

# Generating Diverse High-Fidelity Images with vQ-VAE-2

**Ali Razavi\***  
DeepMind  
alirazavi@google.com

**Aäron van den Oord\***  
DeepMind  
avdnoord@google.com

**Oriol Vinyals**  
DeepMind  
vinyals@google.com

論文URL:<https://arxiv.org/pdf/1906.00446.pdf>

フルペーパー：[https://drive.google.com/file/d/1H2nr\\_Cu7OK18tRemsWn\\_6o5DGMNYentM/view](https://drive.google.com/file/d/1H2nr_Cu7OK18tRemsWn_6o5DGMNYentM/view)

# 背景

深層学習による2つの生成モデルタイプ

- Generative Adversarial Networks (GANs) などの暗黙(implicit)のモデル  
GANは高品質，高解像度な画像を生成できるようになっているが，  
真の分布の多様性を捉えられない，多様性の欠如(mode collapse)を抱えている
- VAE, フローベース，自己回帰モデルなどの尤度を用いたモデル  
尤度ベースは，負の対数尤度NLLを最適化する問題であり，各サンプルの確率に  
対して最大化するため，mode collapseがない利点がある。

負の対数尤度NLL

$$L(\mathbf{y}) = -\log(\mathbf{y})$$

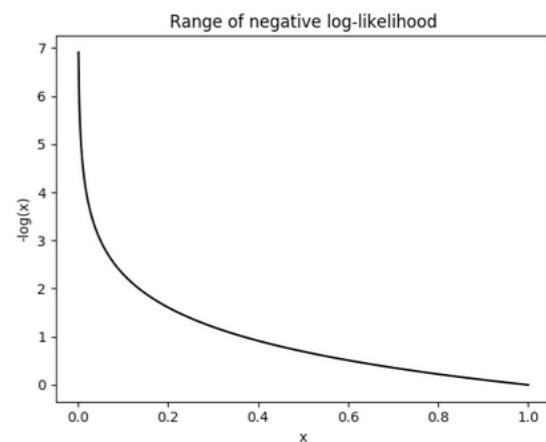


Figure: The loss function reaches infinity when input is 0, and reaches 0 when input is 1.

# 背景

しかし、ピクセル空間における尤度推定を直接行うことは困難

- ・画像の品質は異なり、NLLを用いてクラス間の比較を行うことはできない
- ・画像全域を把握できない

バイアスやマルチスケールを用いた手法があるが、より良い手法が求められる

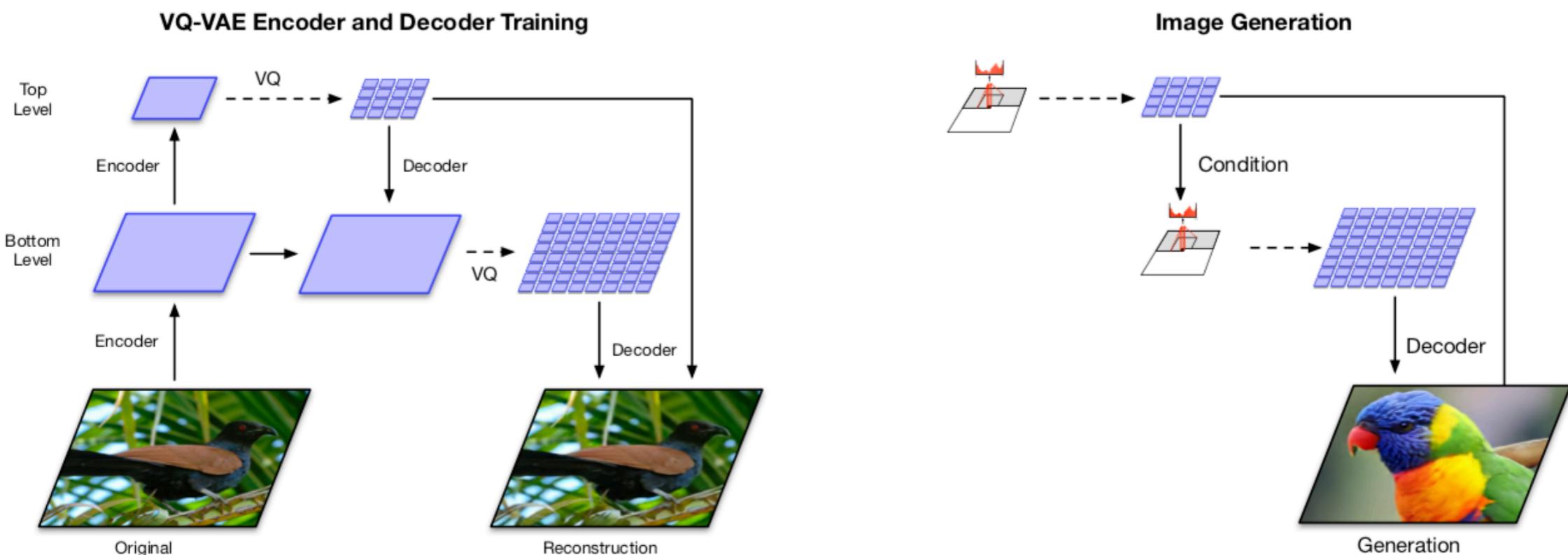
# 論文：Generating Diverse High-Fidelity Images with VQ-VAE-2の概要

