDEs)

#### 2. Improvement in Sample Quality

#### 3. High-Fidelity Image Generation

Using a new architecture and sampling algorithm

#### 4. Proposal of New SDE and Record-Setting Performance

## 2. Background

# 2.1 Denoising Score Matching with Langevin Dynamics (SMLD)

## 1. ノイズスケールの設定

一連の正のノイズスケール

$$\sigma_{\min} = \sigma_1 < \sigma_2 < \dots < \sigma_N = \sigma_{\max}$$

摂動カーネル perturbation kernel

元のデータにランダムなノイズを加えることで、新しいデータを生成するための確率分布

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



新しいデータ分布：

$$p_{\sigma}(\tilde{x}) := \int p_{\text{data}}(x)p_{\sigma}(\tilde{x}|x)dx$$

$$p_{\sigma_{\min}}(x) \approx p_{\text{data}}(x)$$

ノイズほとんど加えない

$$p_{\sigma_{\max}}(x) \approx \mathcal{N}(x; 0, \sigma_{\max}^2 I)$$

非常に強いノイズが加わり、散らばっている。

## 2. Training of the Noise Conditional Score Network (NCSN)

ノイズ条件付きスコアネットワーク (NCSN)  $s_{\theta}(x, \sigma)$  を訓練

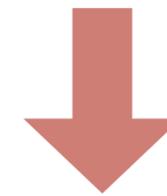
$$\theta^* = \arg \min_{\theta} \sum_{i=1}^N \sigma_i^2 \mathbb{E}_{p_{\text{data}}(x)} \mathbb{E}_{p_{\sigma_i}(\tilde{x}|x)} \left[ \left\| s_{\theta}(\tilde{x}, \sigma_i) - \nabla_{\tilde{x}} \log p_{\sigma_i}(\tilde{x}|x) \right\|_2^2 \right]$$

モデルが予測するスコア      実際のスコア

$\theta$  : モデルのパラメータ

$\theta^*$  : 最適化されたモデルのパラメータ

異なるノイズスケール  $\sigma_i$  に対して、モデルが予測するスコアが実際のデータ分布のスコアにできるだけ近くなるようにモデルのパラメータ  $\theta$  を最適化



ノイズのレベルに応じて正確にデータの特徴を捉えることができる  
モデルが得られる。

# 2.1 Denoising Score Matching with Langevin Dynamics (SMLD)

## 3. Sampling with Langevin MCMC

$$\mathbf{x}_i^m = \mathbf{x}_i^{m-1} + \epsilon_i \mathbf{s}_\theta^*(\mathbf{x}_i^{m-1}, \sigma_i) + \sqrt{2\epsilon_i} \mathbf{z}_i^m, \quad m = 1, 2, \dots, M$$

ドリフト項

最適化されたスコア関数  $\mathbf{s}_\theta^*(\mathbf{x}_i^{m-1}, \sigma_i)$  に基づいてデータ分布の勾配方向に移動

ノイズ項

標準正規分布に従うノイズ  $\mathbf{z}_i^m$  を加える。これにより、サンプルが確率的に探索される。

$\mathbf{x}_i^m$  ノイズスケール  $\sigma_i$  での第  $m$  ステップでのサンプル

$M$  サンプリングステップでの総数

$\mathbf{z}_i^m$  標準正規分布に従うランダムなノイズ

$\epsilon_i$  ステップサイズ

### Initialization

サンプリングの開始の際に、 $\sigma_{\max}$  に対応する標準正規分布に従った初期値  $\mathbf{x}_N^0$  を生成

$$\mathbf{x}_N^0 \sim \mathcal{N}(\mathbf{x} \mid \mathbf{0}, \sigma_{\max}^2 \mathbf{I})$$

### Langevin MCMC Steps

各ノイズスケール  $\sigma_i$  に対して  $i = N, N-1, \dots, 1$  の順に以下を実行

$$\mathbf{x}_i^0 = \mathbf{x}_{i+1}^M \text{ を初期値として設定}$$

$m = 1, 2, \dots, M$  で Langevin MCMC を実行

### Obtaining the final sample

$M \rightarrow \infty \quad \epsilon_i \rightarrow 0$  の時、最終的なサンプル  $\mathbf{x}_1^M$  は

$$p_{\sigma_{\min}}(\mathbf{x}) \approx p_{\text{data}}(\mathbf{x})$$

からの正確なサンプルになる。

モデルが学習したデータ分布に従った新しいデータポイントを生成することができる。

## 2.2 Denoising Diffusion Probabilistic Models (DDPM)

### ■ Constructing noise scales and Markov chain

#### 1. ノイズスケールの定義 The definition of noise scale

$$0 \leq \beta_1, \beta_2, \dots, \beta_N \leq 1$$

#### 2. マルコフ連鎖の構築 Constructing Markov chain

各トレーニングデータポイント  $\mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x})$  について離散マルコフ連鎖  $\{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_N\}$  を構築する

各ステップ  $i$  での遷移確率の定義

$$p(x_i | x_{i-1}) = \mathcal{N}(x_i; x_{i-1}, \sqrt{1 - \beta_i} I) \quad \longrightarrow \quad p_{\alpha_i}(x_i | x_0) = \mathcal{N}(x_i; x_0, \sqrt{\alpha_i} (1 - \alpha_i) I) \quad (\alpha_i := \prod_{j=1}^i (1 - \beta_j))$$

#### 3. ノイズスケールの設定 Setting the noise scale

ノイズスケール  $\beta_i$  は、 $\mathbf{x}_N$  がほぼ  $\mathcal{N}(\mathbf{0}, \mathbf{I})$  に従うように設定。

## 2.2 Denoising Diffusion Probabilistic Models (DDPM)

### ■ Parameterization of Markov chain in the reverse direction

#### 1. 逆方向のマルコフ連鎖 Markov chain in the reverse direction

現在のステップ  $\mathbf{x}_i$  から前のステップ  $\mathbf{x}_{i-1}$  をガウス分布を用いて生成するための条件付き確率分布を定義

$$p_{\theta}(\mathbf{x}_{i-1} | \mathbf{x}_i) = \mathcal{N}\left(\mathbf{x}_{i-1}; \frac{1}{\sqrt{1-\beta_i}}(\mathbf{x}_i + \beta_i \mathbf{s}_{\theta}(\mathbf{x}_i, i)), \beta_i \mathbf{I}\right)$$

ノイズを含むデータから元のデータを再構成するプロセスを効果的にモデル化

#### 2. モデルのトレーニング

the evidence lower bound (ELBO) を用いて訓練する。

$$\theta^* = \arg \min_{\theta} \sum_{i=1}^N (1 - \alpha_i) \mathbb{E}_{p_d(\mathbf{x})} \mathbb{E}_{p_{\alpha_i}(\tilde{\mathbf{x}}|\mathbf{x})} \left[ \|\mathbf{s}_{\theta}(\tilde{\mathbf{x}}, i) - \nabla_{\tilde{\mathbf{x}}} \log p_{\alpha_i}(\tilde{\mathbf{x}} | \mathbf{x})\|_2^2 \right]$$

各時間ステップ  $i$  に対する誤差と重みをかけた総和    モデルが実際のデータ分布の勾配をどれくらい正確に予測しているか

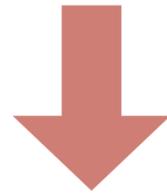
モデルがデータの生成過程を正確に逆方向に再現できるようになる。

## 2.2 Denoising Diffusion Probabilistic Models (DDPM)

### 3. サンプルの取得

$$\mathbf{x}_{i-1} = \frac{1}{\sqrt{1-\beta_i}} (\mathbf{x}_i + \beta_i \mathbf{s}_{\theta^*}(\mathbf{x}_i, i)) + \sqrt{\beta_i} \mathbf{z}_i, \quad i = N, N-1, \dots, 1.$$

初期サンプル  $\mathbf{x}_N$  を用いて逆方向に進み、各ステップでノイズを徐々に減少させながらサンプルを生成  
Generate samples by moving backward using the initial sample  $\mathbf{x}_N$ , gradually reducing noise at each step.



