

Generative Deep Learning

2023/11/10 (金)

本日は画像系の深層生成モデルの歴史と概要をご紹介します

わかること 🧐

- 深層生成モデルの歴史と分類
- 基本的な画像系深層生成モデルの
 - アーキテクチャ概要
 - 性質・特色
 - 理論的背景

Advanced

わからない (詳しくない) こと 🧐

- Neural Network (NN) による深層学習の基本的な仕組み
- 各深層生成モデルの具体的な中身
- 各深層生成モデルの発展とそのユースケース

概要部分は学部初級レベルの数学力で理解できる想定です！

ただし私は deep learning の専門家でも生成モデルの専門家でもないので
説明に不明瞭・不完全な部分があったらすみません... 🧐

Contents

1. What is Generative Modeling?
2. History & Taxonomy of Generative Models
3. Image Generative Models
4. Summary

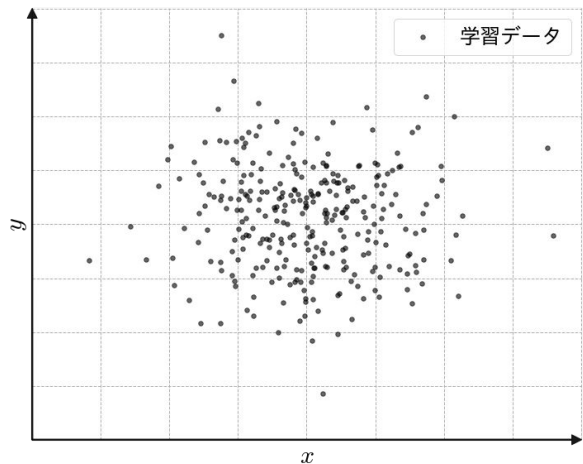
Contents

1. What is Generative Modeling?
2. History & Taxonomy of Generative Models
3. Image Generative Models
4. Summary

1. What is Generative Modeling? (1/3)

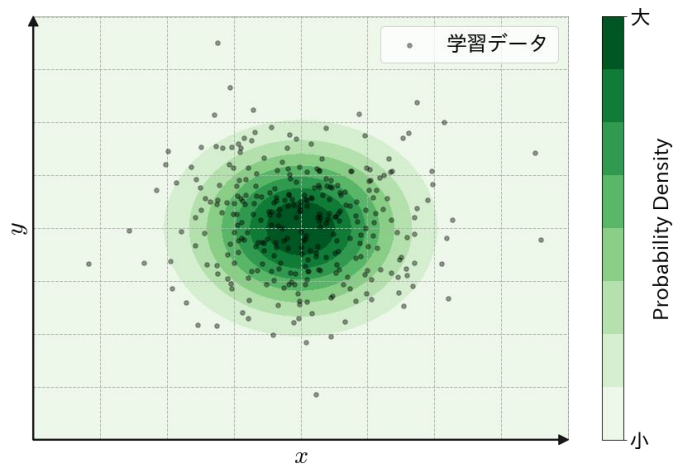
生成モデルの目標は、与えられた学習データ $D = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}\}$ を用いてそのデータを生成している**確率分布** $p(\mathbf{x})$ を**推定**することである

学習データ $D = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(300)}\}$



推定

確率分布
(正規分布) $p(\mathbf{x}) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left[-\frac{(\mathbf{x} - \mu)^2}{2\sigma^2} \right]$

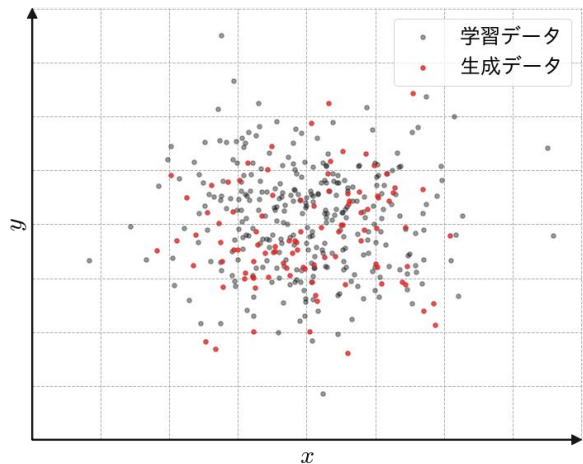


1. What is Generative Modeling? (2/3)

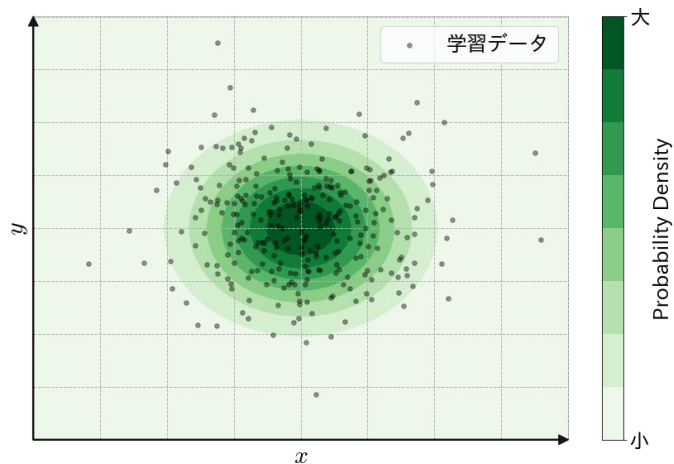
例えば下の例の場合、正規分布を構成する 2 つのパラメータ (平均と分散) $\mathbf{z} = (\mu, \sigma)$ が推定できれば、その分布を用いて**学習データと似て非なるデータを生成**できる

生成データ $D = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(100)}\}$

確率分布
(正規分布)
$$p(\mathbf{x}) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left[-\frac{(\mathbf{x} - \mu)^2}{2\sigma^2} \right]$$



生成



1. What is Generative Modeling? (3/3)

データによる確率分布のパラメータの推定には、一般に**確率分布の“尤もらしさ”を最大化する最尤法**という手法が用いられる※

最尤法とは

ある確率分布 $p(\mathbf{x}; \theta)$ から独立に抽出されたデータセット $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}\}$ が与えられたとき、尤度の対数を取った対数尤度は、

$$L(\theta) \equiv \log \prod_{i=1}^n p(\mathbf{x}^{(i)}; \theta) = \sum_{i=1}^n \log p(\mathbf{x}^{(i)}; \theta).$$

と定義される。

この**対数尤度をどう最大化するか (どう最大化するパラメータを求めるか)** が深層生成モデルの大きな違いになっている。

例. 正規分布の場合

パラメータ未知のある正規分布からデータ $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(n)}\}$ が生成された場合、対数尤度は、

$$\begin{aligned} L(\mu, \sigma) &= \sum_{i=1}^n \log \left[\frac{1}{\sqrt{2\pi}\sigma} \exp \left(-\frac{(x^{(i)} - \mu)^2}{2\sigma^2} \right) \right], \\ &= -\log(\sqrt{2\pi}\sigma) - \sum_{i=1}^n \frac{(x^{(i)} - \mu)^2}{2\sigma^2}. \end{aligned}$$

例えば平均については $\frac{\partial L(\mu, \sigma)}{\partial \mu} = 0$ より、

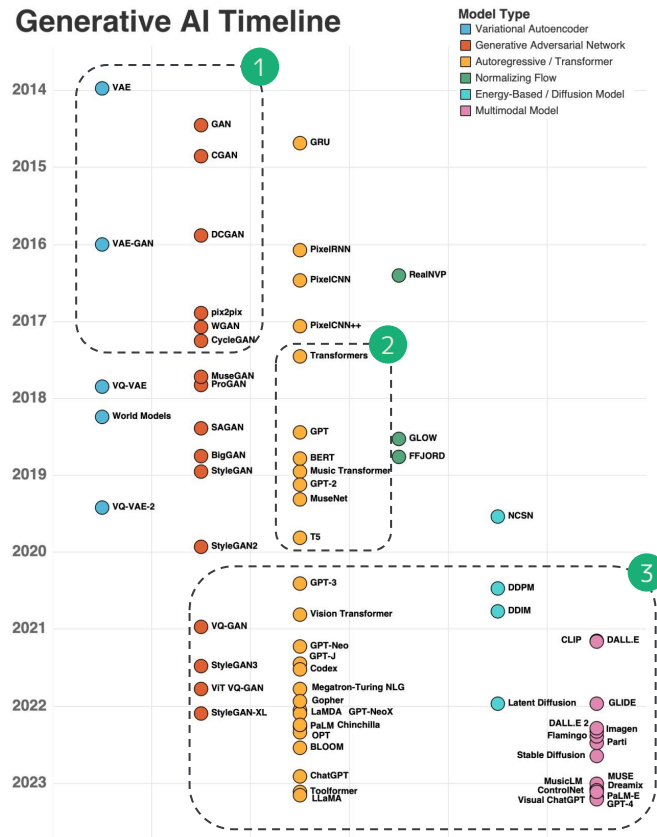
$$\mu = \frac{1}{n} \sum_{i=1}^n x^{(i)}.$$