## 提案手法

VQ-VAEとPixelCNNを用いた生成モデルを提案

VQ-VAEの階層化と、PixelCNNによる尤度推定

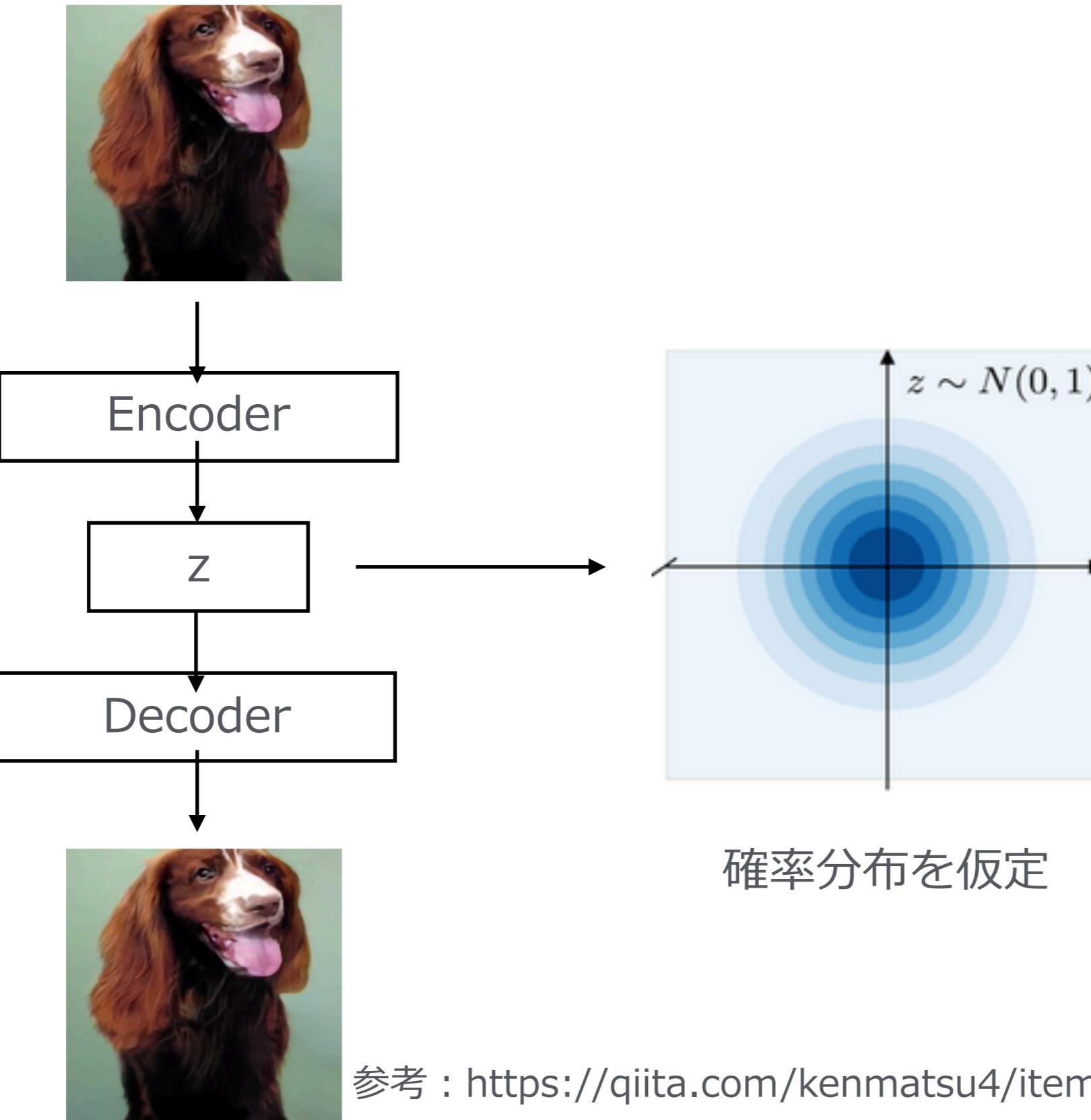
- 生成画像の解像度向上・多様性の獲得・一般的な評価が可能になった。



VQ-VAE-2の概要

# AE • VAE

## Variational AutoEncoder



# VQ-VAE

Vector Quantization Variational AutoEncoder

Quantize:量子化. アナログ信号などの連續量を, 整数などの離散値で近似的に表現すること. 値の精度を落とし, 粗い区間に分け直すこと.

$$\text{Quantize}(E(\mathbf{x})) = \mathbf{e}_k \quad \text{where } k = \arg \min_j ||E(\mathbf{x}) - \mathbf{e}_j|| \quad (1)$$

$E(\mathbf{x})$ : 画像 $\mathbf{x}$ をエンコーダに入力した際の出力ベクトル (潜在ベクトル $\mathbf{z}$ )  
コードブックへのマッピングすることで, 離散的な潜在空間に圧縮

