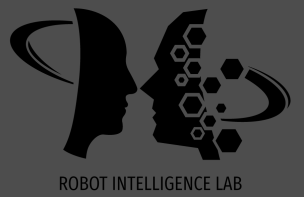


Generative Model

InfoGAN, VQ-VAE, VQ-VAE2

Sungjoon Choi, Korea University

ROBOT INTELLIGENCE LAB

InfoGAN

"InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets," 2016

InfoGAN?

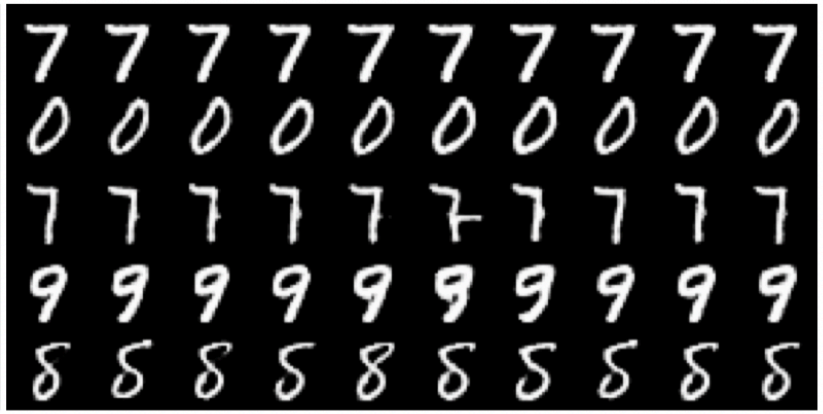
ROBOT INTELLIGENCE LAB

Information-theoretic extension to the GAN to learn **disentangled** representations

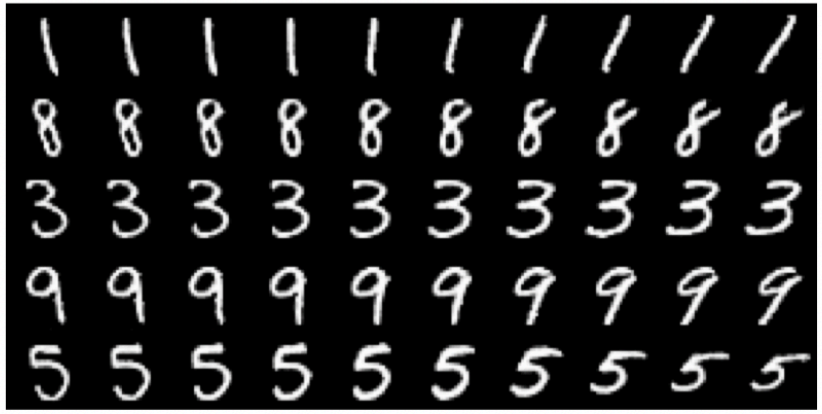
Disentangled Representation

ROBOT INTELLIGENCE LAB

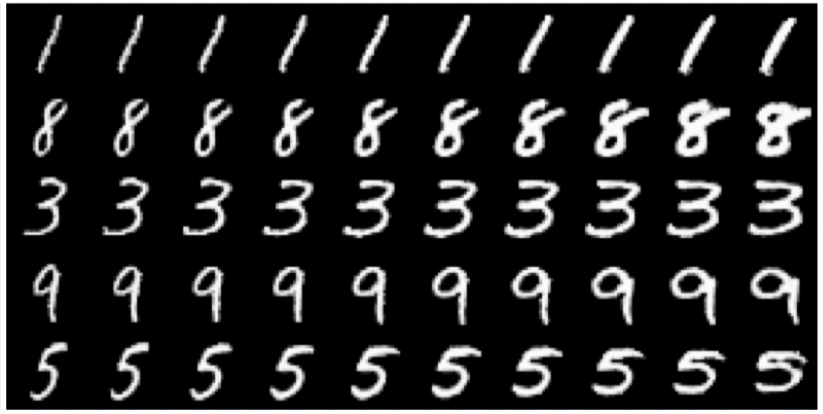
(a) Varying c_1 on InfoGAN (Digit type)



(b) Varying c_1 on regular GAN (No clear meaning)



(c) Varying c_2 from -2 to 2 on InfoGAN (Rotation)



(d) Varying c_3 from -2 to 2 on InfoGAN (Width)

Mutual Information

ROBOT INTELLIGENCE LAB

- The input noise vector is decomposed into z, source of incompressible noise, and c, the latent code.
- In standard GAN, the generator may ignore the latent code (i.e., $P_G(x \mid c) = P_G(x)$). To handle this, latent codes c and generator distribution $G(z, c)$ should have high mutual information. Hence, $I(c; G(z, c))$ should be high.
- The mutual information between X and Y, $I(X; Y)$ can be expressed as:

$$I(X; Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X)$$

  - Intuitively, $I(X; Y)$ is the amount of reduced uncertainty in X when Y is observed.
  - If X and Y are independent, $I(X; Y) = 0$ as knowing one reveals nothing about the other.
  - If X and Y are related by a deterministic, invertible function, then maximal mutual information is attained.

InfoGAN

ROBOT INTELLIGENCE LAB

- The objective of InfoGAN is as follows:

$$\min_G \max_D V_I(D,G) = V(D,G) - \lambda I(c; G(z,c))$$

- Intuitively, given $x \sim P_G(z,c)$, we want $P_G(c \mid x)$ to have a small entropy.
  - However, computing the posterior $P(c \mid x)$ is usually intractable.

$$\begin{aligned} I(c; G(z,c)) &= H(c) - H(c|G(z,c)) \\ &= \mathbb{E}_{x\sim G(z,c)}\left[\mathbb{E}_{c'\sim P(c'|x)}[\log P(c'|x)]\right] + H(c) \\ &= \mathbb{E}_{x\sim G(z,c)}\left[D_{KL}(P(\cdot|x)\|Q(\cdot|x)) + \mathbb{E}_{c'\sim P(c'|x)}[\log Q(c'|x)]\right] + H(c) \\ &\geq \mathbb{E}_{x\sim G(z,c)}\left[\mathbb{E}_{c'\sim P(c'|x)}[\log Q(c'|x)]\right] + H(c) \end{aligned}$$