この方法を祖先サンプリング **ancestral sampling** と呼ぶ

祖先サンプリング **ancestral sampling**

グラフィカルモデル  $\prod_{i=1}^N p_{\theta}(\mathbf{x}_{i-1} | \mathbf{x}_i)$  からのサンプリングを行う  
ことに相当

This corresponds to sampling from a graphical model  $\prod_{i=1}^N p_{\theta}(\mathbf{x}_{i-1} | \mathbf{x}_i)$

重みと摂動カーネルの関係

$$1 - \alpha_i \propto \frac{1}{\mathbb{E}[\|\nabla_x \log p_{\alpha_i}(\tilde{x}|x)\|_2^2]}$$

$$\sigma_i^2 \propto \frac{1}{\mathbb{E}[\|\nabla_x \log p_{\sigma_i}(\tilde{x}|x)\|_2^2]}$$

### 3. Score-Based generative modeling with SDEs

# 3.1 Perturbing data with SDEs

## Diffusion Process Model

### 1. From Data Distribution to Prior Distribution

- 初期状態  $\mathbf{x}(0)$  はデータ分布  $p_0$  に従う
- 最終状態  $\mathbf{x}(T)$  は事前分布  $p_T$  に従う

### 2. Process of Adding Noise

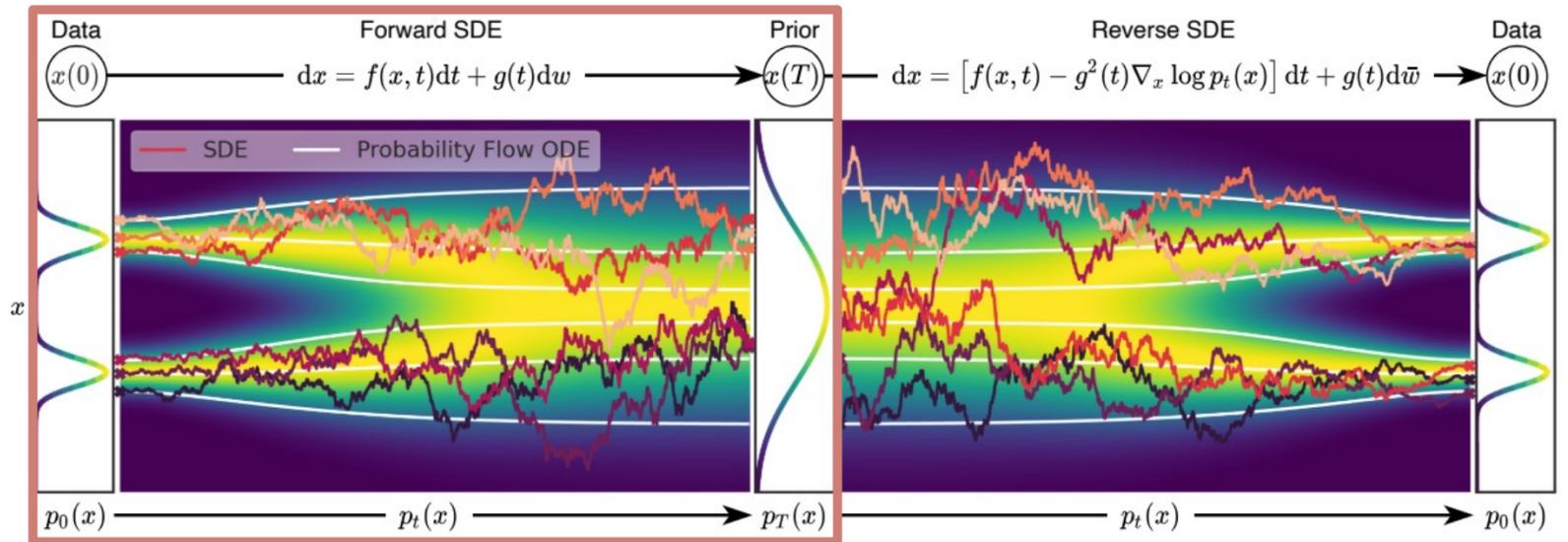
Ito stochastic differential equation (Ito SDE)

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + g(t)d\mathbf{w}$$

**ドリフト係数**  
データの平均的な変動を表す。

**拡散係数**  
スカラー関数。データのランダムな変動の度合いを表す。

The standard Wiener process (Brownian motion)に対する微小変化



Overview of score-based generative modeling through SDEs.

このSDEは、係数が状態および時間の両方においてグローバルリプシッツ連続である限り、唯一の強解を持つ (Øksendal, 2003)

This SDE has a unique strong solution as long as the coefficients are globally Lipschitz in both state and time.

データ分布をある特定の事前分布に変換するためのSDEの構築

# 3.2 Generating samples by reversing the SDE

## 1. サンプルの逆転

$$\mathbf{x}(T) \sim p_T \xrightarrow{\text{逆転}} \mathbf{x}(0) \sim p_0$$

のサンプル                      のサンプル

## 2. 逆時間SDE

拡散プロセスの逆：時間を逆に進む別の拡散プロセス (Anderson(1982))

逆時間確率微分方程式 (reverse-time SDE)

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

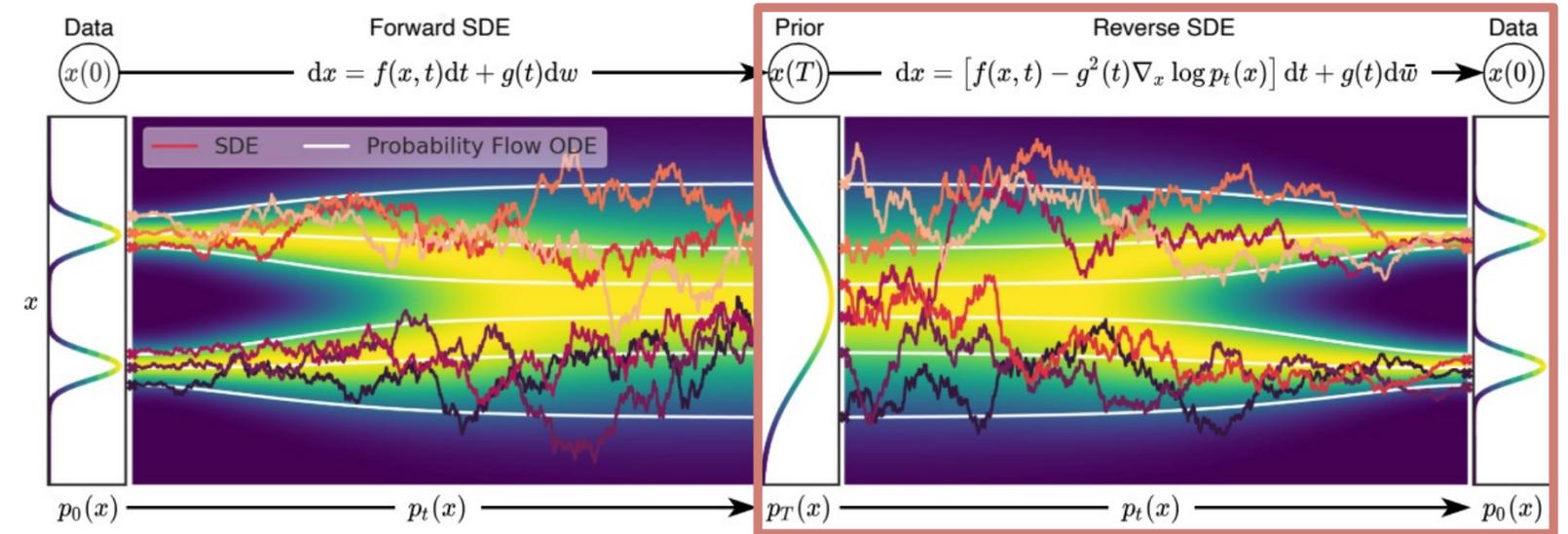
時間 が 逆に進む標準ウィーナー過程

スコア関数

データがノイズから元の分布に向かうよう誘導。これがわかれば、逆拡散プロセス導出可

負のタイムステップ

$p_0$  からサンプルを生成することができるようになる



# 3.3 Estimating scores for the SDE

## ■ Estimating the score of a distribution

$\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ を推定するため、スコアベースモデル  $\mathbf{s}_{\theta}(\mathbf{x}, t)$ を訓練する。

A score-based model is trained to estimate  $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$

スコアマッチング (Hyvarinen, 2005; Song et al., 2019a) を使用してサンプル上でスコアを推定。

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

異なる時刻に対するモデルを調整するための正の重み関数

モデルの予測がどれくらい正確か

$$\lambda = \frac{1}{\mathbb{E} \left[ \left\| \nabla_{\mathbf{x}(t)} \log p_{0t}(\mathbf{x}(t) | \mathbf{x}_0) \right\|_2^2 \right]}$$