# VQ-VAE

論文 : <https://arxiv.org/pdf/1711.00937.pdf>

## 損失関数

$$\mathcal{L}(\mathbf{x}, D(\mathbf{e})) = \|\mathbf{x} - D(\mathbf{e})\|_2^2 + \|sg[E(\mathbf{x})] - \mathbf{e}\|_2^2 + \beta \|sg[\mathbf{e}] - E(\mathbf{x})\|_2^2 \quad (2)$$



計算式

$$N_i^{(t)} := N_i^{(t-1)} * \gamma + n_i^{(t)}(1 - \gamma), \quad m_i^{(t)} := m_i^{(t-1)} * \gamma + \sum_j^{n_i^{(t)}} E(x)_{i,j}^{(t)}(1 - \gamma), \quad e_i^{(t)} := \frac{m_i^{(t)}}{N_i^{(t)}}$$

e: 量子化されたコード

E: エンコーダ関数

D: デコーダ関数

sg: stop-gradient

$\beta$  : ハイパーパラメータ,

コードブックへのマッピングをコントロール

$n_i^{(t)}$ : コードブックiに量子化される

ミニバッチ内の潜在ベクトル

$\gamma : 0.99$

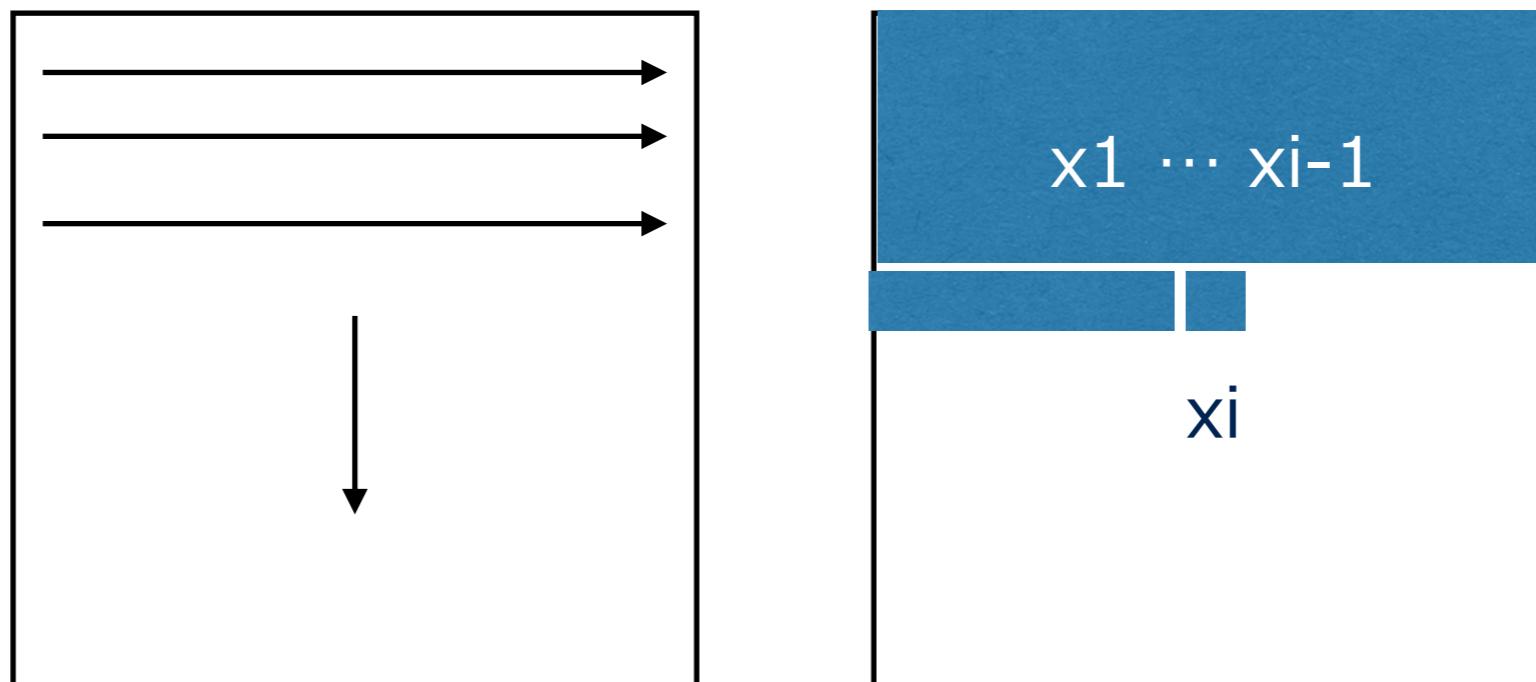
# PixelCNN 論文：<https://arxiv.org/pdf/1601.06759.pdf>).