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

$$H(Y|X) = \int P(X,Y) \log \frac{P(X)}{P(X,Y)} = -\int P(X,Y) \log P(Y|X)$$

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

Variational Bound Trick

ROBOT INTELLIGENCE LAB

$$
\begin{aligned}
I(c; G(z,c)) &= H(c) - H(c|G(z,c)) \\
&= \mathbb{E}_{x\sim G(z,c)}\left[\mathbb{E}_{c'\sim P(c'|x)}\left[\log P(c'|x)\right]\right] + H(c) \\
&= \mathbb{E}_{x\sim G(z,c)}\left[D_{KL}(P(\cdot|x)\|Q(\cdot|x)) + \mathbb{E}_{c'\sim P(c'|x)}\left[\log Q(c'|x)\right]\right] + H(c) \\
&\geq \mathbb{E}_{x\sim G(z,c)}\left[\mathbb{E}_{c'\sim P(c'|x)}\left[\log Q(c'|x)\right]\right] + H(c)
\end{aligned}
$$

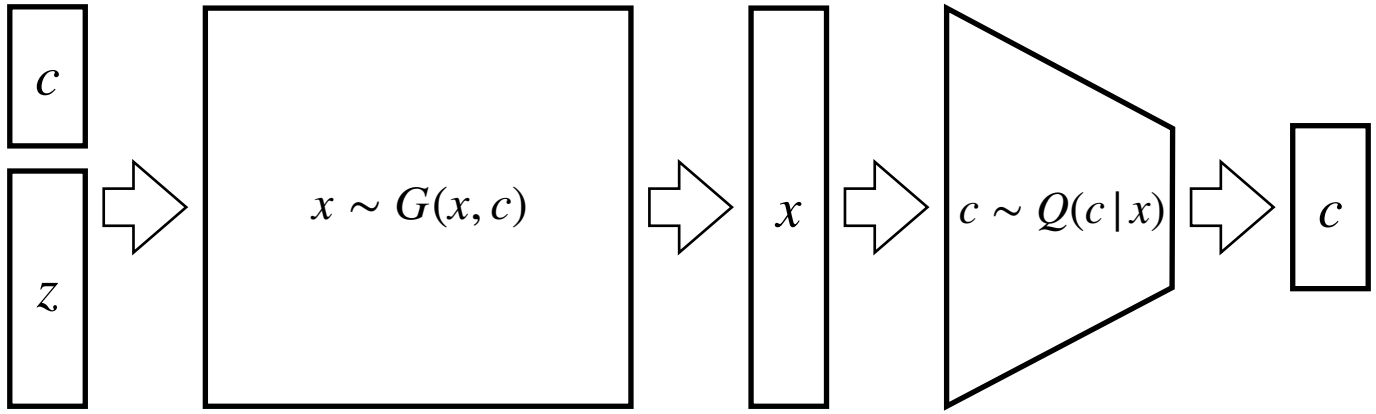
$$
\begin{aligned}
\mathbb{E}_{x\sim p(x)}\left[\log p(x)\right] &= \int p(x)\log p(x)dx \\
&= \int p(x)\log \frac{p(x)q(x)}{q(x)}dx \\
&= \int p(x)\log \frac{p(x)}{q(x)}dx + \int p(x)\log q(x)dx \\
&= D_{KL}(p(x)\|q(x)) + \mathbb{E}_{x\sim p(x)}\left[\log q(x)\right] \\
&\geq \mathbb{E}_{x\sim p(x)}\left[\log q(x)\right]
\end{aligned}
$$

Tractable Variational Lower Bound

ROBOT INTELLIGENCE LAB

- Under suitable regularity conditions:

$$L_I(G, Q) = \mathbb{E}_{x \sim G(z,c), c' \sim P(c|x)}[\log Q(c' \mid x)] = \mathbb{E}_{c \sim P(c), x \sim G(z,c)}[\log Q(c \mid x)]$$



