Deep Generative Models

2022
Multicampus

Il Gu Yi
ModuLabs, Research Scientist
Soochul Park
Vivestudios, Research Scientist

v.1.2

Disclaimer

- Details of convolutional neural networks
  - convolution, pooling operator, …
- Details of recurrent neural networks
  - LSTM, GRU, …
- Various regularization
  - Dropout, Batch normalization, …
- Backpropagation

Contents

- Introduction to Generative Models
- Review of Probability Theory
- Autoregressive Models
- Latent Variable Models
- Flow based Models
- Generative Adversarial Networks

References

- Deep Learning Book, Ian Goodfellow, et. al., 2015
- Stanford cs236 fall18: Deep Generative Models
- A lot of papers and codes

Taxonomy of Generative Models

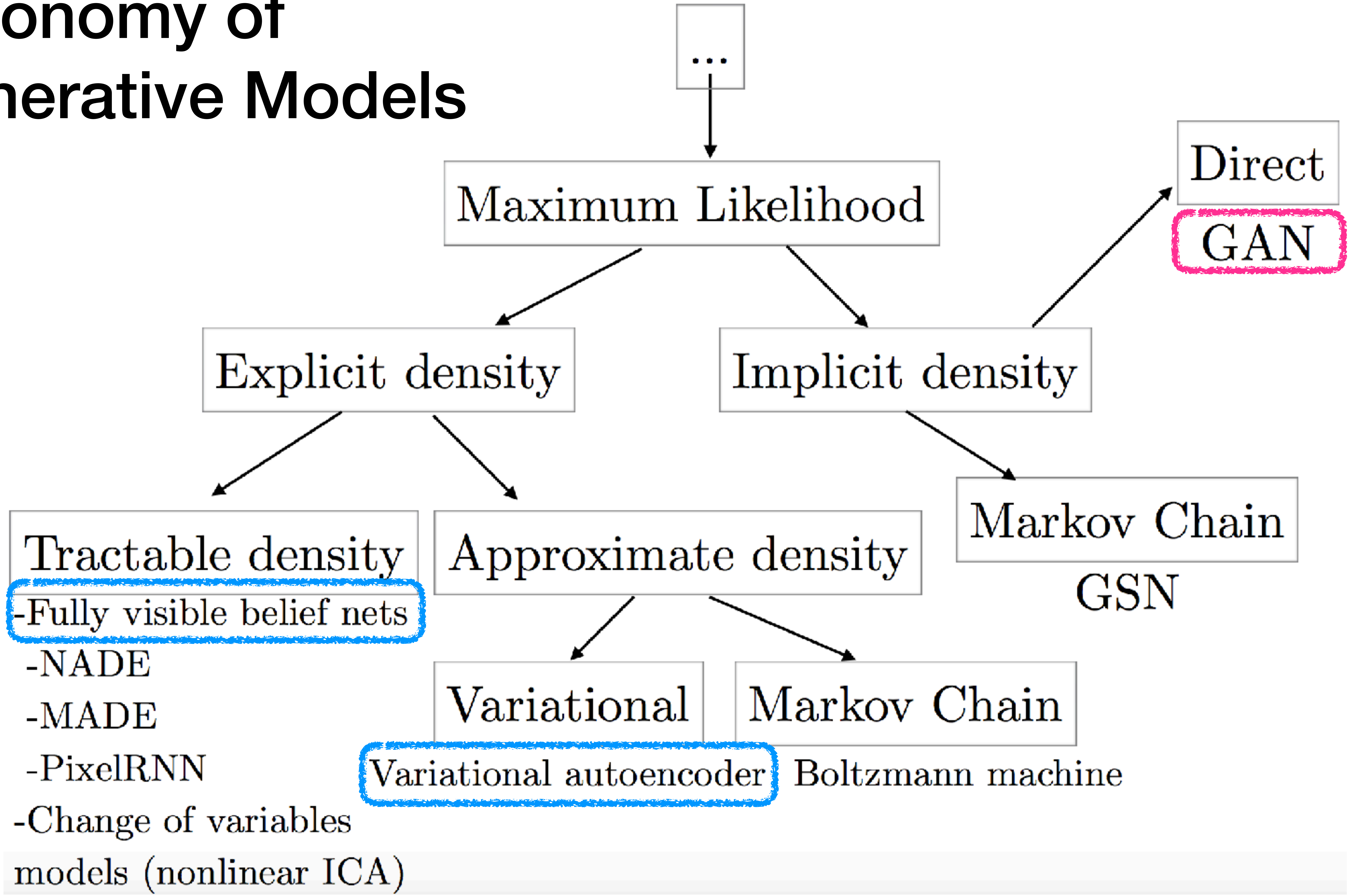


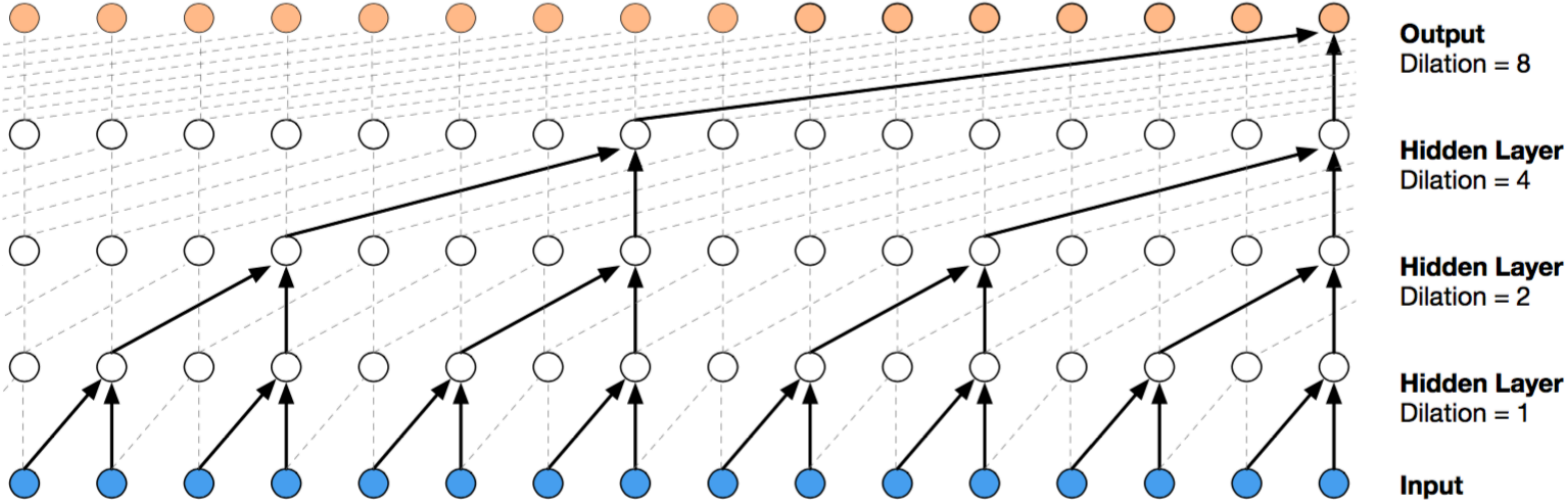
Figure credit: Ian Goodfellow, NIPS 2016 Tutorial: Generative Adversarial Networks

AutoRegressive Models

PixelCNN

Figure credit: Aäron van den Oord, et al., Conditional Image Generation with PixelCNN Decoders

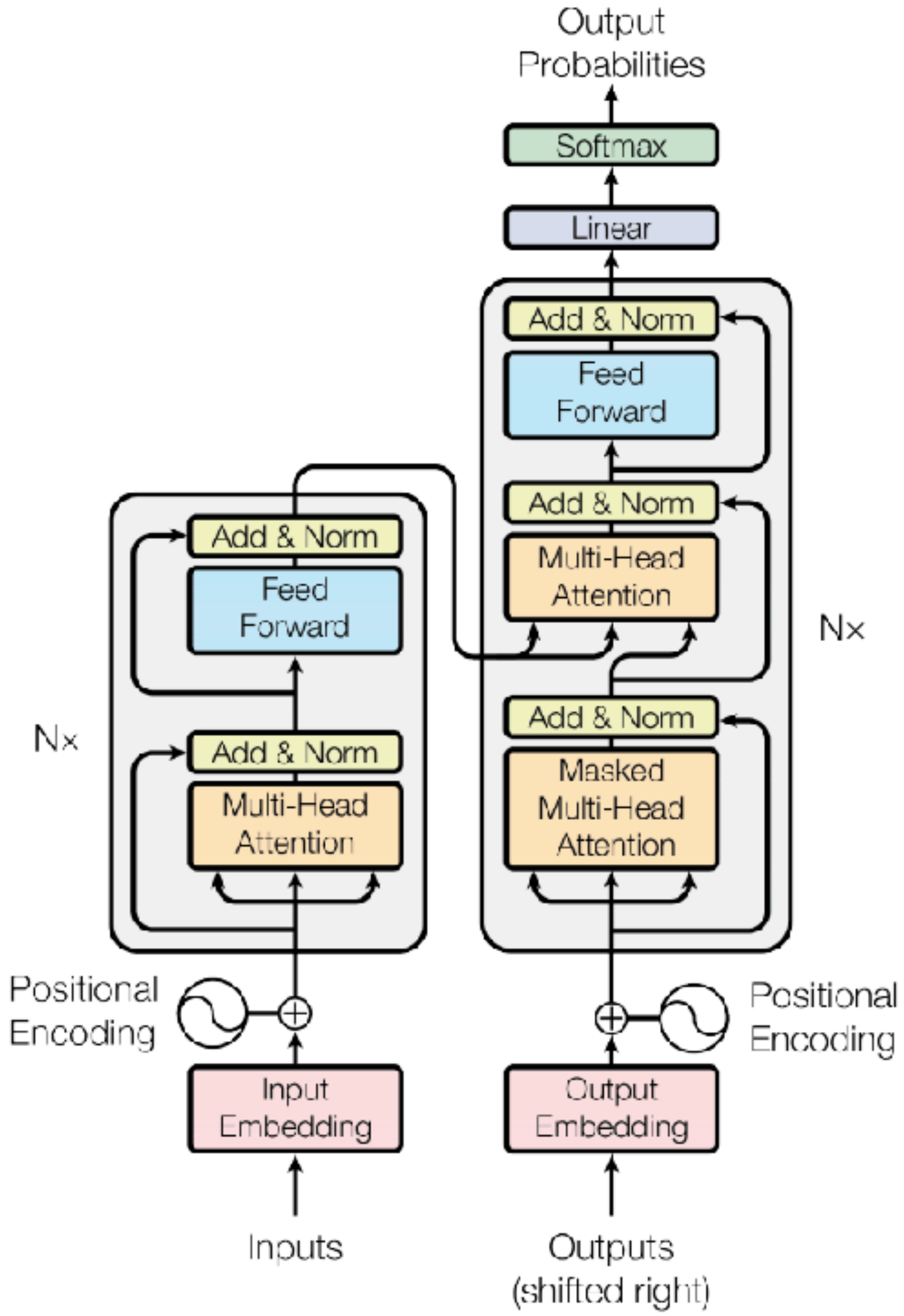
WaveNet



Audio samples
https://google.github.io/tacotron/publications/tacotron2/index.html

Figure credit: Aäron van den Oord, et al., Wavenet: A Generative Model for Raw Audio

