Advanced Computer Vision

Tanveer Hussain
htanveer3797@gmail.com

جامعة الملك عبدالله
للعلوم والتقنية
King Abdullah University of
Science and Technology

KAUST Academy
King Abdullah University of Science and Technology

Course designed by **Naeem Ullah Khan** (naeemullah.khan@kaust.edu.sa), updated by Tanveer Hussain

Generative Adversarial Networks (GANs)

- Variational Autoencoders are based on maximizing likelihood or approximations

$$p_\theta(x) = p(x_\theta \mid z)\, p_\theta(z)$$

- What if we give up on explicitly modeling density, and just want ability to sample?
- **GANs**: don't work with any explicit density function!

  Instead, take game-theoretic approach: learn to generate from training distribution through 2-player game
  - GANs do not explicitly define or model the probability density function of the data. Instead, they implicitly learn to generate samples that follow the same distribution as the training data without explicitly calculating the probability density.

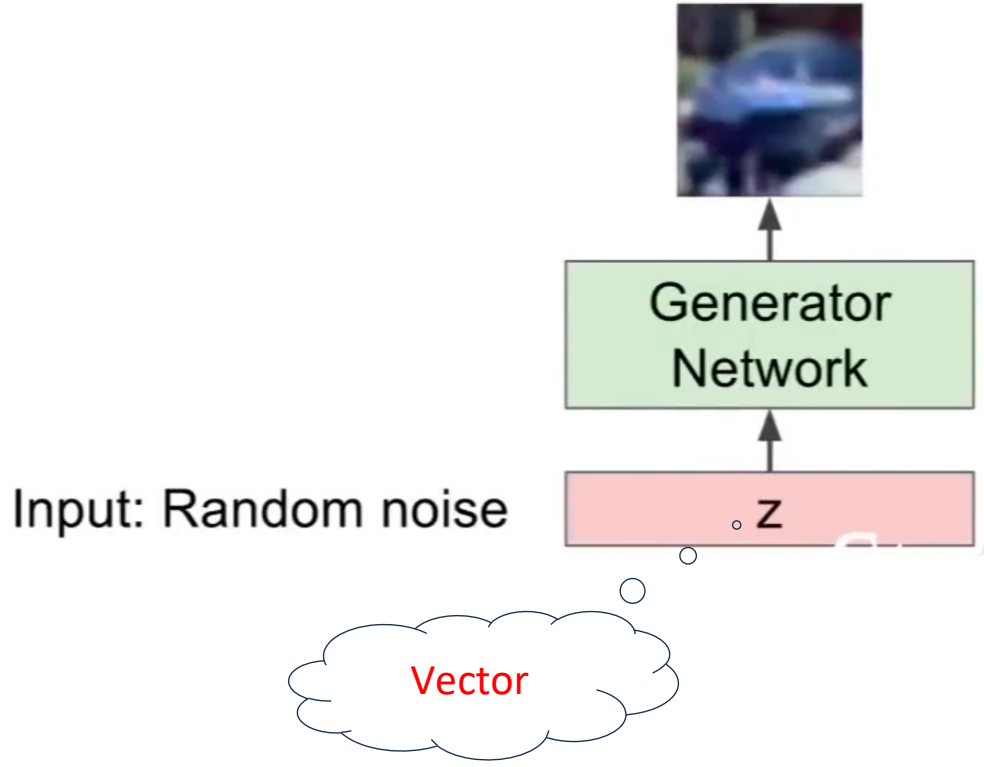
Comparing Distributions via Samples (cont.)

Problem: Want to sample from complex, high-dimensional training distribution. No direct way to do this!

Solution: Sample from a simple distribution, e.g. random noise. Learn transformation to training distribution.

Q: What can we use to represent this complex transformation?

A: A neural network!



Comparing Distributions via Samples (cont.)

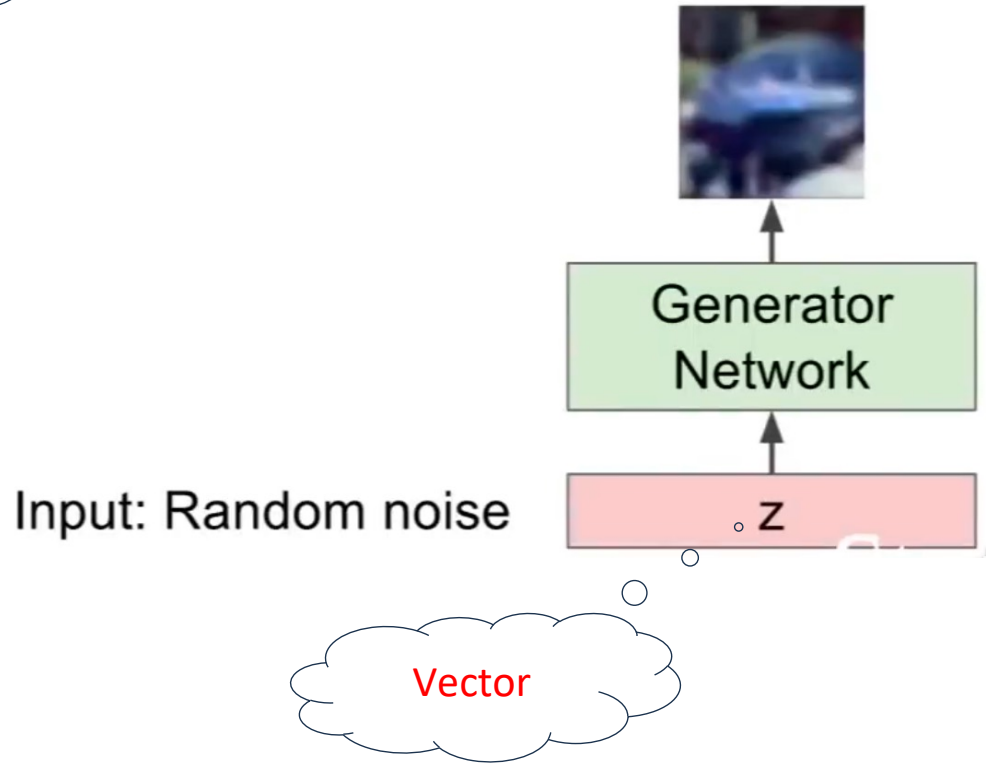
Problem: Want to sample from complex, high-dimensional training distribution. No direct way to do this!

Solution: Sample from a simple distribution, e.g. random noise. Learn transformation to training distribution.

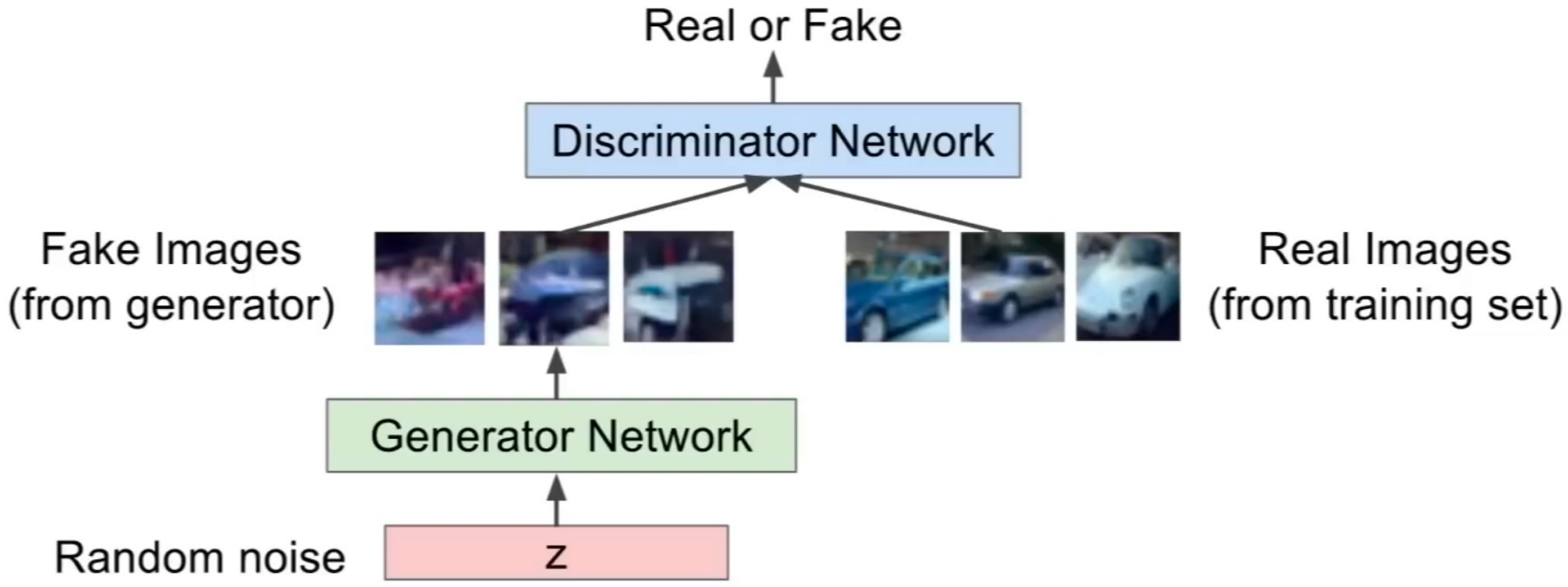
Why complex?

Q: What can we use to represent this complex transformation?

A: A neural network!



Generator network: try to fool the discriminator by generating real-looking images
Discriminator network: try to distinguish between real and fake images



What will be a good GAN model?

GANs (cont.)

- Generative Adversarial Networks were introduced by Ian Goodfellow et al. (2014).
- The idea behind GANs is to train two networks jointly.
- A **discriminator D** to classify samples as “real” or “fake”,
- A **generator G** to map a [simple] fixed distribution to samples that fool **D**.
- The approach is **adversarial** since the two networks have antagonistic objectives.