十分なデータとモデルの容量があれば、スコアマッチングにより、最適な解  $\mathbf{s}_{\theta^*}(\mathbf{x}, t)$  がほぼすべての  $\mathbf{x}$  と  $t$  に対して  $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$  に等しくなる。

With sufficient data and model capacity, score matching ensures that the optimal solution

# 3.4 Examples : VE, VP SDEs and beyond

## ■ SMLDとSDE

### 1. 離散的なノイズ摂動

$$\mathbf{x}_i = \mathbf{x}_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2} \mathbf{z}_{i-1}, \quad i = 1, \dots, N,$$

ノイズスケール  
マルコフ連鎖を表す

ランダムノイズ  
標準正規分布に従う

ステップごとにノイズを追加してデータポイント化  
Add noise at each step to make data points

### 2. 連続的なSDEへの変換

Variance Exploding (VE) SDE

$$d\mathbf{x} = \sqrt{\frac{d[\sigma^2(t)]}{dt}} d\mathbf{w}$$

時間が進むにつれ分散が増加する。

プロセス $\mathbf{x}(t)$ がノイズスケール $\sigma(t)$ により、どのように影響を受けるかを表す。  
Denotes how the process  $\mathbf{x}(t)$  is affected by the noise scale  $\sigma(t)$ .

無限に多いステップ数の場合、離散的なステップが連続的なSDEに収束する。

# 3.4 Examples : VE, VP SDEs and beyond

## DDPMとSDE

### 1. 離散的なノイズ摂動

$$\mathbf{x}_i = \sqrt{1 - \beta_i} \mathbf{x}_{i-1} + \sqrt{\beta_i} \mathbf{z}_{i-1} \quad i = 1, \dots, N.$$

ノイズスケール      ランダムノイズ  
標準正規分布に従う

ステップごとにノイズを追加してデータポイント化  
Add noise at each step to make data points

### 2. 連続的なSDEへの変換

Variance Preserving (VP) SDE

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

時間が進むにつれ分散が一定に保たれる。

プロセス $x(t)$ がノイズスケール $\beta(t)$ により、どのように影響を受けるかを表す。  
Denotes how the process  $x(t)$  is affected by the noise scale  $\beta(t)$ .

無限に多いステップ数の場合、離散的なステップが連続的なSDEに収束する。

# 3.4 Examples : VE, VP SDEs and beyond

## ■ VE SDEとVP SDE

VE SDE

$$dx = \sqrt{\frac{d[\sigma^2(t)]}{dt}} dw$$

時間tが進むにつれて分散が無限に増加する。画像生成やデータ復元などにおいて、ノイズが強くなっていくことを示す。

VP SDE

$$dx = -\frac{1}{2}\beta(t)x dt + \sqrt{\beta(t)} dw.$$

時間tが進むにつれて分散が一定に保たれる。生成されるデータが一定の品質を保つことを意味する。

組み合わせて新しいSDE (**sub-VP SDE**)を提案。  
A new SDE (**sub-VP SDE**) is proposed in combination.

## ■ sub-VP SDE

$$dx = \underbrace{-\frac{1}{2}\beta(t)x dt}_{\text{ドリフト項}} + \underbrace{\sqrt{\beta(t)(1 - e^{-2\int_0^t \beta(s)ds})}}_{\text{拡散項}} dw$$

ドリフト項

拡散項

ドリフト項

$$-\frac{1}{2}\beta(t)x dt$$

- $\mathbf{x}$ の平均的な変化を記述する。

拡散項

$$\sqrt{\beta(t)(1 - e^{-2\int_0^t \beta(s)ds})}$$

- $\mathbf{x}$ のランダムな変化を記述する。
- ランダムな変動が $\beta(t)$ と過去の累積された $\beta(s)$ によってどのように影響されるかを示す。

# 3.4 Examples : VE, VP SDEs and beyond

## ■ sub-VP SDE

$$dx = -\frac{1}{2}\beta(t)x dt + \sqrt{\beta(t)(1 - e^{-2\int_0^t \beta(s)ds})}dw$$

時間と共に拡散項が変化する。

### 1. 初期段階( $t \approx 0$ )

- 累積値  $\int_0^t \beta(s) ds$  は小さい。
- そのため、 $e^{-2\int_0^t \beta(s) ds}$  はほぼ1で  $(1 - e^{-2\int_0^t \beta(s) ds})$  はほぼ0。
- 拡散項の係数は小さくなる。



時間が進むにつれ、累積値は大きく、 $e^{-2\int_0^t \beta(s) ds}$  は指数関数的に0に近づいていく

### 2. 長時間経過後( $t \rightarrow \infty$ )

- 累積値  $\int_0^t \beta(s) ds$  は非常に大きくなる。
- そのため、 $e^{-2\int_0^t \beta(s) ds}$  は非常に小さくなり  $(1 - e^{-2\int_0^t \beta(s) ds})$  はほぼ1。
- 拡散項の係数はほぼ $\beta(t)$  に等しくなる。

拡散項が $\beta(t)$  に制約されるため、SDEの分散は過度に増加せず、適度に制御される。

## ■ VE SDE, VP SDE, sub-VP SDE比較

	VE SDE	VP SDE	sub-VP SD
拡散項	$\sqrt{\frac{d[\sigma^2(t)]}{dt}}$	$\sqrt{\beta(t)}$	$\sqrt{\beta(t)(1 - e^{-2\int_0^t \beta(s) ds})}$
分散の増加	時間と共に増加	一定	制約されるが柔軟
モード崩壊のリスク	高い	低い	非常に低い
数値安定性	不安定になりやすい	安定	非常に安定

### sub-VP SDEまとめ

- 過剰な分散の増加を防ぎつつ、適度なランダム性を保つ。  
Prevent excessive increases in variance while maintaining an appropriate level of randomness.
- 分散の増加を適度に抑えるため、計算コストを効率的に管理できる。  
By adequately controlling the increase in variance, it allows for efficient management of computational costs.

## 4. Solving the reverse SDE

# 4.1 General-purpose numerical SDE solvers

## ■ Numerical Solvers for Stochastic Differential Equations (SDEs)

### オイラー-丸山法 (Euler-Maruyama Method)

- 最も基本的なSDEの数値解法の一つで、オイラー法を拡張したもの  
One of the most basic numerical solvers for SDEs, extending the Euler method.
- 簡単に実装できるが、精度が低い場合がある。  
Easy to implement but may have lower accuracy.

### 確率ルンゲ-クッタ法 (Stochastic Runge-Kutta Method)

- ルンゲ-クッタ法の確率版で、より高精度な解を提供  
The stochastic version of the Runge-Kutta method, providing higher accuracy.
- 計算コストが高くなる場合がある  
The stochastic version of the Runge-Kutta method, providing higher accuracy.

# 4.1 General-purpose numerical SDE solvers

## ■ Samplers

### 祖先サンプリング (Ancestral Sampling)

- DDPM (Denoising Diffusion Probabilistic Models) のサンプリング方法の一つで、逆時間SDEの特定の離散化に対応

One of the sampling methods of DDPM (Denoising Diffusion Probabilistic Models), corresponding to a specific discretization of the reverse-time SDE.

- 新しいSDEに対して祖先サンプリングルールを導出するのが難しい  
This method can be challenging to derive sampling rules for new SDEs.

### 逆拡散サンプラー (Reverse Diffusion Samplers)

- 逆時間SDEを前向きのSDEと同じ方法で離散化することで、サンプリングルールを容易に導出  
By discretizing the reverse-time SDE in the same way as the forward one, sampling rules can be readily derived.
- 新しいSDEに対しても迅速かつ簡単にサンプリングが可能となる  
This method allows for quick and easy sampling even for new SDEs.