Contents

1. What is Generative Modeling?
- 2. History & Taxonomy of Generative Models**
3. Image Generative Models
4. Summary

2. History of Generative Models

Generative AI Timeline



1

2014 - 2017
The VAE & GAN Era

2013/12 の VAE の開発により初めて深層学習により人間の顔などの複雑な画像生成が可能となり、それに続いた GAN の 2 種類を基本としてモデルの改善が行われる

2

2018 - 2019
Transformer Era

2017/12 の Attention 機構の開発により BERT や GPT などの優れた言語モデルが開発されると同時に、それらを応用した音楽生成モデルなども登場する

3

2020 - 2022
The Big Model Era

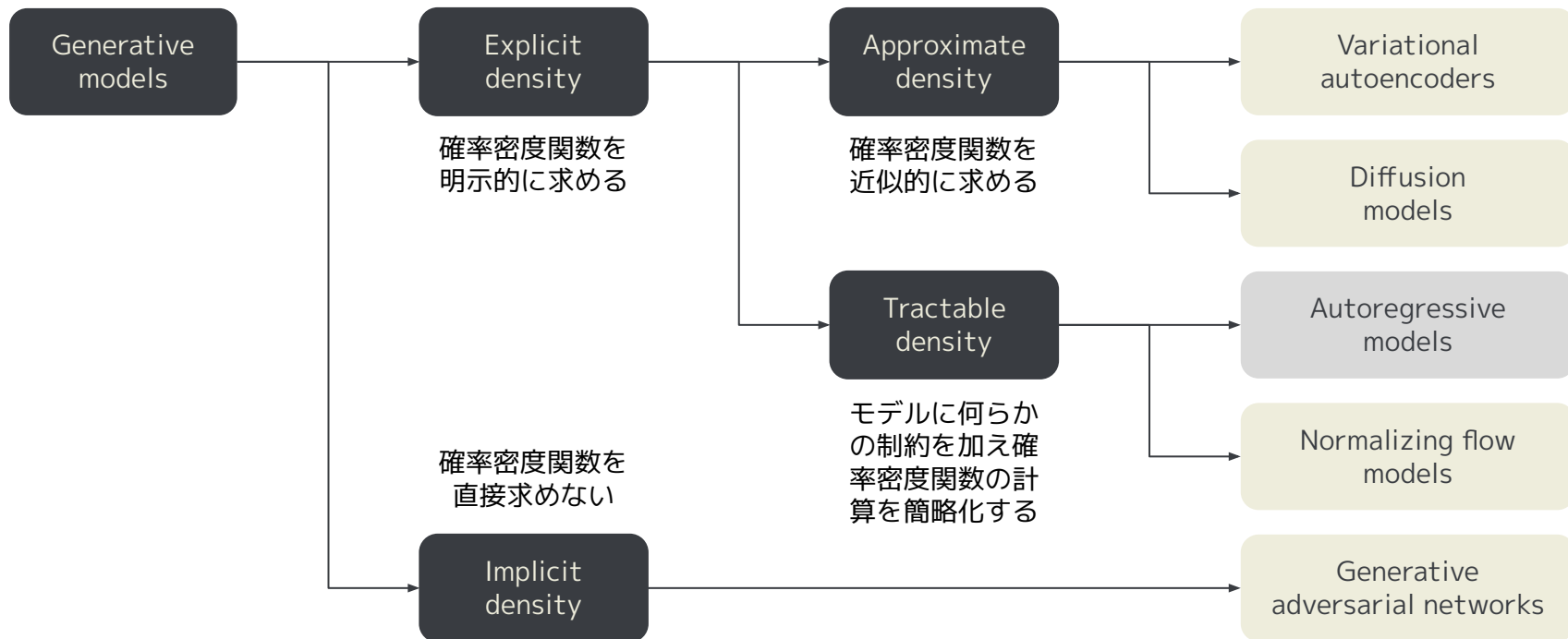
これまでに開発された複数の手法を合わせてより複雑・大規模なモデルが開発されるようになる

参考文献など

- D. P. Kingma, M. Welling, 2013, [Auto-Encoding Variational Bayes](#)
- I. J. Goodfellow, et. al., 2014, [Generative Adversarial Networks](#)
- A. Vaswani, et. al., 2017, [Attention Is All You Need](#)