GANs (cont.)

Generator network: try to fool the discriminator by generating real-looking images
Discriminator network: try to distinguish between real and fake images

Train jointly in minmax game.

Minimax objective function:

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

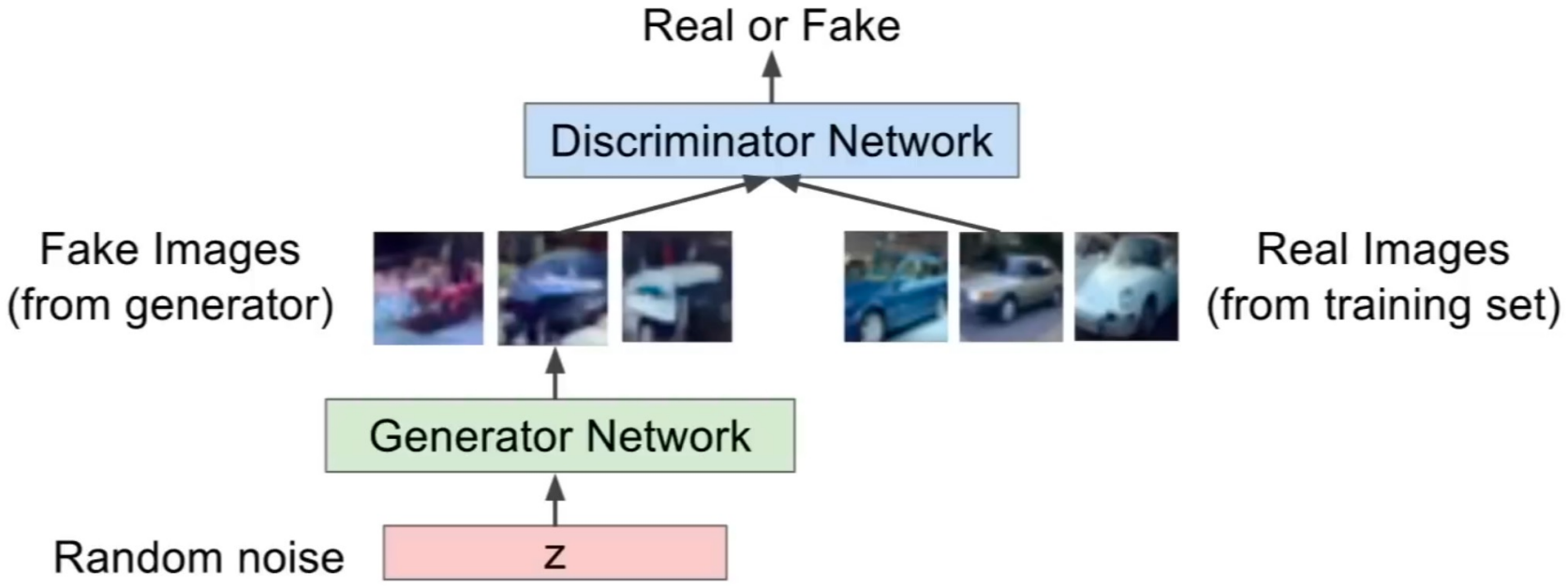
Discriminator outputs likelihood in (0,1) of real image

Minimax objective function:

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

- Discriminator (θ_d) wants to **maximize objective** such that D(x) is close to 1 (real) and D(G(z)) is close to 0 (fake)
- Generator (θ_g) wants to **minimize objective** such that D(G(z)) is close to 1 (discriminator is fooled into thinking generated G(z) is real)

GANs (cont.)

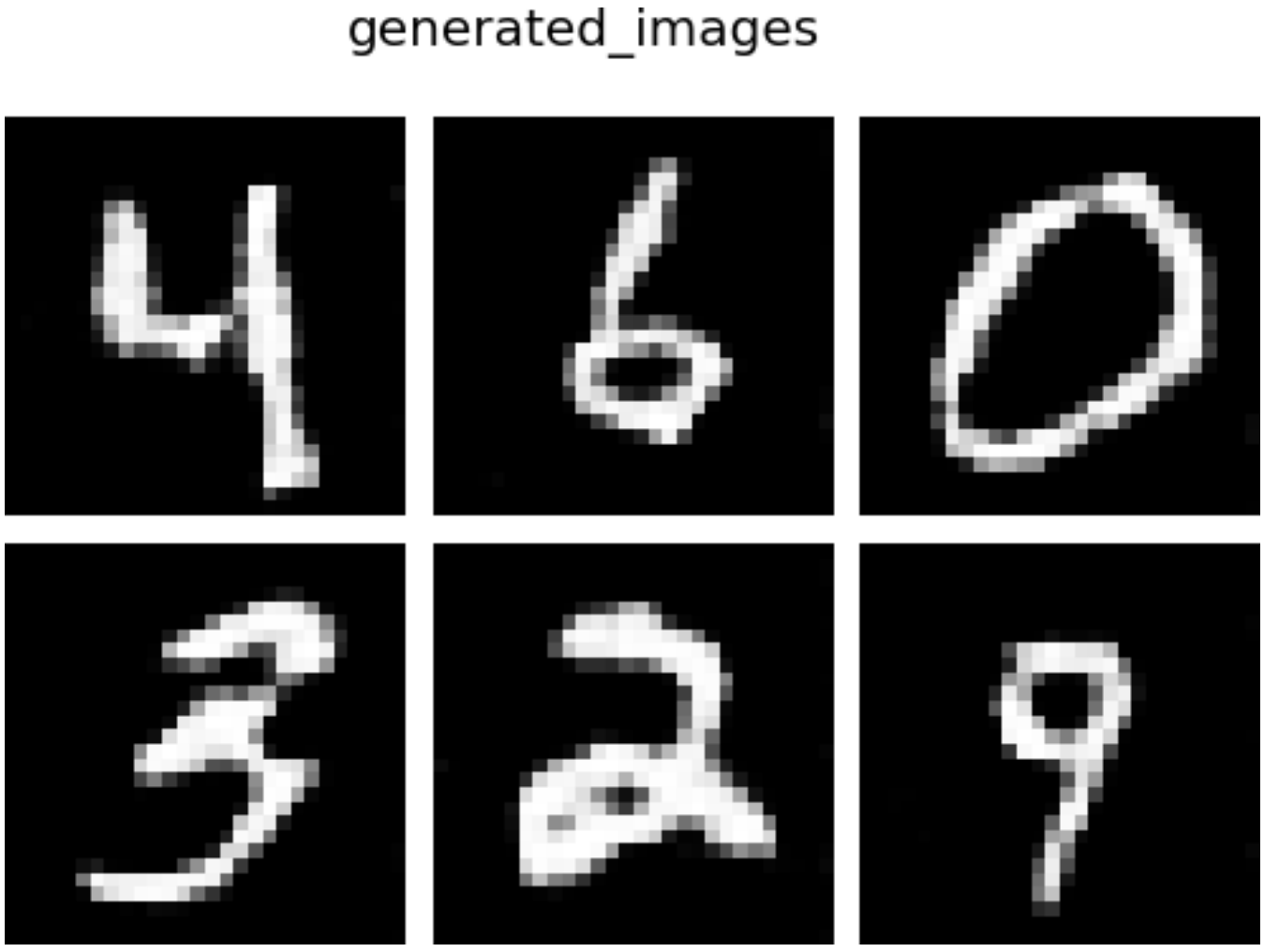


Is ‘z’ learnable? Gets updated?

GANs - Interactive Demo

https://poloclub.github.io/ganlab/

GANs - Results



GAN generated samples for MNIST digits dataset

GANs - Results (cont.)

GAN generated samples for bedroom images

GANs - Results (cont.)

GAN generated samples for anime character faces

Problems with GANs

Alternate between:

1. **Gradient ascent** on discriminator

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

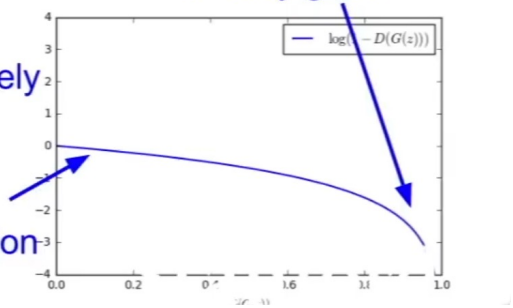
2. **Gradient descent** on generator

$$\min_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))$$

In practice, optimizing this generator objective does not work well!

Gradient signal dominated by region where sample is already good

When sample is likely fake, want to learn from it to improve generator. But gradient in this region is relatively flat!



Alternate between:

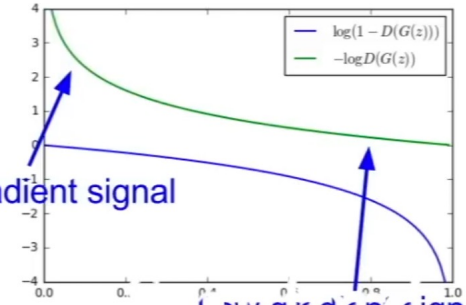
1. **Gradient ascent** on discriminator

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

2. **Instead: Gradient ascent** on generator, different objective

$$\max_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(D_{\theta_d}(G_{\theta_g}(z)))$$

Instead of minimizing likelihood of discriminator being correct, now maximize likelihood of discriminator being wrong.
Same objective of fooling discriminator, but now higher gradient signal for bad samples => works much better! Standard in practice.



High gradient signal

Low gradient signal

Training GANs

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:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D_{\theta_d}(x^{(i)}) + \log(1 - D_{\theta_d}(G_{\theta_g}(z^{(i)}))) \right]$$

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 ascending its stochastic gradient (improved objective):

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log(D_{\theta_d}(G_{\theta_g}(z^{(i)})))$$

end for

Training GANs

Some find k=1 more stable, others use k > 1, no best rule.

