

チュートリアル

# Generative Adversarial Networks の基礎・発展・応用

NTTコミュニケーション科学基礎研究所

金子 卓弘

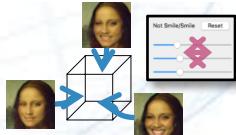
2020年6月10日@JSAI2020 KS-02 画像とAI – MIRU2020 プレビュー

# 自己紹介

- 2014. 4～
- 2017. 4～2020.3

NTTコミュニケーション科学基礎研究所  
東京大学 原田研（社会人博士）

## Image Synthesis



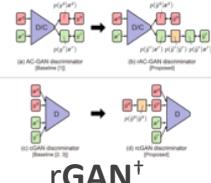
CFGAN

[CVPR 2017]



DTLC-GAN  
[CVPR 2018]

### 制御可能な画像生成



rGAN<sup>+</sup>  
[CVPR 2019]

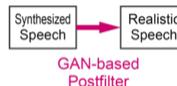
CP-GAN<sup>+</sup>  
[BMVC 2019]



NR-GAN<sup>+</sup>  
[CVPR 2020]

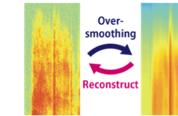
### 不確実データからの画像生成

## Speech Synthesis



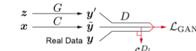
GAN-PF

[ICASSP 2017]



GAN-PF for STFT

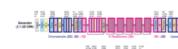
[Interspeech 2017]



GAN-VC

### 高品質な音声合成・音声変換

CycleGAN-VC



CycleGAN-VC  
[EUSIPCO 2018]



CycleGAN-VC2  
[ICASSP 2019]



StarGAN-VC2  
[Interspeech 2019]

### ノンパラレル・マルチドメイン音声変換

<sup>†</sup>社会人博士の研究成果

# 自己紹介

- 2014. 4～
- 2017. 4～2020.3

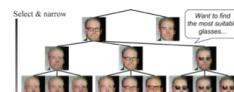
NTTコミュニケーション科学基礎研究所  
東京大学 原田研（社会人博士）



## Image Synthesis

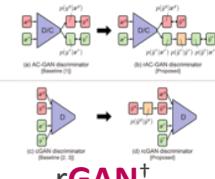


[CVPR 2017]



DTLC-GAN

制御可能な画像生成



[CVPR 2019]

[CVPR 2018]

CP-GAN<sup>†</sup>

[BMVC 2019]

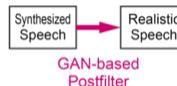


NR-GAN<sup>†</sup>

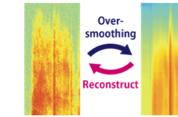
[CVPR 2020]

不確実データからの画像生成

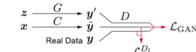
## Speech Synthesis



GAN-PF



GAN-PF for STFT



GAN-VC

[ICASSP 2017]

[Interspeech 2017] [Interspeech 2017]

高音質音声合成・音声変換

CycleGAN-VC

CycleGAN-VC

[EUSIPCO 2018]

CycleGAN-VC2

CycleGAN-VC2

[ICASSP 2019]



StarGAN-VC2

[Interspeech 2019]

ノンパラレル・マルチドメイン音声変換

<sup>†</sup>社会人博士の研究成果

# GANによる画像生成とは？①

- Q1. 計算機が生成したのはどれ?
  - 生成 = 計算機が適当に与えた「乱数」から画像を創出



# GANによる画像生成とは？①

- Q1. 計算機が生成したの？  
– 生成 = 計算機が適当にうそをつくことを創出

正解は全部

StyleGAN2



乱数

→

画像



Generator



# GANによる画像生成とは？②

- Q2. 正解ペアデータが存在しない条件下で画像変換するためには？



絵画 by Monet

写真



?

写真



ウマ

シマウマ



?

シマウマ

# GANによる画像生成とは？②

- Q2. 正解ペアデータが存在しない条件下で画像変換するためには？

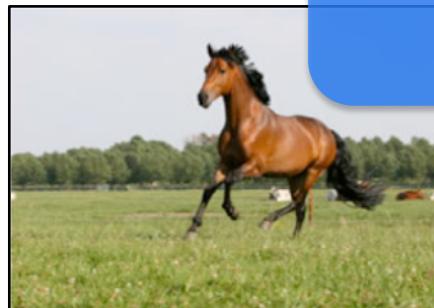


絵画 by Monet

写真

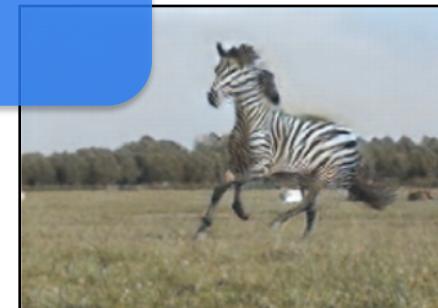


写真



ウマ

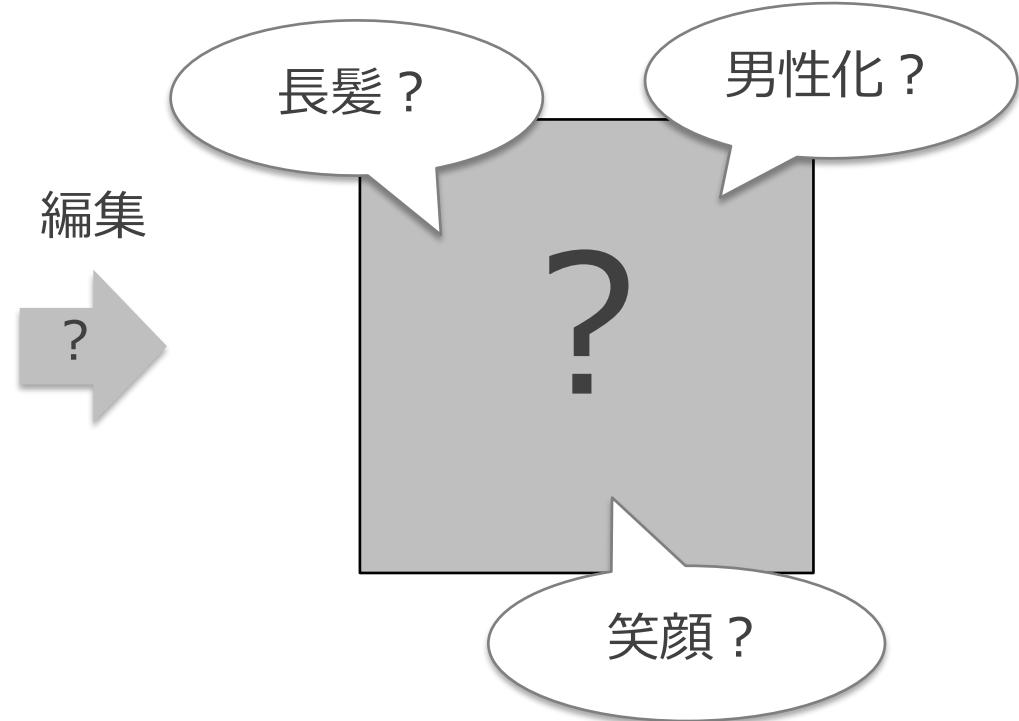
CycleGAN



シマウマ

# GANによる画像生成とは？③

- Q3. 好みの写真に編集するためには？



# GANによる画像生成とは？③

- Q3. 好みの写真に編集する方法は？



デモ動画：<http://www.kecl.ntt.co.jp/people/kaneko.takuhiro/projects/gac/index.html>

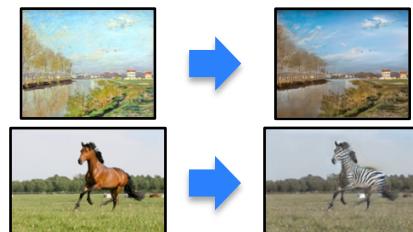
# GANによる画像生成とは？(まとめ)

## 画像生成



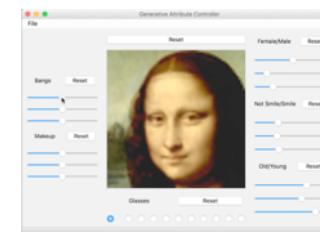
StyleGAN2 [Karras+arXiv2019]

## 画像変換



CycleGAN [Zhu+ICCV2017]

## 画像編集



CFGAN [Kaneko+CVPR2017]

# GANによる画像生成とは？(まとめ)

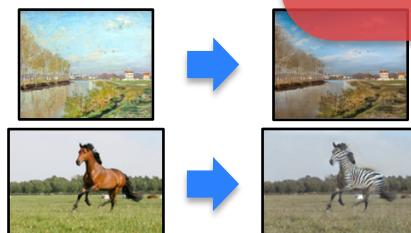
## 画像生成



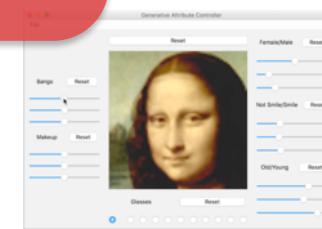
様々な応用が  
可能な技術！

画像変換

画像編集



CycleGAN [Zhu+ICCV2017]



CFGAN [Kaneko+CVPR2017]

GAN  
の基礎

GANの  
発展

GANの  
応用

時間の都合で割愛  
(MIRUのスライド資料<sup>†</sup>ご参照)

<sup>†</sup> <http://www.kecl.ntt.co.jp/people/kaneko.takuhiro/#talks>

# 目次



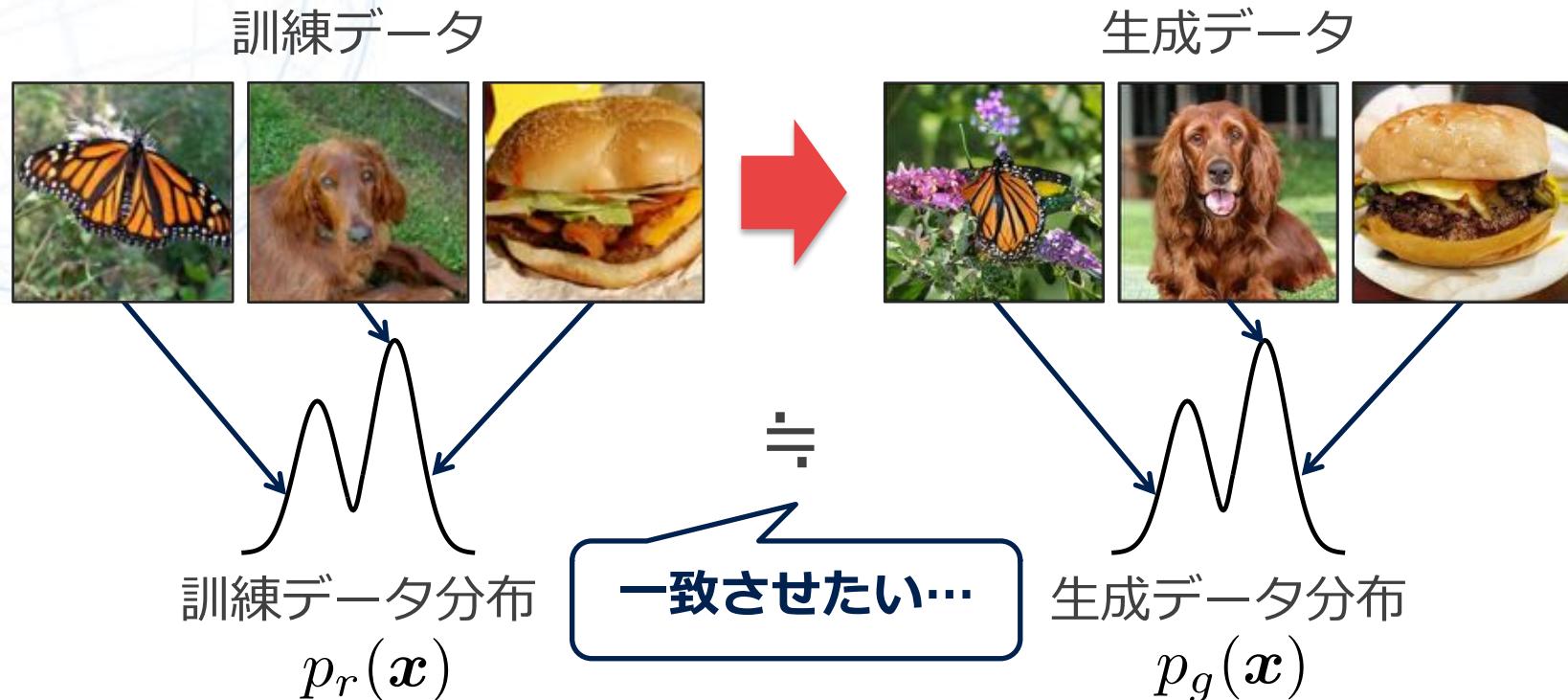
GAN  
の基礎

GANの  
発展

GANの  
応用

# 生成モデルとは？

- 目的：訓練データを再現できる**画像生成器 (Generator)**を学習



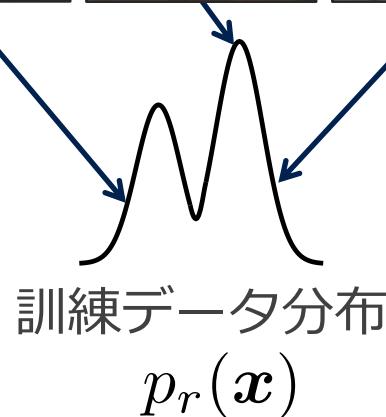
# 生成モデルの学習の難しさ

- 複雑な訓練データをどうやって再現するか

訓練データ



高次元データ (画像・動画など)  
では複雑な形状  
→人手で設計するのは困難

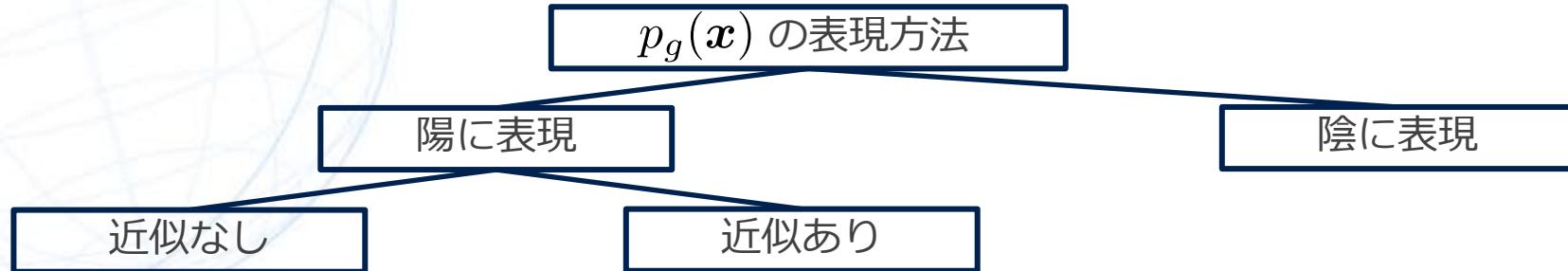


↓  
深層学習の表現力の高さを活用できないか?  
→深層生成モデル



# 深層生成モデル

- 代表的な深層生成モデル : AR, Flow, VAE, GAN



**AR** Autoregressive Model

連鎖律 (Chain Rule)

