Deep Learning for Computer Vision

113-1/Fall 2024

https://cool.ntu.edu.tw/courses/41702 (NTU COOL)

http://vllab.ee.ntu.edu.tw/dlcv.html (Public website)

Yu-Chiang Frank Wang 王鈺強, Professor

Dept. Electrical Engineering, National Taiwan University

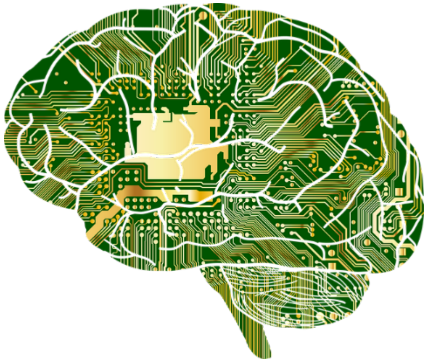
2024/10/8

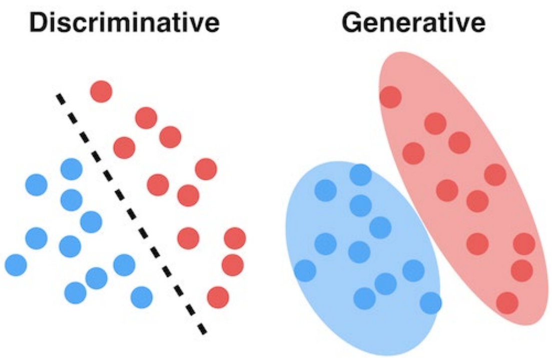
About Final Projects

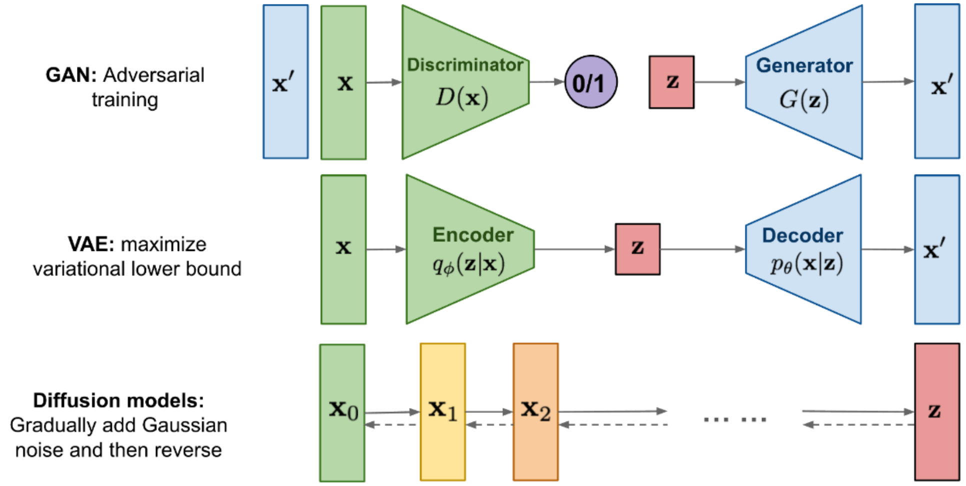
- Updates
  - **1:30pm-5pm, Dec 26th, Thursday**
    (sorry, no space available on 25th)
  - 3~4 people per group
    (team up in mid Nov.)
  - Adapt from latest CVPR/ICCV/ECCV challenges or competitions
  - Poster presentation;
    code required for reproduction
  - Intra/inter-group evaluation
  - Snack/drink provided

What's to Be Covered Today…

- Generative Models
  - Diffusion Model
    - Conditional Diffusion Model
      - Classifier Guidance
      - Classifier-Free Guidance
      - Text/Image Guidance
    - Personalization via Diffusion Model
  - Generative Adversarial Network
  - HW #2 is out! (due 10/29)





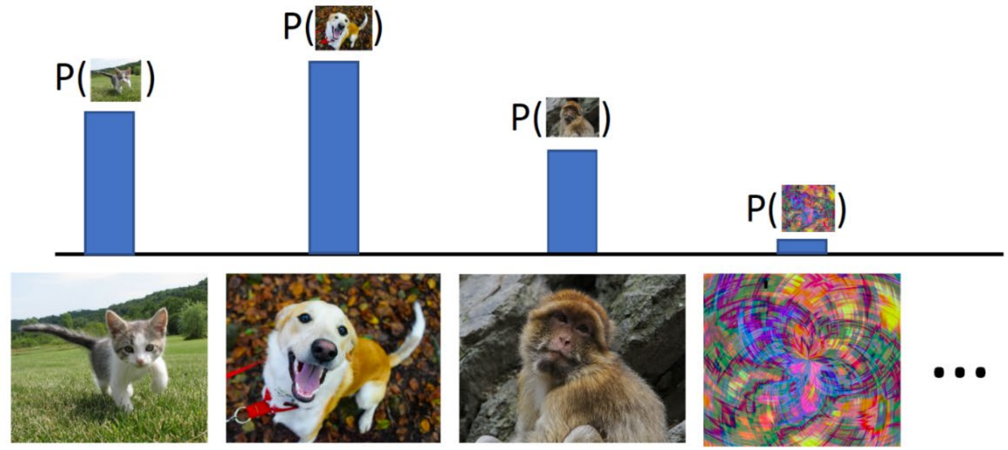


A Quick Recap of Generative Models

Discriminative Model: Learn a probability distribution p(y|x)

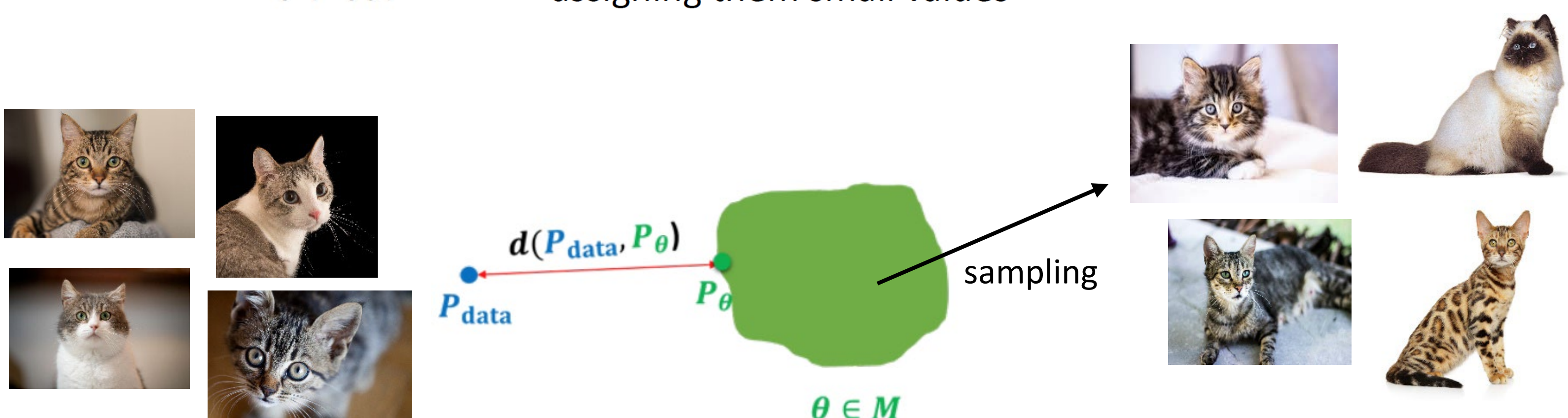
Generative Model: Learn a probability distribution p(x)

Conditional Generative Model: Learn p(x|y)



Generative model: All possible images compete with each other for probability mass

Model can "reject" unreasonable inputs by assigning them small values



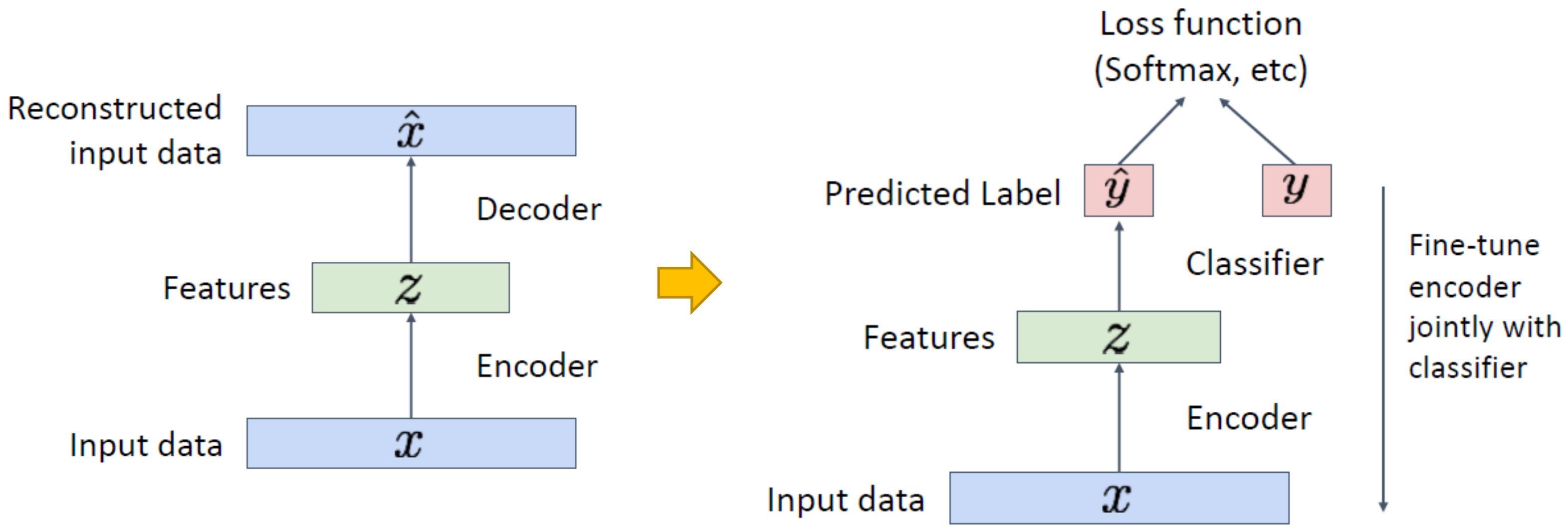
$x_i \sim p_{data}$

$\hat{x}_i \sim p_\theta$

Slide credit: UMich EECS 498-007

Autoencoder (AE)

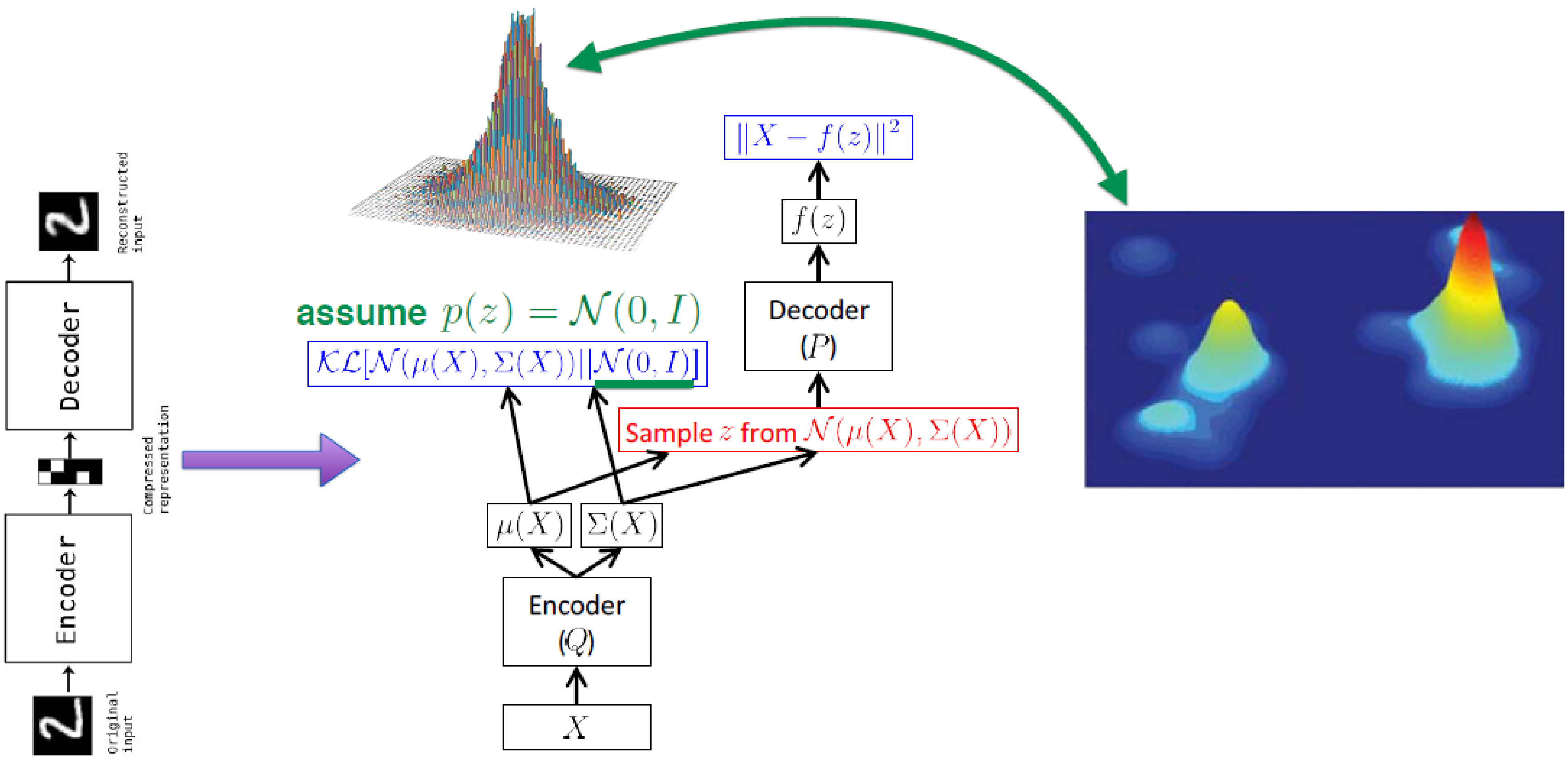
- Unsupervised learning for deriving latent representation
  - Train AE with reconstruction objectives
- Train autoencoder (AE) for downstream tasks
  - After AE training is complete, freeze/finetune the encoder and learn additional modules (e.g., MLP) for downstream tasks
  - E.g., to train a DNN for classification, one can freeze the encoder and learn an additional MLP as the classifier.



Slide credit: UMich EECS 498-007

From AE to Variational Autoencoder (VAE)

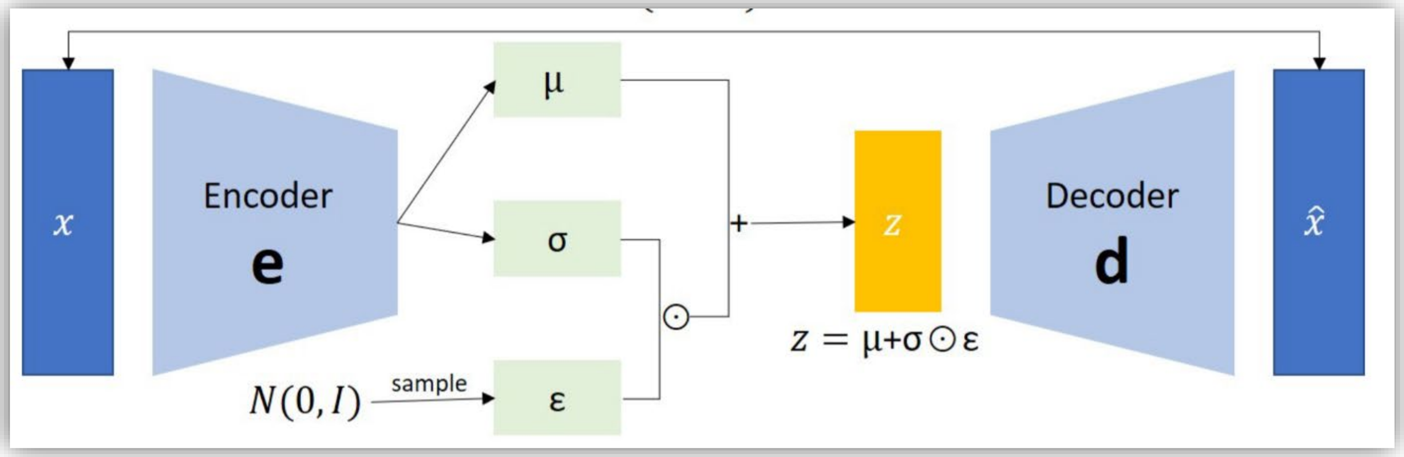
Now is a "*distribution*", we can assume it to be a distribution easy to sample from, e.g. Gaussian



Slide credit: W.-C. Chiu

Reparameterization Trick in VAE

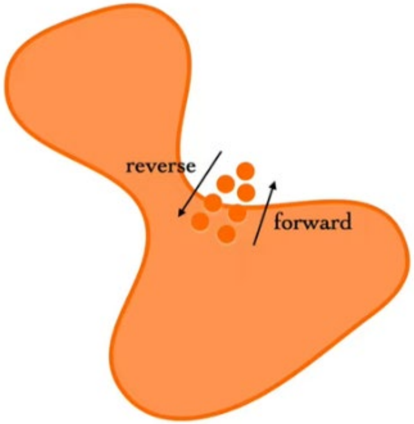
- Remarks
  - Given x, sample z from latent distribution (described by output parameters of encoder), we apply $z = \mu + \sigma \odot \varepsilon$ (ε simply generated by Normal distribution).
  - For training, this enables BP gradients in encoder through **μ** and **σ**; for inference, this introduces generation stochasticity.



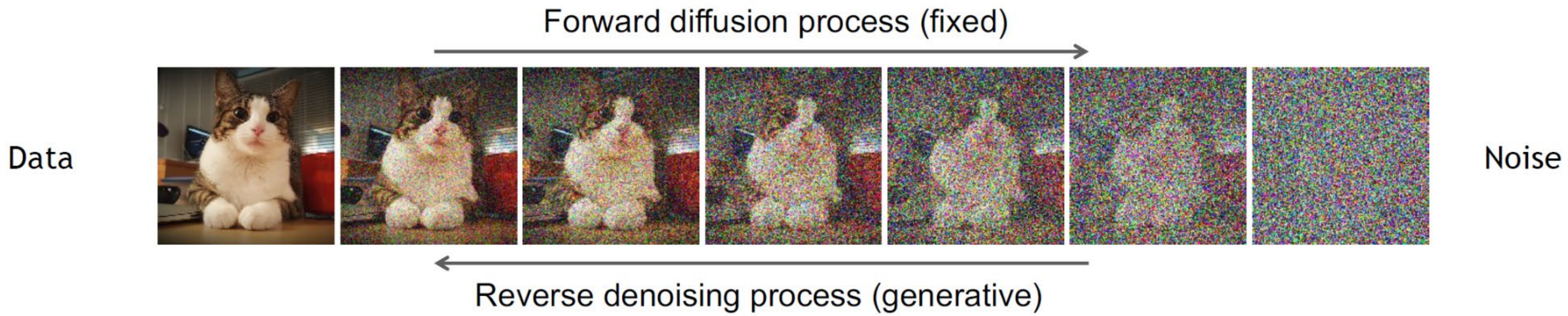
https://medium.com/geekculture/variational-autoencoder-vae-9b8ce5475f68

Denoising Diffusion Probabilistic Models (DDPM)

Learning to generate by denoising



- 2 processes required for training:
  - **Forward diffusion process**
    - gradually add noise to input
  - **Reverse diffusion process**
    - learns to generate/restore data by denoising
    - typically implemented via a **conditional U-net**



Ho et al., Denoising Diffusion Probabilistic Models, NeurIPS 2020
Song et al., Score-Based Generative Modeling through Stochastic Differential Equations, ICLR 2021

Slide credit: Kreis, Gao, & Vahdat

VAE vs. DDPM

Variational Autoencoder

$$\mathbf{x} \rightarrow \boxed{\text{Encoder } \phi} \rightarrow q_\phi(\mathbf{z}|\mathbf{x}) \xrightarrow{\text{sample}} \mathbf{z} \rightarrow \boxed{\text{Decoder } \theta} \rightarrow p_\theta(\mathbf{x}|\mathbf{z}) \xrightarrow{\text{sample}} \hat{\mathbf{x}}$$

Observed	$\mathbf{x}$
Latent	$\mathbf{z}$

$$\text{Maximize } \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\left[\log \frac{p(\mathbf{x},\mathbf{z})}{q(\mathbf{z}|\mathbf{x})}\right]$$

Diffusion model

$$\mathbf{x}_0 \xrightarrow{\text{Encoding}} \mathbf{x}_1 \xrightarrow{\text{Encoding}} \mathbf{x}_2 \cdots \xrightarrow{\text{Encoding}} \mathbf{x}_T \xrightarrow{\text{Decoding } \theta} \mathbf{x}_{T-1} \quad \cdots \xrightarrow{\text{Decoding } \theta} \hat{\mathbf{x}}_0$$

Observed	$\mathbf{x}_0$
Latent	$\mathbf{x}_1, \ldots, \mathbf{x}_T$

$$\text{Maximize } \mathbb{E}_{q(\mathbf{x}_1:\mathbf{x}_T|\mathbf{x}_0)}\left[\log \frac{p(\mathbf{x}_0:\mathbf{x}_T)}{q(\mathbf{x}_1:\mathbf{x}_T|\mathbf{x}_0)}\right]$$

Slide credit: J.-B. Huang

Training DDPM

$$\mathbf{x}_0 \xrightarrow{\text{Encoding}} \mathbf{x}_1 \xrightarrow{\text{Encoding}} \mathbf{x}_2 \cdots \xrightarrow{\text{Encoding}} \mathbf{x}_T \xrightarrow{\text{Decoding } \theta} \mathbf{x}_{T-1} \quad \cdots \quad \xrightarrow{\text{Decoding } \theta} \hat{\mathbf{x}}_0$$

$$\sum_{t=2}^{T} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}[\mathrm{D}_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))]$$

$q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)$

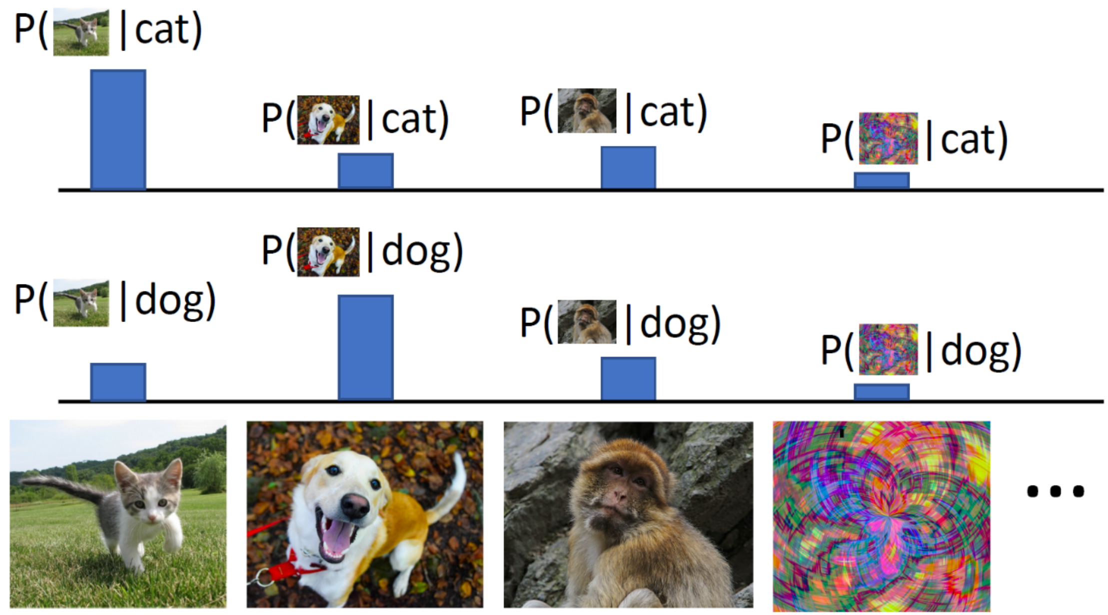
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$

From Unconditional to Conditional Generative Models

Discriminative Model: Learn a probability distribution p(y|x)

Generative Model: Learn a probability distribution p(x)

Conditional Generative Model: Learn p(x|y)



Conditional Generative Model: Each possible label induces a competition among all images

Slide credit: UMich EECS 498-007

$$\mathbf{x}_0 \xrightarrow{\text{Encoding}} \mathbf{x}_1 \xrightarrow{\text{Encoding}} \mathbf{x}_2 \cdots \xrightarrow{\text{Encoding}} \mathbf{x}_T \xrightarrow{\text{Decoding } \theta} \mathbf{x}_{T-1} \quad \cdots \quad \xrightarrow{\text{Decoding } \theta} \hat{\mathbf{x}}_0$$

$$\sum_{t=2}^{T} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}\left[\mathrm{D}_{\mathrm{KL}}\left(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \,||\, p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)\right)\right]$$

$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$

$\mathbf{x}_0$

$\mathbf{x}_{t-1}$

$\mathbf{x}_t$

$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$

$\mathbf{x}_{t-1}$

$\mathbf{x}_t$

$$\mathbf{x}_0 \xrightarrow{\text{Encoding}} \mathbf{x}_1 \xrightarrow{\text{Encoding}} \mathbf{x}_2 \cdots \xrightarrow{\text{Encoding}} \mathbf{x}_T \xrightarrow{\text{Decoding } \theta} \mathbf{x}_{T-1} \quad \cdots \quad \xrightarrow{\text{Decoding } \theta} \hat{\mathbf{x}}_0$$

$$\sum_{\mathrm{t=2}}^{\mathrm{T}} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}[\mathrm{D}_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))]$$

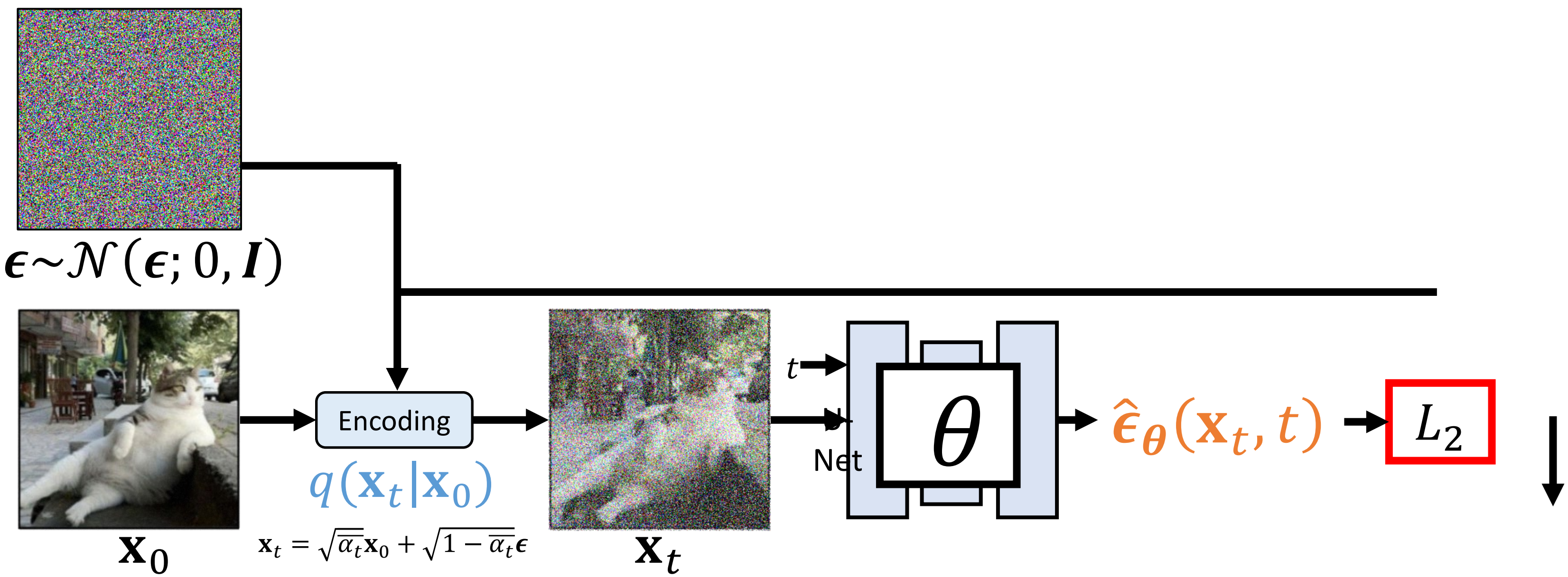
$q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)$

