DEEP LEARNING

Lecture 10: Generative Models

Dr. Yang Lu

Department of Computer Science and Technology

luyang@xmu.edu.cn

Applications of Generative Models

- Image Translation

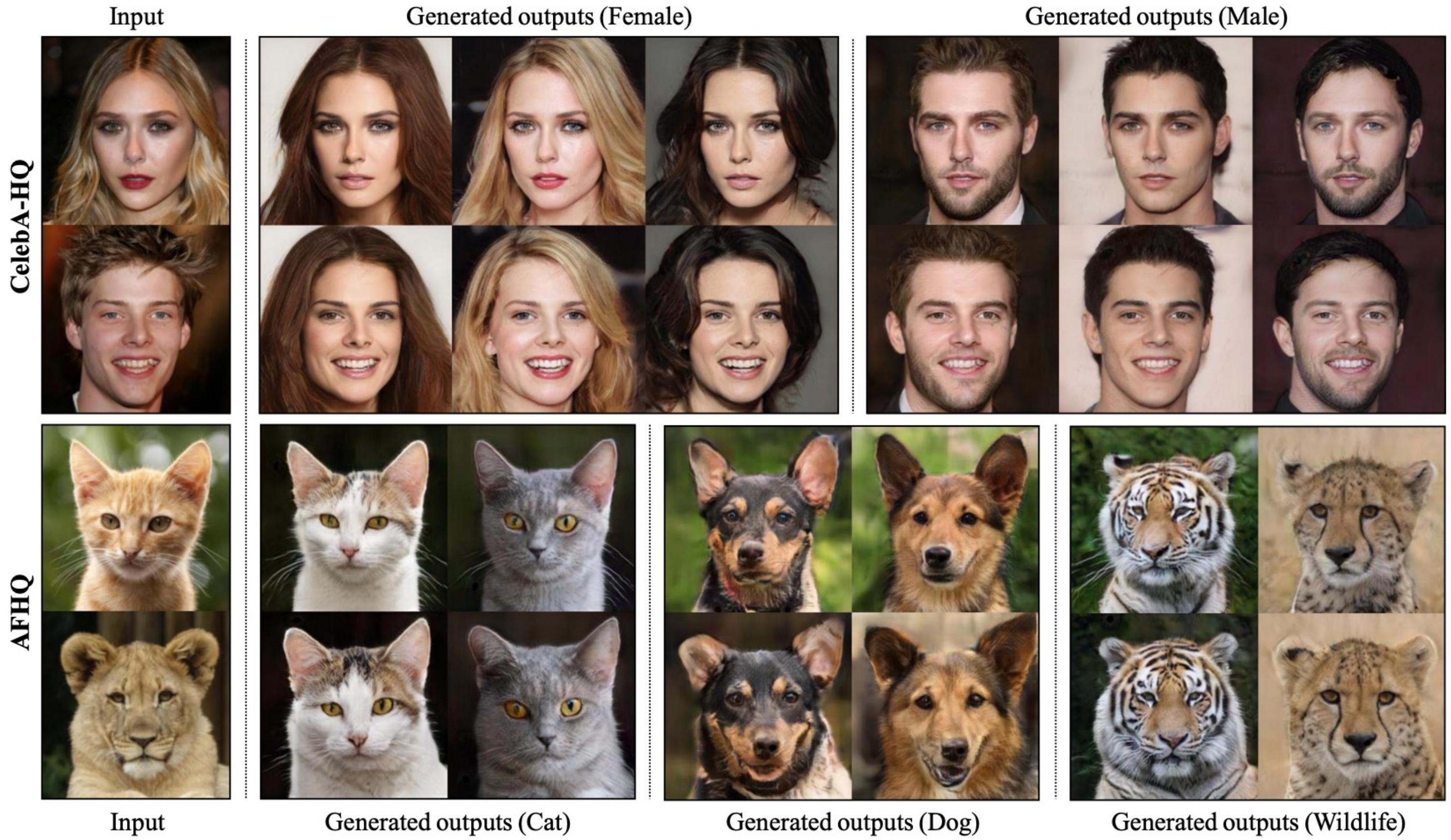


Image source: Choi, Yunjey, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. "Stargan v2: Diverse image synthesis for multiple domains." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8188-8197. 2020.

Applications of Generative Models

- Image Translation

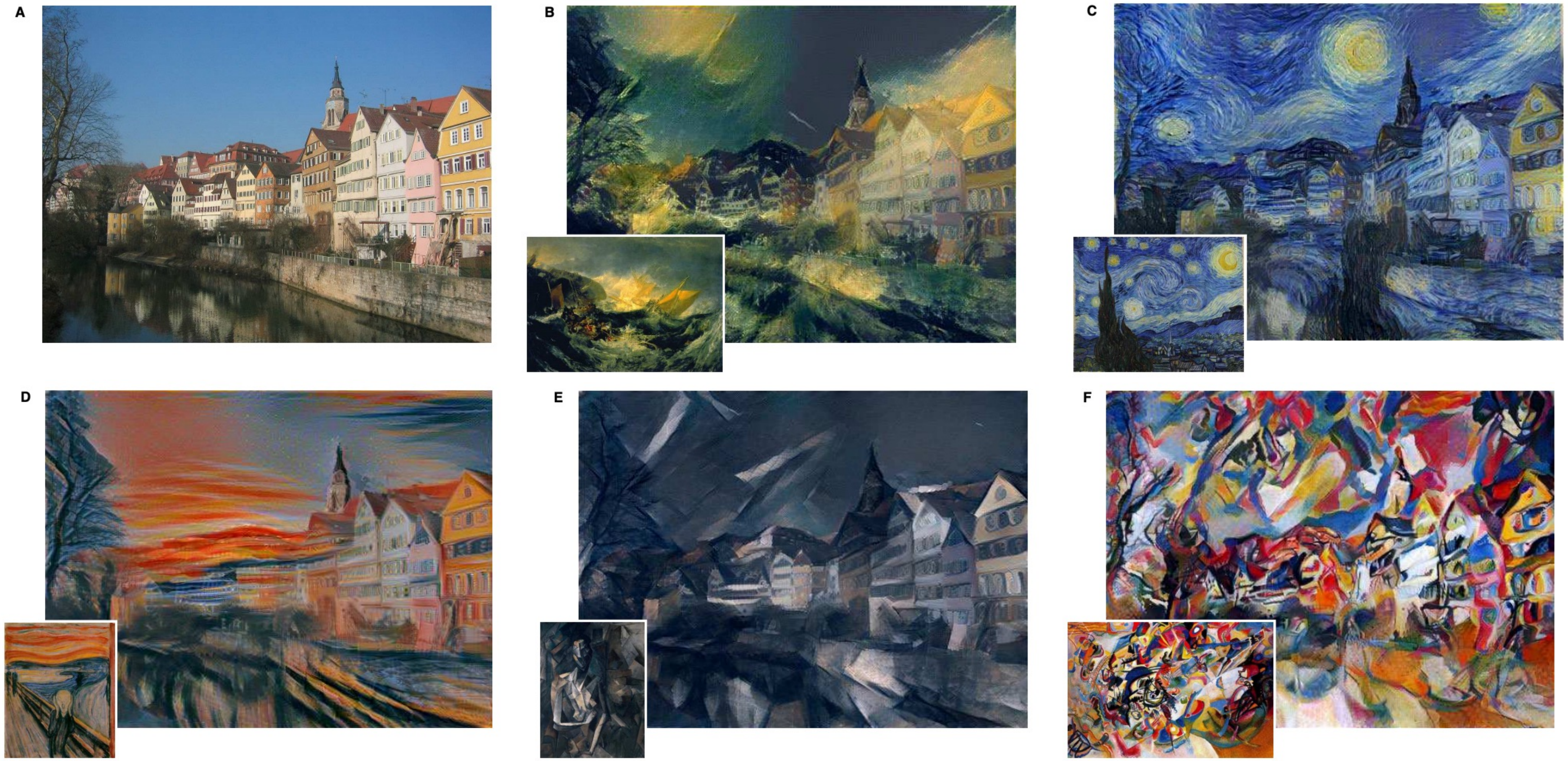


Image source: Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "Image style transfer using convolutional neural networks." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2414-2423. 2016.

Applications of Generative Models

- Scene Generation

Image source: Tang, Hao, Dan Xu, Yan Yan, Philip HS Torr, and Nicu Sebe. "Local class-specific and global image-level generative adversarial networks for semantic-guided scene generation." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7870-7879. 2020.

Applications of Generative Models

- Facial Attribute Manipulation

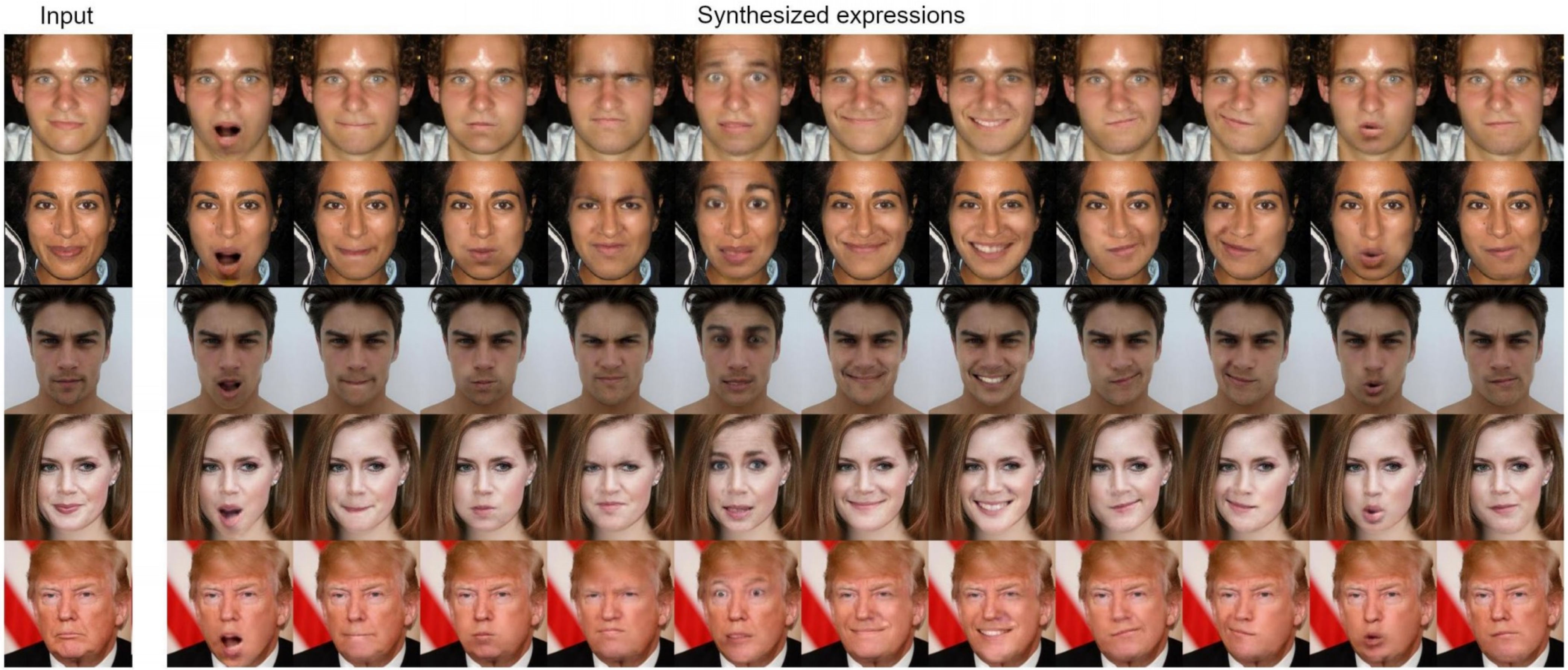


Image source: Geng, Zhenglin, Chen Cao, and Sergey Tulyakov. "3d guided fine-grained face manipulation." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9821-9830. 2019.

Applications of Generative Models

- Facial Attribute Manipulation

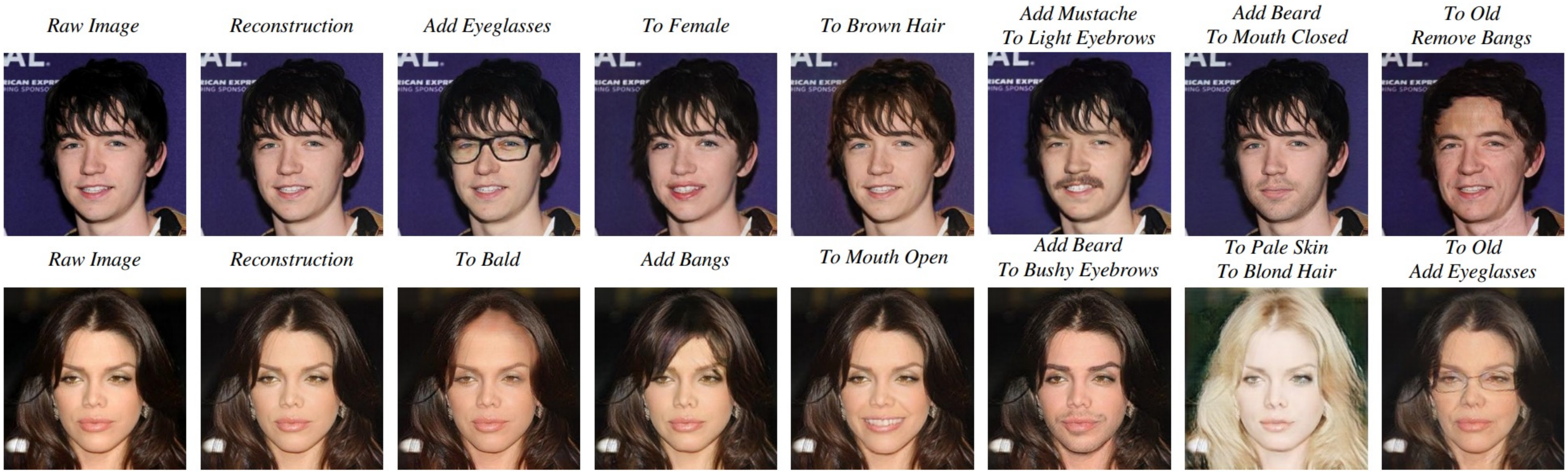


Image source: Liu, Ming, Yukang Ding, Min Xia, Xiao Liu, Errui Ding, Wangmeng Zuo, and Shilei Wen. "Stgan: A unified selective transfer network for arbitrary image attribute editing." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3673-3682. 2019.

Applications of Generative Models

- Gaze Correction

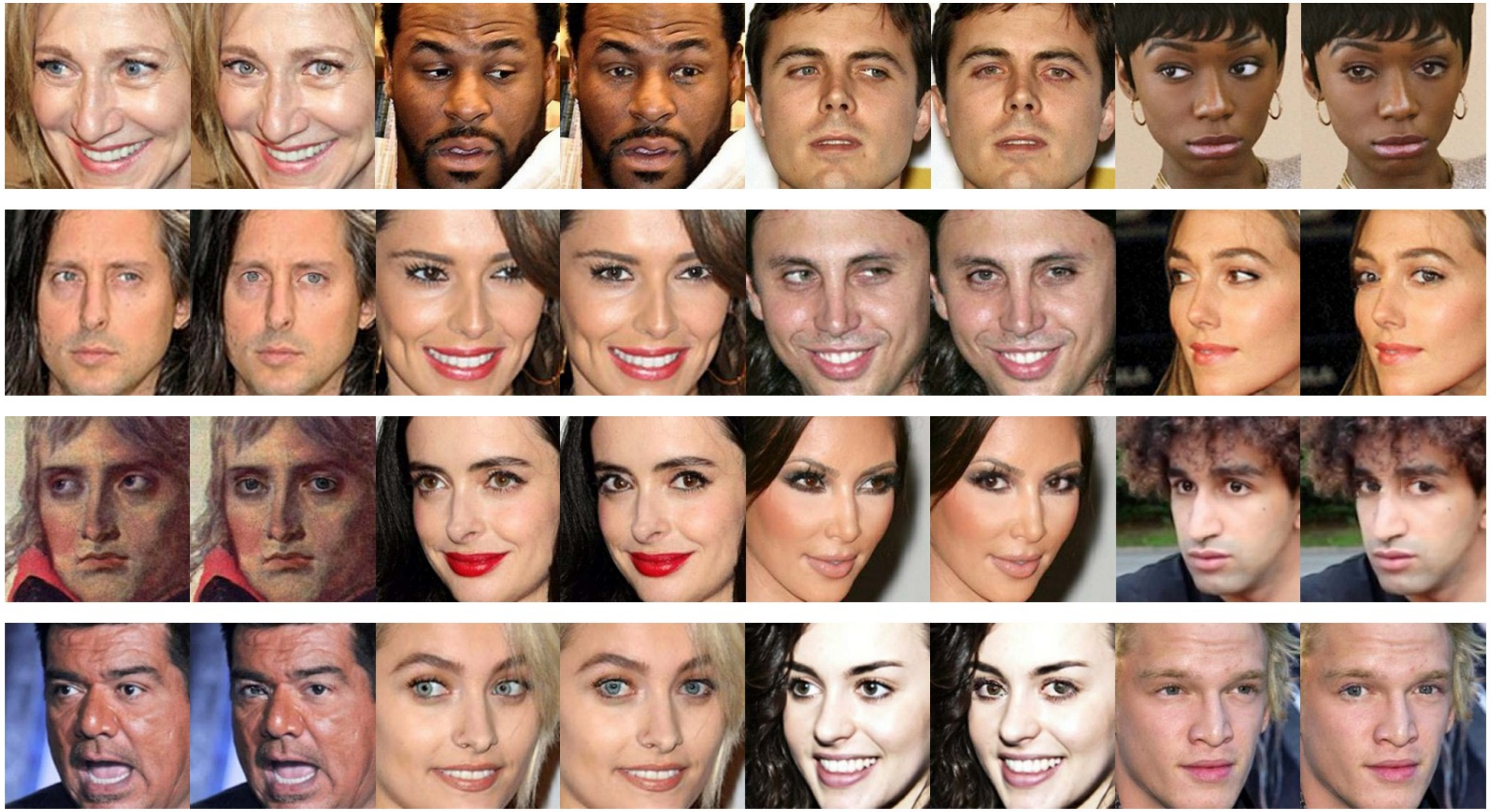


Image source: Zhang, Jichao, Meng Sun, Jingjing Chen, Hao Tang, Yan Yan, Xueying Qin, and Nicu Sebe. "Gazecorrection: Self-guided eye manipulation in the wild using self-supervised generative adversarial networks." arXiv preprint arXiv:1906.00805 (2019).

Applications of Generative Models

- Image Animation

Image source: Siarohin, Aliaksandr, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. "Animating arbitrary objects via deep motion transfer." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2377-2386. 2019.

Applications of Generative Models

- Image Inpainting

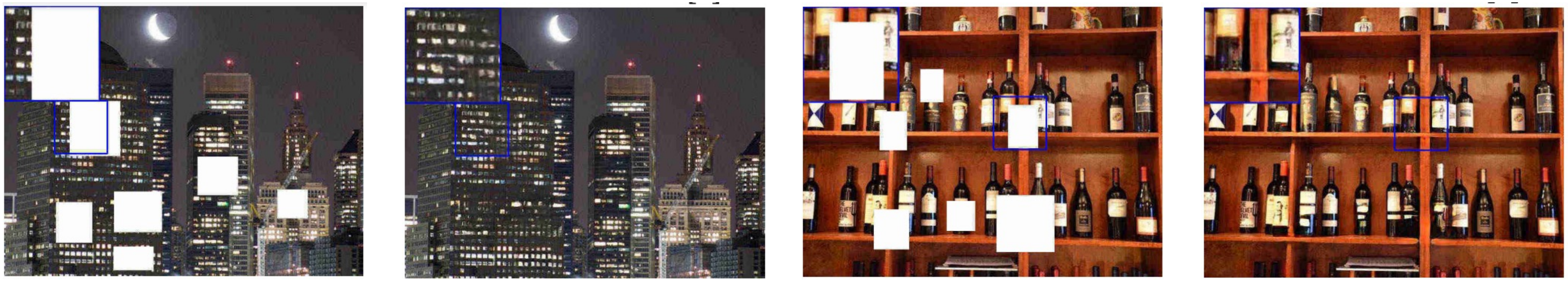


Image source: Yu, Jiahui, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. "Generative image inpainting with contextual attention." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5505-5514. 2018.

Applications of Generative Models

- Image Inpainting

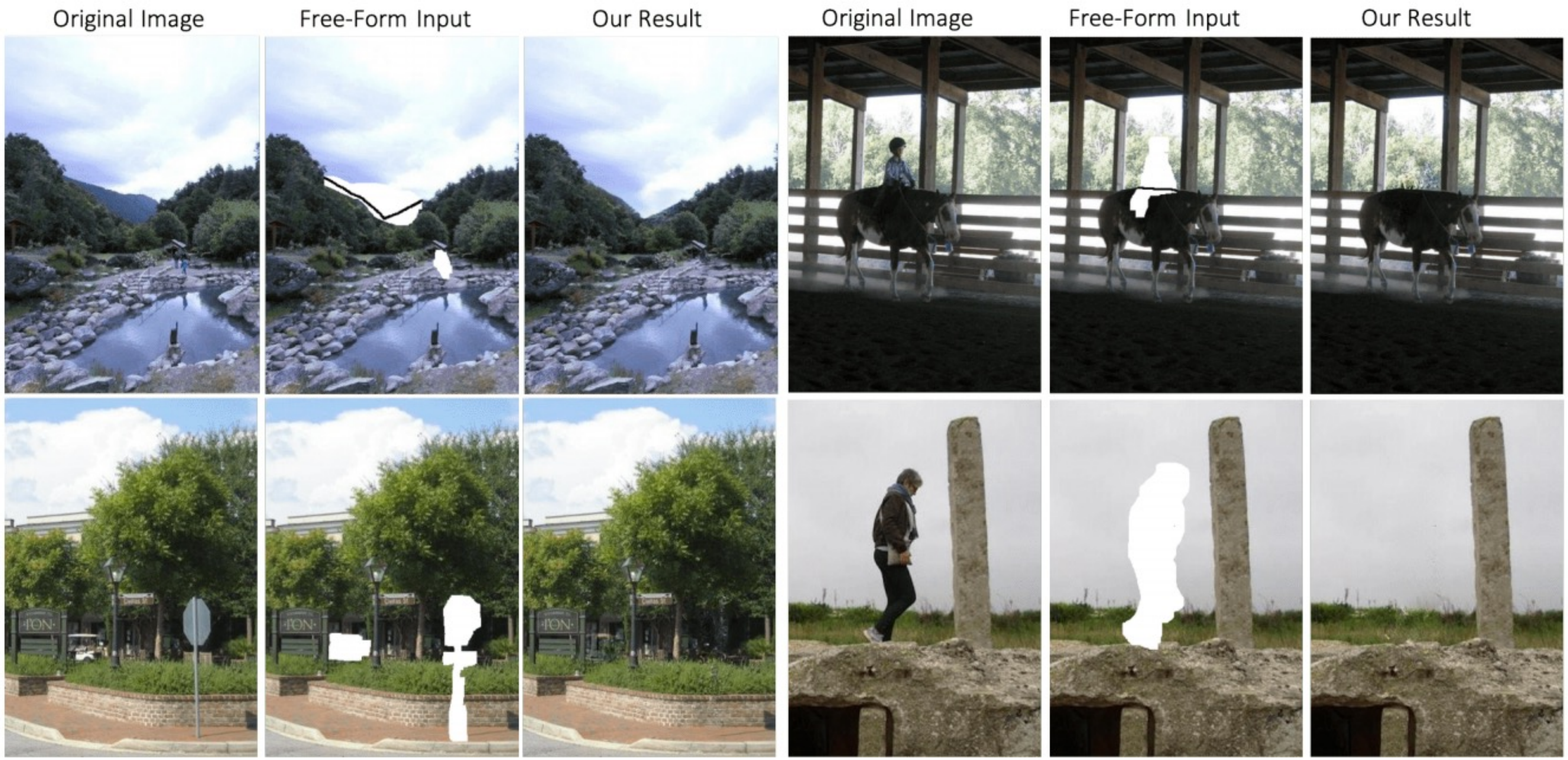


Image source: Yu, Jiahui, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. "Free-form image inpainting with gated convolution." In Proceedings of the IEEE International Conference on Computer Vision, pp. 4471-4480. 2019.