PixelCNN：畳み込みオートエンコーダー。ピクセル・チャンネル単位の自己回帰型モデル

ラベルやタグ、ベクトルを入力とし、各ピクセルの対数尤度を最大にする潜在ベクトルが持つ情報を目的のクラスに条件付き生成できる。

出力はRGBごとに256クラスに分類→ガウシアンノイズよりはっきりした画像に

条件付き確率分布  $p(\mathbf{x}|\mathbf{h}) = \prod_{i=1}^{n^2} p(x_i|x_1, \dots, x_{i-1}, \mathbf{h})$



PixelCNNの画像生成の流れ

# 学習アルゴリズム

VQ-VAEを階層構造にすることで、高解像度を可能にする

---

## Algorithm 1 VQ-VAE training (stage 1)

---

**Require:** Functions  $E_{top}$ ,  $E_{bottom}$ ,  $D$ ,  $\mathbf{x}$   
(batch of training images)

1:  $\mathbf{h}_{top} \leftarrow E_{top}(\mathbf{x})$

▷ quantize with top codebook eq 1

2:  $\mathbf{e}_{top} \leftarrow Quantize(\mathbf{h}_{top})$

3:  $\mathbf{h}_{bottom} \leftarrow E_{bottom}(\mathbf{x}, \mathbf{e}_{top})$

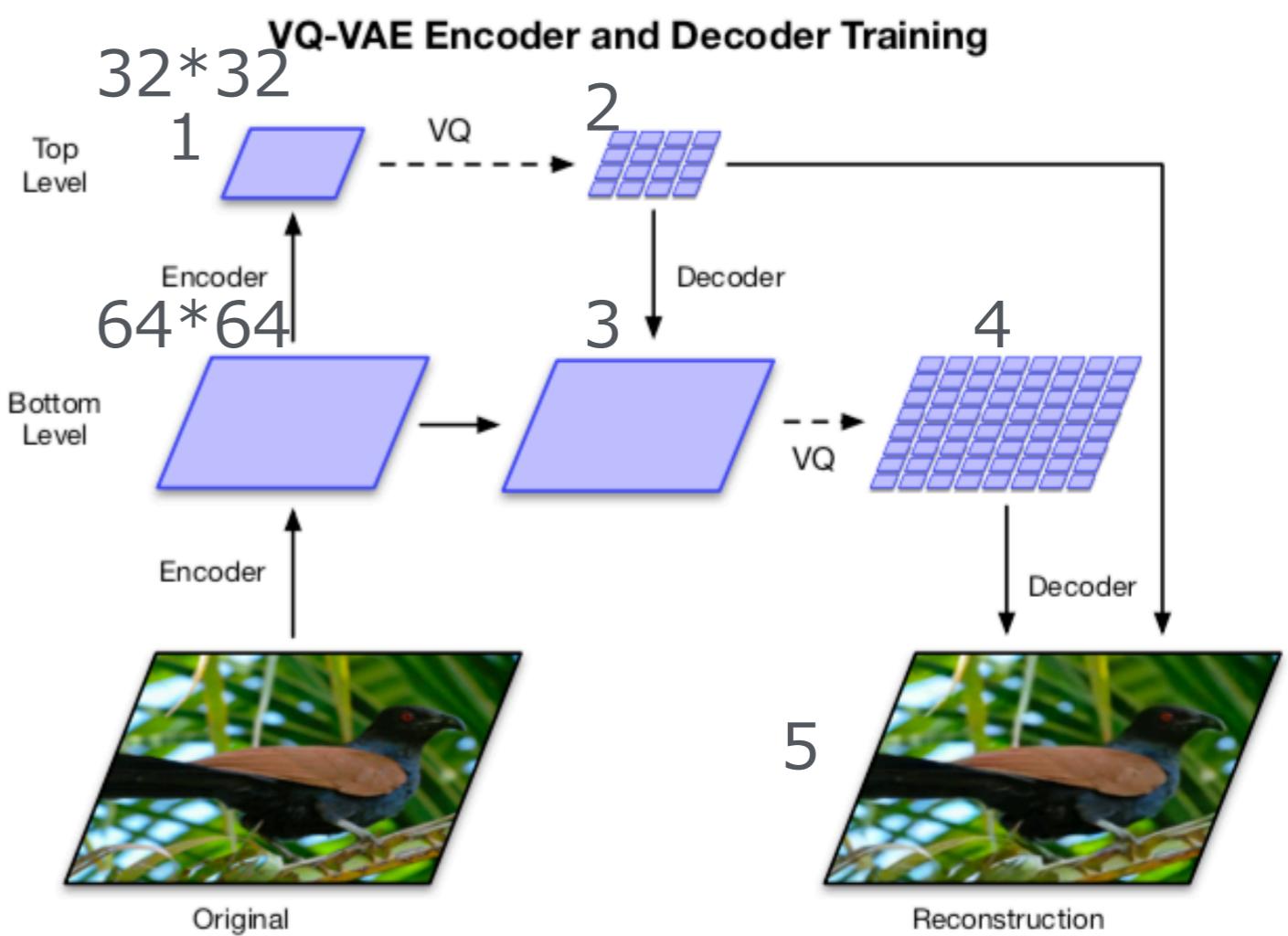
▷ quantize with bottom codebook eq 1

4:  $\mathbf{e}_{bottom} \leftarrow Quantize(\mathbf{h}_{bottom})$

5:  $\hat{\mathbf{x}} \leftarrow D(\mathbf{e}_{top}, \mathbf{e}_{bottom})$

▷ Loss according to eq 2

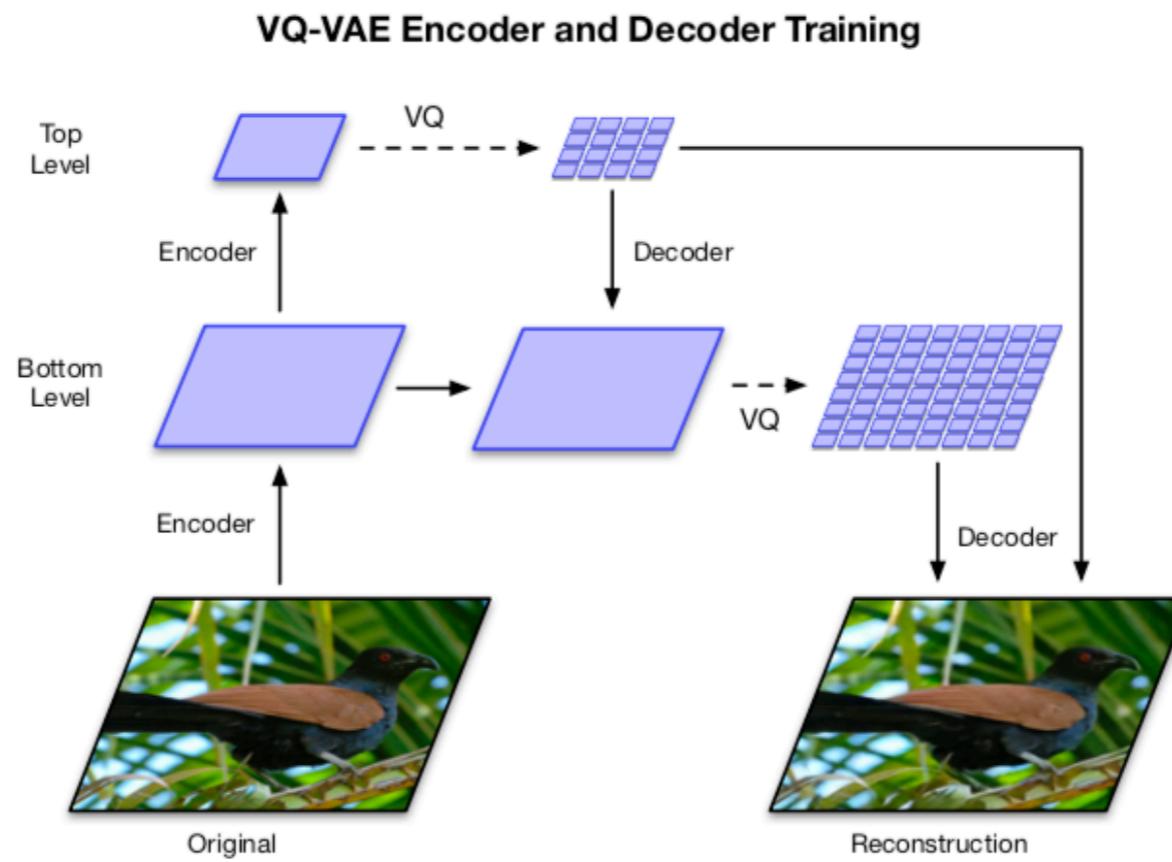
6:  $\theta \leftarrow Update(\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}))$