Recent work (e.g. Wasserstein GAN) alleviates this problem, better stability!

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:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D_{\theta_d}(x^{(i)}) + \log(1 - D_{\theta_d}(G_{\theta_g}(z^{(i)}))) \right]$$

  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 ascending its stochastic gradient (improved objective):

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log(D_{\theta_d}(G_{\theta_g}(z^{(i)})))$$

end for

Problems with GANs (cont.)

- Hard to achieve Nash equilibrium
  - Two models are trained simultaneously to find a Nash equilibrium to a two player non-cooperative game. However, each model updates its cost independently with no respect to another player in the game. Updating the gradient of both models concurrently cannot guarantee a convergence
  - How to Train a GAN? Tips and tricks to make GANs work by Soumith Chintala https://github.com/soumith/ganhacks

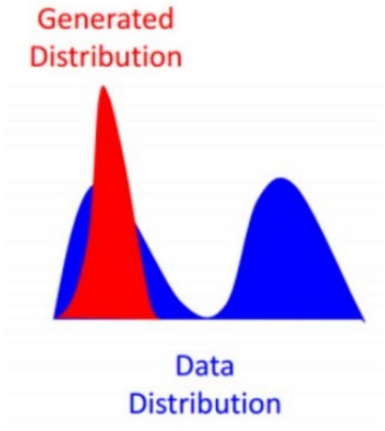
Problems with GANs (cont.)

Mode collapse occurs when the generator in a GAN produces limited variety in its outputs, often repeating the same or very similar samples. This leads to a lack of diversity in the generated data.

Why Mode Collapse Happens?

Generator's Objective: The generator aims to produce outputs that the discriminator classifies as real.

Local Minima: The generator might find a local minimum where producing a few realistic samples is enough to fool the discriminator, leading to repetitive outputs.



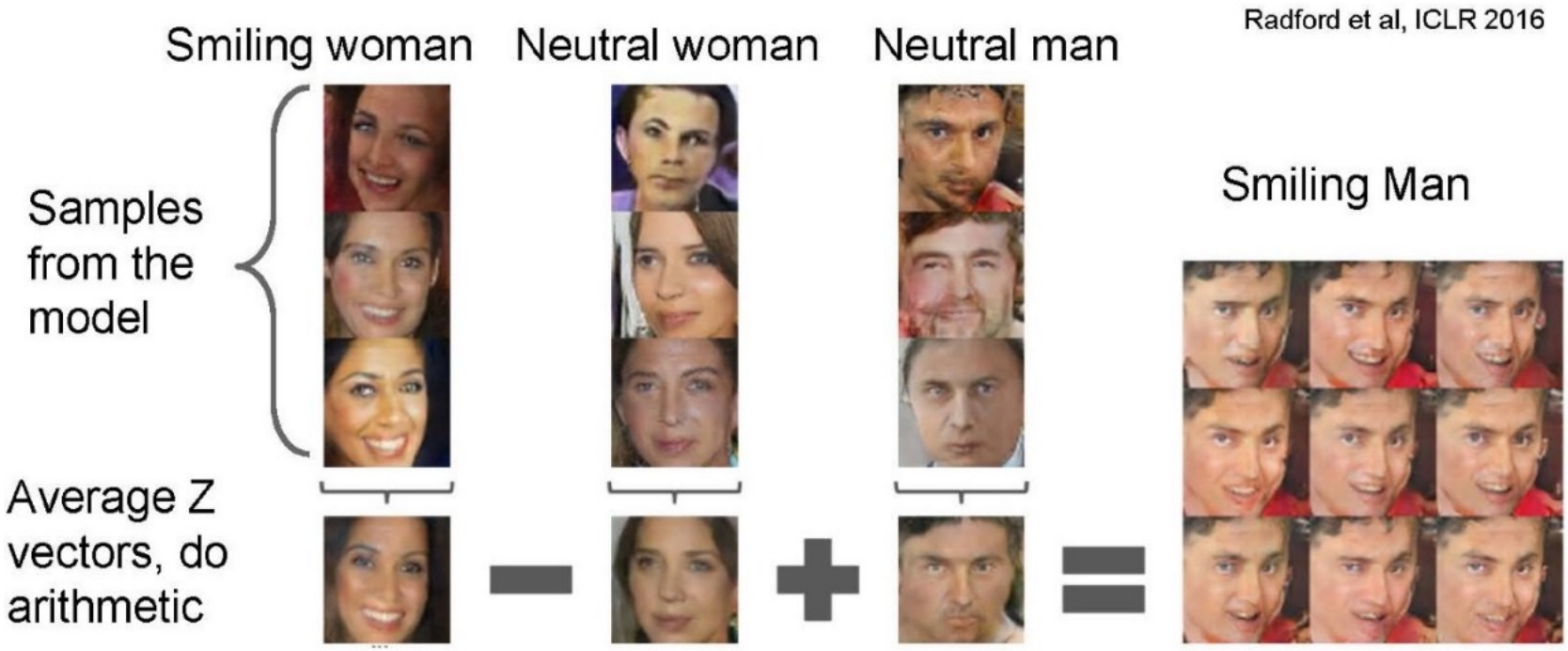
Problems with GANs (cont.)

GAN mode collapse on MNIST digits dataset

GAN Variants

- After the invention of GANs, there has been done a lot of research around them.
- A named list of GANs can be found here.

GANs - Latent space Interpolation



Latent space interploation with GANs

GANs - Latent space Interpolation (cont.)

Latent space interpolation with StyleGAN
(Demo by Xander Steenburgge)
https://colab.research.google.com/drive/1mH70YxGNlnEaSOn0J8LsgkI-QOvsIb3MscrollTo=uEhxBvAR-7y3

Some more GAN results

Image Inpainting. https://www.nvidia.com/en-us/research/ai-demos/

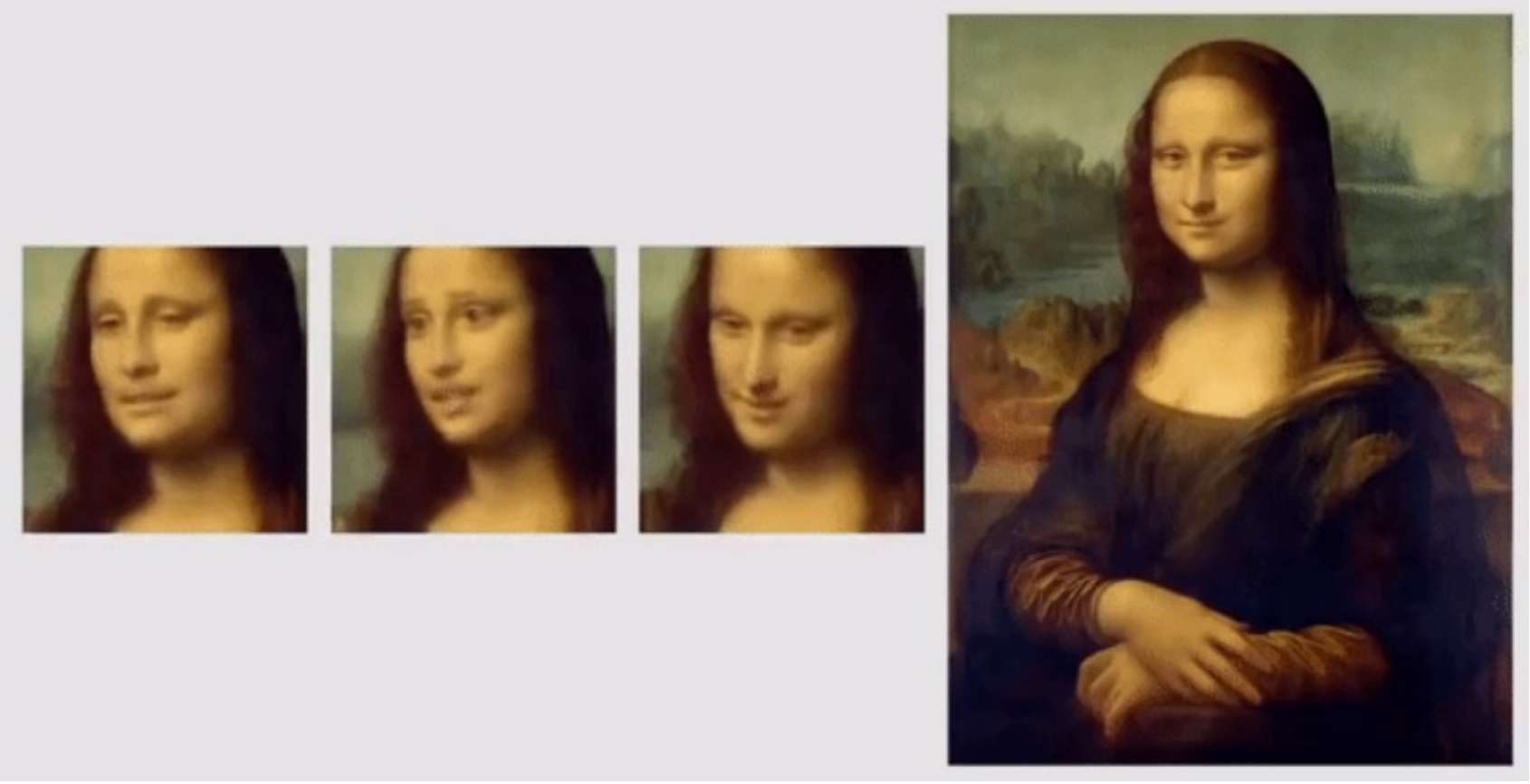
Some more GAN results (cont.)

this small bird has a pink breast and crown, and black primaries and secondaries.

this white and yellow flower have thin white petals and a round yellow stamen

Text to Image Synthesis with GANs

Some more GAN results (cont.)



Living Portraits with GANs

Some more GAN results (cont.)