# 4.2 Predictor corrector samplers

## ■ Predictor-Corrector (PC) samplers

SMLDとDDPMの手法を一般化して予測子と修正子を柔軟に組み合わせる手法

1. 各時間ステップでthe numerical SDE solverでのサンプルの推定値を提供し、**予測子 Predictor**の役割を果たす。
2. その後、スコアベースでのscore-based MCMC approachが推定されたサンプルの周辺分布を修正し、**修正子 Corrector**の役割を果たす。

	Predictor	Corrector
SMLD	恒等関数	annealed Langevin dynamics
DDPM	Ancestral sampling	恒等関数

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

C2000

Correctorステップを2000回実行する。

### 生成モデルの品質



- 予測子ステップに修正子ステップを追加すると計算量は倍になるが、サンプルの質向上
- 予測子と修正子を組み合わせた方が品質が良い。

# 4.3 Probability flow and connection to neural ODEs

## Probability flow

逆時間確率微分方程式(reverse-time SDE)を解くための別の数値的方法

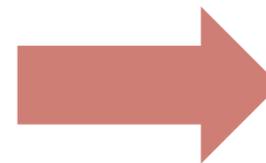
An alternative numerical method for solving the reverse-time stochastic differential equation (reverse-time SDE)

すべての拡散過程に対して、その軌跡がSDEと同じ周辺確率密度を共有する対応する決定論的過程が存在

For all diffusion processes, there exists a corresponding deterministic process whose trajectories share the same marginal probability densities as the SDE.

確率フローODE Probability flow ODE

$$d\mathbf{x} = \left[ \underbrace{\mathbf{f}(\mathbf{x}, t)}_{\text{ドリフト項}} - \frac{1}{2} \underbrace{g(t)^2}_{\text{拡散項}} \nabla_{\mathbf{x}} \underbrace{\log p_t(\mathbf{x})}_{\text{スコア関数}} \right] dt$$



### 決定論的過程 deterministic process

初期条件と時間に依存する一連の規則に従い、その未来の状態が完全に決定される過程。

The process by which its future state is completely determined according to a set of rules that depend on initial conditions and time.

初期条件とスコア関数によって  $\mathbf{x}(t)$  を決定。  
 $\mathbf{x}(t)$  is determined by initial conditions and a score function.

- 決定論的過程であるため、SDEと異なりノイズを含まず、滑らかな軌跡を生成
- 対応するSDEと同じ周辺確率密度を共有するため、統計的性質を保持
- スコア関数がニューラルネットワークに近似される場合、ニューラルODEの一例に

# 4.3 Probability flow and connection to neural ODEs

## Exact likelihood computation

確率フロー-ODE Probability flow ODE

$$d\mathbf{x} = \left[ \mathbf{f}(\mathbf{x}, t) - \frac{1}{2}g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) \right] dt$$



下の式に基づき、時間とともに連続的に変化するデータの密度を計算する

the instantaneous change of variables formula

$$d\mathbf{x} = \left\{ \mathbf{f}(\mathbf{x}, t) - \frac{1}{2} \nabla \cdot [\mathbf{G}(\mathbf{x}, t) \mathbf{G}(\mathbf{x}, t)^T] - \frac{1}{2} \mathbf{G}(\mathbf{x}, t) \mathbf{G}(\mathbf{x}, t)^T \mathbf{s}_{\theta}(\mathbf{x}, t) \right\} dt.$$



正確な尤度を計算できる  
calculate the exact likelihood

Table 2: NLLs and FIDs (ODE) on CIFAR-10.

Model	negative log-likelihoods NLL Test ↓	FID ↓
RealNVP (Dinh et al., 2016)	3.49	-
iResNet (Behrmann et al., 2019)	3.45	-
Glow (Kingma & Dhariwal, 2018)	3.35	-
MintNet (Song et al., 2019b)	3.32	-
Residual Flow (Chen et al., 2019)	3.28	46.37
FFJORD (Grathwohl et al., 2018)	3.40	-
Flow++ (Ho et al., 2019)	3.29	-
DDPM ( $L$ ) (Ho et al., 2020)	$\leq 3.70^*$	13.51
DDPM ( $L_{\text{simple}}$ ) (Ho et al., 2020)	$\leq 3.75^*$	3.17
DDPM	3.28	3.37
DDPM cont. (VP)	3.21	3.69
DDPM cont. (sub-VP)	3.05	3.56
DDPM++ cont. (VP)	3.16	3.93
DDPM++ cont. (sub-VP)	3.02	3.16
DDPM++ cont. (deep, VP)	3.13	3.08
DDPM++ cont. (deep, sub-VP)	<b>2.99</b>	<b>2.92</b>

# 4.3 Probability flow and connection to neural ODEs

## Exact likelihood computation

### DDPM cont.

DDPMモデルと同じアーキテクチャを使用しつつ、新たに提案された連続的な目的関数

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

を用いてモデルを訓練

→尤度が改善 Likelihood improved

### sub-VP SDE

VP SDEよりも、sub-VP SDEの方が常に高い尤度を提供

### DDPM++ cont

DDPM cont. のアーキテクチャを改良

DDPM cont. (deep, sub-VP)が最も高い尤度を提供

↑  
ネットワークの深さが2倍

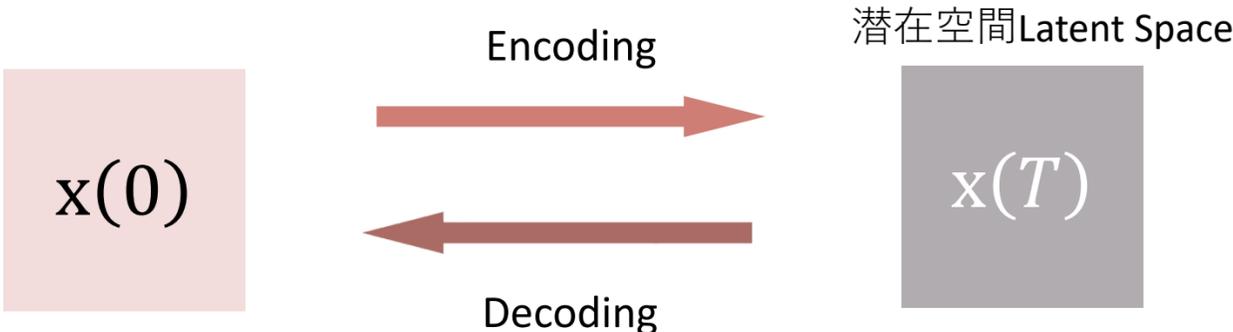
Table 2: NLLs and FIDs (ODE) on CIFAR-10.

Model	NLL Test ↓	FID ↓
RealNVP (Dinh et al., 2016)	3.49	-
iResNet (Behrmann et al., 2019)	3.45	-
Glow (Kingma & Dhariwal, 2018)	3.35	-
MintNet (Song et al., 2019b)	3.32	-
Residual Flow (Chen et al., 2019)	3.28	46.37
FFJORD (Grathwohl et al., 2018)	3.40	-
Flow++ (Ho et al., 2019)	3.29	-
DDPM ( $L$ ) (Ho et al., 2020)	$\leq 3.70^*$	13.51
DDPM ( $L_{\text{simple}}$ ) (Ho et al., 2020)	$\leq 3.75^*$	3.17
DDPM	3.28	3.37
DDPM cont. (VP)	3.21	3.69
DDPM cont. (sub-VP)	3.05	3.56
DDPM++ cont. (VP)	3.16	3.93
DDPM++ cont. (sub-VP)	3.02	3.16
DDPM++ cont. (deep, VP)	3.13	3.08
DDPM++ cont. (deep, sub-VP)	<b>2.99</b>	<b>2.92</b>

# 4.3 Probability flow and connection to neural ODEs

## Manipulating latent representations

### Encoding Process and Decoding Process



#### Encoding into Latent Space

$$d\mathbf{x} = \left[ \mathbf{f}(\mathbf{x}, t) - \frac{1}{2}g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) \right] dt$$

式は確率フローODEを表しており、これを時間  $t$  にわたって積分することで、エンコード。

The equation represents a probability flow ODE, which is encoded by integrating over time  $t$ .

#### Decoding Process

逆方向のSDEに対応する常微分方程式 (ODE) を積分する。  
Integrate the ordinary differential equation (ODE) corresponding to the SDE in the reverse direction.

#### 可逆モデル Invertible Models

可逆モデルでは、データのエンコードとデコードの過程が双方向に可能であり、潜在空間での操作が容易  
In invertible models, the encoding and decoding processes are bidirectional, making it easy to manipulate latent representations.