256\*256

VQ-VAEの構造

# モデルの構造



$h_{\text{top}}$

$h_{\text{top}}, h_{\text{middle}}$

$h_{\text{top}}, h_{\text{middle}}, h_{\text{bottom}}$

Original

# 学習アルゴリズム

オートエンコーダ付きPixelCNNを用いて潜在変数の事前分布を学習

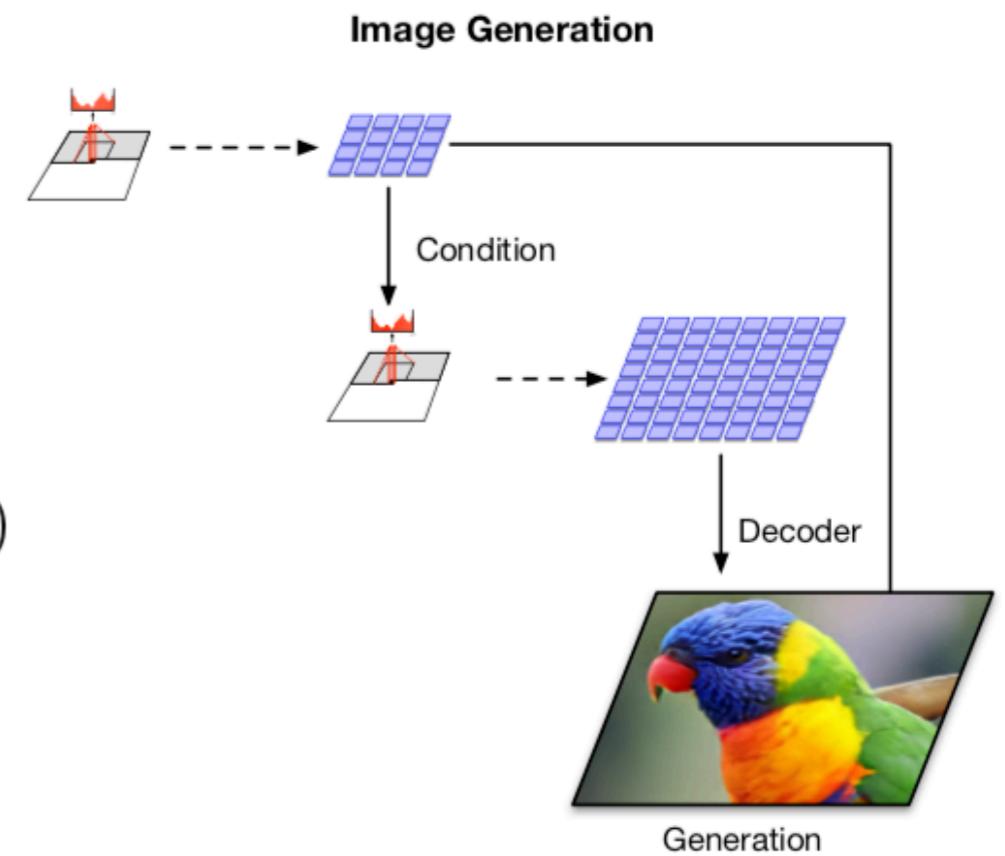
---

## Algorithm 2 Prior training (stage 2)

---

```
1:  $\mathbf{T}_{top}, \mathbf{T}_{bottom} \leftarrow \emptyset$                                 ▷ training set
2: for  $\mathbf{x} \in$  training set do
3:    $\mathbf{e}_{top} \leftarrow Quantize(E_{top}(\mathbf{x}))$ 
4:    $\mathbf{e}_{bottom} \leftarrow Quantize(E_{bottom}(\mathbf{x}, \mathbf{e}_{top}))$ 
5:    $\mathbf{T}_{top} \leftarrow \mathbf{T}_{top} \cup \mathbf{e}_{top}$ 
6:    $\mathbf{T}_{bottom} \leftarrow \mathbf{T}_{bottom} \cup \mathbf{e}_{bottom}$ 
7: end for
8:  $p_{top} = TrainPixelCNN(\mathbf{T}_{top})$ 
9:  $p_{bottom} = TrainCondPixelCNN(\mathbf{T}_{bottom}, \mathbf{T}_{top})$ 
  
    ▷ Sampling procedure
10: while true do
11:    $\mathbf{e}_{top} \sim p_{top}$ 
12:    $\mathbf{e}_{bottom} \sim p_{bottom}(\mathbf{e}_{top})$ 
13:    $\mathbf{x} \leftarrow D(\mathbf{e}_{top}, \mathbf{e}_{bottom})$ 
14: end while
```

---



PixelCNNの構造

# 実験

## データセット

- ImageNet 256\*256 (1400万枚, 2万クラス)

666~1000クラス使用, 枚数不明

- FFHQ 1024\*1024

性別, 肌の色, 年齢, 姿勢, 服装の多様な70000枚の画像

## 比較手法

- BigGAN deep

最大512\*512の高解像度で, 1000クラスを生成することができるSOTAモデル  
(DeepMind製)