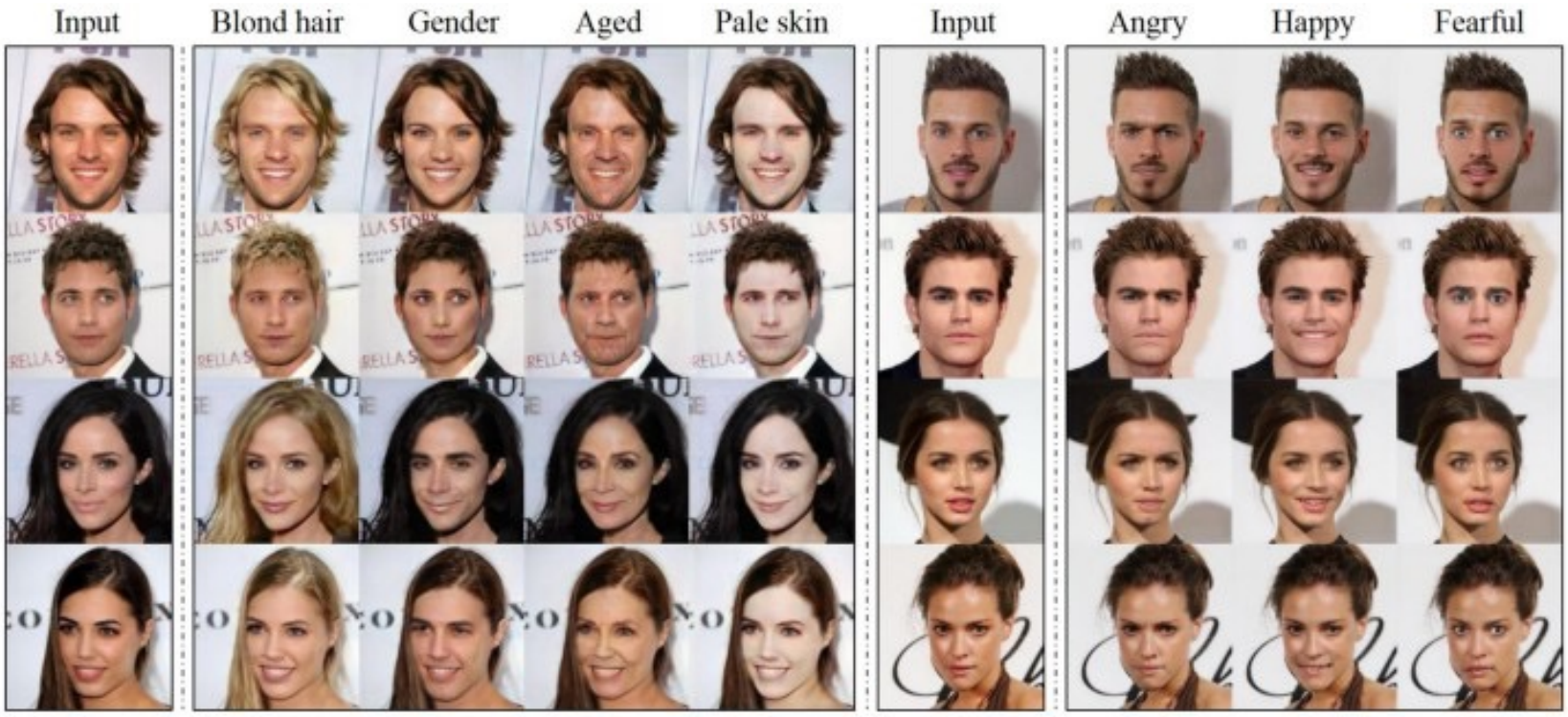


Image to Image translation in multiple domains with StyleGAN (Choi et al.)

Denoising Diffusion Probabilistic Models

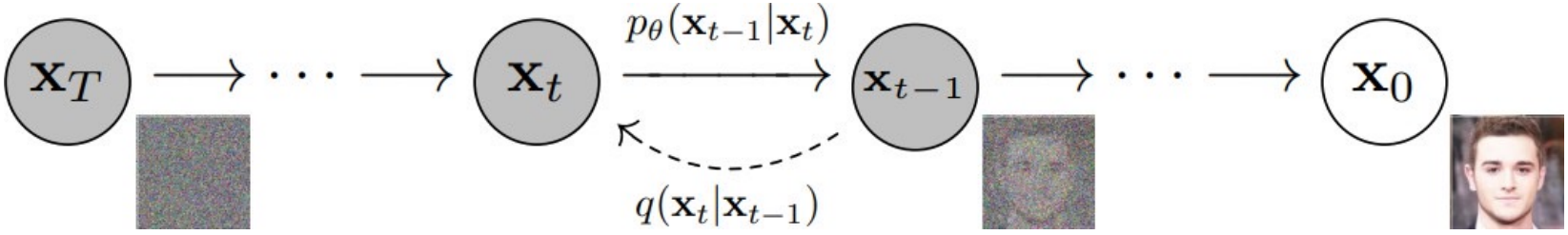
- Denoising diffusion models, now also known as score-based generative models, have recently emerged as a powerful class of generative models. They demonstrate astonishing results in high-fidelity image generation, often even outperforming GANs.

Diffusion Models Beat GANs on Image Synthesis Dharwal & Nichol, OpenAI, 2021

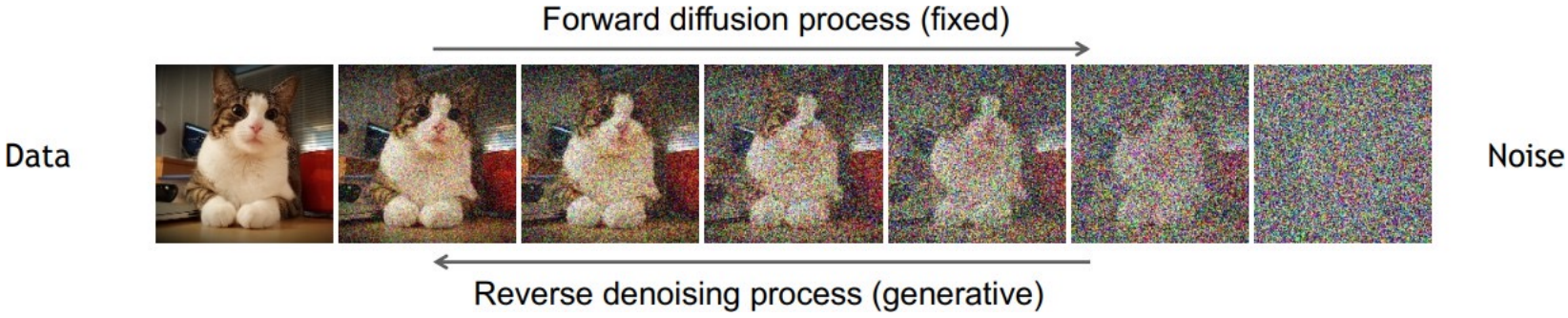
Denoising Diffusion Probabilistic Models (cont.)

Denoising diffusion models consist of two processes:

- A fixed (or predefined) forward diffusion process q of our choosing, that gradually adds Gaussian noise to an image, until you end up with pure noise
- a learned reverse denoising diffusion process p_θ , where a neural network is trained to gradually denoise an image starting from pure noise, until you end up with an actual image.

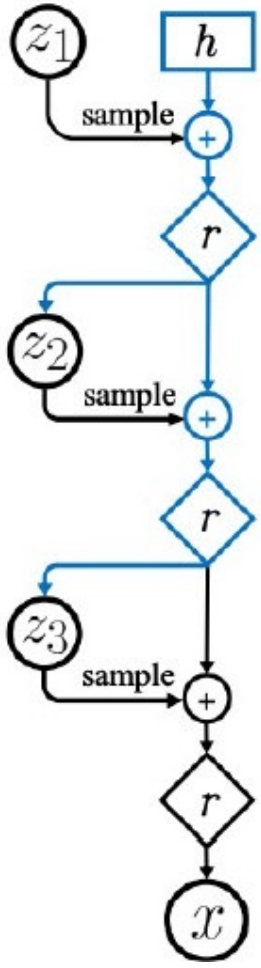


Denoising Diffusion Probabilistic Models (cont.)



Connection to VAEs

- Diffusion models can be considered as a special form of hierarchical VAEs (one VAE after another).
- However, in diffusion models:
  - The encoder is fixed.
  - The latent variables have the same dimension as the data.
  - The denoising model is shared across different timestep.



Problems with Diffusion Models

- For now, it seems that the major limitation of the Diffusion Models is its notoriously slow sampling procedure which normally requires hundreds to thousands of time discretization steps of the learned diffusion process to reach the desired accuracy.

Diffusion Models - Current Research

أكاديمية كاوست
KAUST ACADEMY

- Denoising Diffusion Implicit Model
  - DDIM roughly sketches the final sample then refine it with the reverse process.
- Improved Denoising Diffusion Probabilistic Models
  - Train σ^2 while training the diffusion model instead of fixing it.
- Score-Based Generative Modeling through Stochastic Differential Equations
  - Model the gradient of the log probability density function, a quantity often known as the (Stein) score function.
- Tackling the Generative Learning Trilemma with Denoising Diffusion GANs
  - Introduce denoising diffusion generative adversarial networks (denoising diffusion GANs) that model each denoising step using a multimodal conditional GAN.
- Cascaded Diffusion Models for High Fidelity Image Generation
  - Cascaded diffusion models to boost sample quality.