Results

ROBOT INTELLIGENCE LAB

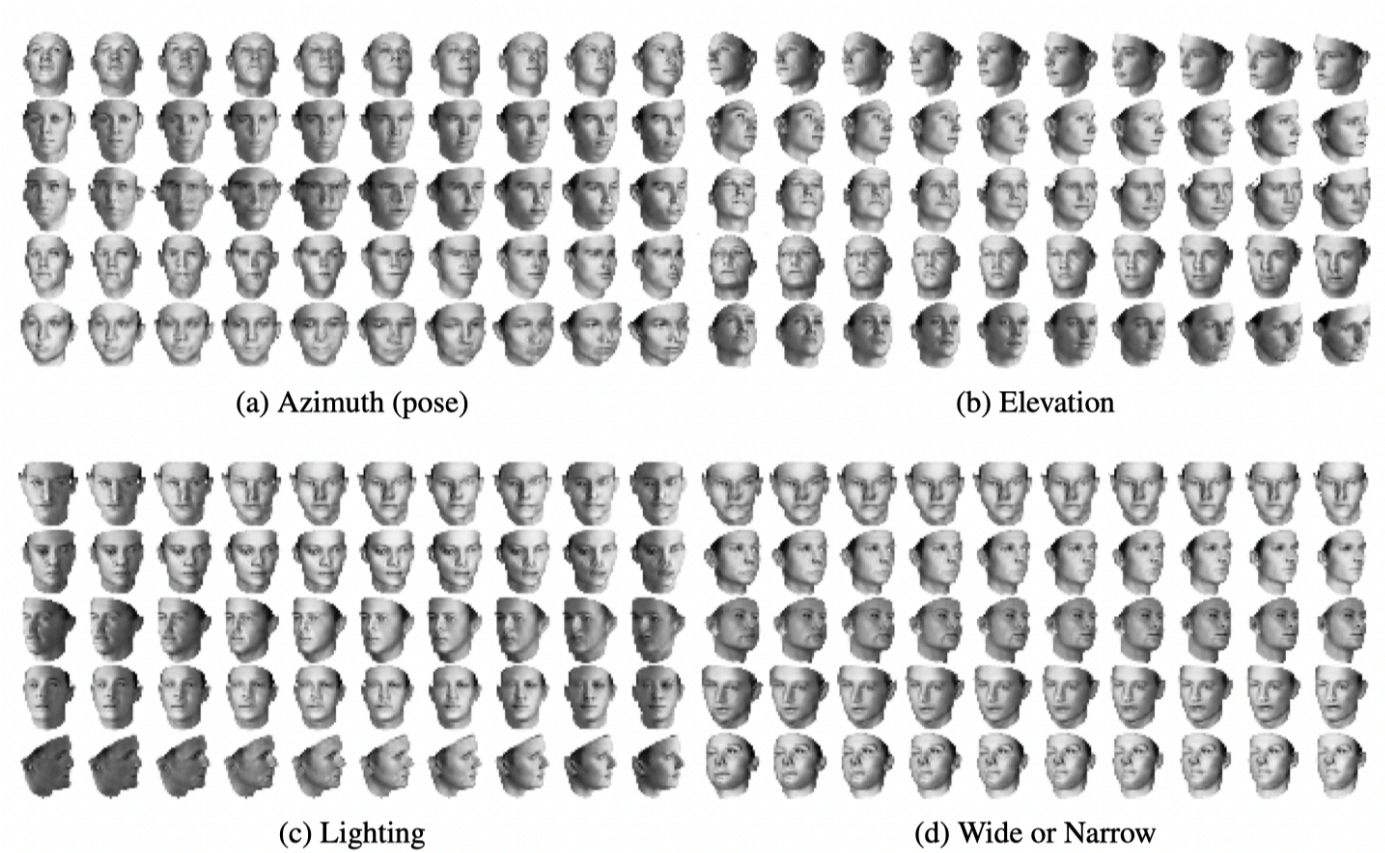


Figure 3: **Manipulating latent codes on 3D Faces:** We show the effect of the learned continuous latent factors on the outputs as their values vary from -1 to 1. In (a), we show that one of the continuous latent codes consistently captures the azimuth of the face across different shapes; in (b), the continuous code captures elevation; in (c), the continuous code captures the orientation of lighting; and finally in (d), the continuous code learns to interpolate between wide and narrow faces while preserving other visual features. For each factor, we present the representation that most resembles prior supervised results [7] out of 5 random runs to provide direct comparison.

Results

ROBOT INTELLIGENCE LAB

(a) Rotation (b) Width

Figure 4: **Manipulating latent codes on 3D Chairs:** In (a), we show that the continuous code captures the pose of the chair while preserving its shape, although the learned pose mapping varies across different types; in (b), we show that the continuous code can alternatively learn to capture the widths of different chair types, and smoothly interpolate between them. For each factor, we present the representation that most resembles prior supervised results [7] out of 5 random runs to provide direct comparison.