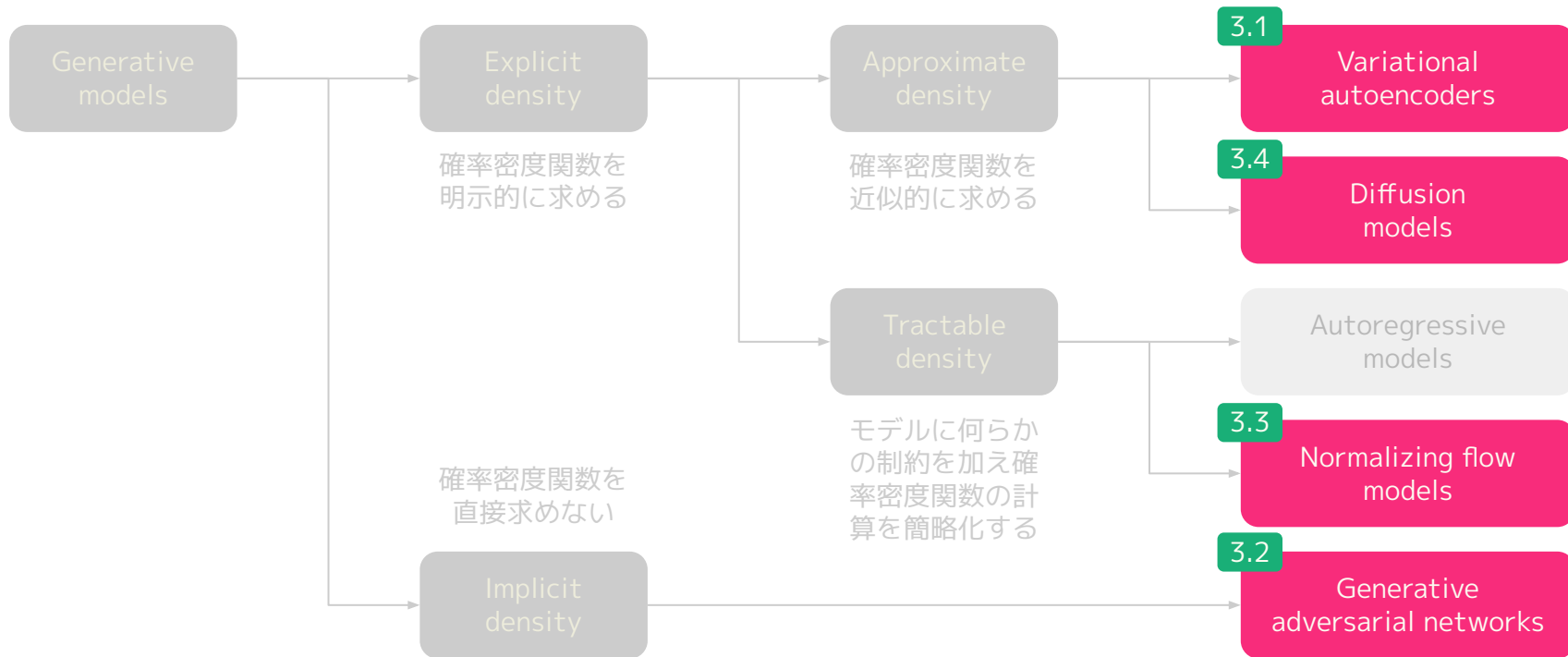
2. Taxonomy of Generative Models

本日は各深層生成モデルを以下の 5 つに分類し、それぞれを概観する



2. Taxonomy of Generative Models

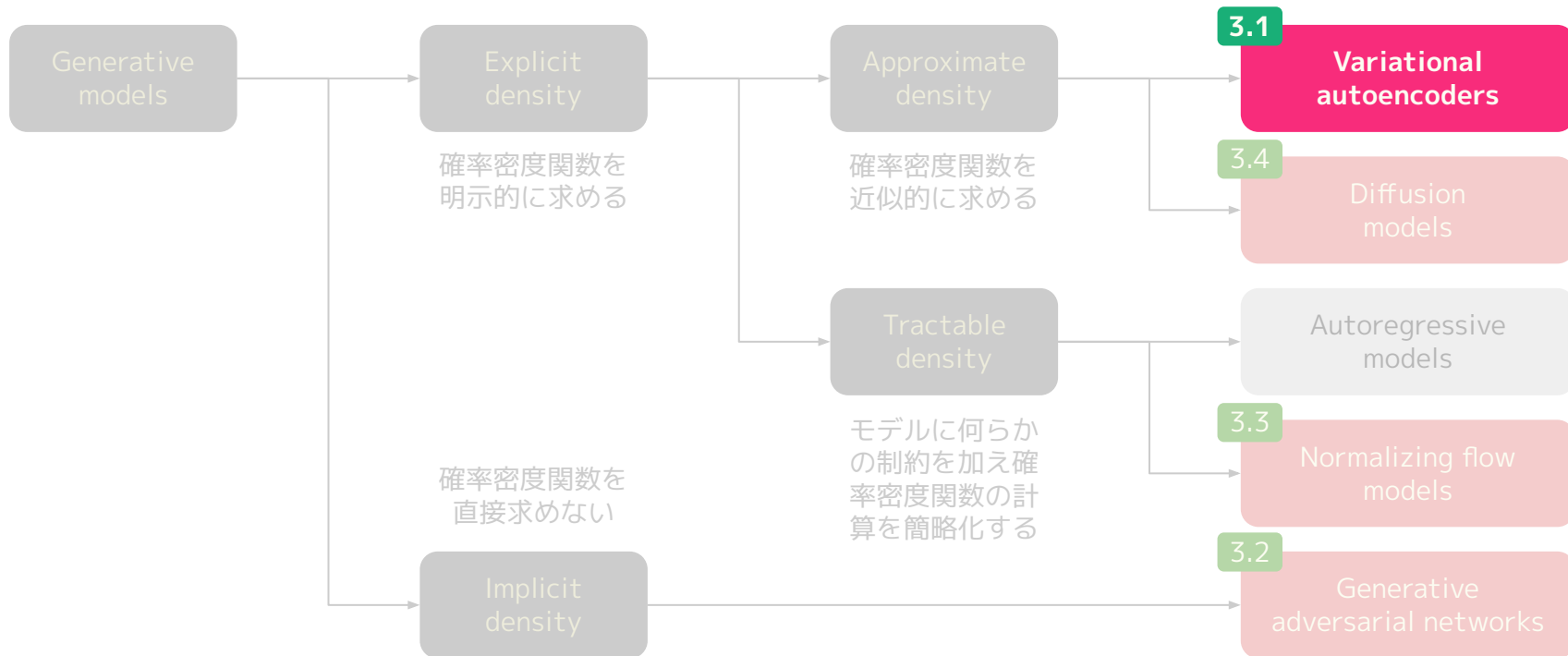
本日は各深層生成モデルを以下の 5 つに分類し、それぞれを概観する



Contents

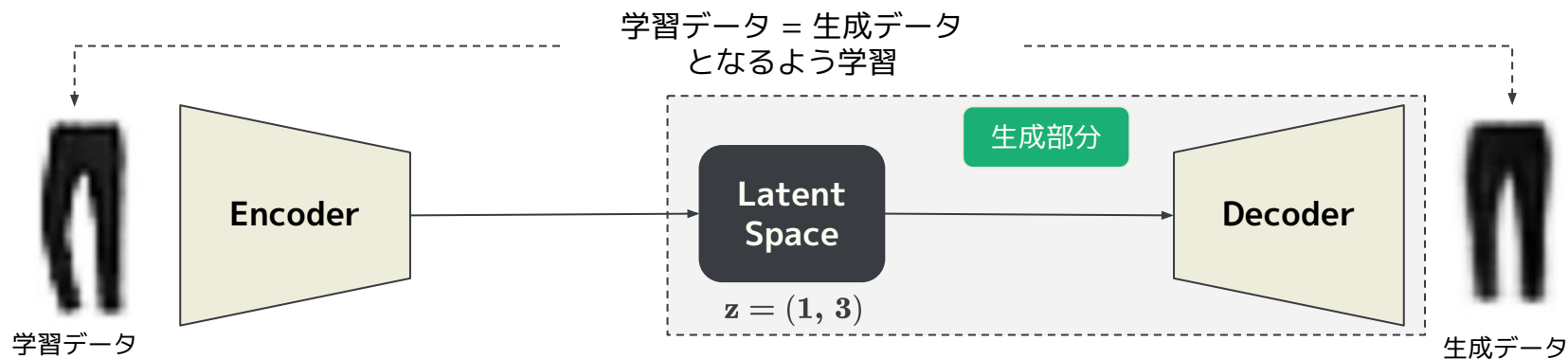
1. What is Generative Modeling?
2. History & Taxonomy of Generative Models
- 3. Image Generative Models**
4. Summary

3. Image Generative Models



3.1. Autoencoders (AE) - Architecture

AE は学習データの低次元空間への写像を介して画像生成モデルを獲得する



Encoding

NN を用いて画像などの高次元データをより低次元データへと圧縮する

Latent Space

多様体仮説※に基づき高次元学習データが埋め込まれている低次元空間

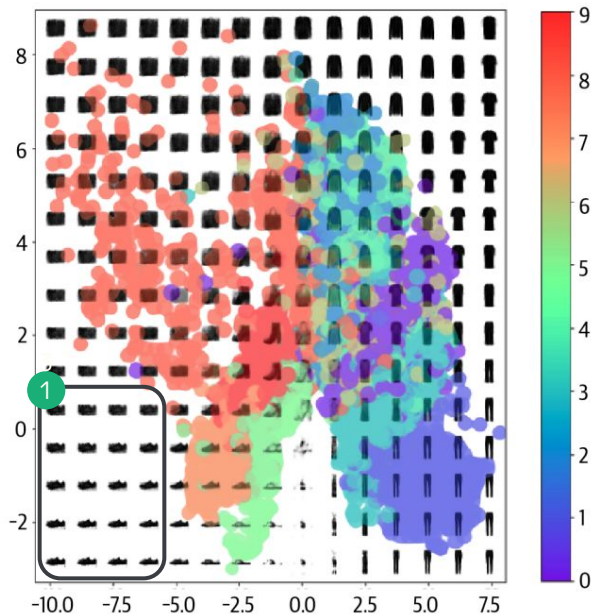
Decoding

NN を用いて圧縮された低次元データを元の学習データへと復元する

3.1. Autoencoders (AE) - Problems

9つの種類の洋服の画像が含まれる [Fashion MNIST](#) のデータセットを用いて AE を学習させると、洋服の画像生成という観点で2つの問題が存在することがわかる

ID	Clothing label
0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot



1

学習データが存在しない点における生成の問題

学習データが存在しない点を入力した場合、近傍の学習データに似た画像が生成される保証もそもそも洋服の画像が生成される保証もない

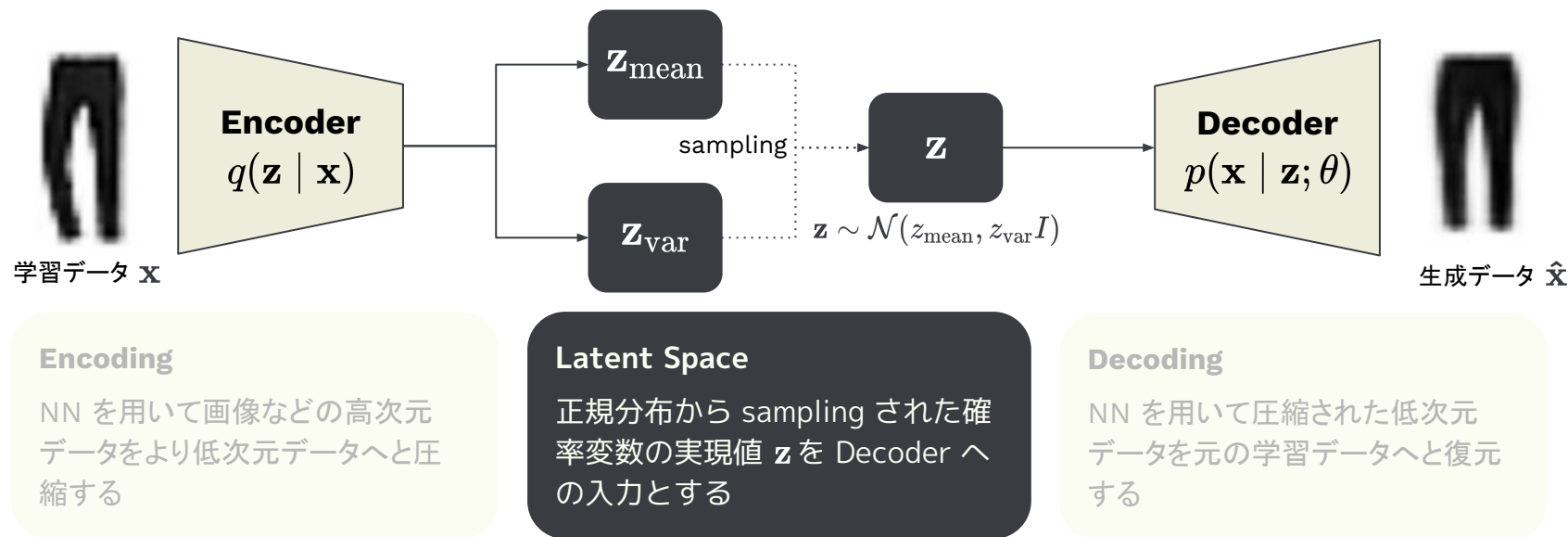
2

潜在空間からのサンプリングができない問題

画像を生成するために潜在空間からサンプリングしようとしてもその定義域すらわからない

3.1. Variational Autoencoders (VAE) - Architecture

VAE では**潜在変数を正規分布に従う確率変数とすること**でサンプリングを可能にし、学習データの近傍が“似た”データを生成することを保証する



3.1. Variational Autoencoders (VAE) - What is “Variational”?

VAE は AE の発展形というよりは Bayes における変分推論の応用として開発された

変分推論とは

\mathbf{x} から \mathbf{z} を推論する encoder $q(\mathbf{z}|\mathbf{x})$ を用いて対数尤度は以下の 2 要素に分解される*。

$$\log p(\mathbf{x}; \theta) = \log \left(\int_{\mathbf{z}} p(\mathbf{x}|\mathbf{z}; \theta) p(\mathbf{z}) d\mathbf{z} \right),$$