Transformer



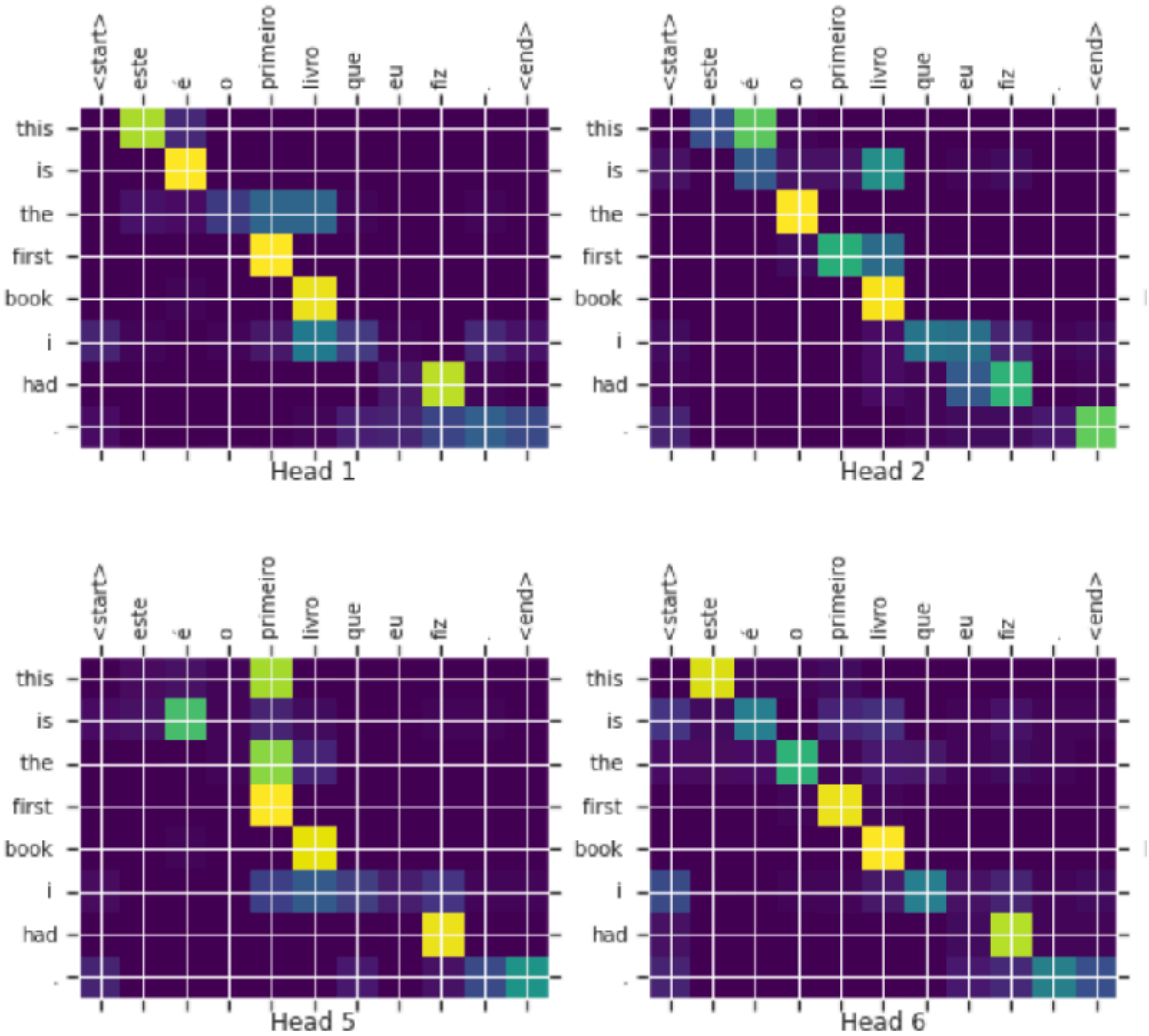
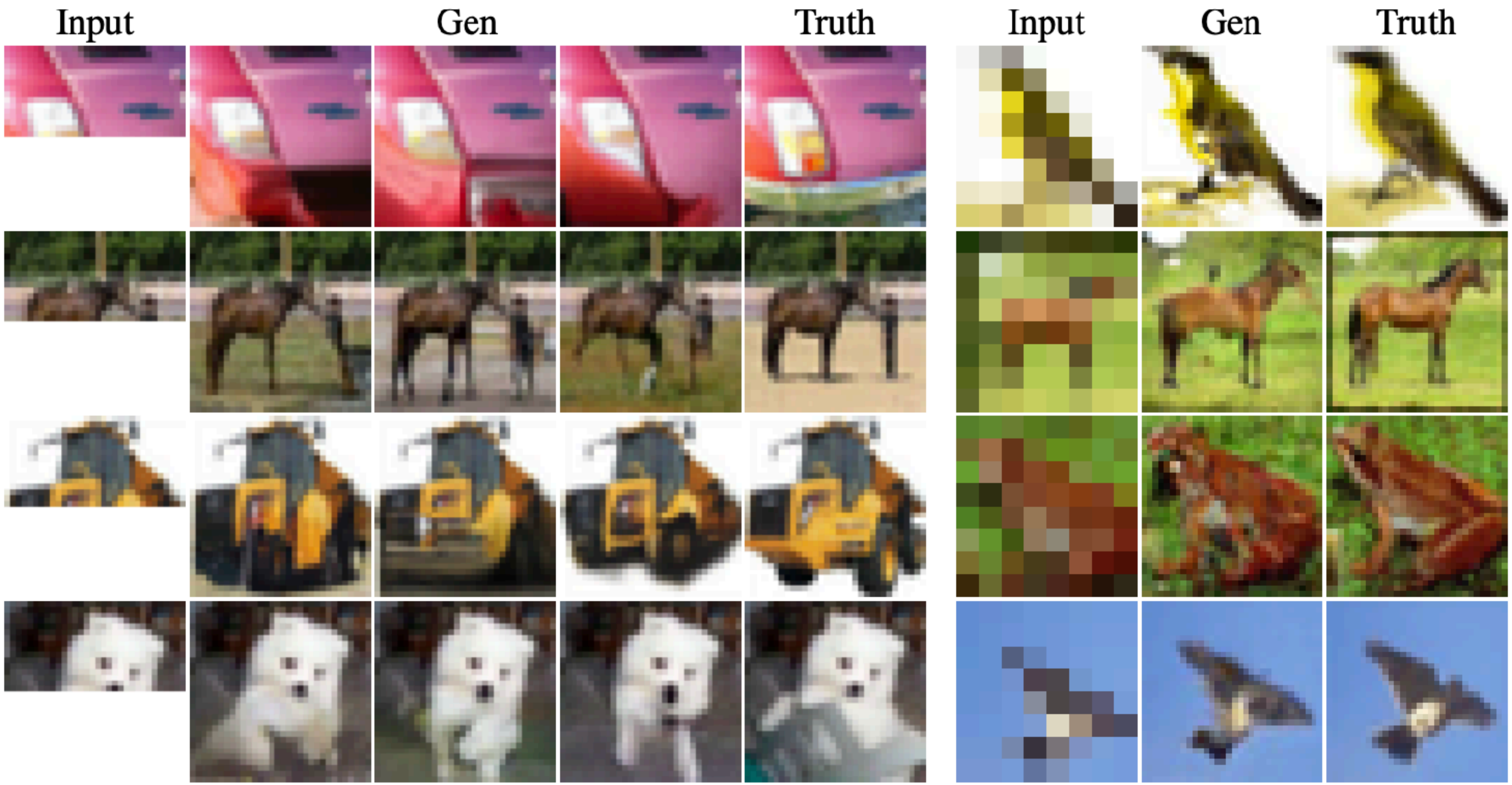


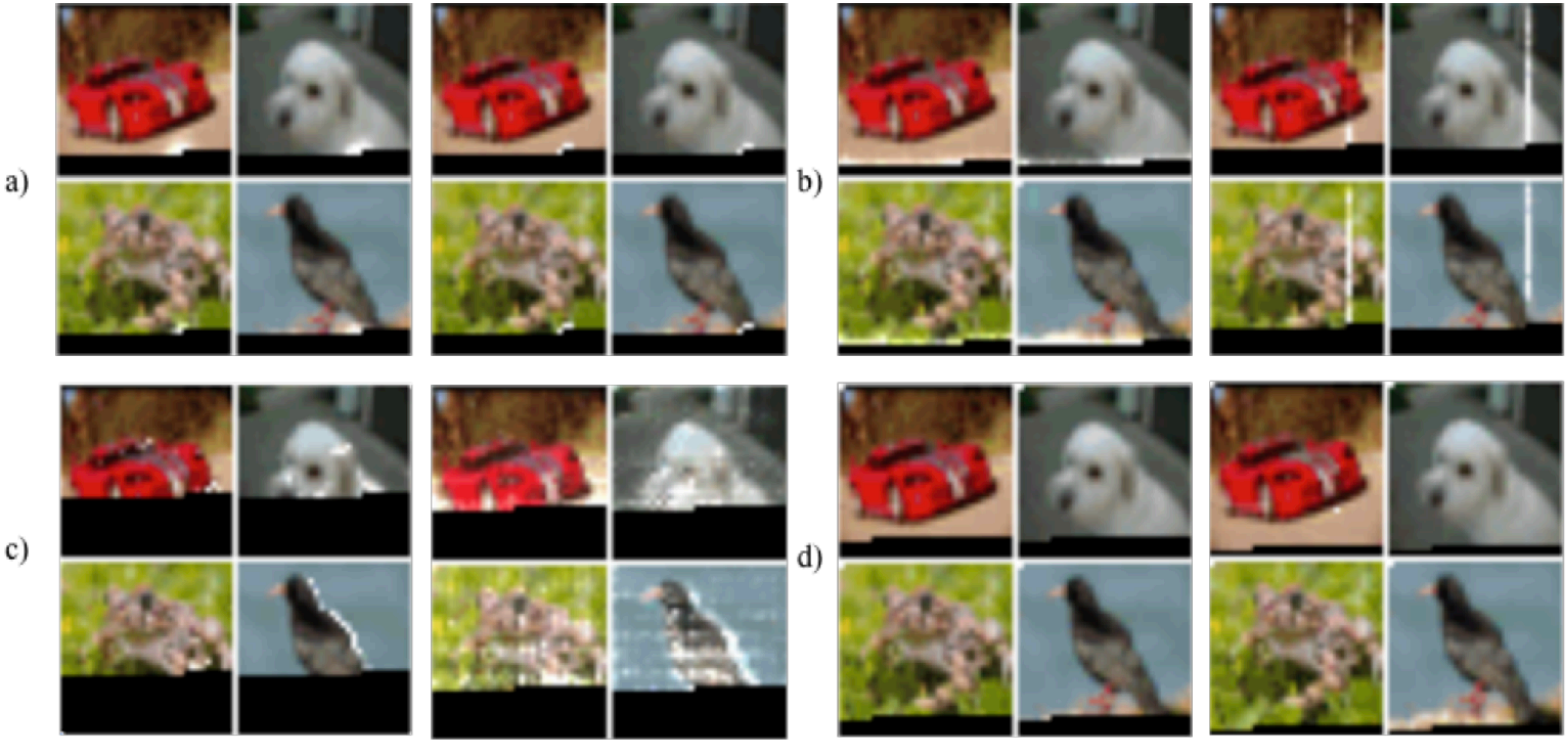
Figure credit :
Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.
https://www.tensorflow.org/text/tutorials/transformer

Image Transformer



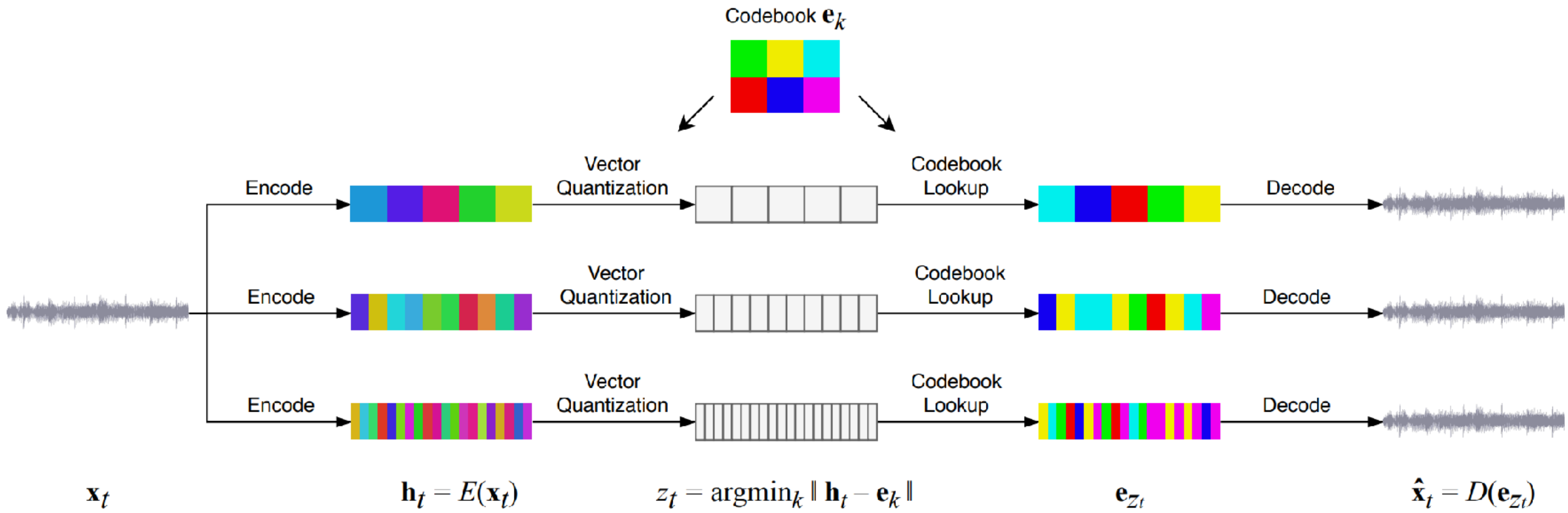
Parmar, Niki, et al. "Image transformer." International Conference on Machine Learning. PMLR, 2018.

Sparse Transformer



Child, Rewon, et al. "Generating long sequences with sparse transformers." arXiv preprint arXiv:1904.10509 (2019).

Jukebox

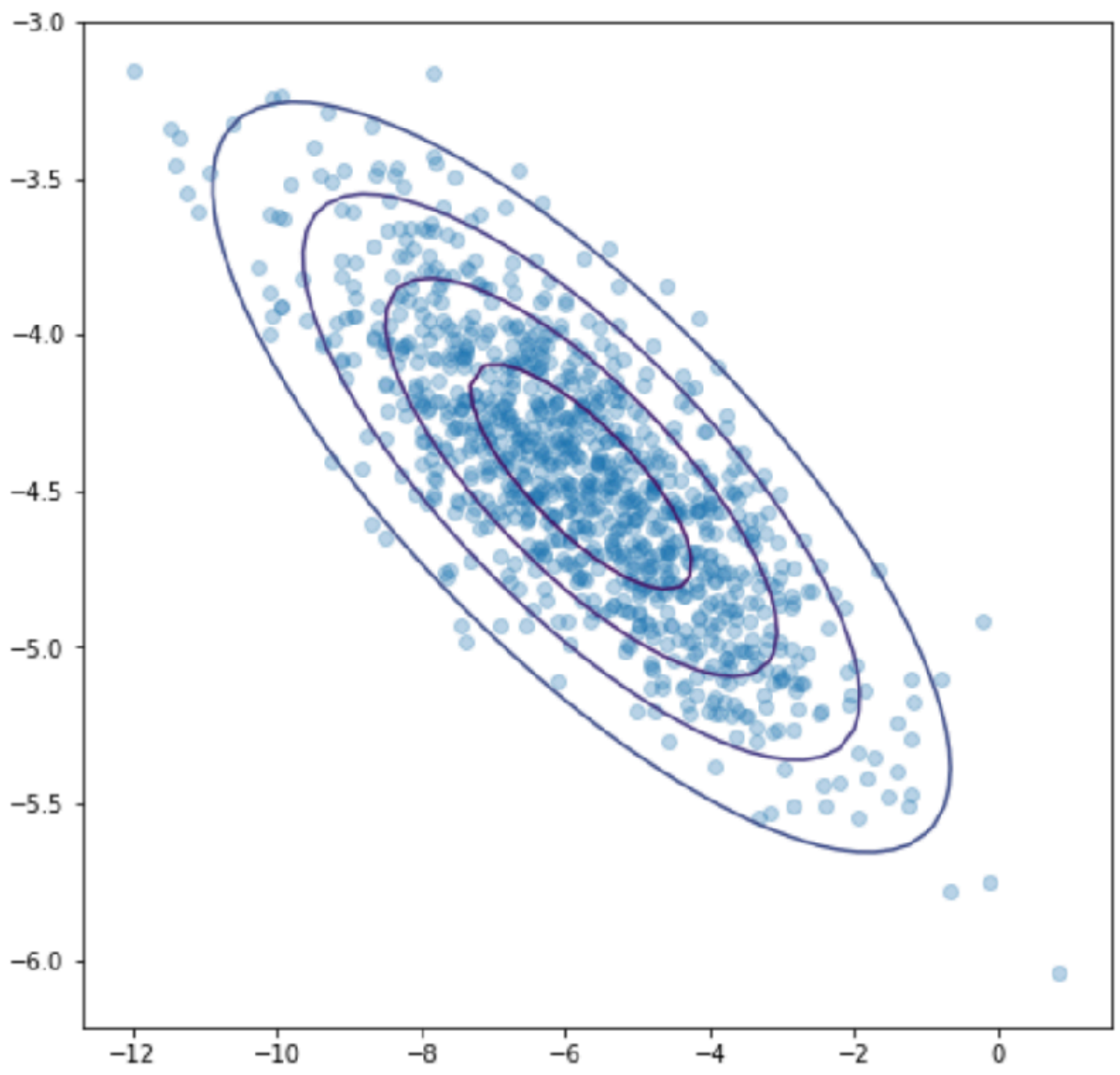


$$p(\mathbf{z}) = p(\mathbf{z}^{\text{top}}, \mathbf{z}^{\text{middle}}, \mathbf{z}^{\text{bottom}})$$
$$= p(\mathbf{z}^{\text{top}})p(\mathbf{z}^{\text{middle}}|\mathbf{z}^{\text{top}})p(\mathbf{z}^{\text{bottom}}|\mathbf{z}^{\text{middle}}, \mathbf{z}^{\text{top}})$$