GLIDE - OpenAI

"a man wearing a white hat"

Image Inpainting with GLIDE

[1] GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided

DALL.E 2 - OpenAI

a shiba inu wearing a beret and black turtleneck

a close up of a handpalm with leaves growing from it

Text to image generation eith DAll.E 2

[1] Hierarchical Text-Conditional Image Generation with CLIP Latents

DALL.E 2 - OpenAI

Fix the CLIP embedding $\mathbf{z}_i$

Decode using different decoder latents $\mathbf{x}_T$

Image Variations

[1] Hierarchical Text-Conditional Image Generation with CLIP Latents

DALL.E 2 - OpenAI

Interpolate image CLIP embeddings $\mathbf{z}_i$

Use different $\mathbf{x}_T$ to get different interpolation trajectories.

Image interpolation

[1] Hierarchical Text-Conditional Image Generation with CLIP Latents

DALL.E 2 - OpenAI

a photo of a cat → an anime drawing of a super saiyan cat, artstation

a photo of a victorian house → a photo of a modern house

a photo of an adult lion → a photo of lion cub

Change the image CLIP embedding towards the difference of the text CLIP embeddings of two prompts.

Decoder latent is kept as a constant.

Text Difference Image interpolation

[1] Hierarchical Text-Conditional Image Generation with CLIP Latents

Imagen - Google

A brain riding a rocketship heading towards the moon.

[1] Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding

Imagen - Google

A dragon fruit wearing karate belt in the snow.

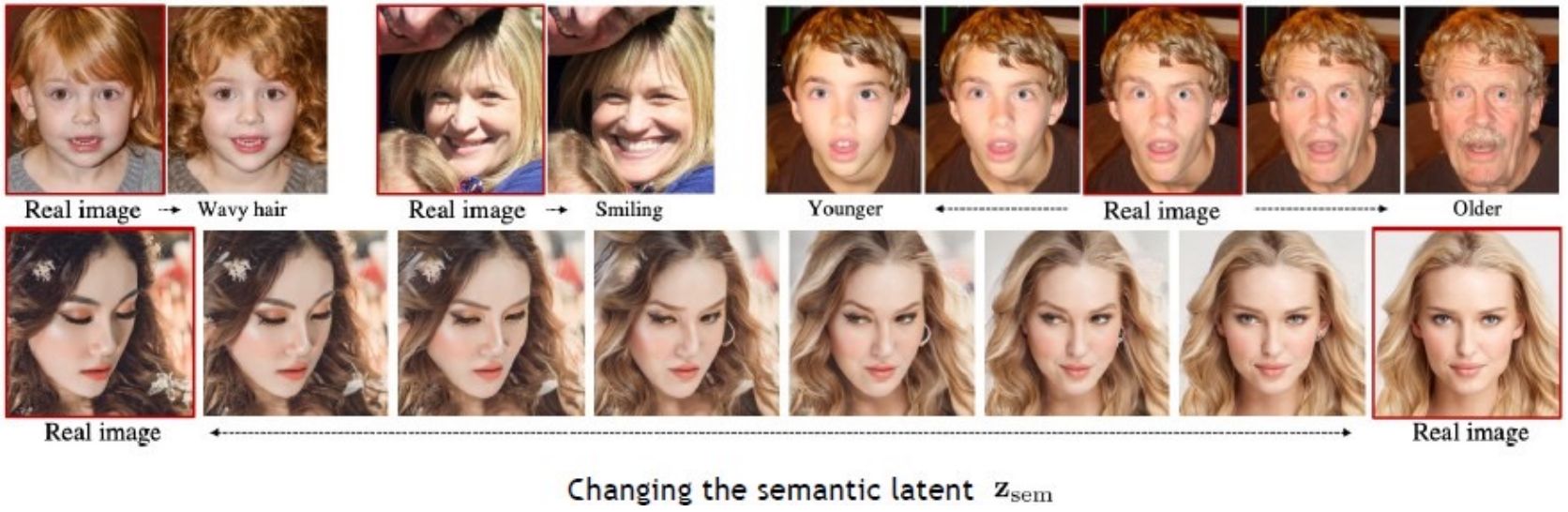
[1] Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding

Imagen - Google

A relaxed garlic with a blindfold reading a newspaper while floating in a pool of tomato soup.