$$p_g(\mathbf{x}) = \prod_{i=1}^n p_g(x_i | x_1, \dots, x_{i-1})$$

**Flow** Flow-based Model

可逆生成

$$z = f(\mathbf{x}) \quad \mathbf{x} = f^{-1}(z)$$

**VAE**

Variational Autoencoder

変分近似

$$\begin{aligned} \log p_g(\mathbf{x}) &\geq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}) \\ &= -KL(q_{\boldsymbol{\phi}}(z|\mathbf{x}) \| p(z)) \\ &\quad + \mathbb{E}_{q_{\boldsymbol{\phi}}(z|\mathbf{x})} [\log p_{\boldsymbol{\theta}}(\mathbf{x}|z)] \end{aligned}$$

**GAN**

Generative Adversarial Networks

敵対的学習

$$\begin{aligned} \min_G \max_D \mathbb{E}_{\mathbf{x} \sim p_r(\mathbf{x})} [\log D(\mathbf{x})] \\ + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \end{aligned}$$

# 深層生成モデル

- 代表的な深層生成モデル : AR, Flow, VAE, GAN

$p_g(\mathbf{x})$  の表現方法

対象となるデータの分布  
を知らなくても良い

陽に表現

陰に表現

近似なし

近似あり

**AR** Autoregressive Model

連鎖律 (Chain Rule)

$$p_g(\mathbf{x}) = \prod_{i=1}^n p_g(x_i | x_1, \dots, x_{i-1})$$

**Flow** Flow-based Model

可逆生成

$$z = f(\mathbf{x}) \quad \mathbf{x} = f^{-1}(z)$$

**VAE**

Variational Autoencoder

変分近似

$$\begin{aligned} \log p_g(\mathbf{x}) &\geq \mathcal{L}(\theta, \phi; \mathbf{x}) \\ &= -KL(q_\phi(z|\mathbf{x}) \| p(z)) \\ &\quad + \mathbb{E}_{q_\phi(z|\mathbf{x})} [\log p_\theta(\mathbf{x}|z)] \end{aligned}$$

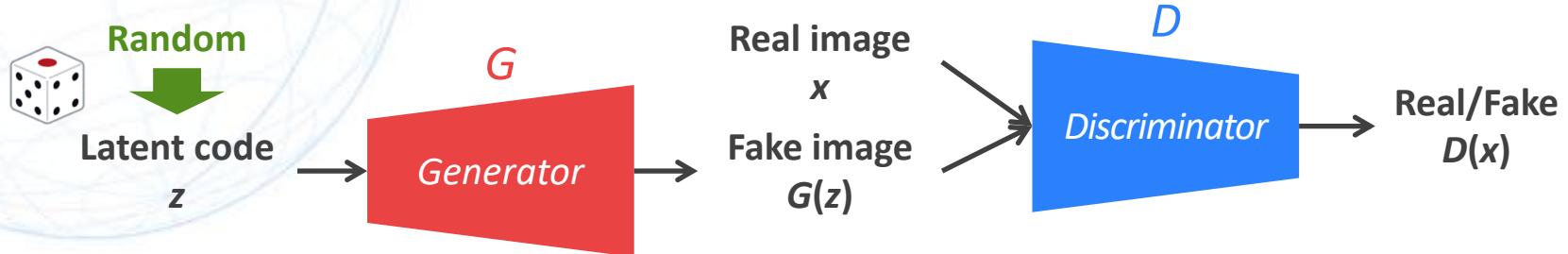
**GAN**

Generative Adversarial Networks

敵対的学習

$$\begin{aligned} \min_G \max_D \mathbb{E}_{\mathbf{x} \sim p_r(\mathbf{x})} [\log D(\mathbf{x})] \\ + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))] \end{aligned}$$

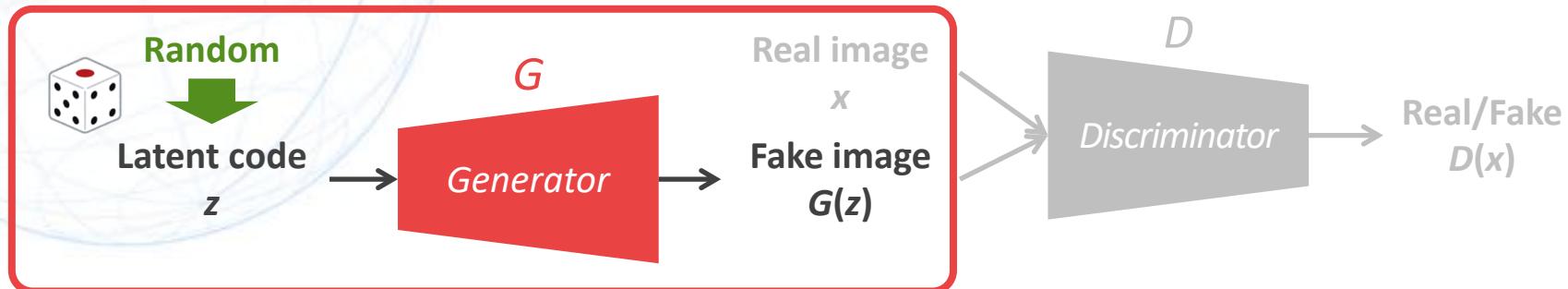
- Generative Adversarial Networks [Goodfellow+2014]
  - 生成器 (*Generator*) と識別器 (*Discriminator*) を敵対的に学習



- Min-Max最適化 (*Generator* vs. *Discriminator*)

$$\min_G \max_D \mathbb{E}_{x \sim p_r(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

- Generative Adversarial Networks [Goodfellow+2014]
  - 生成器 (*Generator*) と識別器 (*Discriminator*) を敵対的に学習



乱数から画像生成

- Min-Max最適化 (*Generator* vs. *Discriminator*)

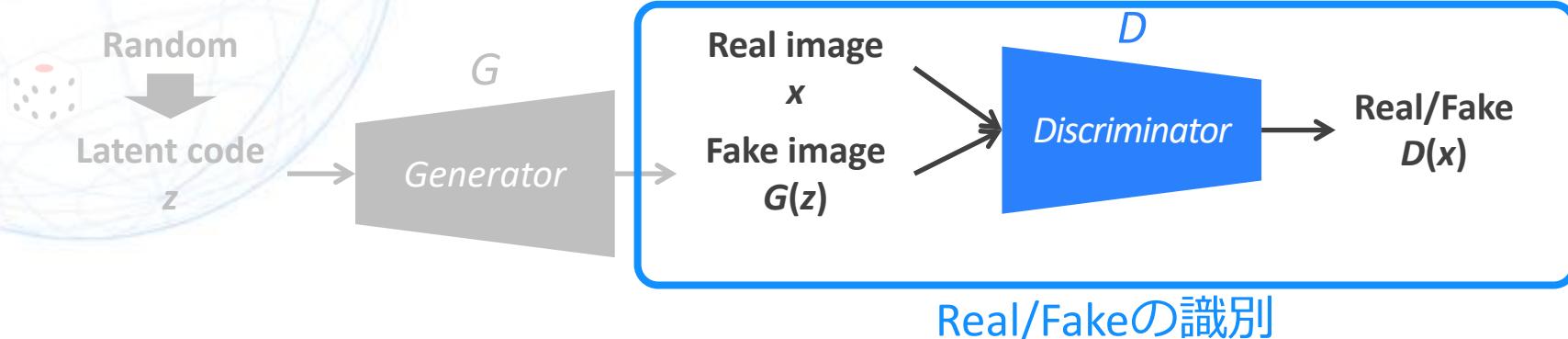
Fakeでなくする

$$\min_G \max_D \mathbb{E}_{x \sim p_r(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

*Discriminator*を騙せるように最適化

# GAN 3/4

- Generative Adversarial Networks [Goodfellow+2014]
  - 生成器 (Generator) と識別器 (Discriminator) を敵対的に学習

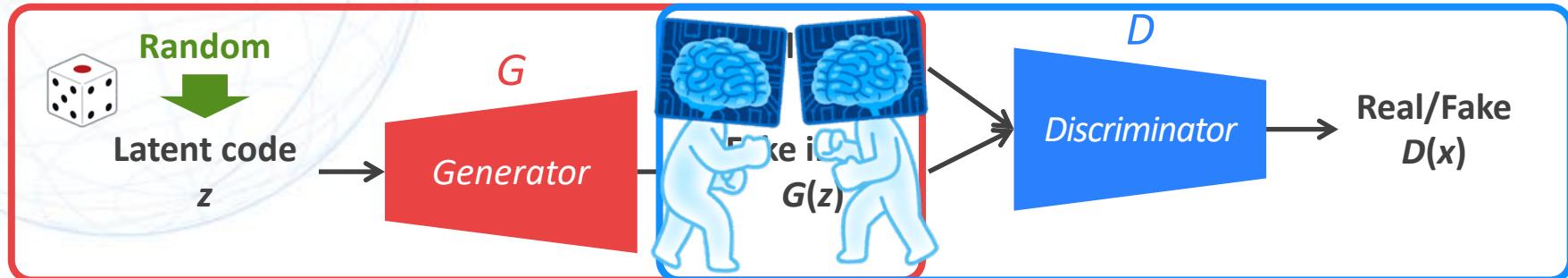


- Min-Max最適化 (Generator vs. Discriminator)  
Real/Fakeの二値識別

$$\min_G \max_D \mathbb{E}_{x \sim p_r(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

Generatorに騙されないように最適化

- Generative Adversarial Networks [Goodfellow+2014]
  - 生成器 (Generator) と識別器 (Discriminator) を敵対的に学習



乱数から画像生成

競争

Real/Fakeの識別

- Min-Max最適化 (Generator vs. Discriminator)

GeneratorとDiscriminatorをMin-Maxで競争

$$\min_G \max_D \mathbb{E}_{x \sim p_r(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