他にはneural ODEs, normalizing flows (Dinh et al., 2016; Kingma & Dhariwal, 2018)など

# 4.3 Probability flow and connection to neural ODEs

## ■ Manipulating latent representations

### Applications in Image Editing

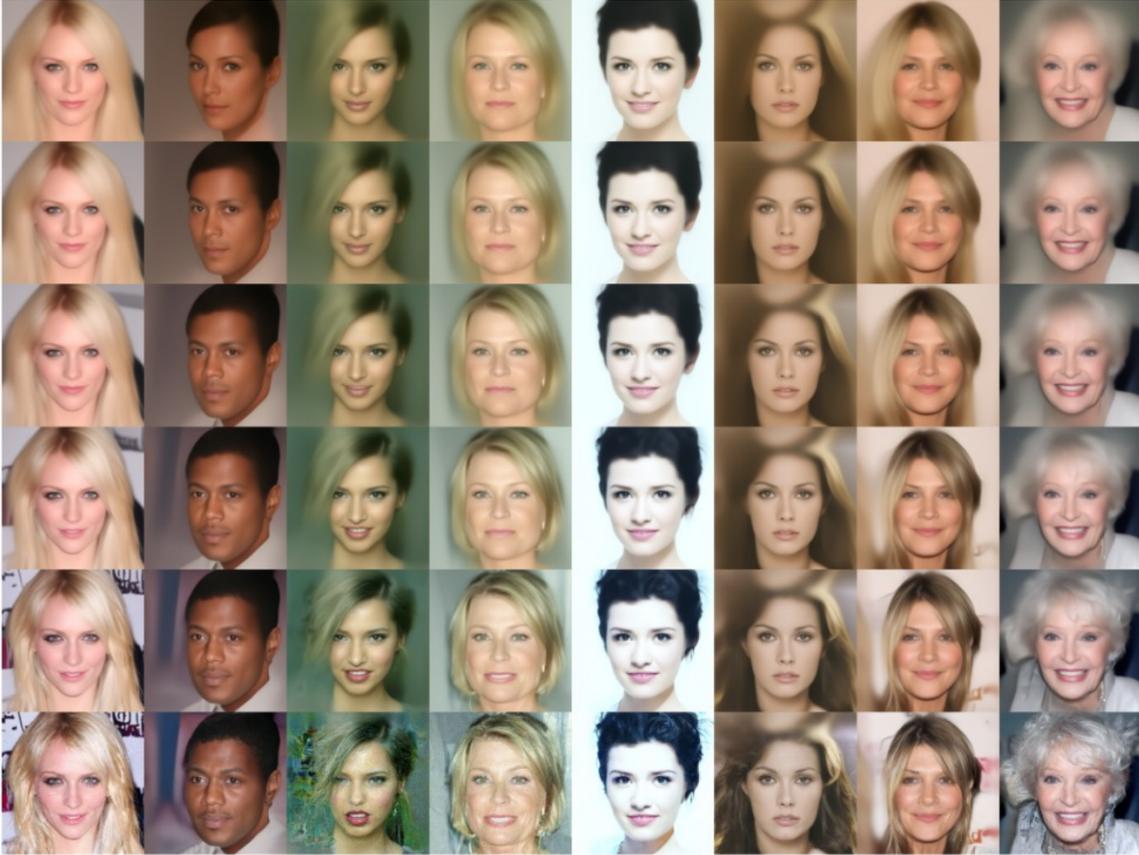
#### 補間 (Interpolation)

2つの画像の間を補間することで、新しい画像を生成。



#### 温度スケールリング (Temperature Scaling)

潜在空間での確率分布の温度を変更することで生成画像の多様性を調整する。



# 4.3 Probability flow and connection to neural ODEs

## ■ Uniquely identifiable encoding

### Uniquely identifiable encoding

- 特定の入力データに対する潜在空間へのエンコードが唯一。
- 同じ入力データが常に同じエンコードにマッピング。モデルの予測が安定し、一貫性が保たれる。

"Uniquely identifiable" means that for a given input data point, the encoding into the latent space is unique. In other words, the same input data is always mapped to the same encoding.

### Characteristics of the proposed model

#### Forward SDE

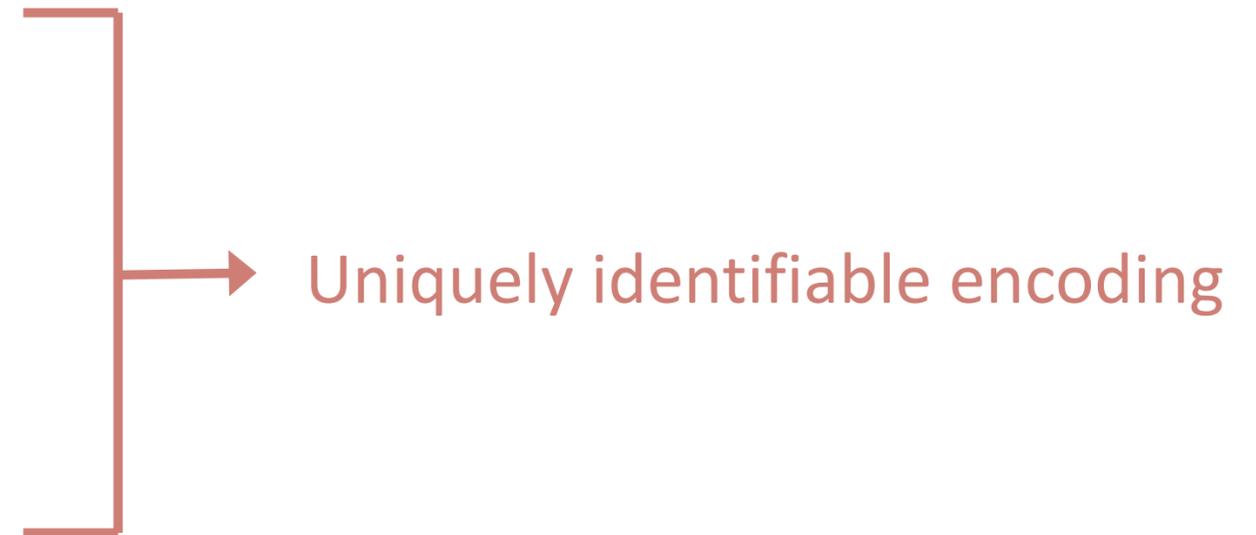
$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + g(t)d\mathbf{w}$$

訓練されたパラメータを持たない。

#### Probability flow ODE

$$d\mathbf{x} = \left[ \mathbf{f}(\mathbf{x}, t) - \frac{1}{2}g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) \right] dt$$

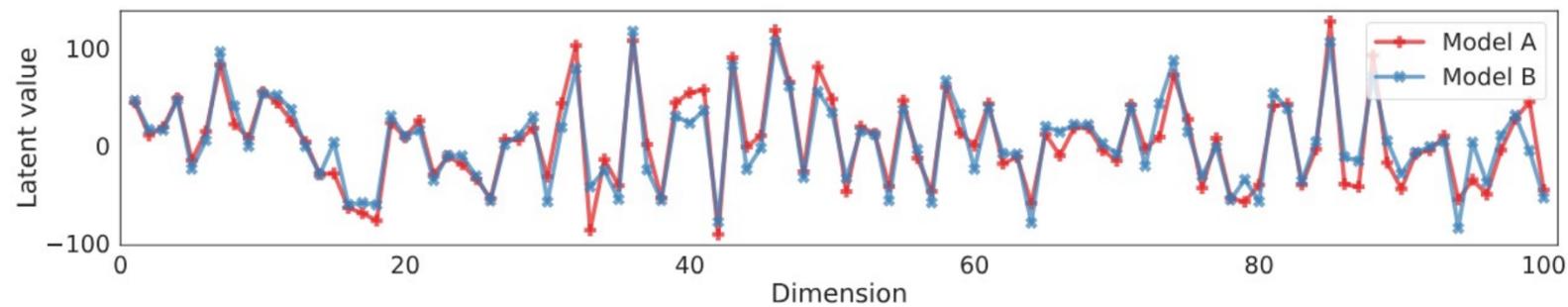
入力データに対するエンコードが一意に決定



# 4.3 Probability flow and connection to neural ODEs

## Uniquely identifiable encoding

### Uniquely identifiable encoding in the model



横軸 (Dimension)

潜在コードの次元。最初の100次元を表示

縦軸 (Latent Value)

潜在空間の各次元の値

ModelA:

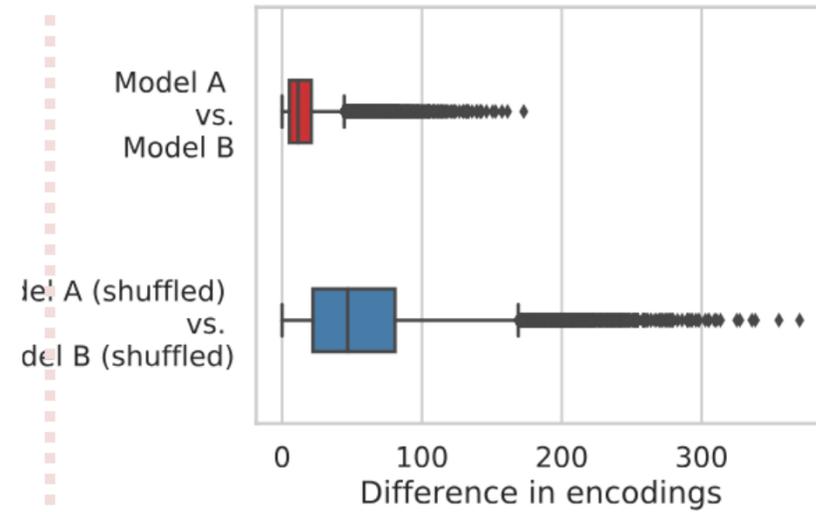
NCSN++モデルで、各解像度に対して4層のレイヤーを持つ

ModelB:

NCSN++モデルで、各解像度に対して8層のレイヤーを持つ

モデルAとモデルBは同じ入力に対して非常に近いエンコーディングを提供する。

VE SDEを用いたスコアベースの生成モデルが、異なるモデルアーキテクチャであっても一貫性のあるエンコーディングを生成する能力を持つ



横軸 (Dimension)

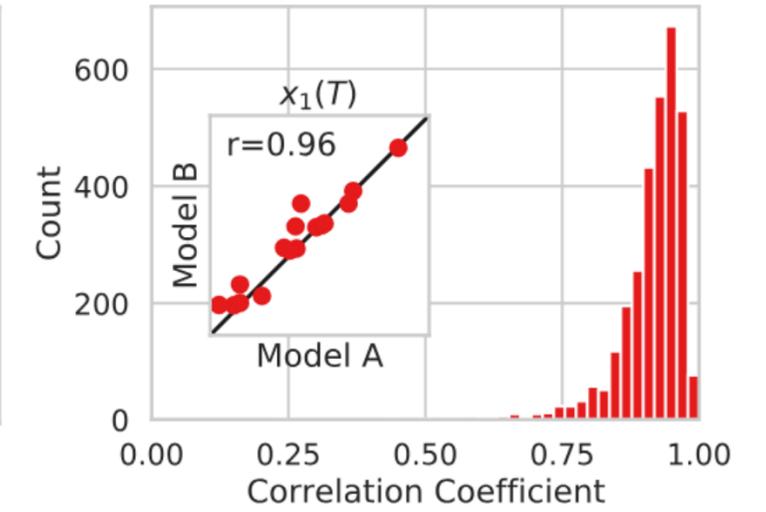
潜在空間の次元ごとの違いの大きさ

縦軸 (Latent Value)

比較対象のモデルペア

ModelAとModelBの潜在コードの違いが小さく、シャッフルデータの差との差は大きい

モデルAとモデルBの潜在コードが非常に似ている。



横軸 (Dimension)

相関係数

縦軸 (Latent Value)

ModelAとModelBの相関係数の頻度

相関係数は0.96

# 4.3 Probability flow and connection to neural ODEs

## ■ Efficient sampling neural ODEsの枠組みを利用してデータサンプリングを効率的に行う

### Sample Generation Using Probability Flow ODEs

#### 1. Prerequisites

Forward SDE

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + \mathbf{G}(t)d\mathbf{w}$$

The discretization of this SDE

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{f}_i(\mathbf{x}_i) + \mathbf{G}_i \mathbf{z}_i, \quad i = 0, 1, \dots, N - 1$$

#### 2. Probability flow ODE

Probability flow ODE

$$d\mathbf{x} = \left\{ \mathbf{f}(\mathbf{x}, t) - \frac{1}{2} \mathbf{G}(t) \mathbf{G}(t)^\top \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) \right\} dt.$$

#### 3. Discretization and sample generation

$$\mathbf{x}_i = \mathbf{x}_{i+1} - \mathbf{f}_{i+1}(\mathbf{x}_{i+1}) + \frac{1}{2} \mathbf{G}_{i+1} \mathbf{G}_{i+1}^\top \mathbf{s}_{\theta^*}(\mathbf{x}_{i+1}, i + 1), \\ i = 0, 1, \dots, N - 1.$$

#### 4. Application to SMLD and DDPM

Application to SMLD

$$\mathbf{x}_i = \mathbf{x}_{i+1} + \frac{1}{2} (\sigma_{i+1}^2 - \sigma_i^2) \mathbf{s}_{\theta^*}(\mathbf{x}_{i+1}, \sigma_{i+1}), \\ i = 0, 1, \dots, N - 1.$$

Application to DDPM

$$\mathbf{x}_i = (2 - \sqrt{1 - \beta_{i+1}}) \mathbf{x}_{i+1} + \frac{1}{2} \beta_{i+1} \mathbf{s}_{\theta^*}(\mathbf{x}_{i+1}, i + 1), \\ i = 0, 1, \dots, N - 1.$$

# 4.3 Probability flow and connection to neural ODEs

## ■ Efficient sampling

### Utilizing Black-Box ODE Solvers

#### 高品質なサンプルの生成 Generation of high quality samples

Dormand-Prince法（1980年）などのブラックボックスODEソルバーを使用することで、高品質なサンプルを生成できる。

Black box ODE solvers such as the Dormand-Prince method (1980) can be used to generate high-quality samples.

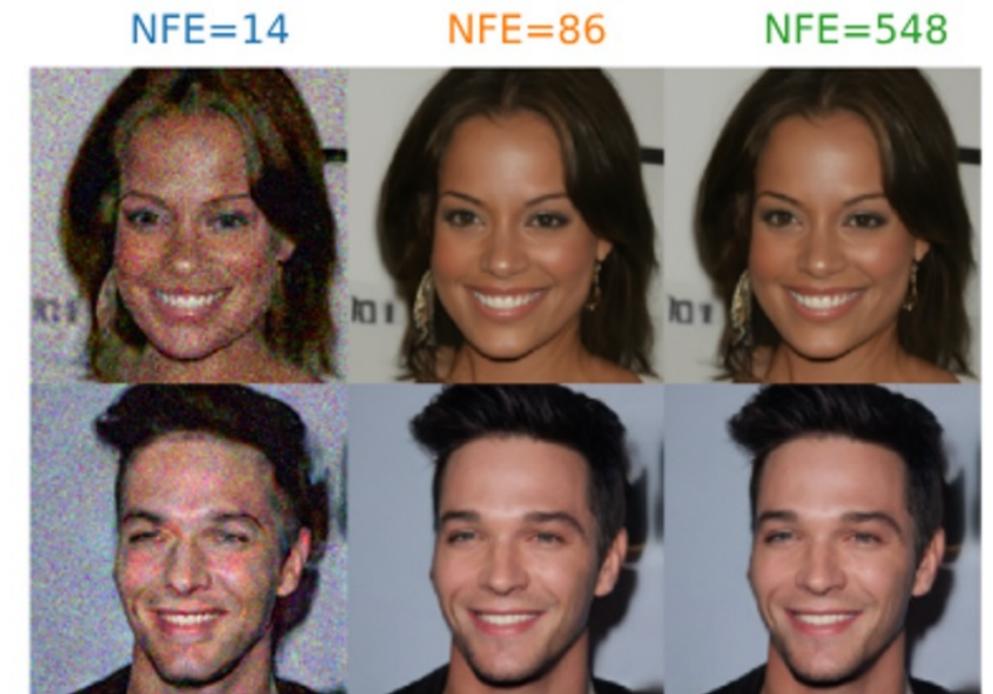
#### 精度と効率のトレードオフ Trade-off between accuracy and efficiency

精度と計算効率のトレードオフを明示的に調整できる。エラー許容度を大きく設定すると、関数評価の回数を90%以上削減できる場合がある。視覚的なサンプルの品質に影響を与えることなく、計算資源の効率的な利用が可能。

The tradeoff between accuracy and computational efficiency can be explicitly adjusted. A large error tolerance may reduce the number of function evaluations by 90% or more. Efficient use of computational resources without affecting visual sample quality.