Applications of Generative Models

- Image Blending

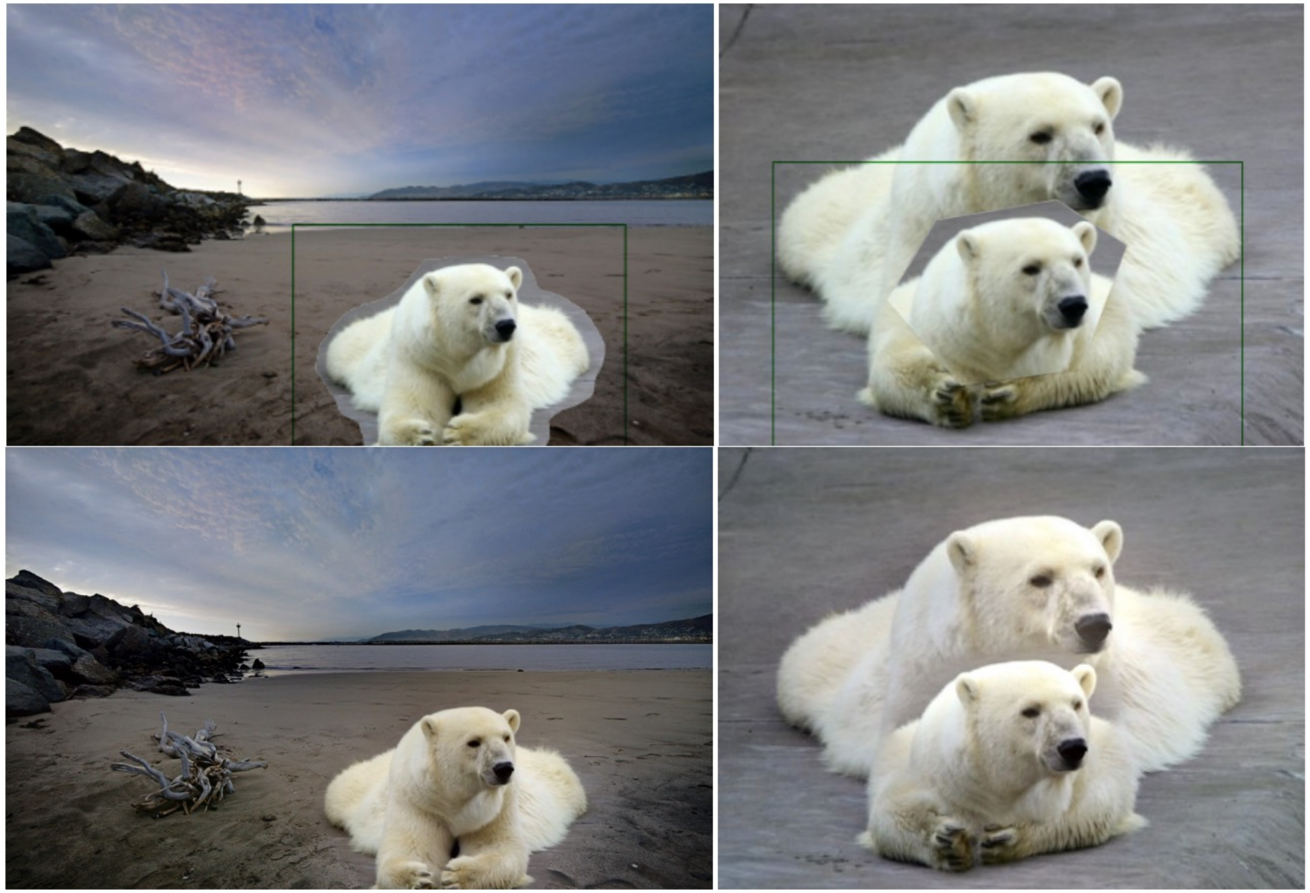


Image source: Wu, Huikai, Shuai Zheng, Junge Zhang, and Kaiqi Huang. "Gp-gan: Towards realistic high-resolution image blending." In Proceedings of the 27th ACM International Conference on Multimedia, pp. 2487-2495. 2019.

Applications of Generative Models

- Image Super-Resolution

Image source: Ledig, Christian, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken et al. "Photo-realistic single image super-resolution using a generative adversarial network." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4681-4690. 2017.
https://www.theverge.com/21298762/face-depixelizer-ai-machine-learning-tool-pulse-stylegan-obama-bias

Applications of Generative Models

- Makeup

Image source: Li, Tingting, Ruihe Qian, Chao Dong, Si Liu, Qiong Yan, Wenwu Zhu, and Liang Lin. "Beautygan: Instance-level facial makeup transfer with deep generative adversarial network." In Proceedings of the 26th ACM international conference on Multimedia, pp. 645-653. 2018.

Applications of Generative Models

- Text-to-image generation

"a hedgehog using a calculator"

"a corgi wearing a red bowtie and a purple party hat"

"robots meditating in a vipassana retreat"

"a fall landscape with a small cottage next to a lake"

Image source: AlexanderQuinnNichol,PrafullaDhariwal,AdityaRamesh,PranavShyam,PamelaMishkin,BobMcgrew,IlyaSutskever,andMarkChen.2022. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. In International Conference on Machine Learning. 16784–16804.

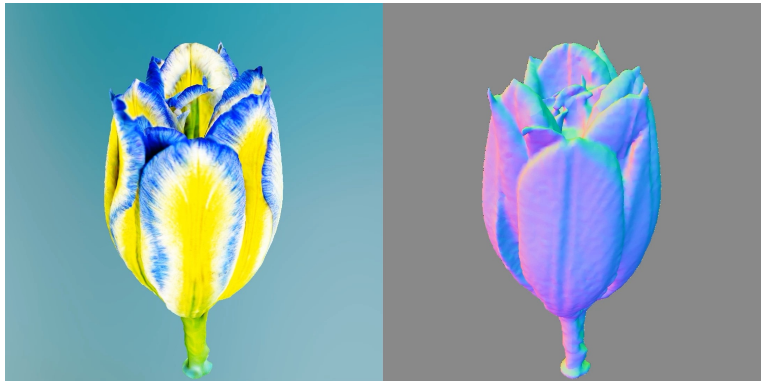
Applications of Generative Models

- Video generation

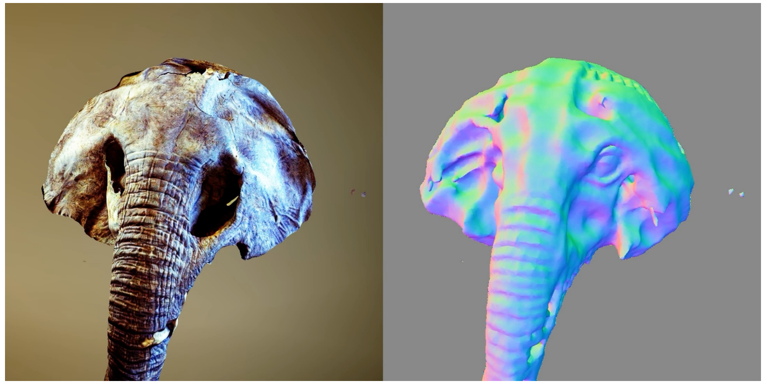
Video source: https://github.com/Picsart-AI-Research/Text2Video-Zero

Applications of Generative Models

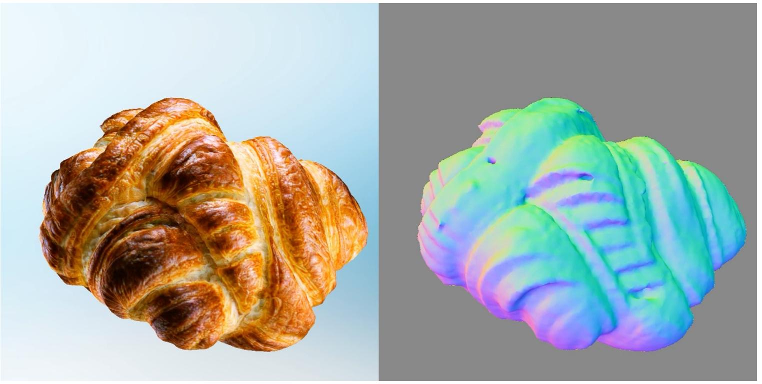
- Text-to-3D generation



A blue tulip



An elephant skull

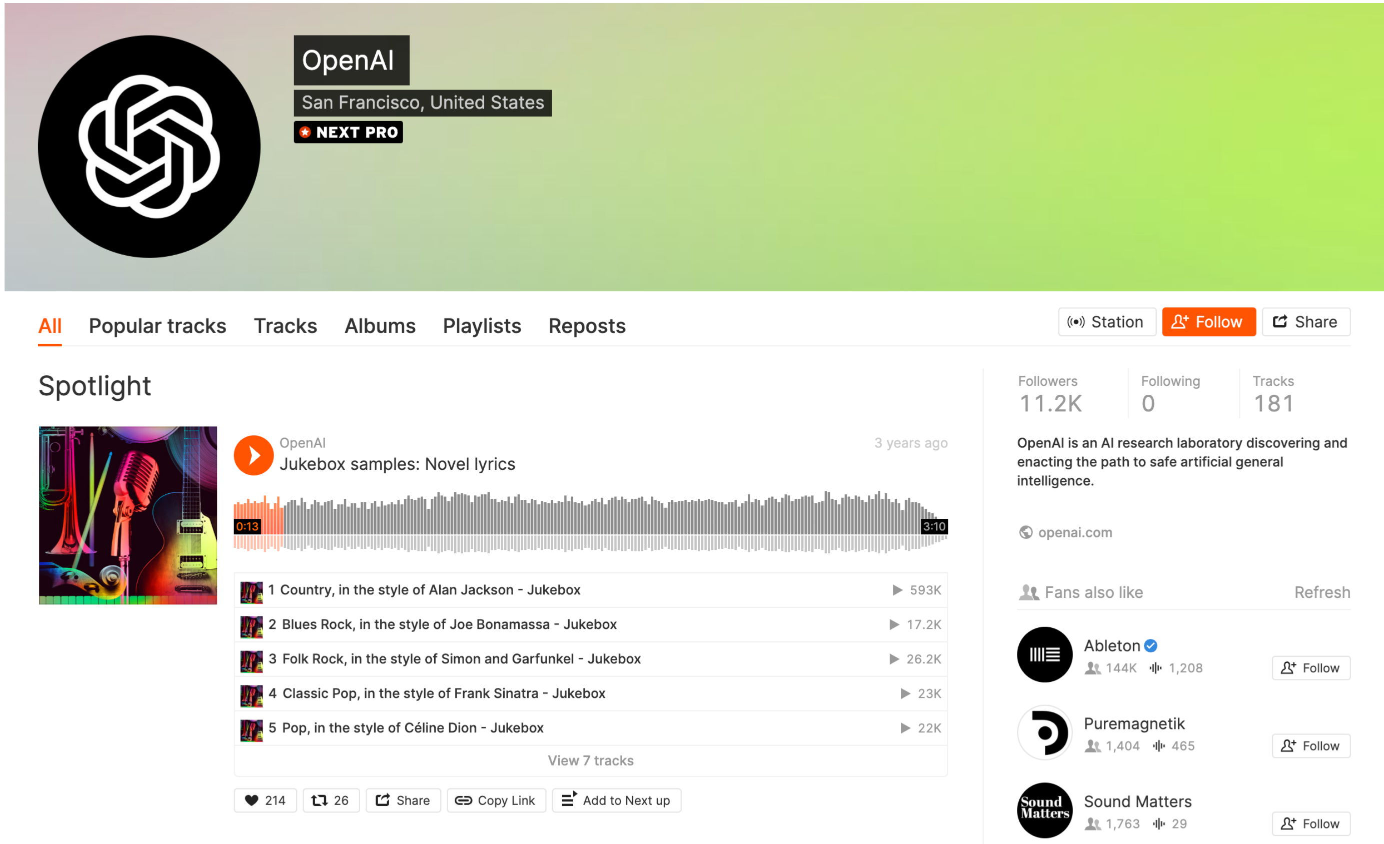


A delicious croissant

Michelangelo style statue of dog reading news on a cellphone

Applications of Generative Models

Music Generation



Generated music samples: https://soundcloud.com/openai_audio

Generative vs. Discriminative

- In machine learning, two main approaches are called the generative approaches and the discriminative approaches.
- Given an observable variable X and a target variable Y:
  - A generative model is a statistical model of the data distribution $P(X)$ or the joint probability distribution on $X \times Y$: $P(X, Y)$.
  - A discriminative model is a model of the conditional distribution of Y given X: $P(Y|X = x)$.

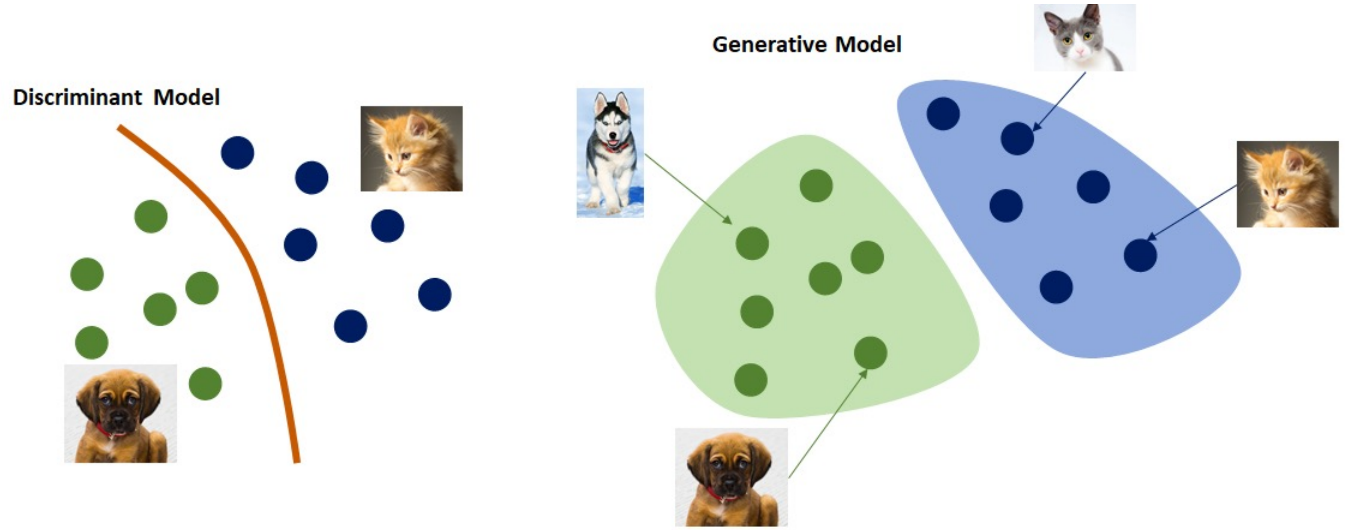


Image source: https://medium.com/@jordi299/about-generative-and-discriminative-models-d8958b67ad32

Discriminative Approaches

- Most supervised learning methods fall into discriminative approaches.
  - Given data: (x, y), x is data, y is label.
  - Goal: Learn a function to map $x \rightarrow y$, namely posterior probability $P(Y|X = x)$.
  - Examples: Classification, regression, object detection, face recognition, sentiment classification, etc.

$\longrightarrow$ cat

$\longrightarrow$ car

Image source: https://www.autoexpress.co.uk/tesla/model-3
https://timesofindia.indiatimes.com/life-style/relationships/pets/5-things-that-scare-and-stress-your-cat/articleshow/67586673.cms

Generative Approaches

- Given training data, generate new samples from same distribution.
- Objectives:

1. Learn $p_{model}(x)$ that approximates $p_{data}(x)$.
2. Sample a new x from $p_{model}(x)$.

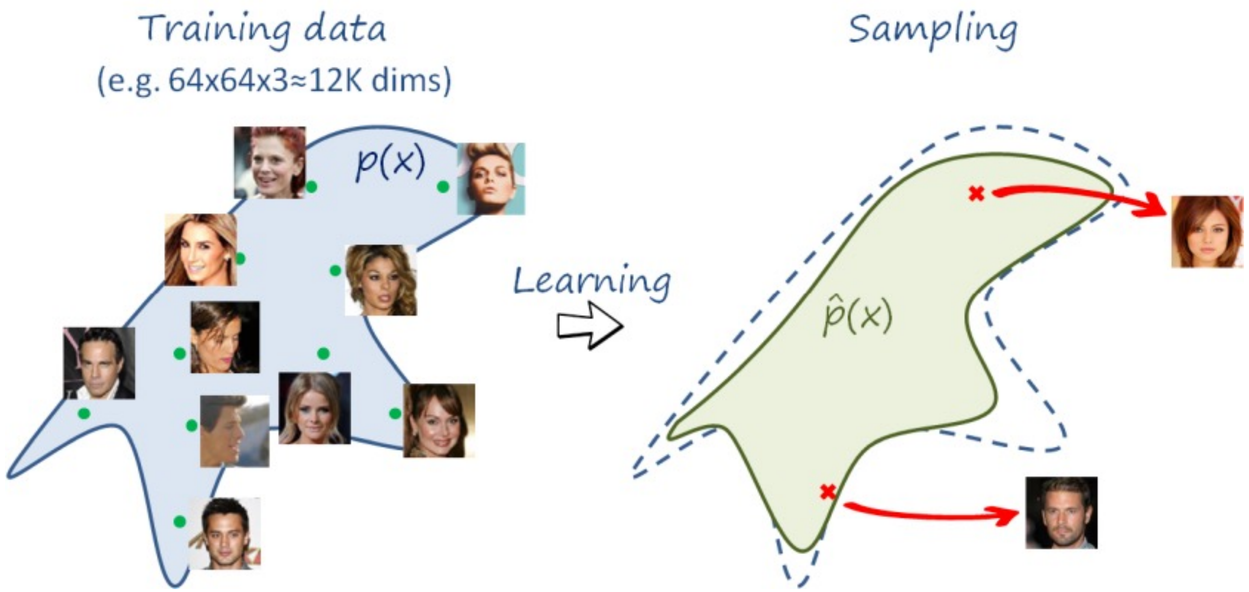
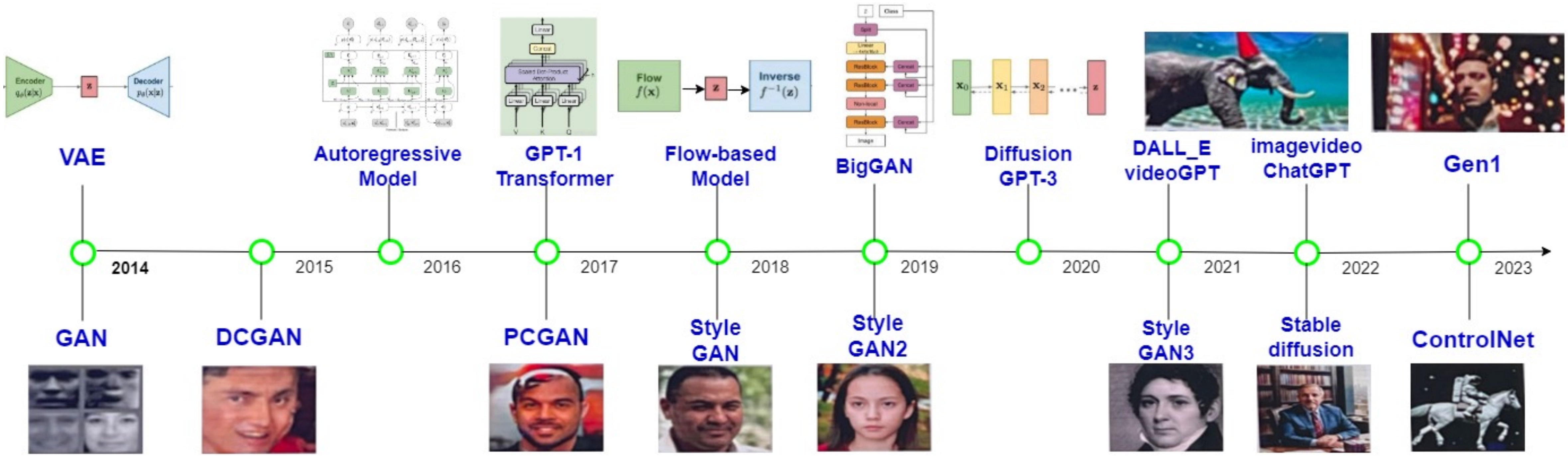


Image source: http://www.lherranz.org/2018/08/07/imagetranslation/

Generative Models

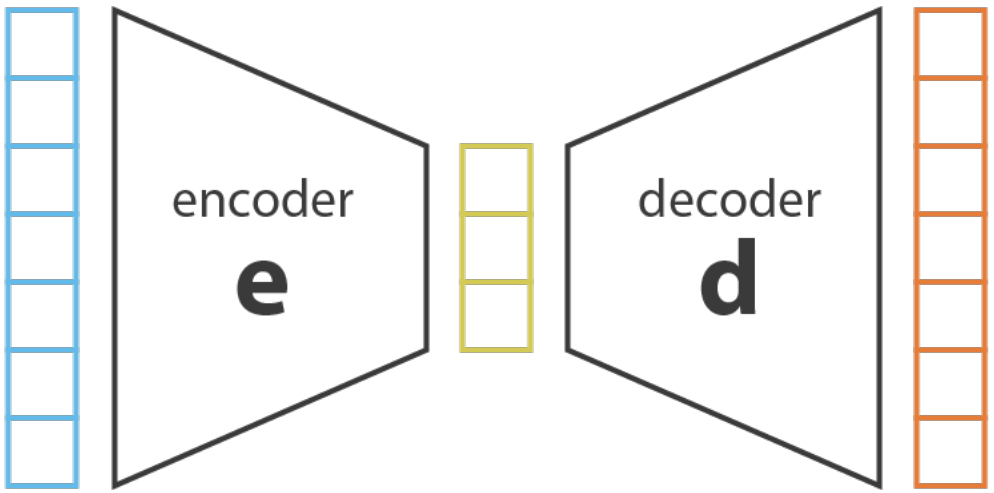


Outlines

- Variational Autoencoder
- Generative Adversarial Nets
- Diffusion Model

VARIATIONAL AUTOENCODER

Autoencoder



x | $e(x)$ | $d(e(x))$

initial data
in space R^n

encoded data
in latent space R^m (with $m<n$)

encoded-decoded data
back in the initial space R^n

$x = d(e(x))$ ➡ **lossless encoding**
no information is lost when reducing the number of dimensions