$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$

Observation #2

$$\mathbf{x}_0 \xrightarrow{\text{Encoding}} \mathbf{x}_1 \xrightarrow{\text{Encoding}} \mathbf{x}_2 \cdots \xrightarrow{\text{Encoding}} \mathbf{x}_T \xrightarrow[\ ]{\text{Decoding } \theta} \mathbf{x}_{T-1} \quad \cdots \quad \xrightarrow[\ ]{\text{Decoding } \theta} \hat{\mathbf{x}}_0$$

$$\Rightarrow \sum_{t=2}^{T} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}\left[\ w(t) \ \left[\ || \ \hat{\epsilon}_{\theta}(\mathbf{x}_t, t) \ - \ \epsilon \ ||_2^2 \ \right] \right]$$



Training vs. Inference

- Summary

Algorithm 1 Training

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

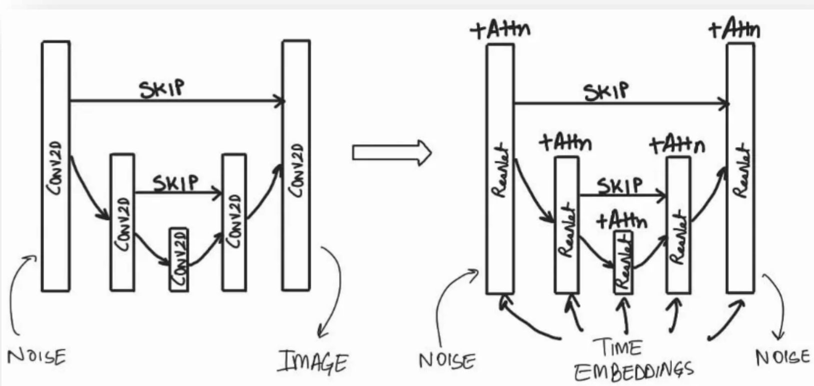
X_t

Algorithm 2 Sampling

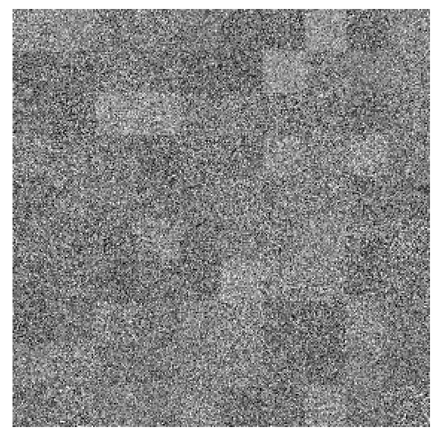
1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
4: $\mathbf{x}_{t-1} = \boxed{\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t,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

allowing generation diversity

$$x_t \xrightarrow{\text{model}} \epsilon_\theta(x_t, t) \xrightarrow{P(x_t|x_0) \to P(x_0|x_t, \epsilon_\theta)} \hat{x}_0(x_t, \epsilon_\theta) \longrightarrow \mu(x_t, \hat{x}_0), \beta_t \xrightarrow{P(x_{t-1}|x_t, x_0)} \hat{x}_{t-1}$$



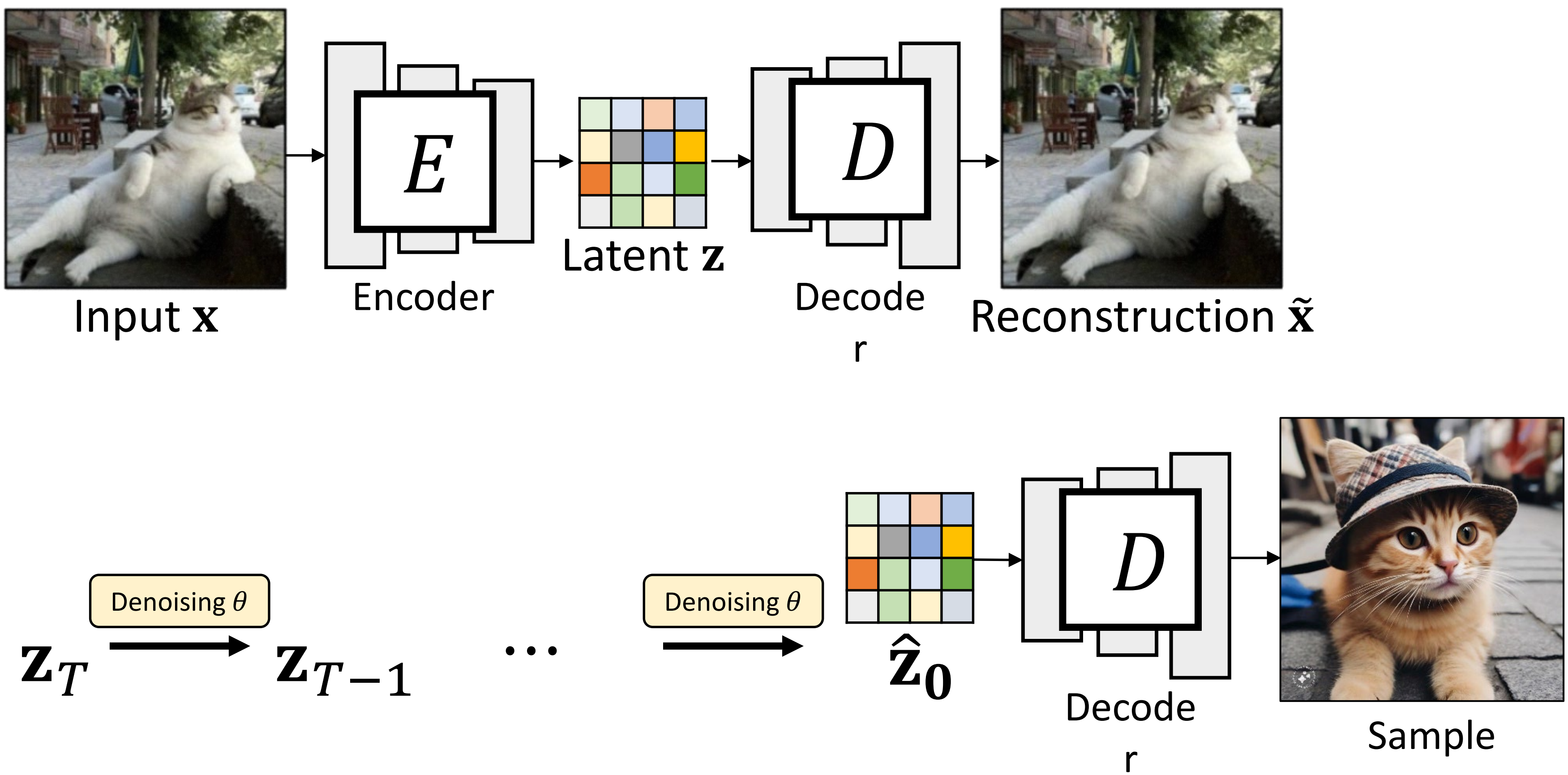
Training U-Net for DDPM noise prediction



MNIST handwritten image data

Slide credit: Kreis, Gao, & Vahdat

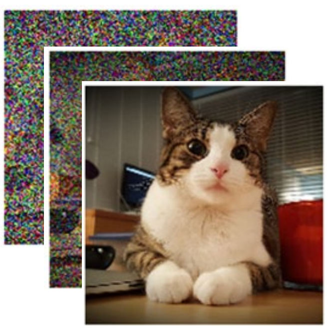
Latent Diffusion Models



From DDPM to DDIM: Denoising Diffusion Implict Models

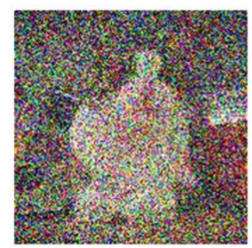
- DDIM
  - Sampling process for generation

$$\boldsymbol{x}_{t-1} = \sqrt{\alpha_{t-1}} \underbrace{\left(\frac{\boldsymbol{x}_t - \sqrt{1-\alpha_t}\,\epsilon_\theta^{(t)}(\boldsymbol{x}_t)}{\sqrt{\alpha_t}} \right)}_{\text{“ predicted } \boldsymbol{x}_0\text{”}} + \underbrace{\sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\epsilon_\theta^{(t)}(\boldsymbol{x}_t)}_{\text{“direction pointing to } \boldsymbol{x}_t\text{”}} + \underbrace{\sigma_t\epsilon_t}_{\text{random noise}}$$

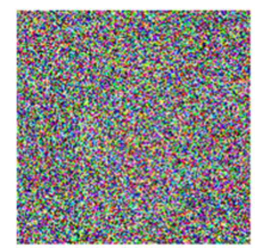


Step 0

Step 123



Step 456

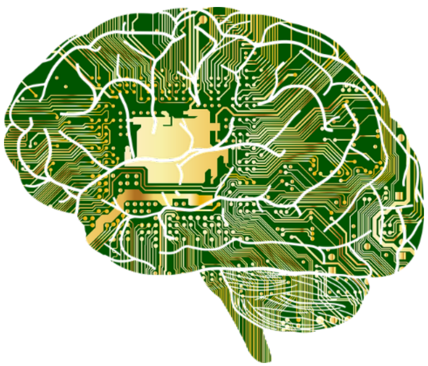


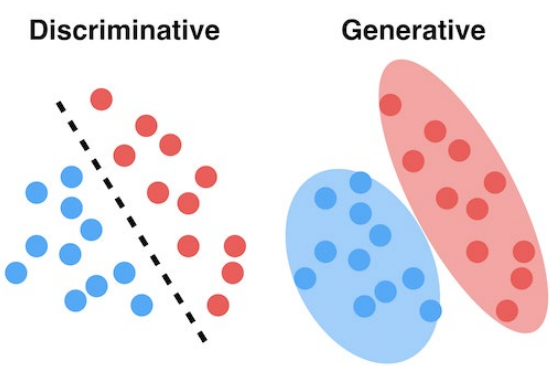
Step 999

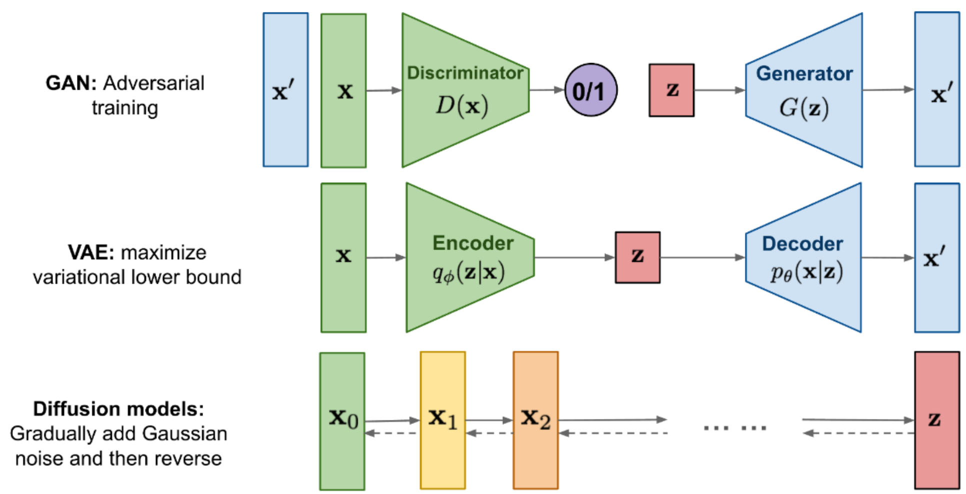
- Additional comment on σ_t^2 : *stochastic* vs. *deterministic* generation process
- Since DDIM and DDPM share the same objective function,
  so one can use a ***pretrained*** DDPM for DDIM generation.

What’s to Be Covered Today…

- Generative Models
  - Diffusion Model
    - Conditional Diffusion Model
      - Classifier Guidance
    - Personalization via Diffusion Model
  - Generative Adversarial Network

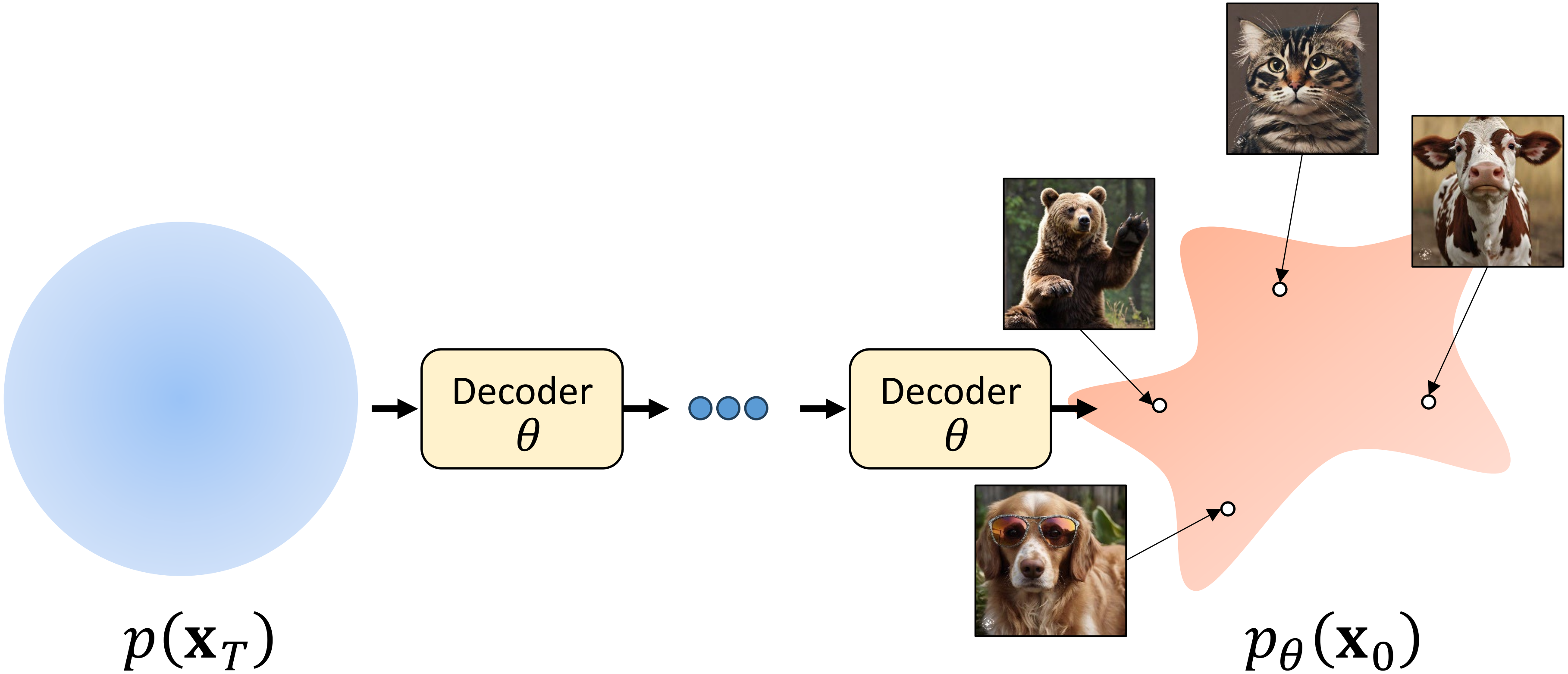




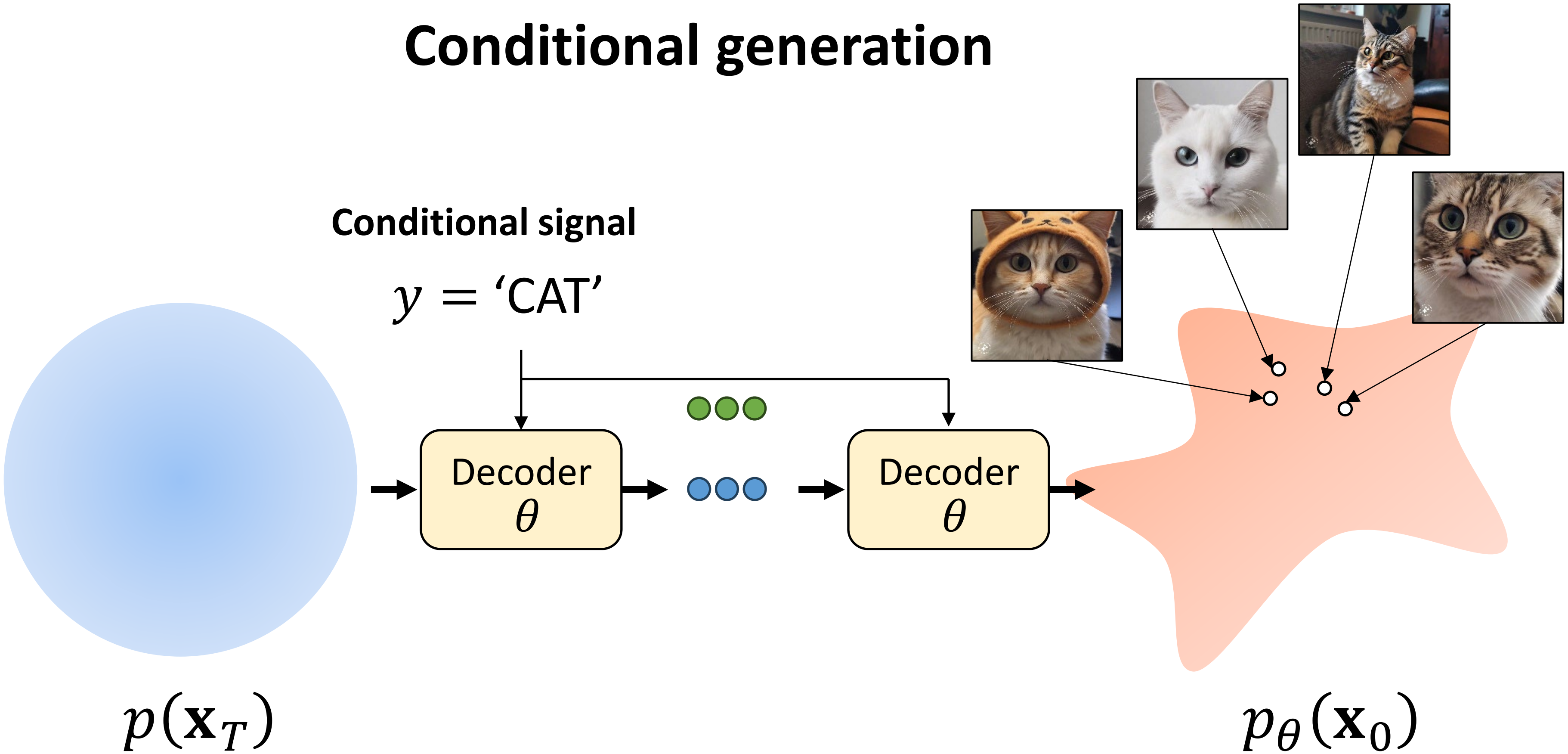


Slide credit: Prof. J.-B. Huang @ U Maryland

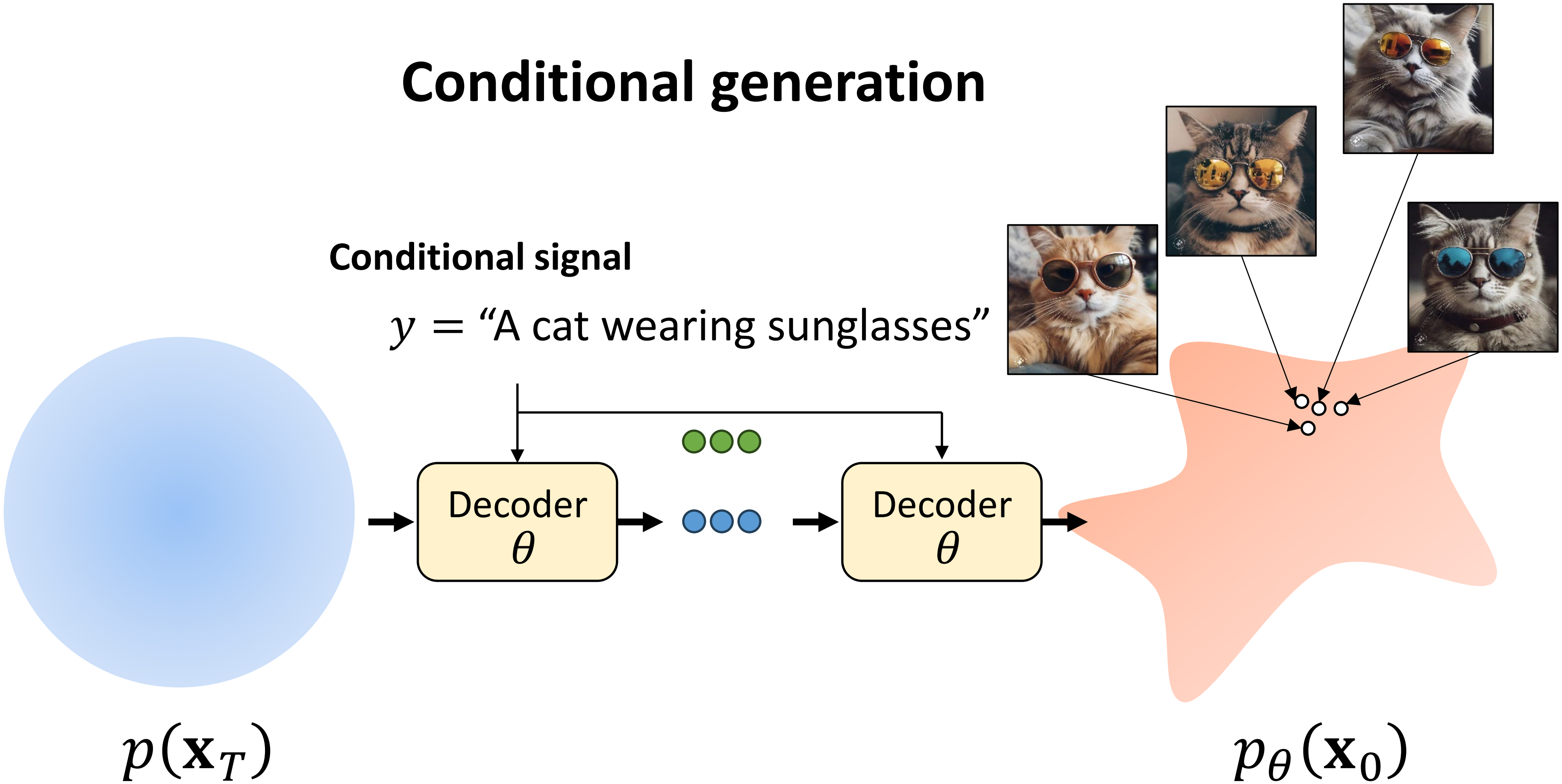
Unconditional generation



Conditional generation



Conditional generation



Unconditional generation

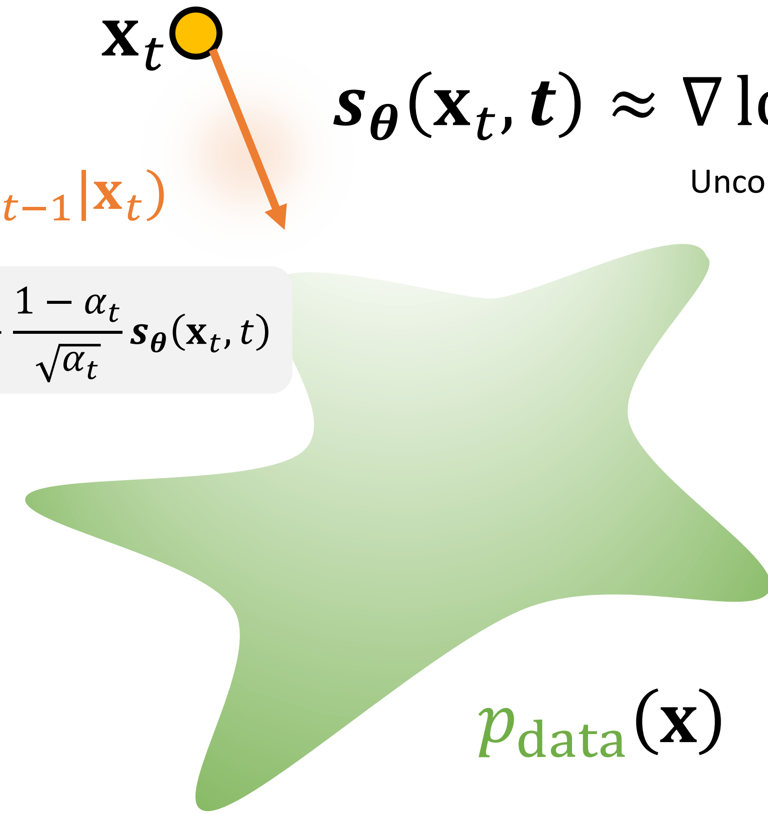
$\mathbf{x}_t$

$$\boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, \boldsymbol{t}) \approx \nabla \log p(\mathbf{x}_t)$$

Unconditional score

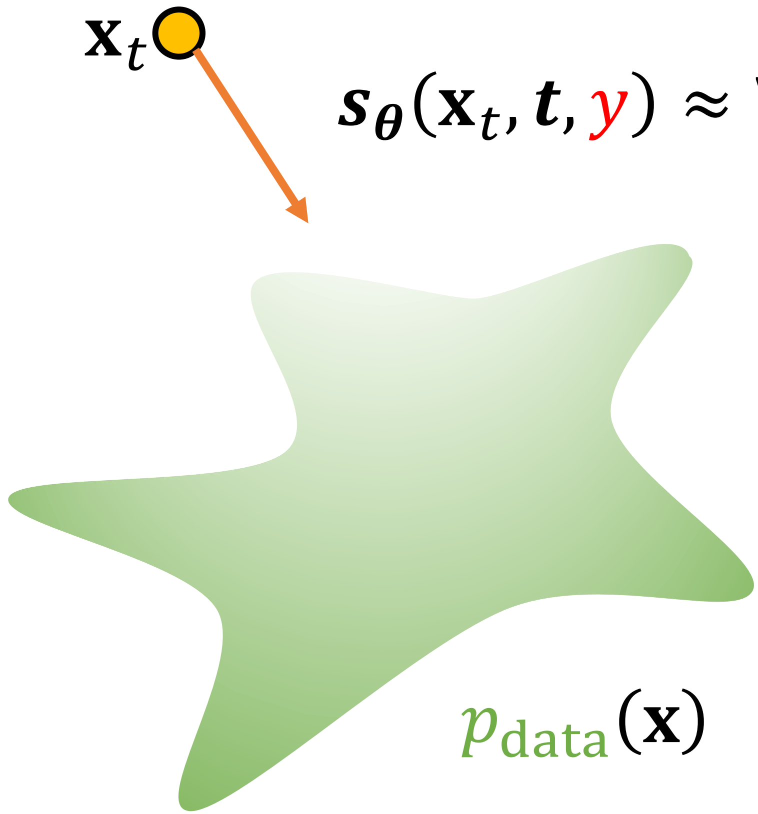
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$

$$\mu_\theta(\mathrm{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\mathbf{x}_t + \frac{1-\alpha_t}{\sqrt{\alpha_t}}\boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$$