ブラックボックスODEソルバー：

内部の具体的な実装詳細に依存せず、指定された入力に対して適切にODE（常微分方程式）を解くことができるソルバー



# 4.4 Architecture improvement

## three different metrics

### NLL(Negative Log-Likelihood)

生成モデルの性能の評価。低いほどモデルを表現

Evaluate the performance of the generative model. The smaller the value, the more representative the model

### FID(Frechet Inception Distance)

生成する画像を評価するための指標。低いほど実画像に近い。

An index for evaluating the image to be generated. The smaller the value, the closer to the real image.

### IS(Inception Score)

生成画像の品質と多様性の評価。高いほど高品質で多様

Evaluation of the quality and diversity of the generated images. The higher the value, the higher the quality and diversity.

Table2

- 最終トレーニングチェックポイントの結果。
- サンプルはブラックボックスODEソルバーを用いて生成。

Table 2: NLLs and FIDs (ODE) on CIFAR-10.

Model	NLL Test ↓	FID ↓
RealNVP (Dinh et al., 2016)	3.49	-
iResNet (Behrmann et al., 2019)	3.45	-
Glow (Kingma & Dhariwal, 2018)	3.35	-
MintNet (Song et al., 2019b)	3.32	-
Residual Flow (Chen et al., 2019)	3.28	46.37
FFJORD (Grathwohl et al., 2018)	3.40	-
Flow++ (Ho et al., 2019)	3.29	-
DDPM ( $L$ ) (Ho et al., 2020)	$\leq 3.70^*$	13.51
DDPM ( $L_{\text{simple}}$ ) (Ho et al., 2020)	$\leq 3.75^*$	3.17
DDPM	3.28	3.37
DDPM cont. (VP)	3.21	3.69
DDPM cont. (sub-VP)	3.05	3.56
DDPM++ cont. (VP)	3.16	3.93
DDPM++ cont. (sub-VP)	3.02	3.16
DDPM++ cont. (deep, VP)	3.13	3.08
DDPM++ cont. (deep, sub-VP)	<b>2.99</b>	<b>2.92</b>

Table3

- 最も低いFIDを持つチェックポイントの結果。
- サンプルはPCサンプラーを用いて生成。

Table 3: CIFAR-10 sample quality.

Model	FID↓	IS↑
<b>Conditional</b>		
BigGAN (Brock et al., 2018)	14.73	9.22
StyleGAN2-ADA (Karras et al., 2020a)	<b>2.42</b>	<b>10.14</b>
<b>Unconditional</b>		
StyleGAN2-ADA (Karras et al., 2020a)	2.92	9.83
NCSN (Song & Ermon, 2019)	25.32	8.87 ± .12
NCSNv2 (Song & Ermon, 2020)	10.87	8.40 ± .07
DDPM (Ho et al., 2020)	3.17	9.46 ± .11
DDPM++	2.78	9.64
DDPM++ cont. (VP)	2.55	9.58
DDPM++ cont. (sub-VP)	2.61	9.56
DDPM++ cont. (deep, VP)	2.41	9.68
DDPM++ cont. (deep, sub-VP)	2.41	9.57
NCSN++	2.45	9.73
NCSN++ cont. (VE)	2.38	9.83
NCSN++ cont. (deep, VE)	<b>2.20</b>	<b>9.89</b>

# 4.4 Architecture improvement

## Main Architectures

NCSN++ is the optimal architecture designed for VE SDE.

DDPM++ is the optimal architecture designed for VP and sub-VP SDE.

## Comparison of Results

VE SDEs generally provide better sample quality than VP/sub-VP SDEs, but it has been empirically observed that their actual likelihoods are worse than their VP/sub-VP SDE counterparts.

Sample quality	Likelihood
VE SDEs	VP, sub-VP SDEs

Better SDE

NLL(Negative Log-Likelihood)

生成モデルの性能の評価。  
低いほどモデルを表現

FID(Frechet Inception Distance)

生成する画像を評価するための  
指標。低いほど実画像に近い。

IS(Inception Score)

生成画像の品質と多様性の評  
価。高いほど高品質で多様

Table 2: NLLs and FIDs (ODE) on CIFAR-10.

Model	NLL Test ↓	FID ↓
RealNVP (Dinh et al., 2016)	3.49	-
iResNet (Behrmann et al., 2019)	3.45	-
Glow (Kingma & Dhariwal, 2018)	3.35	-
MintNet (Song et al., 2019b)	3.32	-
Residual Flow (Chen et al., 2019)	3.28	46.37
FFJORD (Grathwohl et al., 2018)	3.40	-
Flow++ (Ho et al., 2019)	3.29	-
DDPM ( $L$ ) (Ho et al., 2020)	$\leq 3.70^*$	13.51
DDPM ( $L_{\text{simple}}$ ) (Ho et al., 2020)	$\leq 3.75^*$	3.17
DDPM	3.28	3.37
DDPM cont. (VP)	3.21	3.69
DDPM cont. (sub-VP)	3.05	3.56
DDPM++ cont. (VP)	3.16	3.93
DDPM++ cont. (sub-VP)	3.02	3.16
DDPM++ cont. (deep, VP)	3.13	3.08
DDPM++ cont. (deep, sub-VP)	<b>2.99</b>	<b>2.92</b>

Table 3: CIFAR-10 sample quality.

Model	FID↓	IS↑
<b>Conditional</b>		
BigGAN (Brock et al., 2018)	14.73	9.22
StyleGAN2-ADA (Karras et al., 2020a)	<b>2.42</b>	<b>10.14</b>
<b>Unconditional</b>		
StyleGAN2-ADA (Karras et al., 2020a)	2.92	9.83
NCSN (Song & Ermon, 2019)	25.32	8.87 ± .12
NCSNv2 (Song & Ermon, 2020)	10.87	8.40 ± .07
DDPM (Ho et al., 2020)	3.17	9.46 ± .11
DDPM++	2.78	9.64
DDPM++ cont. (VP)	2.55	9.58
DDPM++ cont. (sub-VP)	2.61	9.56
DDPM++ cont. (deep, VP)	2.41	9.68
DDPM++ cont. (deep, sub-VP)	2.41	9.57
NCSN++	2.45	9.73
NCSN++ cont. (VE)	2.38	9.83
NCSN++ cont. (deep, VE)	<b>2.20</b>	<b>9.89</b>

# 4.4 Architecture improvement

## Best Model for Sample Quality

**NCSN++ cont. (deep, VE)** doubled the network depth and set new records for both Inception Score and FID in unconditional generation on CIFAR-10.

## Best Model for Likelihood:

**DDPM++ cont. (deep, sub-VP)** doubled the network depth and achieved a log-likelihood of 2.99 bits/dim on uniformly dequantized CIFAR-10 using the continuous objective in Eq. (7).

NLL(Negative Log-Likelihood)

生成モデルの性能の評価。  
低いほどモデルを表現

FID(Frechet Inception Distance)

生成する画像を評価するための  
指標。低いほど実画像に近い。

IS(Inception Score)

生成画像の品質と多様性の評  
価。高いほど高品質で多様

Table 2: NLLs and FIDs (ODE) on CIFAR-10.

Model	NLL Test ↓	FID ↓
RealNVP (Dinh et al., 2016)	3.49	-
iResNet (Behrmann et al., 2019)	3.45	-
Glow (Kingma & Dhariwal, 2018)	3.35	-
MintNet (Song et al., 2019b)	3.32	-
Residual Flow (Chen et al., 2019)	3.28	46.37
FFJORD (Grathwohl et al., 2018)	3.40	-
Flow++ (Ho et al., 2019)	3.29	-
DDPM ( $L$ ) (Ho et al., 2020)	$\leq 3.70^*$	13.51
DDPM ( $L_{\text{simple}}$ ) (Ho et al., 2020)	$\leq 3.75^*$	3.17
DDPM	3.28	3.37
DDPM cont. (VP)	3.21	3.69
DDPM cont. (sub-VP)	3.05	3.56
DDPM++ cont. (VP)	3.16	3.93
DDPM++ cont. (sub-VP)	3.02	3.16
DDPM++ cont. (deep, VP)	3.13	3.08
DDPM++ cont. (deep, sub-VP)	<b>2.99</b>	<b>2.92</b>

Table 3: CIFAR-10 sample quality.