[1] Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding

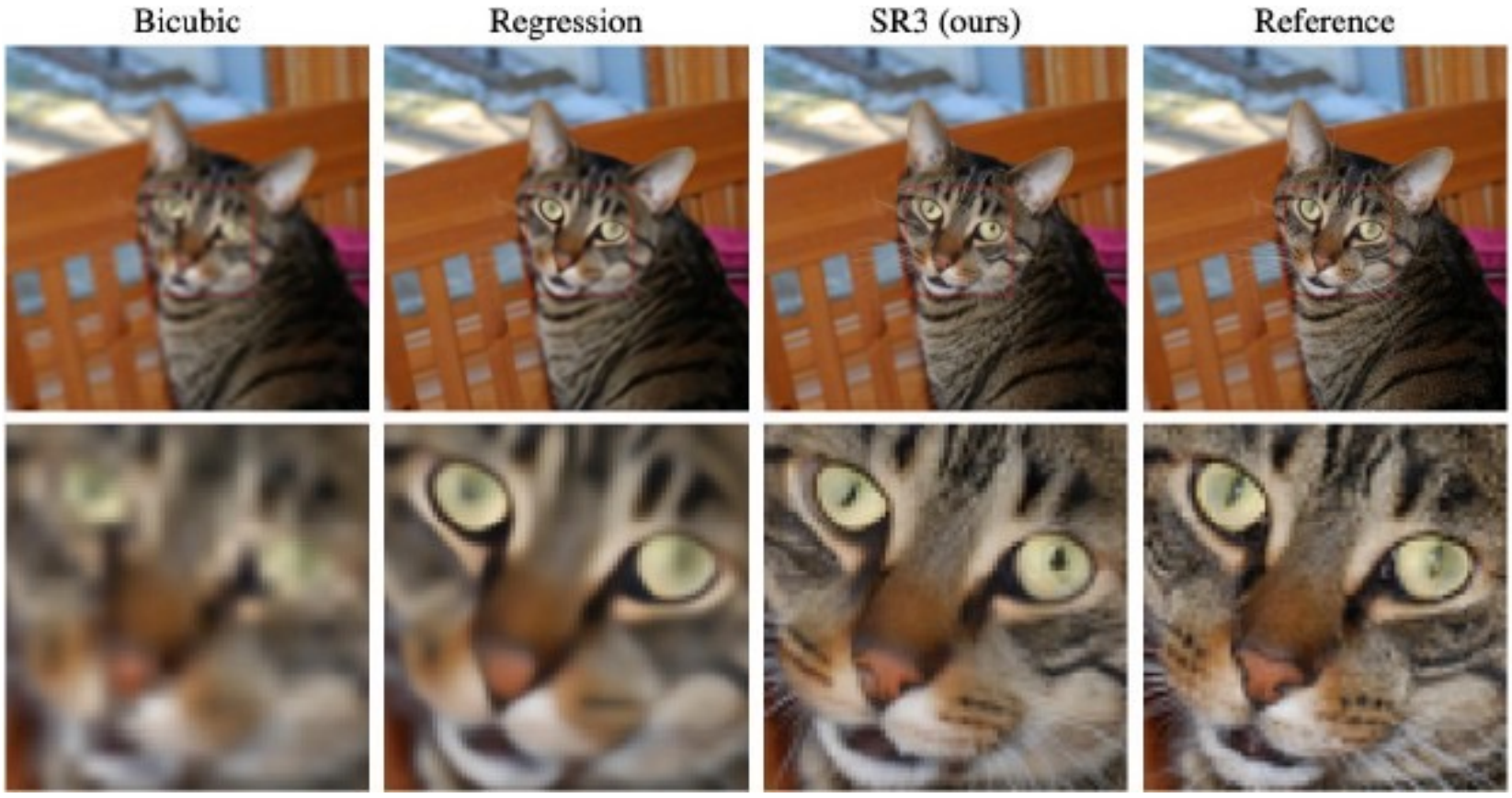
Diffusion Autoencoders



Learning semantic meaningful latent representations in diffusion models

[1] Diffusion Autoencoders: Toward a Meaningful and Decodable Representation

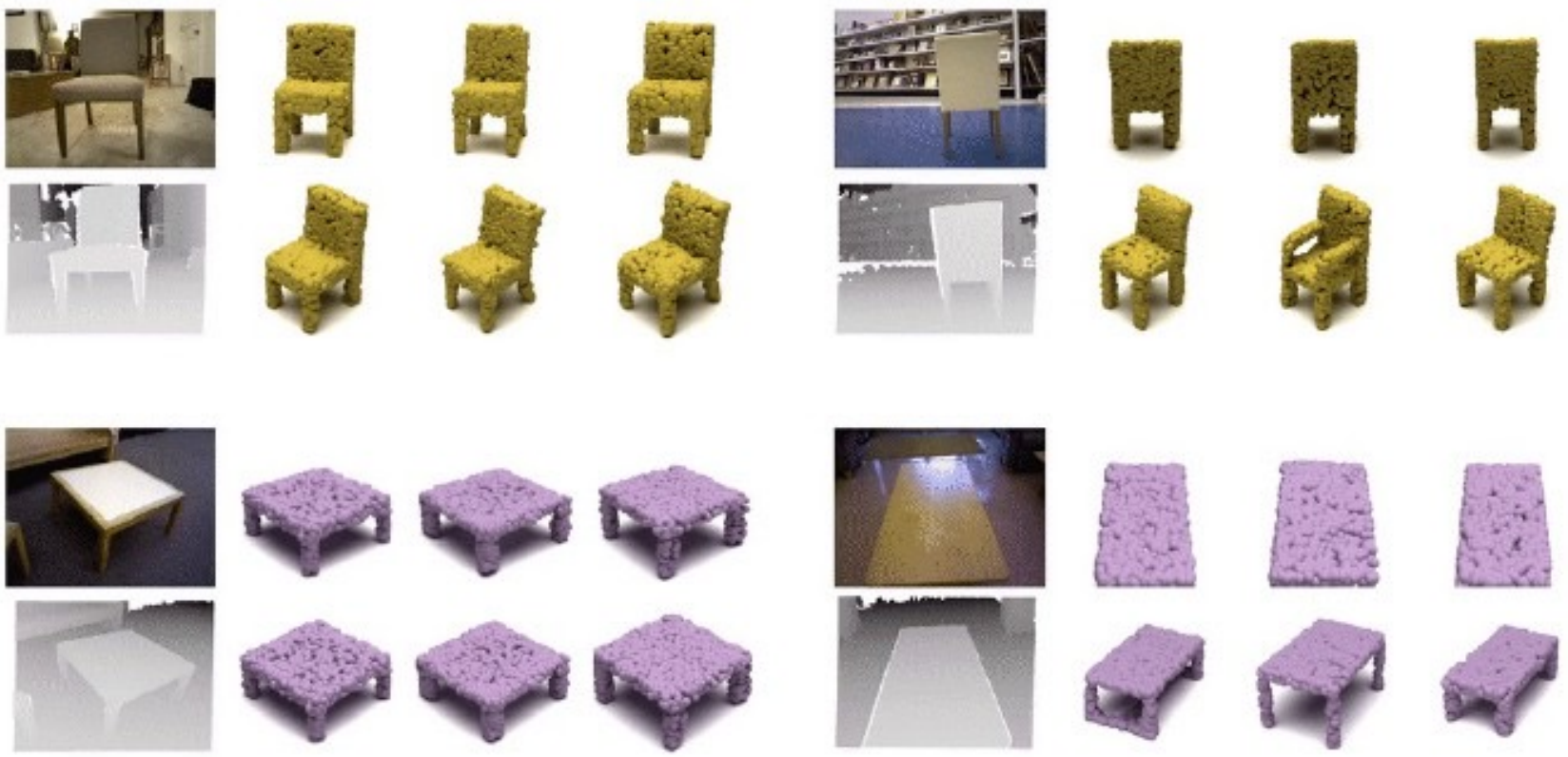
Super Resolution



Natural Image Super Resolution $64 \times 64 \longrightarrow 256 \times 256$

[1] Image Super-Resolution via Iterative Refinement

3D Shape Generation



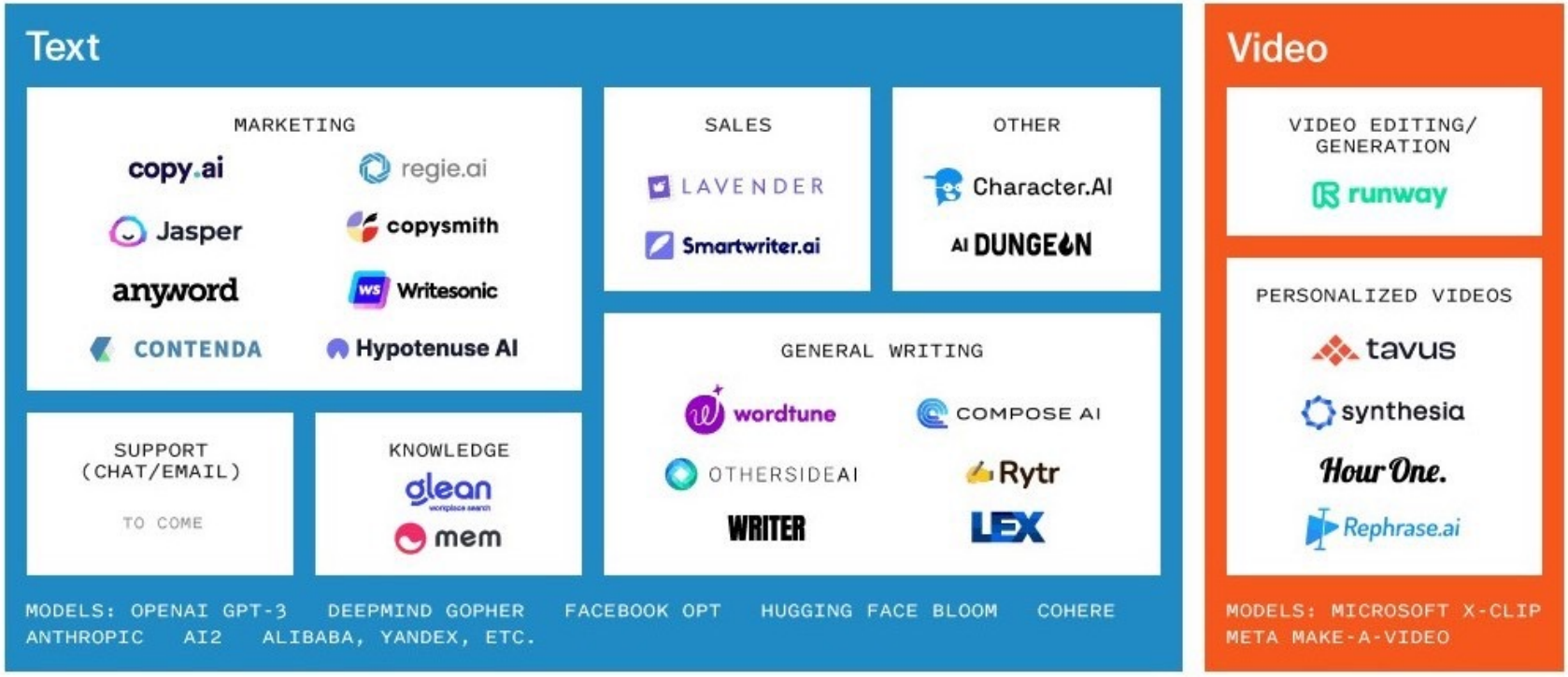
[1] 3D Shape Generation and Completion through Point-Voxel Diffusion

Try It Yourself!

Text to Image generation with stable diffusion.
https://huggingface.co/spaces/stabilityai/stable-diffusion

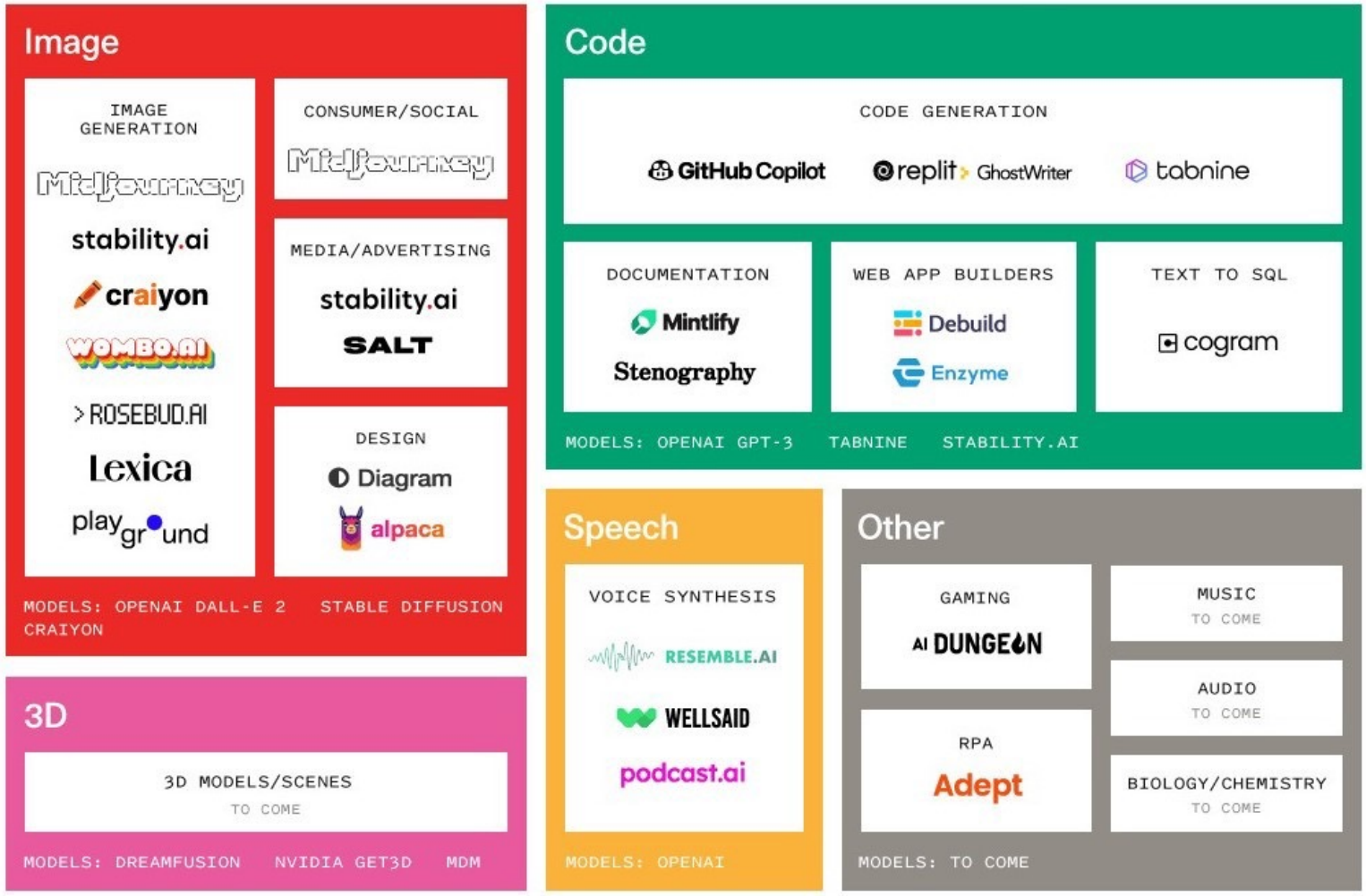
[1] High-Resolution Image Synthesis with Latent Diffusion Models

The Generative AI Landscape



[1] Sonya Huang (@sonyatweetybird)