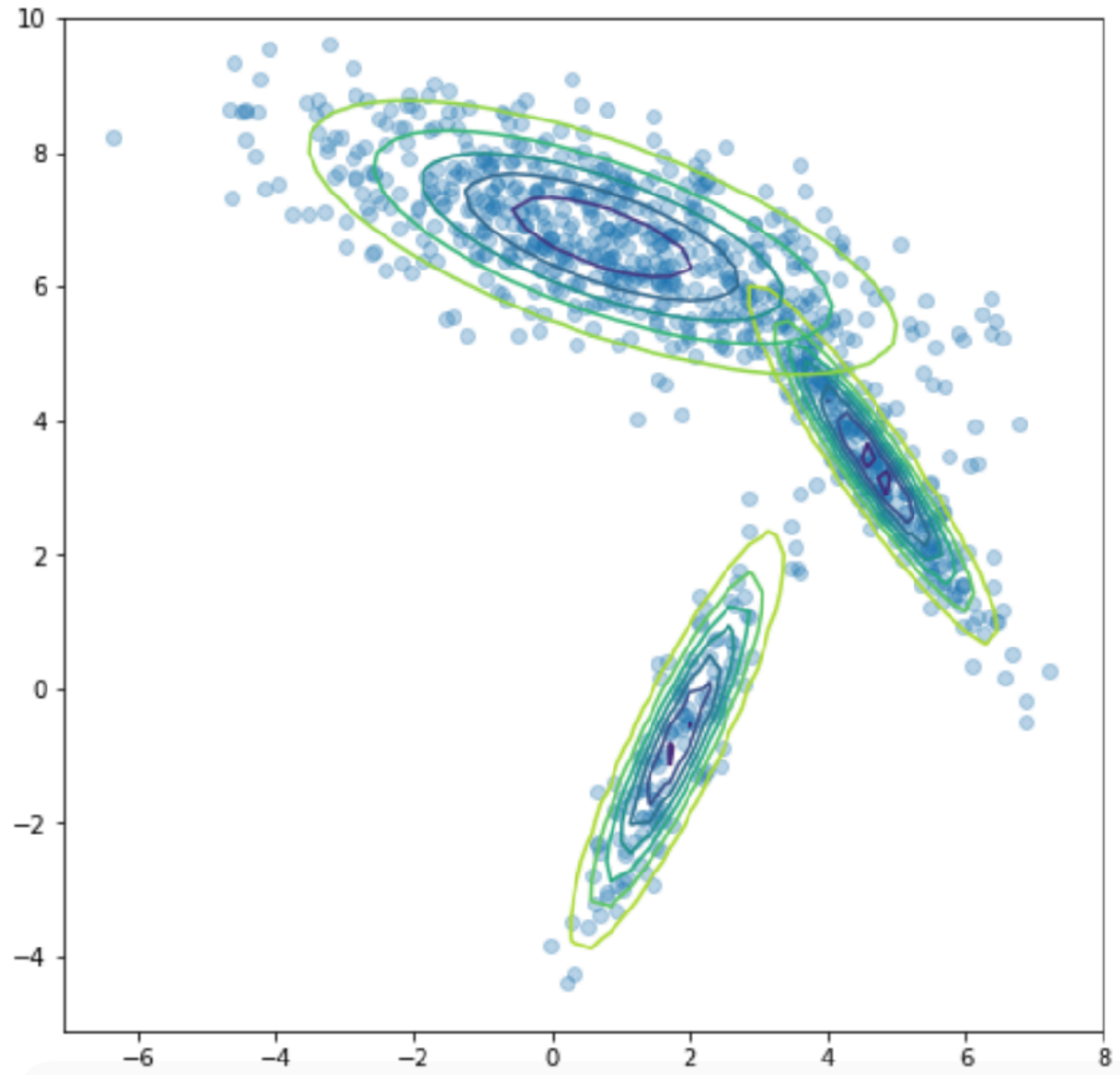
Dhariwal, Prafulla, et al. "Jukebox: A generative model for music." *arXiv preprint arXiv:2005.00341* (2020).

Latent Models

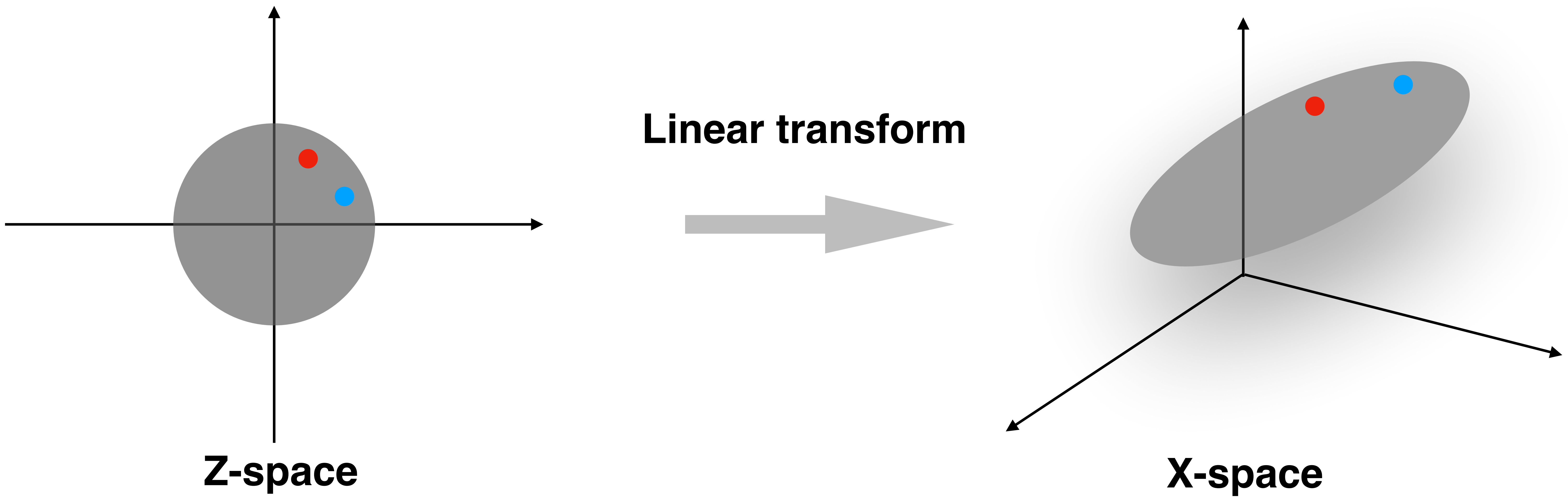
Gaussian Model



Gaussian Mixture Models



Probabilistic PCA

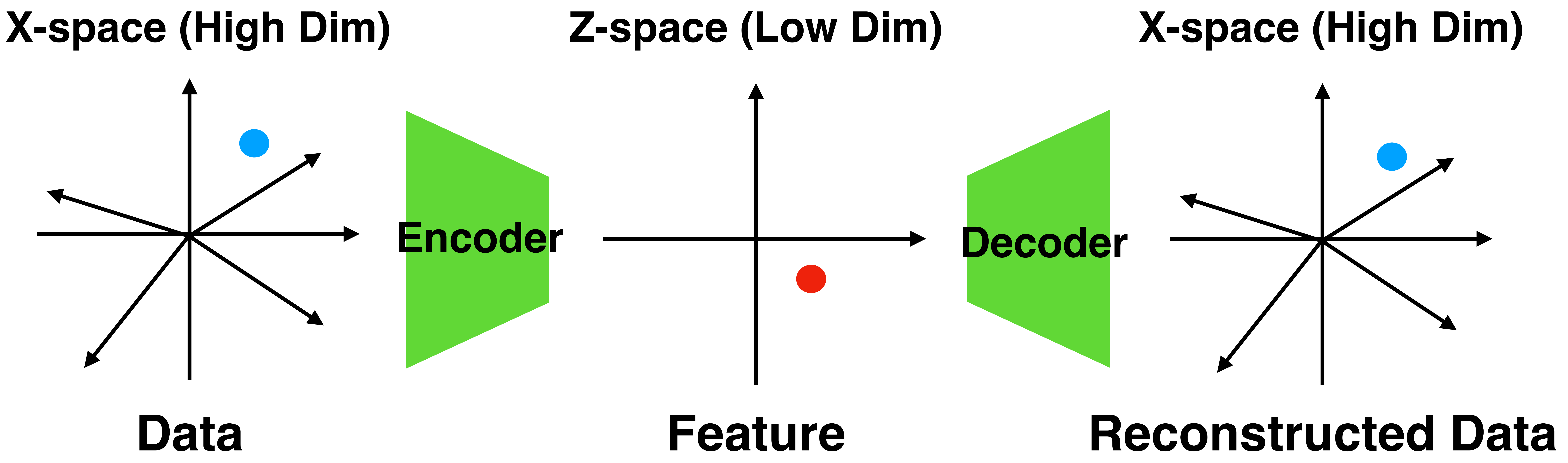


Prior	$p_\theta(z) = N(z \mid 0, I)$
Conditional	$p_\theta(x \mid z) = N(x \mid Wz + \mu, \sigma^2 I)$

(Linear-Gaussian Model)

References :
Michael E. Tipping and Christopher M. Bishop. Probabilistic Principal Component Analysis. 1999.
Christopher M. Bishop. Pattern Recognition and Machine Learning. p.570-580