最終的にGeneratorは本物と見分けのつかない画像を生成

# 目次



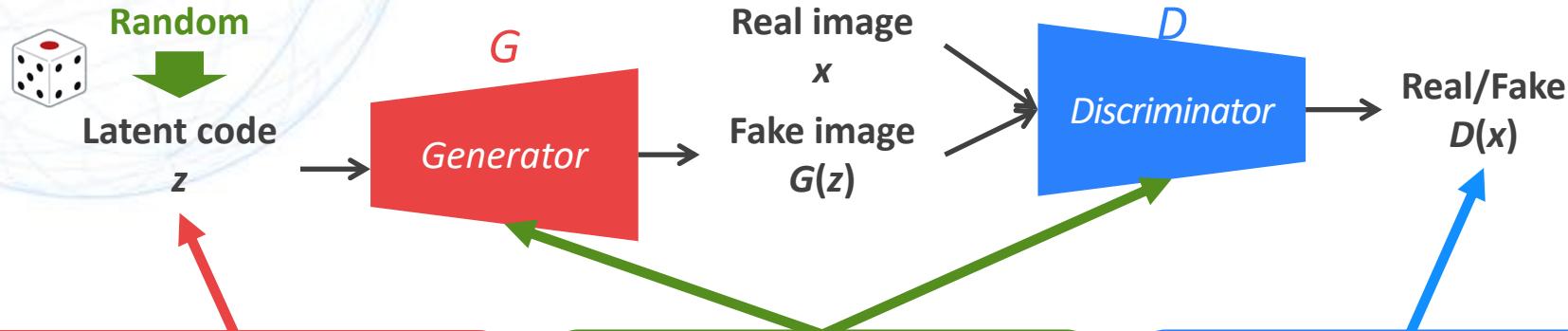
GAN  
の基礎

GANの  
発展

GANの  
応用

# GANの発展

- 4つの発展



**潜在変数構造の工夫**  
操作性向上  
cGAN, AC-GAN, InfoGAN,  
CFGAN, DTLC-GAN

**データ設定の工夫**  
不完全データに対応  
rGAN, CP-GAN, S<sup>3</sup>GAN,  
AmbientGAN, NR-GAN

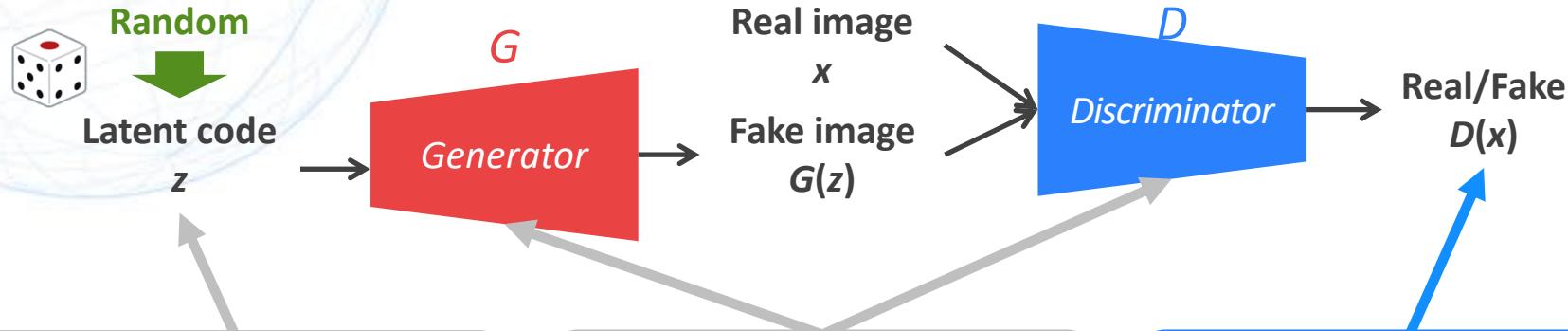
**ネットワークの工夫**  
高解像度化  
DCGAN, LAPGAN, StackGAN,  
SAGAN, ProGAN, StyleGAN,  
MSG-GAN, StyleGAN2

**目的関数の工夫**  
学習の安定化  
LSGAN, WGAN, WGAN-GP,  
SN-GAN, R<sub>1</sub> GP, Hinge GAN,  
Relativistic GAN, (BigGAN)

※GANの研究は近年非常に盛んであり他にも様々な面白い試みあり

# GANの発展

- 4つの発展



潜在変数構造の工夫  
操作性向上  
cGAN, AC-GAN, InfoGAN,  
CFGAN, DTLC-GAN

データ設定の工夫  
不完全データに対応  
rGAN, CP-GAN, S<sup>3</sup>GAN,  
AmbientGAN, NR-GAN

ネットワークの工夫  
高解像度化  
DCGAN, LAPGAN, StackGAN,  
SAGAN, ProGAN, StyleGAN,  
MSG-GAN, StyleGAN2

目的関数の工夫  
学習の安定化  
LSGAN, WGAN, WGAN-GP,  
SN-GAN, R<sub>1</sub> GP, Hinge GAN,  
Relativistic GAN, (BigGAN)

※GANの研究は近年非常に盛んであり他にも様々な面白い試みあり

- Min-Max最適化のため学習が不安定
  - 理想的な条件下ではJensen-Shannon Divergence (JSD) 基準で収束
    - 十分な表現力のあるGeneratorとDiscriminator
    - 十分なサイズの訓練データ

$$\begin{aligned}
 & \mathbb{E}_{\mathbf{x} \sim p_r(\mathbf{x})} [\log D_G^*(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} [\log(1 - D_G^*(G(\mathbf{z})))] \\
 &= -\log(4) + KL \left( p_r(\mathbf{x}) \middle\| \frac{p_r(\mathbf{x}) + p_g(\mathbf{x})}{2} \right) + KL \left( p_g(\mathbf{x}) \middle\| \frac{p_r(\mathbf{x}) + p_g(\mathbf{x})}{2} \right) \\
 &= -\log(4) + 2 \cdot JSD(p_r(\mathbf{x}) \| p_g(\mathbf{x}))
 \end{aligned}$$

↑                   ↑  
訓練データ分布 生成データ分布

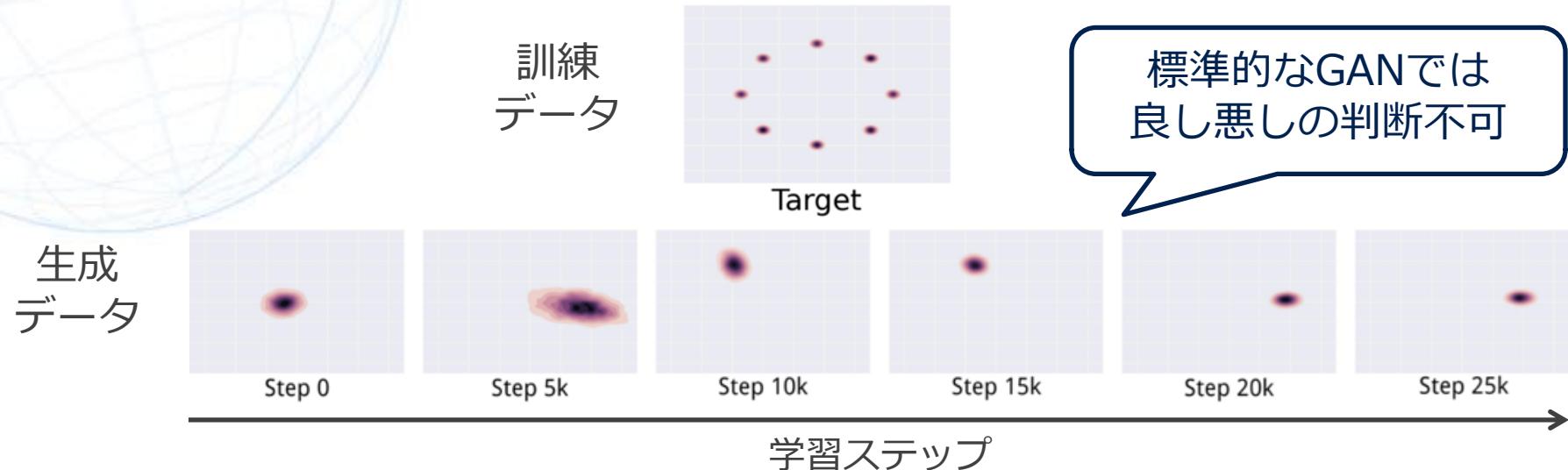
- しかし、実際には以下の条件
  - 有限な表現力のGeneratorとDiscriminator
  - 有限サイズの訓練データ



学習は不安定化  
 • Mode Collapseなど  
 (次ページ)

# Mode Collapse

- Mode Collapse [Metz+2017]
  - サンプルが特定箇所に集中



RealとFakeの分布間距離がうまく測れていない？

- Least Squares GAN [Mao+2017]
  - Sigmoid Cross Entropyの代わりにLeast Squares Lossを利用

### Discriminator

$$\text{GAN} \quad \min_D \mathbb{E}_{\mathbf{x} \sim p_r(\mathbf{x})}[-\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})}[-\log(1 - D(G(\mathbf{z})))]$$



$$\text{LSGAN} \quad \min_D \mathbb{E}_{\mathbf{x} \sim p_r(\mathbf{x})}[(D(\mathbf{x}) - 1)^2] + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})}[D(G(\mathbf{z}))^2]$$

Realは1に近づける              Fakeは0に近づける

### Generator

$$\text{GAN} \quad \min_G \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})}[-\log D(G(\mathbf{z}))]$$



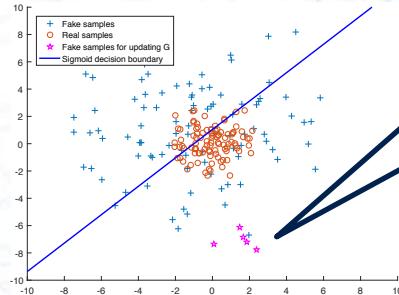
$$\text{LSGAN} \quad \min_G \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})}[(D(G(\mathbf{z})) - 1)^2]$$

Fakeを1 (Real) に近づける

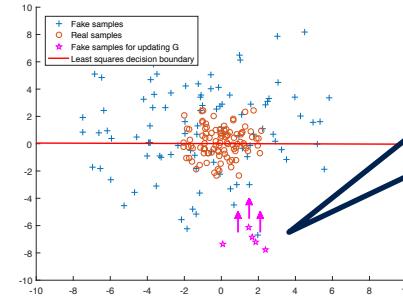
※ LSGANは  $p_r(\mathbf{x}) + p_g(\mathbf{x})$  と  $2p_g(\mathbf{x})$  のChi-square Distanceを最小化

# LSGAN (続き)

- 直感的解釈



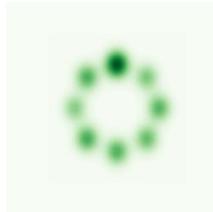
GAN: Sigmoid Cross Entropy



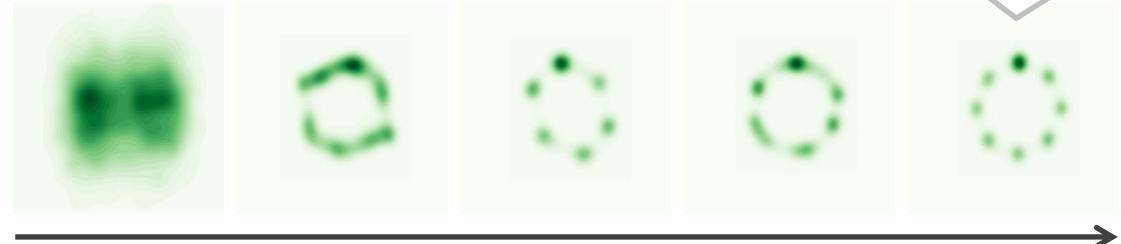
LSGAN: Least Squares Loss

- 実験結果

訓練  
データ



生成  
データ



Mode Collapseを回避

- Wasserstein GAN [Arjovsky+2017]
  - Classifier関数の代わりにCritic関数を利用

### Discriminator

$$\text{GAN} \quad \min_D \mathbb{E}_{\mathbf{x} \sim p_r(\mathbf{x})}[-\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})}[-\log(1 - D(G(\mathbf{z})))]$$



$$\text{WGAN} \quad \max_D \mathbb{E}_{\mathbf{x} \sim p_r(\mathbf{x})}[D(\mathbf{x})] - \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})}[D(G(\mathbf{z}))]$$

RealとFakeの距離をなるべく正確に測る

$D$ はLipschitz制約を満たすことが必要  
→ Weight Clipping

### Generator

$$\text{GAN} \quad \min_G \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})}[-\log D(G(\mathbf{z}))]$$



$$\text{WGAN} \quad \max_G \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})}[D(G(\mathbf{z}))]$$

距離を最小化

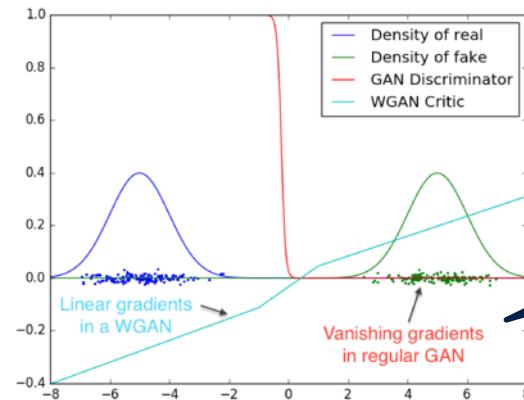
※ WGANは  $p_g(\mathbf{x})$  と  $p_r(\mathbf{x})$  のEarth Mover Distanceを最小化

# WGAN (続き)

- 直感的解釈

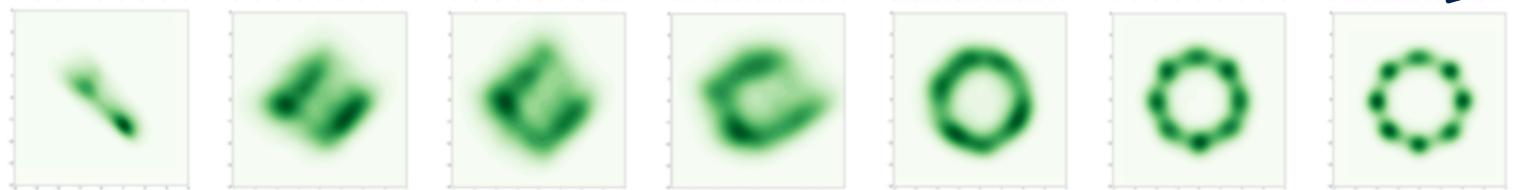
WGAN (水色線): 線形の勾配

Wasserstein距離は緩い制約  
のもと連続・微分可能



GAN (赤線): 勾配消失

- 実験結果



Mode Collapseを回避

- WGANの課題
  - $D$ にLipschitz制約を課すためWeight Clipping (きつい制約)  
1-Lipschitz       $\|f(\mathbf{x}) - f(\mathbf{y})\| \leq \|\mathbf{x} - \mathbf{y}\|$  for all  $\mathbf{x}, \mathbf{y}$

→ モデルが過度に簡易化

Weight Clippingにより  $f$  の範囲を制約

- WGAN Gradient Penalty [Gulrajani+2017]
  - 背景 :  $f$  が連続で微分可能  $\rightarrow f$  が 1-Lipschitz になる必要十分条件は

$$\|\nabla f\| \leq 1 \text{ everywhere}$$

- 提案 : Gradient Penalty (GP) (緩い制約)  
$$\lambda \mathbb{E}_{\hat{\mathbf{x}} \sim p_{\hat{\mathbf{x}}}(\hat{\mathbf{x}})}[(\|\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})\|_2 - 1)^2]$$
  - 财献 : 様々なモデル (FC, Conv, ResNet) でも学習可能に

- Spectral Normalization GAN [Miyato+2018]
  - 背景：ニューラルネットワークは重み $W$ と活性化関数 $a$ の組合せ各層がLipschitz  $\rightarrow f$ もLipschitz

$$f(\mathbf{x}) = W^{L+1}(a_L(W^L(a_{L-1}(W^{L-1}(\dots a_1(W^1\mathbf{x})\dots))))$$

活性化関数は元からLipschitz

$W$ をLipschitzにするには？

- 提案：Spectral Normalization (SN)

$$W_{\text{SN}} = W / \sigma(W)$$

$\sigma(W)$ ：最大特異値

- 貢献：複雑なデータ (ImageNet) でも学習可能に



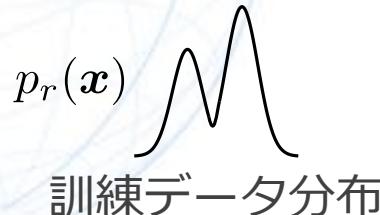
- Large Scale GAN [Brock+2019]
  - 計算機パワーを活かして
    - バッチサイズ（一度の学習で見るデータ量）
    - チャンネルサイズ（モデルの表現能力）を大きくして学習  
→ 学習が安定化し、より高精細な画像の生成が可能に



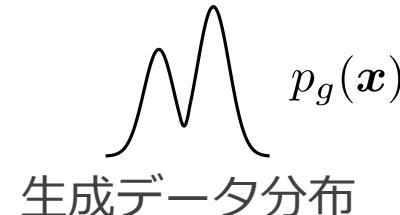
A. Brock et al., "Large Scale GAN Training for High Fidelity Natural Image Synthesis," ICLR 2019.