$x \neq d(e(x))$ ➡ **lossy encoding**
some information is lost when reducing the number of dimensions and can't be recovered later

$$\text{loss} = \| x - \hat{x} \|^2 = \| x - d(z) \|^2 = \| x - d(e(x)) \|^2$$

Variational Autoencoder

Auto-encoding variational bayes
DP Kingma, M Welling - arXiv preprint arXiv:1312.6114, 2013 - arxiv.org
… **variational** lower bound yields a simple differentiable unbiased estimator of the lower bound; this SGVB (Stochastic Gradient **Variational Bayes**… , we propose the **AutoEncoding** VB (AEVB…

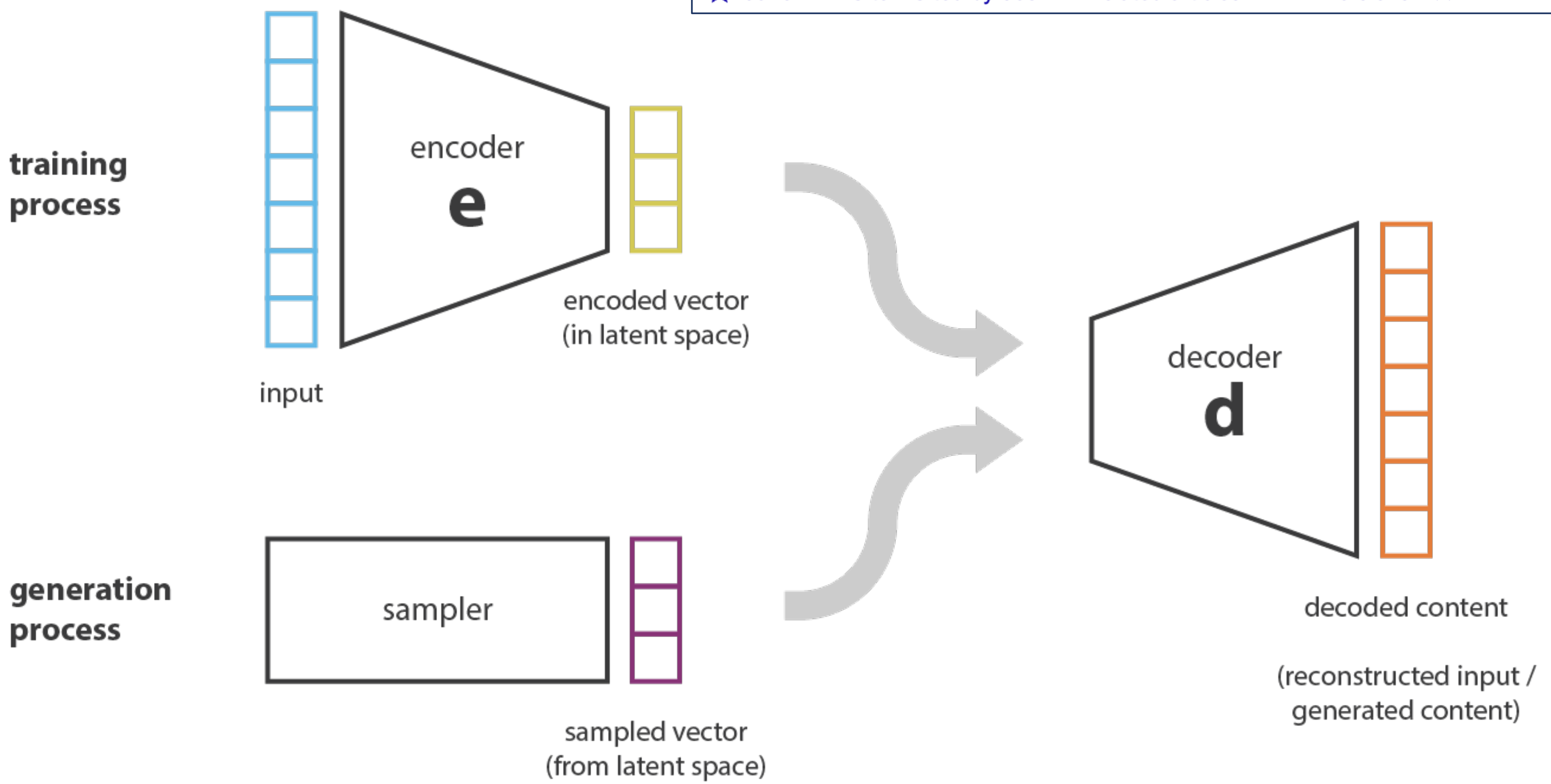


Image source: https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

Variational Autoencoder

- The autoencoder is solely trained to encode and decode with as few loss as possible, no matter how the latent space is organized.

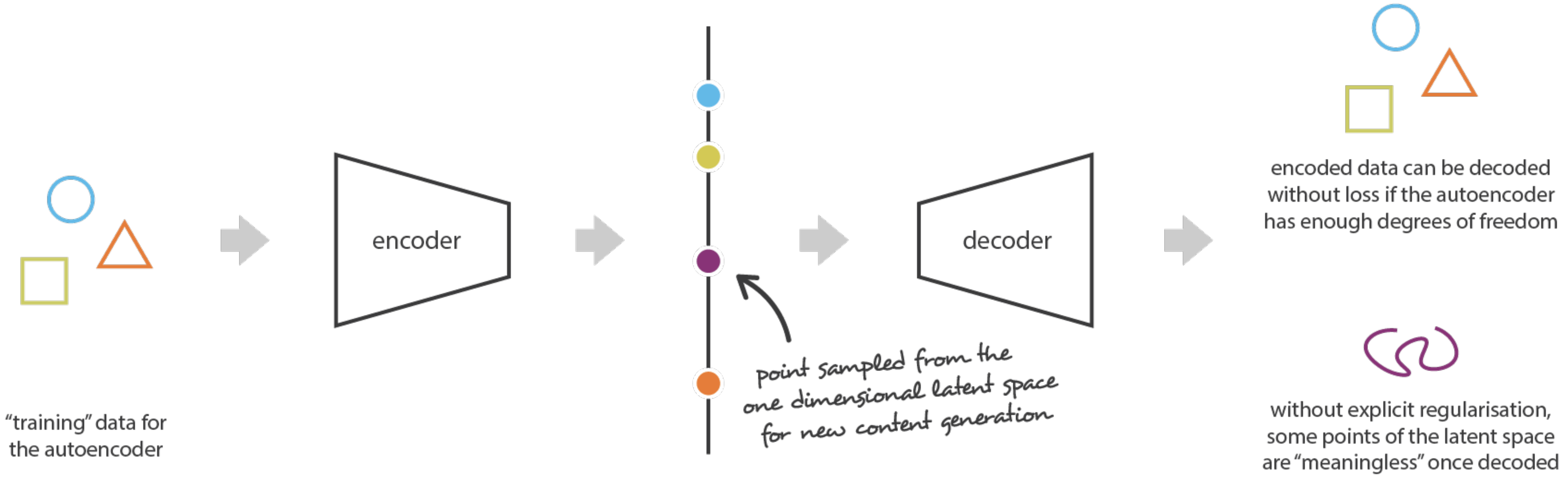


Image source: https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

Variational Autoencoder

- A variational autoencoder can be defined as being an autoencoder whose training is:
  - regularised to avoid overfitting;
  - ensure that the latent space has good properties that enable generative process.

Variational Autoencoder

- Instead of encoding an input as a single point, we encode it as a **distribution over the latent space**.

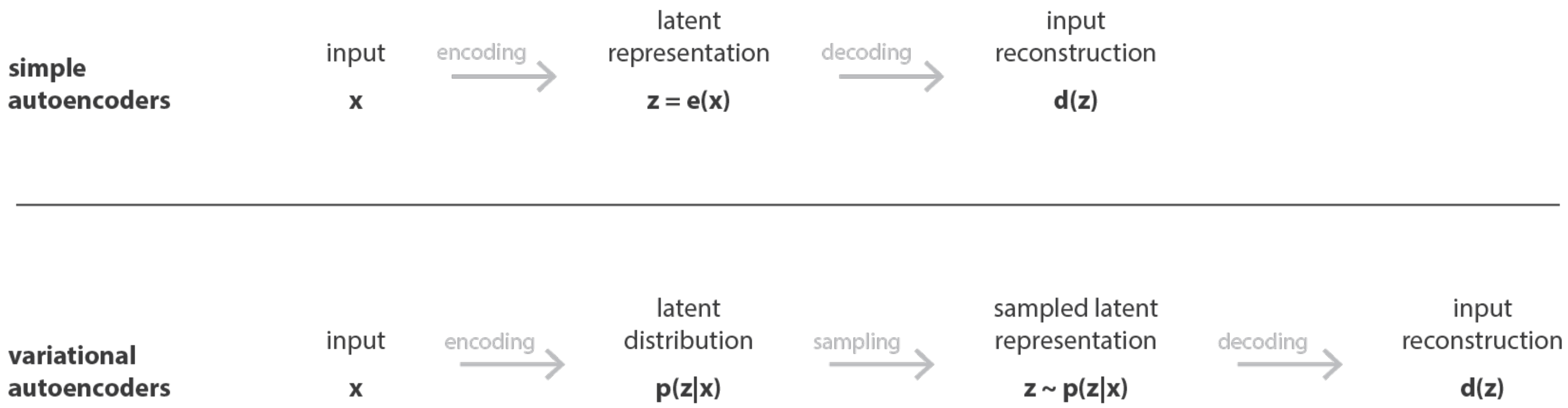
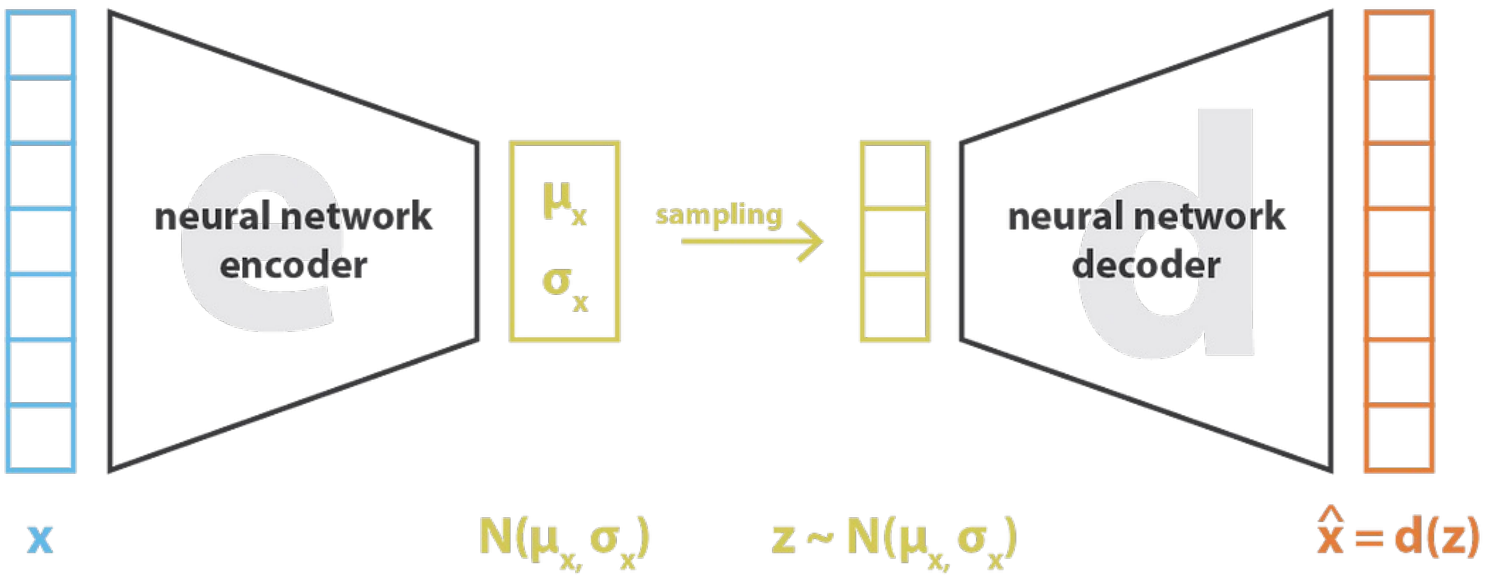


Image source: https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

Variational Autoencoder

- Regularize the organization of the latent space by making the distributions returned by the encoder close to a standard normal distribution.



$$\text{loss} = \| x - \hat{x} \|^2 + \text{KL}[\, N(\mu_x, \sigma_x), N(0, I)\,] = \| x - d(z) \|^2 + \text{KL}[\, N(\mu_x, \sigma_x), N(0, I)\,]$$

Image source: https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

Variational Autoencoder

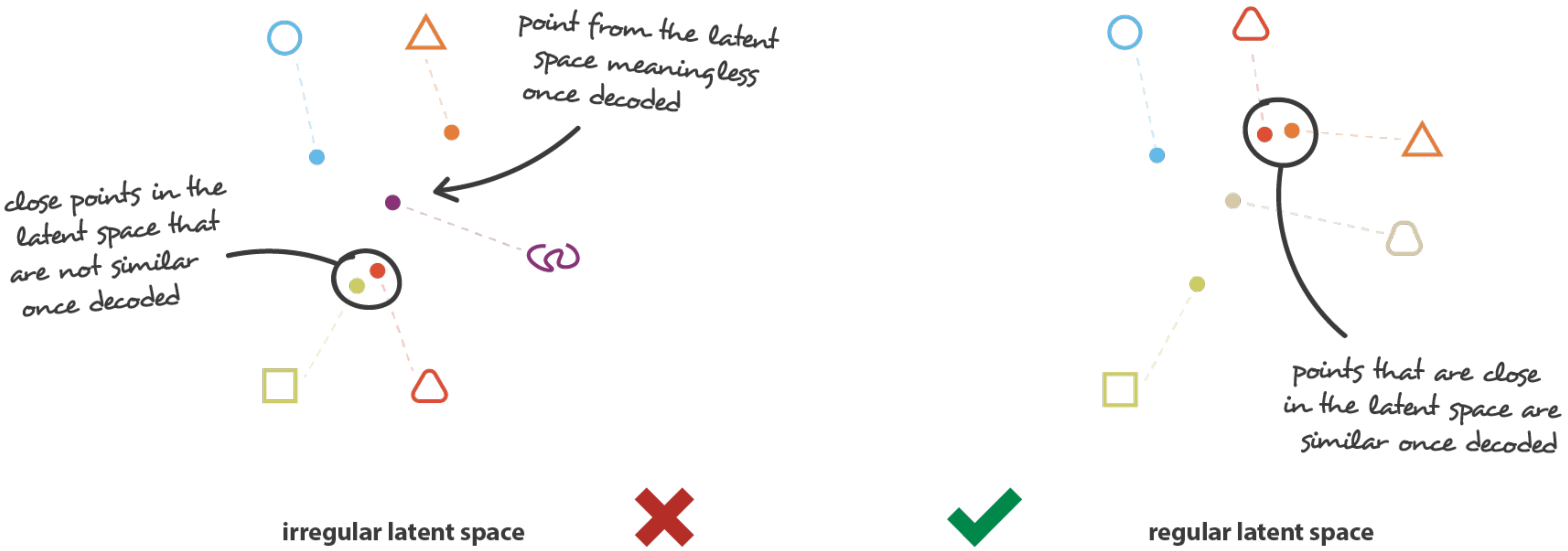


Image source: https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

Variational Autoencoder

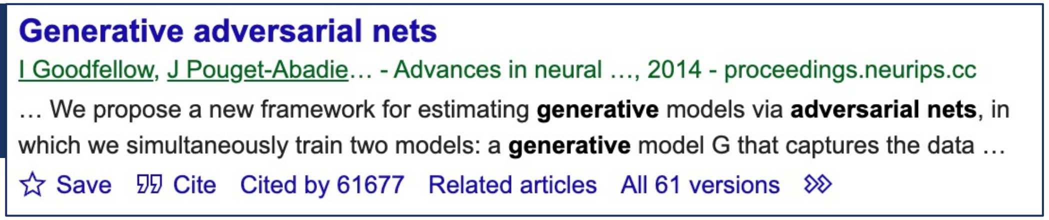
- The loss of the autoencoder is to minimize both:
  - the reconstruction loss (how similar the autoencoder's output to its input);
  - the latent loss (how close its hidden nodes were to a normal distribution).
- It doesn't guarantee the quality of the generated image. A major drawback of VAEs is the blurry outputs that they generate. VAE models tend to produce unrealistic, blurry samples.

Human faces generated by VAEs

Image source: https://medium.com/sciforce/generative-models-under-a-microscope-comparing-vaes-gans-and-flow-based-models-344f20085d83#:~:text=A%20major%20drawback%20of%20VAEs,to%20produce%20unrealistic%2C%20blurry%20samples.

GAN

GAN


Generative adversarial nets
I Goodfellow, J Pouget-Abadie… - Advances in neural …, 2014 - proceedings.neurips.cc
… We propose a new framework for estimating **generative** models via **adversarial nets**, in which we simultaneously train two models: a **generative** model G that captures the data …
Save Cite Cited by 61677 Related articles All 61 versions

- GAN was proposed by Ian Goodfellow in 2014.
- Yann LeCun described GANs as “*the most interesting idea in the last 10 years in machine learning*”.
- Ian presented and explained his paper in NIPS 2016 with a 2-hour presentation.

Ian in NIPS 2016

Image source: https://youtu.be/HGYYEUSm-0Q

GAN: How to Do

- GAN is composed by a generator and a discriminator. They are both neural networks.
  - Generator network: try to fool the discriminator by generating real-looking images.
  - Discriminator network: try to distinguish between real and fake images.

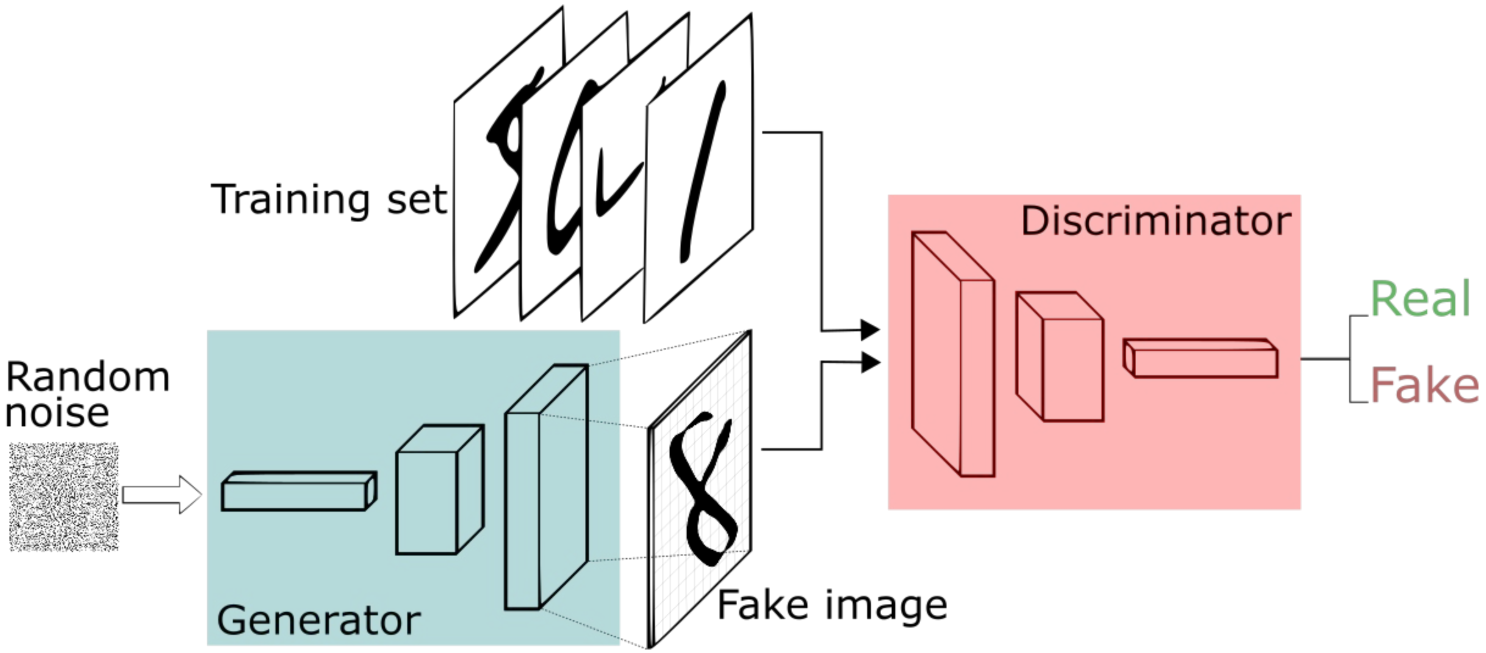


Image source: https://sthalles.github.io/intro-to-gans/

GAN: How to Do

- Generator and discriminator tells each other where it was wrong.
  - Generator tells discriminator how I fool you.
  - Discriminator tells generator how I detect you.