Auto-Encoder



Variational Auto-Encoder

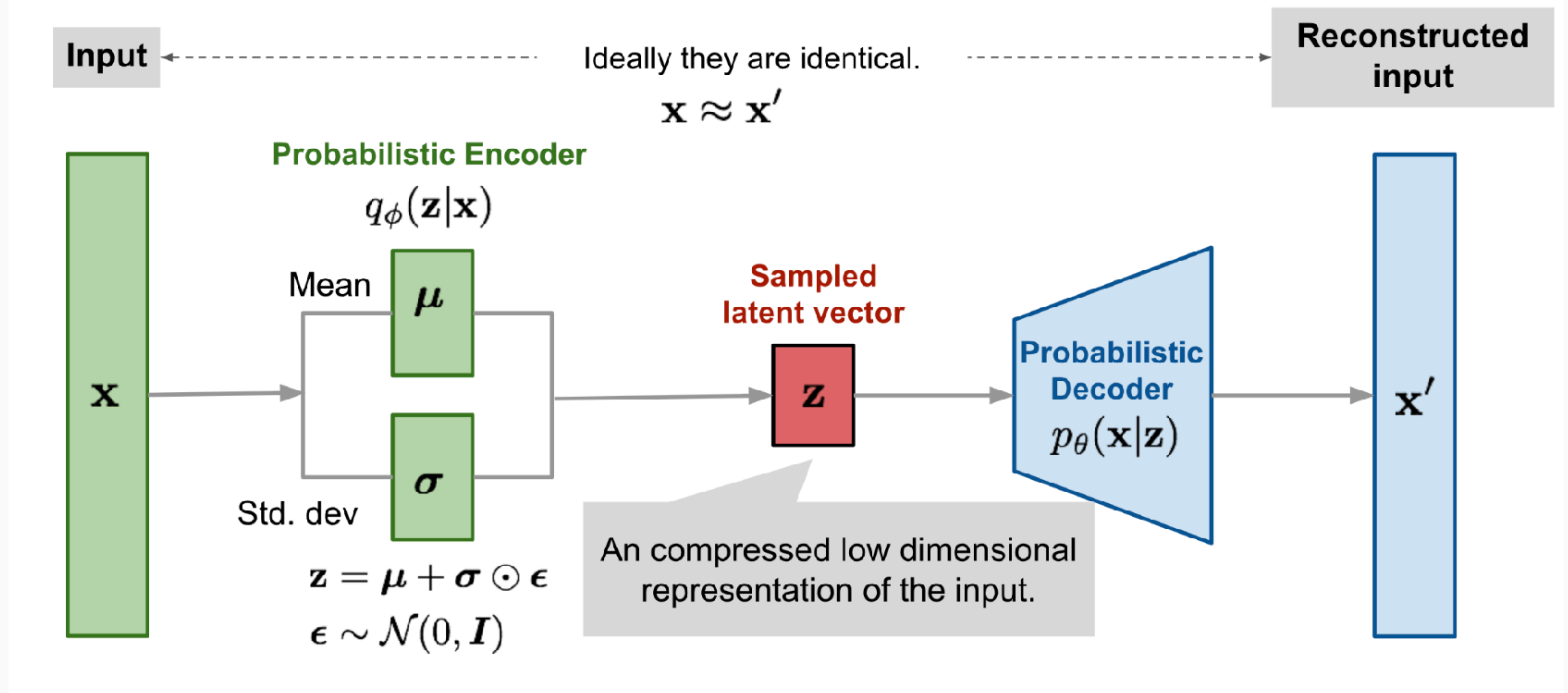
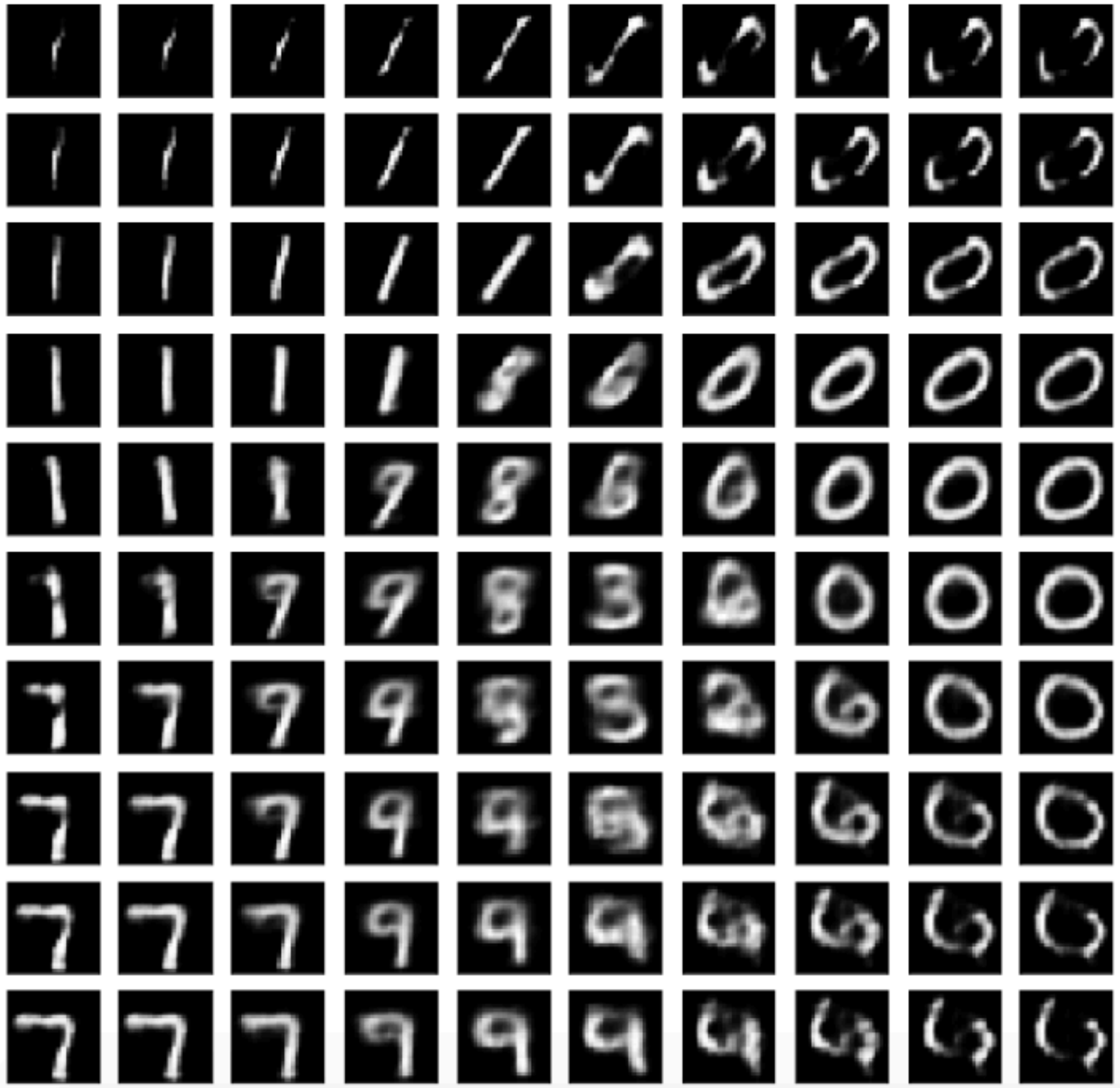
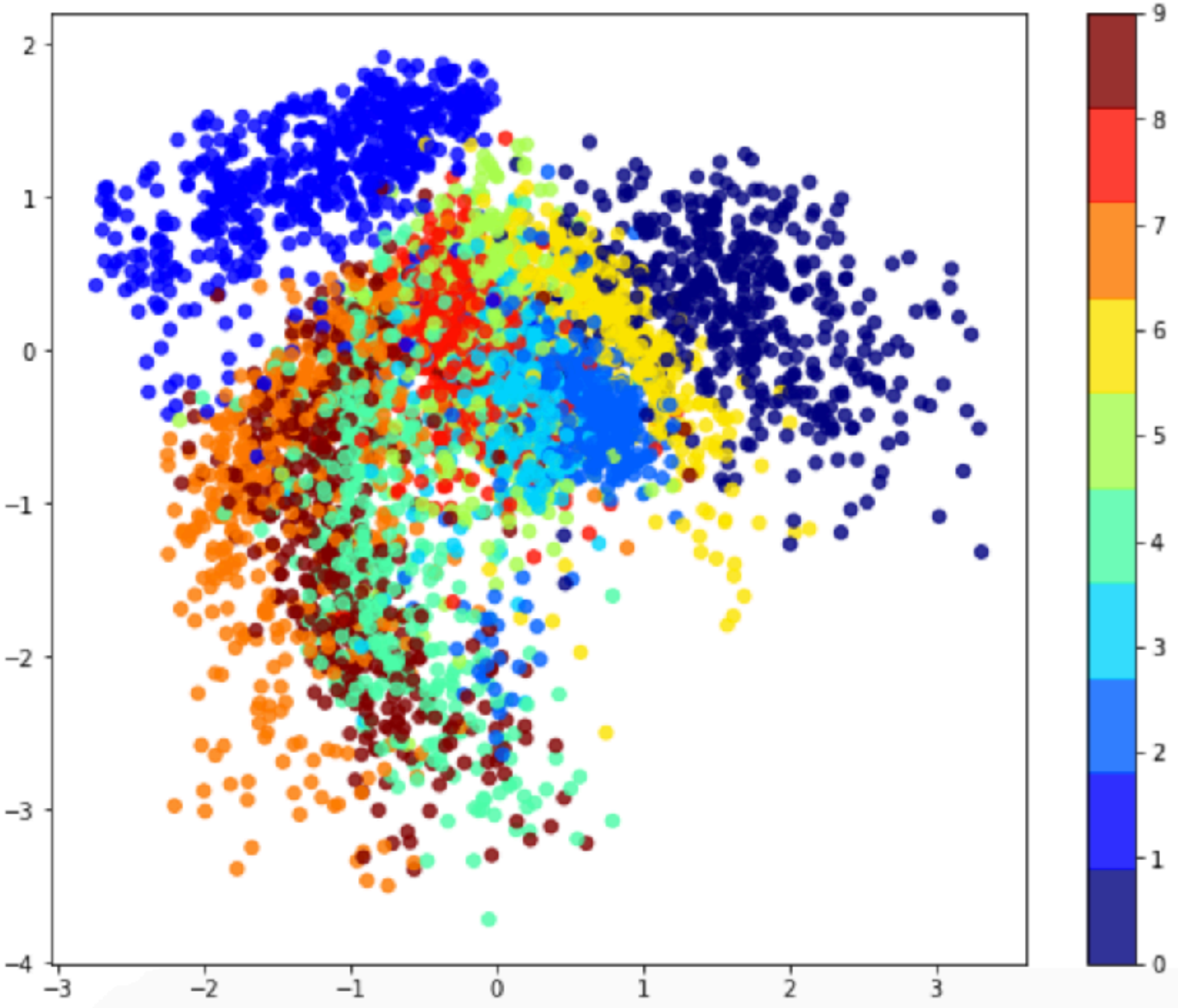


Fig. 9. Illustration of variational autoencoder model with the multivariate Gaussian assumption.

https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html

Variational Auto-Encoders

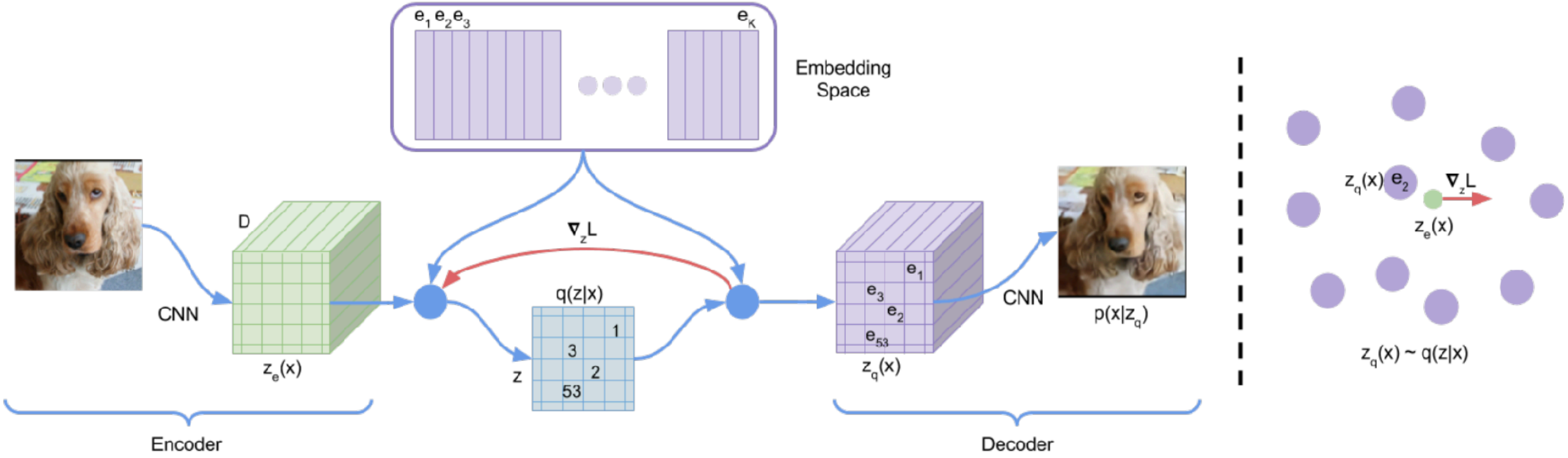




Hierachical VAE

Child, Rewon. "Very deep vaes generalize autoregressive models and can outperform them on images." arXiv preprint arXiv:2011.10650 (2020).

VQ-VAE

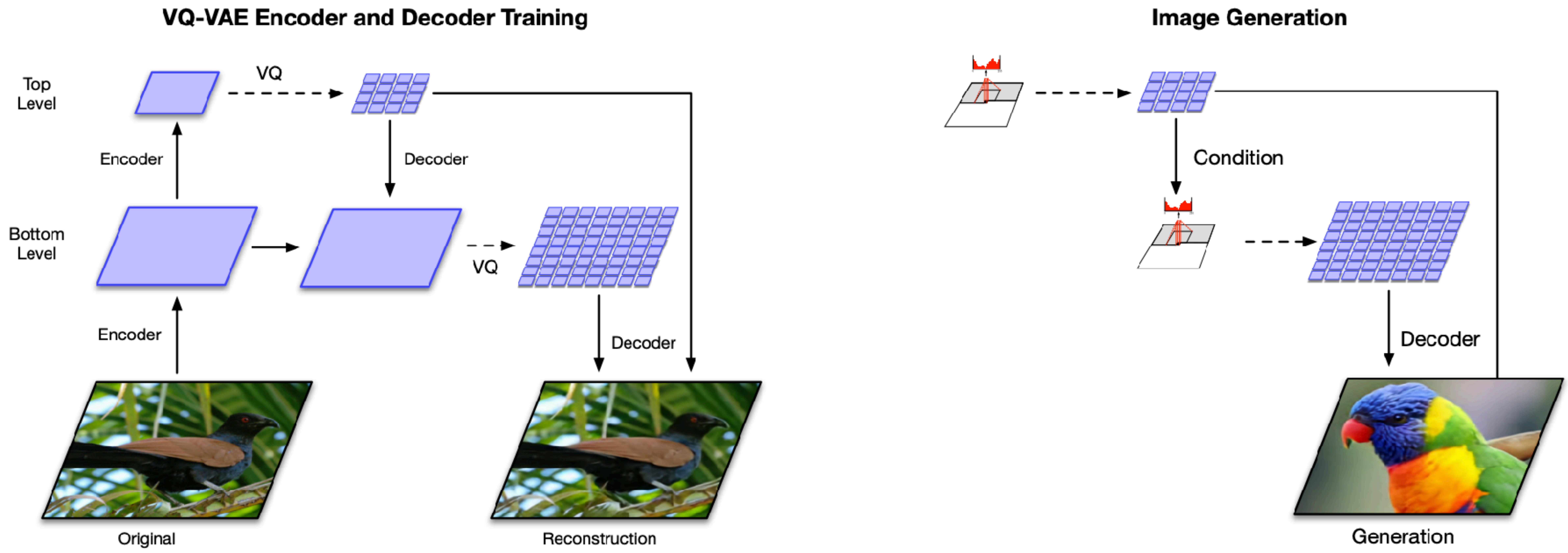


$$\mathcal{L}(\mathbf{x}, D(\mathbf{e})) = ||\mathbf{x} - D(\mathbf{e})||_2^2 + ||sg[E(\mathbf{x})] - \mathbf{e}||_2^2 + \beta||sg[\mathbf{e}] - E(\mathbf{x})||_2^2$$

Reconstruction loss **Codebook loss** **Commitment loss**

Oord, Aaron van den, Oriol Vinyals, and Koray Kavukcuoglu. "Neural discrete representation learning." arXiv preprint arXiv:1711.00937 (2017).