# 目的関数のまとめ

- GANの課題①
  - Min-Max最適化のため学習が不安定



÷



どうやって一致させるか  
(距離基準) が鍵

## 目的関数の発展

**GAN**  
Jensen-Shannon  
Divergence

距離基準  
の改善

**LSGAN**  
Chi-Square  
Divergence

Lipschitz  
制約の改善

**WGAN-GP**  
Gradient Penalty

**WGAN**  
Earth Mover Distance  
+ Weight Clipping

**SN-GAN**  
Spectral Normalization

学習の安定化

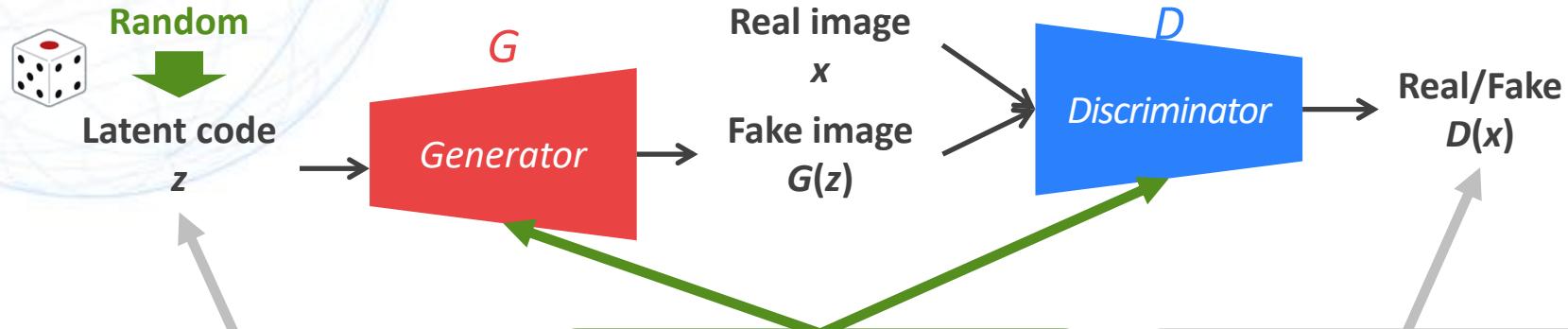
計算機パワー

**BigGAN**  
Scale Up

※他 : R<sub>1</sub> GP, Hinge GAN, Relativistic GANなど

# GANの発展

- 4つの発展



潜在変数構造の工夫  
操作性向上  
cGAN, AC-GAN, InfoGAN,  
CFGAN, DTLC-GAN

データ設定の工夫  
不完全データに対応  
rGAN, CP-GAN, S<sup>3</sup>GAN,  
AmbientGAN, NR-GAN

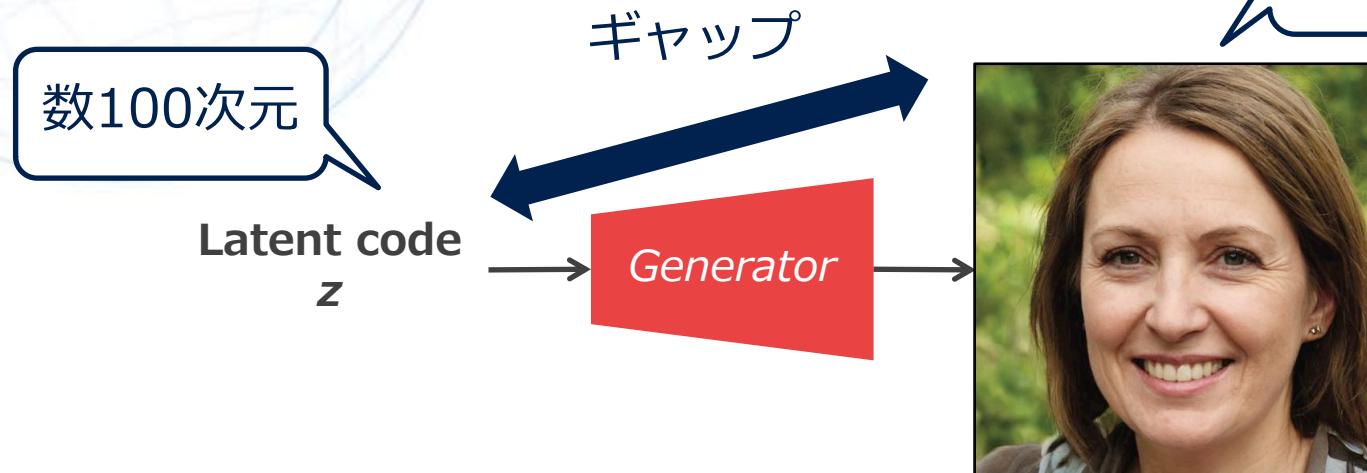
ネットワークの工夫  
高解像度化  
DCGAN, LAPGAN, StackGAN,  
SAGAN, ProGAN, StyleGAN,  
MSG-GAN, StyleGAN2

目的関数の工夫  
学習の安定化  
LSGAN, WGAN, WGAN-GP,  
SN-GAN, R<sub>1</sub> GP, Hinge GAN,  
Relativistic GAN, (BigGAN)