The Generative AI Landscape

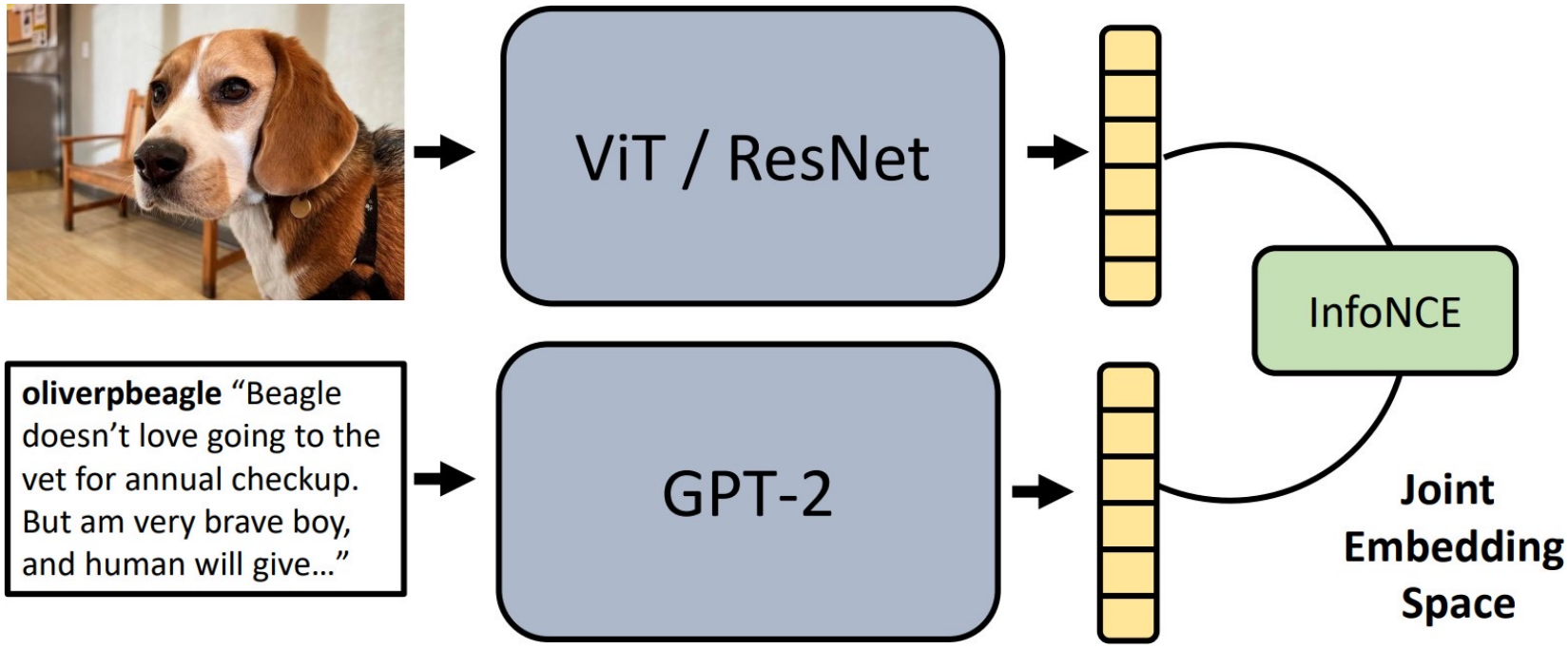


[1] Sonya Huang (@sonyatweetybird)

Image and Text

CLIP – Connecting Images and Text

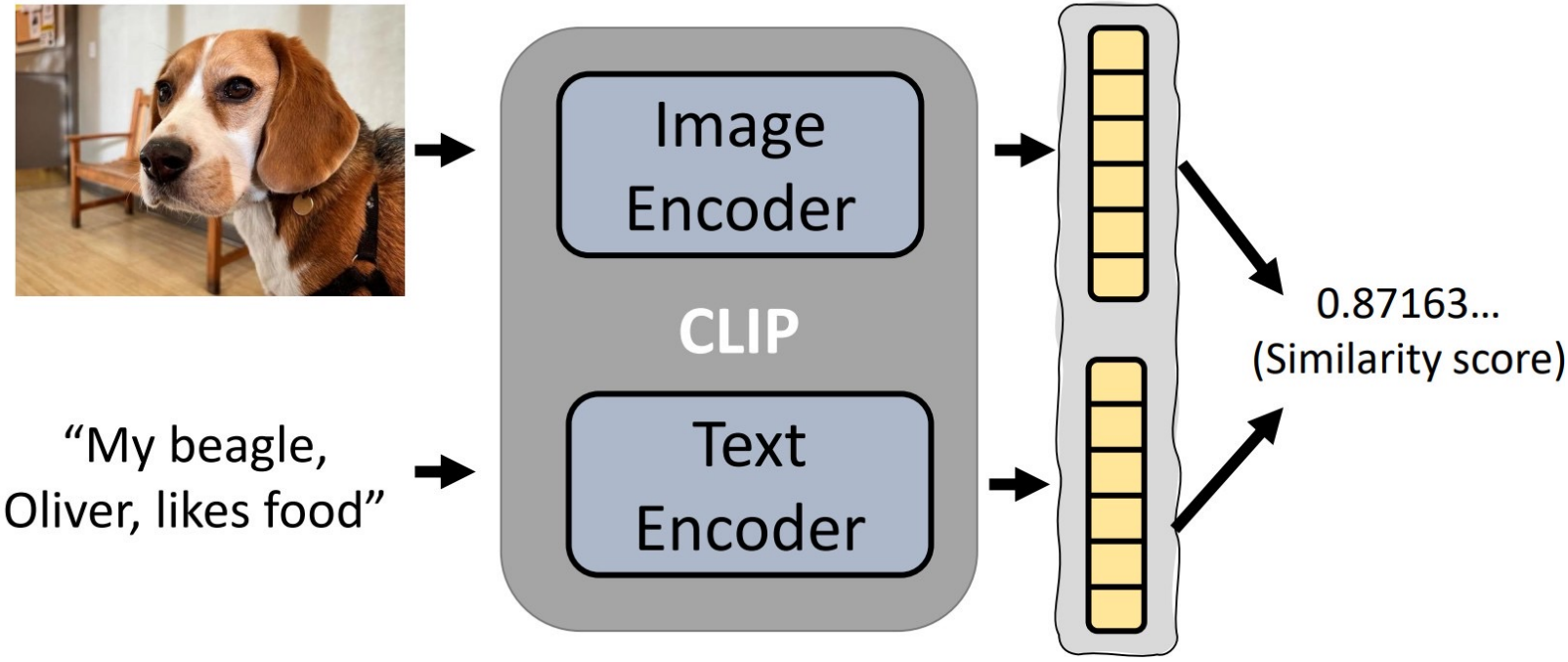
Information Noise Contrastive Estimation



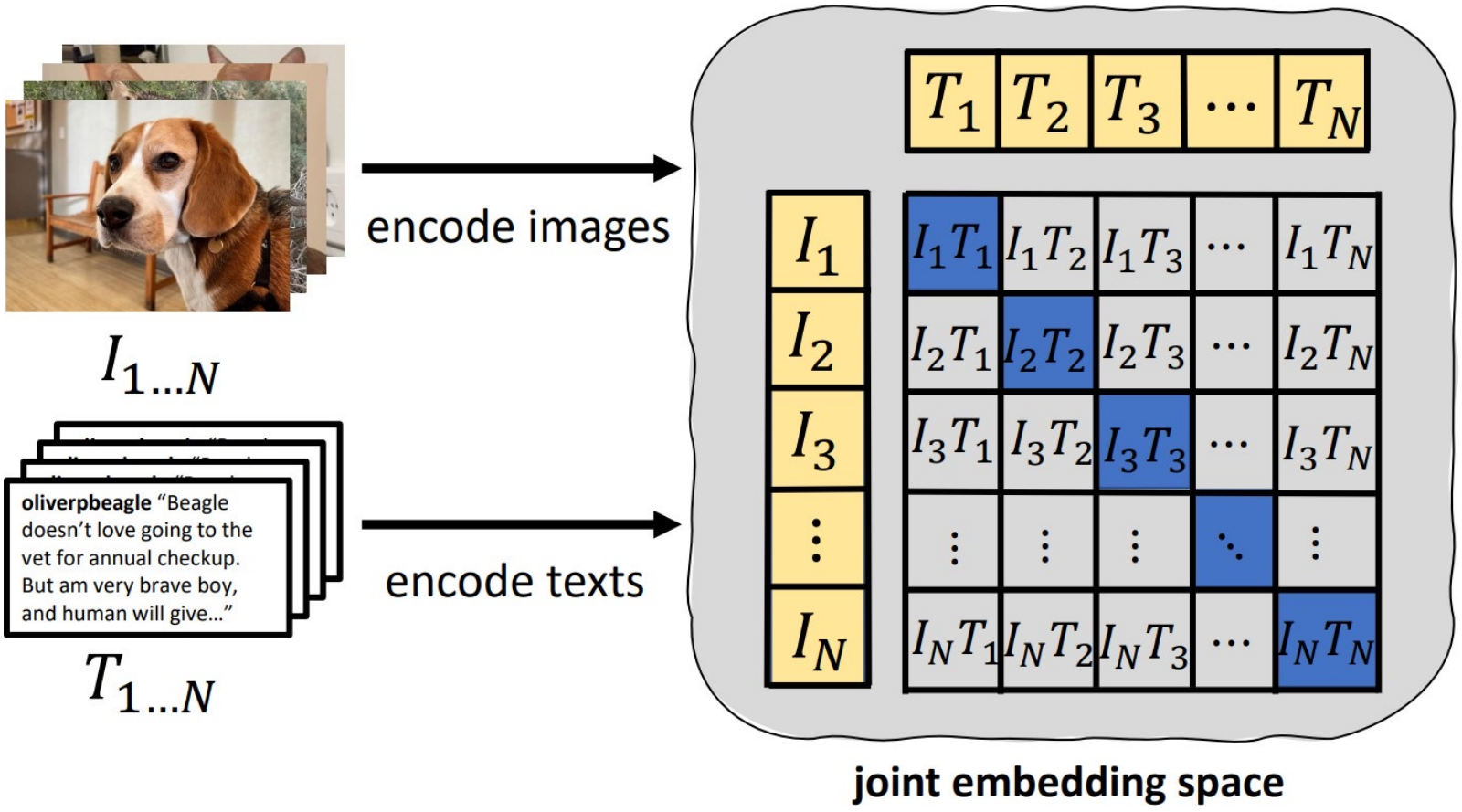
[0]Radford et. al, Learning Transferable Visual Models From Natural Language Supervision, ArXiv'21