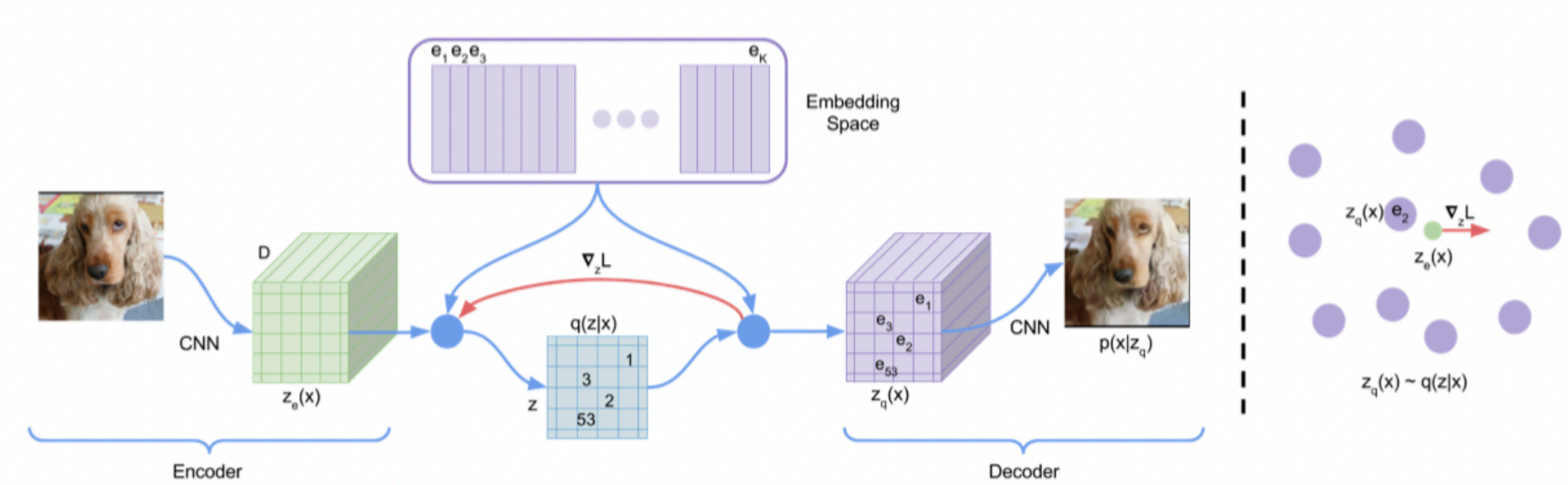
ROBOT INTELLIGENCE LAB

VQ-VAE

"Neural Discrete Representation Learning," 2018

VQ-VAE

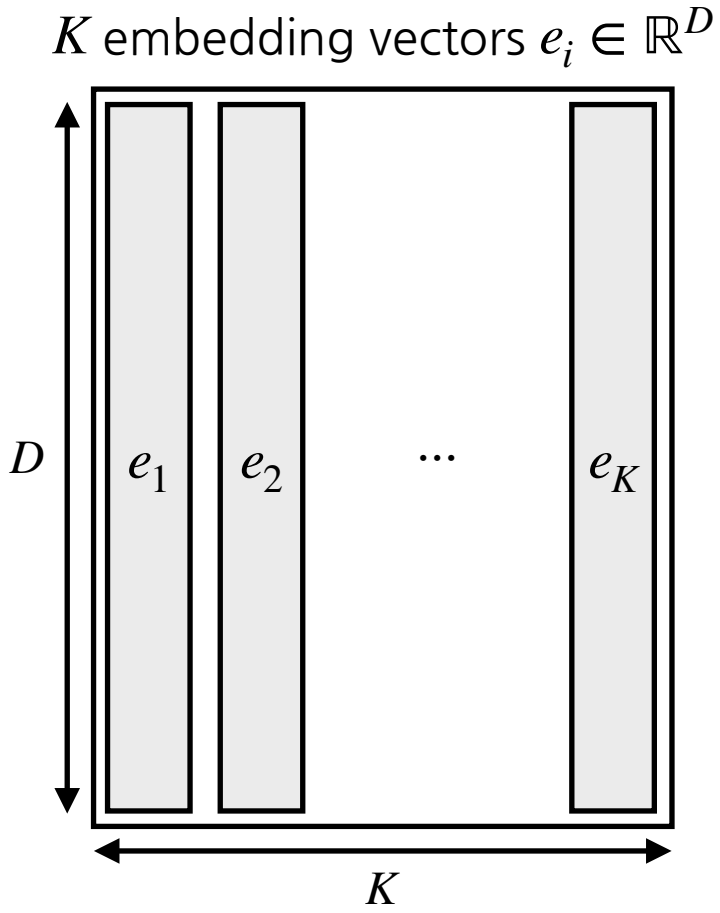
ROBOT INTELLIGENCE LAB



- VQ-VAE combines the variational auto-encoder (VAE) framework with discrete latent representations using vector quantization (VQ).

Discrete Latent Variables

ROBOT INTELLIGENCE LAB



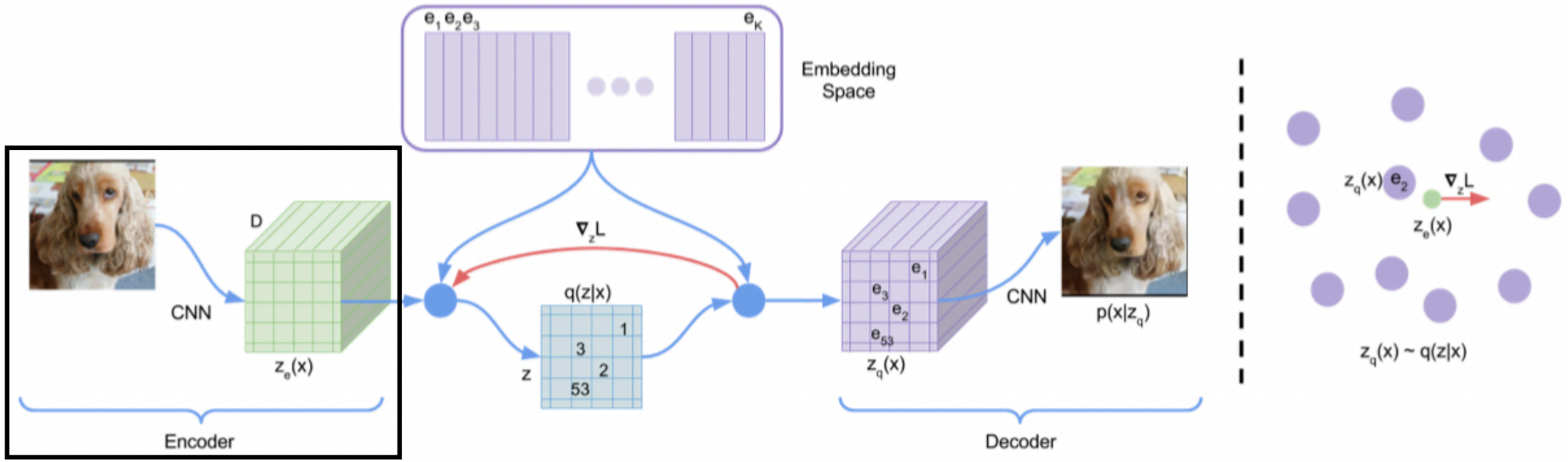
Discrete Latent Variables

ROBOT INTELLIGENCE LAB

- Latent embedding space consists of K embedding vectors $e_i \in \mathbb{R}^D$, $i \in 1,2,\ldots,K$.
- Inference
  - Given an input x, an encoder outputs $z_e(x) \in \mathbb{R}^D$ and the closest embedding vector e_k is selected.
  - In other words, $z_q(x) = e_k$ where where $k = \arg\min_j \|z_e(x) - e_j\|_2$.
- Learning
  - Since $\arg\min$ is not differentiable, the **straight-through estimator** that simply copies gradient from the decoder input $z_q(x)$ to encoder output $z(e)$.
  - To update embeddings, Vector Quantization (VQ) that uses the l_2 error to move the embedding vector towards the encoder outputs is used.

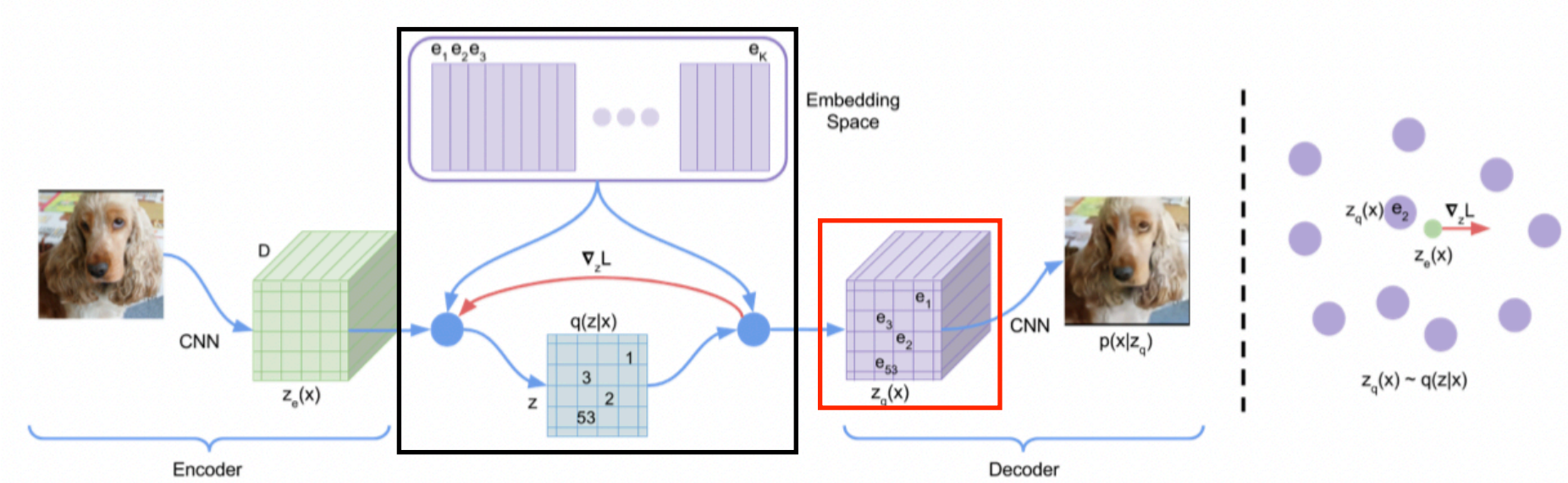
$$L = \log p(x|z_q(x)) + \|\mathrm{sg}[z_e(x)] - e\|_2^2 + \beta\|z_e(x) - \mathrm{sg}[e]\|_2^2,$$