$p_{\mathrm{data}}(\mathbf{x})$

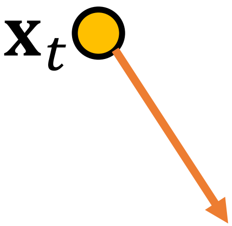
Conditional generation



$$\boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, \boldsymbol{t}, y) \approx \nabla \log p(\mathbf{x}_t | y)$$

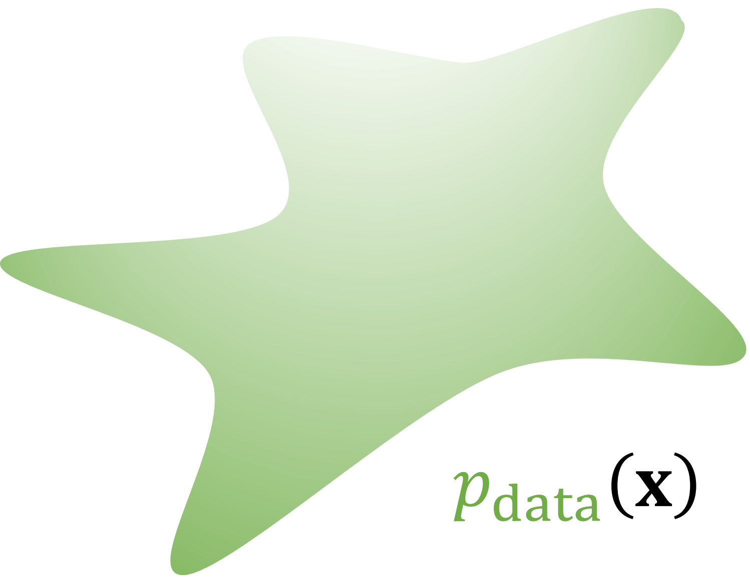
Conditional score

Conditional generation

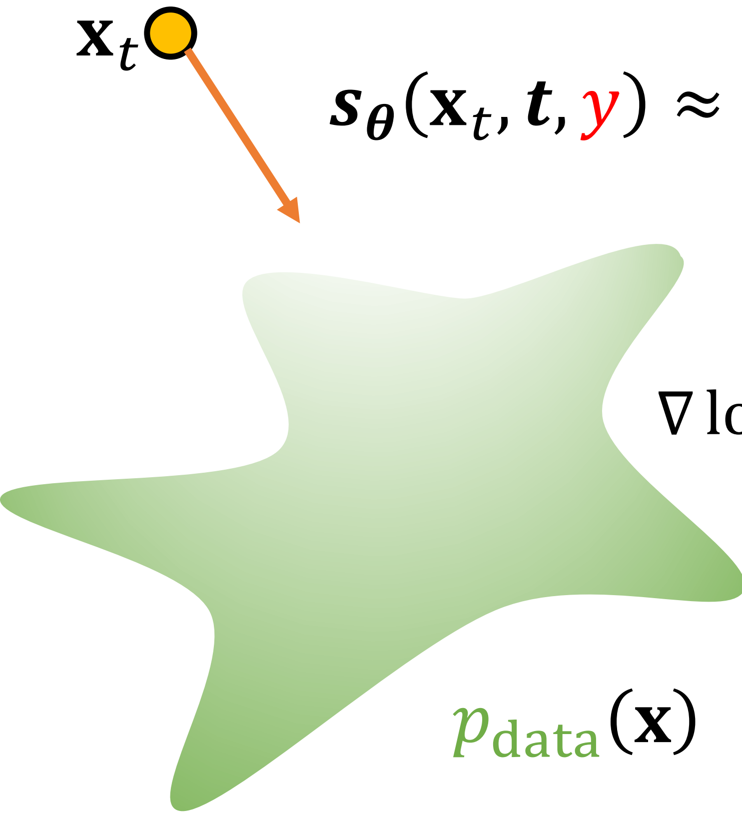


$$\boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, \boldsymbol{t}, y) \approx \nabla \log p(\mathbf{x}_t | y)$$

Conditional score



Conditional generation



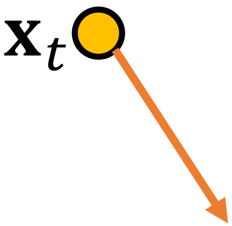
$$\boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, \boldsymbol{t}, y) \approx \nabla \log p(\mathbf{x}_t | y)$$

Conditional score

$$\nabla \log p(\mathbf{x}_t | y) = \nabla \log \left(\frac{p(\mathbf{x}_t) p(y | \mathbf{x}_t)}{p(y)} \right)$$

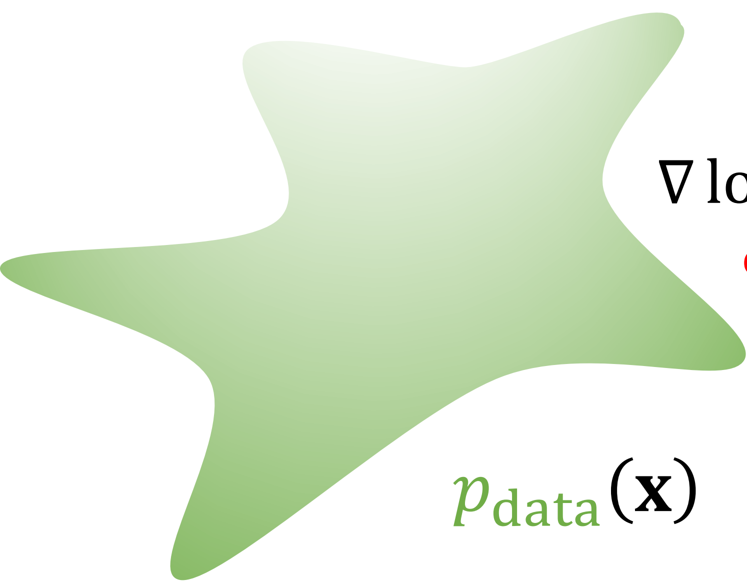
Conditional score

Conditional generation



$$\boldsymbol{s_\theta}(\mathbf{x}_t, \boldsymbol{t}, y) \approx \nabla \log p(\mathbf{x}_t | y)$$

Conditional score

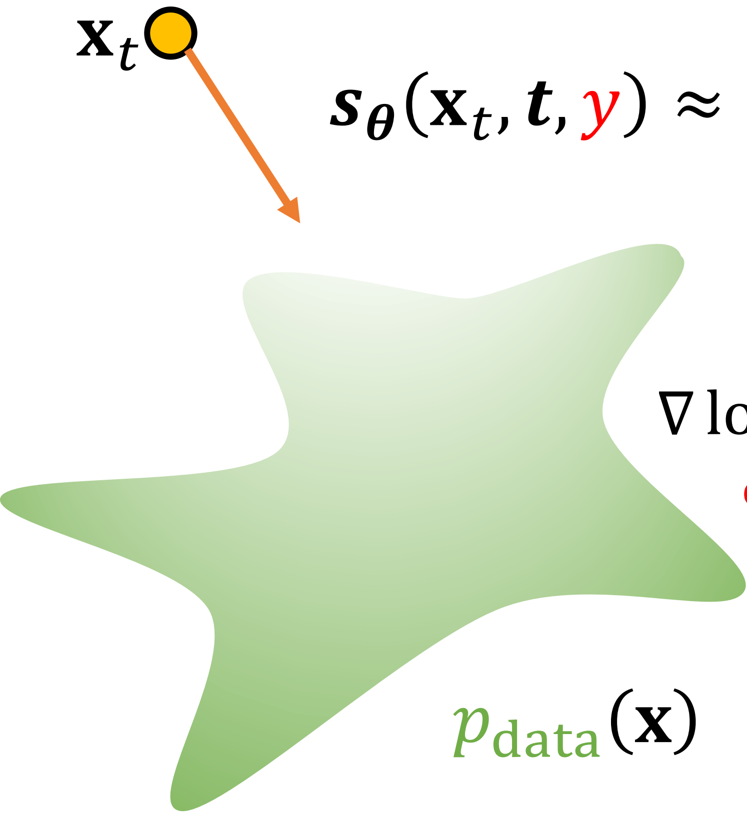


$$\nabla \log p(\mathbf{x}_t | y) = \nabla \log p(\mathbf{x}_t) + \nabla \log p(y | \mathbf{x}_t) - \nabla \log p(y)$$

Conditional score

Conditional generation

Classifier Guidance



$$\boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, \boldsymbol{t}, y) \approx \nabla \log p(\mathbf{x}_t | y)$$

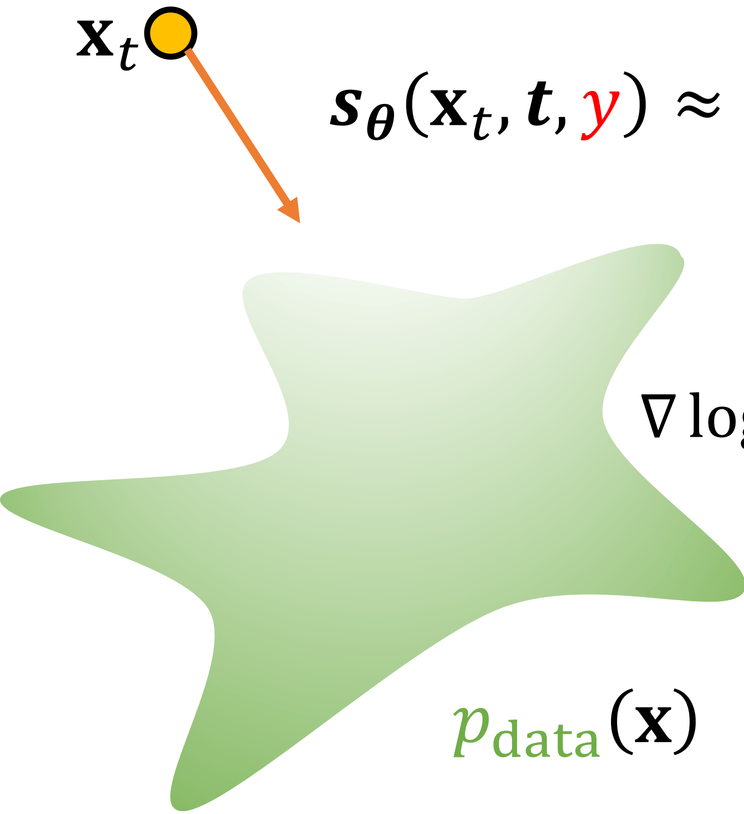
Conditional score

$$\nabla \log p(\mathbf{x}_t | y) = \nabla \log p(\mathbf{x}_t) + \nabla \log p(y | \mathbf{x}_t)$$

Conditional score — Unconditional score — Adversarial gradient

Conditional generation

Classifier Guidance



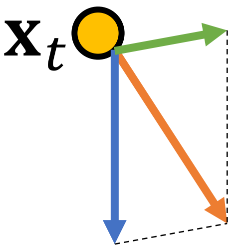
$$\boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, \boldsymbol{t}, {\color{red}y}) \approx \nabla \log p(\mathbf{x}_t | {\color{red}y})$$

Conditional score

$$\underbrace{\nabla \log p_\gamma(\mathbf{x}_t | {\color{red}y})}_{\text{Conditional score}} = \underbrace{\nabla \log p(\mathbf{x}_t)}_{\text{Unconditional score}} + \gamma \underbrace{\nabla \log p({\color{red}y} | \mathbf{x}_t)}_{\text{Adversarial gradient}}$$

Conditional generation

Classifier Guidance



$$\boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, \boldsymbol{t}, y) \approx \nabla \log p(\mathbf{x}_t | y)$$

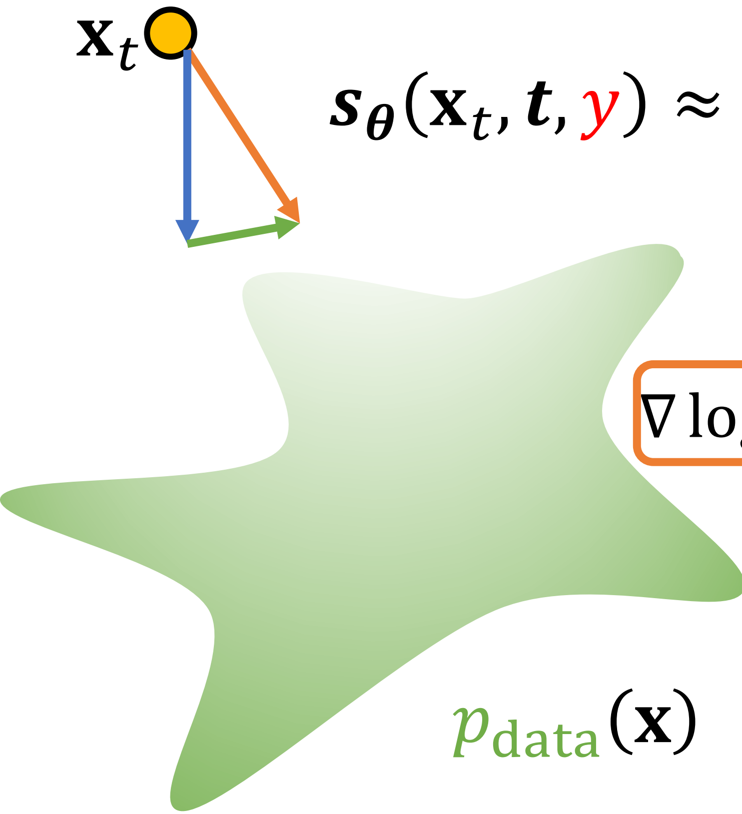
Conditional score

$$\boxed{\nabla \log p_\gamma(\mathbf{x}_t | y)} = \boxed{\nabla \log p(\mathbf{x}_t)} + \boxed{\gamma \ \nabla \log p(\mathrm{y} | \mathbf{x}_t)}$$

Conditional score | Unconditional score | Adversarial gradient

Conditional generation

Classifier Guidance



$$\boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, \boldsymbol{t}, y) \approx \nabla \log p(\mathbf{x}_t | y)$$

Conditional score

$$\boxed{\nabla \log p_\gamma(\mathbf{x}_t | y)} = \boxed{\nabla \log p(\mathbf{x}_t)} + \boxed{\gamma \ \nabla \log p(y | \mathbf{x}_t)}$$

Conditional score — Unconditional score — Adversarial gradient

Conditional generation

Classifier

Classifier guidance scale = 1

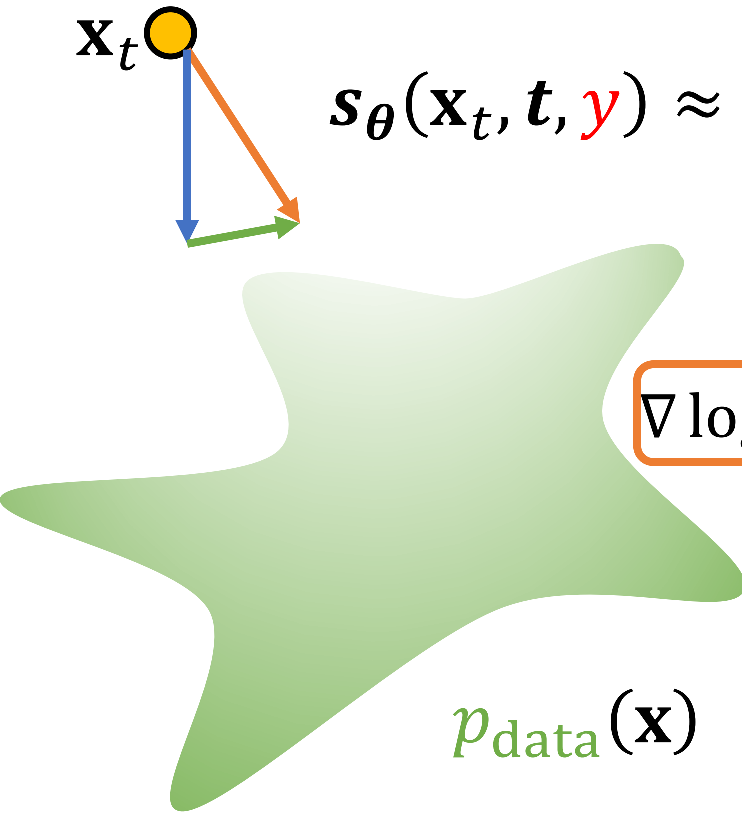
Classifier guidance scale = 10

y = “Pembroke Welsh corgi”

[Dhariwal & Nichol 2021]

Conditional generation

Classifier Guidance



$$\boldsymbol{s_\theta}(\mathbf{x}_t, \boldsymbol{t}, y) \approx \nabla \log p(\mathbf{x}_t | y)$$

Conditional score

$$\boxed{\nabla \log p_\gamma(\mathbf{x}_t | y)} = \boxed{\nabla \log p(\mathbf{x}_t)} + \boxed{\gamma \ \nabla \log p(y | \mathbf{x}_t)}$$

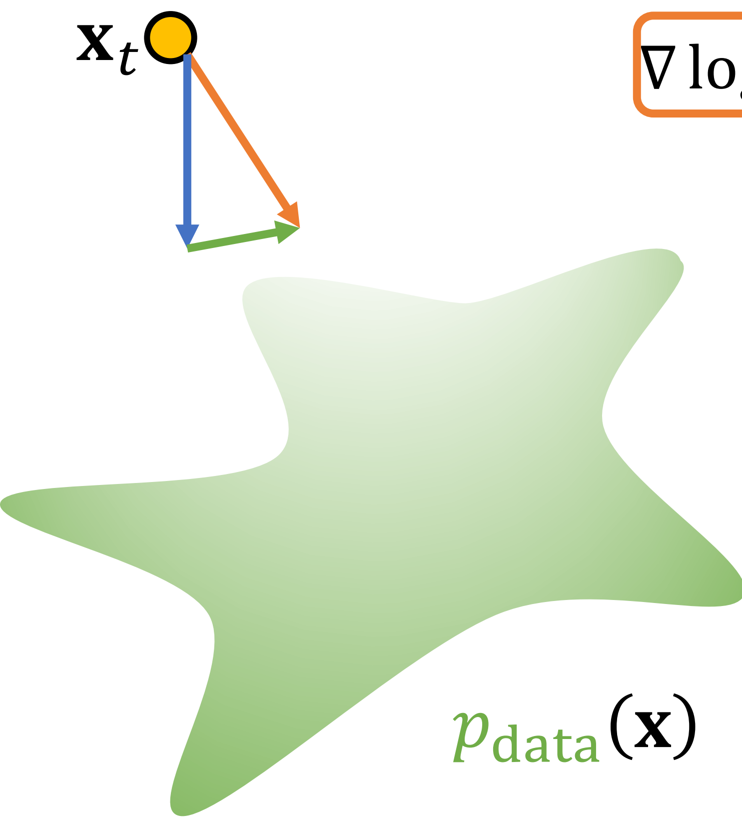
Conditional score — Unconditional score — Adversarial gradient

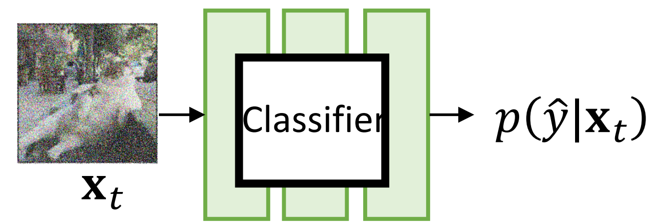
Conditional generation

Classifier Guidance

$$\boxed{\nabla \log p_\gamma(\mathbf{x}_t|y)} = \boxed{\nabla \log p(\mathbf{x}_t)} + \boxed{\gamma\ \nabla \log p(y|\mathbf{x}_t)}$$

Conditional score — Unconditional score — Adversarial gradient

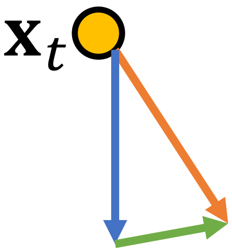




Training optional*

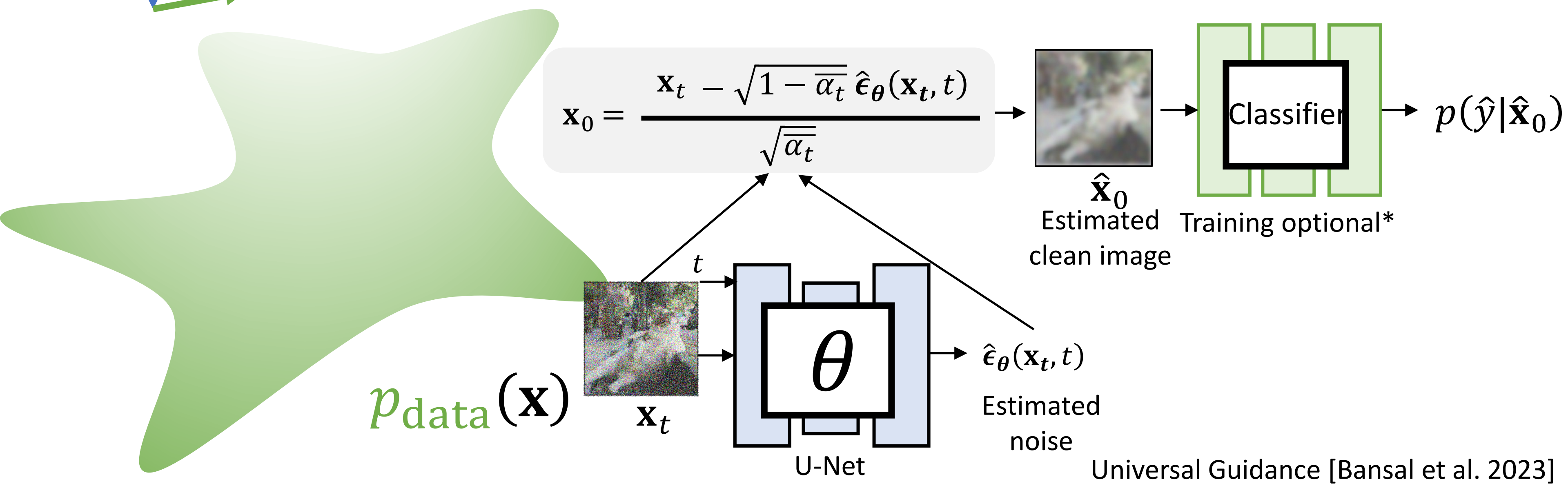
Conditional generation

Classifier Guidance



$$\boxed{\nabla \log p_\gamma(\mathbf{x}_t|y)} = \boxed{\nabla \log p(\mathbf{x}_t)} + \boxed{\gamma\ \nabla \log p(y|\hat{\mathbf{x}}_0)}$$

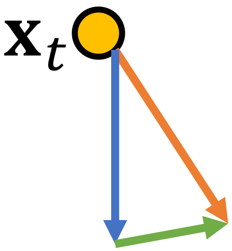
Conditional score | Unconditional score | Adversarial gradient



Universal Guidance [Bansal et al. 2023]

Conditional Generation with Classifier-Guidance

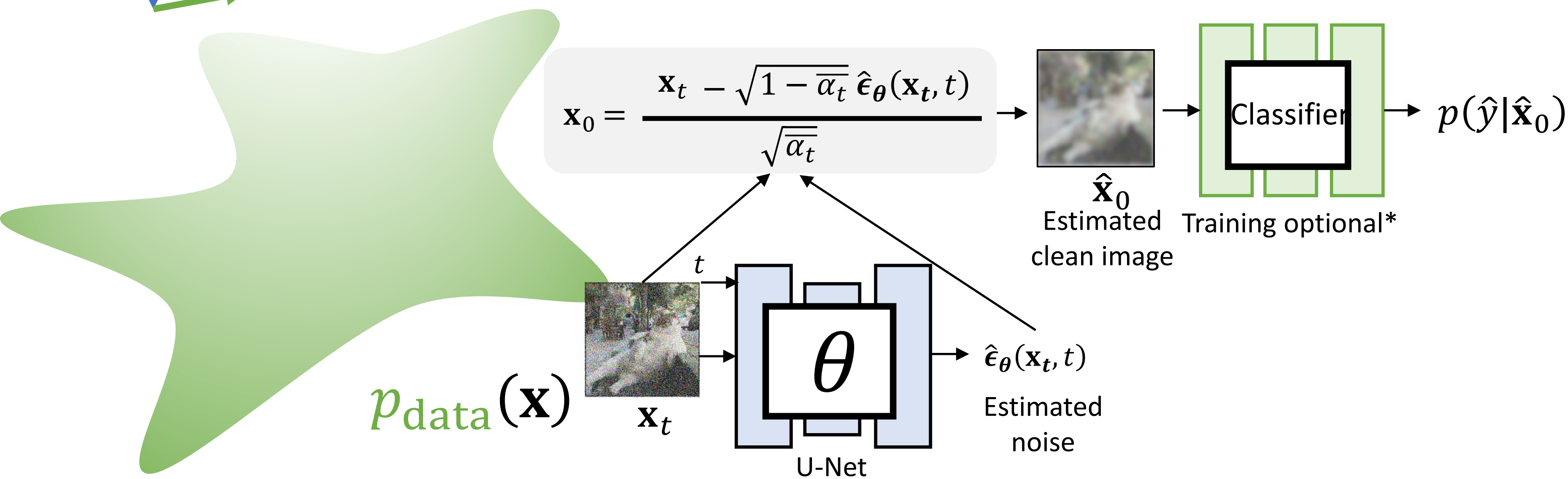
- Any price to pay?