Evidence Lower Bound (ELBO) = $\int_{\mathbf{z}} q(\mathbf{z} | \mathbf{x}) \log \frac{p(\mathbf{x}|\mathbf{z}; \theta) p(\mathbf{z})}{q(\mathbf{z} | \mathbf{x})} d\mathbf{z}$

KL divergence (各分布の“距離”) + $\int_{\mathbf{z}} q(\mathbf{z} | \mathbf{x}) \log \frac{q(\mathbf{z} | \mathbf{x})}{p(\mathbf{z} | \mathbf{x})} d\mathbf{z}.$

対数尤度は $q(\mathbf{z}|\mathbf{x})$ に依らないため、ELBO の最大化は KL divergence の最小化 $q(\mathbf{z}|\mathbf{x}) = p(\mathbf{z}|\mathbf{x})$ を意味する。従って、**ELBO を最大化する $q(\mathbf{z}|\mathbf{x})$** を求めれば ELBO は対数尤度と等しくなると同時に decoder の学習も達成される。

Bayes における変分推論の利用

何らかの事後分布 $p(\mathbf{z}|\mathbf{x})$ を求める際に利用され、ELBO に対し以下の平均場近似と変分法を用いて計算する

$$p(\mathbf{z}|\mathbf{x}) \simeq q'(\mathbf{z}) \simeq \prod_i q'_i(\mathbf{z}_i)$$

VAE における変分推論の利用

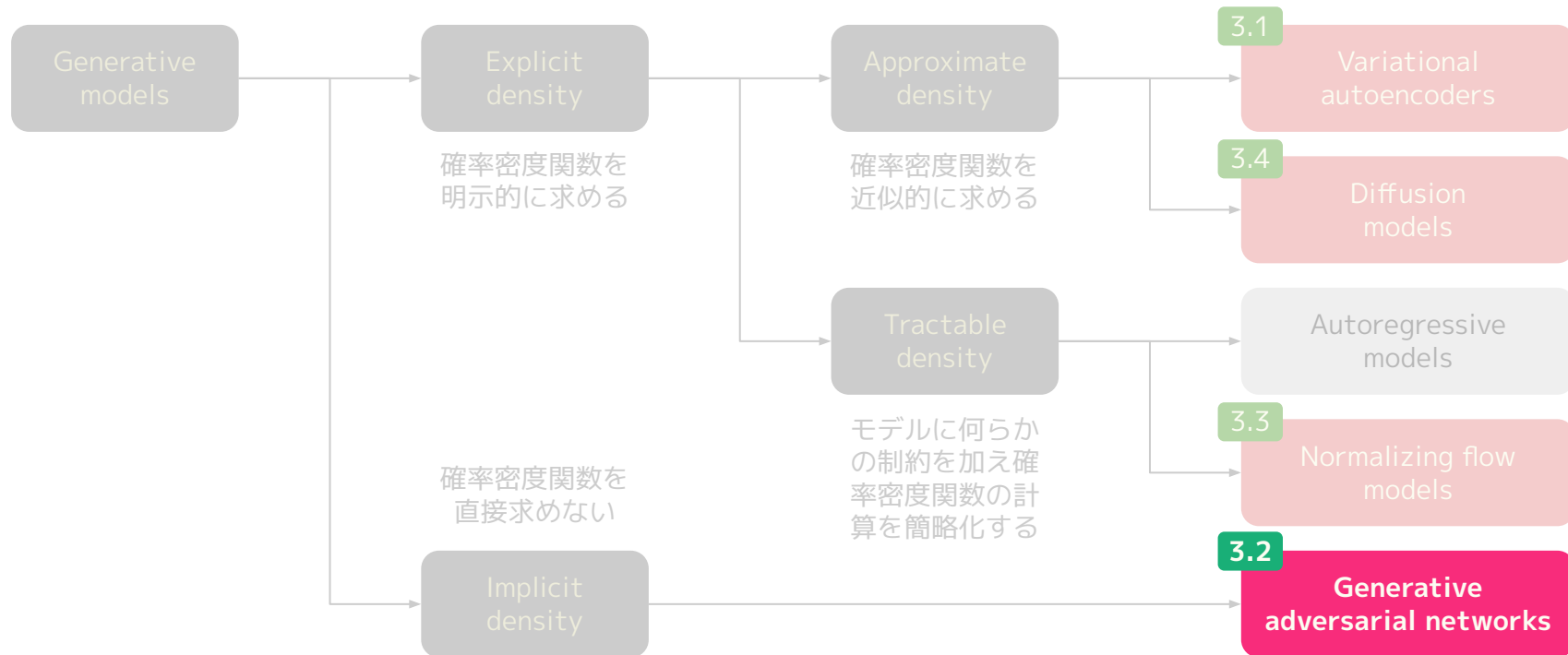
ELBO を分解した以下の式を目的関数として学習する

$$(\text{ELBO}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} [\log p(\mathbf{x} | \mathbf{z})] - KL(q(\mathbf{z} | \mathbf{x}) || p(\mathbf{z}))$$