Inference (1/3)



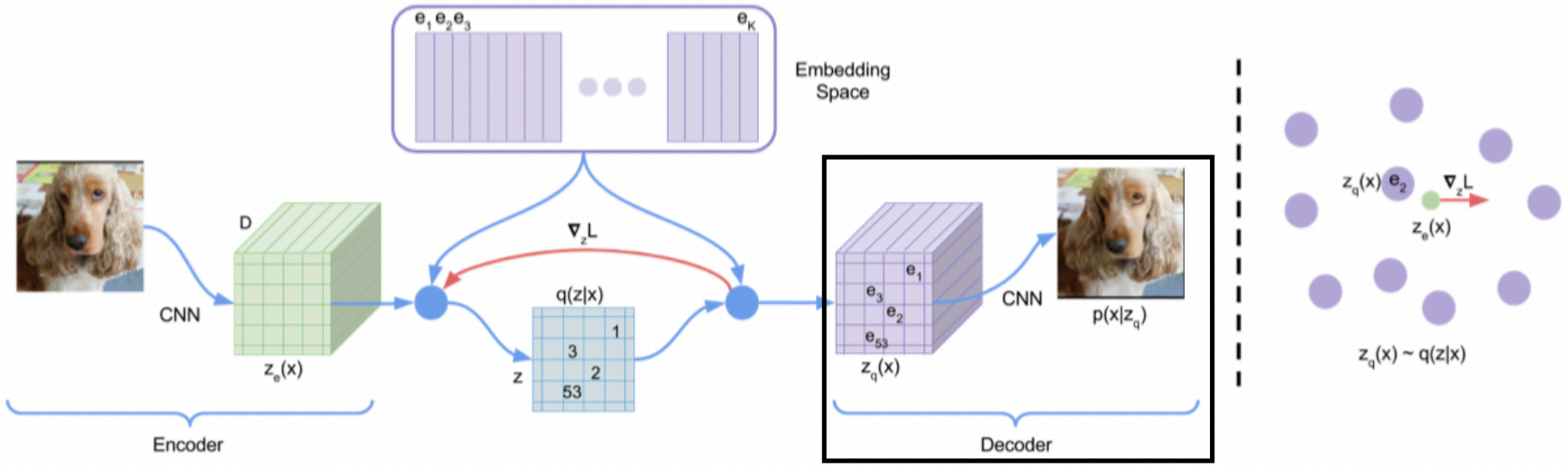
- Given an input image $x \in \mathbb{R}^{128\times128\times3}$, the encoder outputs $z_e(x) \in \mathbb{R}^{32\times32\times D}$.

Inference (2/3)



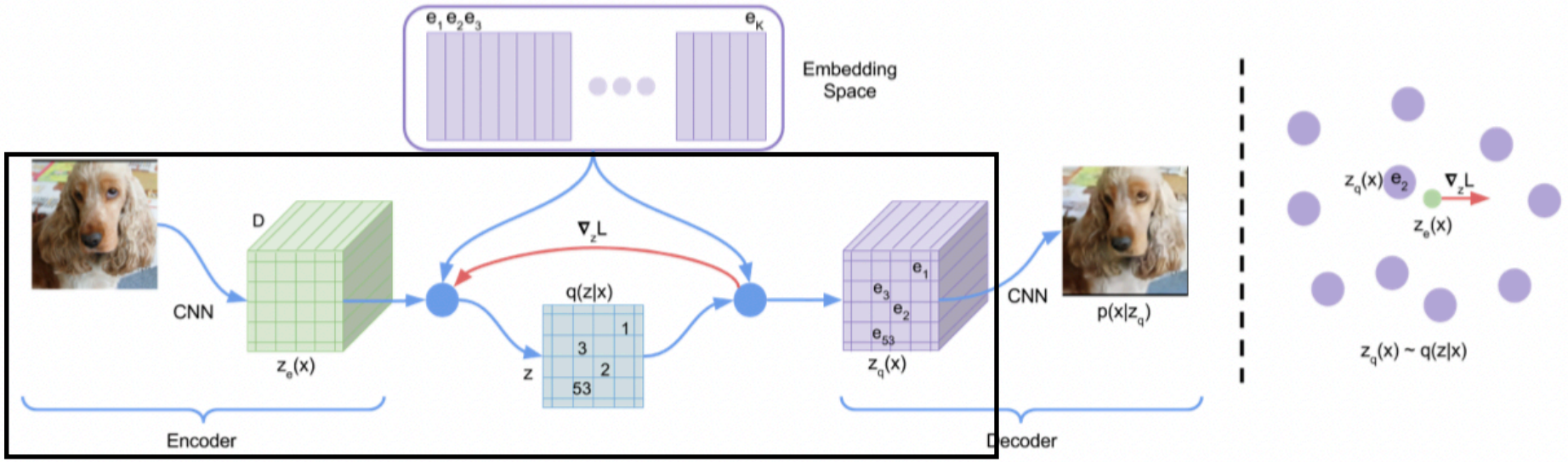
- Given an input image $x \in \mathbb{R}^{128\times128\times3}$, the encoder outputs $z_e(x) \in \mathbb{R}^{32\times32\times D}$.
- Each 32×32 embedding vector is mapped to its closest $e_k \in \mathbb{R}^D$ in the Embedding space.