Classifier Guidance

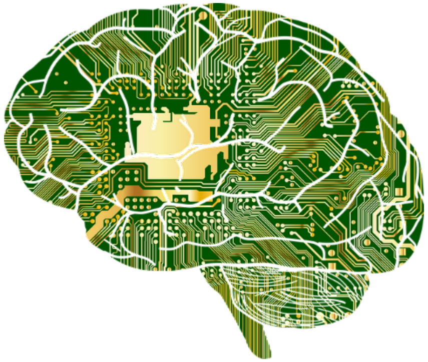
$$\boxed{\nabla \log p_{\gamma}(\mathbf{x}_t | y)} = \boxed{\nabla \log p(\mathbf{x}_t)} + \boxed{\gamma \ \nabla \log p(y | \hat{\mathbf{x}}_0)}$$

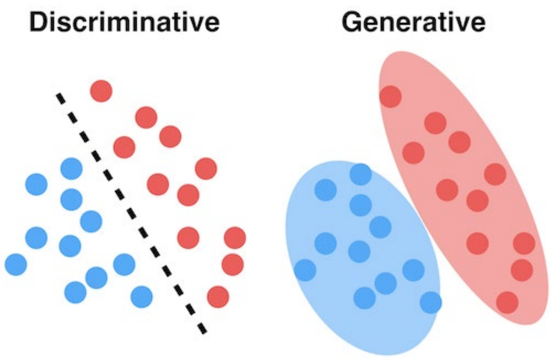
Conditional score | Unconditional score | Adversarial gradient

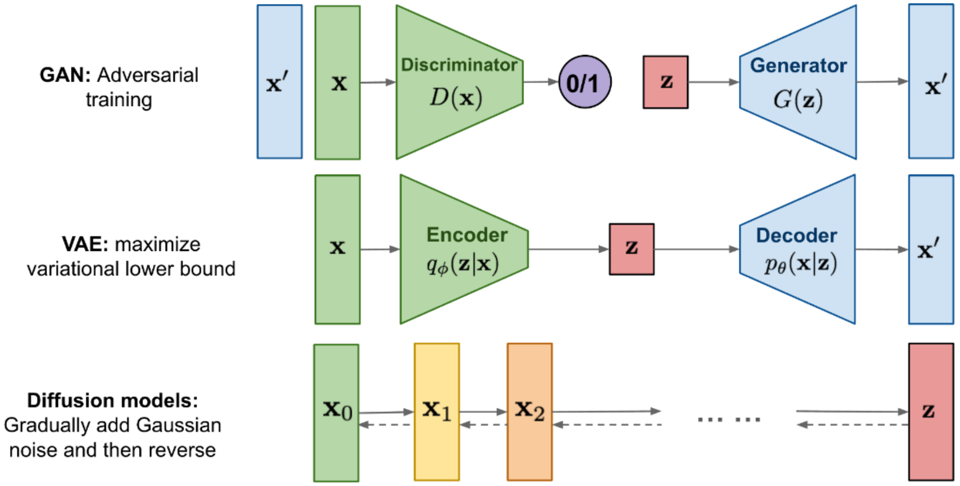


What's to Be Covered Today…

- Generative Models
  - Diffusion Model
    - Conditional Diffusion Model
      - Classifier Guidance
      - Classifier-Free Guidance
    - Personalization via Diffusion Model
  - Generative Adversarial Network





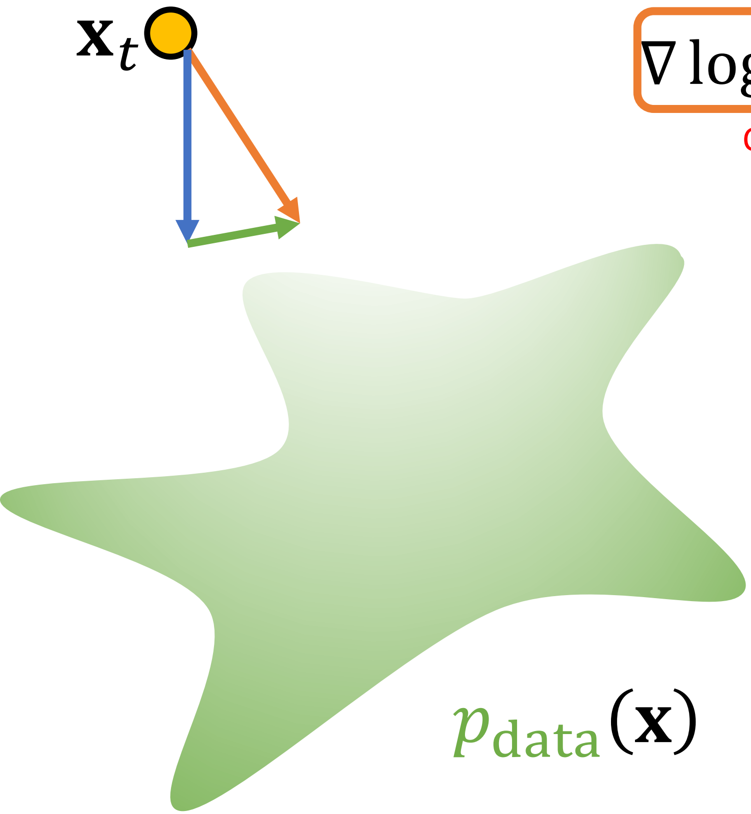


Slide credit: Prof. J.-B. Huang @ U Maryland

Conditional generation

Classifier Guidance

$$\underbrace{\boxed{\nabla \log p_\gamma(\mathbf{x}_t|{\color{red}y})}}_{\text{{\color{red}Conditional} score}} = \underbrace{\boxed{\nabla \log p(\mathbf{x}_t)}}_{\text{Unconditional score}} + \underbrace{\boxed{\gamma\ \nabla \log p({\color{red}y}|\mathbf{x}_t)}}_{\text{Adversarial gradient}}$$



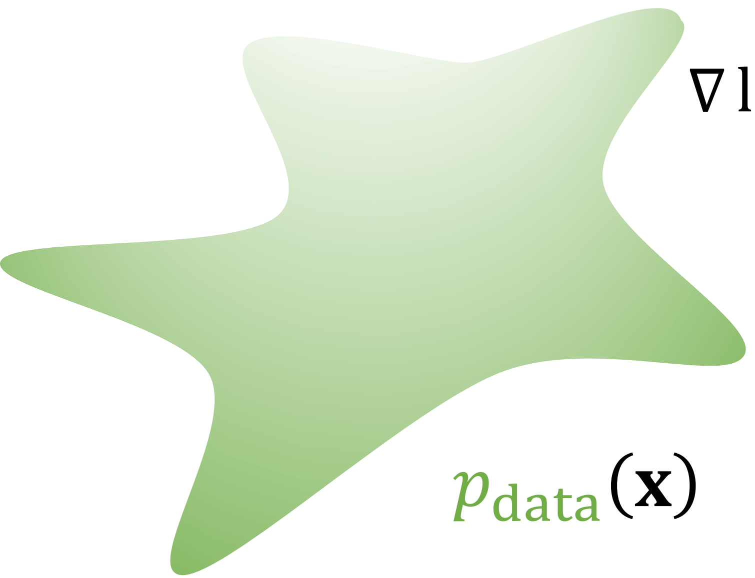
Conditional generation

$\mathbf{x}_t$

$$\nabla \log p_\gamma(\mathbf{x}_t|y) = \nabla \log p(\mathbf{x}_t) + \gamma\, \nabla \log p(y|\mathbf{x}_t)$$

Conditional score — Unconditional score — Adversarial gradient

$$\nabla \log p(y|\mathbf{x}_t) = \nabla \log\left(\frac{p(\mathbf{x}_t|y)\,p(y)}{p(\mathbf{x}_t)}\right)$$



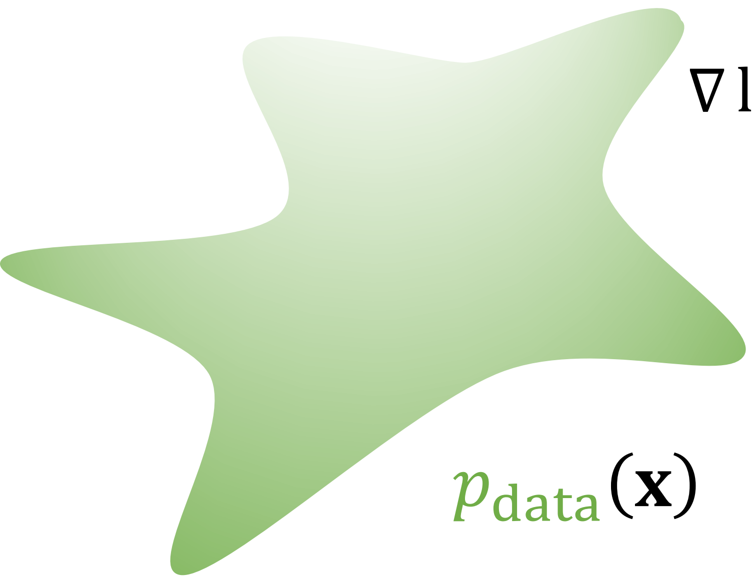
Conditional generation

$\mathbf{x}_t$

$$\nabla \log p_\gamma(\mathbf{x}_t|y) = \nabla \log p(\mathbf{x}_t) + \gamma \, \nabla \log p(y|\mathbf{x}_t)$$

Conditional score | Unconditional score | Adversarial gradient

$$\nabla \log p(y|\mathbf{x}_t) = \nabla \log p(\mathbf{x}_t|y) - \nabla \log p(\mathbf{x}_t) + \nabla \log p(y)$$



Conditional generation

$\mathbf{x}_t$

$$\nabla \log p_\gamma(\mathbf{x}_t|y) = \nabla \log p(\mathbf{x}_t) + \gamma\ \nabla \log p(y|\mathbf{x}_t)$$

Conditional score | Unconditional score | Adversarial gradient

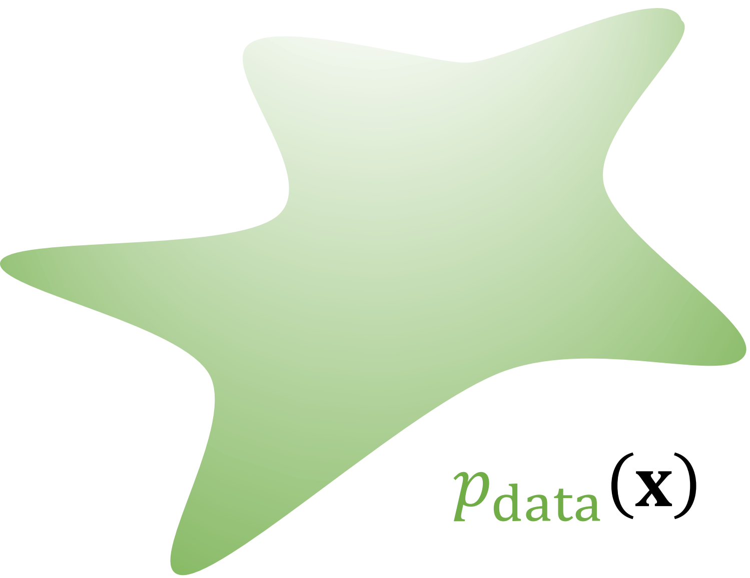
$$\nabla \log p(y|\mathbf{x}_t) = \nabla \log p(\mathbf{x}_t|y) - \nabla \log p(\mathbf{x}_t)$$

Conditional generation

$\mathbf{x}_t$

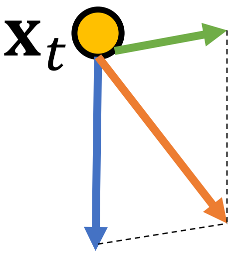
$$\nabla \log p_\gamma(\mathbf{x}_t|y) = \nabla \log p(\mathbf{x}_t) + \gamma(\nabla \log p(\mathbf{x}_t|y) - \nabla \log p(\mathbf{x}_t))$$

Conditional score — Unconditional score — Adversarial gradient

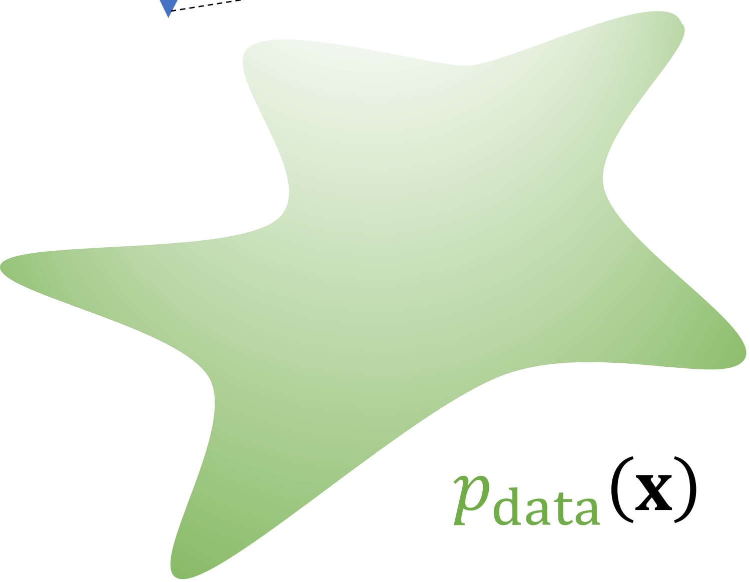


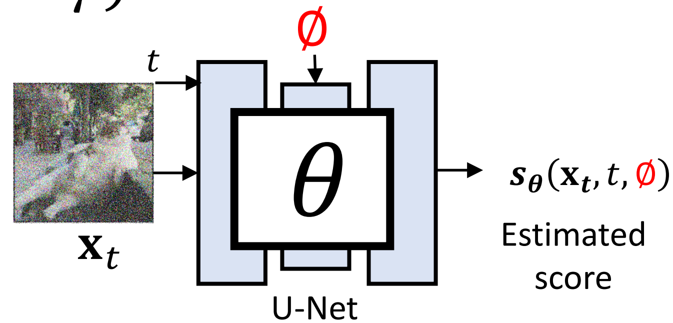
Conditional generation

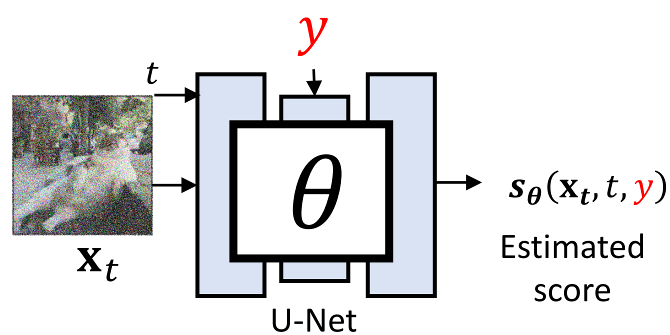
Classifier-free Guidance



$$\underbrace{\nabla \log p_\gamma(\mathbf{x}_t|y)}_{\text{Conditional score}} = \underbrace{(1-\gamma)\,\nabla \log p(\mathbf{x}_t)}_{\text{Unconditional score}} + \underbrace{\gamma\,\nabla \log p(\mathbf{x}_t|y)}_{\text{Conditional score}}$$



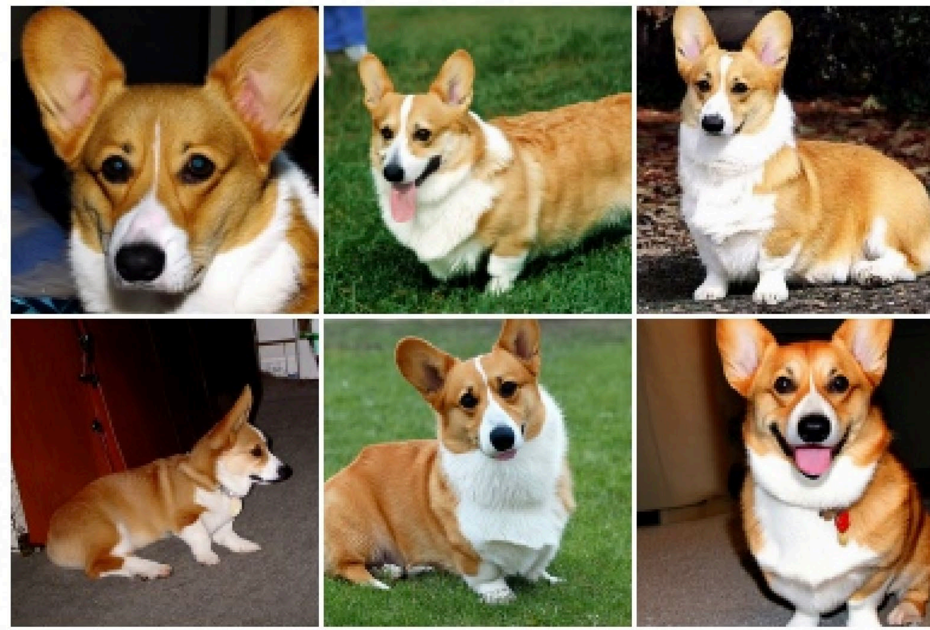




Conditional generation

Classifier-free Guidance

Unguided samples



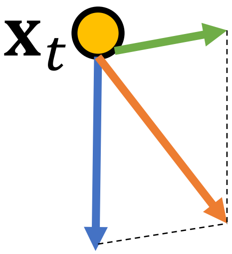
Guided samples

[Ho and Salimans 2021]

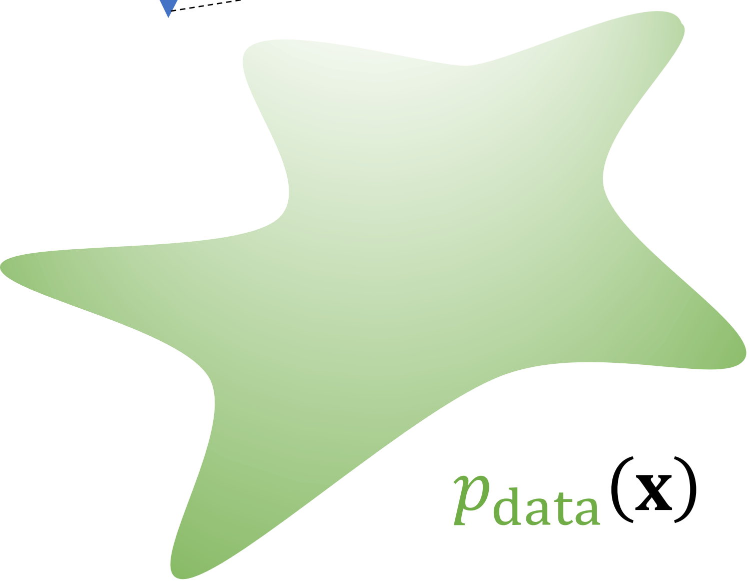
Conditional Generation with Classifier-Free Guidance

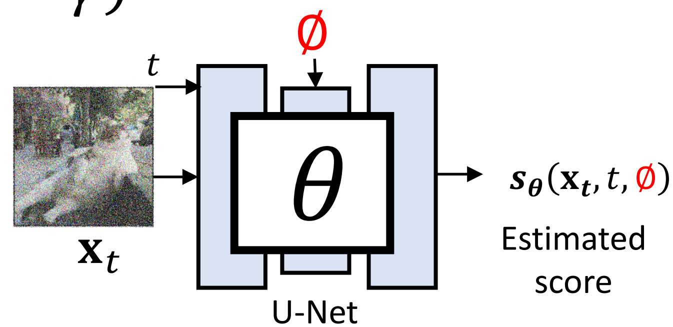
- Any price to pay?

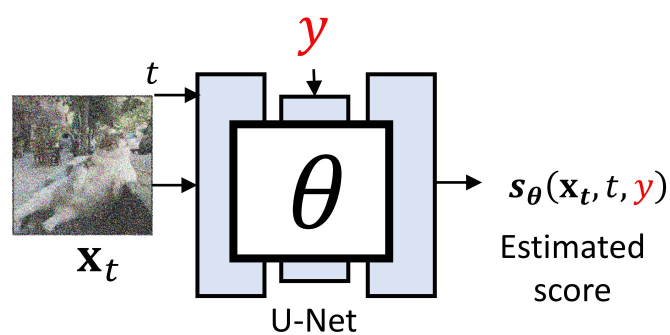
Classifier-free Guidance



$$\underbrace{\boxed{\nabla \log p_\gamma(\mathbf{x}_t|y)}}_{\text{Conditional score}} = \underbrace{\boxed{(1-\gamma)\,\nabla \log p(\mathbf{x}_t)}}_{\text{Unconditional score}} + \underbrace{\boxed{\gamma\,\nabla \log p(\mathbf{x}_t|y)}}_{\text{Conditional score}}$$

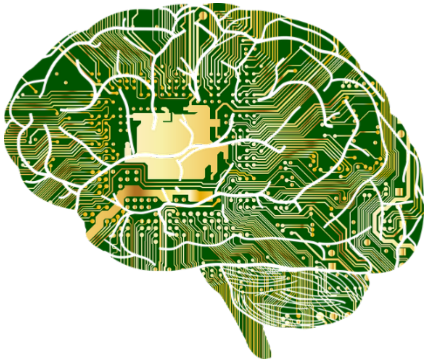


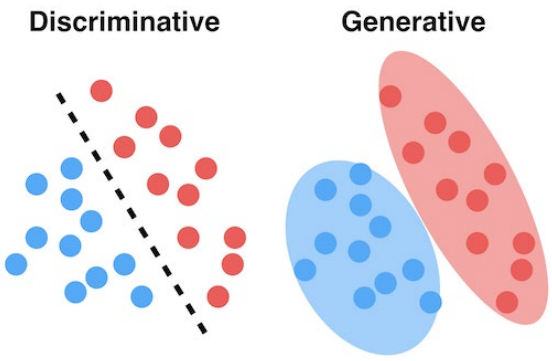


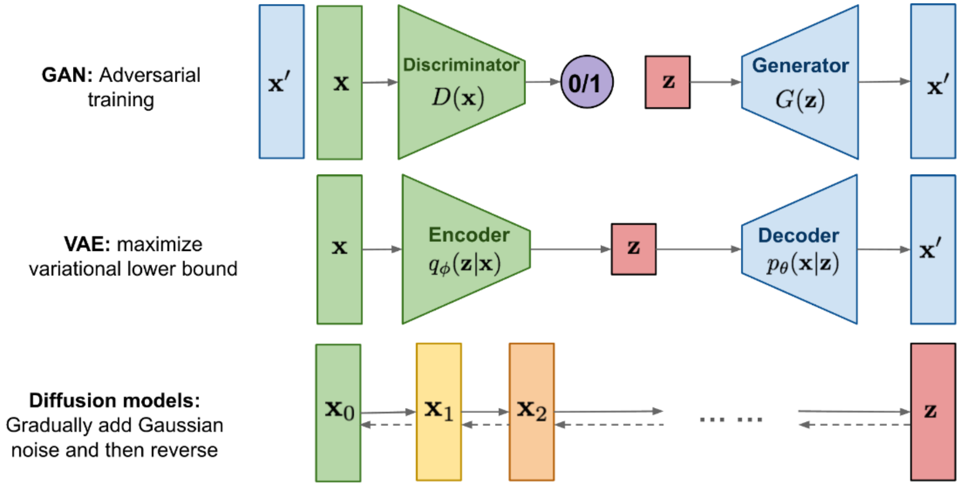


What's to Be Covered Today…

- Generative Models
  - Diffusion Model
    - Conditional Diffusion Model
      - Classifier Guidance
      - Classifier-Free Guidance
      - Text/Image Guidance
    - Personalization via Diffusion Model
  - Generative Adversarial Network

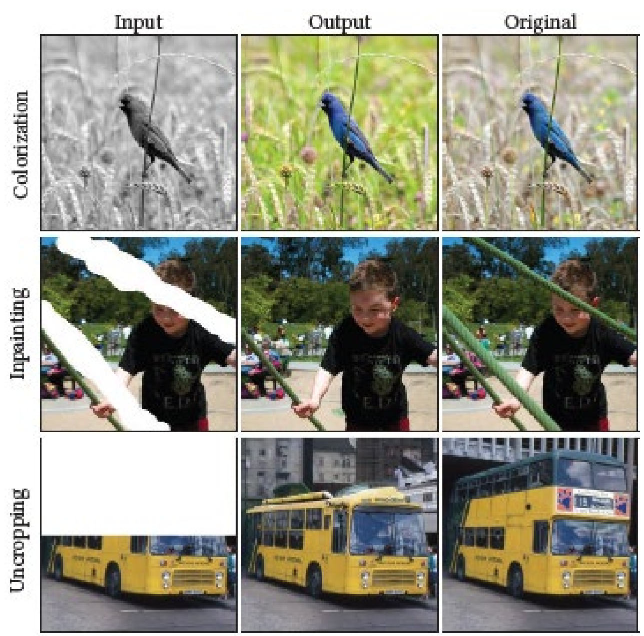


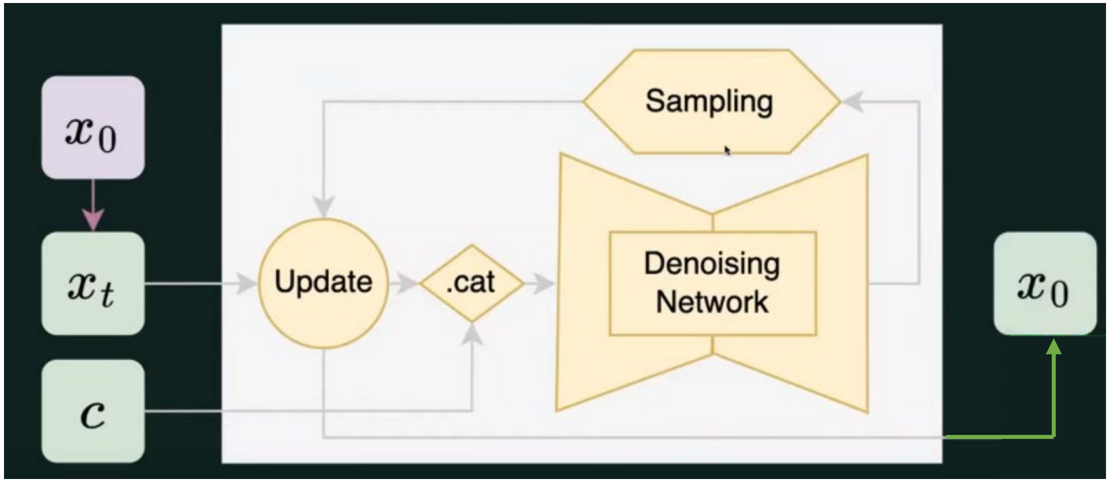




Conditional Generation with Image Guidance

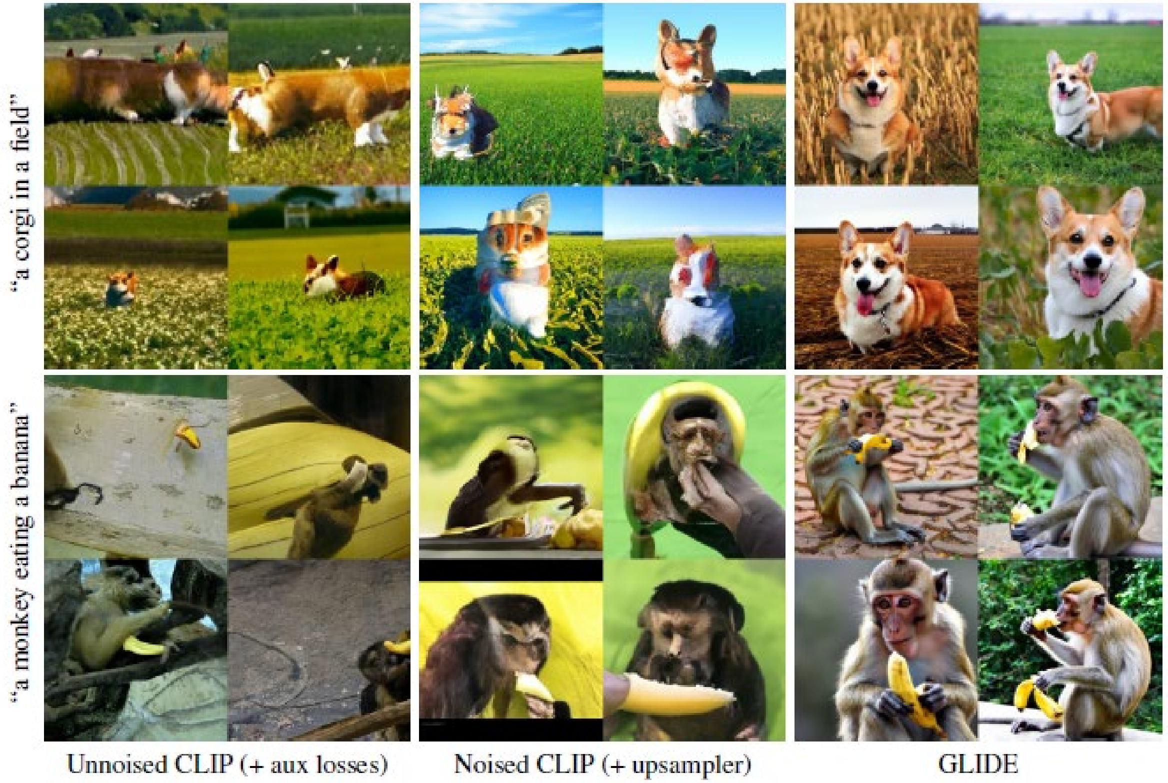
- *Palette: Image-to-Image Diffusion Models*, Google Research, arXiv 2022
  - Applications: colorization, inpainting, outpainting, etc.
  - Input image as condition (via concatenation)