※GANの研究は近年非常に盛んであり他にも様々な面白い試みあり

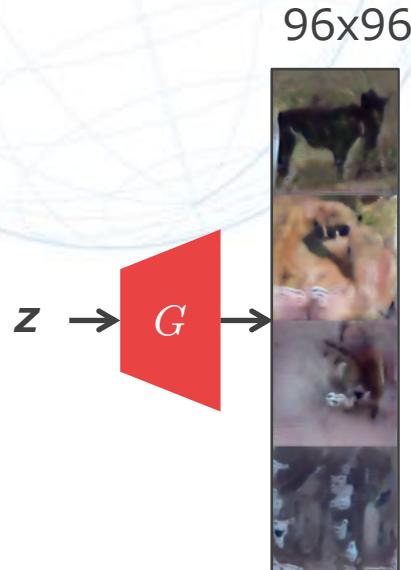
# GANの課題②

- 入力と出力のギャップ
  - 非常に複雑な関数を学習することが必要
    - 特に、解像度が高くなればなるほどギャップ大

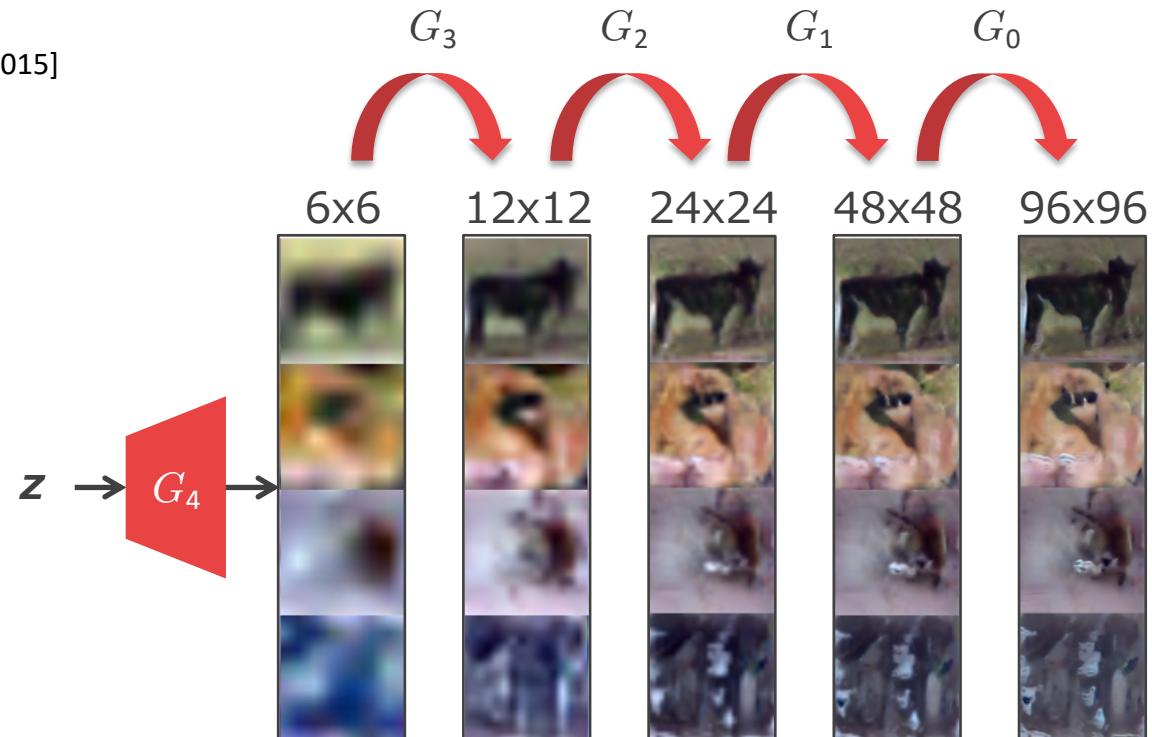


生成しやすいネットワーク構造にすることが重要

- Laplacian GAN [Denton+2015]
  - 段階的に画像生成



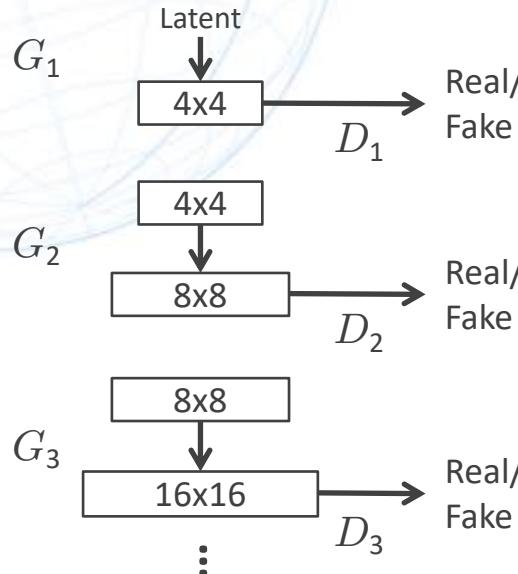
$G$ : 複雑な関数が必要



$G_0, \dots, G_4$ : 比較的簡易な関数でよい

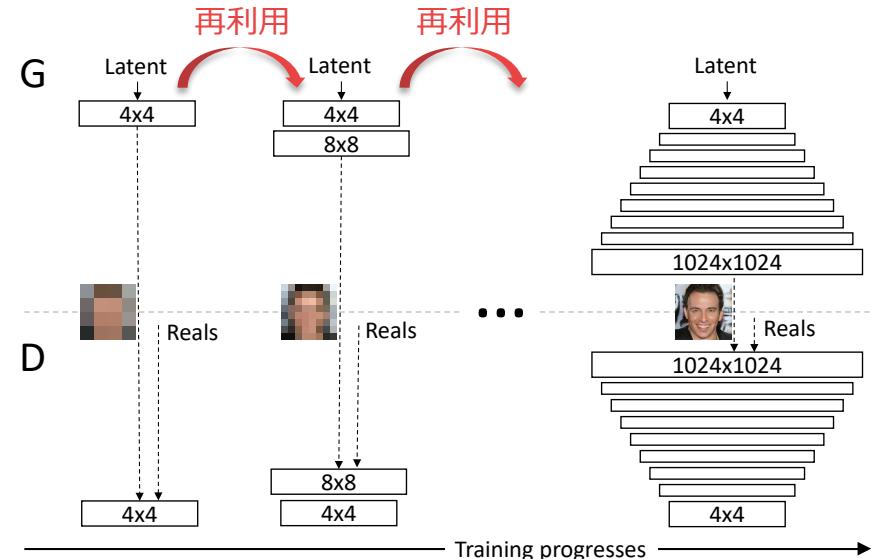
# ProGAN

- Progressive Growing of GANs [Karras+2018]
  - 単一のネットワーク内で、層を追加しながら高解像度化



LAPGAN

解像度ごとに  $G$  と  $D$  → パラメータ/学習時間大



ProGAN

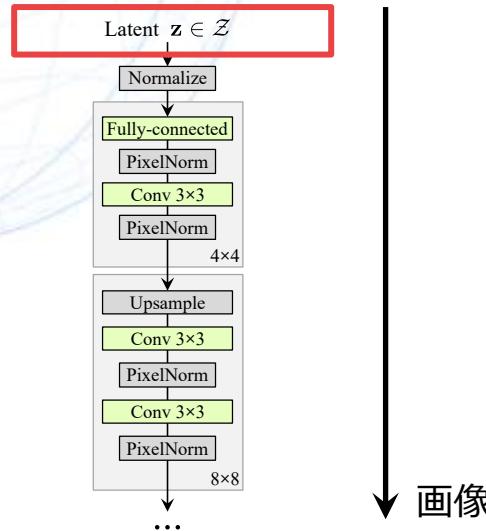
単一の  $G$  と  $D$  → パラメータ/学習時間小

# ProGAN (続き)

- 画像生成例 1024x1024の高解像度画像の生成が可能



- Style-Based Generator for GANs [Karras+2019]
  - Generatorの入力を多段階化



# StyleGAN (続き)

- 画像生成例 さらに高精細な画像の生成が可能



# ネットワーク構造のまとめ

- GANの課題②
  - 入力と出力のギャップ（特に、高解像度化したいときに課題）



- ネットワーク構造の発展

LAPGAN → ProGAN → StyleGAN → StyleGAN2  
段階的に生成 → 層を追加しながら生成 → 入力を多段階化 → 改良

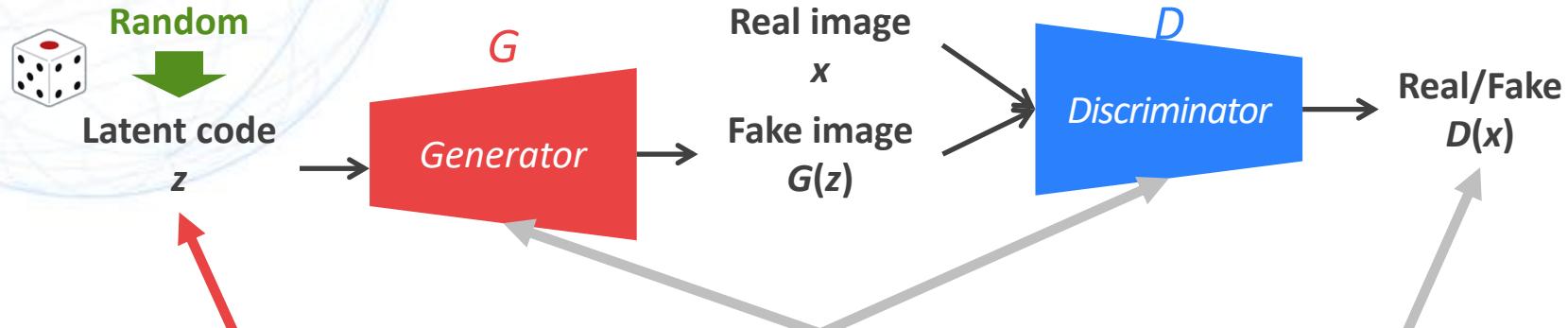


高解像度化



# GANの発展

- 4つの発展



**潜在変数構造の工夫**  
操作性向上  
cGAN, AC-GAN, InfoGAN,  
CFGAN, DTLC-GAN

**データ設定の工夫**  
不完全データに対応  
rGAN, CP-GAN, S<sup>3</sup>GAN,  
AmbientGAN, NR-GAN

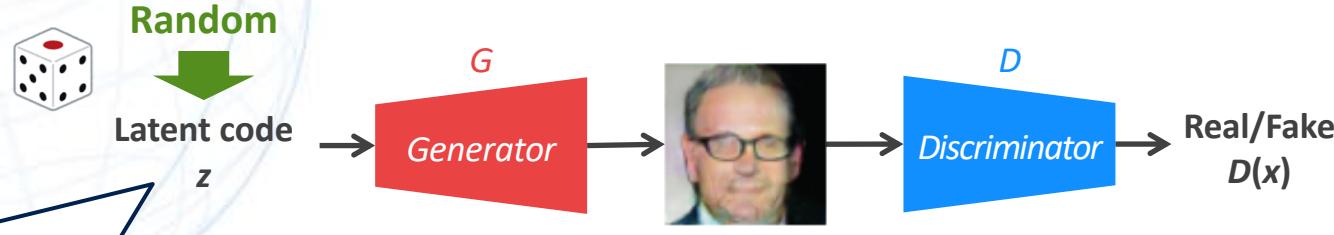
**ネットワークの工夫**  
高解像度化  
DCGAN, LAPGAN, StackGAN,  
SAGAN, ProGAN, StyleGAN,  
MSG-GAN, StyleGAN2

**目的関数の工夫**  
学習の安定化  
LSGAN, WGAN, WGAN-GP,  
SN-GAN, R<sub>1</sub> GP, Hinge GAN,  
Relativistic GAN, (BigGAN)

※GANの研究は近年非常に盛んであり他にも様々な面白い試みあり

# GANの課題③

- 画像生成の制御が困難



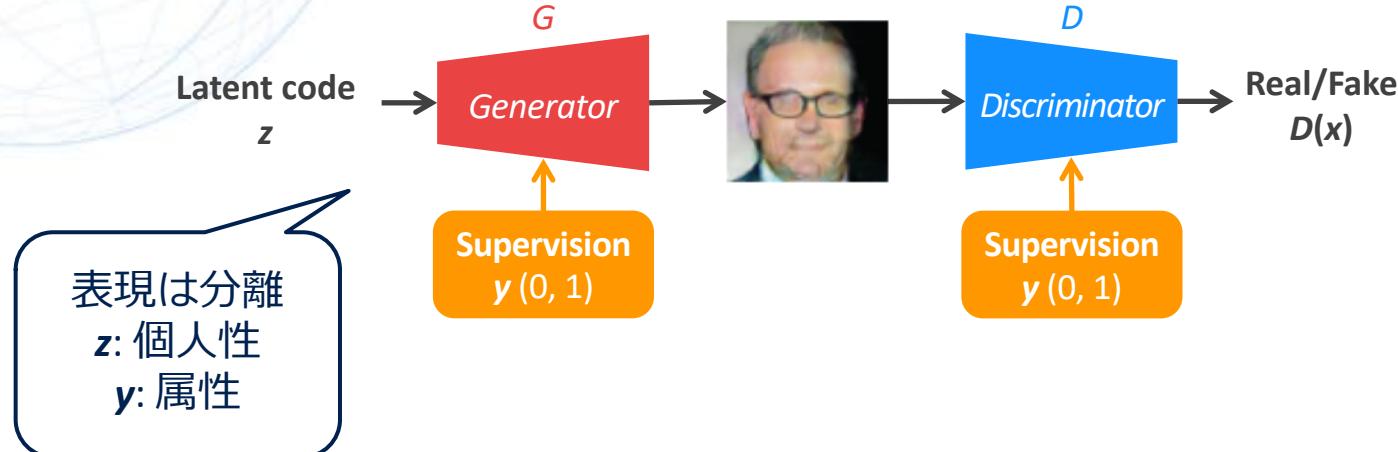
入力は乱数  
→ 表現（個人性、属性など）が混在

?



どんな画像が生成されるかはランダム

- Conditional GAN [Mirza+2014]
  - 属性の有無(例: 笑顔かどうか)を表す**教師情報  $y$** を導入  
→ ON/OFFの制御



# cGAN (続き)

- 画像生成例



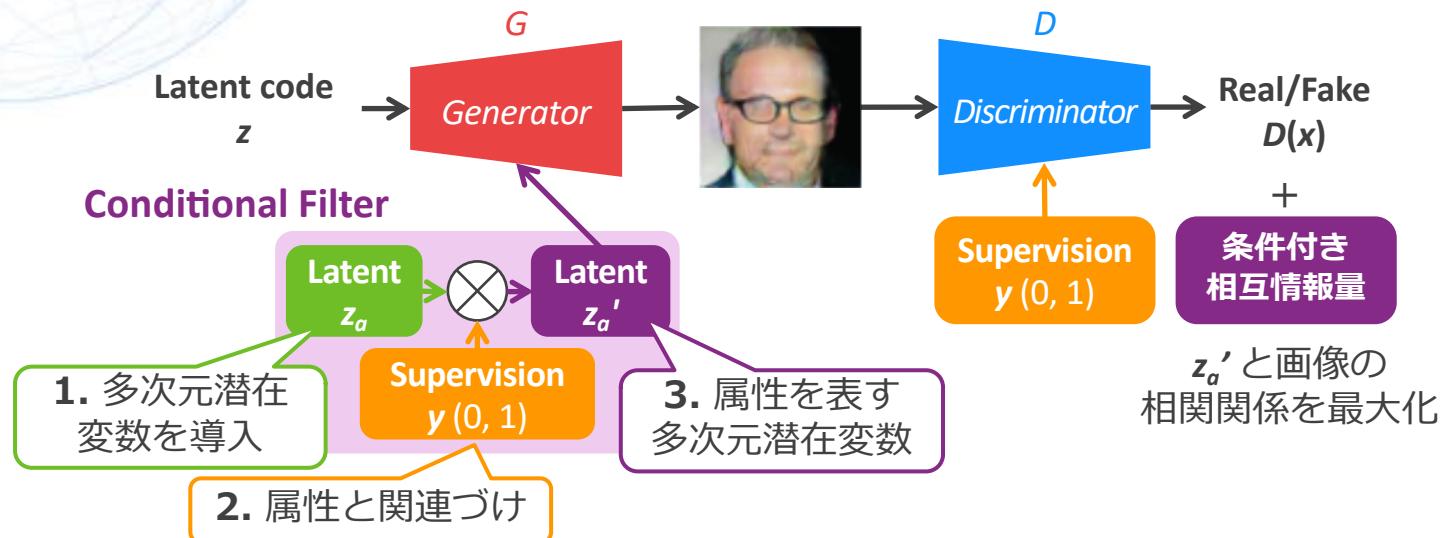
教師情報に基づづく制御（属性のON/OFF）に限定

M. Mirza & S. Osindero, "Conditional Generative Adversarial Nets," arXiv 2014.

Pictures: T. Kaneko et al., "Generative Attribute Controller with Conditional Filtered Generative Adversarial Networks," CVPR 2017.

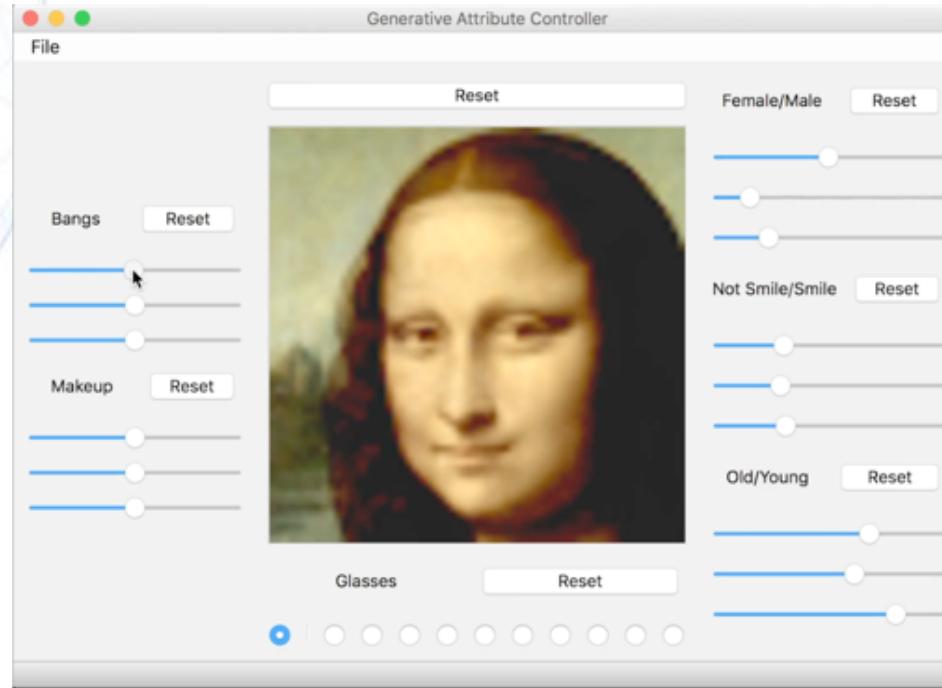
- Conditional Filtered GAN [Kaneko+2017]

- 新たに多次元潜在変数  $z_a$  を導入
- 教師情報  $y$  でフィルタリングすることで属性と関連づけ
- 属性を表す多次元潜在変数  $z_a'$  を構成 → 多次元的に制御