ベルヌーイ分布を仮定すると出力 と入力の cross entropy になる 正規分布

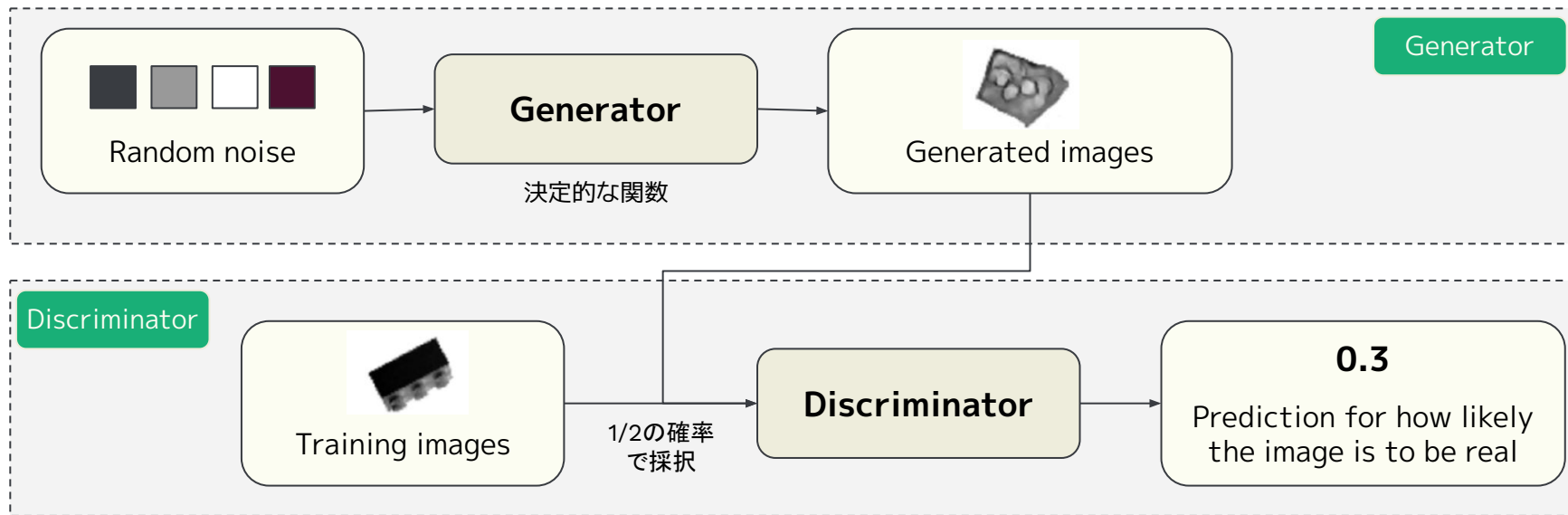
※ 式変形は色々なところで掲載されているので調べてみてください

3. Image Generative Models



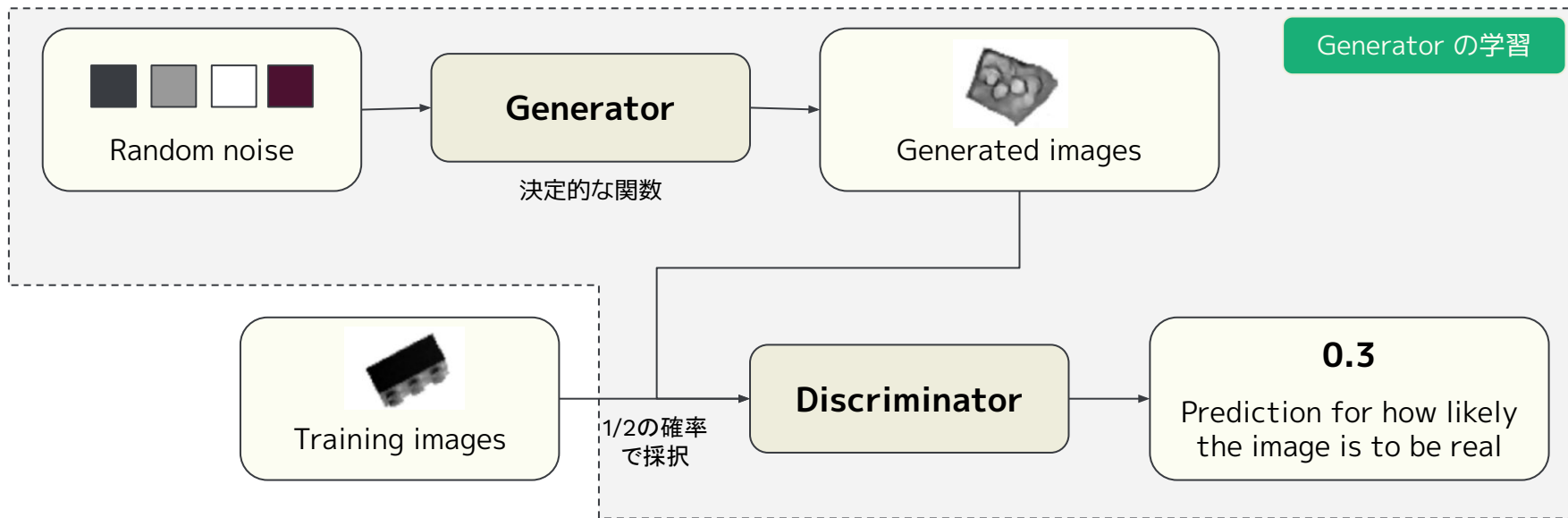
3.2. Generative Adversarial Networks (GAN) - Architecture

GAN では、生成器は識別器を騙せるよう本物そっくりのデータを生成するように、識別器は生成器のわずかな誤りも見つけられるように競合して学習する



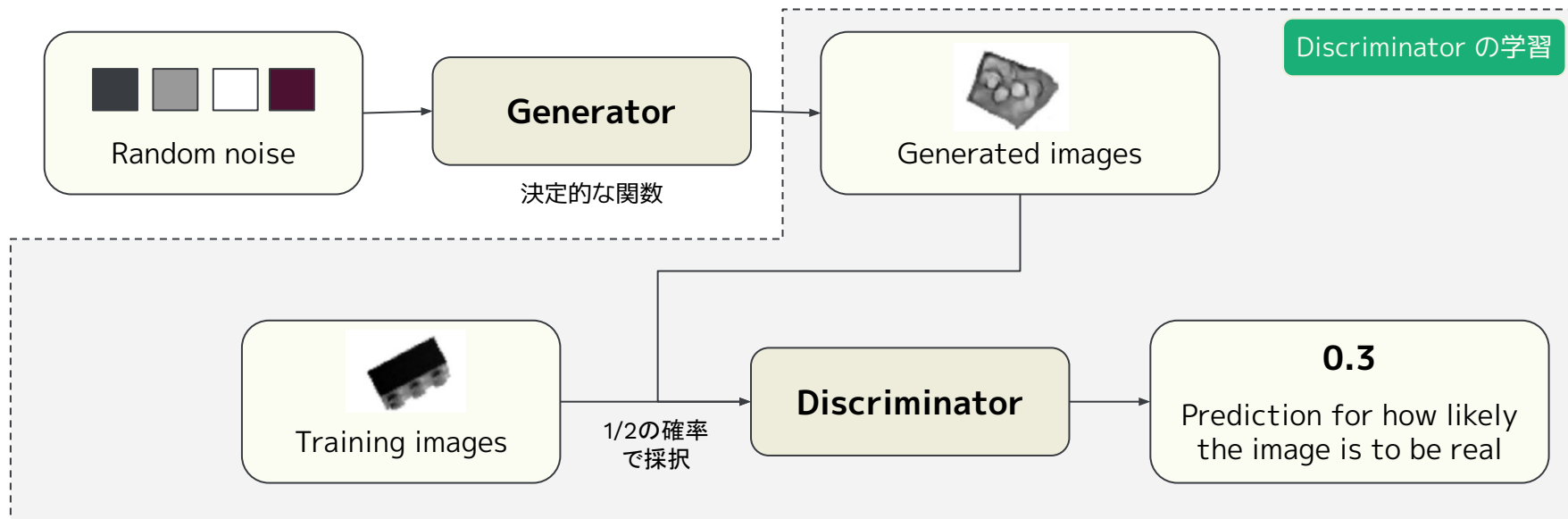
3.2. Generative Adversarial Networks (GAN) - Learning (1/3)

Generator を生成した画像が実物と判定されるように学習を進めるため、Generator の学習には Discriminator の出力をそのまま損失関数に用いる



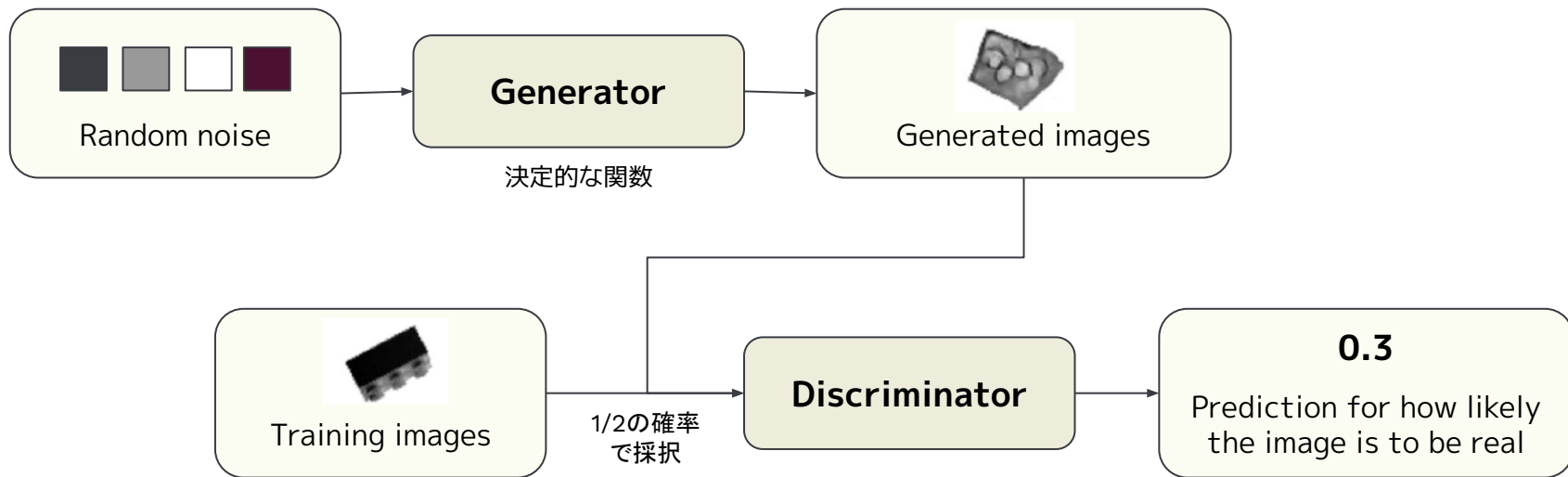
3.2. Generative Adversarial Networks (GAN) - Learning (2/3)

Discriminator は通常の 2 値分類の分類器と同様に学習を行う



3.2. Generative Adversarial Networks (GAN) - Learning (3/3)

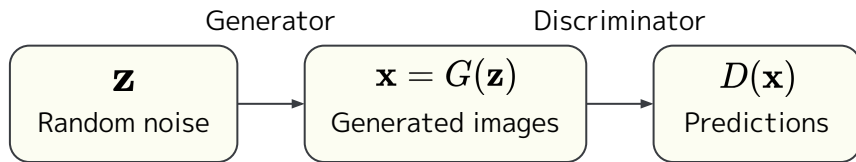
Generator/Discriminator のどちらか一方が弱くなってしまうことを防ぐため、これら 2 つの学習は同時ではなく交互に行われる



3.2. Generative Adversarial Networks (GAN) - Background

GAN は generator の生成分布 $Q(\mathbf{x})$ と訓練データのデータ分布 $P(\mathbf{x})$ の KL divergence を最小化するように学習している