VQ-VAE 2



Razavi, Ali, Aaron van den Oord, and Oriol Vinyals. "Generating diverse high-fidelity images with vq-vae-2." Advances in neural information processing systems. 2019.

DALL·E

(a) a tapir made of accordion. a tapir with the texture of an accordion.

(b) an illustration of a baby hedgehog in a christmas sweater walking a dog

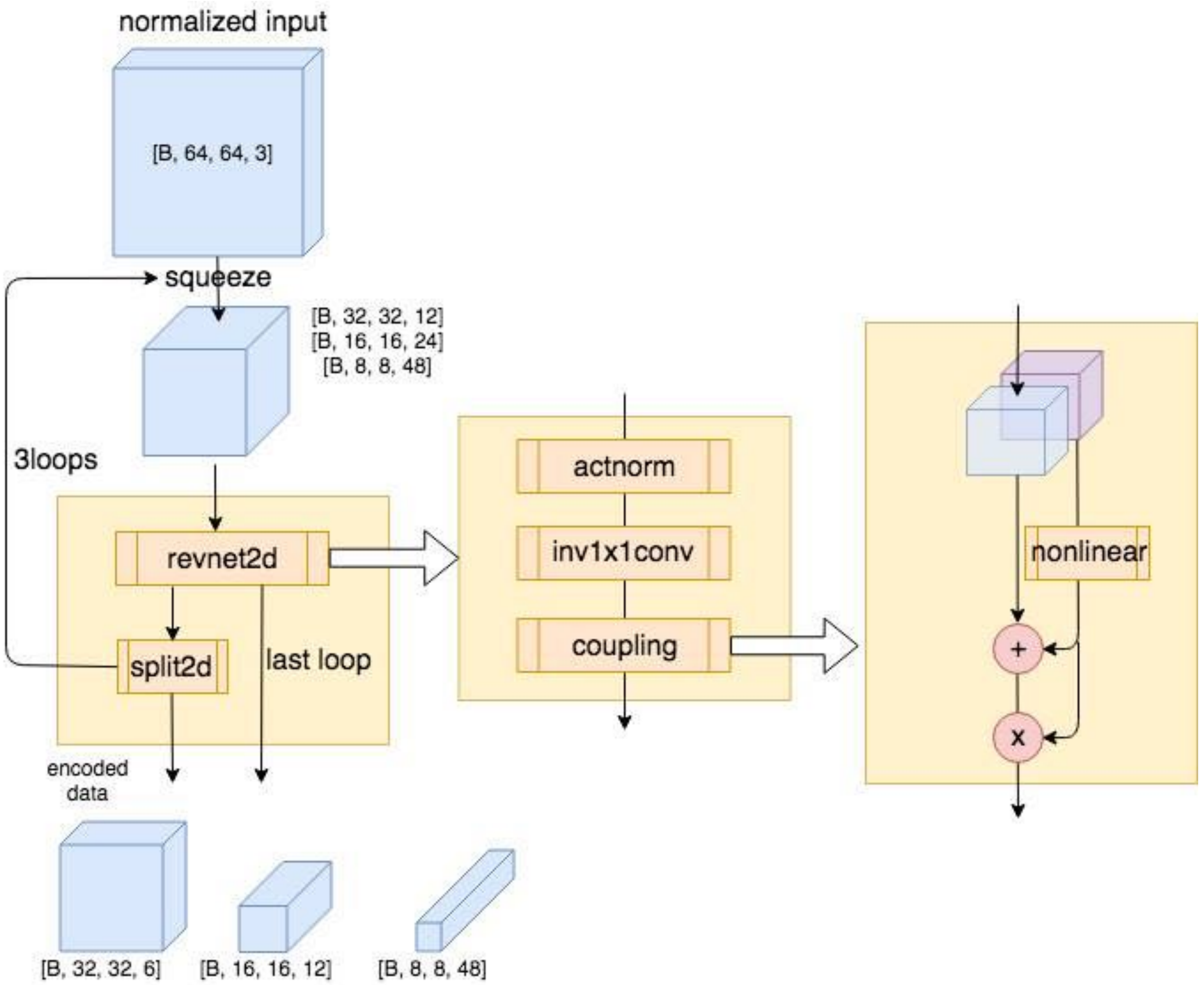
(c) a neon sign that reads "backprop". a neon sign that reads "backprop". backprop neon sign

(d) the exact same cat on the top as a sketch on the bottom

Ramesh, Aditya, et al. "Zero-shot text-to-image generation." arXiv preprint arXiv:2102.12092 (2021).

Flow Models

GLOW networks



Results via NICE

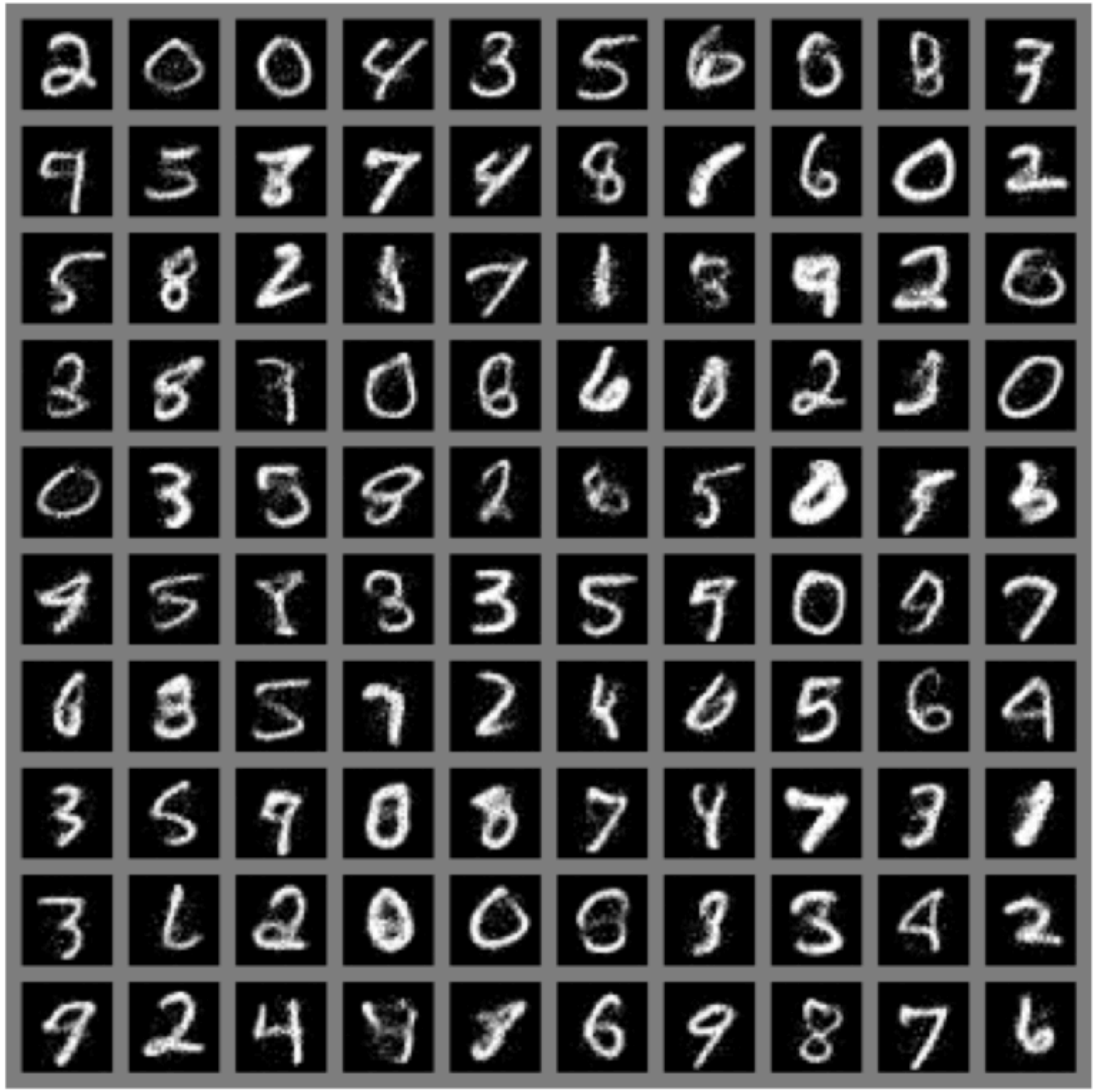
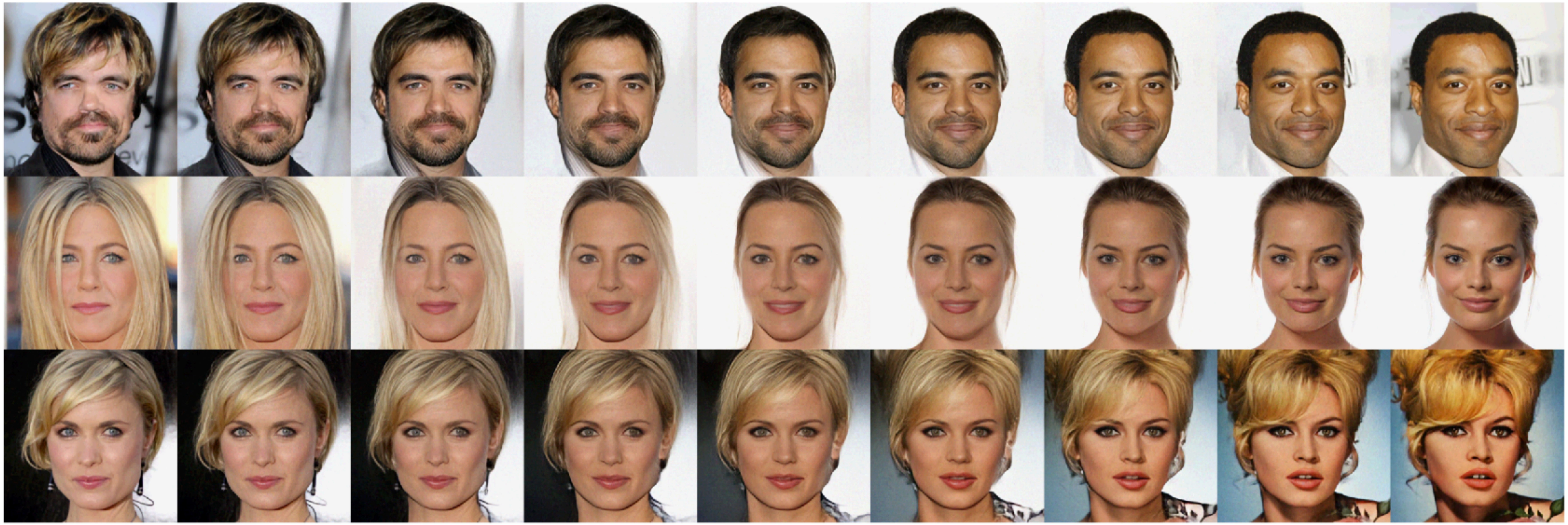
 

Figure credit: L. Dinh, et. al., NICE: Non-Linear Independent Components Estimation

Interpolations via RealNVP

Figure credit: L. Dinh, et. al., Density estimation using Real NVP

Interpolations via Glow



Demo: https://openai.com/blog/glow/
Figure credit: D. Kingma, et. al., Glow: Generative Flow with Invertible 1x1 Convolutions

BeautyGLOW

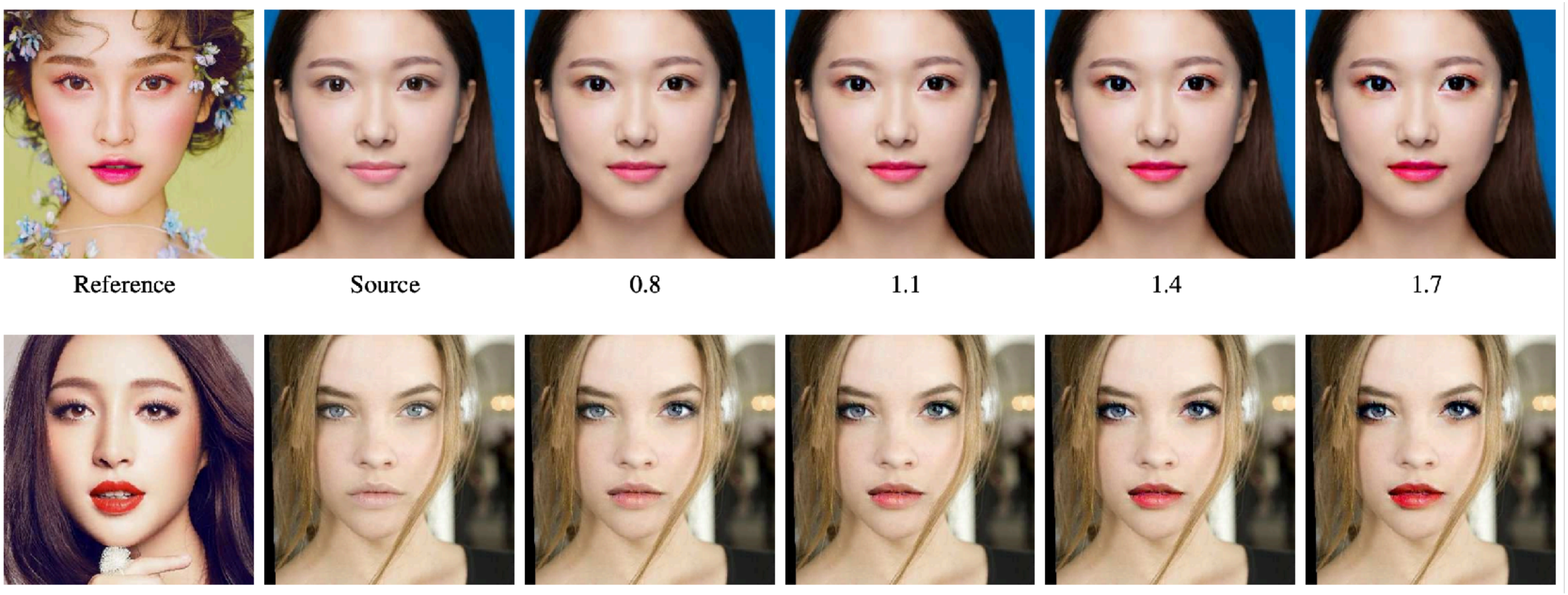


Figure credit: Hung-Jen Chen et. al., BeautyGlow: On-Demand Makeup Transfer Framework with Reversible Generative Network