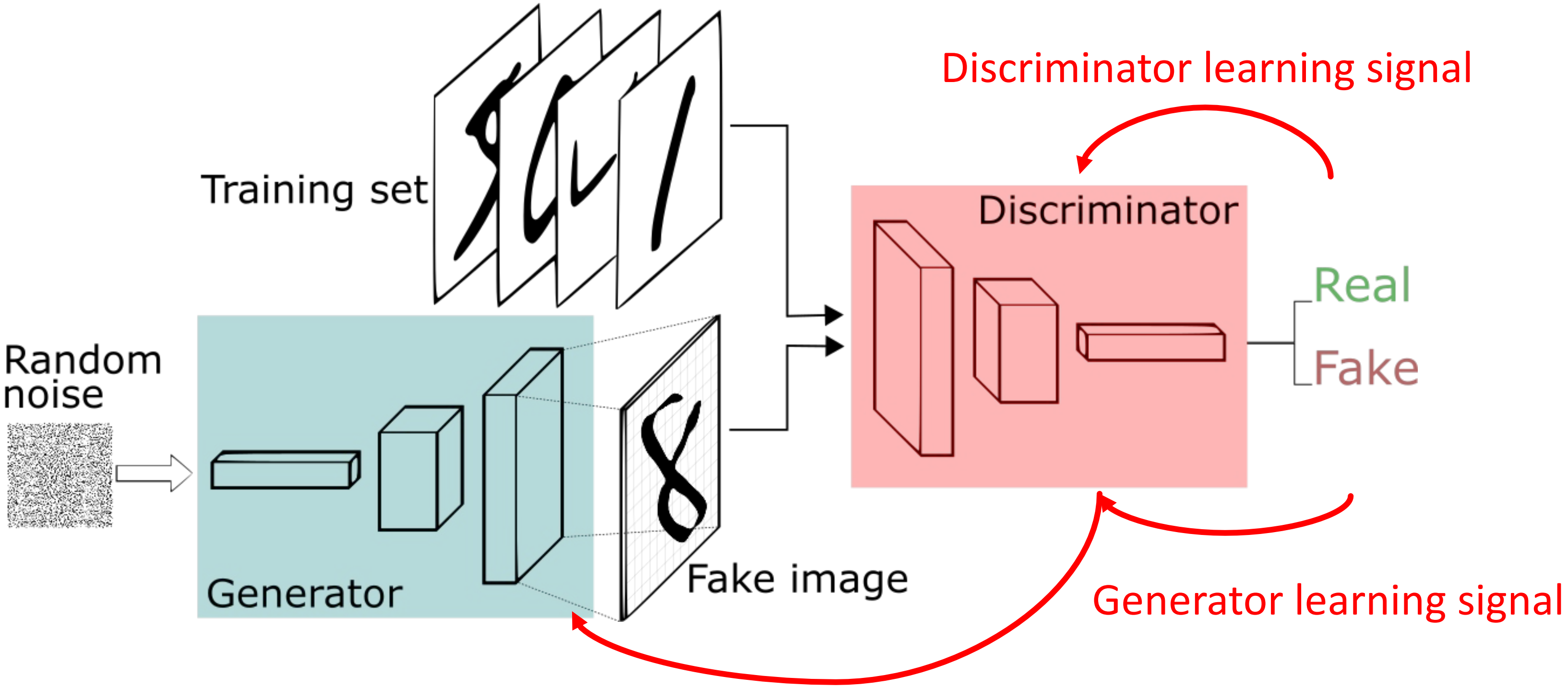


Image source: https://sthalles.github.io/intro-to-gans/

GAN: How to Learn

- Given a prior on input noise variables $\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})$, generator $G_{\theta_g}(\boldsymbol{z})$ take $\boldsymbol{z}$ as input and map it into data space.
- Discriminator $D_{\theta_d}(\boldsymbol{x})$ takes the data $\boldsymbol{x} \sim p_{data}$ as input and output probability that $\boldsymbol{x}$ came from the real data rather than generated data $G_{\theta_g}(\boldsymbol{z})$.
- D_{θ_d} and G_{θ_g} have different goals (1 for real, 0 for fake):
  - Generator wants: $D_{\theta_d}\left(G_{\theta_g}(\boldsymbol{z})\right) \rightarrow 1.$
  - Discriminator wants: $D_{\theta_d}(\boldsymbol{x}) \rightarrow 1,\ D_{\theta_d}\left(G_{\theta_g}(\boldsymbol{z})\right) \rightarrow 0.$

GAN: How to Learn

- By maximizing the log-likelihood, the overall objective is to simultaneously train over all $\boldsymbol{x}$ with random generated $\boldsymbol{z}$:
  - train G_{θ_g} to minimize $\log\left(1 - D_{\theta_d}\left(G_{\theta_g}(\boldsymbol{z})\right)\right)$;
  - train D_{θ_d} to maximize $\log D_{\theta_d}(\boldsymbol{x})$ and $\log\left(1 - D_{\theta_d}\left(G_{\theta_g}(\boldsymbol{z})\right)\right)$.
- In other words, D_{θ_d} and G_{θ_g} play the following two-player minimax game:

$$\min_{\theta_g} \max_{\theta_d} \left[\underbrace{\mathbb{E}_{\boldsymbol{x}\sim p_{data}} \log D_{\theta_d}(\boldsymbol{x})}_{\text{Discriminator output for real data } \boldsymbol{x}} + \underbrace{\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})} \log\left(1 - D_{\theta_d}\left(G_{\theta_g}(\boldsymbol{z})\right)\right)}_{\text{Discriminator output for generated fake data } G_{\theta_g}(\boldsymbol{z})} \right]$$

GAN: How to Learn

Objective:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{\boldsymbol{x} \sim p_{data}} \log D_{\theta_d}(\boldsymbol{x}) + \mathbb{E}_{\boldsymbol{z} \sim p_z(\boldsymbol{z})} \log\left(1 - D_{\theta_d}\left(G_{\theta_g}(\boldsymbol{z})\right)\right) \right]$$

Alternate between:

- Gradient ascent on discriminator:

$$\max_{\theta_d} \left[\mathbb{E}_{\boldsymbol{x} \sim p_{data}} \log D_{\theta_d}(\boldsymbol{x}) + \mathbb{E}_{\boldsymbol{z} \sim p_z(\boldsymbol{z})} \log\left(1 - D_{\theta_d}\left(G_{\theta_g}(\boldsymbol{z})\right)\right) \right]$$

- Gradient descent on generator:

$$\min_{\theta_g} \left[\mathbb{E}_{\boldsymbol{z} \sim p_z(\boldsymbol{z})} \log\left(1 - D_{\theta_d}\left(G_{\theta_g}(\boldsymbol{z})\right)\right) \right]$$

GAN: How to Learn

- After several steps of training, if G and D have enough capacity, they will reach a point at which both cannot improve because $p_g = p_{data}$.
  - Generator can generate real image.
  - Discriminator is unable to differentiate between the two distributions, i.e. $D_{\theta_d}(\boldsymbol{x}) = 1/2$.

discriminative distribution

data distribution

generative distribution

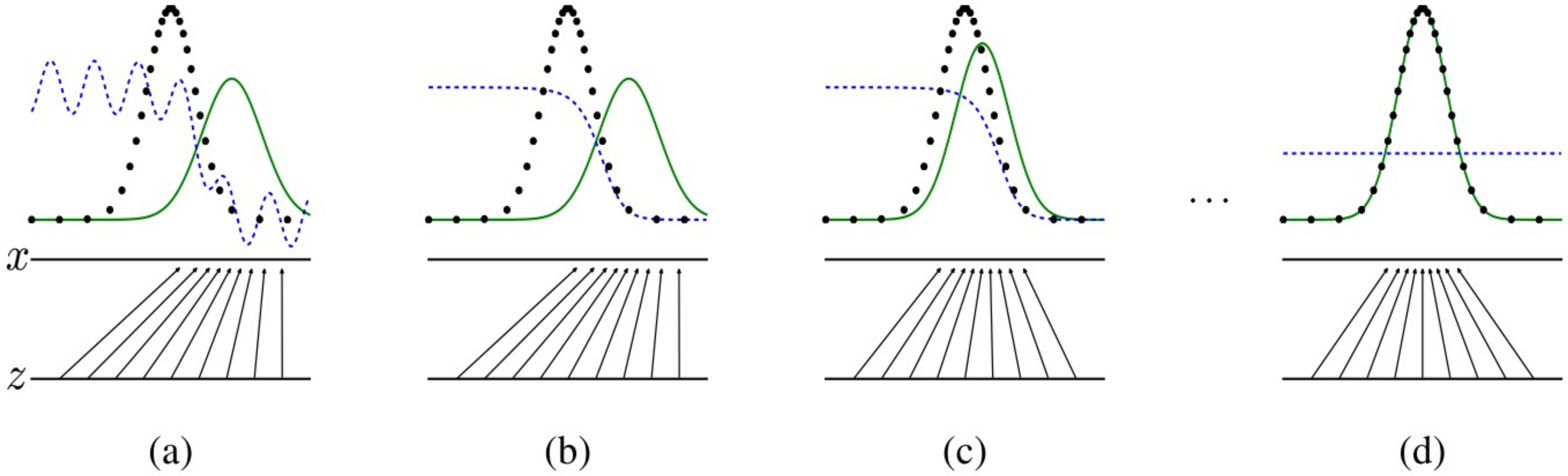


Image source: Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets." In Advances in neural information processing systems, pp. 2672-2680. 2014.

GAN: Algorithm

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

end for

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

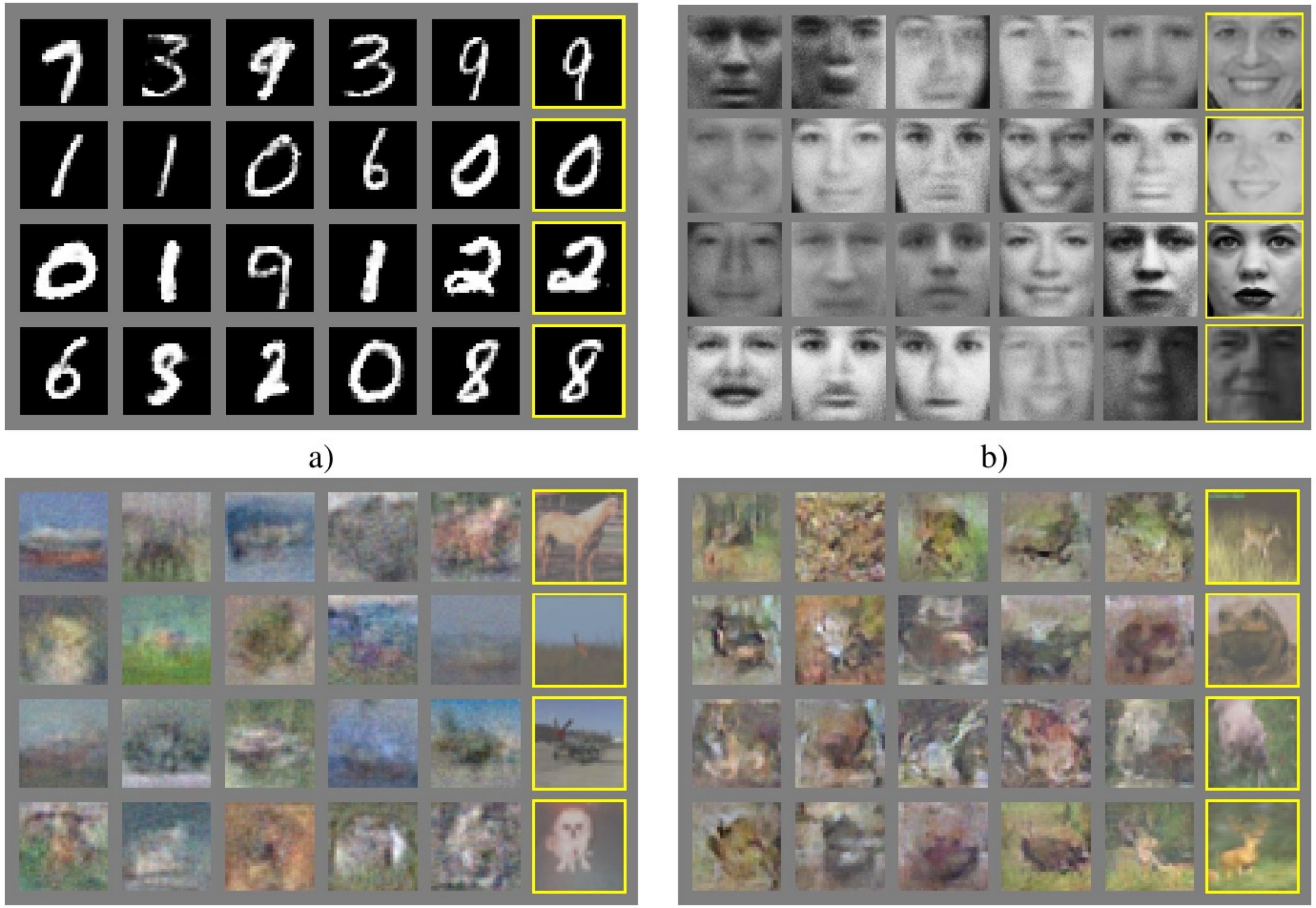
$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Image source: Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets." In Advances in neural information processing systems, pp. 2672-2680. 2014.

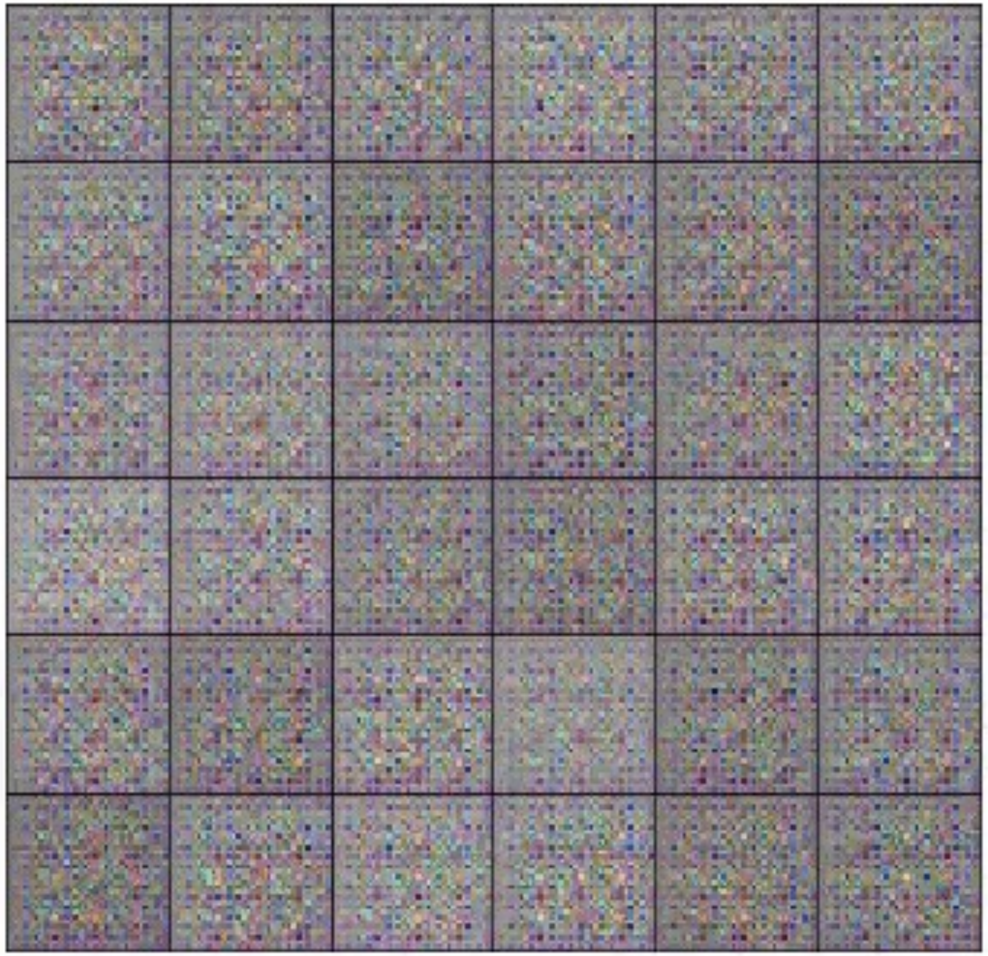
GAN: Result



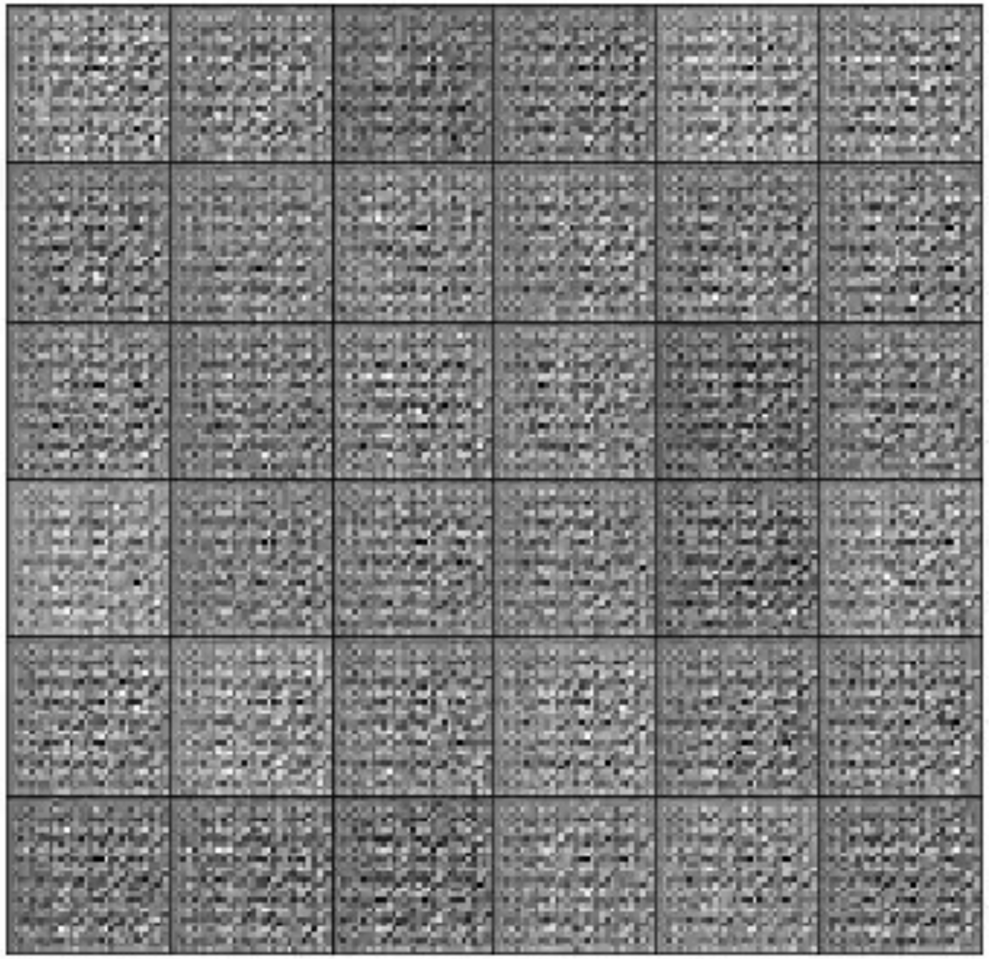
Rightmost column shows the nearest training example of the neighboring sample,
in order to demonstrate that the model has not memorized the training set.

Image source: Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets." In Advances in neural information processing systems, pp. 2672-2680. 2014.

GAN: Result



SVHNs



MNIST

Image source: https://sthalles.github.io/intro-to-gans/

GAN Starts an Era

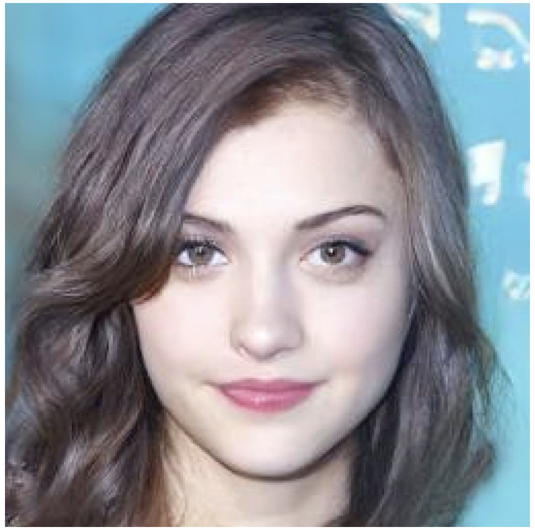

2014


2015


2016


2017


2018


2020


2023

Image source: Ian Goodfellow. Samples from Goodfellow et al., 2014, Radford et al., 2015, Liu et al., 2016, Karras et al., 2017, Karras et al., 2018, Ho et al. 2020, Zhang et al. 2023

DCGAN

Unsupervised representation learning with deep convolutional generative adversarial networks
A Radford, L Metz, S Chintala - arXiv preprint arXiv:1511.06434, 2015 - arxiv.org
… **REPRESENTATION LEARNING** FROM UNLABELED DATA **Unsupervised representation learning** … A classic approach to **unsupervised representation learning** is to do clustering on the …

- Vanilla GAN simply uses MLP, rather than CNN in both generator and discriminator.
- CNN can be easily applied to discriminator.
- Now the problem is: how can CNN be used as a generator?
  - Pooling leads to downsampling, how to upsampling?

Fractionally-Strided Convolutions

Input: 2×2

1	2
3	4

×

Filter: 3×3

10	20	30
40	50	60
30	20	10

=

Output: 4×4

10	20	30	
40	50	60	
30	20	10	

Fractionally-Strided Convolutions

Input: 2×2

1	2
3	4

×

Filter: 3×3

10	20	30
40	50	60
30	20	10

=

Output: 4×4

10	20+ 20	30+ 40	60
40	50+ 80	60+ 100	120
30	20+ 60	10+ 40	20

Fractionally-Strided Convolutions

Input: 2×2

1	2
3	4

×

Filter: 3×3

10	20	30
40	50	60
30	20	10

=

Output: 4×4

10	40	70	60
40+30	130+60	160+90	120
30+120	80+150	50+180	20
90	60	30	

Fractionally-Strided Convolutions

Input: 2×2

1	2
3	4

×

Filter: 3×3

10	20	30
40	50	60
30	20	10

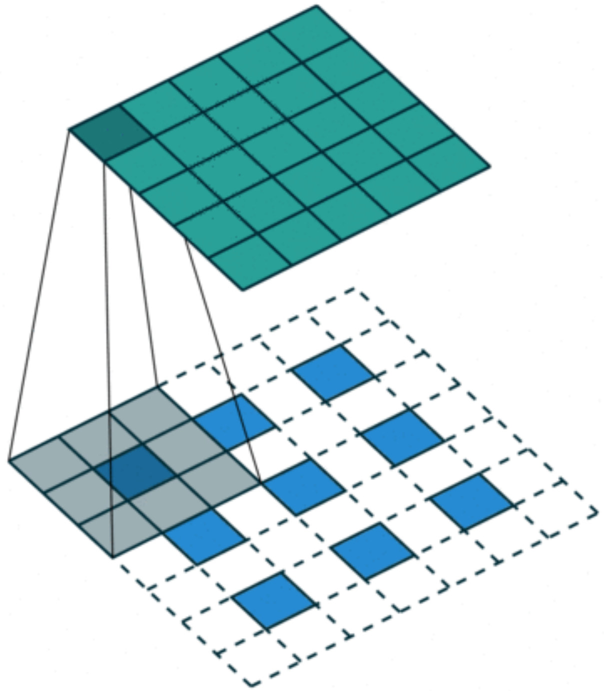
=

Output: 4×4

10	40	70	60
70	190 +40	250 +80	120+ 120
150	230+ 160	230+ 200	20+ 240
90	60+ 120	30+ 80	40

Fractionally-Strided Convolutions

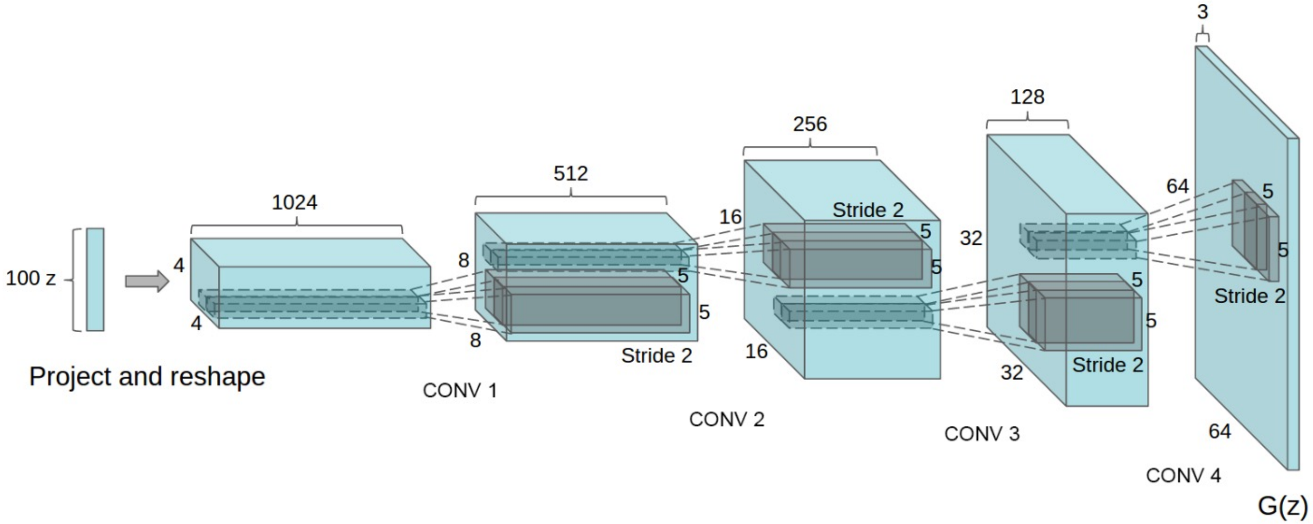
- Fractionally-strided convolutions are also called transposed convolutions.
  - PyTorch: `torch.nn.ConvTranspose2d`.
  - TensorFlow: `tf.keras.layers.Conv2DTranspose`.
- Some researchers are used to call deconvolutions. However, true deconvolutions are the inverse operation of convolution, which is not the same as fractionally-strided convolutions.