# CFGAN (続き)

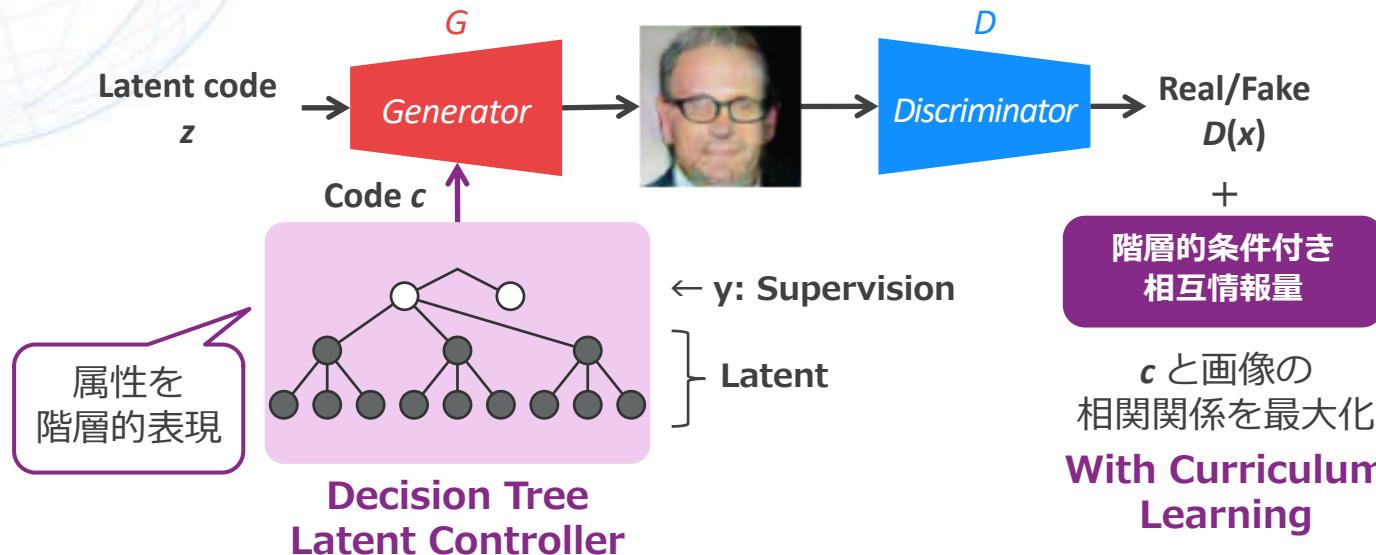
- 画像生成例



デモ動画: <http://www.kecl.ntt.co.jp/people/kaneko.takuhiro/projects/gac/index.html>

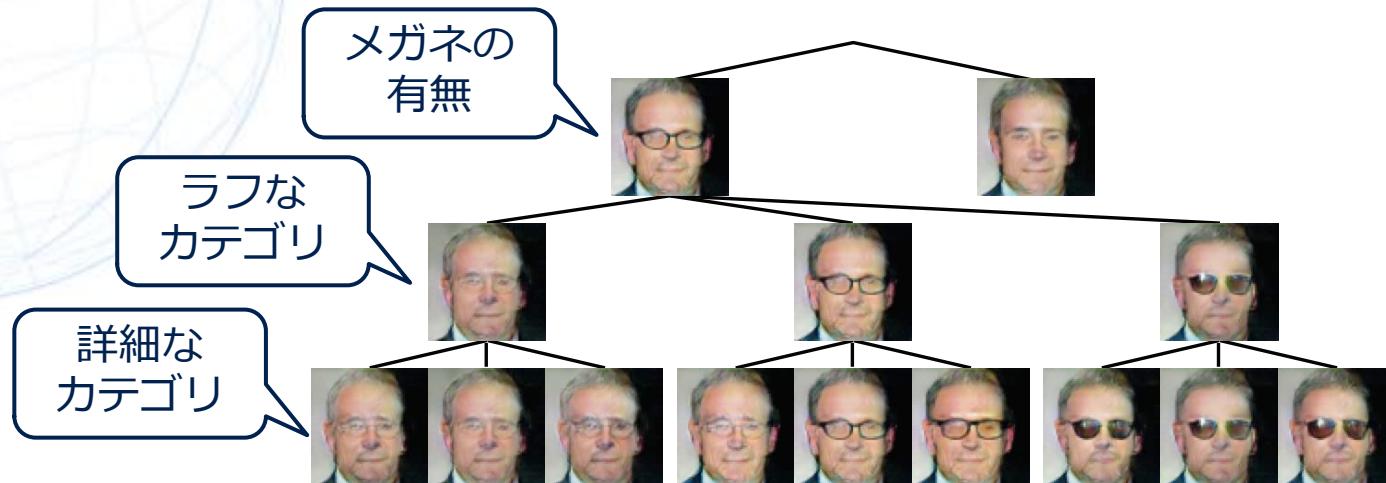
多次元的に制御可能

- Decision Tree Latent Controller GAN [Kaneko+2018]
  - 階層的なサンプリング構造を導入 → 階層的に制御
  - 学習のカリキュラムを導入 → ラフな表現～詳細な表現を順に学習



# DTLC-GAN (続き)

- 画像生成例

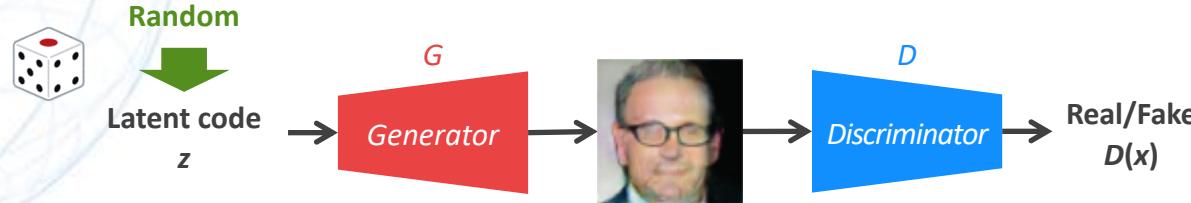


教師情報は最上位 (メガネの有無) のみ  
詳細なカテゴリはデータから発見

階層的に制御可能

# 潜在変数構造のまとめ

- GANの課題③
  - 画像生成のコントロールが困難



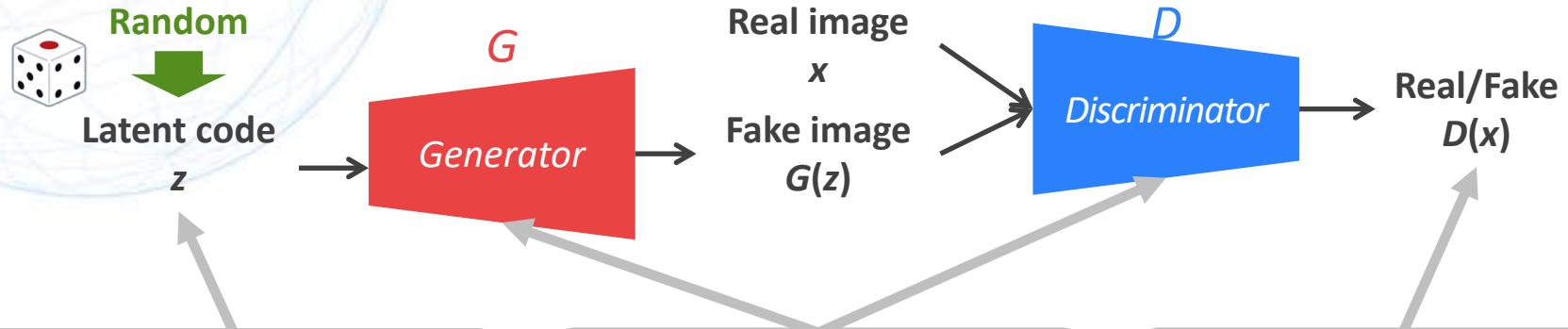
- 潜在変数構造の発展

操作性向上

層数	0	1	2, 3, ...
教師なし	<b>GAN</b> [Goodfellow+2014] $z \rightarrow G \rightarrow \text{Image}$ 分離なし	<b>InfoGAN</b> [Chen+2016] $z, C \rightarrow G \rightarrow \text{Image}$ 一層の潜在変数 <b>CFGAN</b> [Kaneko+2017] $z, C \rightarrow \otimes \rightarrow G \rightarrow \text{Image}$	<b>DTLC-GAN</b> [Kaneko+2018] $z \rightarrow \text{Tree} \rightarrow G \rightarrow \text{Image}$ 多階層の潜在変数
(弱) 教師あり	<b>cGAN</b> [Mirza+2014] <b>AC-GAN</b> [Odena+2017] $z, y \rightarrow G \rightarrow \text{Image}$ 教師情報に制約	$z, C \rightarrow \otimes \rightarrow G \rightarrow \text{Image}$ $y \rightarrow \text{Supervision}$	

# GANの発展

- 4つの発展



**潜在変数構造の工夫**  
操作性向上  
cGAN, AC-GAN, InfoGAN,  
CFGAN, DTLC-GAN

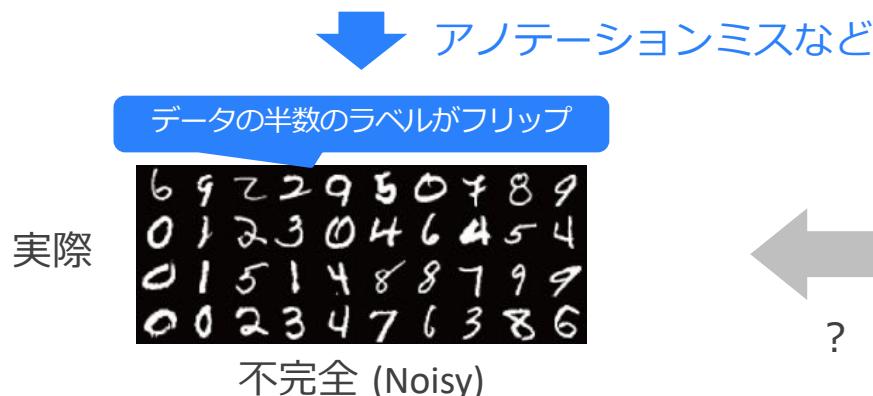
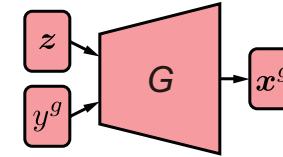
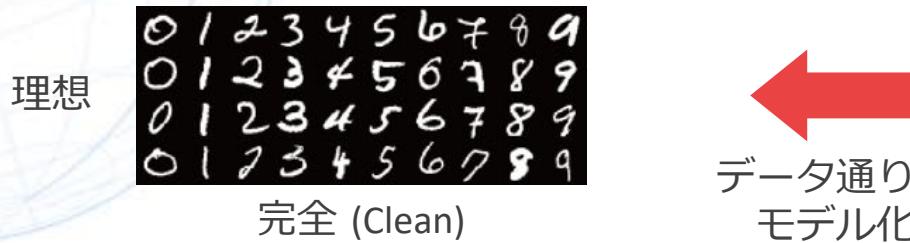
**データ設定の工夫**  
不完全データに対応  
rGAN, CP-GAN, S<sup>3</sup>GAN,  
AmbientGAN, NR-GAN

**ネットワークの工夫**  
高解像度化  
DCGAN, LAPGAN, StackGAN,  
SAGAN, ProGAN, StyleGAN,  
MSG-GAN, StyleGAN2

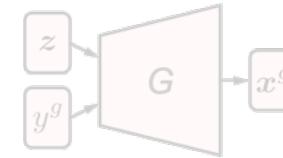
**目的関数の工夫**  
学習の安定化  
LSGAN, WGAN, WGAN-GP,  
SN-GAN, R<sub>1</sub> GP, Hinge GAN,  
Relativistic GAN, (BigGAN)

# GANの課題④ 1/4

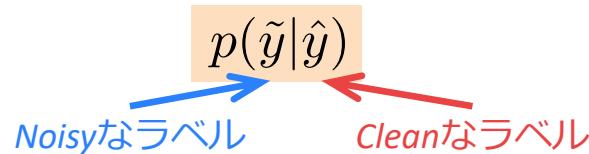
- 学習データは完全であることを仮定
  - 例 1：ラベルがCleanであることを仮定



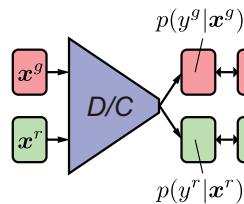
どう学習？



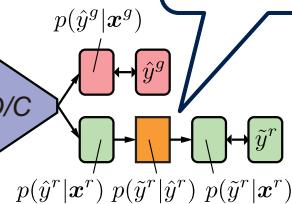
- Label-Noise Robust GAN [Kaneko+2019]
  - Noise Transition ModelをAC-GAN [Odena+2017] とcGAN [Mirza+2014, Miyato+2018] に導入



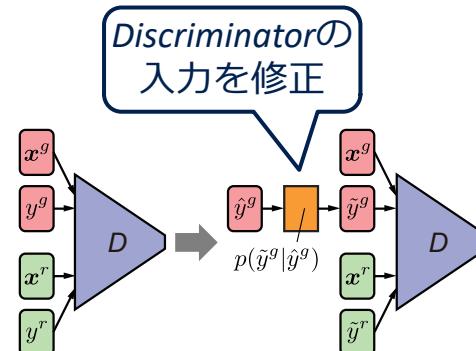
→ モデルの入出力を修正



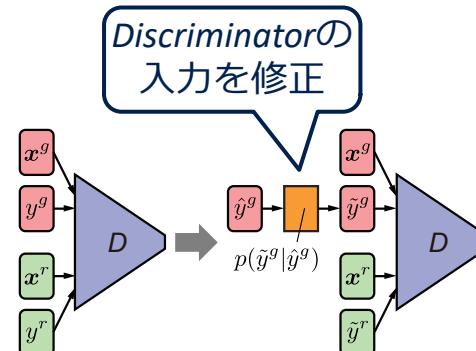
AC-GAN  
(従来)



rAC-GAN  
(提案)



cGAN  
(従来)



rcGAN  
(提案)