Generative Adversarial Networks

Generative Adversarial Networks Framework

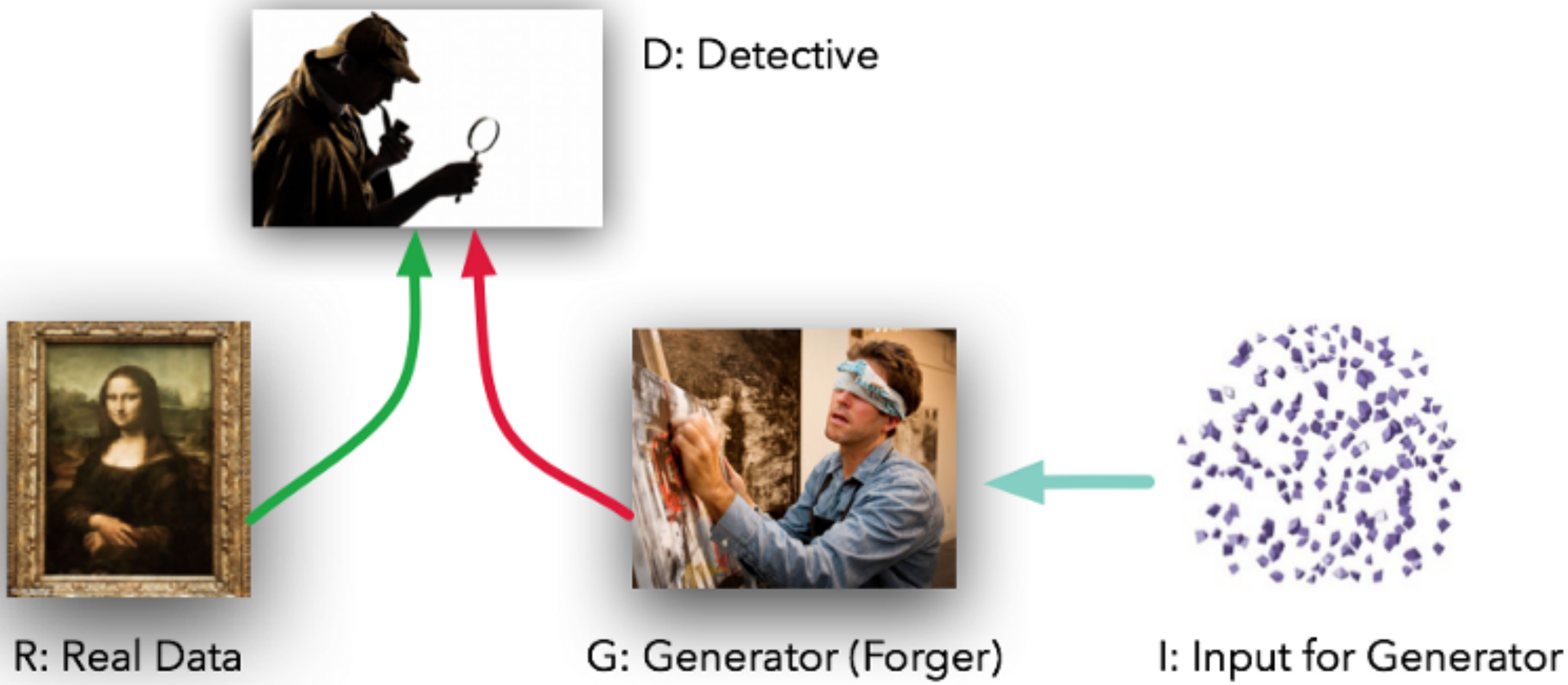
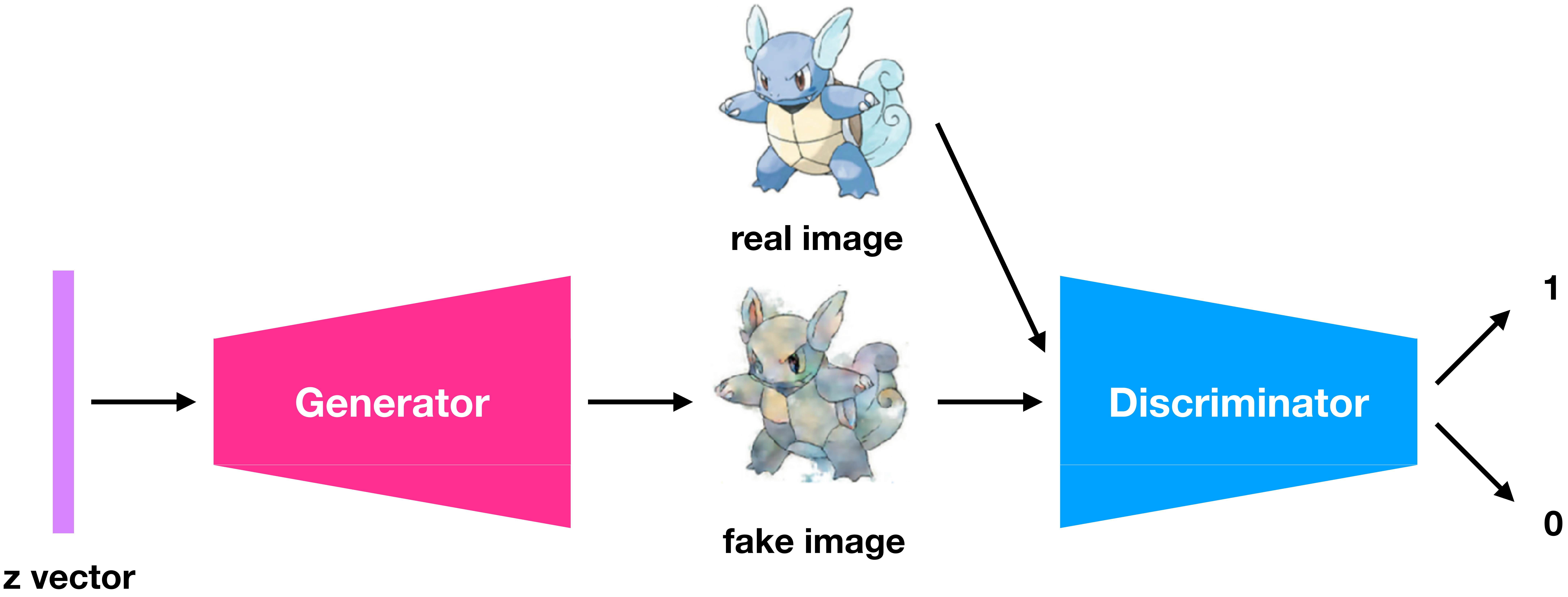


Figure credit: Blog. Generative Adversarial Networks (GANs) in 50 lines of code (PyTorch)

Generative Adversarial Networks Framework



DCGAN Architectures

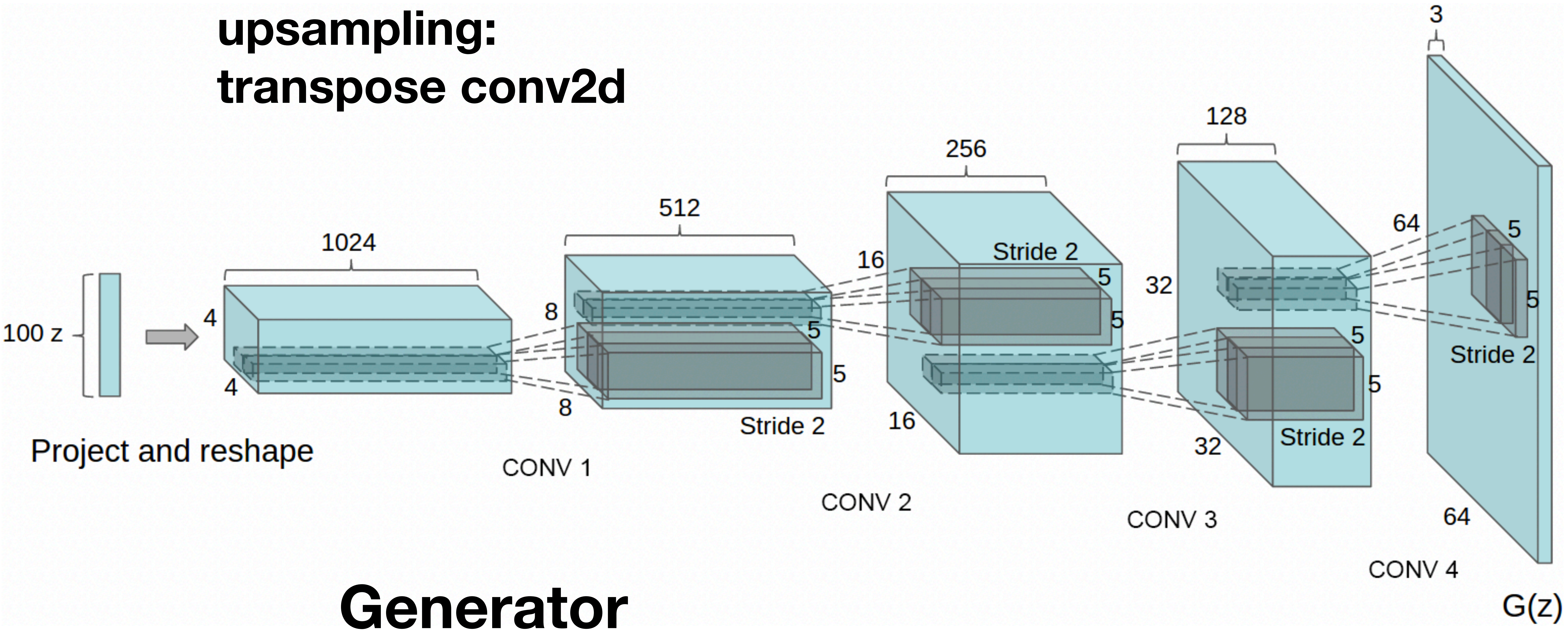


Figure credit: A. Radford, et al., Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks

DCGAN Results

Figure credit: A. Radford, et al., Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks

Conditional GAN Results

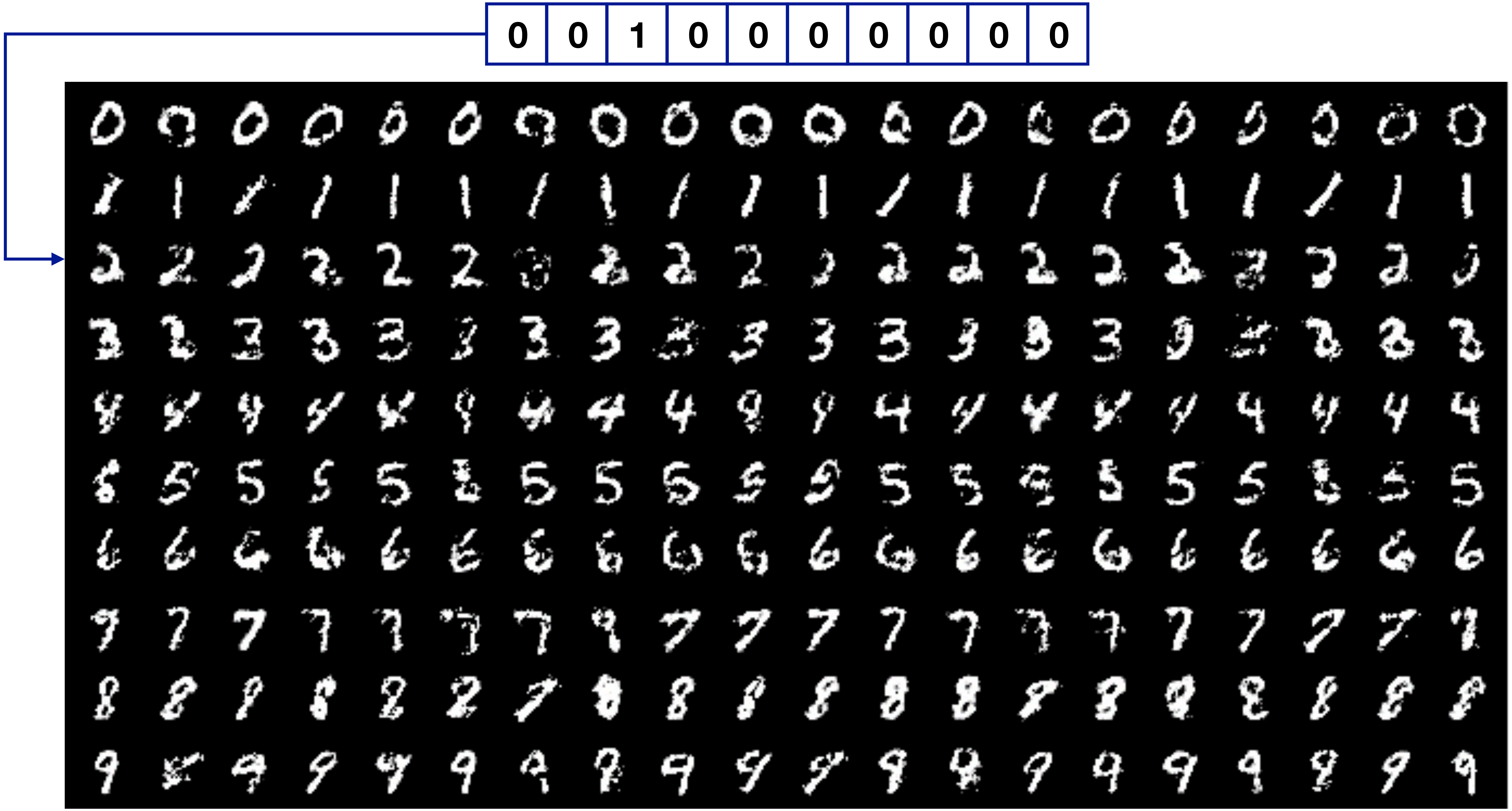
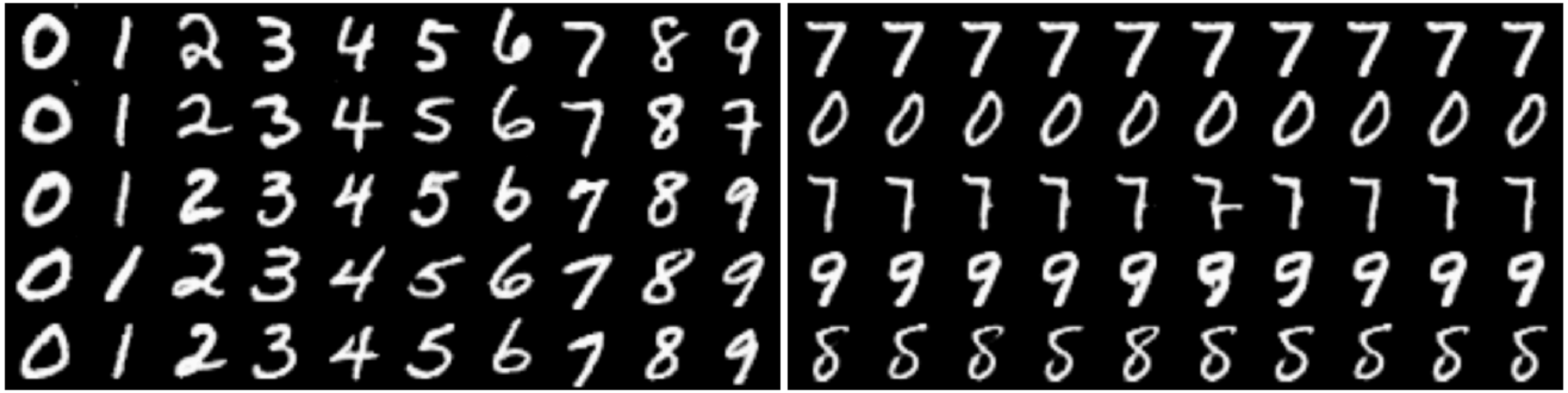
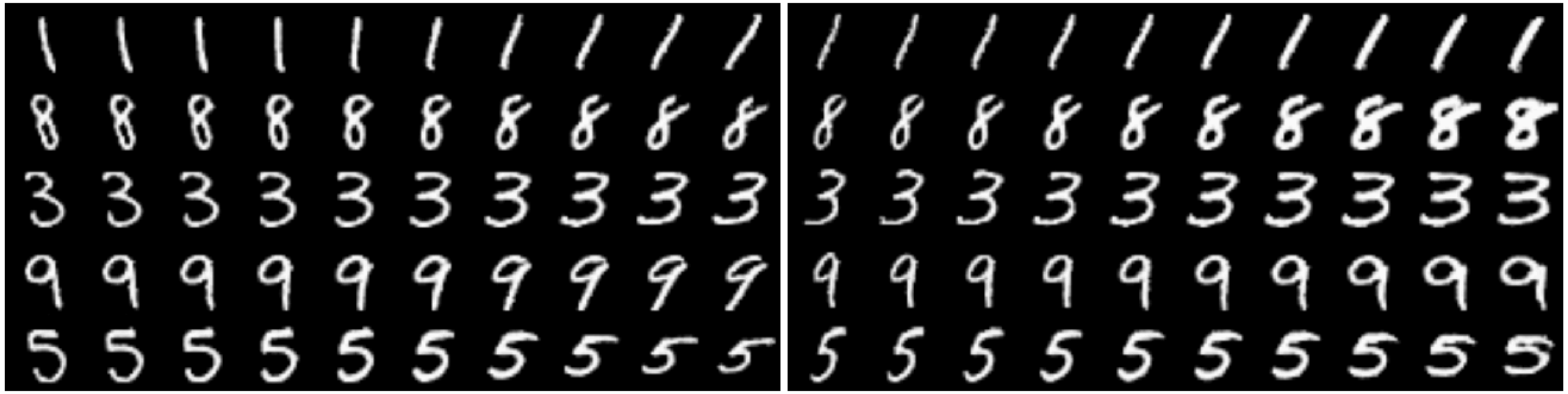


Figure credit: M. Mirza, et al., Conditional Generative Adversarial Nets

InfoGAN Results



(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)

Figure credit: X. Chen, et al., InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets

InfoGAN Results

(a) Azimuth (pose) (b) Elevation

(c) Lighting (d) Wide or Narrow

BiGAN Results

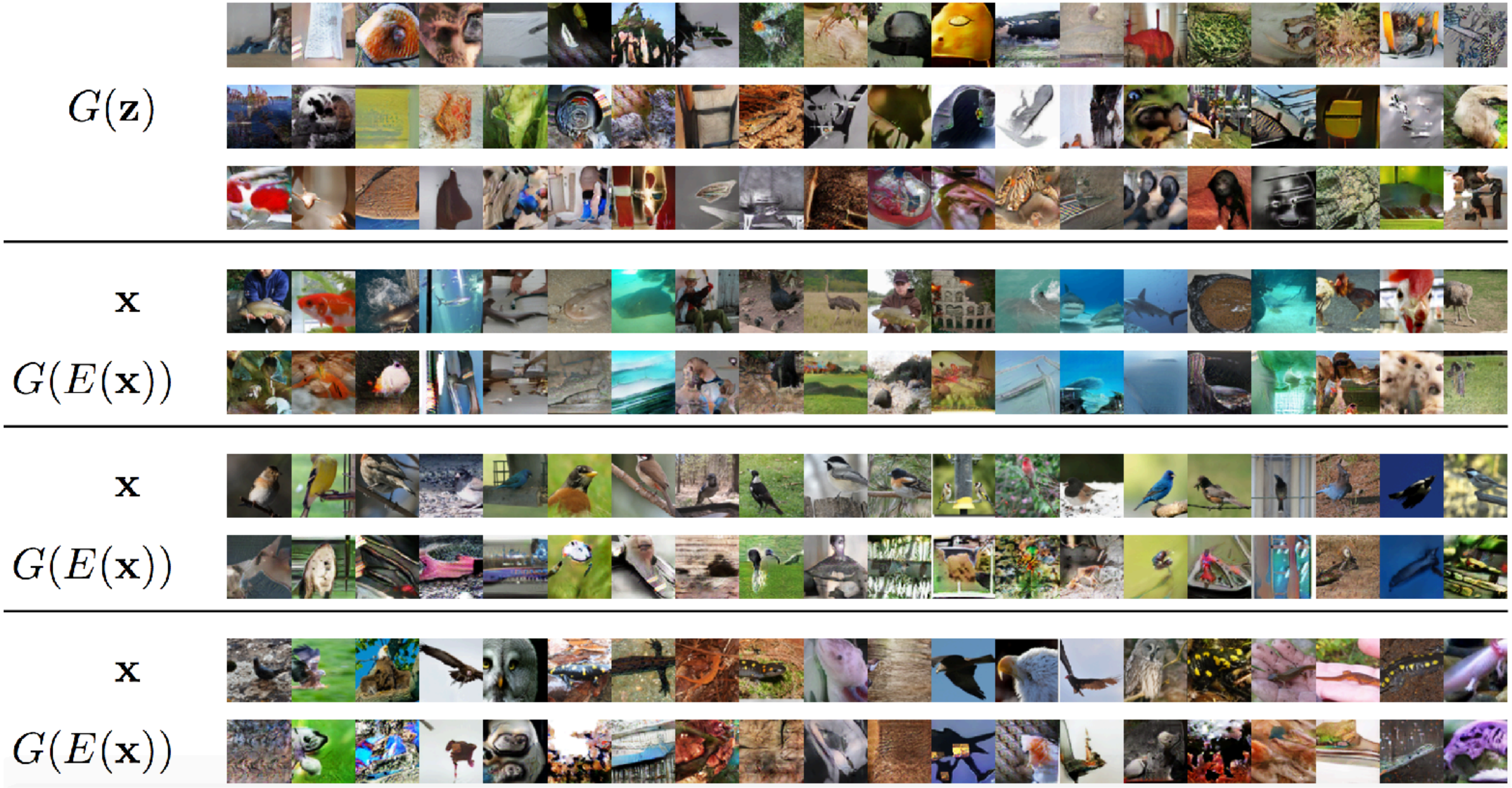
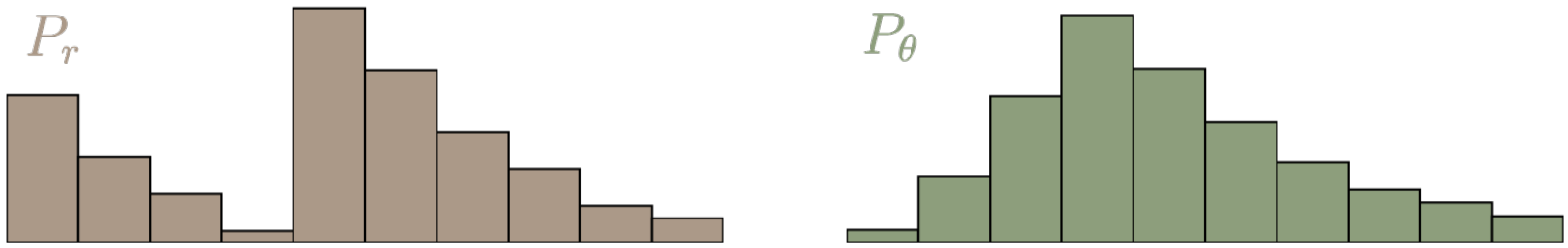


Figure credit: Jeff Donahue, et al., Adversarial Feature Learning

Wasserstein-GAN



Consider the distribution as if it were a block and
move the blocks to make the two distributions the same
"Minimal effort"

$$\mathrm{EMD}(p_{\mathrm{data}}, p_g) = \inf_{\gamma \in \Pi(p_{\mathrm{data}}, p_g)} \mathbb{E}_{(x,y)\sim\gamma}\left[||x - y||\right]$$

$$W(p_{\mathrm{data}}, p_g) = \sup_{||f||_{L \leq 1}} \mathbb{E}_{x \sim p_{\mathrm{data}}}[f(x)] - \mathbb{E}_{x \sim p_g}[f(x)]$$

Figure credit: https://vincentherrmann.github.io/blog/wasserstein/

WGAN Results

Figure 5: Algorithms trained with a DCGAN generator. Left: WGAN algorithm. Right: standard GAN formulation. Both algorithms produce high quality samples.

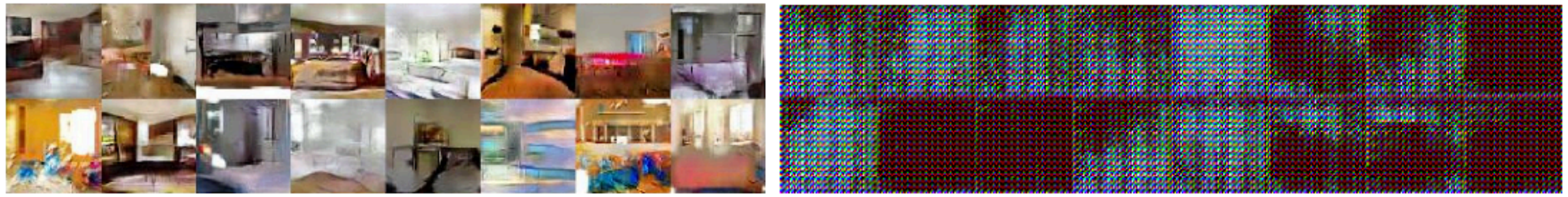


Figure 6: Algorithms trained with a generator without batch normalization and constant number of filters at every layer (as opposed to duplicating them every time as in [18]).