GAN が最適化する目的関数



GAN が最適化している目的関数は以下。

$$L(G, D) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})} [\log (1 - D(G(\mathbf{z})))]$$

訓練データに対する discriminator の判定結果 (大きくする) 生成データに対する discriminator の判定結果 (小さくする)

学習初期から勾配を大きくするため生成器には以下の目的関数を利用。

$$L_{\text{gen}}(G, D) = -\mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})} [\log D(G(\mathbf{z}))]$$

Generator が行う最適化

左の 2 つの目的関数の和を取る。

$$L_{\text{KL}}(G, D) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})} \left[\log \frac{1 - D(G(\mathbf{z}))}{D(G(\mathbf{z}))} \right]$$

Discriminator がベイズ最適※を達成しているとき最適な識別戦略が $D^*(\mathbf{x}) = \frac{P(\mathbf{x})}{P(\mathbf{x}) + Q(\mathbf{x})}$ であることを用いると、上式における generator の最適化は KL divergence の最小化と一致する。

$$\begin{aligned} \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})} \left[\log \frac{1 - D^*(G(\mathbf{z}))}{D^*(G(\mathbf{z}))} \right] &= \mathbb{E}_{\mathbf{x} \sim Q(\mathbf{x})} \left[\log \frac{Q(\mathbf{x})}{P(\mathbf{x})} \right] \\ &= KL(Q||P) \end{aligned}$$

3.2. Generative Adversarial Networks (GAN) - Pros & Cons

VAE と異なり GAN は確率分布からのサンプリングを行わないため**高精細な画像を生成できる一方、学習が困難**※であることが知られている

GAN における学習の難点

1

Generator/Discriminator のどちらかが強力になりすぎることによって学習が進まなくなってしまう

2

Generator/Discriminator の対は常に進化していくため loss では学習の進捗がわからない

3

Weight や dropout などの hyperparameter の調節によって容易に性質が変化してしまう

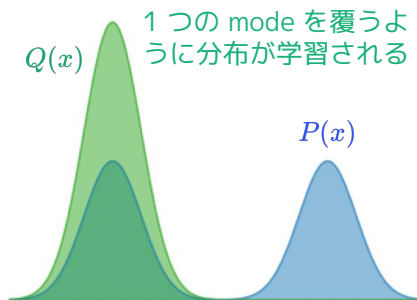
Mode collapse (① の特別な場合)

Advanced

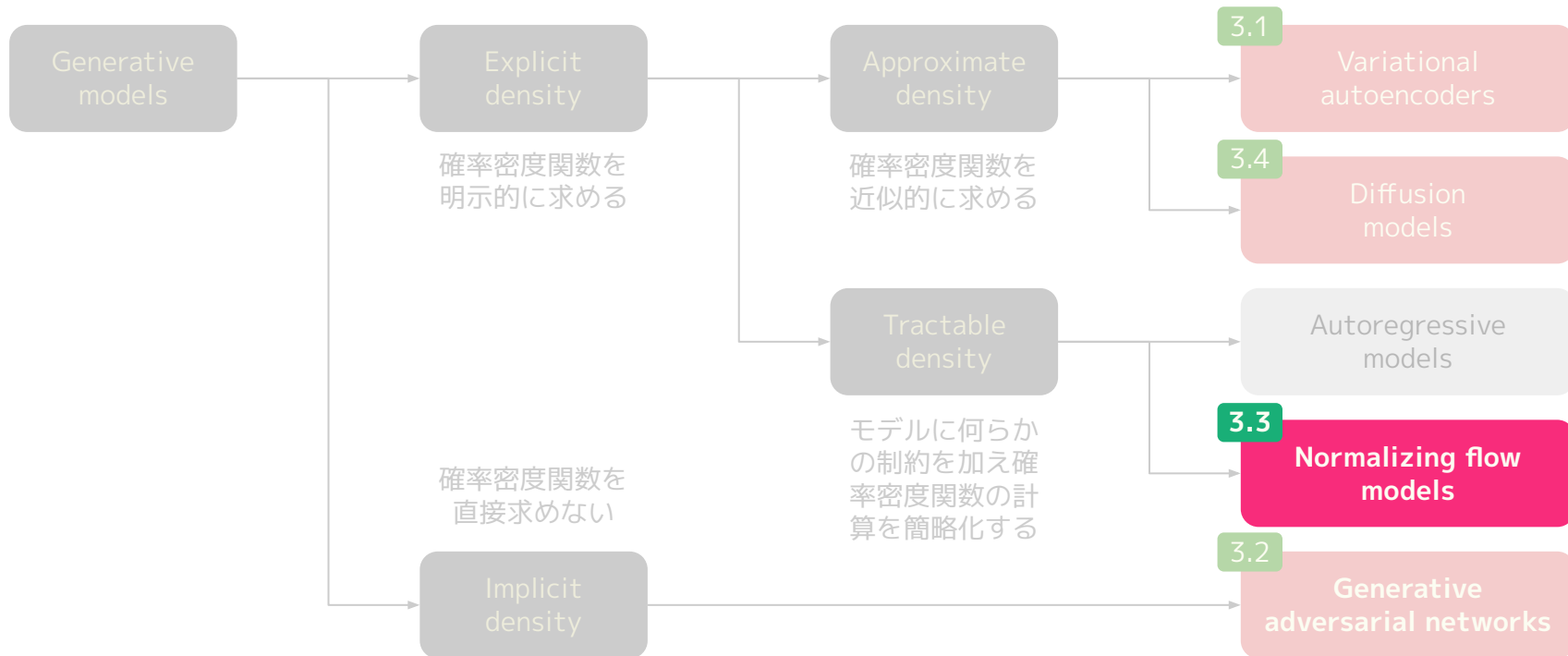
Generator が強力になり類似した特定のサンプルで discriminator を容易に騙せるようになることを mode collapse という。これは先に見た通り generator が **を最小化** するよう学習するために起こる。

$$KL(Q||P) = \int Q(x) \log \frac{Q(x)}{P(x)} dx$$

Wherever $Q(\cdot)$ has high probability, $P(\cdot)$ must also have high probability.

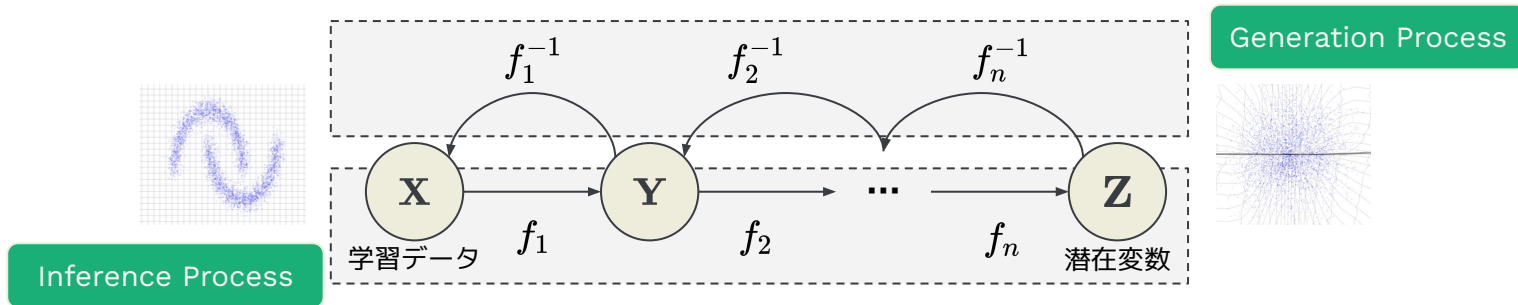


3. Image Generative Models



3.3. Normalizing Flow Models - Idea

簡単な初期分布に逆変換が容易に計算できる可逆変換を繰り返し適用することで、複雑な分布を表現すると同時に**対数尤度を直接的に計算可能にする**



確率変数の変換

確率密度関数 $p(\mathbf{x})$ は変数変換 $\mathbf{x} = f(\mathbf{z})$ の下で以下のように変換される。

$$p(\mathbf{x}) = p(\mathbf{z}) \left| \det \left(\frac{d\mathbf{z}}{d\mathbf{x}} \right) \right|$$

Jacobian の行列式

学習時における対数尤度の計算

一般に潜在変数 \mathbf{z} には正規分布などを用いるため、対数尤度を計算するには **Jacobian の行列式が簡単に求まれば良い**

$$\begin{aligned} \log p(\mathbf{x}) &= \log \left[p(\mathbf{z}) \prod_{k=0}^{n-1} \left| \det \left(\frac{d\mathbf{z}_{k+1}}{d\mathbf{z}_k} \right) \right| \right], \\ &= \log p(\mathbf{z}) + \sum_{k=0}^{n-1} \log \left| \det \left(\frac{d\mathbf{z}_{k+1}}{d\mathbf{z}_k} \right) \right|. \end{aligned}$$