T. Kaneko et al., "Label-Noise Robust Generative Adversarial Networks," CVPR 2019.

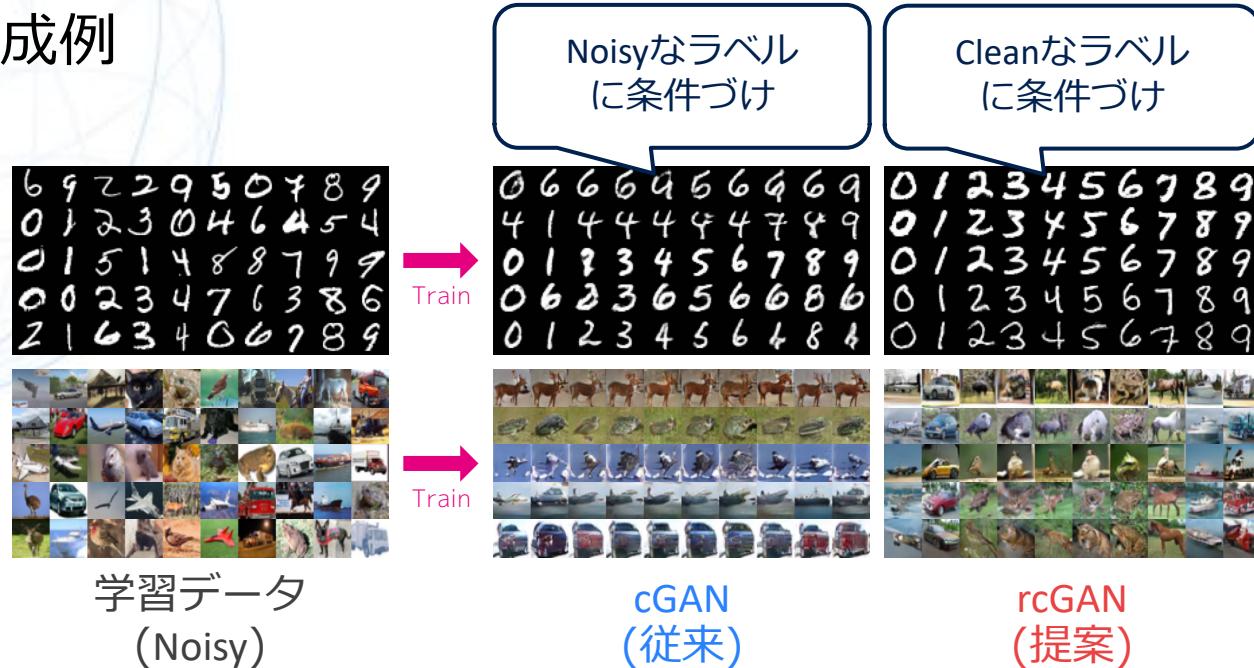
A. Odena et al., "Conditional Image Synthesis With Auxiliary Classifier GANs," ICML 2017.

M. Mirza & S. Osindero, "Conditional Generative Adversarial Nets," arXiv 2014.

T. Miyato & M. Koyama, "cGANs with Projection Discriminator," ICLR 2018.

# rGAN (続き)

- 画像生成例



ラベルレノイズに頑健な画像生成が可能

# GANの課題④ 2/4

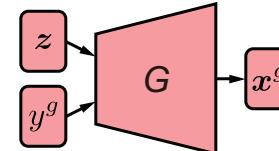
- 学習データは完全であることを仮定
  - 例2：Class-Separateであることを仮定

理想

0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9

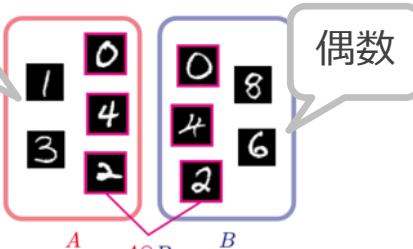
完全 (Class-Separate)

離散的なラベルに基づきモデル化



Generator

実際



5以下

偶数

A

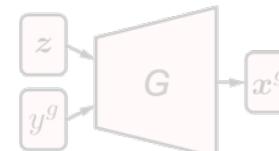
$A \cap B$

B

不完全 (Class-Overlapping)

ラベルの基準が曖昧

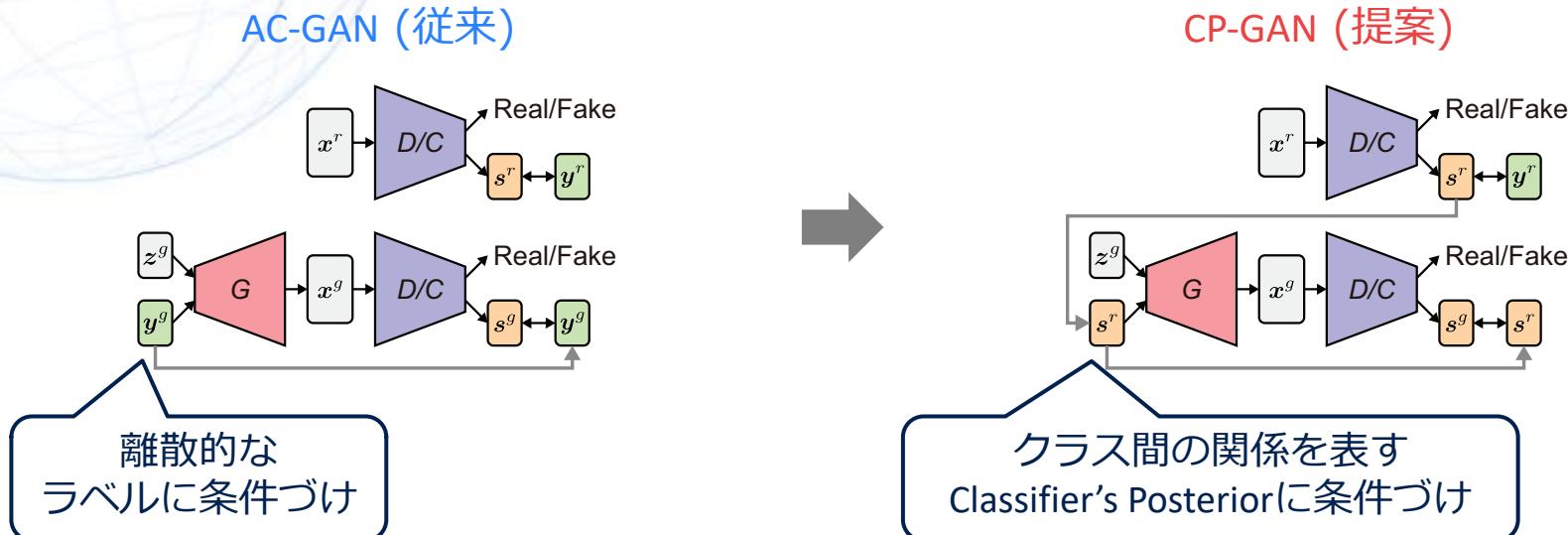
どう学習？



Generator

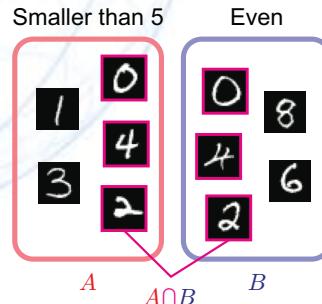
  
Knitwear ? Sweater  
Clothing1M  
Xiao et al.,  
CVPR 2015.

- Classifier's Posterior GAN [Kaneko+2019]
  - Classifier's Posteriorを使いAC-GAN [Odena+2017] の入力と目的関数を再設計

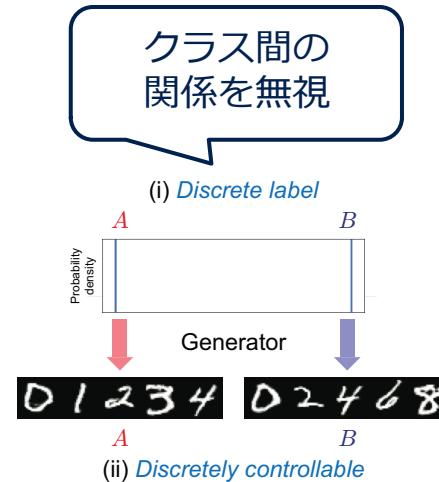


# CP-GAN (続き)

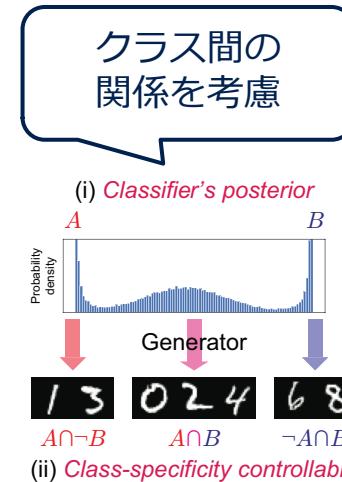
- 画像生成例



学習データ  
(Class-Overlapping)



AC-GAN  
(従来)



CP-GAN  
(提案)

コード: <https://github.com/takuhirok/CP-GAN/>

クラス間の関係を考慮した画像生成が可能

# GANの課題④ 3/4

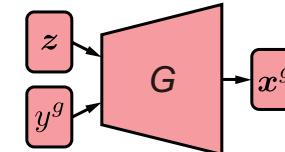
- 学習データは完全であることを仮定
  - 例 3：全データにラベル付与を仮定

理想



全データにラベル付与

ラベルを用いて  
モデル化



Generator

アノテーションが不完全

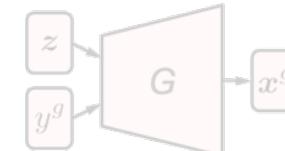
実際



不完全 (少数ラベル)

?

どう学習？



Generator

- Semi-Supervised GAN with Self-Supervision [Lucic+2019]
  - 準教師あり学習と自己教師あり学習をGANに導入
- 生成画像例

BigGAN  
(従来)  
全ラベルを使用



S<sup>3</sup>GAN  
(提案)  
10%のラベルを使用



少数ラベルだけでも全ラベルありに匹敵する画像生成が可能

# GANの課題④ 4/4

- 学習データは完全であることを仮定
  - 例 4 : 画像が“Clean”であることを仮定

理想



完全 (Clean)

実際



不完全 (Noisy)