## 実験 定量的評価

ImageNet

尤度ベースの生成モデルは、一般化できる客観的尺度としてNLLを用いることができ、過適合を評価できる。

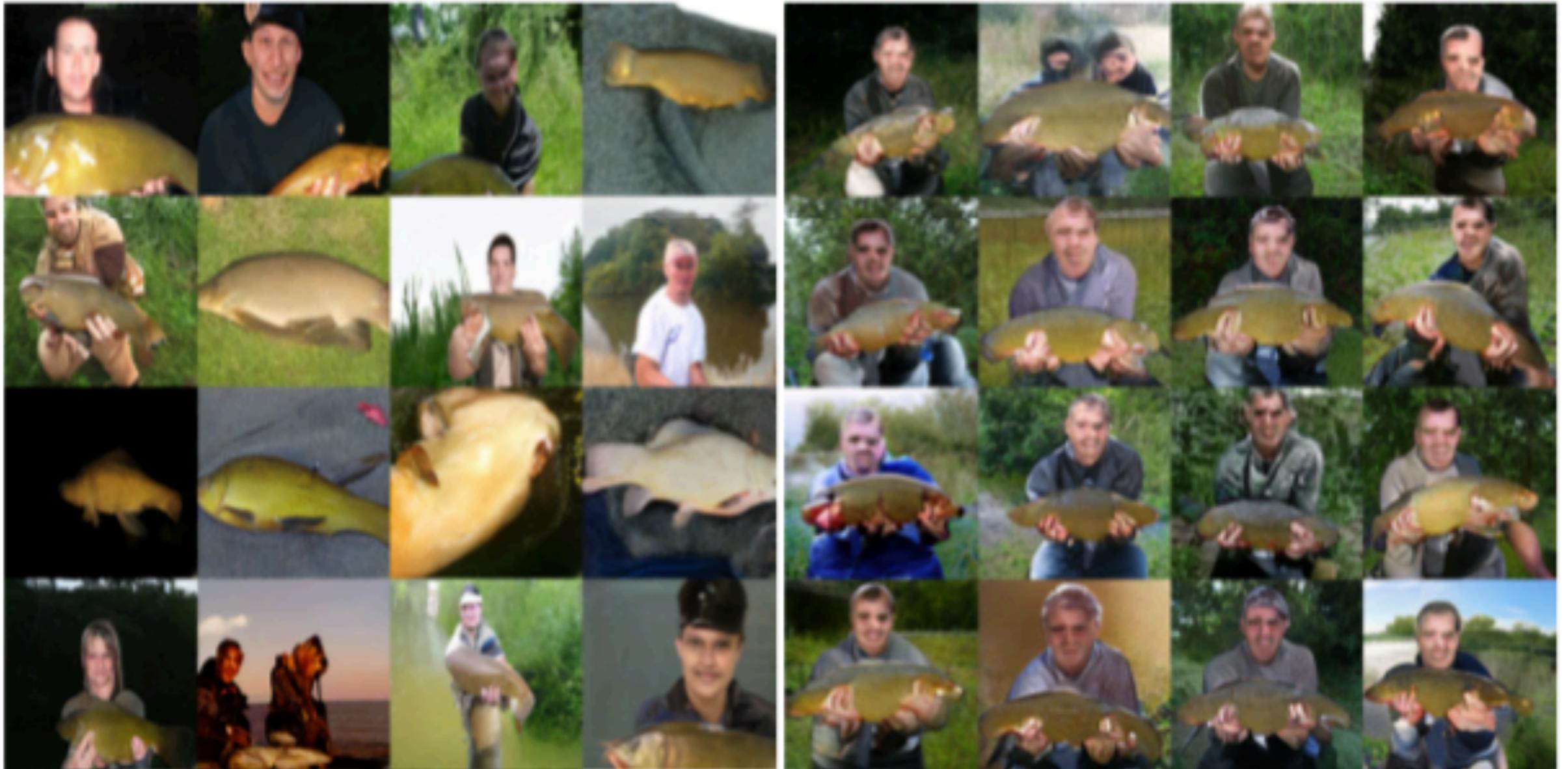
	Train NLL	Validation NLL	Train MSE	Validation MSE
Top prior	3.40	3.41	-	-
Bottom prior	3.45	3.45	-	-
VQ Decoder	-	-	0.0047	0.0050

生成モデルが生成した画像セットで学習した分類器を用いて評価

→品質と多様性を評価

	Top-1 Accuracy	Top-5 Accuracy
BigGAN deep	42.65	65.92
VQ-VAE	54.83	77.59
VQ-VAE after reconstructing	58.74	80.98
Real data	73.09	91.47