Inference (3/3)



- Given an input image $x \in \mathbb{R}^{128\times128\times3}$, the encoder outputs $z_e(x) \in \mathbb{R}^{32\times32\times D}$.
- Each 32×32 embedding vector is mapped to its closest $e_k \in \mathbb{R}^D$ in the Embedding space.
- The mapped $z_q(x) \in \mathbb{R}^{128\times128\times D}$ is fed into the decoder to reconstruct the given input image.
- The bit reduction is $\frac{128 \times 128 \times 3 \times 8}{32 \times 32 \times 9} \approx 42.6$ where 8 comes from $256 = 2^8$ color resolutions and 9 comes from $K = 512 = 2^9$ embedding vectors.

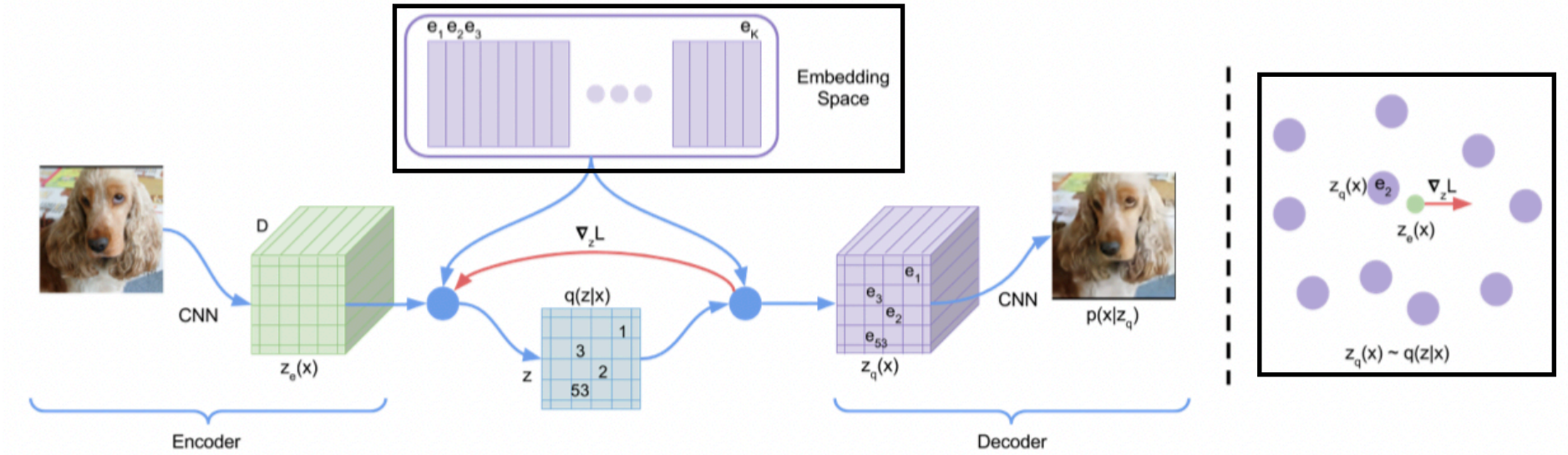
Train (1/3)



- The **straight-through estimator** that simply copies gradient from the decoder input $z_q(x)$ to the encoder output $z(e)$ is used to train the encoder part.

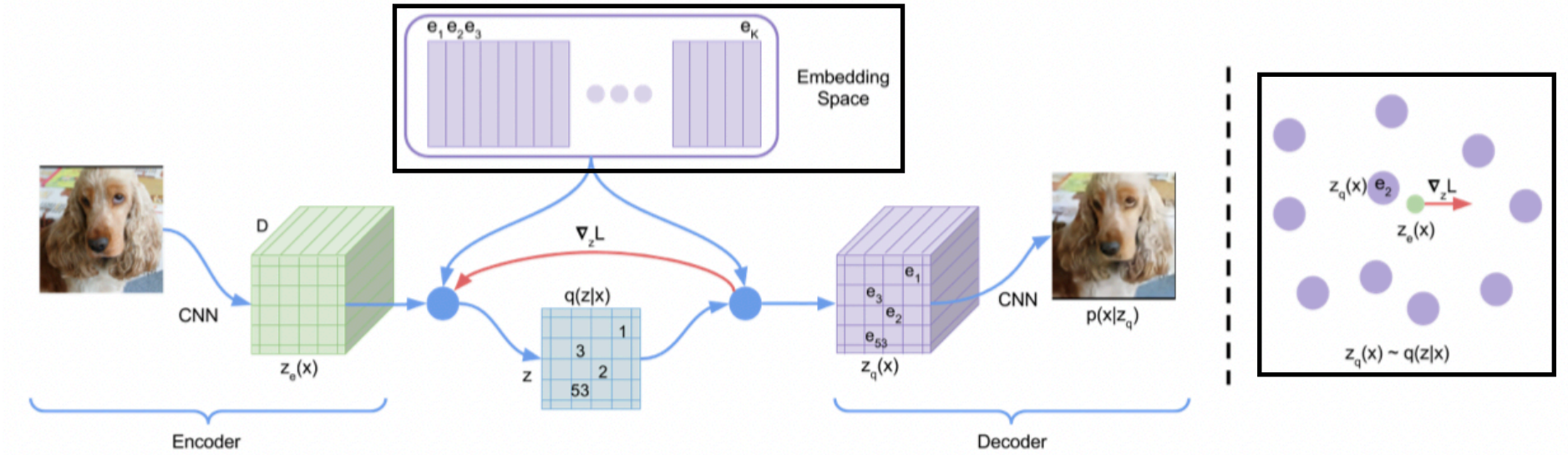
Train (2/3)

ROBOT INTELLIGENCE LAB



- The **straight-through estimator** that simply copies gradient from the decoder input $z_q(x)$ to the encoder output $z(e)$ is used to train the encoder part.
- To learn the embedding space, Vector Quantization (VQ) is used. The VQ objective uses the l_2 error to move the embedding vectors e_i towards the encoder outputs $z_e(x)$.

Train (3/3)



- The **straight-through estimator** that simply copies gradient from the decoder input $z_q(x)$ to the encoder output $z(e)$ is used to train the encoder part.
- To learn the embedding space, Vector Quantization (VQ) is used. The VQ objective uses the l_2 error to move the embedding vectors e_i towards the encoder outputs $z_e(x)$.
- After training, an autoregressive distribution over z, $p(z)$, is fitted so that we can generate z via ancestral sanplning.