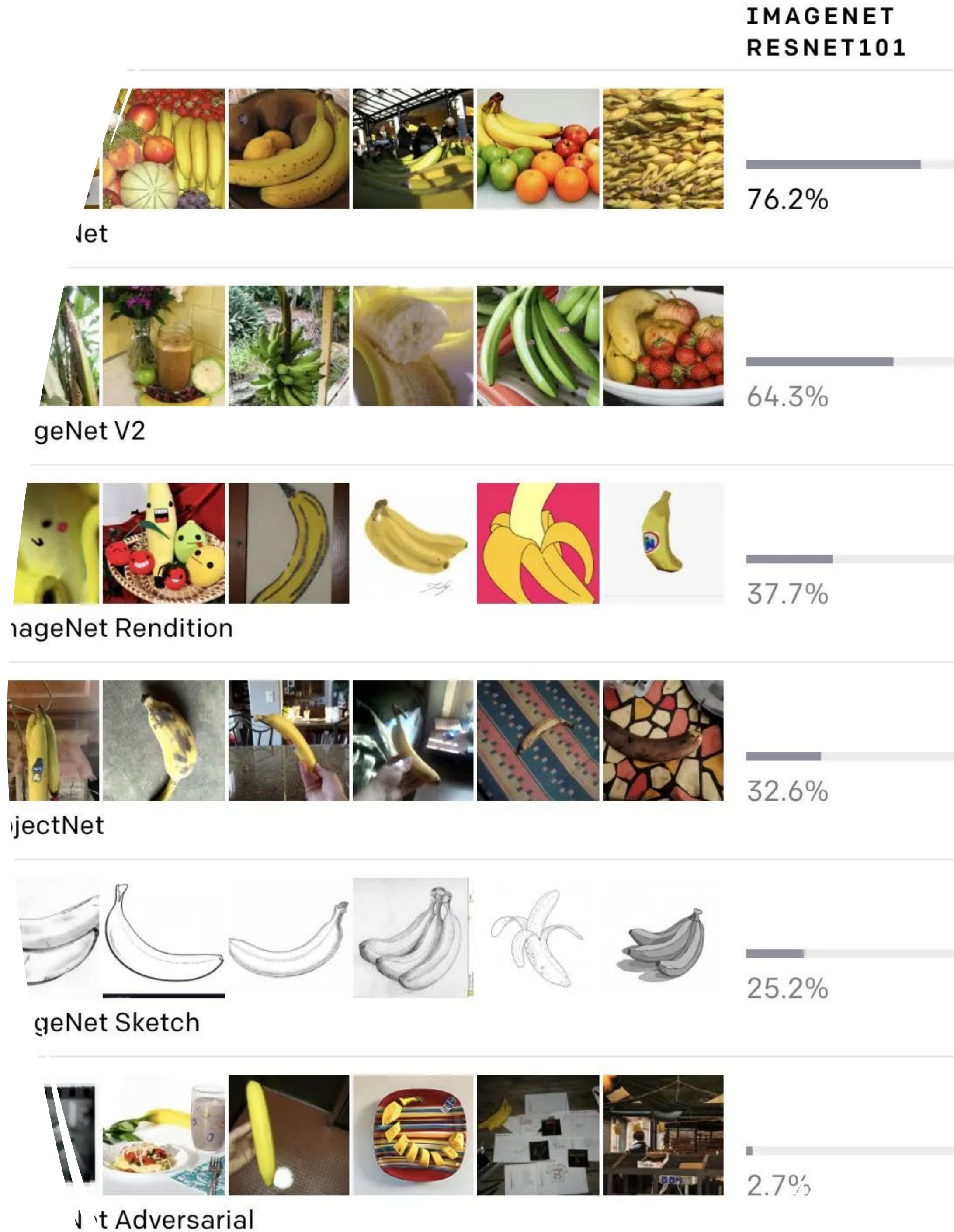
Conditional Generation with Text Guidance

- *GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models*, OpenAI, arXiv 2022
  - CLIP (Contrastive Language-Image Pretraining) is previously proposed to measure alignment between text and image inputs
  - Classifier guidance -> CLIP guidance (not training-free)
  - What is CLIP? (see next slide)



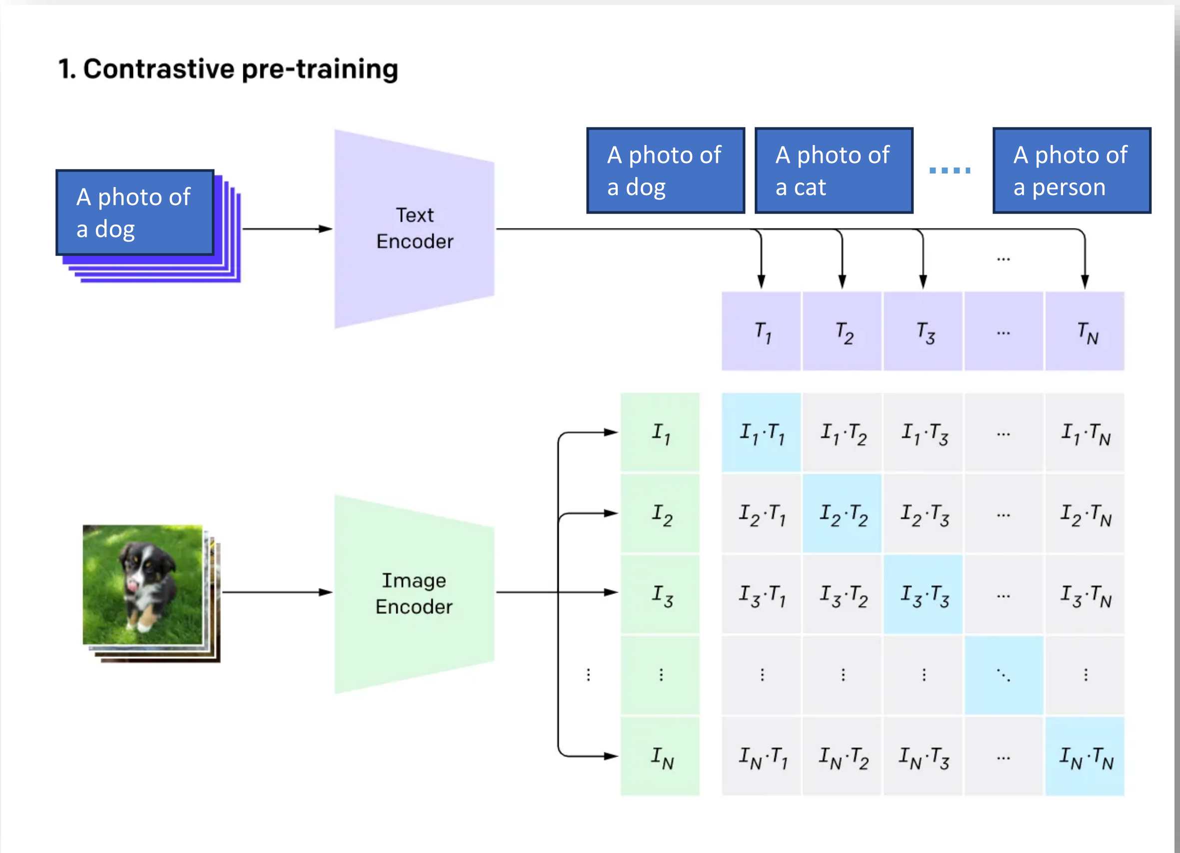
CLIP: Contrastive Language-Image Pretraining

- *Learning Transferable Visual Models From Natural Language Supervision*, OpenAI, NeurIPS WS 2021 (w/ 22000+ citations)
- Why not just CNN?
    - Require **annotated data** for training image classification
    - **Domain gap** between closed and open-world domain data
    - Lack of ability for **zero-shot classification**



CLIP (cont'd)

- Objectives
  - Cross-domain contrastive learning from large-scale image-language data
  - Next-token prediction (what's this & why?);
    will talk more about this for the lecture of Transformer

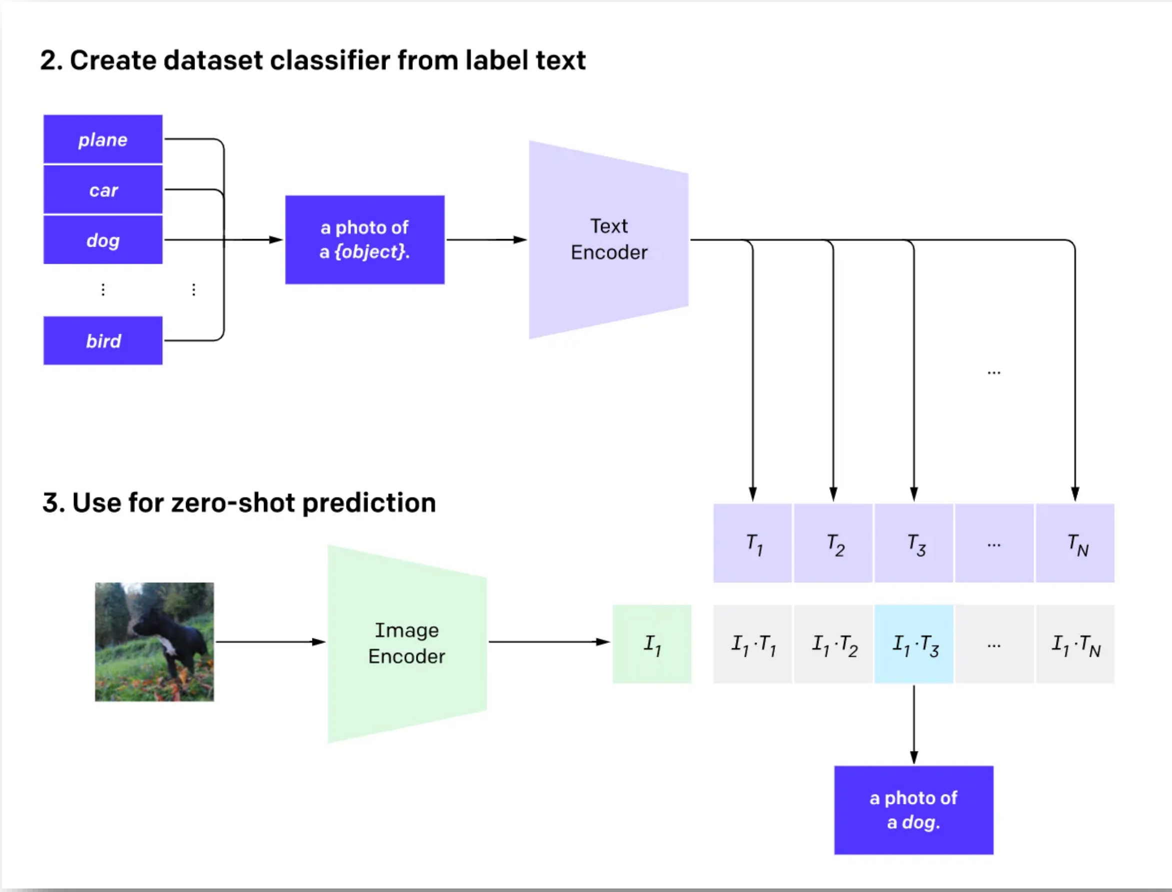


2. Next-token prediction

e.g., a ___;
a photo ____;
a photo of _____, etc.

CLIP (cont'd)

- (Zero-shot) Inference:



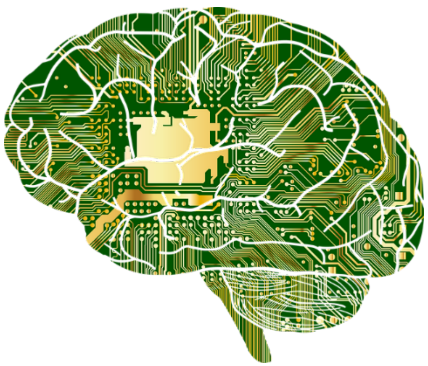
- Potential concerns/disadvantages of CLIP?

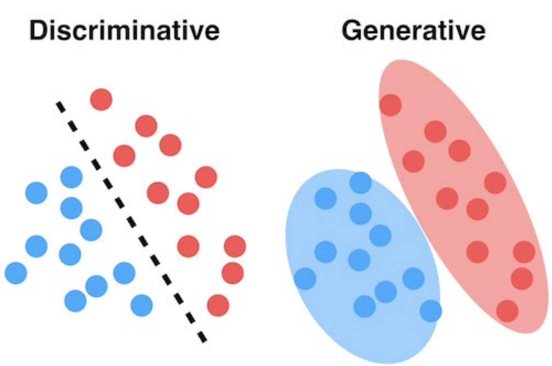
Questions for Image Generation

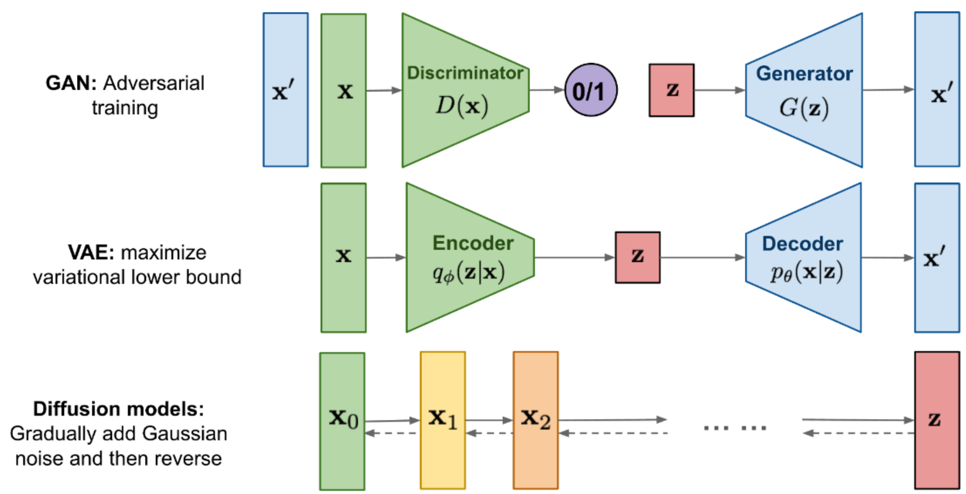
- How to evaluate your unconditional image generation results?
- How to evaluate your conditional image generation results?
- Any objective/subjective and quantitative/qualitative evaluation?

What’s to Be Covered Today…

- Generative Models
  - Diffusion Model
    - Conditional Diffusion Model
    - Personalization via Diffusion Model
  - Generative Adversarial Network







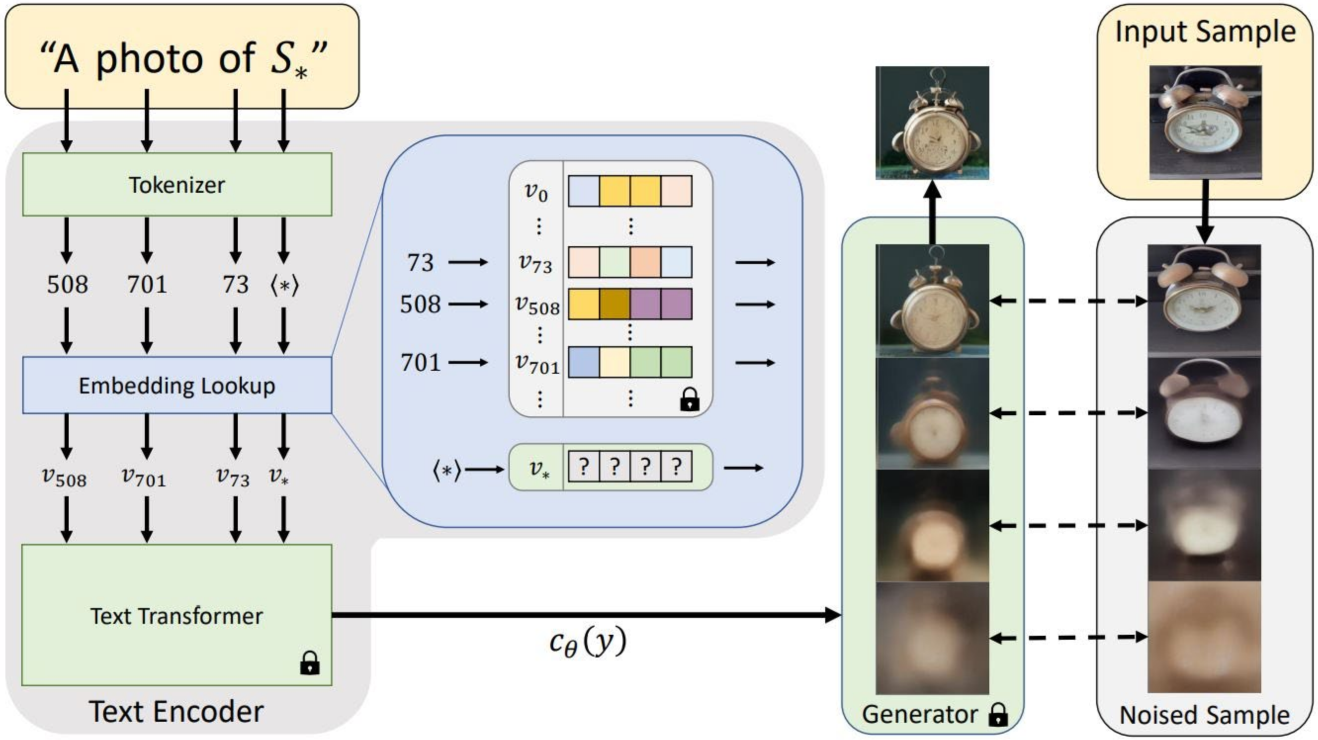
Diffusion Model for Personalization (1): Textual Inversion

- Proposed by NV Research, ICLR 2023
- Goal: Learn a special token (e.g., S*) to represent the concept of interest

An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion

Diffusion Model for Personalization (1): Textual Inversion (cont'd)

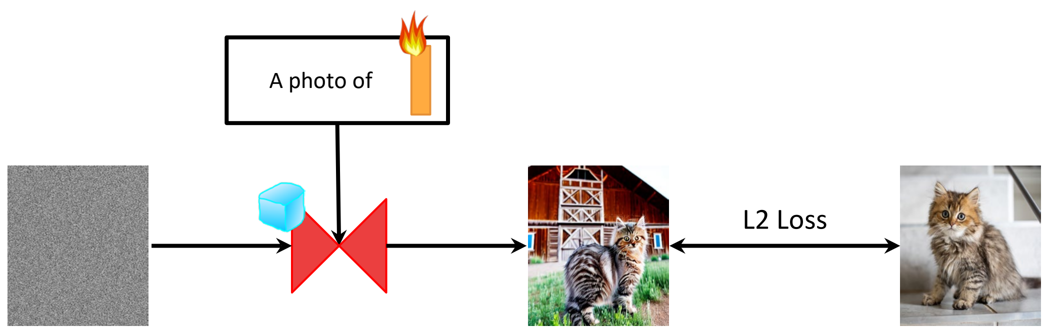
- Learning of special token S*
  - Pre-train and fix text encoder & diffusion model (i.e., generator)
  - Randomly initialize a token as the text encoder input
  - Optimize this token via image reconstruction objectives



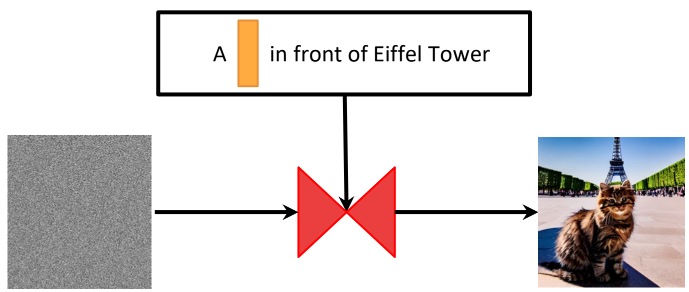
An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion

Diffusion Model for Personalization (1): Textual Inversion (cont'd)

- Learning of special token S*
  - Pre-train and fix text encoder & diffusion model (i.e., generator)
  - Randomly initialize a token as the text encoder input
  - Optimize this token via image reconstruction objectives
- Training:



- Inference:

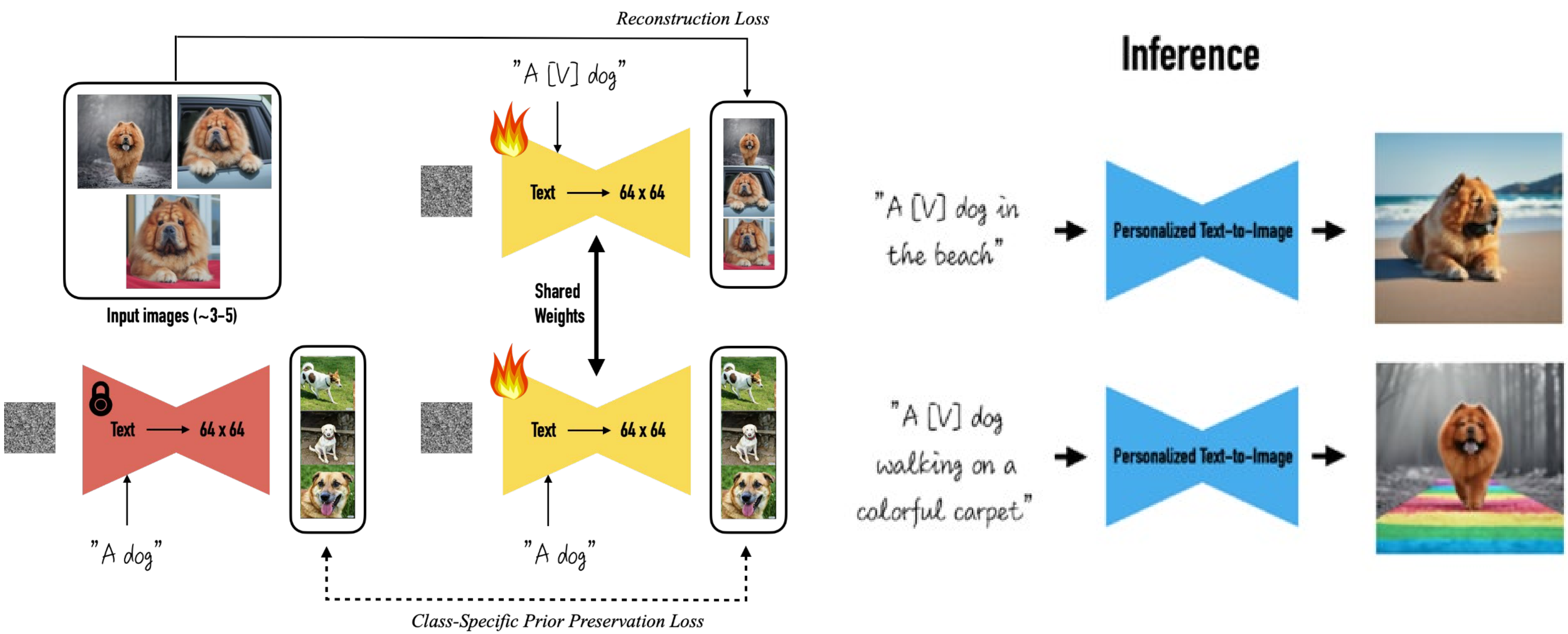


- Any potential concern?

An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion

Diffusion Model for Personalization (2): DreamBooth

- Proposed by Google Research, CVPR 2023
- Finetune the diffusion model w/ a fixed token to represent the image concept
    - Determine and fix a rare token (e.g., [V])
    - Finetune the diffusion model for image restoration objectives
    - Enforce a class-specific prior (why?)

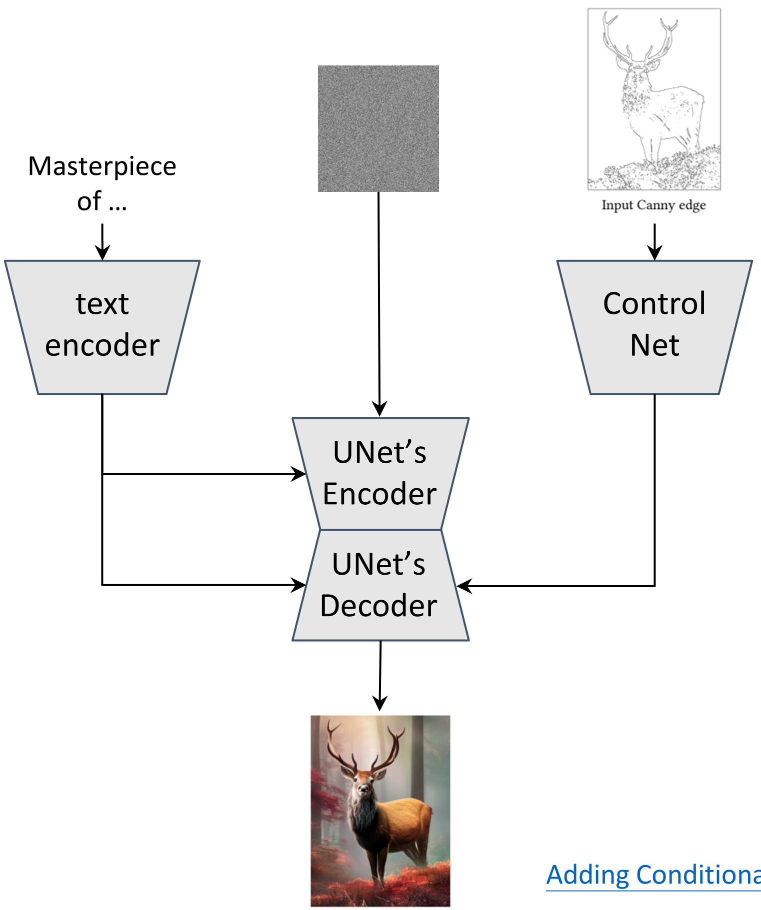


- Any concern?

DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation

Diffusion Model for Personalization (3): ControlNet

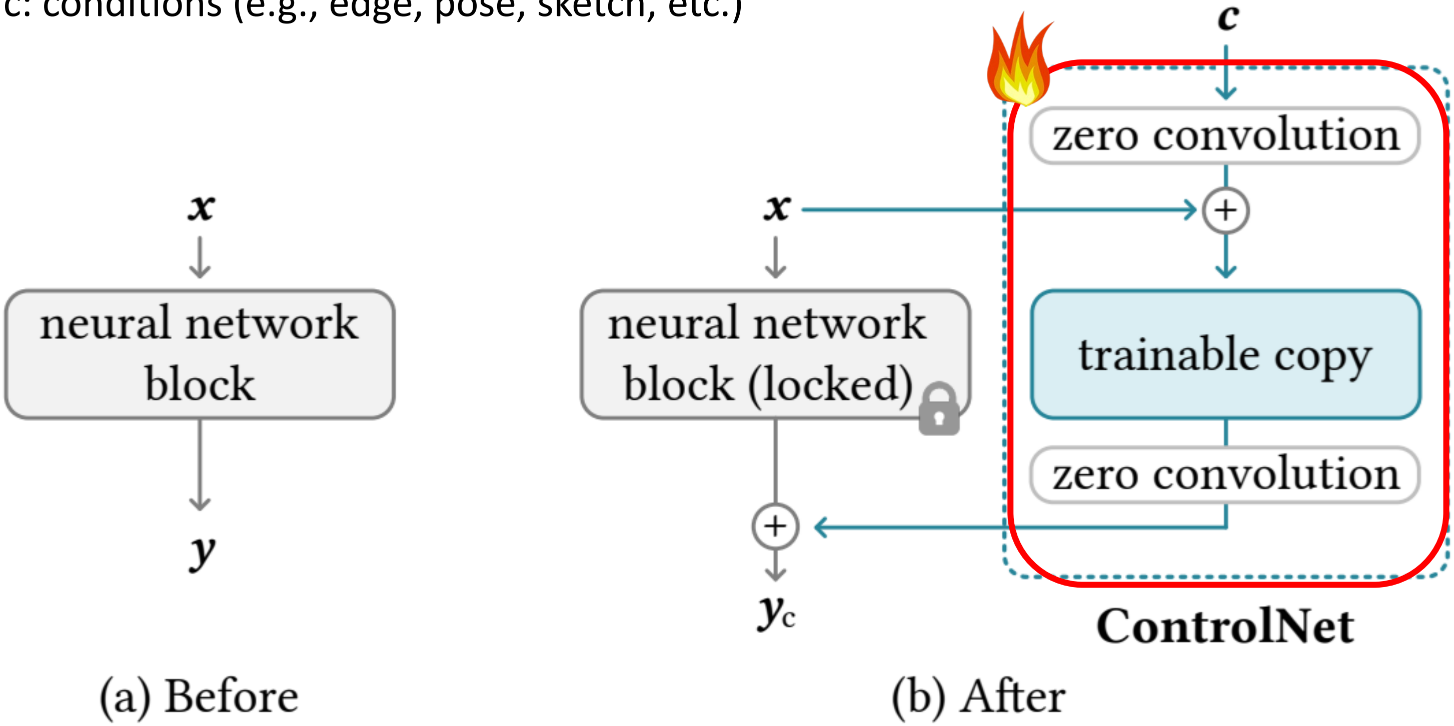
- Proposed by Stanford, ICCV 2023
- Goal: personalization via user-determined condition



[Adding Conditional Control to Text-to-Image Diffusion Models]

Diffusion Model for Personalization (3): ControlNet

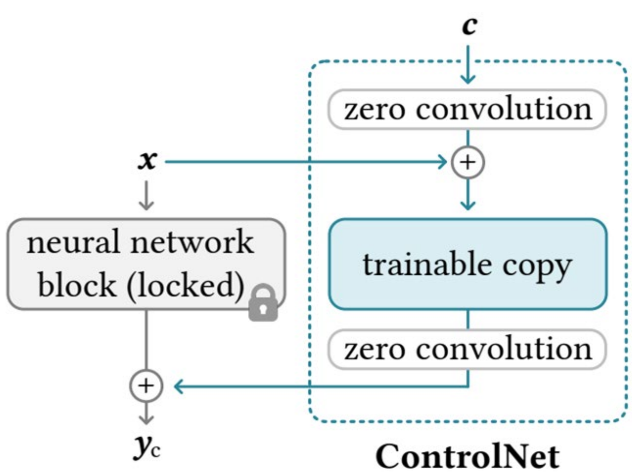
- Initialized from UNet's encoder
- Notations:
  - x: input noise of each layer
  - y: output noise of each layer
  - c: conditions (e.g., edge, pose, sketch, etc.)



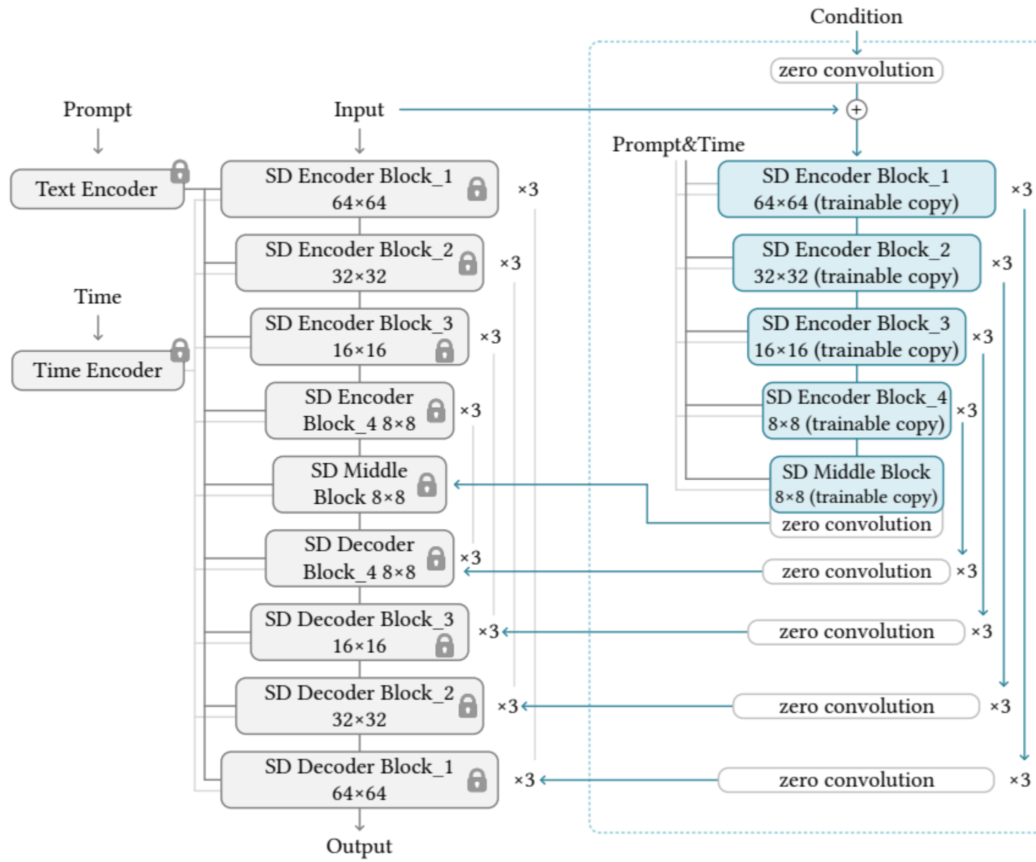
Adding Conditional Control to Text-to-Image Diffusion Models

Diffusion Model for Personalization (3): ControlNet

- Initialized from UNet's encoder
- Notations:
  - x: input noise of each layer
  - y: output noise of each layer
  - c: conditions (e.g., edge,, etc.)



(b) After

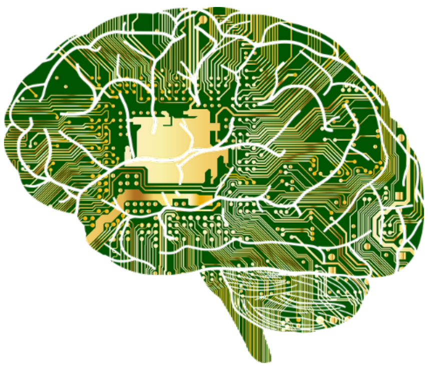


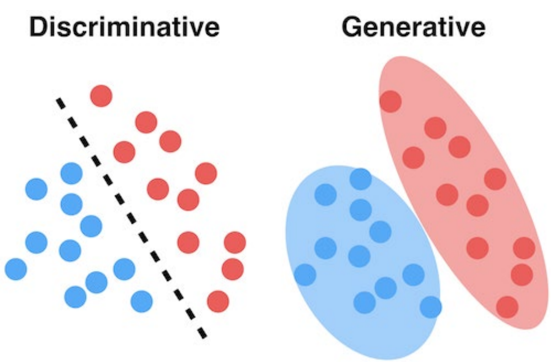
(a) Stable Diffusion (b) ControlNet

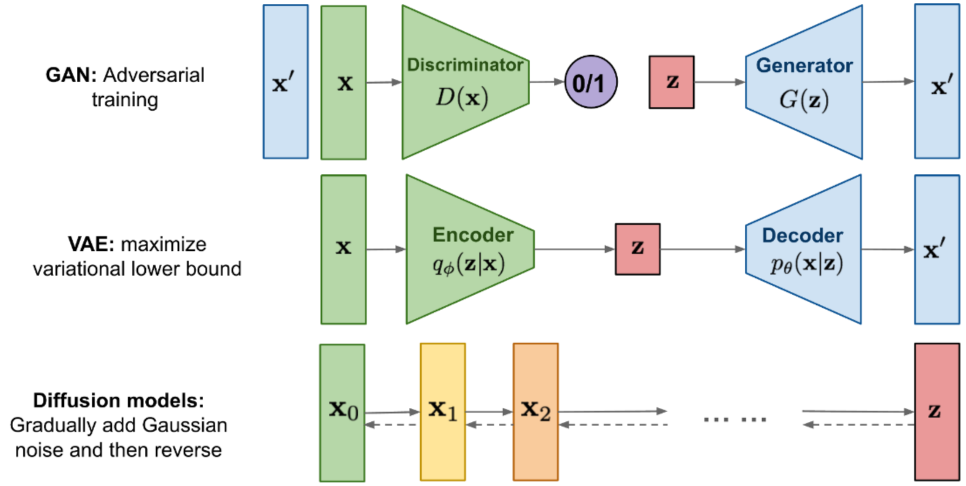
Adding Conditional Control to Text-to-Image Diffusion Models

What's to Be Covered Today…

- Generative Models
  - Diffusion Model
    - Conditional Diffusion Model
      - Classifier Guidance
      - Classifier-Free Guidance
      - Text/Image Guidance
    - Personalization via Diffusion Model
  - Generative Adversarial Network

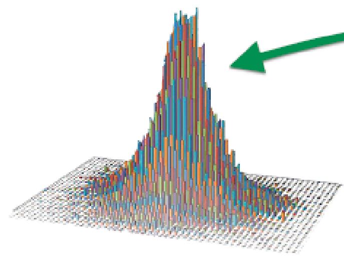


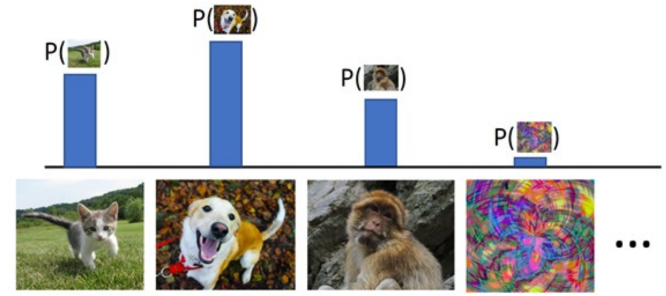


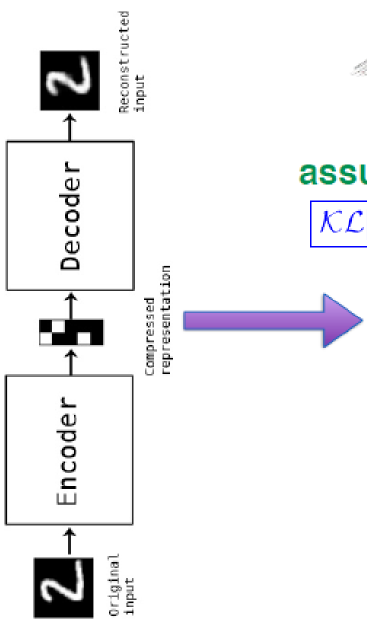


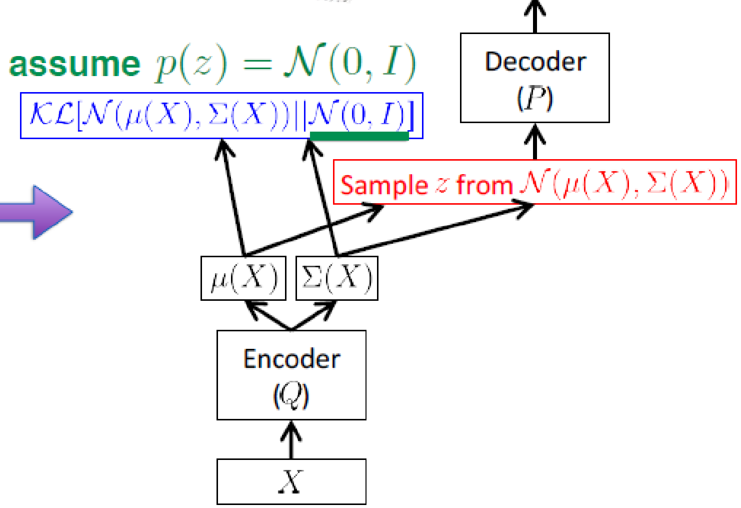
From VAE to *Generative Adversarial Networks (GAN)*

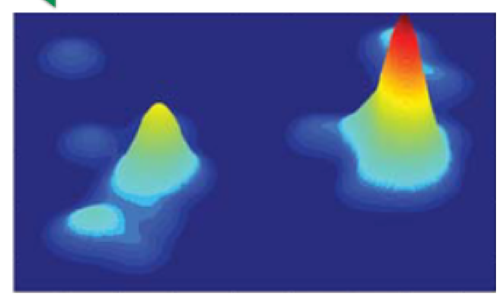
Now is a "*distribution*", we can assume it to be a distribution easy to sample from, e.g. Gaussian

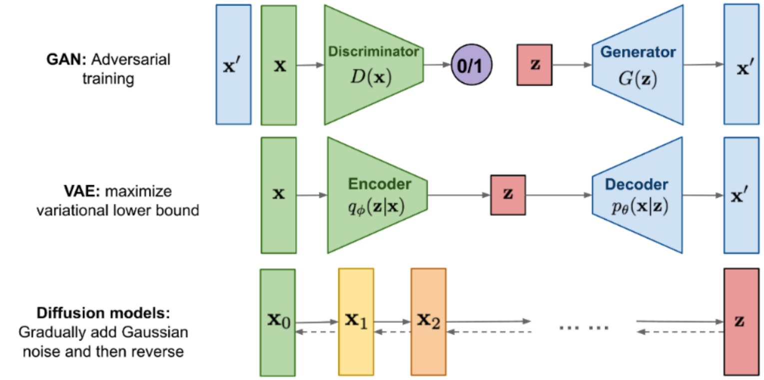






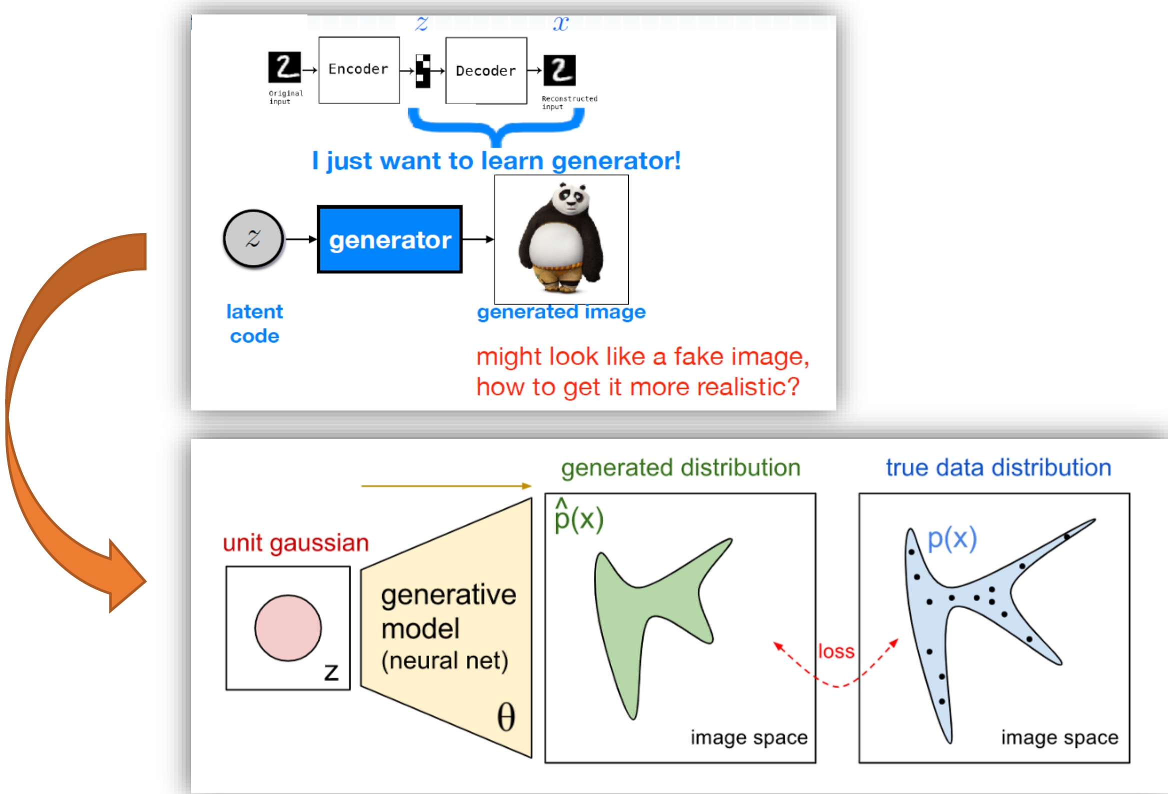






From VAE to GAN

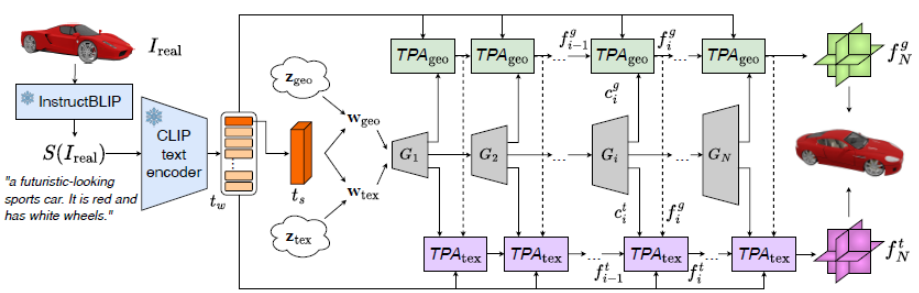
- Remarks
  - We only need the decoder/generator in practice.
  - We prefer fast generation.
  - How do we know if the output images are sufficiently good?

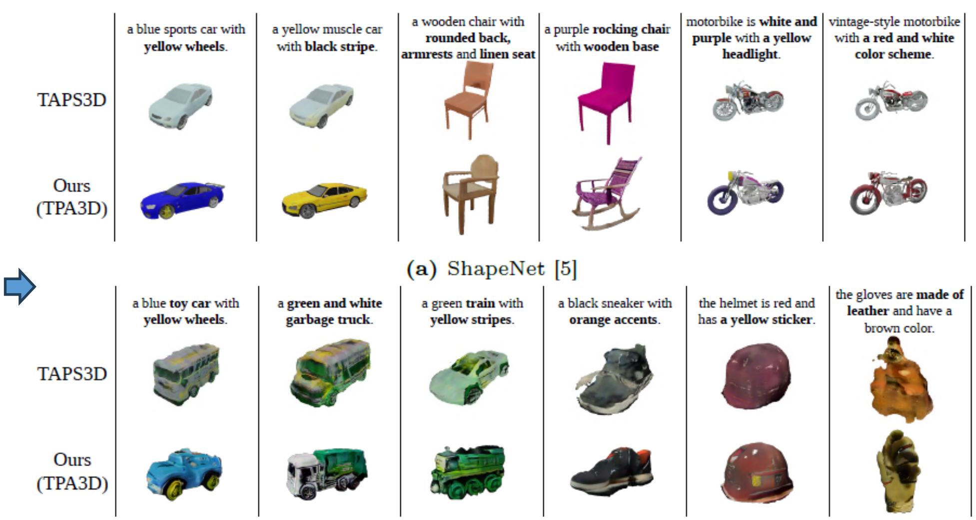


Slide credit: W. Chiu

GAN is NOT an outdated DL technology

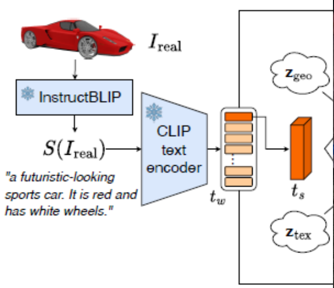
- Remarks
  - We only need the decoder/generator in practice.
  - We prefer fast generation.
  - How do we know if the output images are sufficiently good?
- Example
  - TPA3D: Triplane Attention for Fast Text-to-3D Generation, ECCV 2024
  - Bin-Shih Wu, Hong-En Chen, Shen-Yu Huang, and Y.-C. Frank Wang

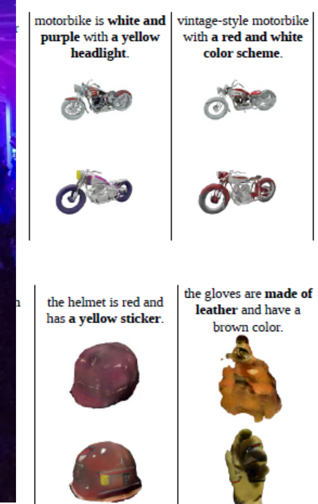




GAN is NOT an outdated DL technology

- Remark
  - We … r in pr
  - We
  - Ho … es are
- Exampl
  - TP/ … ext-to-3
  - Bir … u Huan





Generative Adversarial Network

- Idea
  - **Generator** to convert a vector z (sampled from P_z) into fake data x (from P_G), while we need $P_G = P_{data}$
  - **Discriminator** classifies data as real or fake (1/0)
  - How? Impose an **adversarial loss** on the observed data distribution!

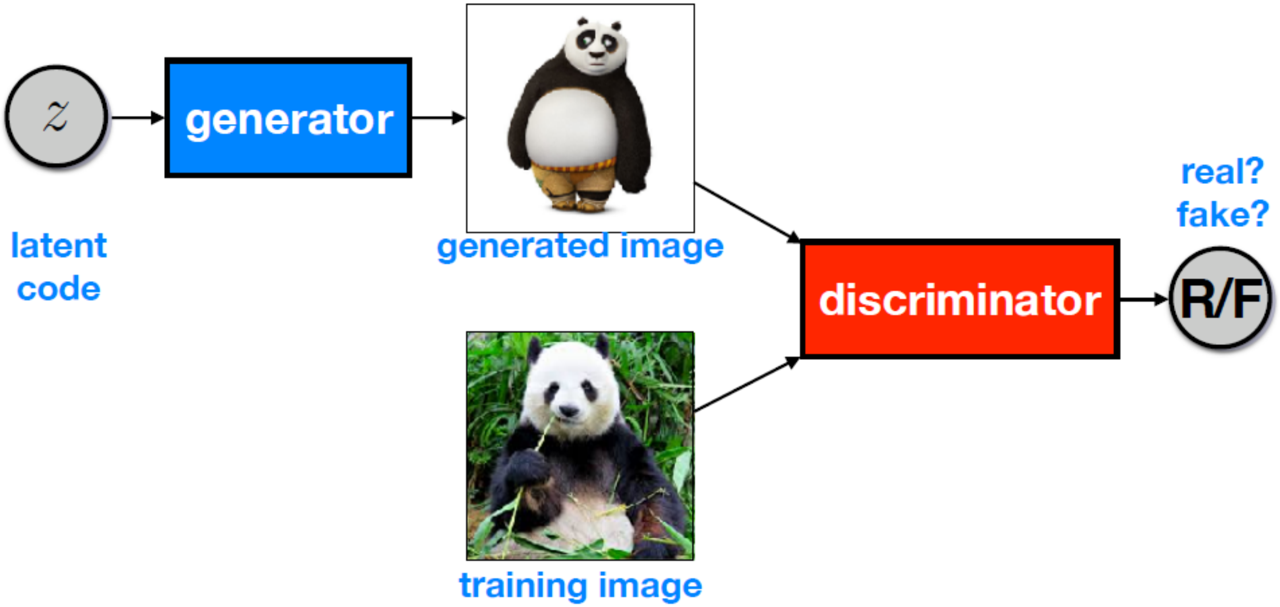


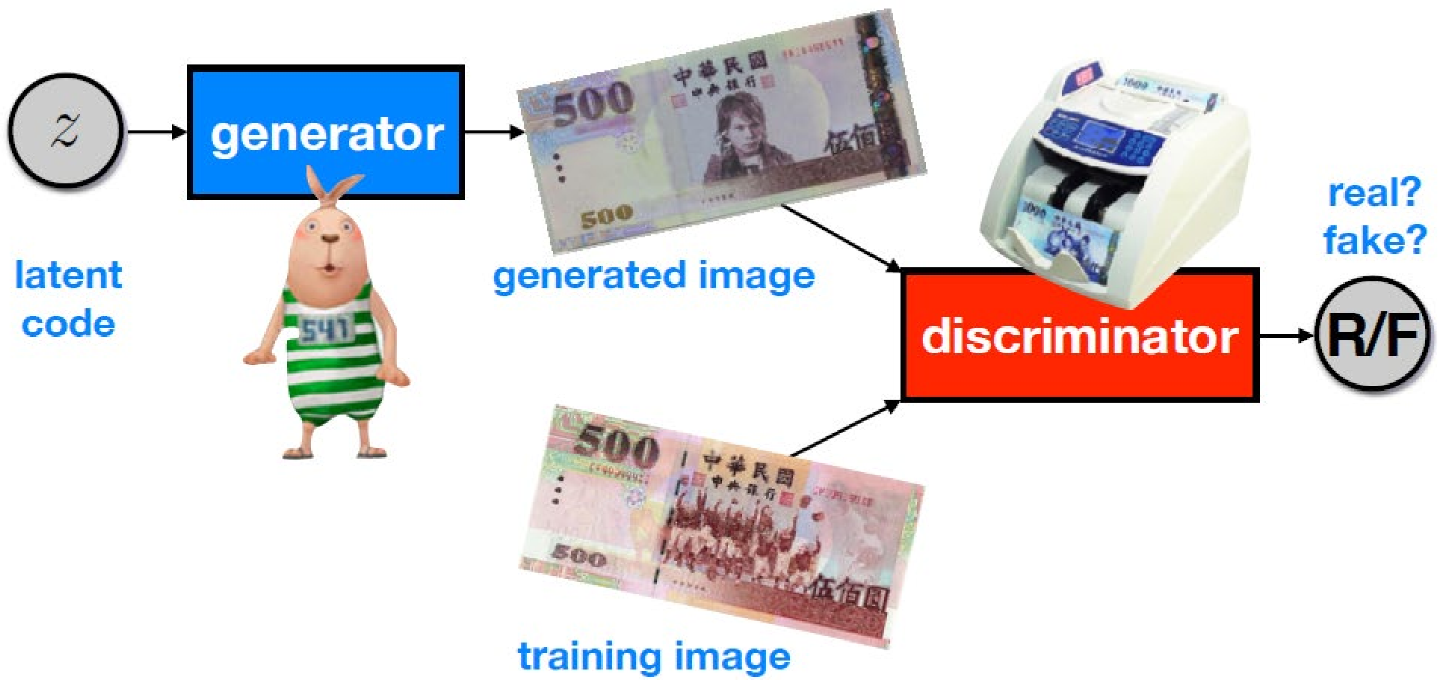
Image credit: W. Chiu

Generative Adversarial Network (cont'd)