# 実験 定性的評価



VQ-VAE-2

Big GAN deep

# 実験 定性的評価



VQ-VAE-2

Big GAN deep

# 実験 定性的評価

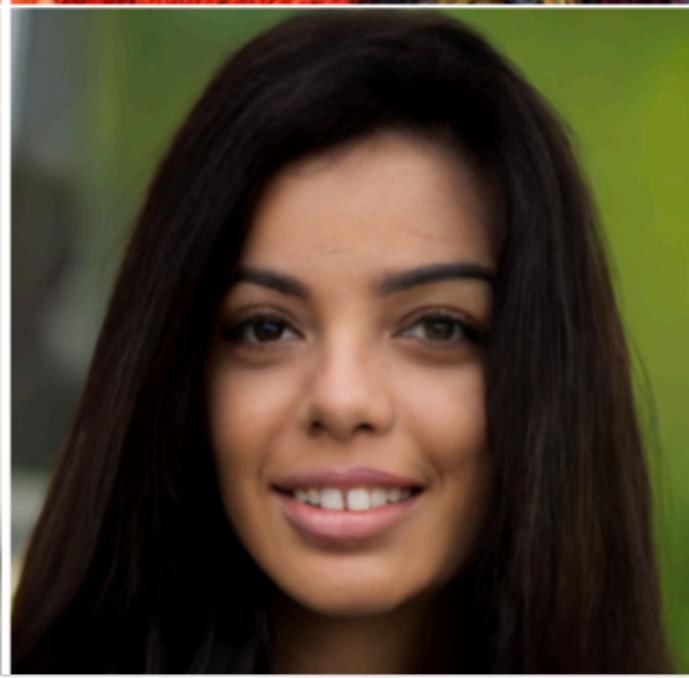
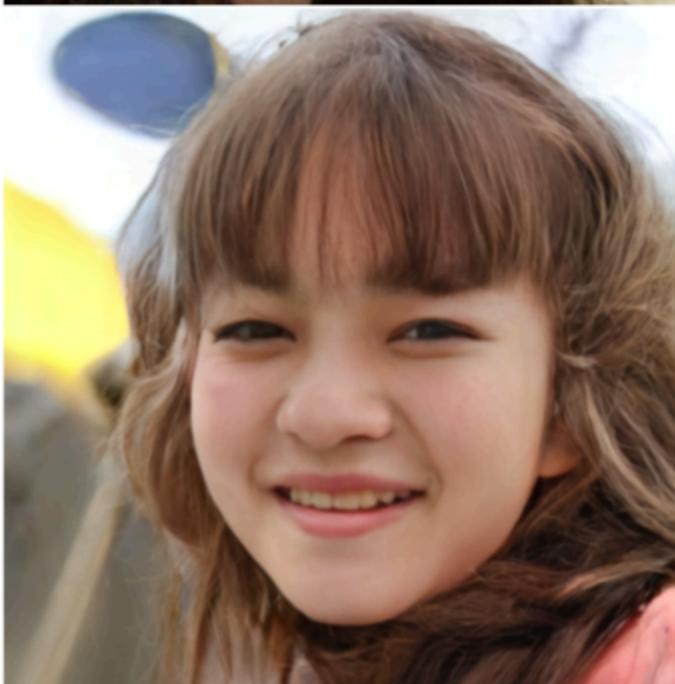
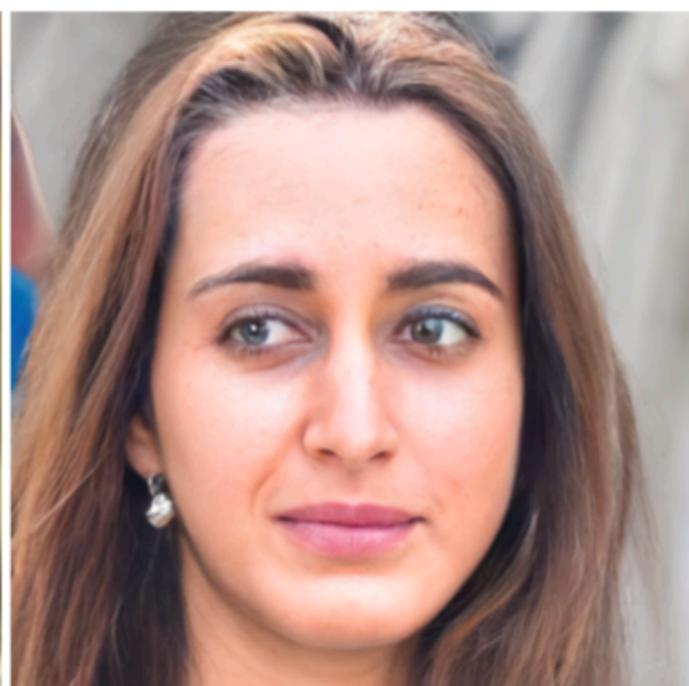
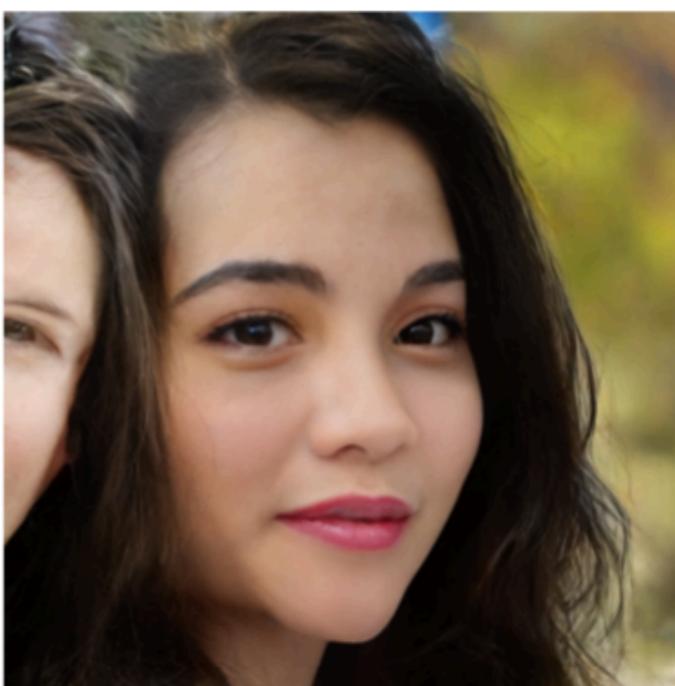


VQ-VAE-2

Big GAN deep

FFHQで学習した生成画像

1024 \* 1024



## 考察・まとめ

- ・他手法で見られる評価基準は一般化の問題を無視しており、本手法ではその点を尤度ベースの生成モデルを用いてNLLを基準にすることで改善した
- ・通常のVQ-VAEに階層マルチスケールな潜在マップを使用することで、生成する解像度を向上させ、BigGANよりも多様な画像を生成できるように
- ・多様性と品質の尺度はまだ研究段階であり、目視の定性的評価はまだ必要

生成画像例  
( $256 \times 256$ )



生成画像例  
( $256 \times 256$ )



潜在マップが，局所，大域でそれぞれ捉えている様子