Transposed convolution with stride is equivalent to convolving with zero-padding and inserting zeros.

Image source: https://github.com/vdumoulin/conv_arithmetic
More linear algebra details: https://arxiv.org/pdf/1603.07285.pdf

DCGAN



Architecture guidelines for stable Deep Convolutional GANs

- Replace any pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator).

stride=2 everywhere

- Use batchnorm in both the generator and the discriminator.
- Remove fully connected hidden layers for deeper architectures.
- Use ReLU activation in generator for all layers except for the output, which uses Tanh.
- Use LeakyReLU activation in the discriminator for all layers.

Image source: Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv preprint arXiv:1511.06434 (2015).

DCGAN: Visual Results

The generated bedrooms look very nice (at that time)!

Image source: Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv preprint arXiv:1511.06434 (2015).

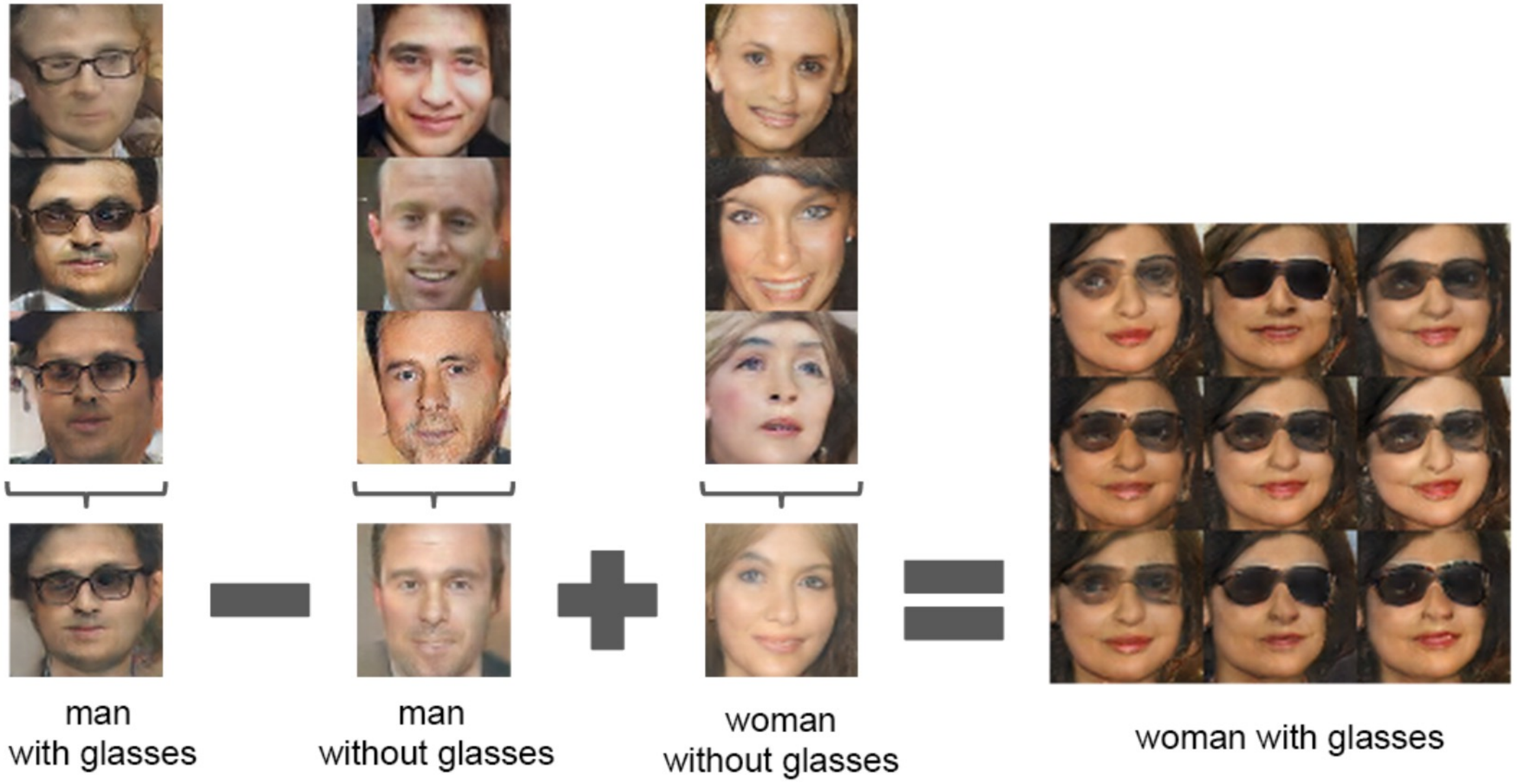
DCGAN: Walking in the Latent Space

- If walking in this latent space results in semantic changes to the image generations (such as objects being added and removed), we can reason that the model has learned relevant and interesting representations.

Interpolation between a series of 9 random points in Z show that the space learned has smooth transitions.

Image source: Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv preprint arXiv:1511.06434 (2015).

DCGAN: Vector Arithmetic



For each column, the Z vectors of samples are averaged. Arithmetic was then performed on the mean vectors creating a new vector Y.

Image source: Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv preprint arXiv:1511.06434 (2015).

DCGAN: Use as Feature Extractor

- Train on Imagenet-1k and then use the discriminator's convolutional features from all layers.
- Maxpooling each layers representation to produce a 4 × 4 spatial grid.
- These features are then flattened and concatenated to form a 28672 dimensional vector.
- A regularized linear L2-SVM classifier is trained on top of them.

Model	Accuracy	Accuracy (400 per class)	max # of features units
1 Layer K-means	80.6%	63.7% (±0.7%)	4800
3 Layer K-means Learned RF	82.0%	70.7% (±0.7%)	3200
View Invariant K-means	81.9%	72.6% (±0.7%)	6400
Exemplar CNN	84.3%	77.4% (±0.2%)	1024
DCGAN (ours) + L2-SVM	82.8%	73.8% (±0.4%)	512

CGAN

Conditional generative adversarial nets
M Mirza, S Osindero - arXiv preprint arXiv:1411.1784, 2014 - arxiv.org
… way to train **generative** models. In this work we introduce the **conditional** version of **generative**
… We show that this model can generate MNIST digits **conditioned** on class labels. We also …

- We can't control what we generate from the vanilla GAN.
  - Noise is the only input and it is totally random.
- How can we tell GAN what we want it to generate?
- Straightforward solution: replace data distribution by conditional distribution.

$$p(\boldsymbol{x}) \rightarrow p(\boldsymbol{x}|y).$$

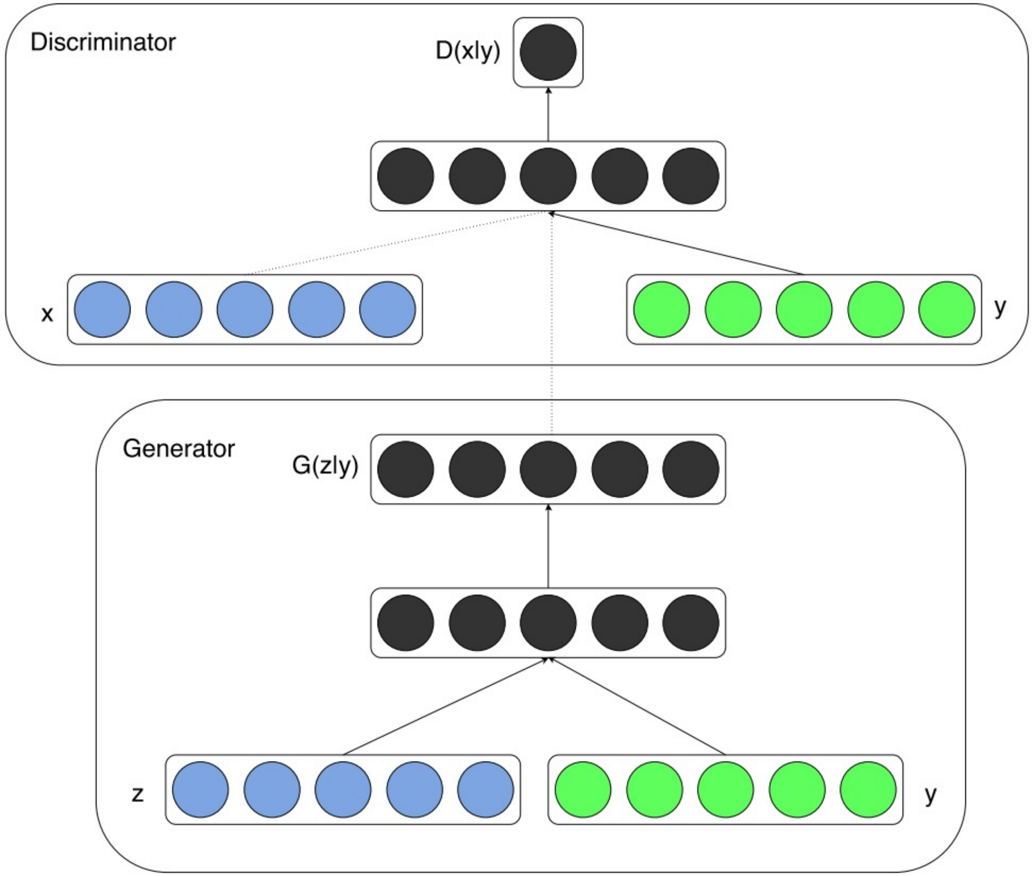
- Now, the problem becomes:
  - Generator: generate a sample for class y.
  - Discriminator: distinguish the real sample in class y and the generated sample in class y.

CGAN

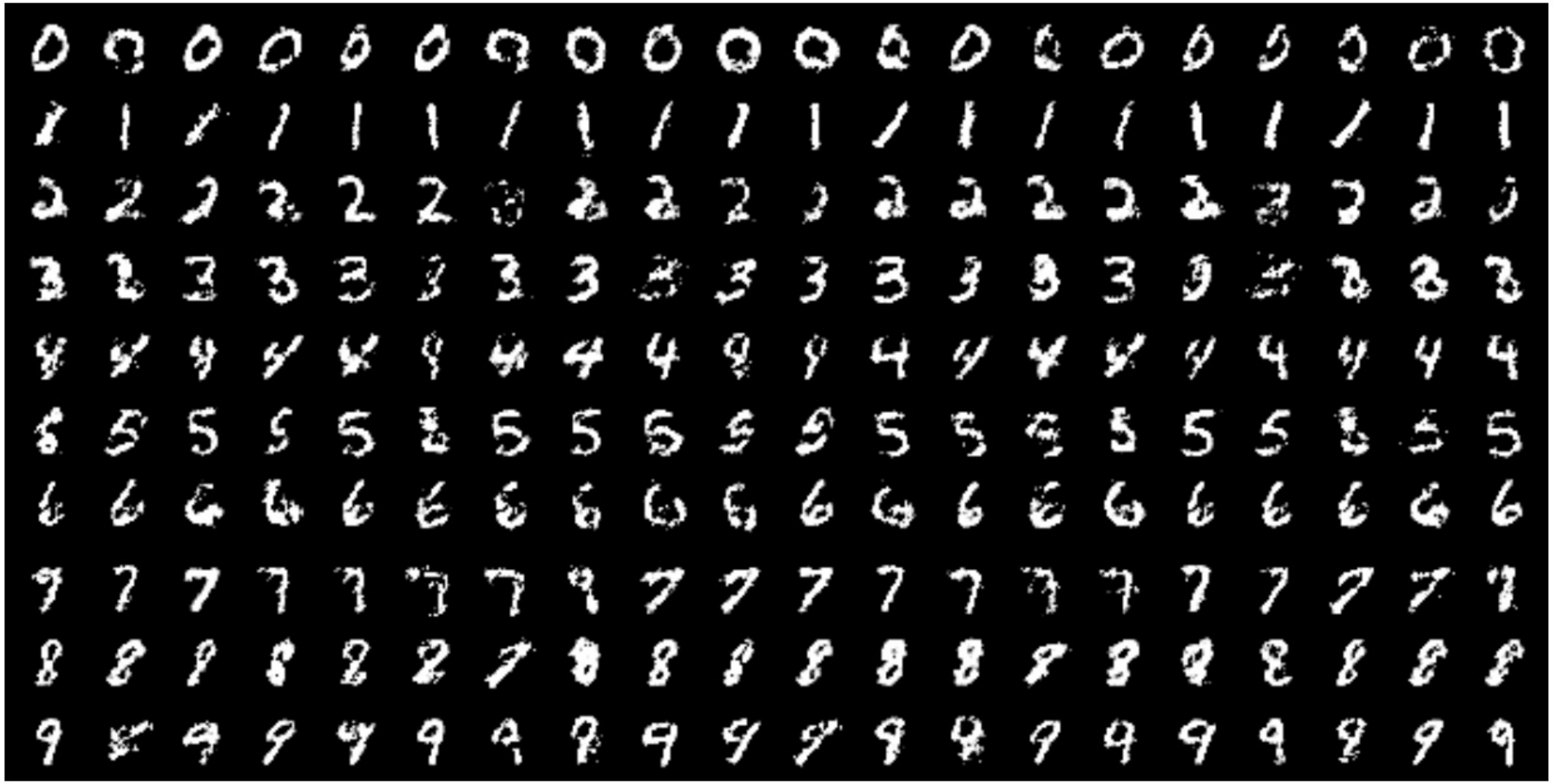
- Both generator and discriminator are conditioned on some extra information $\boldsymbol{y}$:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(\boldsymbol{x}|\boldsymbol{y}) + \mathbb{E}_{z \sim p(z)} \log \left(1 - D_{\theta_d}\left(G_{\theta_g}(\boldsymbol{z}|\boldsymbol{y}) \right) \right) \right].$$

- $\boldsymbol{y}$ could be any kind of auxiliary information, such as class labels or data from other modalities.
  - E.g. the speech of saying that class.



CGAN



Generate MNIST digits by directly feeding one-hot class label.

pix2pix

Image-to-image translation with conditional adversarial networks
P Isola, JY Zhu, T Zhou… - Proceedings of the IEEE …, 2017 - openaccess.thecvf.com
… In analogy to automatic language translation, we define automatic **image-to-image** translation as the problem of translating one possible representation of a scene into another, given …
☆ Save 99 Cite Cited by 20433 Related articles All 24 versions »

- Given a pair of images, transfer the style of one image to another.

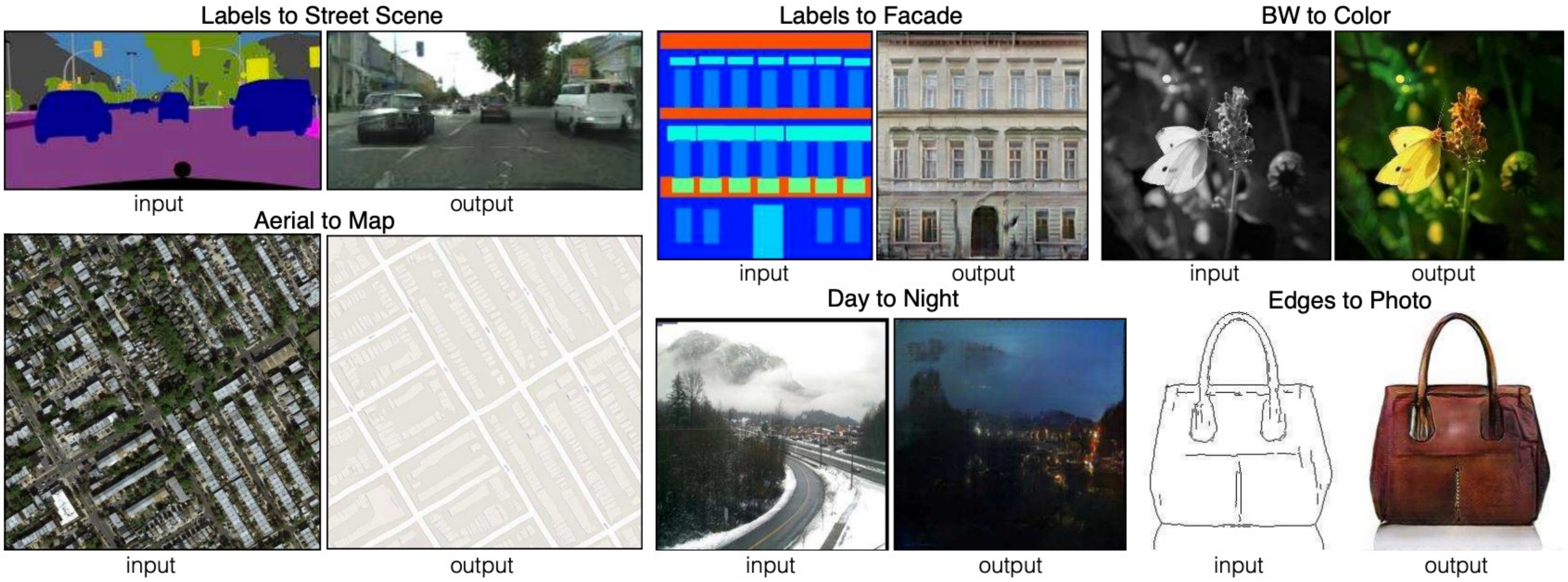


Image source: Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. "Image-to-image translation with conditional adversarial networks." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1125-1134. 2017.

pix2pix

- The model is based on CGAN.
- As an improvement, the generator is tasked to not only fool the discriminator but also to be near the ground truth output.
- L_1 penalization is added to the loss of CGAN to encourage less blurring:

$$L_{L_1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x,z)\|_1].$$

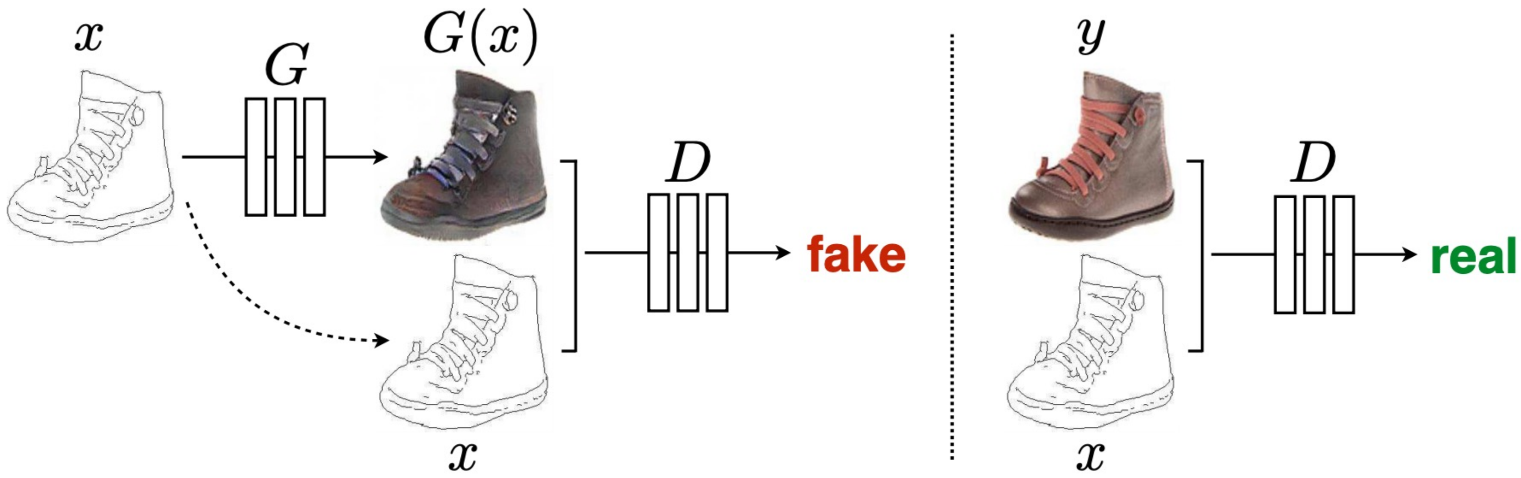


Image source: Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. "Image-to-image translation with conditional adversarial networks." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1125-1134. 2017.