Overall Loss

ROBOT INTELLIGENCE LAB

$$L = \log p(x \mid z_q(x)) + \|sg[z_e(x)] - e\|_2^2 + \beta\|z_e(x) - sg[e]\|_2^2$$

reconstruction

Vector Quantization

Results

ROBOT INTELLIGENCE LAB

Reconstruction

Figure 2: Left: ImageNet 128x128x3 images, right: reconstructions from a VQ-VAE with a 32x32x1 latent space, with K=512.

Results

ROBOT INTELLIGENCE LAB

Figure 3: Samples (128x128) from a VQ-VAE with a PixelCNN prior trained on ImageNet images. From left to right: kit fox, gray whale, brown bear, admiral (butterfly), coral reef, alp, microwave, pickup.

Results

ROBOT INTELLIGENCE LAB

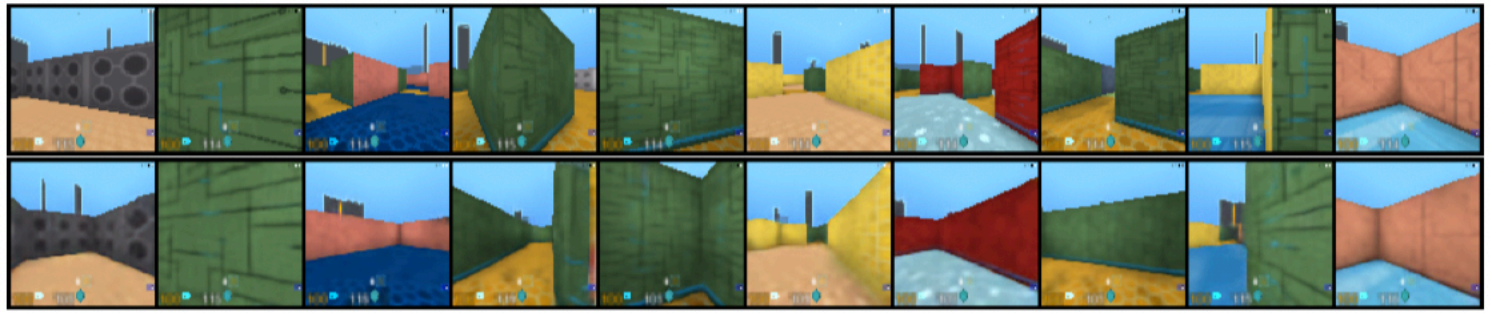


Figure 5: Top original images, Bottom: reconstructions from a 2 stage VQ-VAE, with 3 latents to model the whole image (27 bits), and as such the model cannot reconstruct the images perfectly. The reconstructions are generated by sampled from the second PixelCNN prior in the 21x21 latent domain of first VQ-VAE, and then decoded with standard VQ-VAE decoder to 84x84. A lot of the original scene, including textures, room layout and nearby walls remain, but the model does not try to store the pixel values themselves, which means the textures are generated procedurally by the PixelCNN.

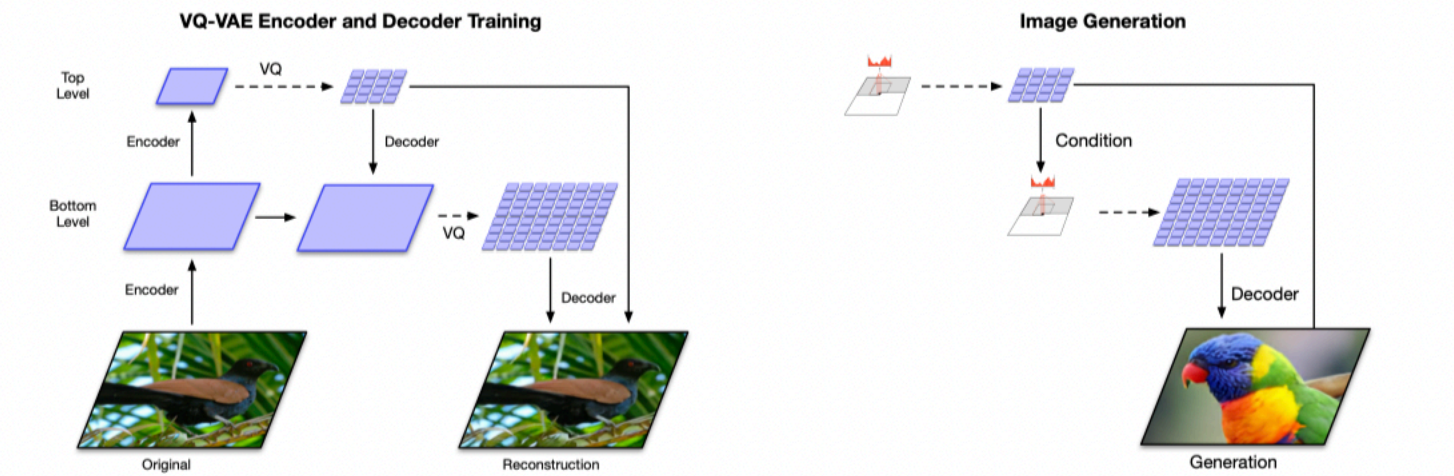
ROBOT INTELLIGENCE LAB

VQ-VAE2

"Generating Diverse High-Fidelity Images with VQ-VAE-2," 2019

Hierarchical Latent Codes

ROBOT INTELLIGENCE LAB



(a) Overview of the architecture of our hierarchical VQ-VAE. The encoders and decoders consist of deep neural networks. The input to the model is a 256×256 image that is compressed to quantized latent maps of size 64×64 and 32×32 for the *bottom* and *top* levels, respectively. The decoder reconstructs the image from the two latent maps.

(b) Multi-stage image generation. The top-level PixelCNN prior is conditioned on the class label, the bottom level PixelCNN is conditioned on the class label as well as the first level code. Thanks to the feed-forward decoder, the mapping between latents to pixels is fast. (The example image with a parrot is generated with this model).

Figure 2: VQ-VAE architecture.