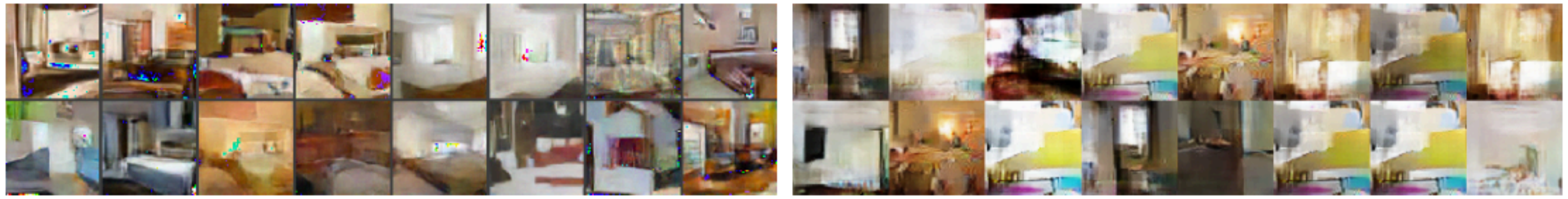


Figure 7: Algorithms trained with an MLP generator with 4 layers and 512 units with ReLU nonlinearities. The number of parameters is similar to that of a DCGAN, but it lacks a

WGAN-GP Results

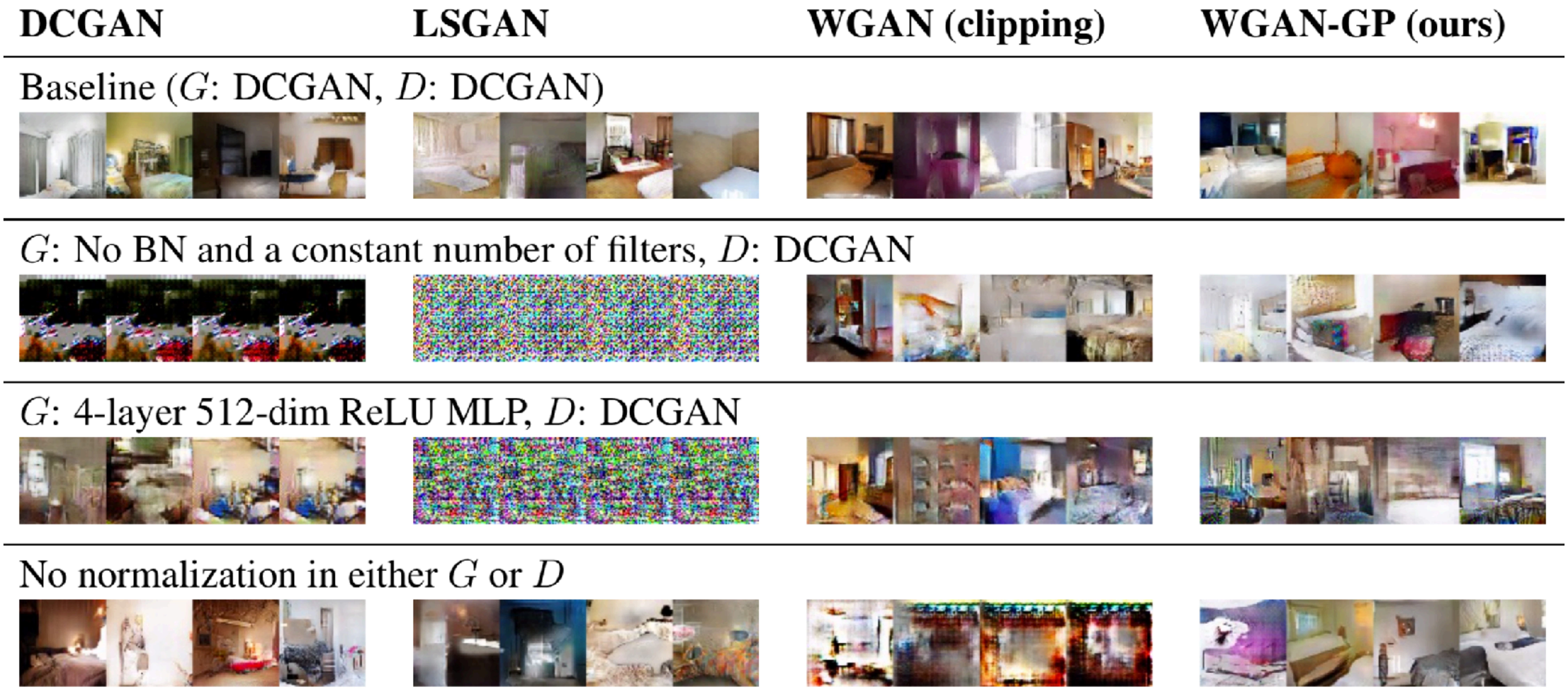


Figure credit: Ishaan Gulrajani, et al., Improved Training of Wasserstein GANs

Pix2Pix

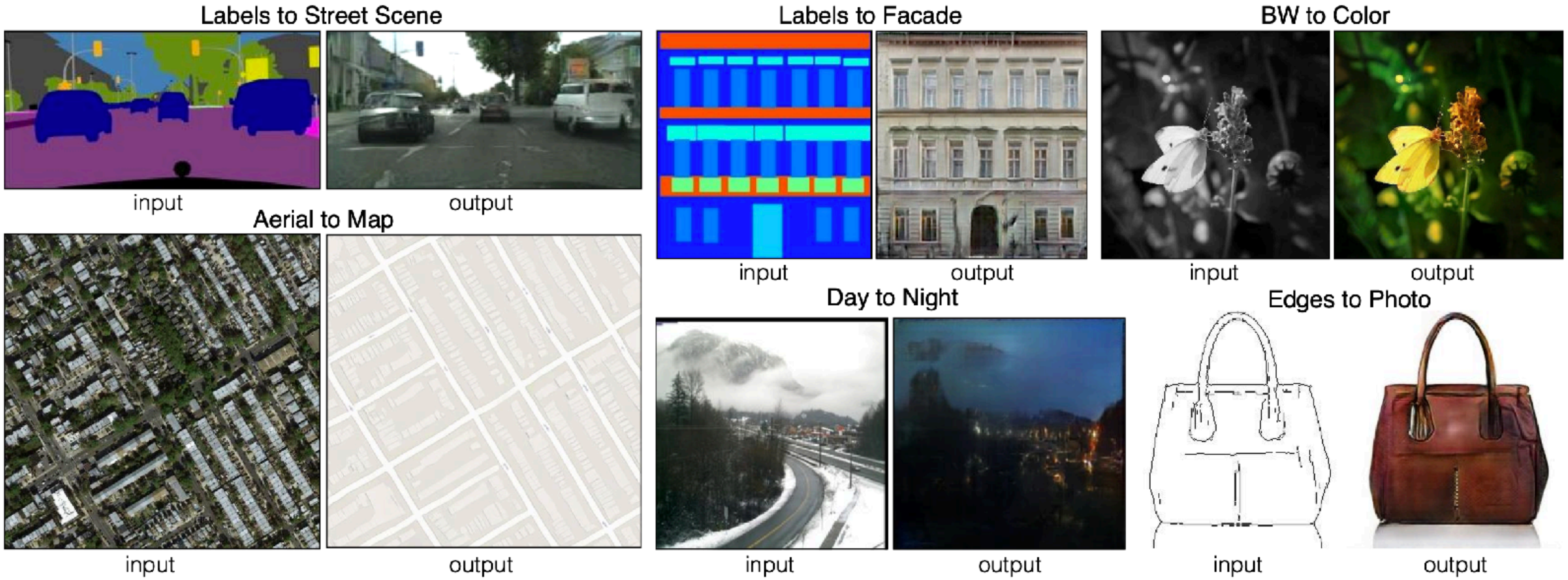


Figure credit: P. Isola, et al., Image-to-Image Translation with Conditional Adversarial Networks

CycleGAN

Figure credit: J-Y. Zhu, et al., Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

Style Transfer

Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "A neural algorithm of artistic style." *arXiv preprint arXiv:1508.06576* (2015).

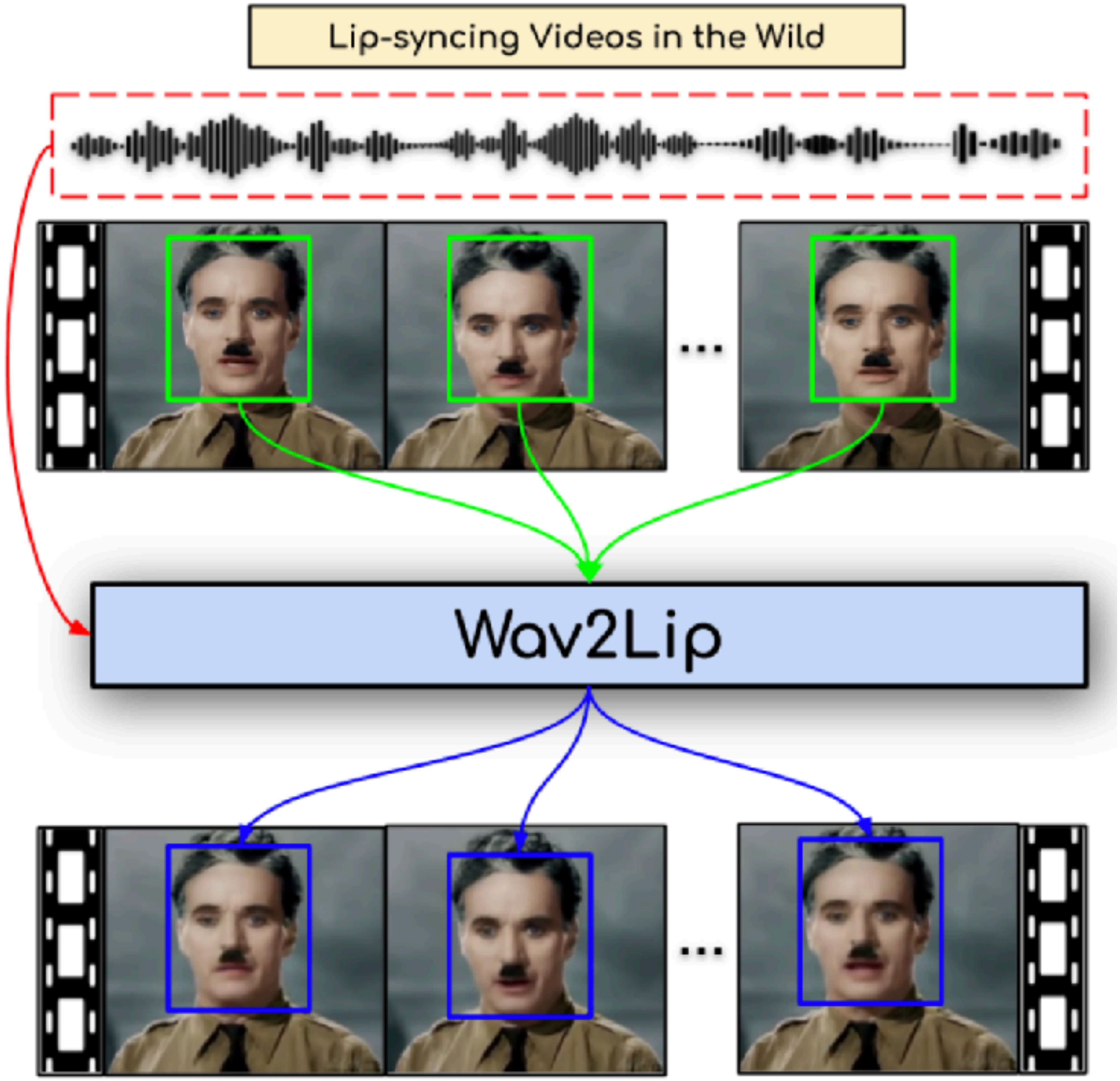
ProgressiveGAN results

1024x1024 images generated using the CELEBA-HQ dataset.

Figure credit: Tero Karras, et al., Progressive Growing of GANs for Improved Quality, Stability, and Variation

StyleGAN Coarse Styles

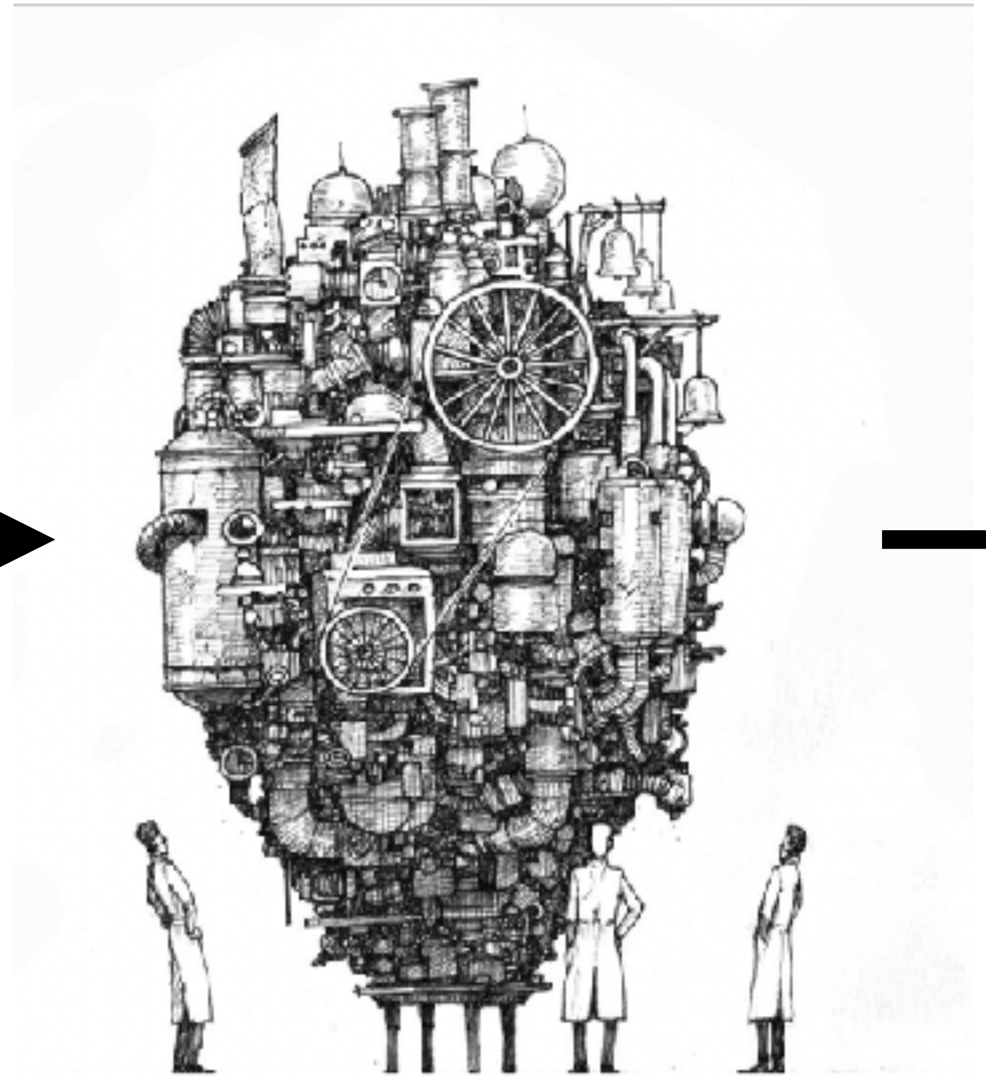
Wav2Lip



- Demo : https://youtu.be/0fXaDCZNOJc

Discriminative vs Generative Models

$$p(y|x) \qquad\qquad p(x|y)$$



“Cat”

Discriminative Models

Generative Models