pix2pix: More Applications

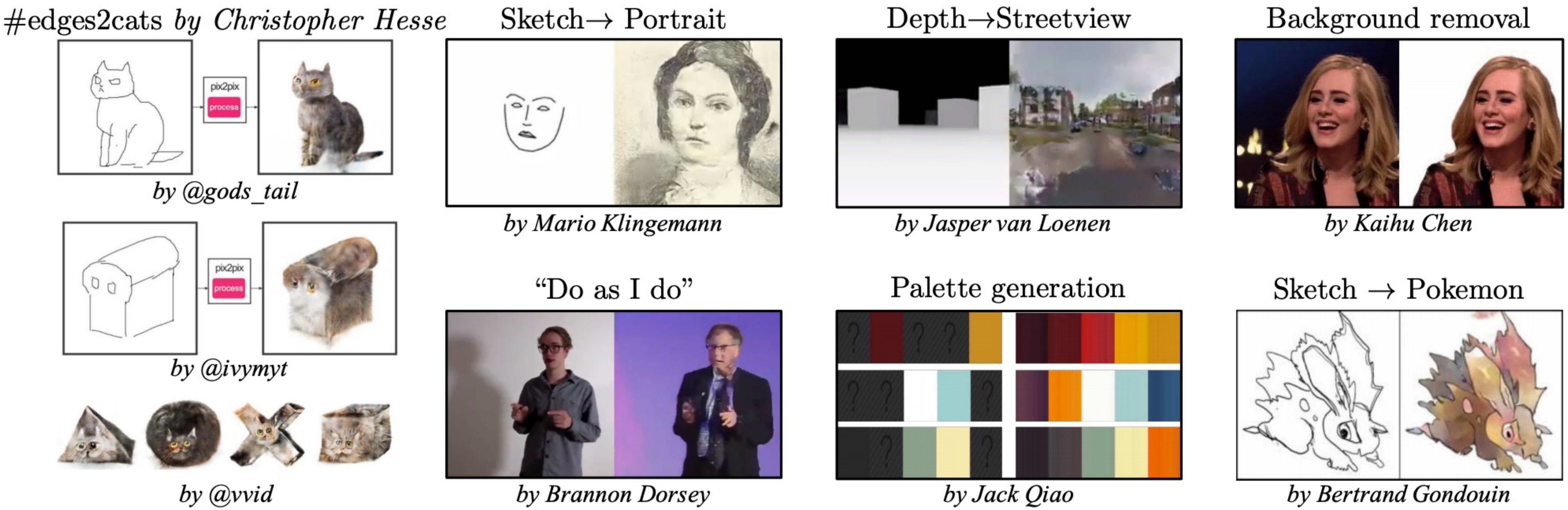
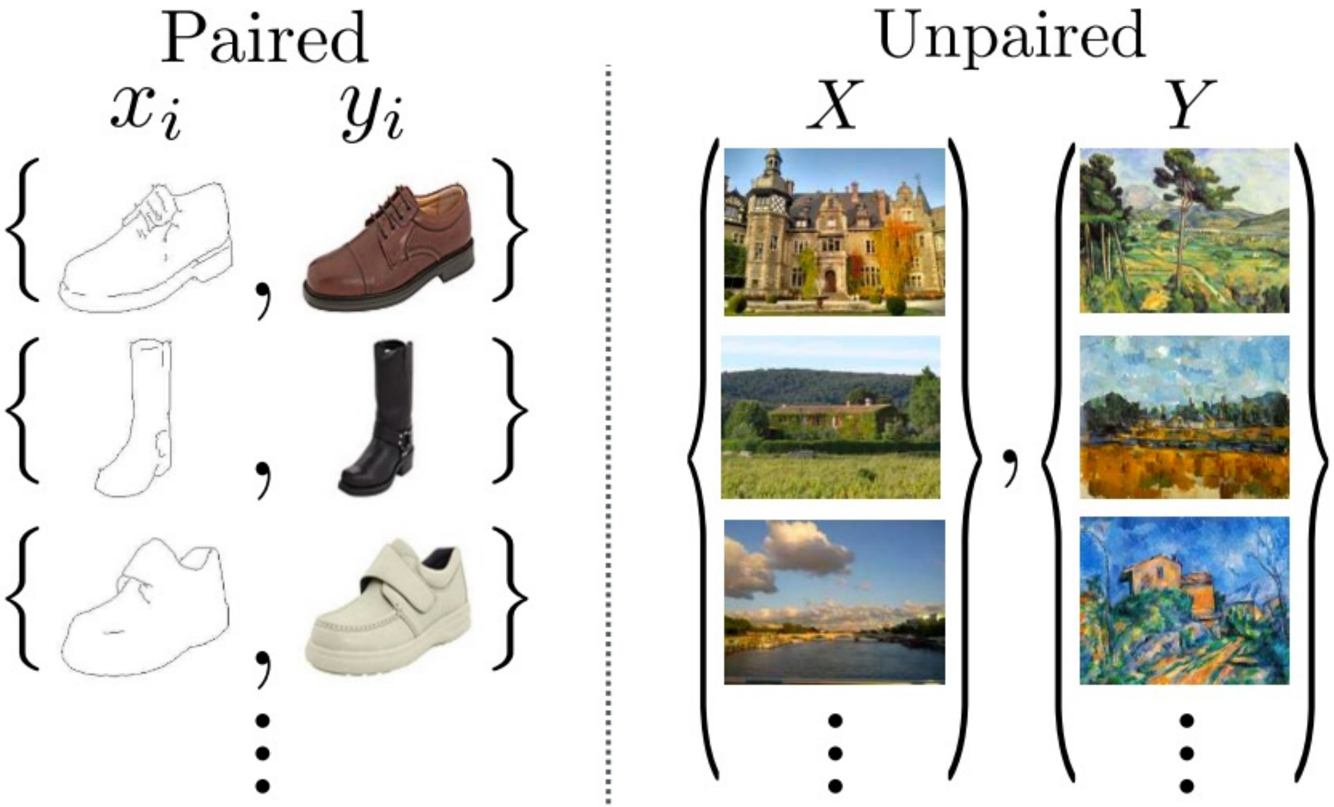


Image source: Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. "Image-to-image translation with conditional adversarial networks." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1125-1134. 2017.

CycleGAN

Unpaired image-to-image translation using cycle-consistent adversarial networks
JY Zhu, T Park, P Isola, AA Efros - Proceedings of the IEEE …, 2017 - openaccess.thecvf.com
… losses on domains X and Y yields our full objective for **unpaired image-to-image** translation. … We first compare our approach against recent methods for **unpaired image-to-image** …
☆ Save 99 Cite Cited by 20082 Related articles All 26 versions

- Paired examples can be expensive to obtain.
- Can we translate from $X \leftrightarrow Y$ in an unsupervised manner?



Paired vs. unpaired examples

CycleGAN

- Two generators:
  - $G: X \rightarrow Y$;
  - $F: Y \rightarrow X$.
- Two discriminators:
  - D_X aims to distinguish between images $\{x\}$ and translated images $\{F(y)\}$;
  - D_Y aims to discriminate between $\{y\}$ and $\{G(x)\}$.

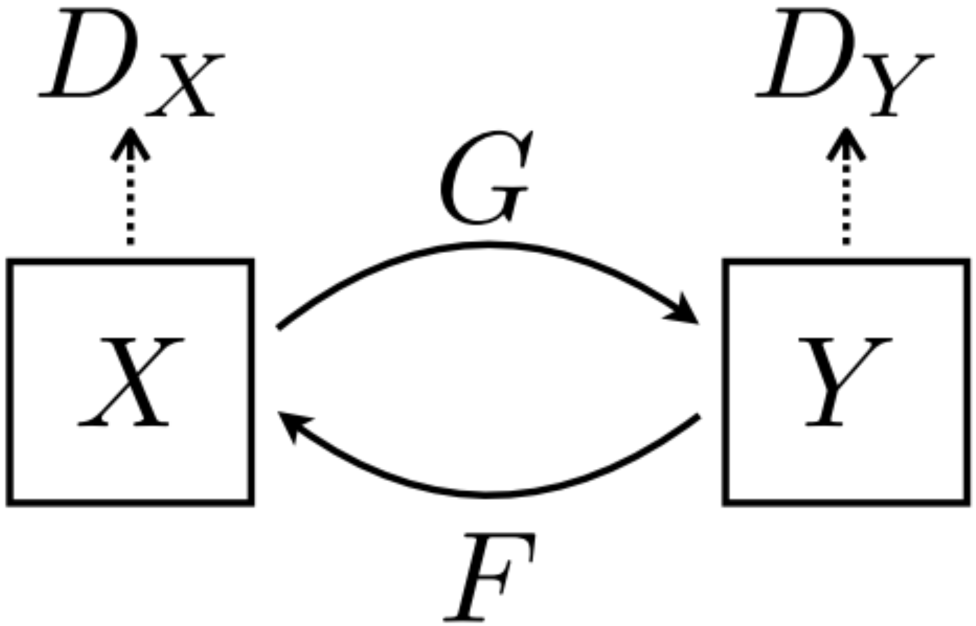


Image source: Zhu, Jun-Yan, Taesung Park, Phillip Isola, and Alexei A. Efros. "Unpaired image-to-image translation using cycle-consistent adversarial networks." In Proceedings of the IEEE international conference on computer vision, pp. 2223-2232. 2017.

CycleGAN

- If we can go from X to $\hat{Y}$ via G, then it should also be possible to go from $\hat{Y}$ back to X via F.
- Cycle consistency loss is added to the original adversarial loss:

$$L_{\text{cyc}}(G,F) = \mathbb{E}_x\left[\|F(G(x)) - x\|_1\right] + \mathbb{E}_y\left[\|G(F(x)) - y\|_1\right].$$

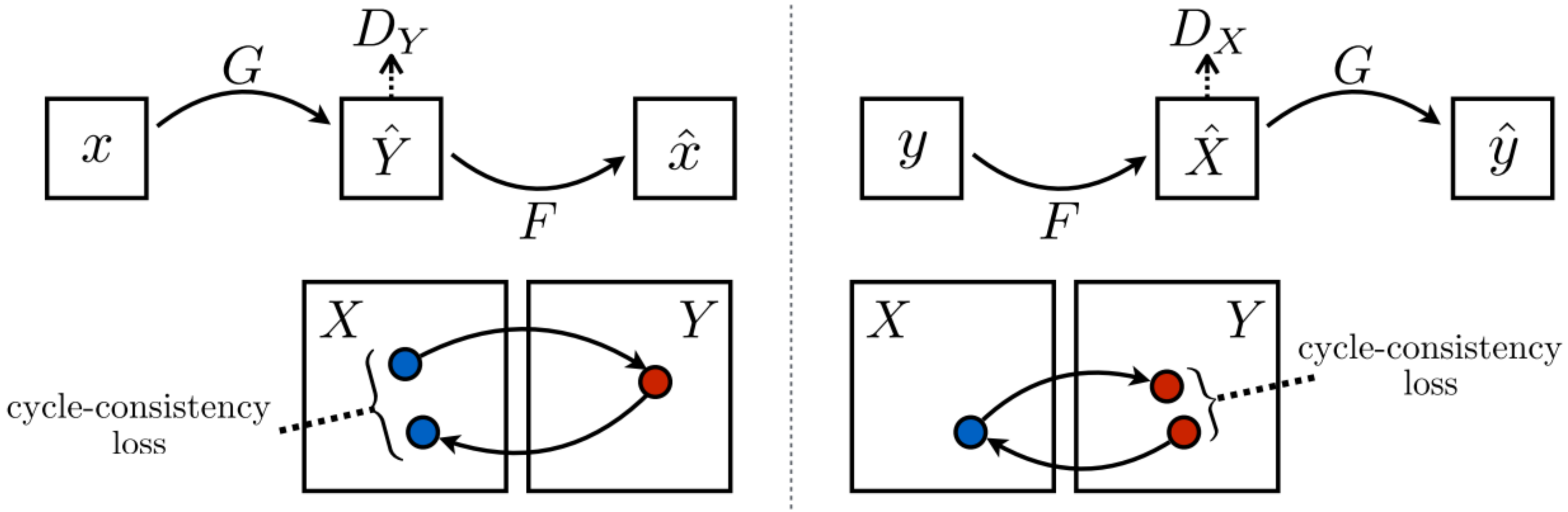


Image source: Zhu, Jun-Yan, Taesung Park, Phillip Isola, and Alexei A. Efros. "Unpaired image-to-image translation using cycle-consistent adversarial networks." In Proceedings of the IEEE international conference on computer vision, pp. 2223-2232. 2017.

CycleGAN

Failed case:

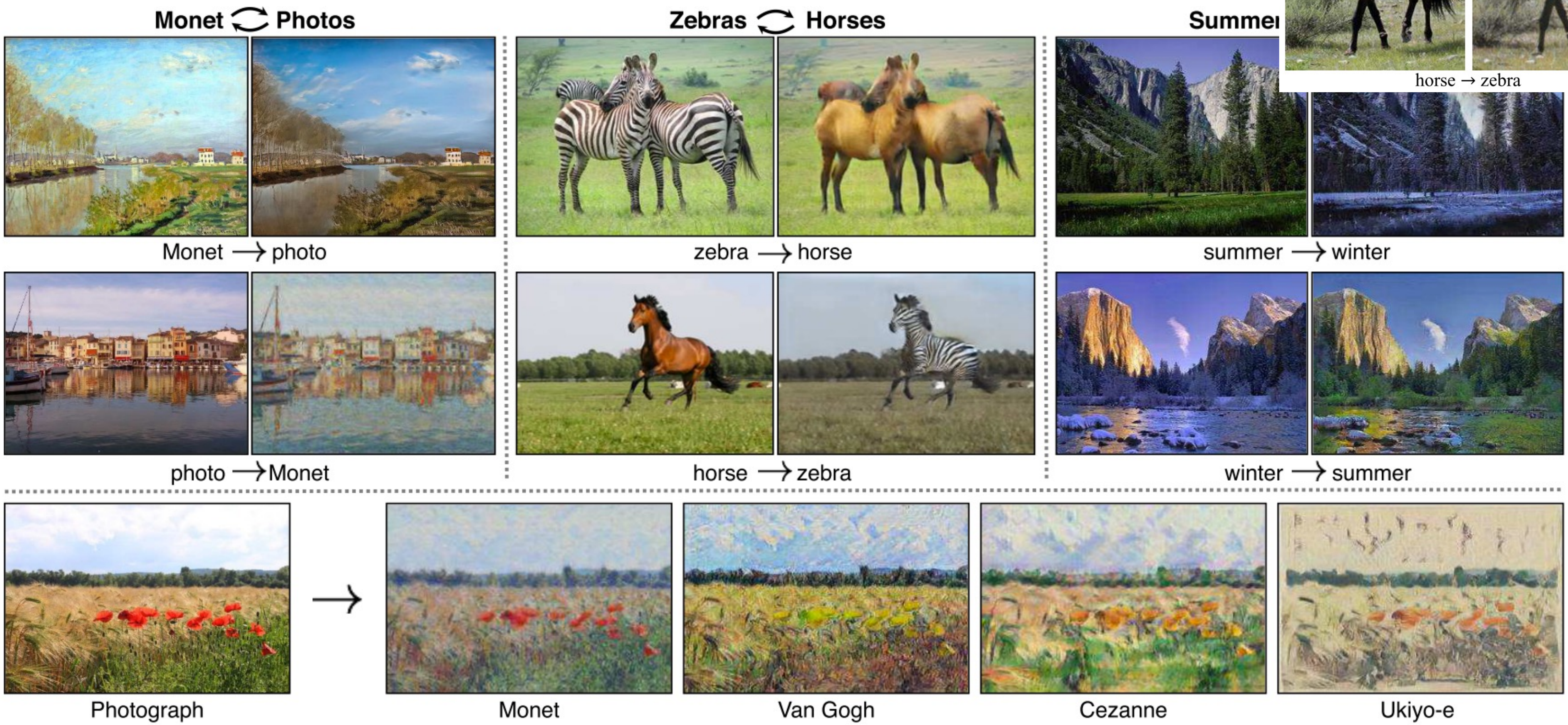
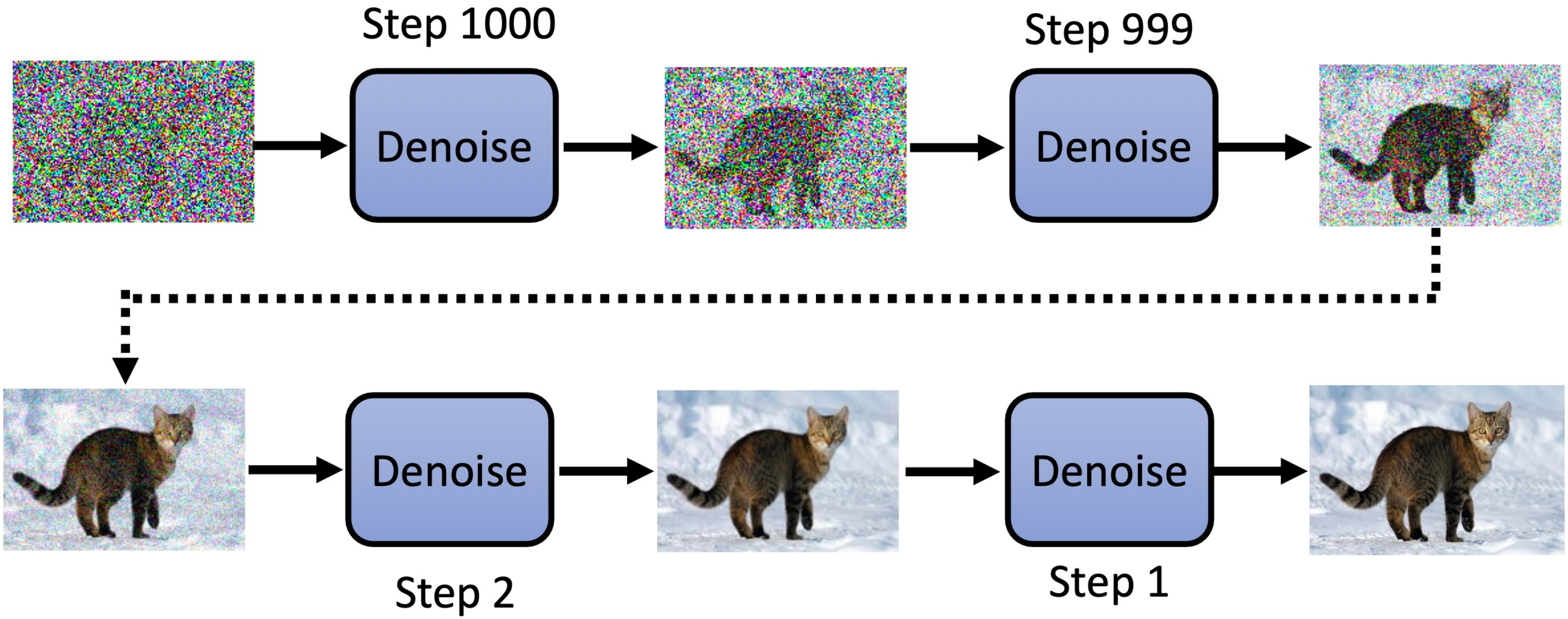


Image source: Zhu, Jun-Yan, Taesung Park, Phillip Isola, and Alexei A. Efros. "Unpaired image-to-image translation using cycle-consistent adversarial networks." In Proceedings of the IEEE international conference on computer vision, pp. 2223-2232. 2017.

DIFFUSION MODEL

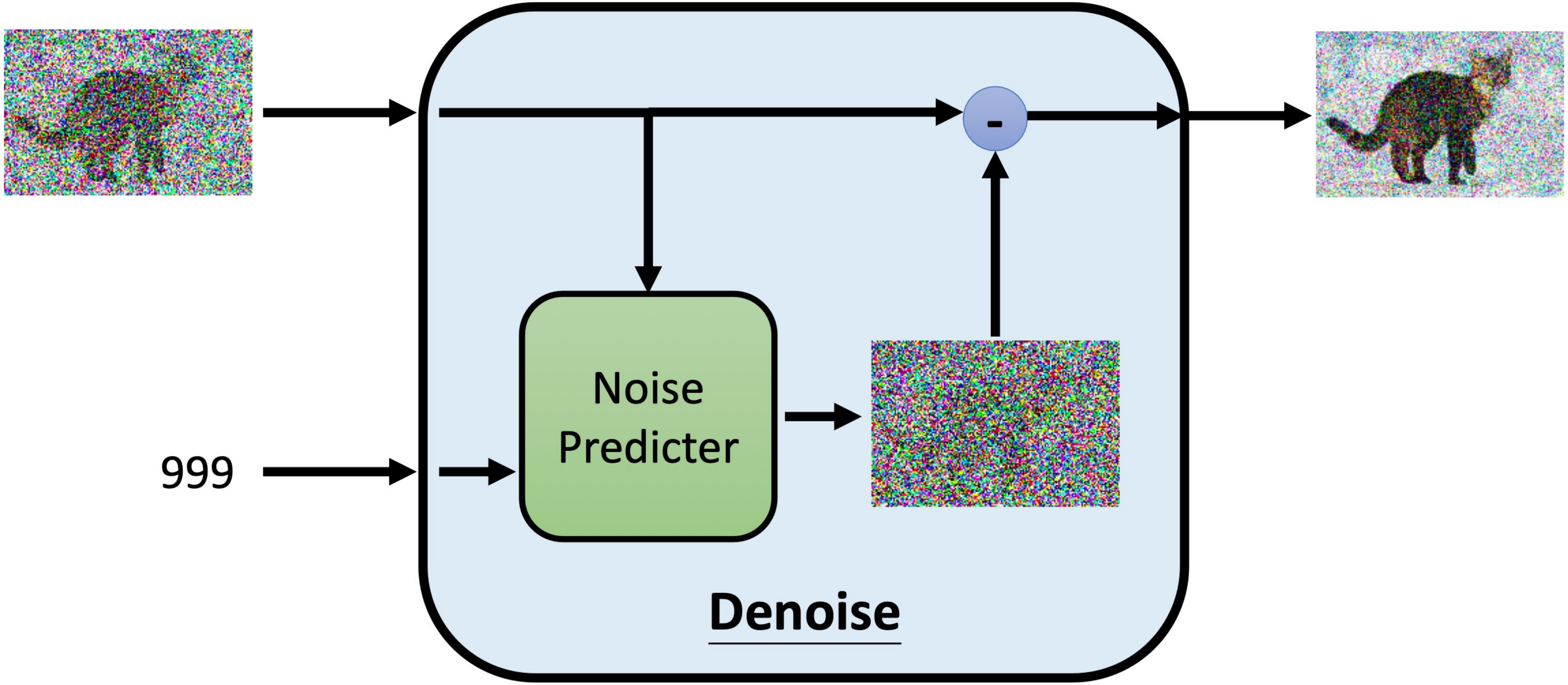
Diffusion Model

Denoising diffusion probabilistic models
J Ho, A Jain, P Abbeel - Advances in neural information …, 2020 - proceedings.neurips.cc
… This paper presents progress in **diffusion probabilistic models** [53]. A **diffusion probabilistic model** (which we will call a “**diffusion model**” for brevity) is a parameterized Markov chain …
☆ Save 99 Cite Cited by 4825 Related articles All 6 versions



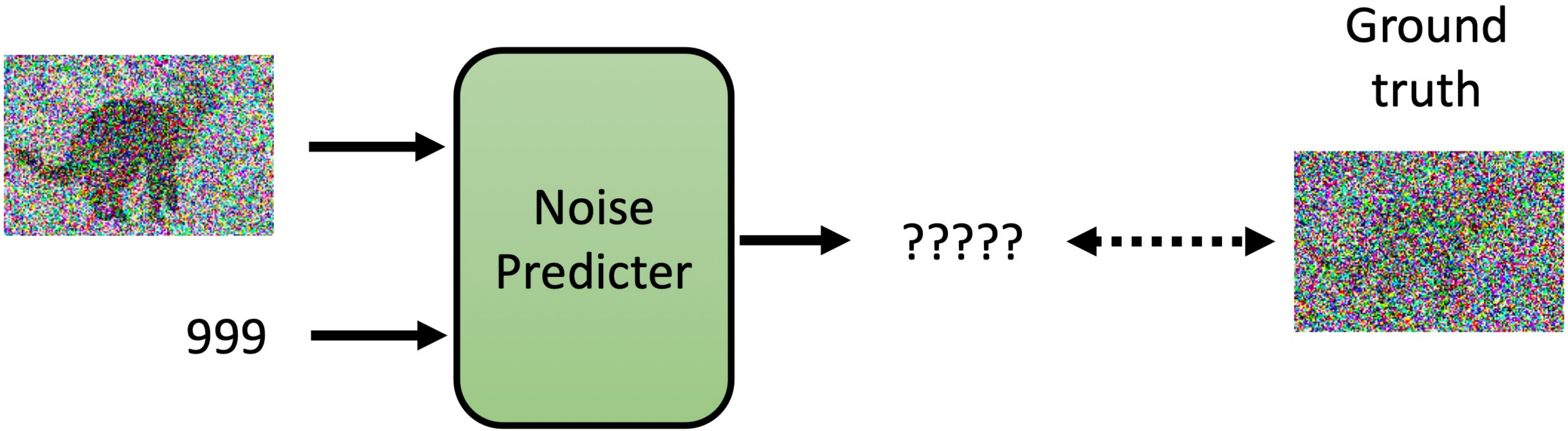
Reference: Hung-yi Lee, Diffusion model (https://speech.ee.ntu.edu.tw/~hylee/ml/ml2023-course-data/DiffusionModel%20(v2).pdf)