- Key idea:
  - Impose *adversarial loss* on data distribution
  - Let's see a practical example…

generator: try to generate more realistic images to cheat discriminator

discriminator: try to distinguish whether the image is generated or real

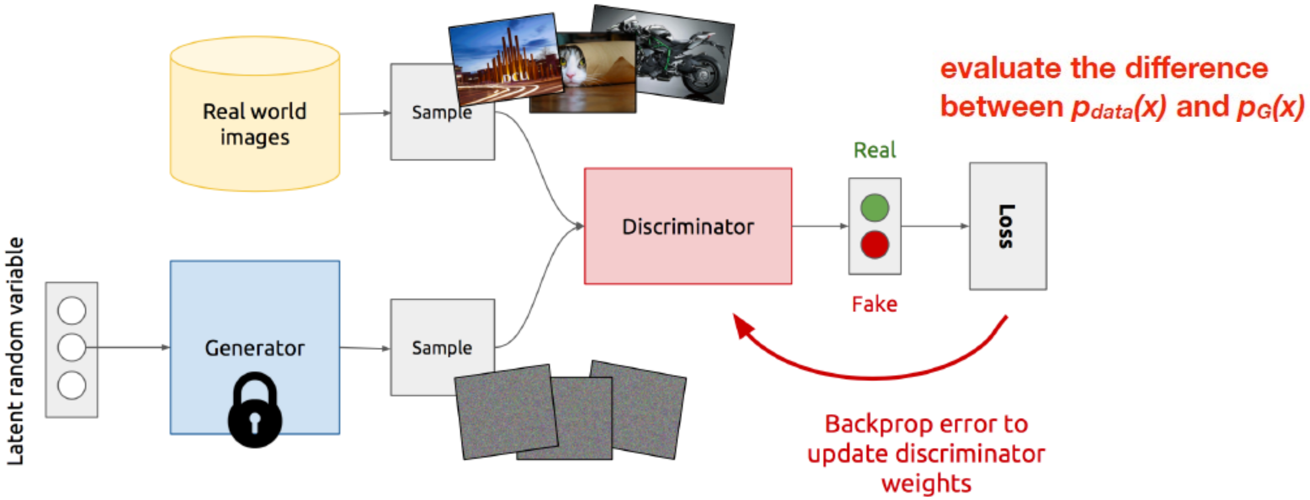


Slide credit: W. Chiu

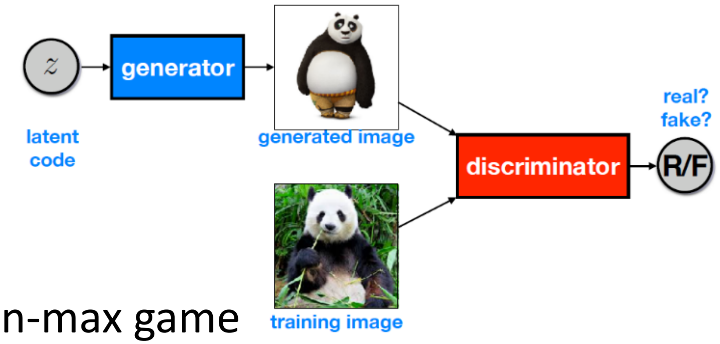
GAN (cont'd)

- Remarks
  - A function maps **normal distribution** $\boldsymbol{N(0, I)}$ to $\boldsymbol{P_{data}}$
  - How good we are in mapping P_g to P_{data}?
    - Train & ask the discriminator!
  - Conduct a two-player min-max game (see next slide for more details)

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

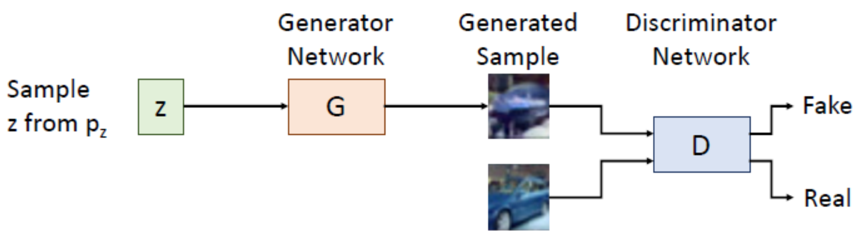


Training Objective of GAN



- Jointly train generator G and discriminator D with a min-max game

$$\min_{G}\max_{D}\left(\overbrace{E_{x\sim p_{data}}[\log D(x)]}^{\text{Discriminator wants } D(x) = 1 \text{ for real data}} + E_{z\sim p(z)}\left[\overbrace{\underbrace{\log\left(1 - D(G(z))\right)}_{\text{Generator wants } D(x) = 1 \text{ for fake data}}}^{\text{Discriminator wants } D(x) = 0 \text{ for fake data}}\right]\right)$$



- Train G & D with alternating gradient updates

$$\min_{G}\max_{D} V(G, D)$$

For t in 1, … T:

1. (Update D) $D = D + \alpha_D \frac{\partial V}{\partial D}$
2. (Update G) $G = G - \alpha_G \frac{\partial V}{\partial G}$

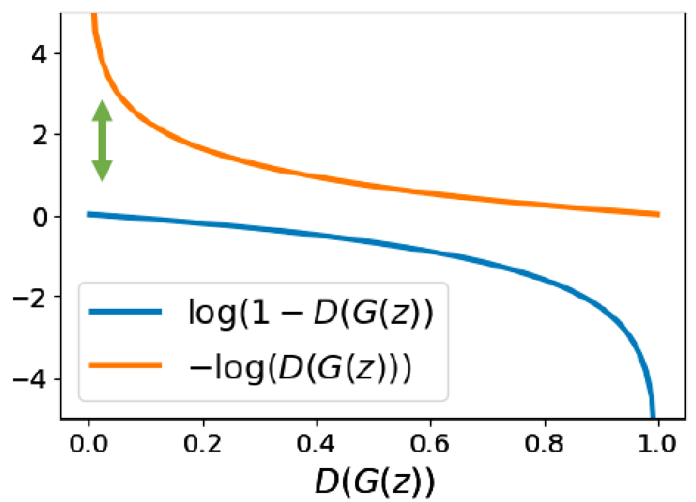
Slide credit: I. Goodfellow

Training Objective of GAN *(optional trick)*

- Potential Problem

$$\min_{G} \max_{D} \left(E_{x \sim p_{data}} [\log D(x)] + E_{z \sim p(z)} \left[\log \left(1 - D(G(z)) \right) \right] \right)$$

  - At start of training, G is not OK yet (obviously);
    D easily tells apart real/fake data (i.e., D(G(z)) close to 0).
  - Possible Solution:
    - Instead of training G to minimize log(1-D(G(z))) in the beginning,
      we train G to minimize -log(D(G(z)).
    - With strong gradients from G, we start the training of the above min-max game.



Slide credit: I. Goodfellow

Optimality of GAN

- Why the min-max game as objective a good idea?

$$\min_G \max_D \left(E_{x\sim p_{data}}[\log D(x)] + E_{z\sim p(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right]\right)$$

$$= \min_G \max_D \left(E_{x\sim p_{data}}[\log D(x)] + E_{x\sim p_G}[\log(1 - D(x))]\right)$$

$$= \min_G \int_X \max_D \left(p_{data}(x) \log D(x) + p_G(x) \log(1 - D(x))\right) dx$$

$$\left.\begin{aligned} f(y) &= a \log y + b \log(1-y) \\ f'(y) &= \frac{a}{y} - \frac{b}{1-y} \end{aligned}\right\} \quad f'(y) = 0 \Leftrightarrow y = \frac{a}{a+b} \text{ (local max)}$$

Optimal Discriminator: $D_G^*(x) = \dfrac{p_{data}(x)}{p_{data}(x) + p_G(x)}$

Slide credit: UMich EECS 498-007

Optimality of GAN

- Why the min-max game as objective a good idea? (cont'd)

$$\min_{G}\max_{D}\left(E_{x\sim p_{data}}[\log D(x)] + E_{z\sim p(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right]\right)$$

$$\Rightarrow \min_{G}\int_{X}\left(p_{data}(x)\log\frac{p_{data}(x)}{p_{data}(x)+p_{G}(x)} + p_{G}(x)\log\frac{p_{G}(x)}{p_{data}(x)+p_{G}(x)}\right)dx$$

$$= \min_{G}\left(E_{x\sim p_{data}}\left[\log\frac{2}{2}\frac{p_{data}(x)}{p_{data}(x)+p_{G}(x)}\right] + E_{x\sim p_{G}}\left[\log\frac{2}{2}\frac{p_{G}(x)}{p_{data}(x)+p_{G}(x)}\right]\right)$$

$$= \min_{G}\left(E_{x\sim p_{data}}\left[\log\frac{2 * p_{data}(x)}{p_{data}(x)+p_{G}(x)}\right] + E_{x\sim p_{G}}\left[\log\frac{2 * p_{G}(x)}{p_{data}(x)+p_{G}(x)}\right] - \log 4\right)$$

Slide credit: UMich EECS 498-007

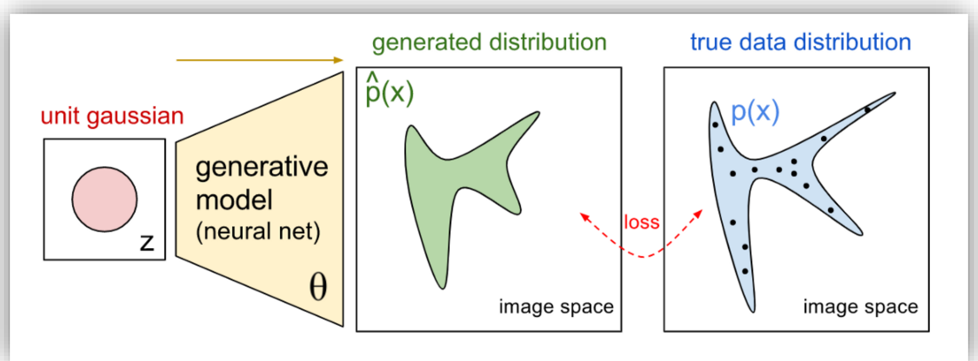
Optimality of GAN

- Why the min-max game as objective a good idea? (cont'd)

$$\min_{G} \max_{D} \left(E_{x\sim p_{data}}[\log D(x)] + E_{z\sim p(z)} \left[\log\left(1 - D\left(G(z)\right)\right)\right]\right)$$

$$= \min_{G} \left(E_{x\sim p_{data}} \left[\log \frac{2 * p_{data}(x)}{p_{data}(x) + p_G(x)}\right] + E_{x\sim p_G} \left[\log \frac{2 * p_G(x)}{p_{data}(x) + p_G(x)}\right] - \log 4\right)$$

$$= \min_{G} \left(KL\left(p_{data}, \frac{p_{data} + p_G}{2}\right) + KL\left(p_G, \frac{p_{data} + p_G}{2}\right) - \log 4\right)$$



Kullback-Leibler Divergence:

$$KL(p, q) = E_{x\sim p}\left[\log \frac{p(x)}{q(x)}\right]$$

Optimality of GAN

- Why the min-max game as objective a good idea? (cont'd)

$$\min_G \max_D \left(E_{x\sim p_{data}}[\log D(x)] + E_{z\sim p(z)}\left[\log\left(1 - D(G(z))\right)\right]\right)$$

$$= \min_G \left(E_{x\sim p_{data}}\left[\log \frac{2 * p_{data}(x)}{p_{data}(x) + p_G(x)}\right] + E_{x\sim p_G}\left[\log \frac{2 * p_G(x)}{p_{data}(x) + p_G(x)}\right] - \log 4\right)$$

$$= \min_G \left(KL\left(p_{data}, \frac{p_{data} + p_G}{2}\right) + KL\left(p_G, \frac{p_{data} + p_G}{2}\right) - \log 4\right)$$

$$= \min_G (2 * JSD(p_{data}, p_G) - \log 4)$$

JSD is always nonnegative, and zero only when the two distributions are equal! Thus $p_{data} = p_G$ is the global min, QED

Jensen-Shannon Divergence:

$$JSD(p, q) = \frac{1}{2} KL\left(p, \frac{p+q}{2}\right) + \frac{1}{2} KL\left(q, \frac{p+q}{2}\right)$$

Slide credit: UMich EECS 498-007

Remarks on Optimality of GAN

$$\min_G \max_D \left(E_{x\sim p_{data}}[\log D(x)] + E_{z\sim p(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right]\right)$$

$$= \min_G (2 * JSD(p_{data}, p_G) - \log 4)$$

- Summary
  - The global min of the minmax game happens when

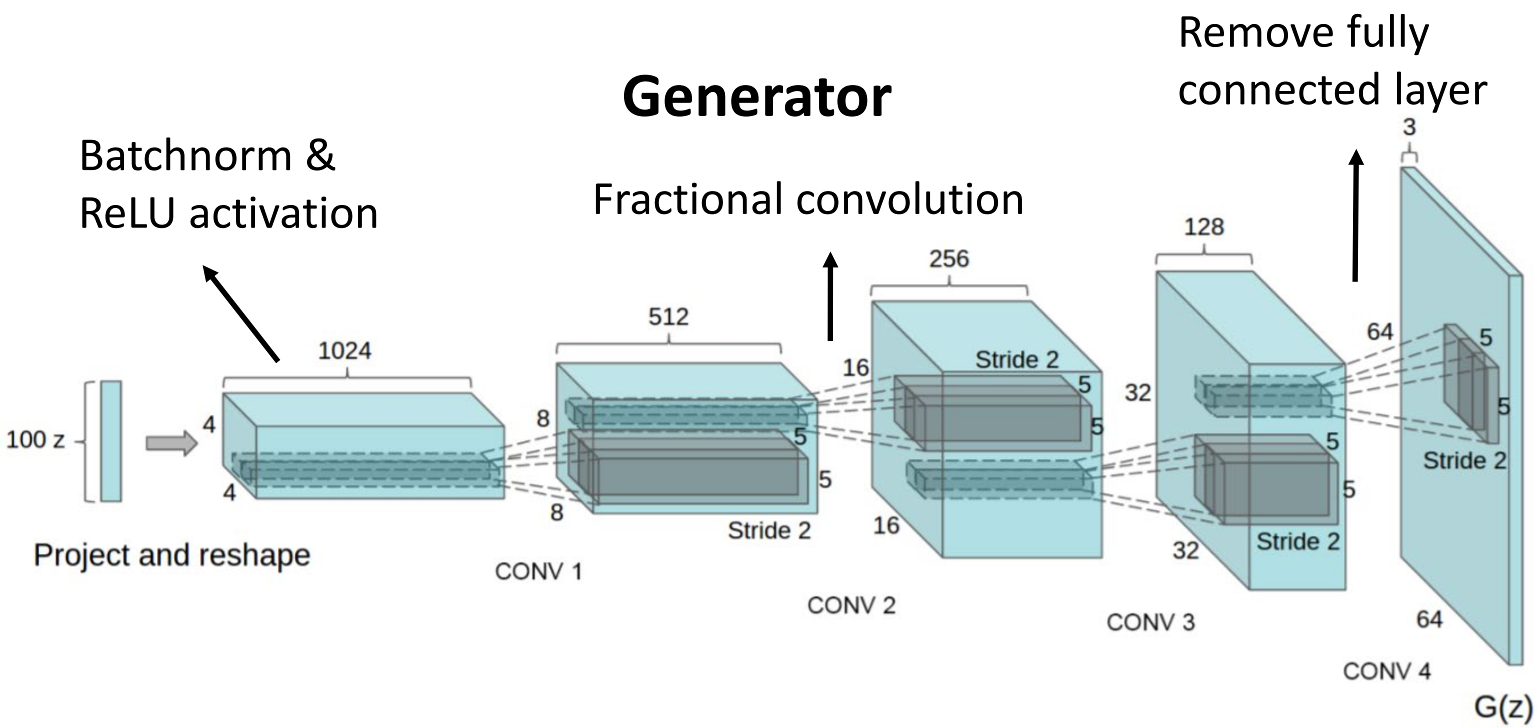
1. $D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x)+p_G(x)}$ (Optimal discriminator for any G)
2. $p_G(x) = p_{data}(x)$ (Optimal generator for optimal D)

- **Caution!**
  - G and D are learned models (i.e., DNNs) with fixed architectures.
    We don't know whether we can actually represent the optimal D & G.
  - Optimality of GAN does not tell anything about convergence to the optimal D/G.

Slide credit: UMich EECS 498-007

Deep Convolutional GAN (DC-GAN)

- Remarks
  - ICLR 2016
  - A CNN+GAN architecture
  - Empirically make training of GAN more stable



Deep Convolutional GAN (DC-GAN)

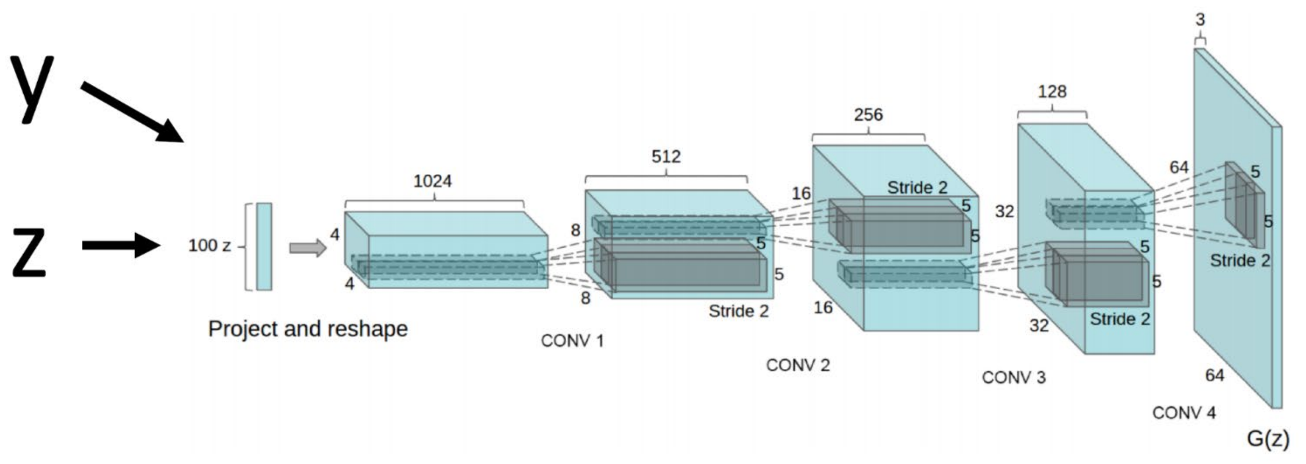
- Example Results

Collected face dataset

LSUN dataset

Conditional GANs

- Remarks
  - ICLR 2016
  - Conditional generative model p(x|y) instead of p(x)
  - Both G and D take the label y as an additional input… i.e., a conditional discriminator is deployed….Why?
  - Why not just use D as designed in the standard GAN?



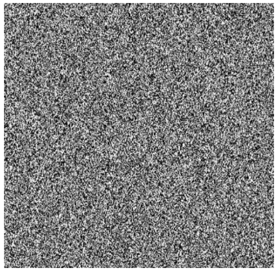
Conditional GANs

- Example Results

Miyato et al, "Spectral Normalization for Generative Adversarial Networks", ICLR 2018

Problems in Training GANs: Vanishing Gradients

- What Might Go Wrong?
  - GAN training is often unstable.
  - In other words, training might not converge properly.
  - The discriminator which we prefer is…



0

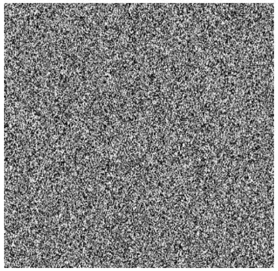
0.2

0.6

1

Problems in Training GANs: Vanishing Gradients (cont'd)

- What Might Go Wrong?
  - GAN training is often unstable.
  - In other words, training might not converge properly.
  - The discriminator we trained might be as follows.
    In other words, no gradient to guide the generator to output proper images.



0

0

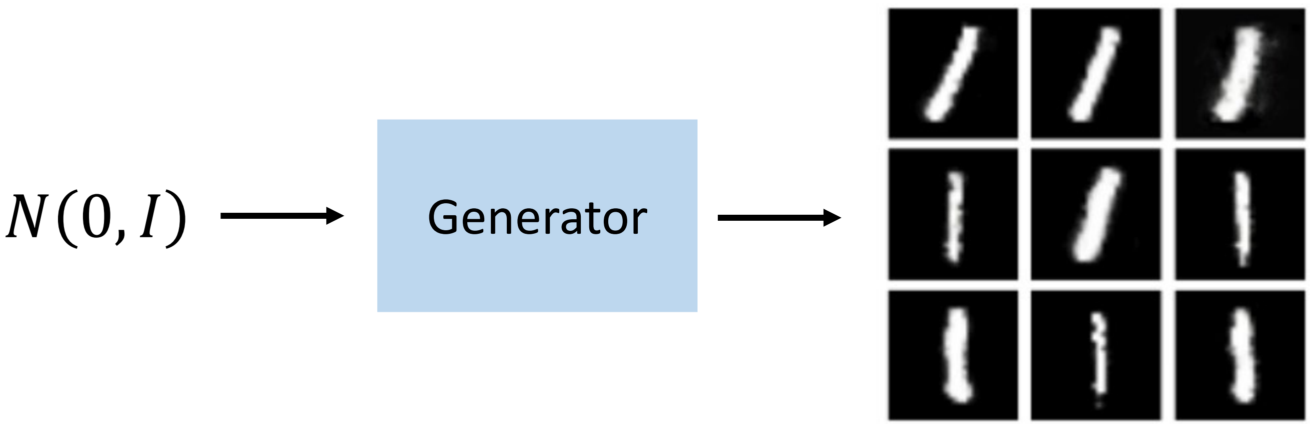
0

1

- This is known as the problem of *vanishing gradients*.

Problems in Training GANs: Mode Collapse

- Remarks
  - The generator only outputs a limited number of image variants regardless of the inputs.



Mode collapse!

Problems in Training GANs: Mode Collapse (cont'd)

- Remarks
  - The generator only outputs a limited number of image variants regardless of the inputs.

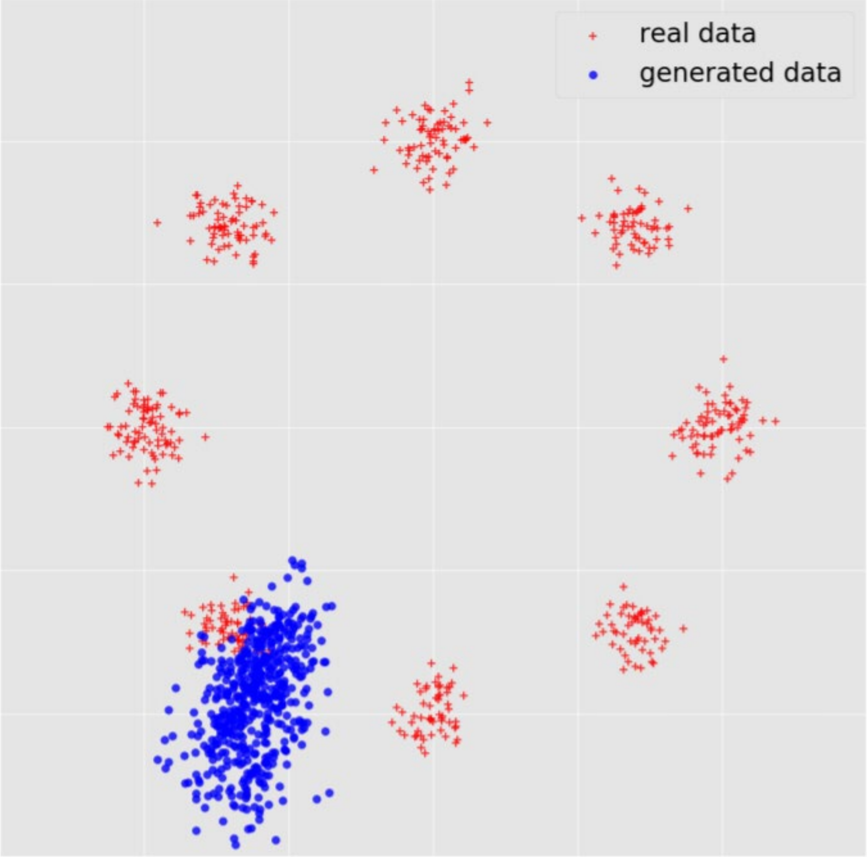
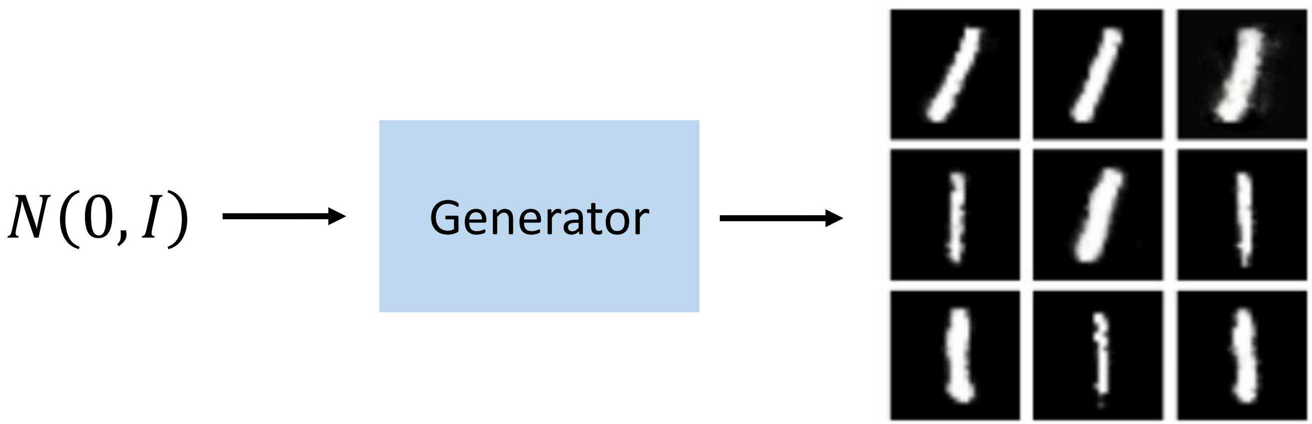


Photo credit: https://openreview.net/pdf?id=rkmu5b0a-

Problems in Training GANs: Mode Collapse (cont'd)

- Why Mode Collapse Happens?
  - The objective of GANs assesses the image authenticity, not diversity.
  - Imbalance training between generator/discriminator (exploding/vanishing gradients)



Energy-Based GAN

- Energy Function
  - Converting input data into scalar outputs, viewed as energy values
  - Desired configuration is expected to output low energy values & vice versa.
- Energy Function as Discriminator
  - Use of autoencoder; can be pre-trained!
  - Reconstruction loss outputs a range of values instead of binary logistic loss.
  - Empirically better convergence