CLIP – Connecting Images and Text

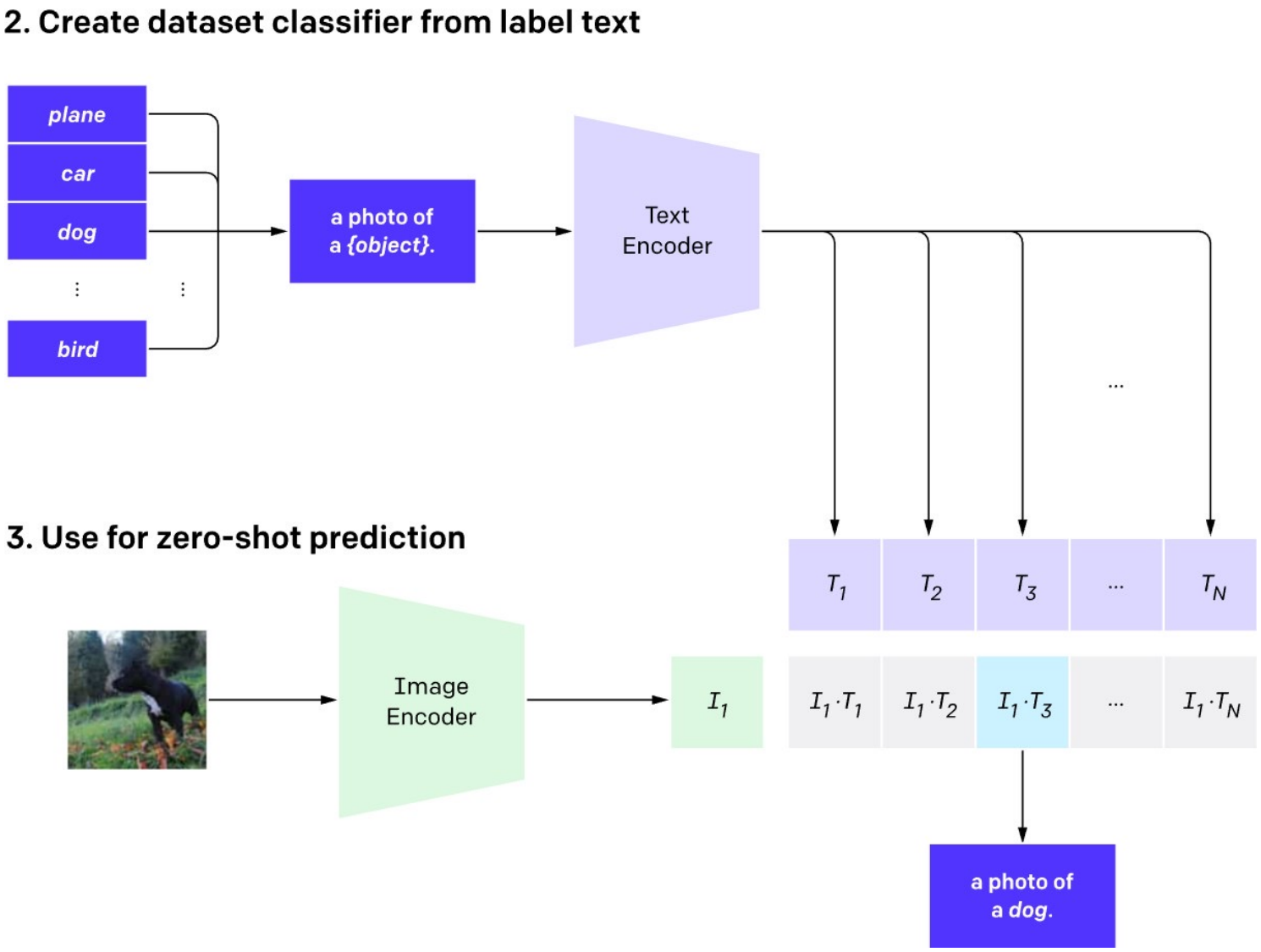
- Contrastive Language Image Pretraining



CLIP – Connecting Images and Text



CLIP - Zero Shot Capabilities



[0]https://openai.com/research/clip

Using CLIP for generative tasks

Generation

A beautiful painting of a building in a serene landscape

Editing

"A cake made of ice"

Using CLIP for generative tasks

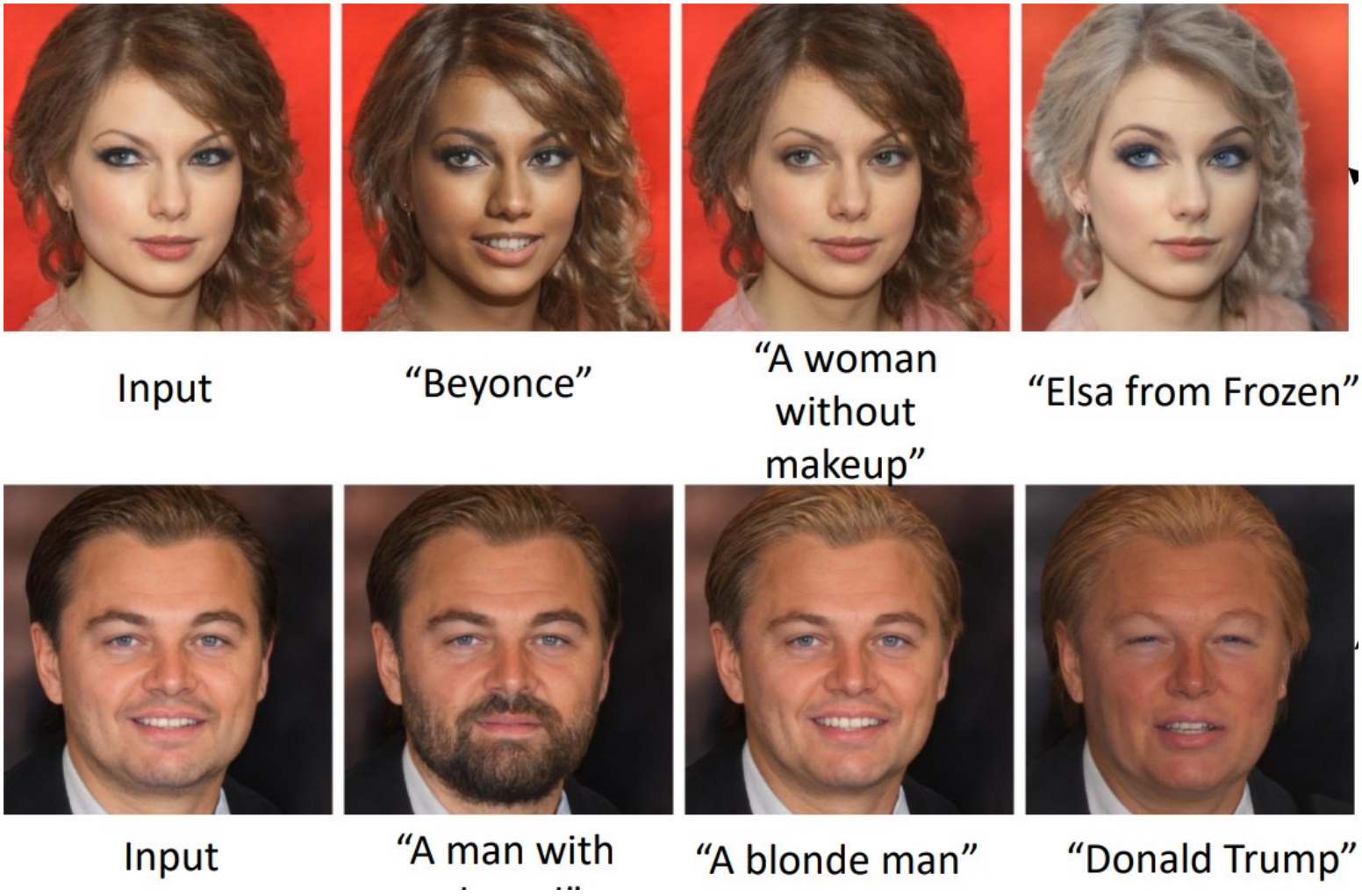
Generation

A beautiful painting of a building in a serene landscape

Editing

"A cake made of ice"

StyleCLIP - Results



[0]StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery, Patashnik and Wu et al. ICCV 2021

StyleCLIP - Results

[0]StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery, Patashnik and Wu et al. ICCV 2021

Text2Live - Results

[0]Text2LIVE: Text-Driven Layered Image and Video Editing, Bar-Tal Ofri-Amar and Fridman et al. ECCV 2022

References

Reference Slides

- Fei-Fei Li ”Generative Deep Learning” CS231
- Hao Dong ”Deep Generative Models”
- Hung-Yi Lee ”Machine Learning”
- Francois Fleuret ”Deep Learning” EE559
- Murtaza Taj ”Deep Learning” CS437
- Kreis, Gao & Vahdat CVPR 2022 Tutorial
- Rogge & Rasul The Annotated Diffusion Model