Model	FID ↓	IS ↑
<b>Conditional</b>		
BigGAN (Brock et al., 2018)	14.73	9.22
StyleGAN2-ADA (Karras et al., 2020a)	<b>2.42</b>	<b>10.14</b>
<b>Unconditional</b>		
StyleGAN2-ADA (Karras et al., 2020a)	2.92	9.83
NCSN (Song & Ermon, 2019)	25.32	8.87 ± .12
NCSNv2 (Song & Ermon, 2020)	10.87	8.40 ± .07
DDPM (Ho et al., 2020)	3.17	9.46 ± .11
DDPM++	2.78	9.64
DDPM++ cont. (VP)	2.55	9.58
DDPM++ cont. (sub-VP)	2.61	9.56
DDPM++ cont. (deep, VP)	2.41	9.68
DDPM++ cont. (deep, sub-VP)	2.41	9.57
NCSN++	2.45	9.73
NCSN++ cont. (VE)	2.38	9.83
NCSN++ cont. (deep, VE)	<b>2.20</b>	<b>9.89</b>

# 5. Controllable generation

# 5 Controllable generation

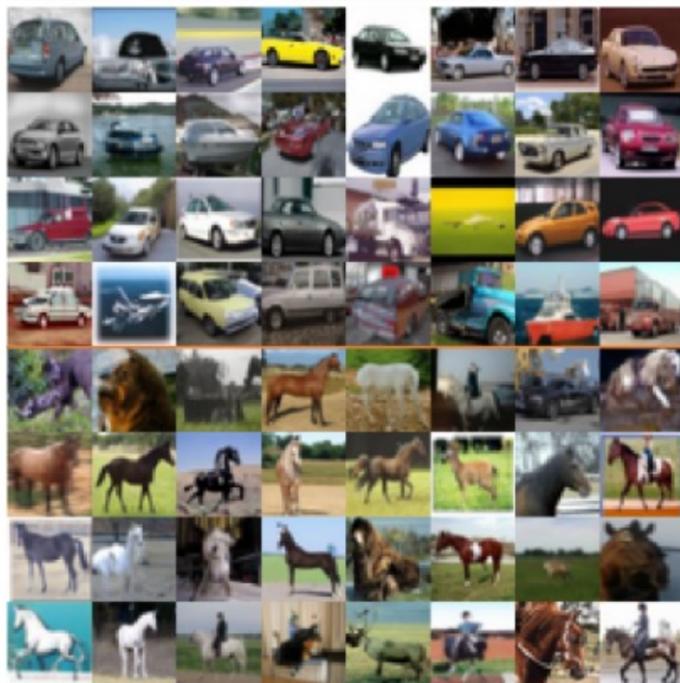
## Formulation of Conditional Reverse-time SDE

Conditional Reverse-time SDE  $p_t(\mathbf{x}(t) | \mathbf{y})$  からサンプリングできる

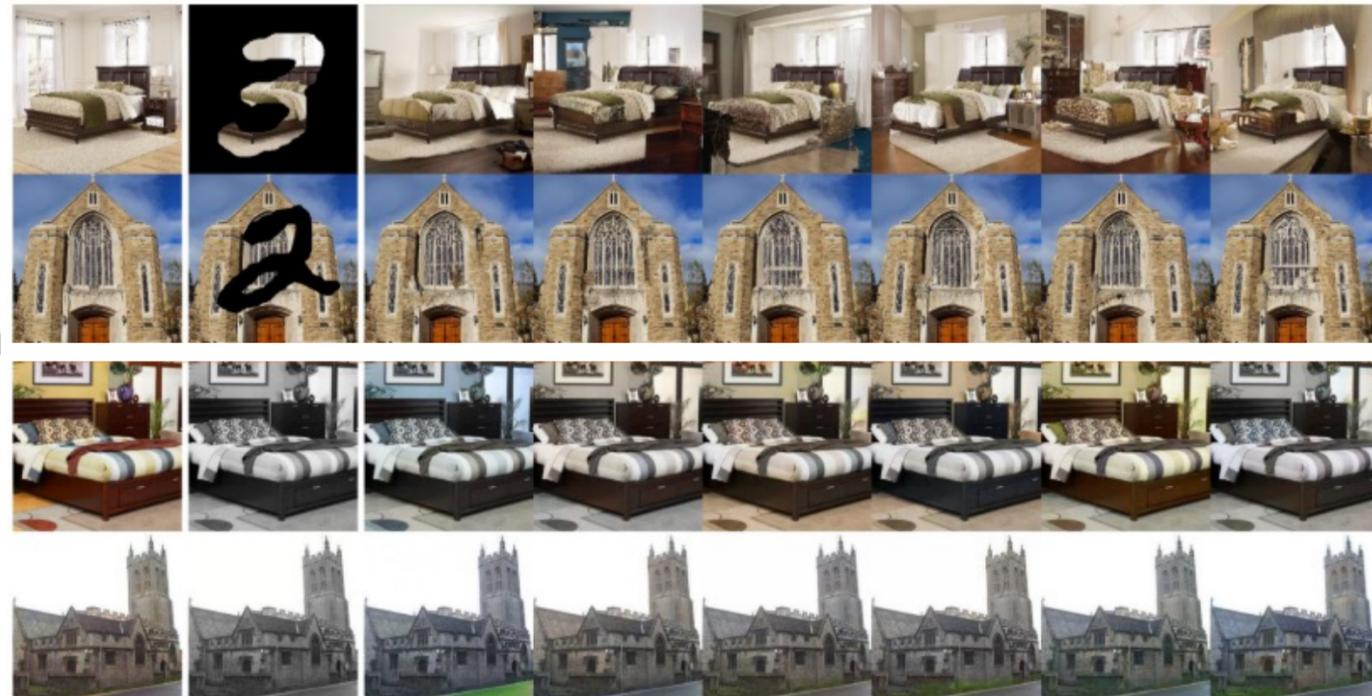
$$d\mathbf{x} = \{\mathbf{f}(\mathbf{x}, t) - g(t)^2 [\nabla_{\mathbf{x}} \log p_t(\mathbf{x}) + \nabla_{\mathbf{x}} \log p_t(\mathbf{y} | \mathbf{x})]\} dt + g(t) d\bar{\mathbf{w}}$$



提案したモデルは  $p_0$  からだけではなく  $p_0(\mathbf{x}(0) | \mathbf{y})$  からデータサンプルが生成できる。  
( $p_t(\mathbf{y} | \mathbf{x}(t))$  が既知の場合)



クラス条件付き生成  
class-conditional generation



補完  
Inpainting

カラー化  
colorization

# 6. Conclusion

# 6 Conclusion

## Features and Benefits of the Framework

Improved Understanding  
of Existing Approaches

New Sampling Algorithms

Exact Likelihood Computation

Uniquely Identifiable  
Encoding

Latent Code Manipulation

New Conditional  
Generation Abilities

# 6 Conclusion

## Current Limitations

### Slower Sampling Speed

提案されたサンプリング手法は結果を改善し、効率的なサンプリングを可能にするが、サンプリング速度はGAN (Goodfellow et al., 2014) よりも遅い

## Future Research Directions

### Combining Stable Learning of Score-Based Generative Models with Fast Sampling of GANs

スコアベースの生成モデルの安定した学習と、GANのような高速サンプリングを統合する方法を見つける。

### Automatic Selection and Tuning of Hyperparameters

スコア関数の利用で使用可能なサンプラーの種類が広がり、多くのハイパーパラメータが導入される。これらを自動的に選択し、調整するための改善された方法が求められる。

### Investigation on the Merits and Limitations of Various Samplers

さまざまなサンプラーのメリットと限界について、より広範な調査が必要。これにより、特定のタスクやデータセットに最適なサンプリング手法を選択できるようになる。

# 所感

- とても難しく苦勞したが、機械学習への理解が深まって良かった。
- 最後の結論部分でサンプリング速度について言及されていたが、他の部分で速度に関してあまり言及されていなかったためもう少し詳細に知りたかった。
- 生成モデルは画像以外でも音声やグラフに使えるとあったが、それに対しての例が無かったので、その辺ももっと学べると面白そう。
- 様々なモデルが出てきたが、説明がないモデルもあったため、そのようなモデルももう少し、比較してみたい。