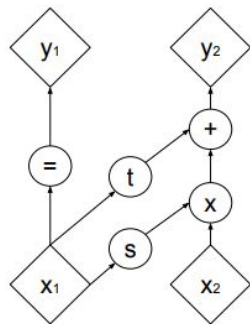
3.3. Normalizing Flow Models - Real NVP

Read NVP では Jacobian が三角行列になるため容易に尤度が計算可能

(Affine) Coupling layer の導入

D 次元ベクトル \mathbf{x} において、
1 ~ d 次元目の要素を $x_{1:d}$ と表
すことにする。

このとき、Real NVP は以下の変
換を施す coupling layer を組み
合わせて構成される。



Forward propagation

$$\begin{cases} y_{1:d} &= x_{1:d}, \\ y_{d+1:D} &= x_{d+1:D} \odot \exp(s(x_{1:d})) + t(x_{1:d}). \end{cases}$$

Coupling layer の変換関数は必要な 2 つの条件を満たす

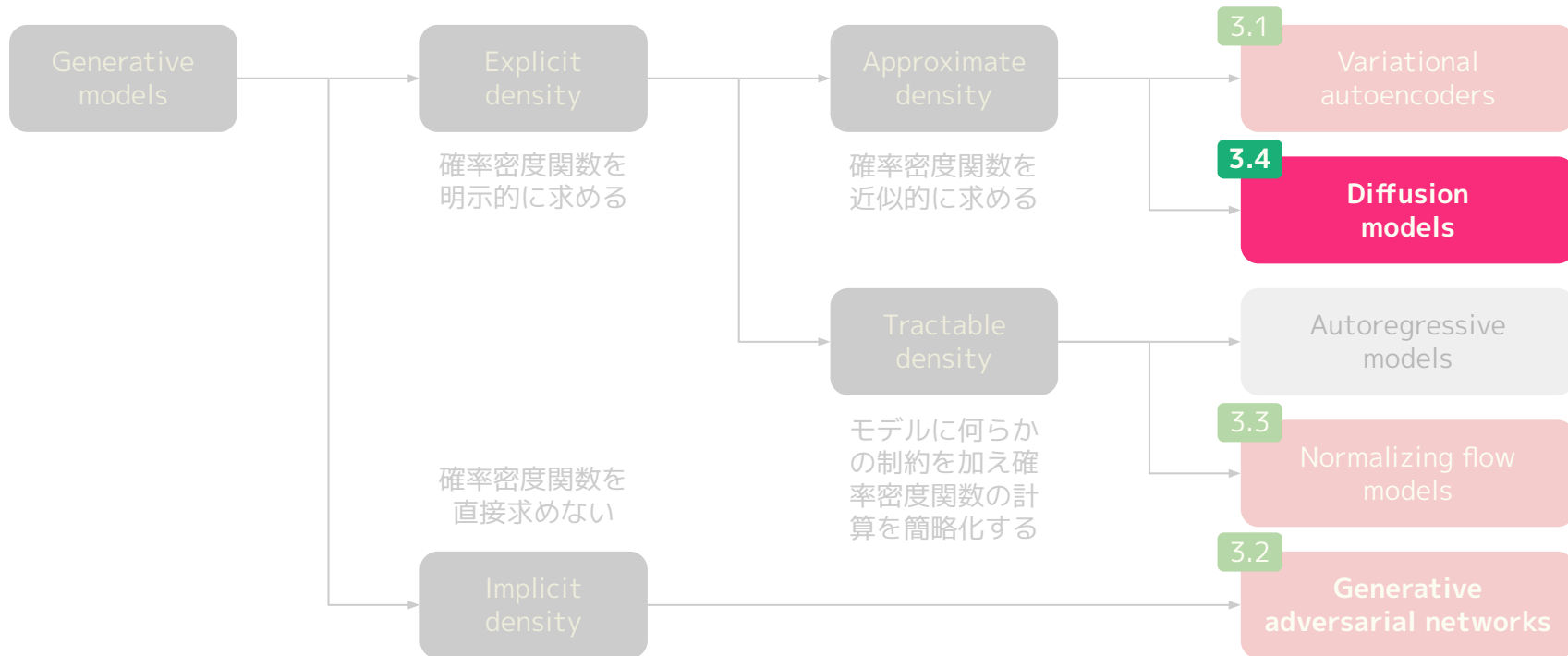
1 逆変換が容易に計算可能

$$\begin{cases} x_{1:d} &= y_{1:d}, \\ x_{d+1:D} &= (y_{d+1:D} - t(x_{1:d})) \odot \exp(-s(x_{1:d})). \end{cases}$$

2 Jacobian の行列式が容易に計算可能

$$\begin{aligned} \frac{\partial \mathbf{y}}{\partial \mathbf{x}} &= \begin{pmatrix} \mathbb{I}_d & \mathbf{0} \\ \frac{\partial y_{d+1:D}}{\partial x_{1:d}} & \text{diag}[\exp(s(x_{1:d}))] \end{pmatrix} \\ \rightarrow \det\left(\frac{\partial \mathbf{y}}{\partial \mathbf{x}}\right) &= \exp\left(\sum_j s(x_{1:d})_j\right) \end{aligned}$$

3. Image Generative Models

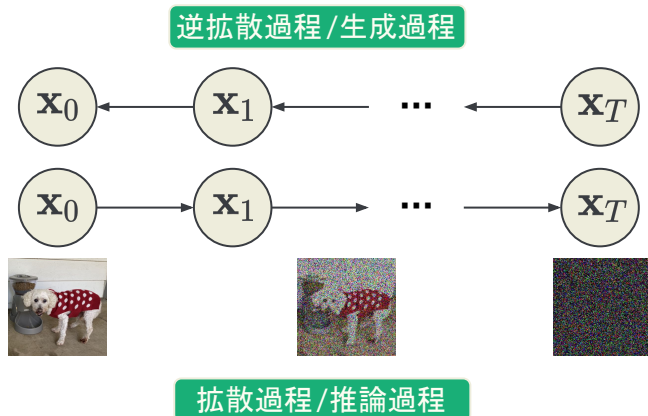


3.4. Diffusion Models - Taxonomy

Diffusion model には大きく score based model (SBM) と diffusion probabilistic model (DPM) の 2 つがあるが、ここでは DPM を紹介する

Denoising Diffusion Probabilistic Model (DDPM)

データに徐々にノイズを加えていく拡散過程を逆向きにたどる逆拡散過程でデータを生成する。



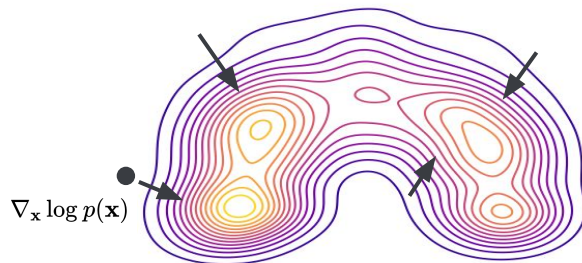
Score Based Model (SBM)

対数尤度 $\log p(\mathbf{x})$ の入力 \mathbf{x} について勾配

実は学習に使う
目的関数は同じ！

$$\mathbf{s}(\mathbf{x}) \equiv \nabla_{\mathbf{x}} \log p(\mathbf{x}) = \frac{\nabla_{\mathbf{x}} p(\mathbf{x})}{p(\mathbf{x})}$$

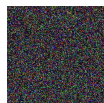
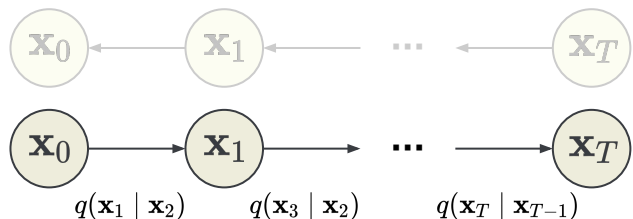
をスコアと呼び、スコアを求めることで勾配法を用いて対数尤度が大きくなる点を直接求める。



3.4. Denoising Diffusion Probabilistic Model (DDPM) (1/3)

DDPM の拡散過程では任意のステップの状態を逐次処理することなく直接計算できるため、学習が高速に行えるようになっている

逆拡散過程 / 生成過程



Gaussian ノイズを加えていく

拡散過程 / 推論過程

学習データに、徐々に大きくなる Gaussian ノイズを加えていく過程は、 $0 < \beta_1 < \dots < \beta_T < 1$ として以下で表される。

$$\begin{cases} q(\mathbf{x}_{1:T} | \mathbf{x}_0) &= \prod_{t=1}^T q(\mathbf{x}_t | \mathbf{x}_{t-1}), \\ q(\mathbf{x}_t | \mathbf{x}_{t-1}) &= \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t I). \end{cases}$$