Diffusion Model



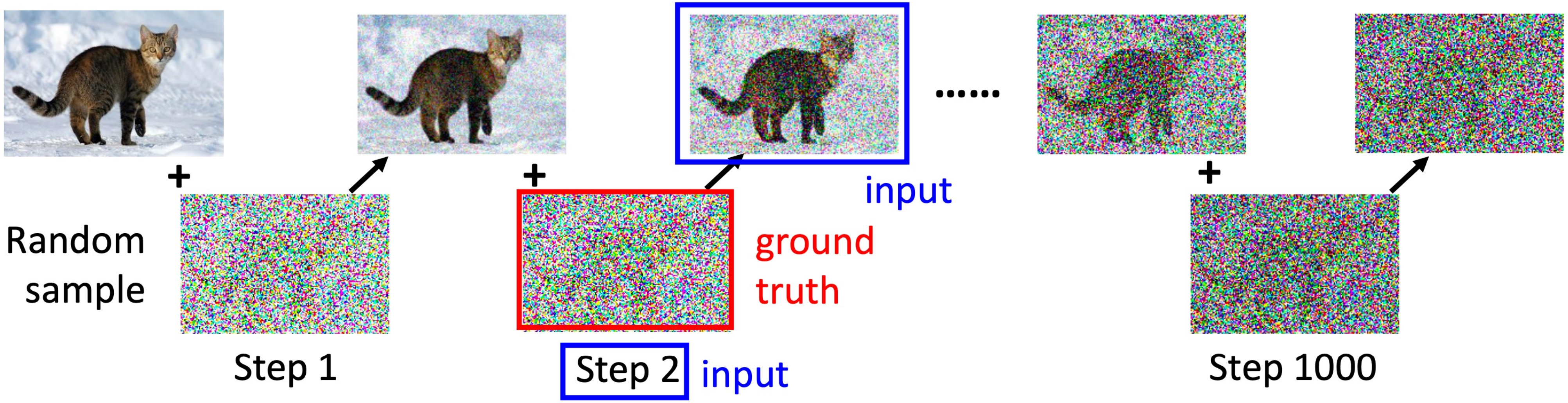
Reference: Hung-yi Lee, Diffusion model (https://speech.ee.ntu.edu.tw/~hylee/ml/ml2023-course-data/DiffusionModel%20(v2).pdf)

Diffusion Model

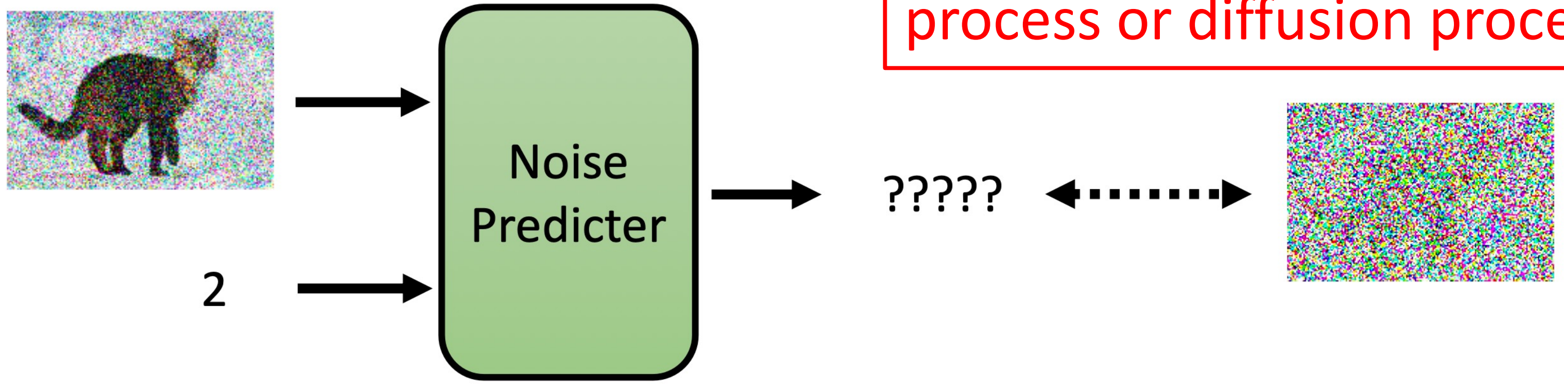


Reference: Hung-yi Lee, Diffusion model (https://speech.ee.ntu.edu.tw/~hylee/ml/ml2023-course-data/DiffusionModel%20(v2).pdf)

Diffusion Model

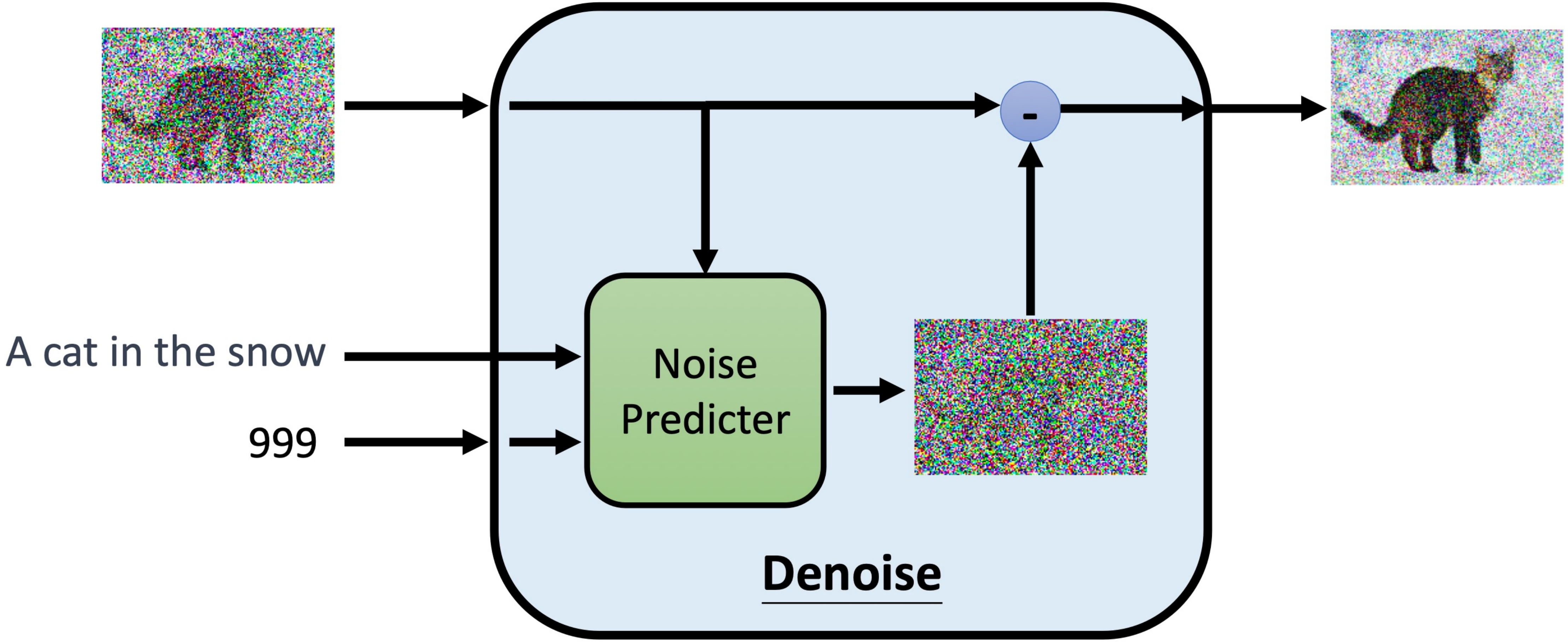


This is the process called the forward process or diffusion process



Reference: Hung-yi Lee, Diffusion model (https://speech.ee.ntu.edu.tw/~hylee/ml/ml2023-course-data/DiffusionModel%20(v2).pdf)

Diffusion Model



Reference: Hung-yi Lee, Diffusion model (https://speech.ee.ntu.edu.tw/~hylee/ml/ml2023-course-data/DiffusionModel%20(v2).pdf)

Diffusion Model

Algorithm 1 Training

1: **repeat**
2: $\quad \mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $\quad t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
4: $\quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: $\quad$ Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

Add ground truth noise

Ground truth noise

Denoise model

Algorithm 2 Sampling

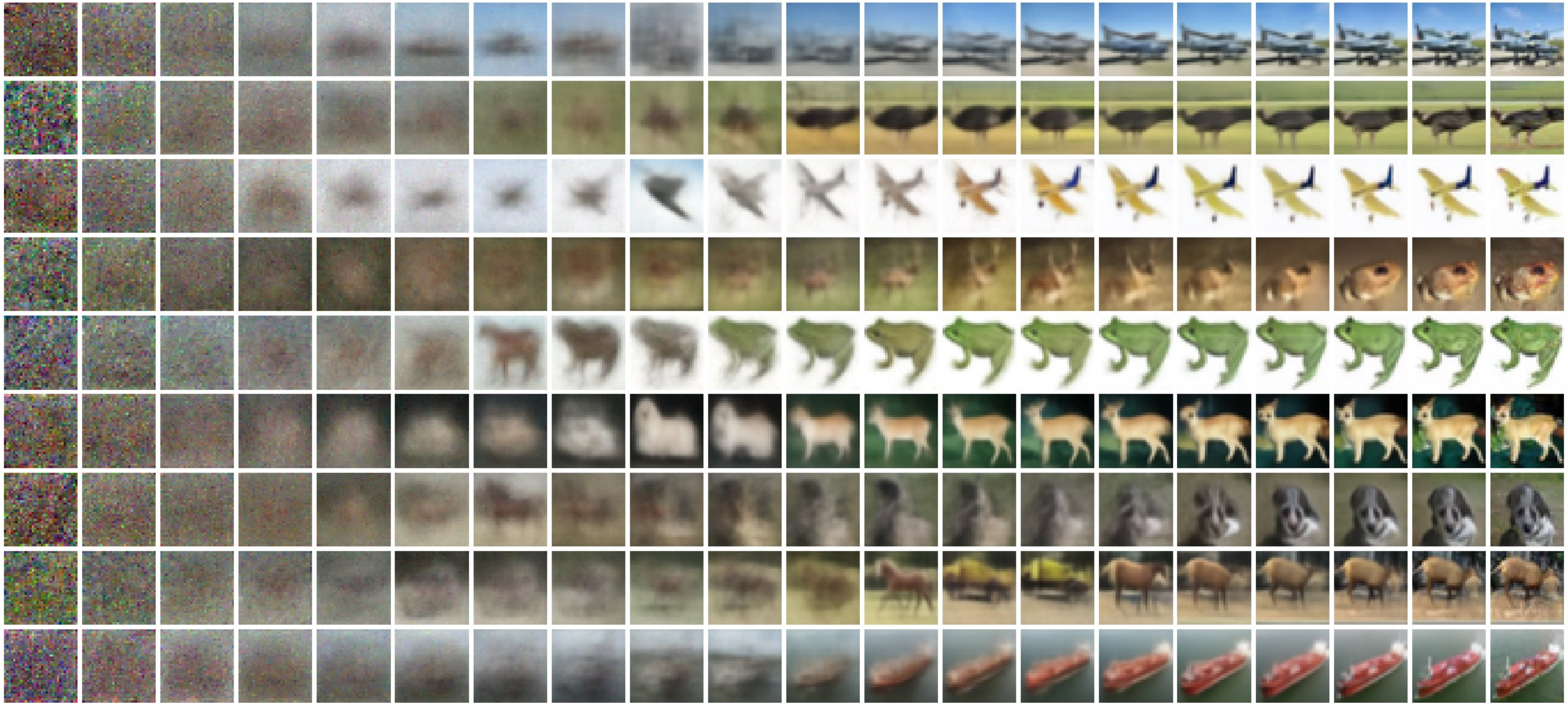
1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\quad \mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

Subtract predicted noise

Denoise model

Image source: Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

Diffusion Model



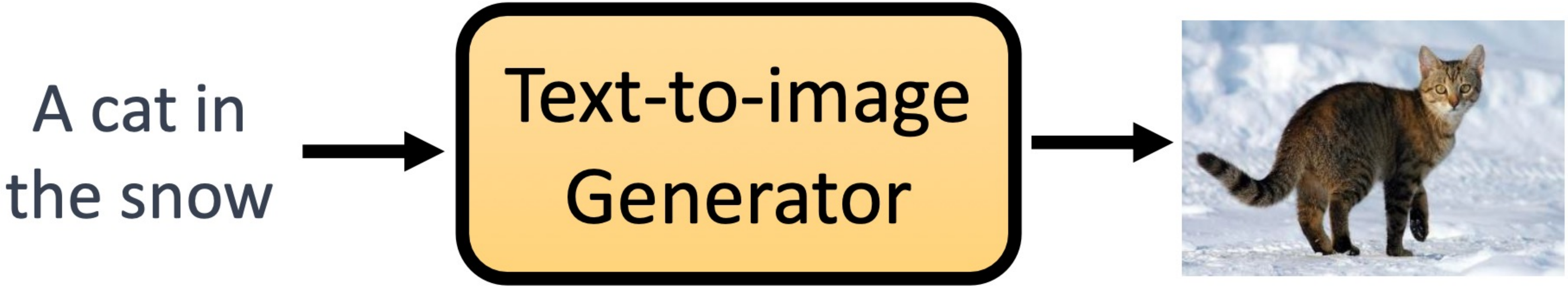
Unconditional CIFAR10 progressive generation

Image source: Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

Diffusion Model

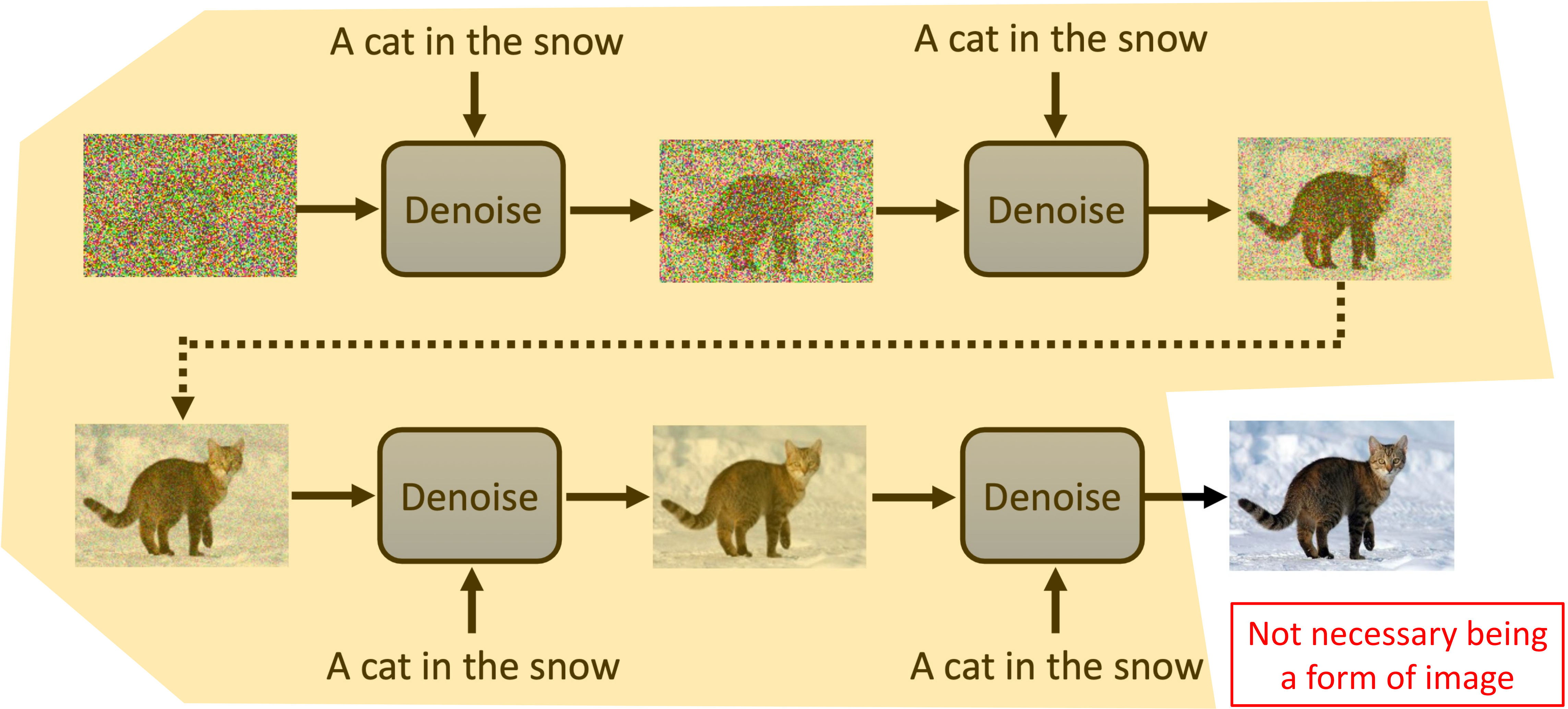
Image source: https://hojonathanho.github.io/diffusion/

- How can we generate desired image?



Reference: Hung-yi Lee, Diffusion model (https://speech.ee.ntu.edu.tw/~hylee/ml/ml2023-course-data/DiffusionModel%20(v2).pdf)

Diffusion Model

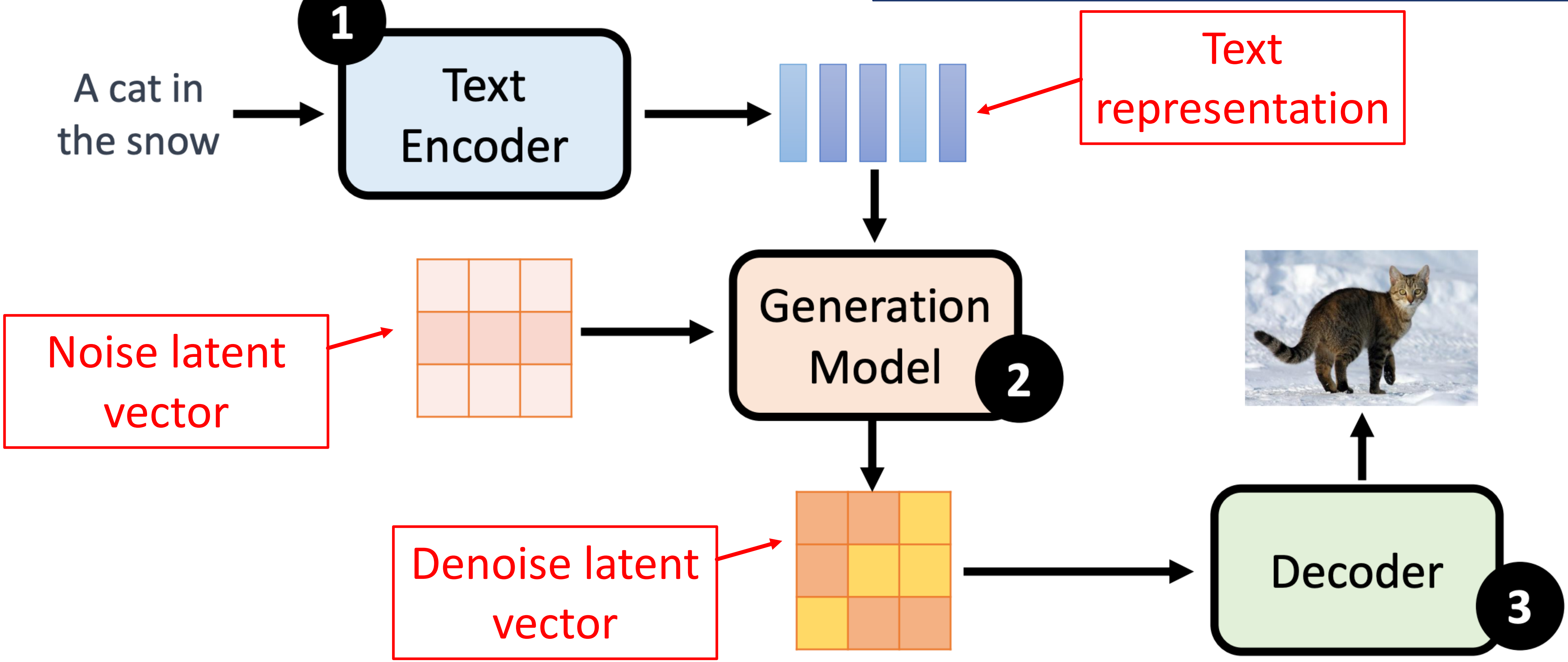


Not necessary being a form of image

Reference: Hung-yi Lee, Diffusion model (https://speech.ee.ntu.edu.tw/~hylee/ml/ml2023-course-data/DiffusionModel%20(v2).pdf)

Latent Diffusion

High-resolution image synthesis with latent diffusion models
R Rombach, A Blattmann, D Lorenz… - Proceedings of the …, 2022 - openaccess.thecvf.com
… To lower the computational demands of training diffusion models towards **high-resolution image synthesis**, we observe that although diffusion models allow to ignore perceptually …
Save Cite Cited by 3600 Related articles All 8 versions



Reference: Hung-yi Lee, Diffusion model (https://speech.ee.ntu.edu.tw/~hylee/ml/ml2023-course-data/DiffusionModel%20(v2).pdf)