Results

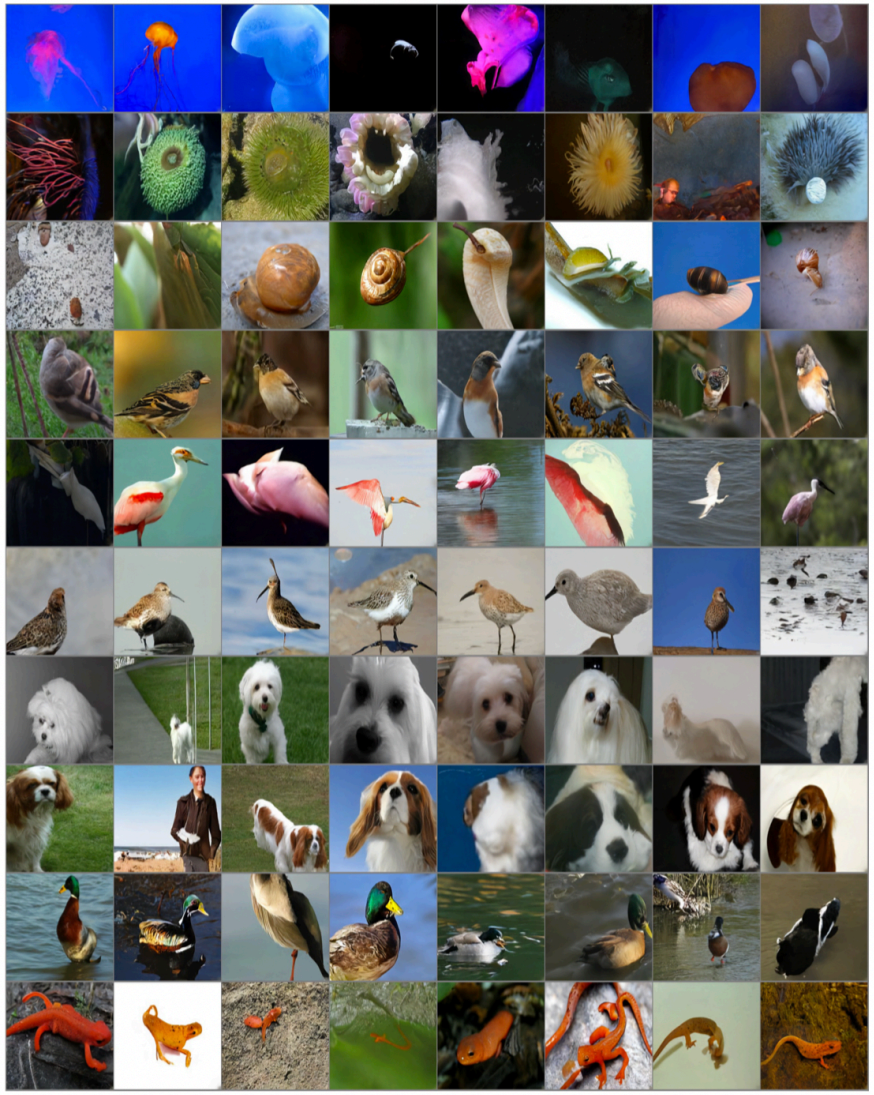


Figure 4: Class conditional random samples. Classes from the top row are: 108 sea anemone, 109 brain coral, 114 slug, 11 goldfinch, 130 flamingo, 141 redshank, 154 Pekinese, 157 papillon, 97 drake, and 28 spotted salamander.

ROBOT INTELLIGENCE LAB

Results

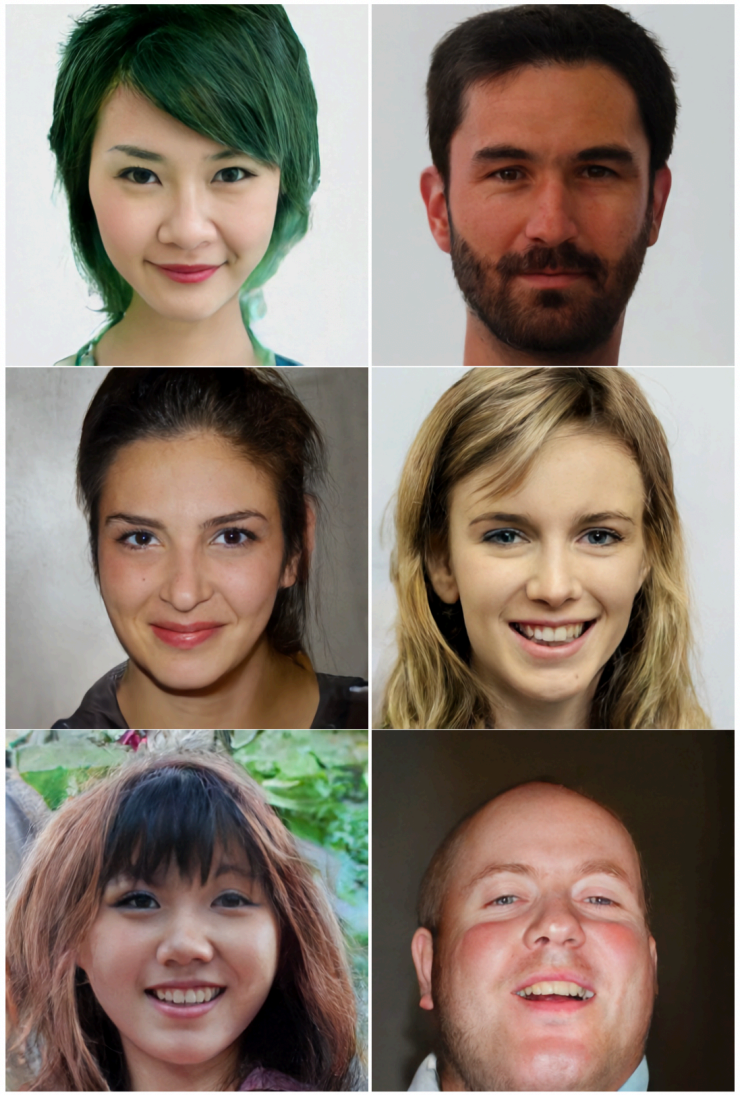


Figure 6: Representative samples from the three level hierarchical model trained on FFHQ-1024 × 1024. The model generates realistic looking faces that respect long-range dependencies such as matching eye colour or symmetric facial features, while covering lower density modes of the dataset (e.g., green hair). Please refer to the supplementary material for more samples, including full resolution samples.

ROBOT INTELLIGENCE LAB

Thank You