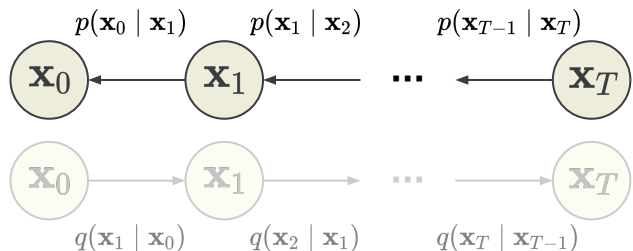
Gauss 分布の共役性より、あるステップへの遷移確率 $q(\mathbf{x}_t | \mathbf{x}_0)$ 直接計算可能である。

$$q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \bar{\beta}_t} \mathbf{x}_0, \bar{\beta}_t I) \quad \left(1 - \bar{\beta}_t \equiv \prod_{s=1}^t (1 - \beta_s) \right)$$

3.4. Denoising Diffusion Probabilistic Model (DDPM) (2/3)

DDPM の逆拡散過程においても同様に遷移確率が正規分布で近似可能である

逆拡散過程 / 生成過程



Gaussian ノイズを加えていく

拡散過程 / 推論過程

同様に、逆拡散過程は以下のように定式化できる。

$$\begin{cases} p_{\theta}(\mathbf{x}_{0:T}) &= p(\mathbf{x}_T) \prod_{t=1}^T p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_t), \\ p(\mathbf{x}_T) &= \mathcal{N}(\mathbf{x}_{t-1}; \mathbf{0}, \mathbf{I}). \end{cases}$$

ここで、 β_t が十分小さければ拡散過程と逆拡散過程が同じ関数系をもつことが証明でき、遷移確率は以下で表される。

$$p(\mathbf{x}_{t-1} | \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_{\theta}(\mathbf{x}_t, t), \Sigma_{\theta}(\mathbf{x}_t, t))$$

尤度最大化の下で平均を
求めれば良い！

パラメータに依存しない固定値
 $\sigma_t \mathbf{I}$ を用いることが多い

3.4. Denoising Diffusion Probabilistic Model (DDPM) (3/3)

DDPM の目的関数は実際のノイズと予測ノイズの二乗誤差と書き表される

DDPM では VAE と同様、対数尤度 $\log p(\mathbf{x}_0)$ を最大化、つまり ELBO を最大化するように学習する。

$$\log p(\mathbf{x}_0; \theta) \geq \underbrace{\mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \frac{p(\mathbf{x}_{0:T}; \theta)}{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} \right]}_{\text{ELBO}} \equiv L_0 + L_1 + \cdots L_T,$$

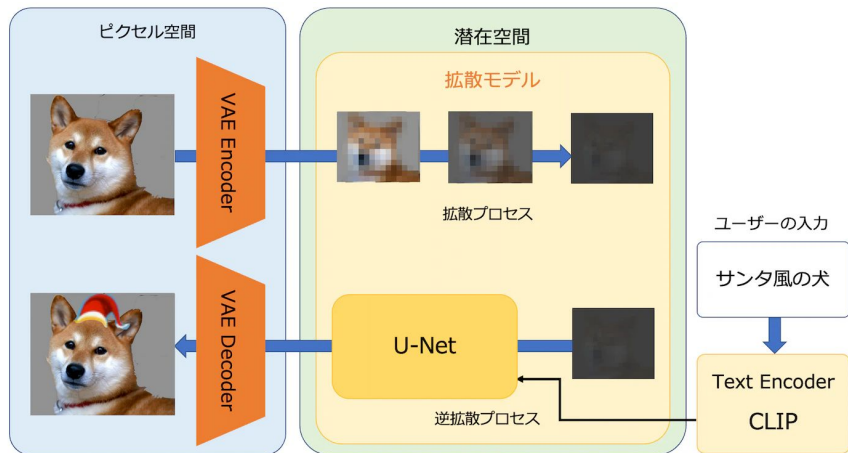
In $t = 1, 2, \dots, T-1,$

$$\begin{aligned} L_t &\equiv \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} [KL(q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) || p(\mathbf{x}_{t-1} | \mathbf{x}_t))], \\ &= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\frac{1}{2\sigma_t^2} \|\tilde{\mu}(\mathbf{x}_t, \mathbf{x}_0) - \mu_\theta(\mathbf{x}_t, t)\|^2 \right] + C, \\ &= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\frac{\beta_t^2}{2(1-\beta_t)\tilde{\beta}_t\sigma_t^2} \|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|^2 \right] + C, \\ &= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\frac{\beta_t^2}{2(1-\beta_t)\tilde{\beta}_t\sigma_t^2} \left\| \epsilon - \epsilon_\theta \left(\sqrt{1-\tilde{\beta}_t}\mathbf{x}_0 + \sqrt{\tilde{\beta}_t}\epsilon_t, t \right) \right\|^2 \right] + C. \end{aligned}$$

$q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\mu}(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t I) \quad \left(\tilde{\mu}_t = \frac{1}{\sqrt{1-\beta_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{\tilde{\beta}_t}} \epsilon \right) \right)$
 $\mu_\theta(\mathbf{x}_t, t) \equiv \frac{1}{\sqrt{1-\beta_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{\tilde{\beta}_t}} \epsilon_\theta(\mathbf{x}_t, t) \right)$
 ノイズを予測する問題に変換

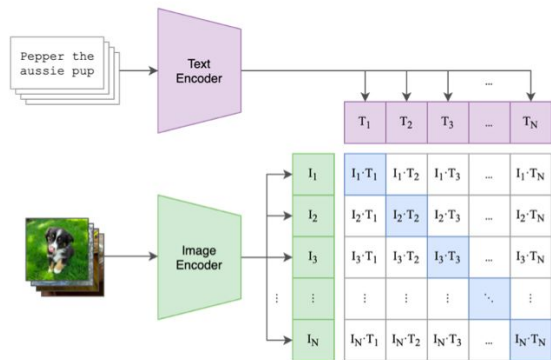
3.4. (オマケ) Stable Diffusion

画像生成で一世を風靡した (している) Stable Diffusion は拡散過程の前に VAE を導入することで学習コストを大幅に削減した

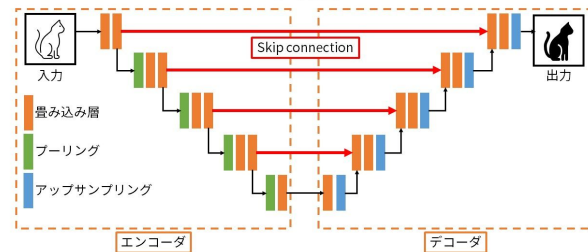


From “[図で見てわかる！画像生成AI「Stable Diffusion」の仕組み](#)”.

CLIP の概要



U-Net



エンコーダで特徴を抽出した後、デコーダで元の解像度に戻します
その際、skip connectionで元の高解像度の画像の情報も使います

Contents

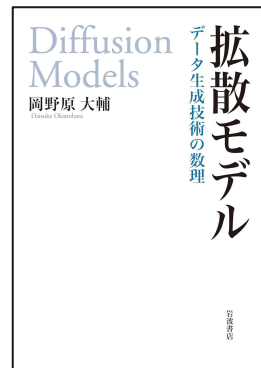
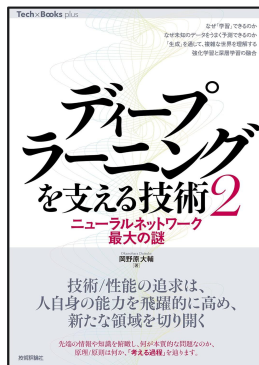
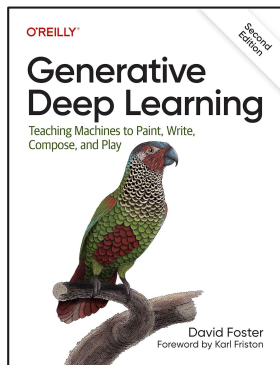
1. What is Generative Modeling?
2. History & Taxonomy of Generative Models
3. Image Generative Models
4. Summary

Summary

	Likeli- hood	Learning Stability	Fidelity	Generation Speed	Summary
3.1 Variational autoencoders	△	○	△	○	潜在空間を確率空間として学習するため、 精細な画像の生成が難しい
3.4 Diffusion models	○	○	○	×	少しずつノイズ除去をするため、生成に多 層 (1000 層程度) の処理が必要
3.3 Normalizing flow models	○	△	△	○	計算簡略化のために NN の構造が制限され ているので表現力が弱い
3.2 Generative adversarial networks	×	×	○	○	生成器と識別器のバランスの上に成り立っ ているため、学習が安定しづらい

References

- [Generative Deep Learning, 2nd Edition](#)
 - NN の基本から応用・実装まで書かれている良書。2023/05 発売なので英語版しかない。
- [ディープラーニングを支える技術 2](#)
 - 深層生成モデルや深層強化学習などニッチな話題を取り扱っている。一般書に見えて程よく数式で理論的背景を説明してくれる良書だが、これだけで全てを理解するのは難しい。
- [拡散モデル データ生成技術の数理](#)
 - 数式が多くあり読むのは大変。拡散モデルの概要を知りたいだけなら他を当たるべし。



Thank you! :)