Latent Diffusion

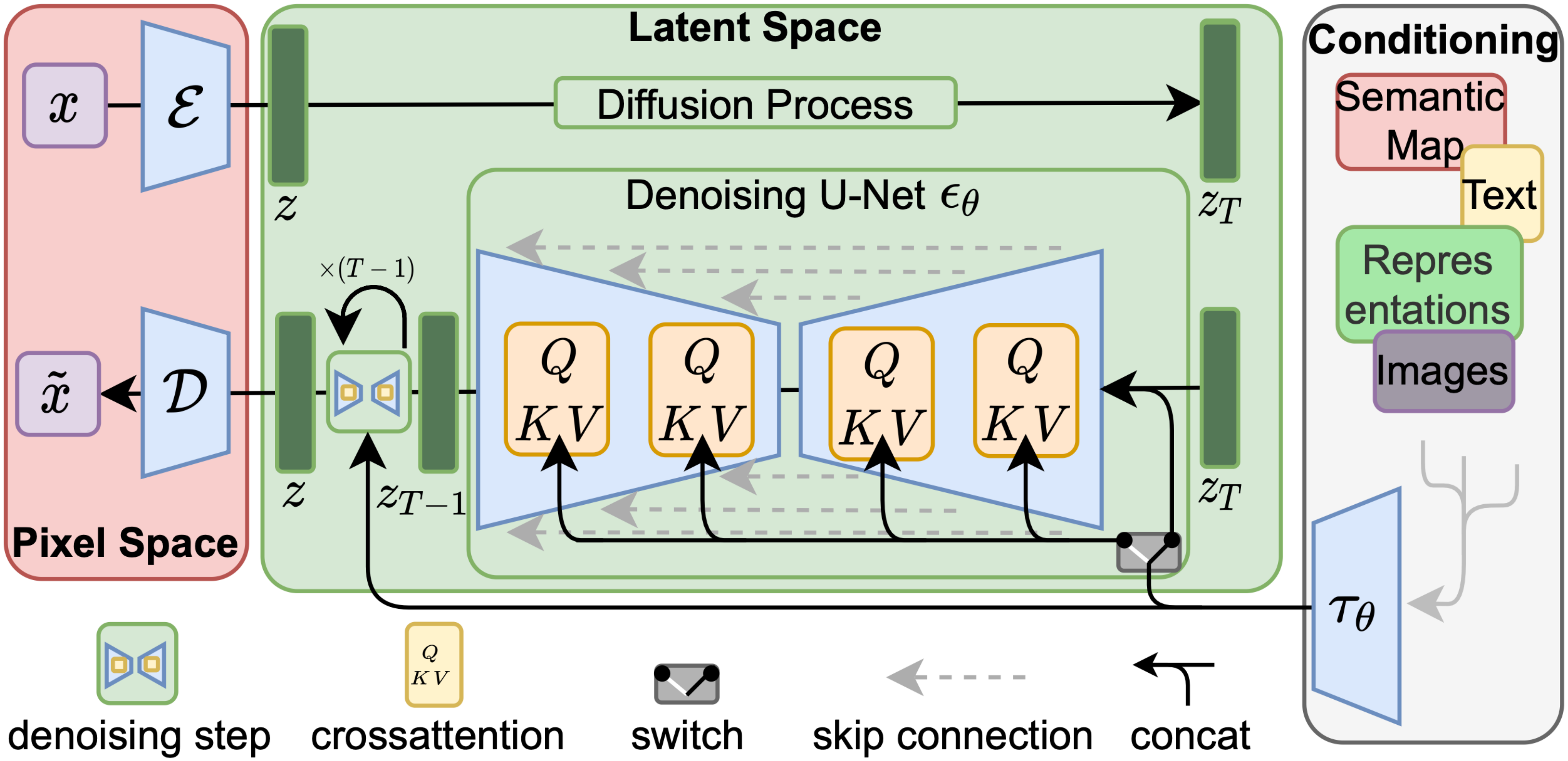
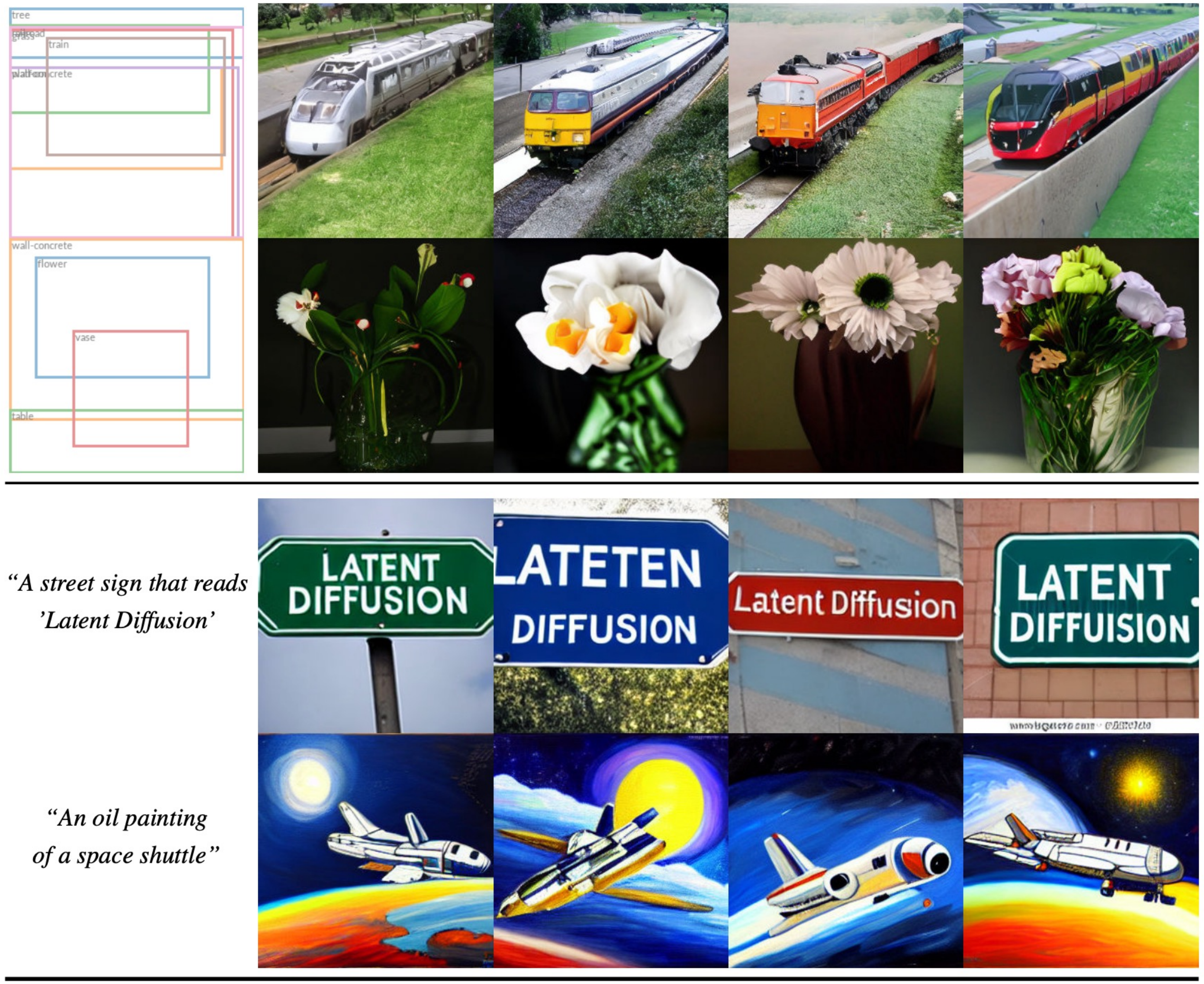


Image source: Rombach, Robin, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. "High-resolution image synthesis with latent diffusion models." In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10684-10695. 2022.

Latent Diffusion



Layout-to-image synthesis on COCO

Text-to-image *LDM* model for user-defined text prompts

Image source: Rombach, Robin, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. "High-resolution image synthesis with latent diffusion models." In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10684-10695. 2022.

Latent Diffusion

Semantic synthesis of landscape images

Image source: Rombach, Robin, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. "High-resolution image synthesis with latent diffusion models." In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10684-10695. 2022.

Latent Diffusion

ImageNet 64→256 super-resolution

Image source: Rombach, Robin, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. "High-resolution image synthesis with latent diffusion models." In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10684-10695. 2022.

Latent Diffusion

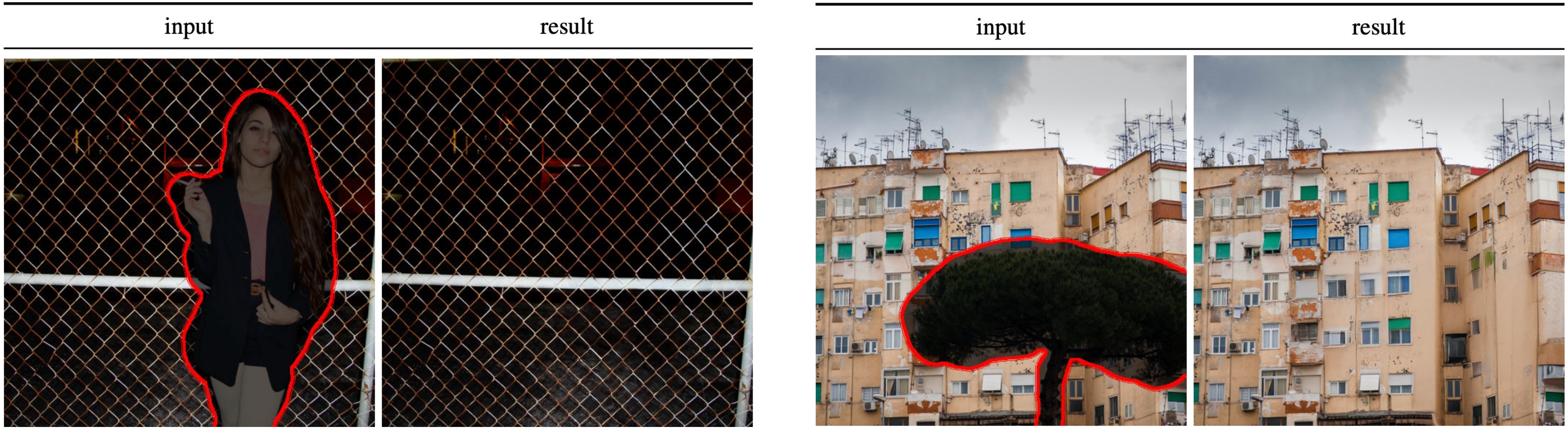
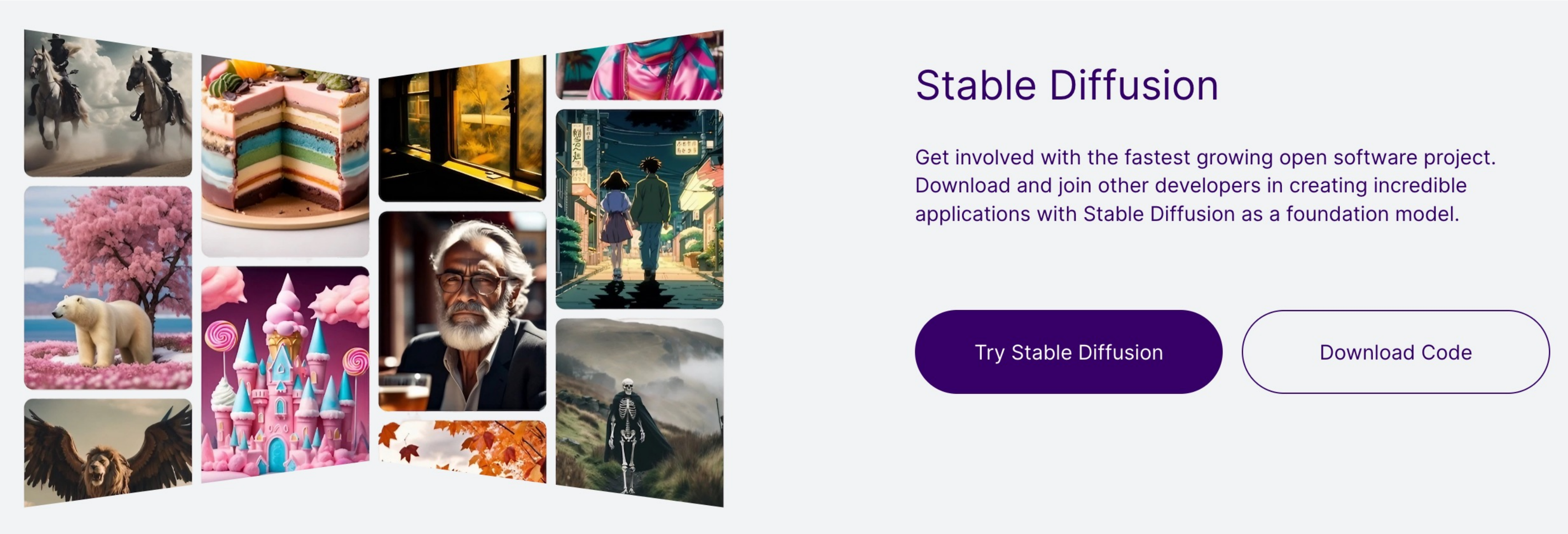


Image inpainting with latent diffusion

Image source: Rombach, Robin, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. "High-resolution image synthesis with latent diffusion models." In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10684-10695. 2022.

Stable Diffusion

- Stable diffusion is developed by researchers from the CompVis Group at Ludwig Maximilian University of Munich and Runway with a compute donation by Stability AI and training data from non-profit organizations.



DALL·E

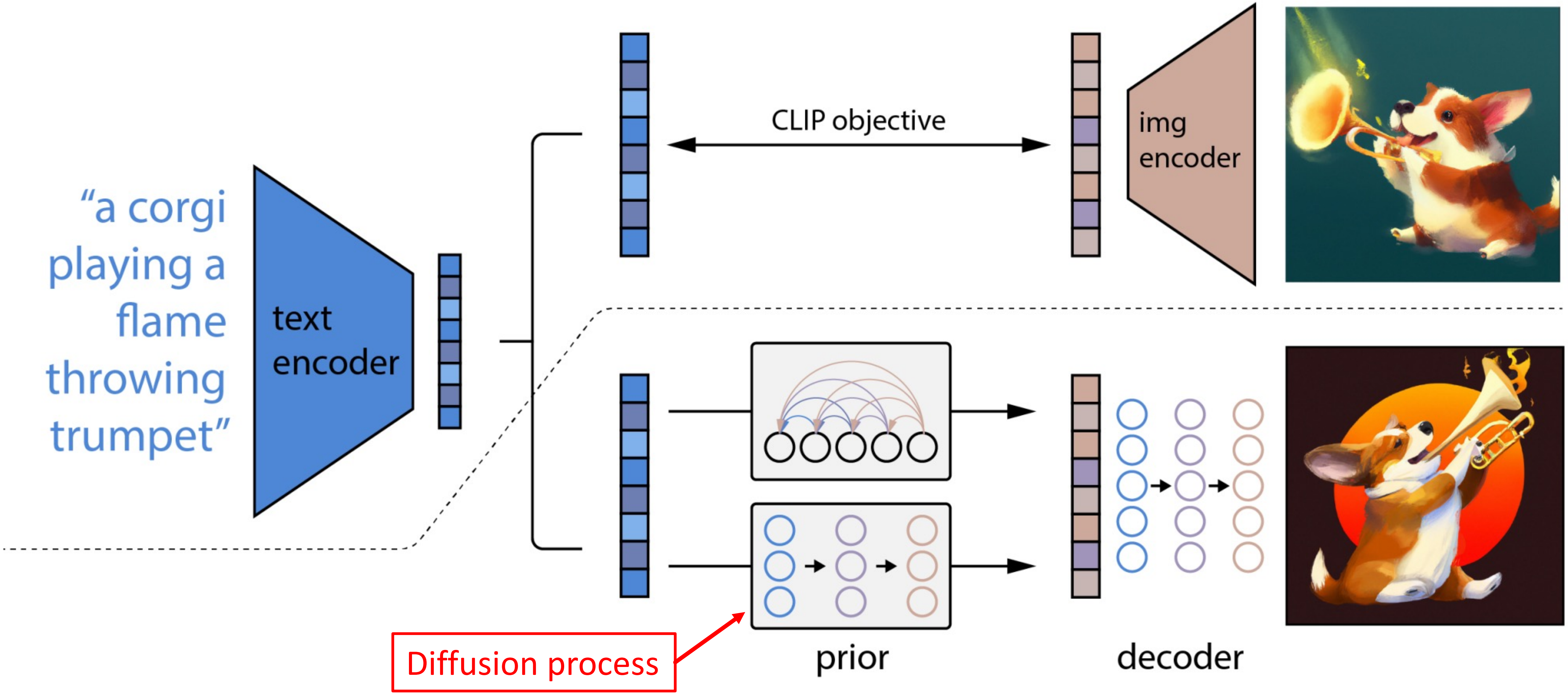
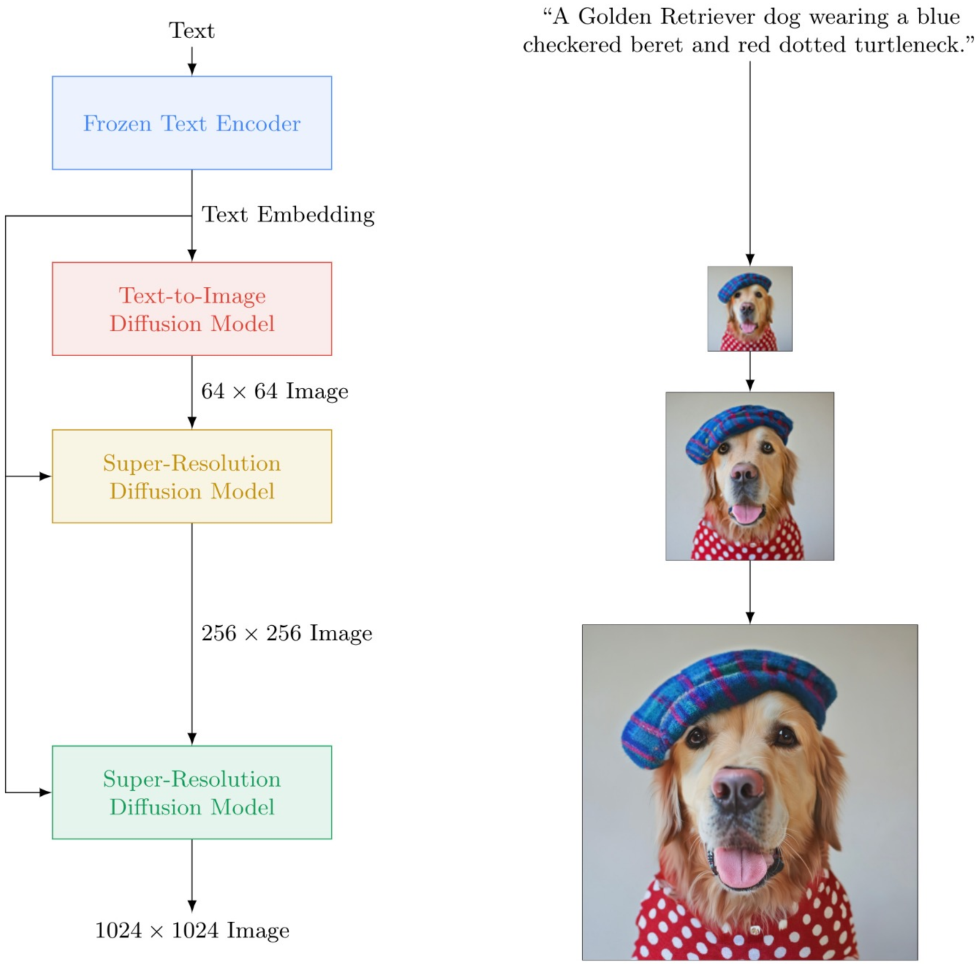


Image source: Ramesh, Aditya, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. "Hierarchical text-conditional image generation with clip latents." *arXiv preprint arXiv:2204.06125* 1, no. 2 (2022): 3.

Imagen



Sprouts in the shape of text ‘Imagen’ coming out of a fairytale book.

A photo of a Shiba Inu dog with a backpack riding a bike. It is wearing sunglasses and a beach hat.

A high contrast portrait of a very happy fuzzy panda dressed as a chef in a high end kitchen making dough. There is a painting of flowers on the wall behind him.

Teddy bears swimming at the Olympics 400m Butter-fly event.

A cute corgi lives in a house made out of sushi.

A cute sloth holding a small treasure chest. A bright golden glow is coming from the chest.

A brain riding a rocketship heading towards the moon.

A dragon fruit wearing karate belt in the snow.

A strawberry mug filled with white sesame seeds. The mug is floating in a dark chocolate sea.

Reference: https://imagen.research.google/

Midjourney

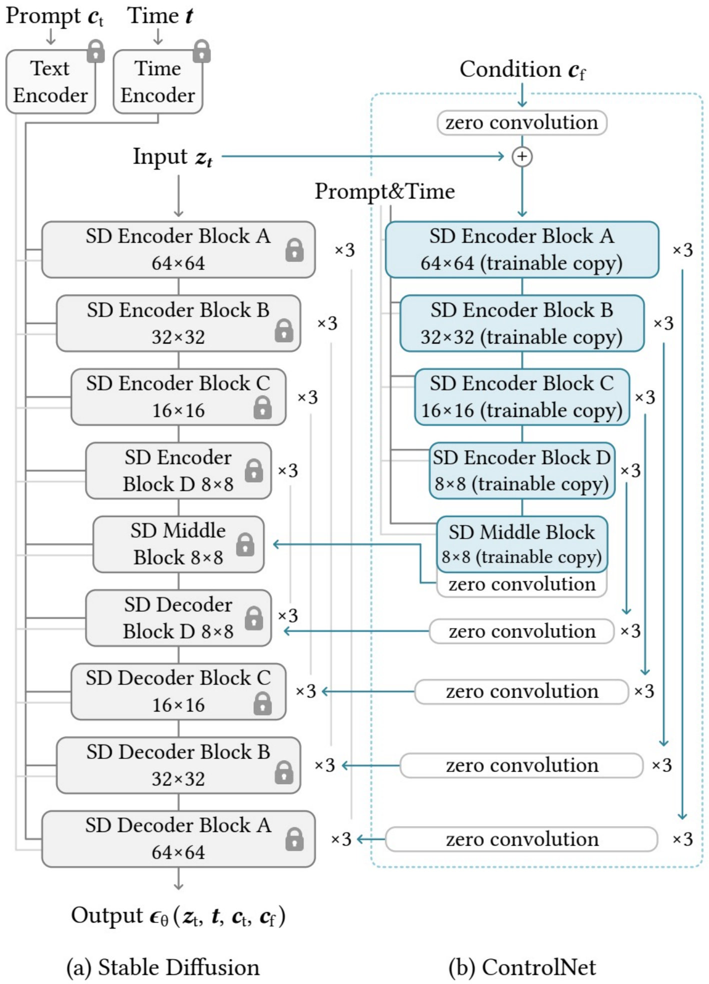
These fake images of the Pope and Donald Trump have gone viral in recent weeks, both were created on Midjourney v5.

Image source: https://petapixel.com/2023/03/30/midjourney-ends-free-trials-after-fake-ai-images-go-viral/

ControlNet

Adding conditional control to text-to-image diffusion models
L Zhang, A Rao, M Agrawala - Proceedings of the IEEE/CVF …, 2023 - openaccess.thecvf.com
… Learning **conditional controls** for large text-to-image diffusion … network architecture that learns **conditional controls** for large … for learning diverse **conditional controls**. The trainable copy …

- A new concept called zero convolution layers, with weights initialized to zeros so that they progressively grow during the training.
- This architecture ensures that harmful noise is not added to the deep features of the large diffusion model.



Source: Zhang, Lvmin, Anyi Rao, and Maneesh Agrawala. "Adding conditional control to text-to-image diffusion models." In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3836-3847. 2023.

Conclusion

After this lecture, you should know:

- What is a generative model?
- How generator and discriminator improve each other?
- How does transposed convolution work?
- How to design application specific loss to train with adversarial loss?
- What is the main principle of diffusion model?

Suggested Reading

- Adversarial Nets Papers
- Tips and tricks to make GANs work
- 令人拍案叫绝的Wasserstein GAN
- 李宏毅讲Diffusion Model

Assignment 4

- Assignment 4 is released. The deadline is **18:00, 11st December.**

Thank you!

- Any question?
- Don’t hesitate to send email to me for asking questions and discussion. ☺