データ通りに  
モデル化



*Image  
Generator*

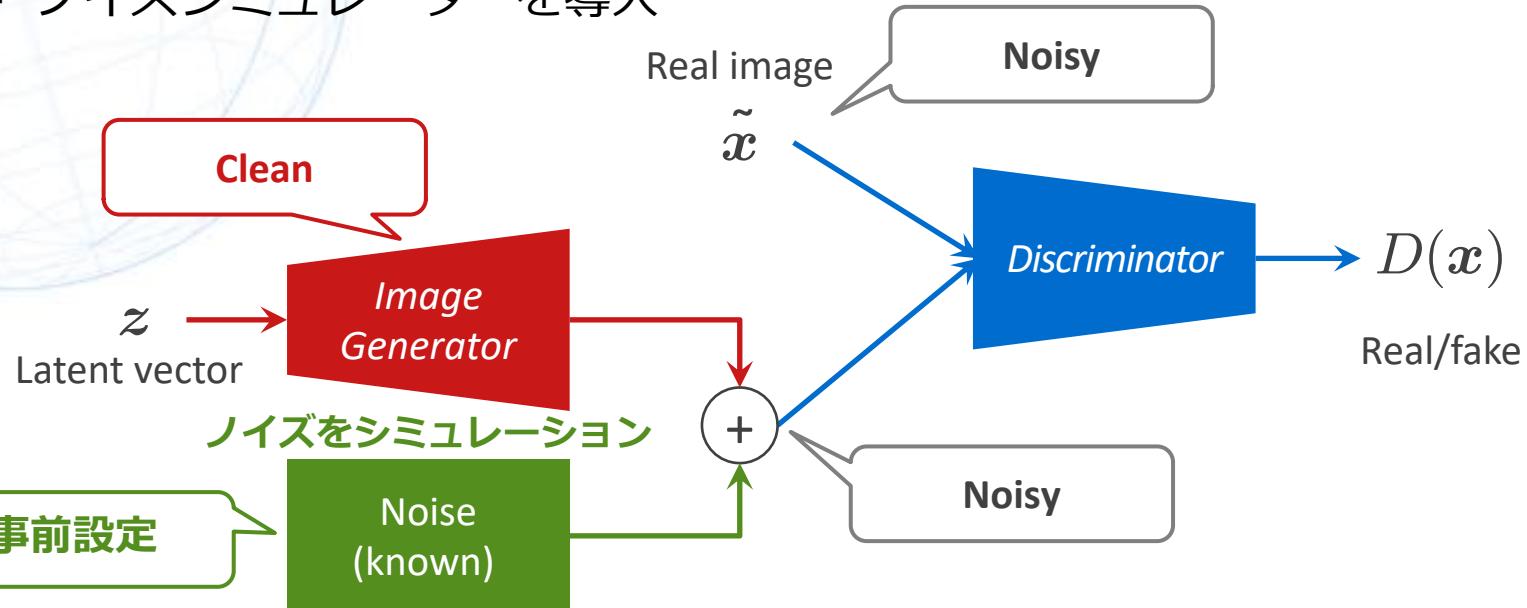
どう学習？



*Image  
Generator*

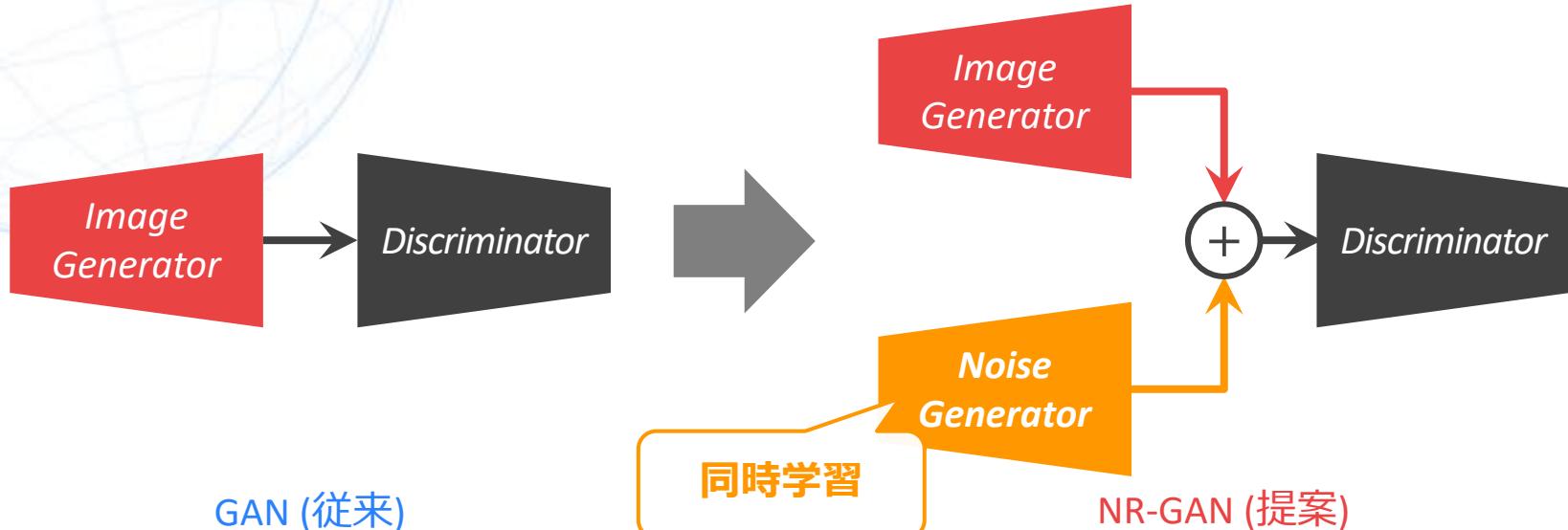
# AmbientGAN

- AmbientGAN [Bora+2018]
  - ノイズシミュレーターを導入



ノイズパラメータを事前に知っていることが必要

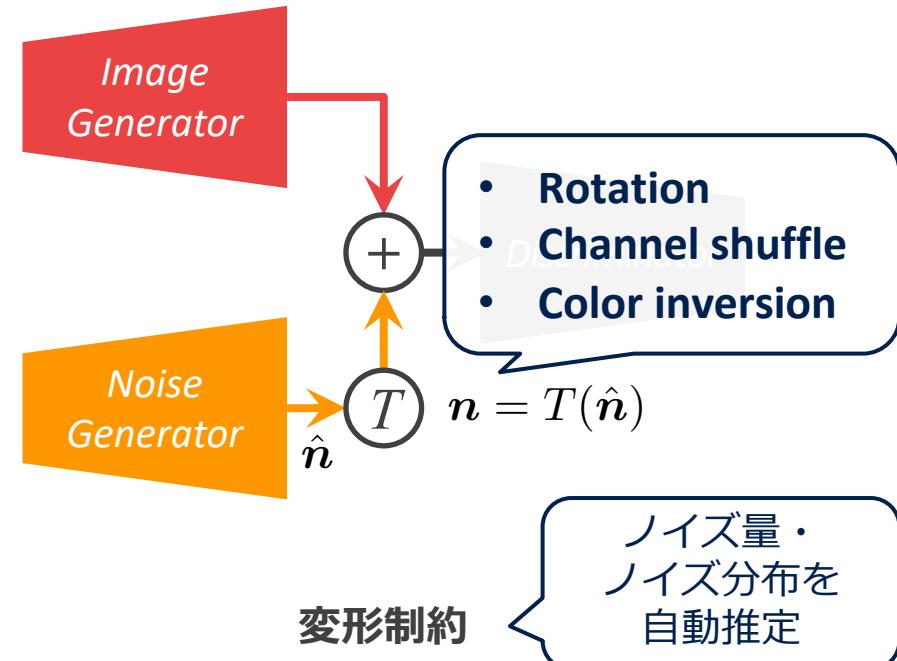
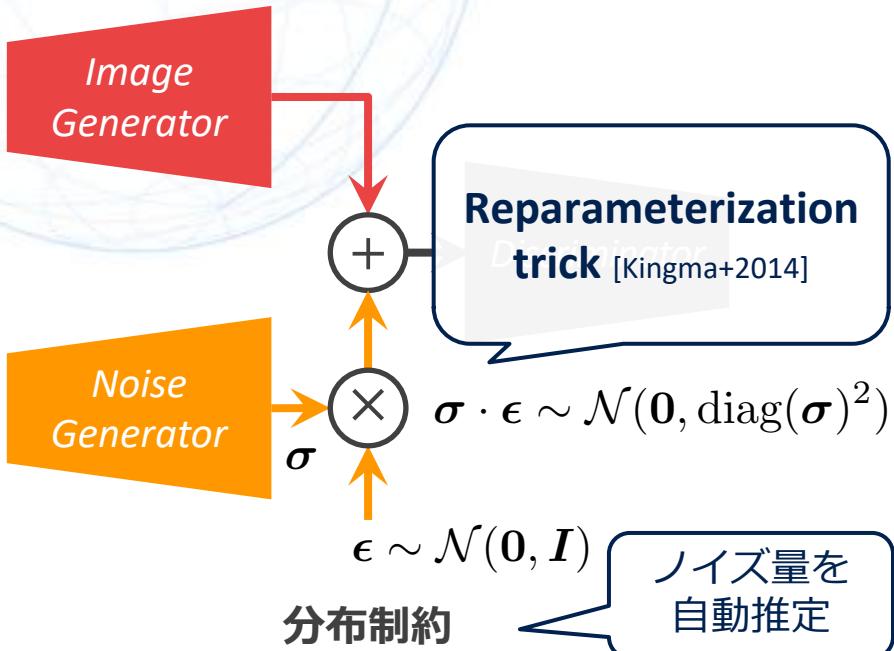
- Noise Robust GAN [Kaneko+2020]
  - 画像生成器とノイズ生成器の2つのGeneratorから構成されるモデルを導入



どうやって画像とノイズを別々に生成するか？

# NR-GAN (続き)

- ノイズ生成器に分布・変形制約を導入



T. Kaneko et al., "Noise Robust Generative Adversarial Networks," CVPR 2020.  
 Kingma & Welling, "Auto-Encoding Variational Bayes," ICLR 2014.

# NR-GAN (続き)

- 画像生成例

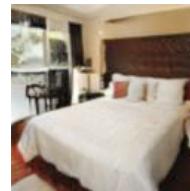
同一モデルで学習

画像非依存のノイズ

Additive Gaussian noise



Brown Gaussian noise



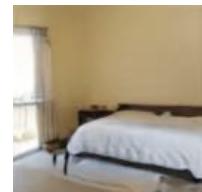
訓練データ

GAN  
(従来)

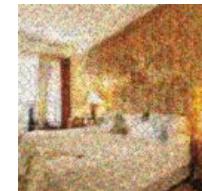
NR-GAN  
(提案)

同一モデルで学習

画像依存のノイズ



Multiplicative Gaussian noise



Poisson noise



訓練データ

GAN  
(従来)

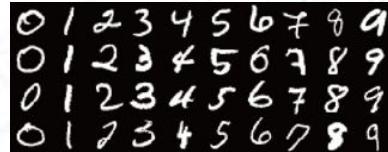
NR-GAN  
(提案)

コード: <https://github.com/takuhirok/NR-GAN/>

ノイズに頑健な画像生成が可能

# データ設定のまとめ

- GANの課題④
  - 学習データは完全であることを仮定



ラベルが  
①Clean  
②Class-Separate  
③全データにラベル



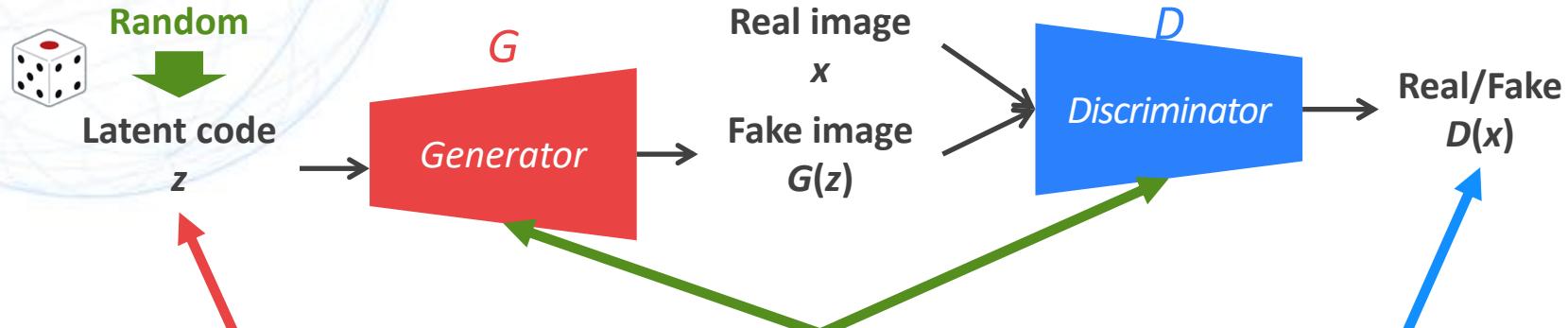
画像がClean

- データ設定の発展



# GANの発展 (まとめ)

- 4つの発展



**潜在変数構造の工夫**  
操作性向上  
cGAN, AC-GAN, InfoGAN,  
CFGAN, DTLC-GAN

**データ設定の工夫**  
不完全データに対応  
rGAN, CP-GAN, S<sup>3</sup>GAN,  
AmbientGAN, NR-GAN

**ネットワークの工夫**  
高解像度化  
DCGAN, LAPGAN, StackGAN,  
SAGAN, ProGAN, StyleGAN,  
MSG-GAN, StyleGAN2

**目的関数の工夫**  
学習の安定化  
LSGAN, WGAN, WGAN-GP,  
SN-GAN, R<sub>1</sub> GP, Hinge GAN,  
Relativistic GAN, (BigGAN)

※GANの研究は近年非常に盛んであり他にも様々な面白い試みあり

GAN  
の基礎

GANの  
発展

GANの  
応用

時間の都合で割愛  
(MIRUのスライド資料<sup>†</sup>ご参照)

<sup>†</sup> <http://www.kecl.ntt.co.jp/people/kaneko.takuhiro/#talks>

# まとめ

- 基礎：GANとは？
  - 騙す $G$  vs. 騙されない $D$ ：敵対的に学習



- 発展：様々な拡張モデルが提案

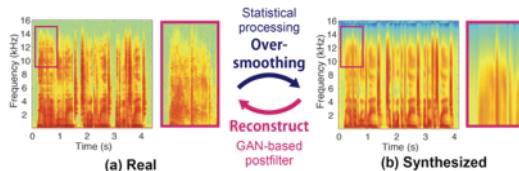
潜在変数構造の工夫  
操作性向上

ネットワークの工夫  
高解像度化

目的関数の工夫  
学習の安定化

データ設定の工夫  
不完全データに対応

- 応用：GANによって精度向上・新しいタスクの実現



高品質なスペクトログラム変換 [Kaneko+2017]



ペアデータなし  
で変換器を学習

今後も様々なタスクの  
解決に役立つ可能性あり！

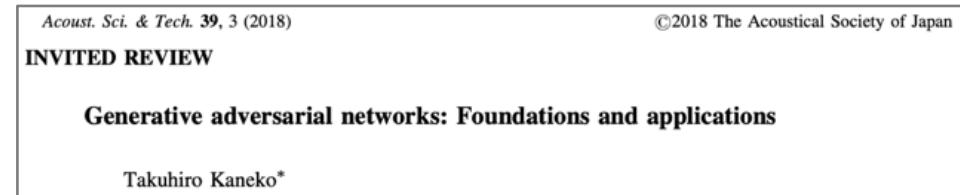
## 共同研究・インターンシップ募集 ぜひ一緒にGAN研究を盛り上げていきましょう！

E-mail: takuhiro.kaneko.tb at hco.ntt.co.jp

- 解説記事



[https://www.jstage.jst.go.jp/article/jasi/74/4/74\\_208/](https://www.jstage.jst.go.jp/article/jasi/74/4/74_208/)



[https://www.jstage.jst.go.jp/article/ast/39/3/39\\_E183003/\\_pdf/-char/en](https://www.jstage.jst.go.jp/article/ast/39/3/39_E183003/_pdf/-char/en)

- スライド資料・デモ動画など

**Takuhiro Kaneko**  
Distinguished Researcher  
NTT Communication Science Laboratories, NTT Corporation  
takuhiro.kaneko.tb at hco.ntt.co.jp  
[Google Scholar]

<http://www.kecl.ntt.co.jp/people/kaneko.takuhiro/index.html>