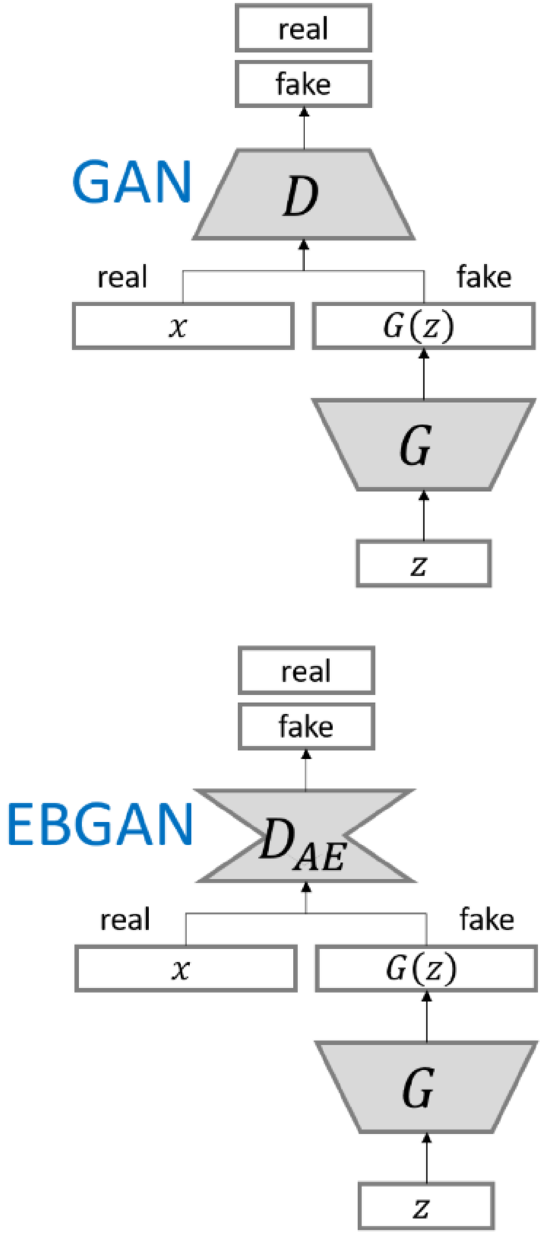
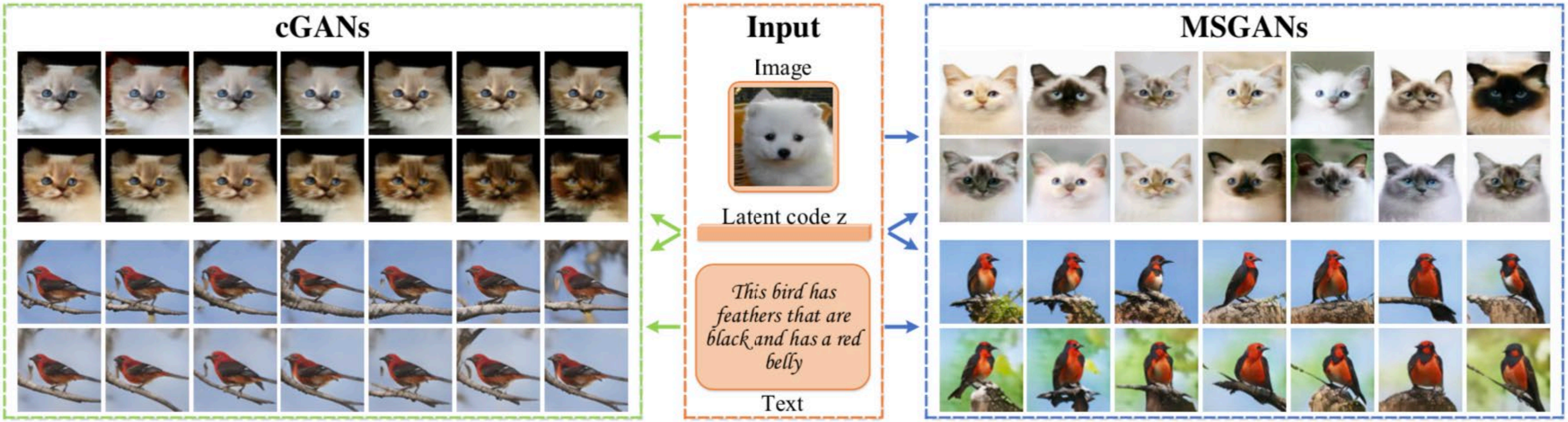


Photo credit: https://github.com/znxlwm/pytorch-generative-model-collections

MSGAN

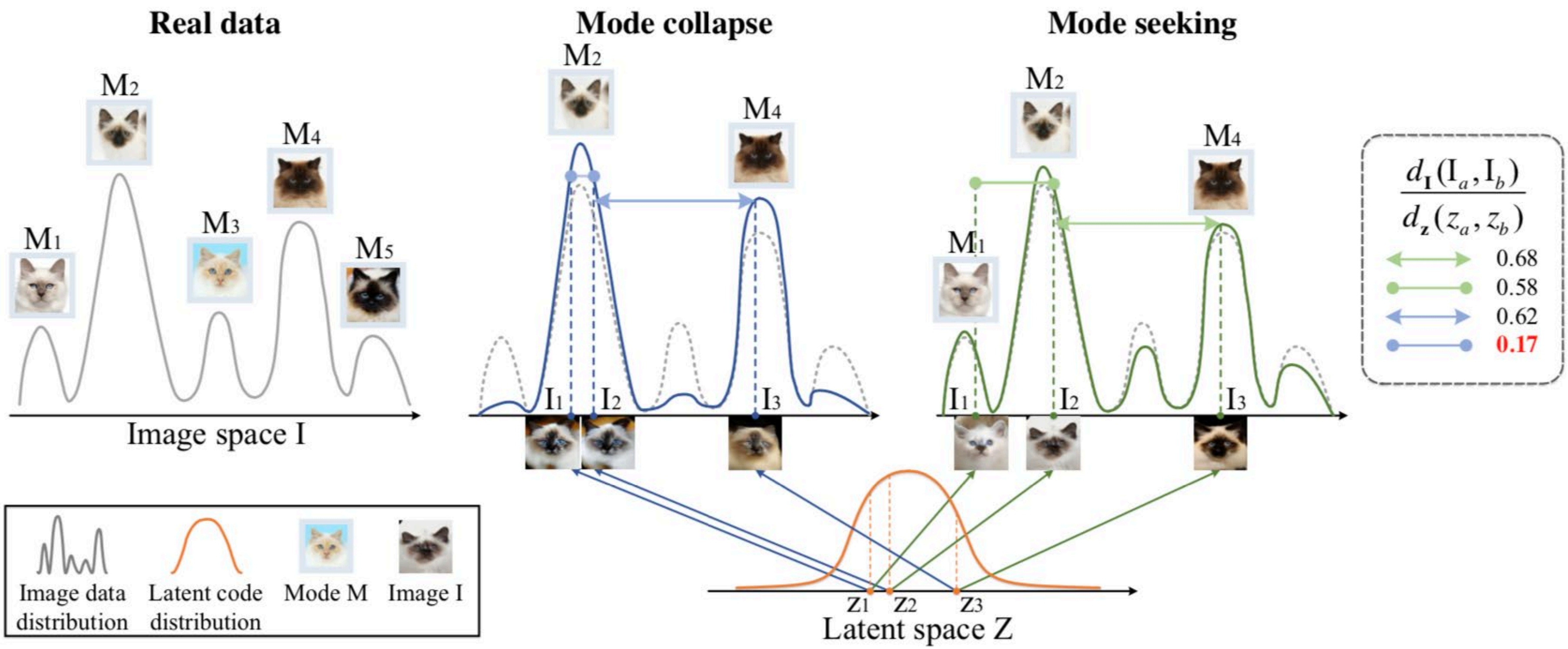
- To address the **mode collapse** issue by conditional GANs
- Mode Seeking Generative Adversarial Networks
  for Diverse Image Synthesis
- With the goal of producing **diverse** image outputs.



Mao *et al.* " Mode Seeking Generative Adversarial Networks for Diverse Image Synthesis. " *CVPR* 2019

MSGAN (cont'd)

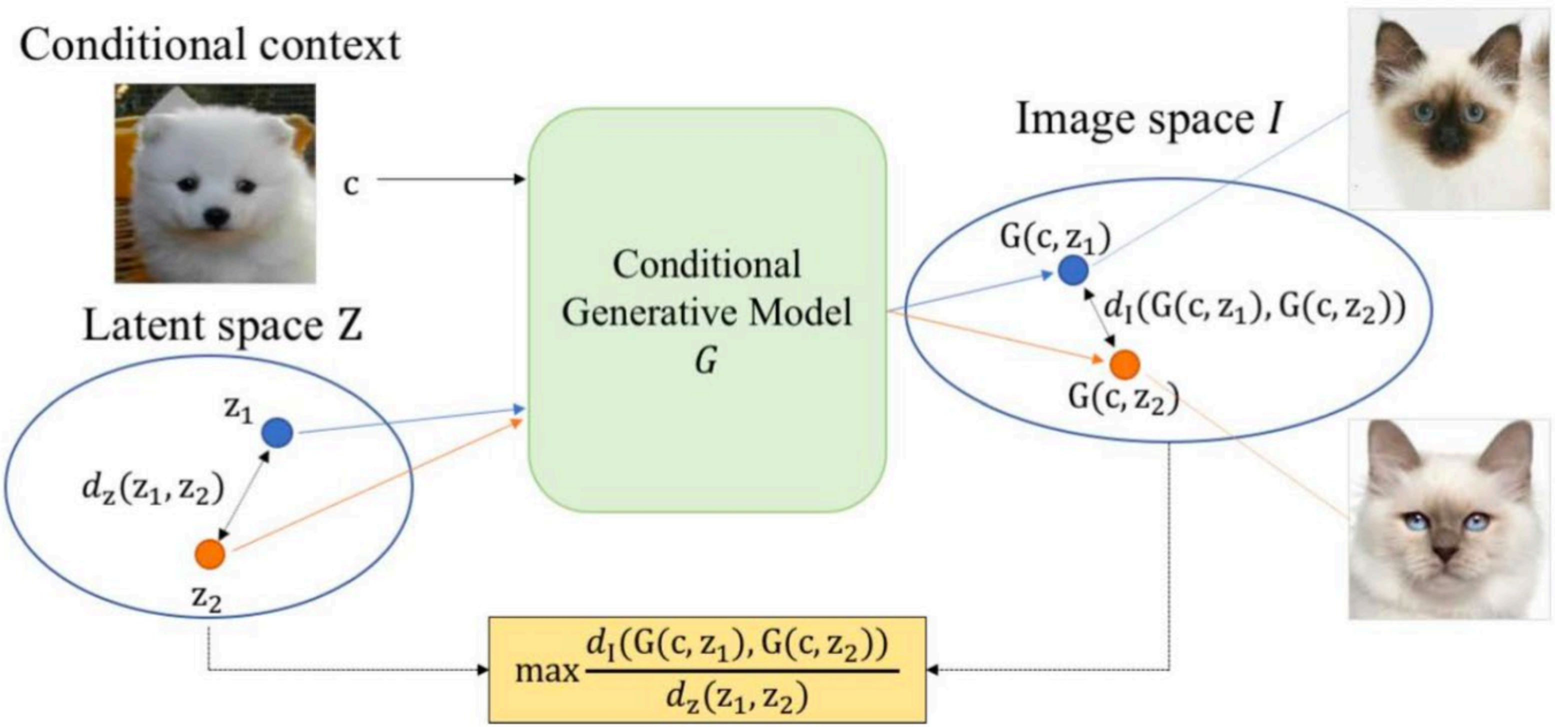
- Motivation (for unconditional GAN)



Mao *et al.* " Mode Seeking Generative Adversarial Networks for Diverse Image Synthesis. " *CVPR* 2019

MSGAN (cont'd)

- Proposed Regularization (for conditional GAN)



Mao *et al.* " Mode Seeking Generative Adversarial Networks for Diverse Image Synthesis. " *CVPR* 2019

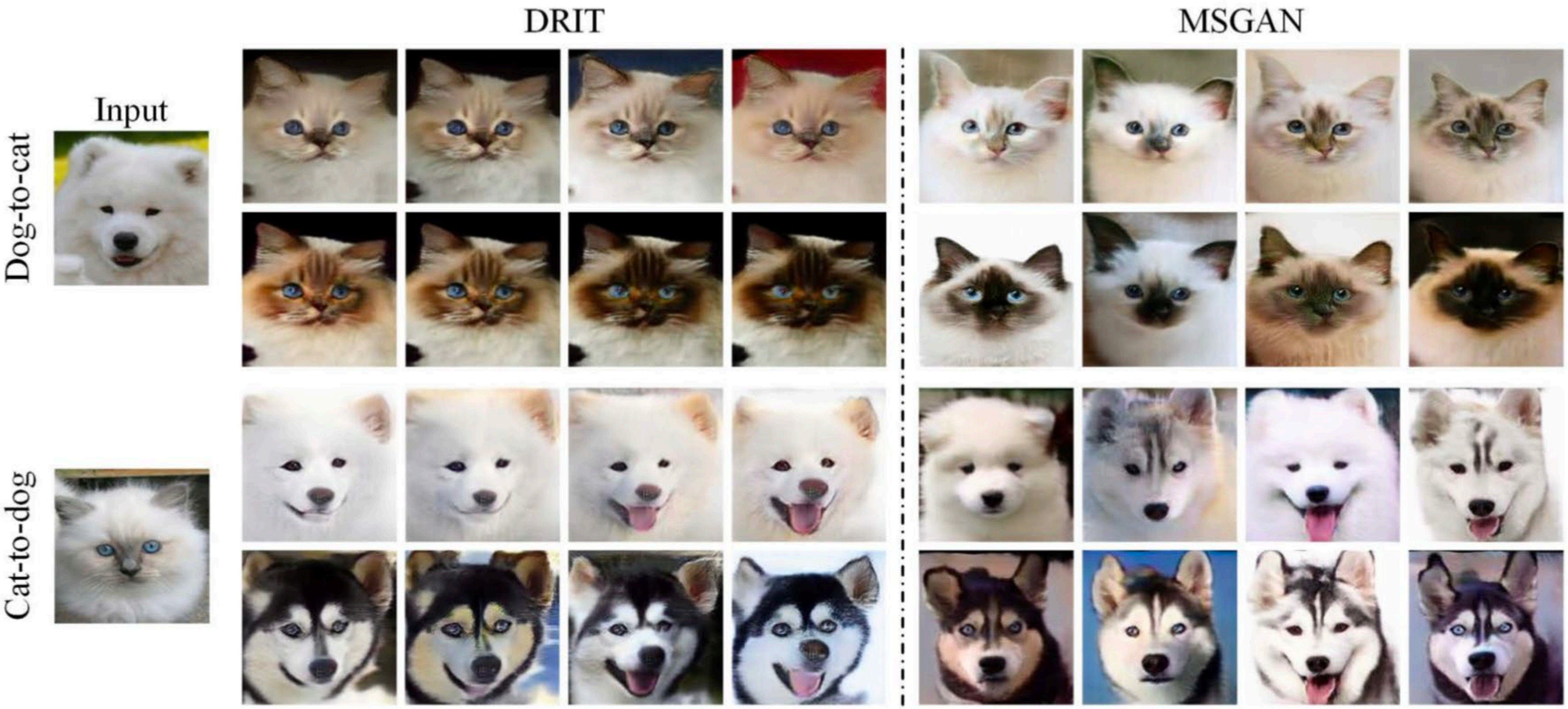
MSGAN (cont'd)

- Qualitative results
  - Conditioned on paired images

Mao *et al.* " Mode Seeking Generative Adversarial Networks for Diverse Image Synthesis. " *CVPR* 2019

MSGAN (cont’d)

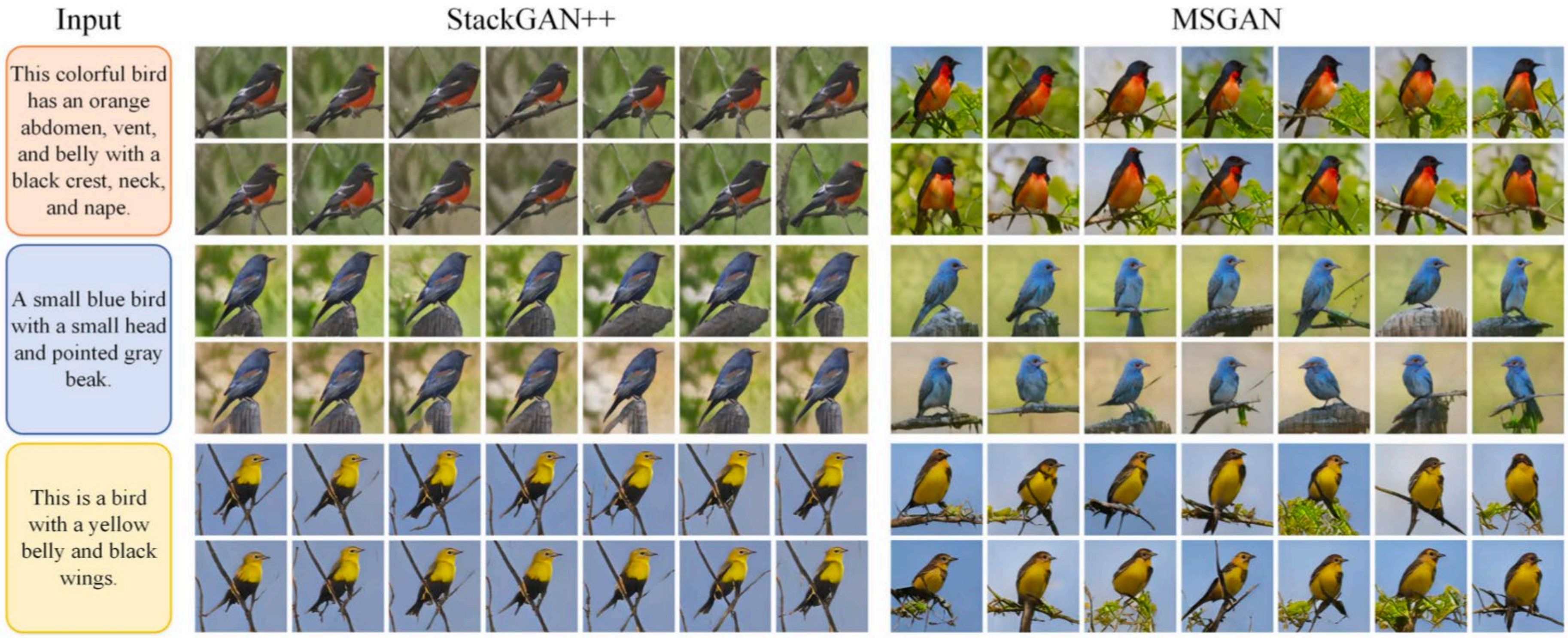
- Qualitative results
  - Conditioned on unpaired images



Mao *et al.* ” Mode Seeking Generative Adversarial Networks for Diverse Image Synthesis. ” *CVPR* 2019

MSGAN (cont'd)

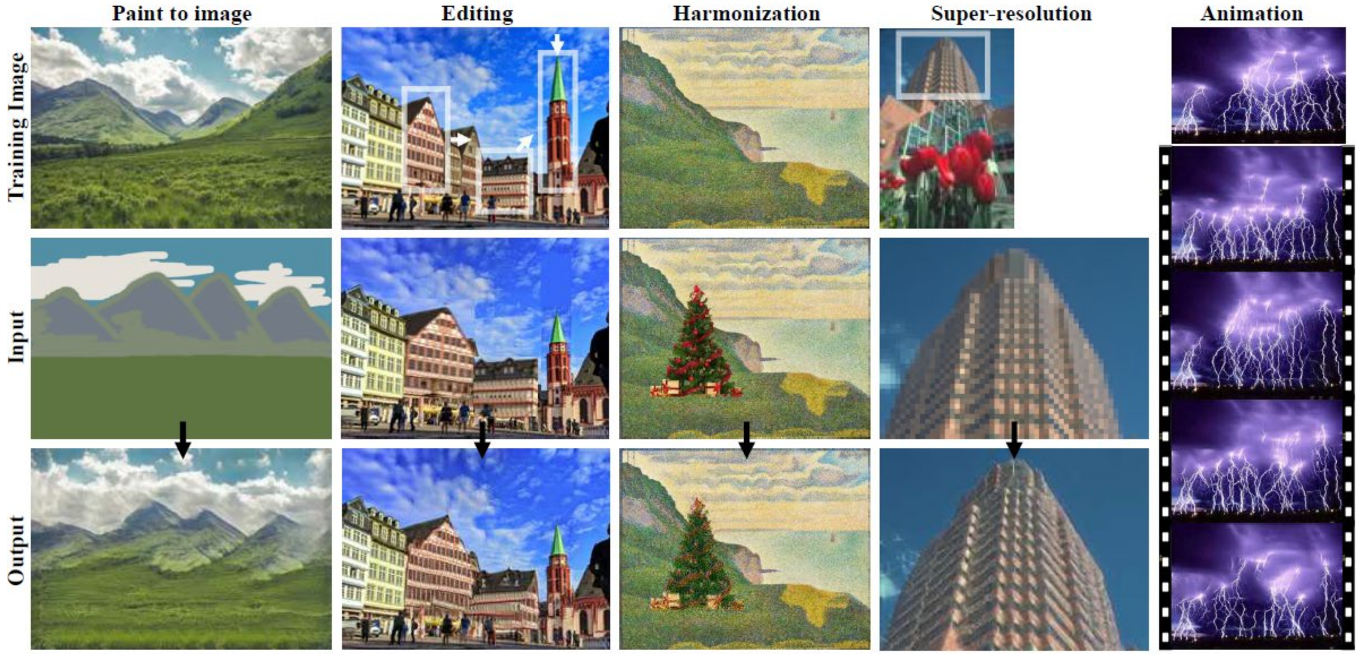
- Qualitative results
  - Conditioned on text (will talk about Vision & Language later this semester)



Mao *et al.* " Mode Seeking Generative Adversarial Networks for Diverse Image Synthesis. " *CVPR* 2019

SinGAN (if time permits)**:**
Learning a Generative Model from a Single Natural Image

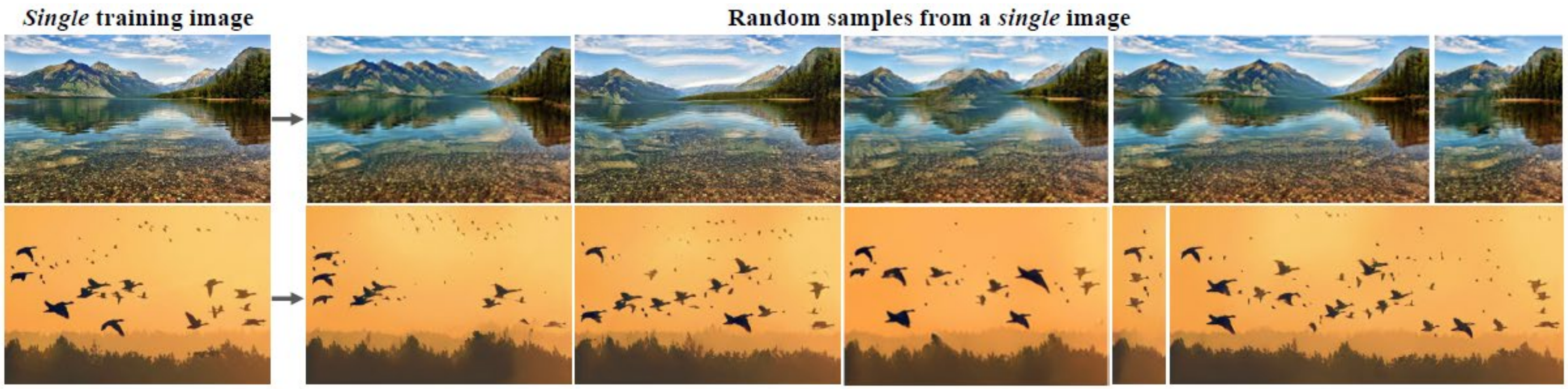
- ICCV 2019 Best Paper Award
- Remarks:
  - Learning from a **single image**
  - Handle **multiple image manipulation tasks**
    - Super-resolution, style conversion, harmonization, image editing, et.



Shaham et al., SinGAN: Learning a Generative Model from a Single Natural Image, ICCV 2019

SinGAN:
Learning a Generative Model from a Single Natural Image

- Goal
  - Output images with arbitrary sizes and aspect ratios (via fully conv models) by changing dimensions of noise and the input size



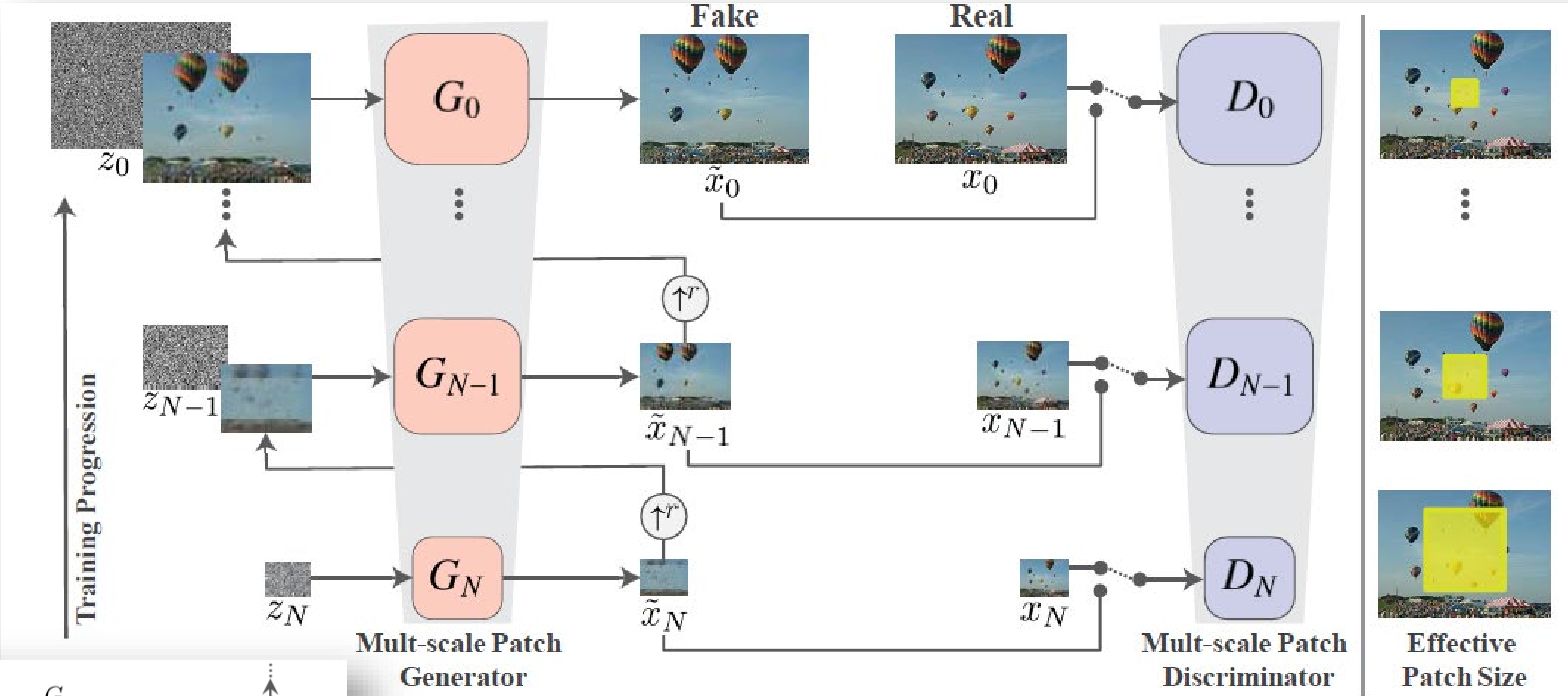
Shaham et al., SinGAN: Learning a Generative Model from a Single Natural Image, ICCV 2019

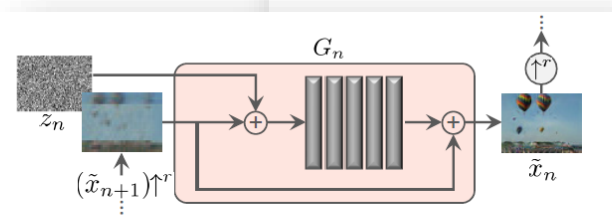
SinGAN:
Learning a Generative Model from a Single Natural Image

- Framework

$$\min_{G_n} \max_{D_n} \mathcal{L}_{\text{adv}}(G_n, D_n) + \alpha \mathcal{L}_{\text{rec}}(G_n)$$

fix kernel (receptive field) size at each scale:
capture structures of decreasing size as we go up

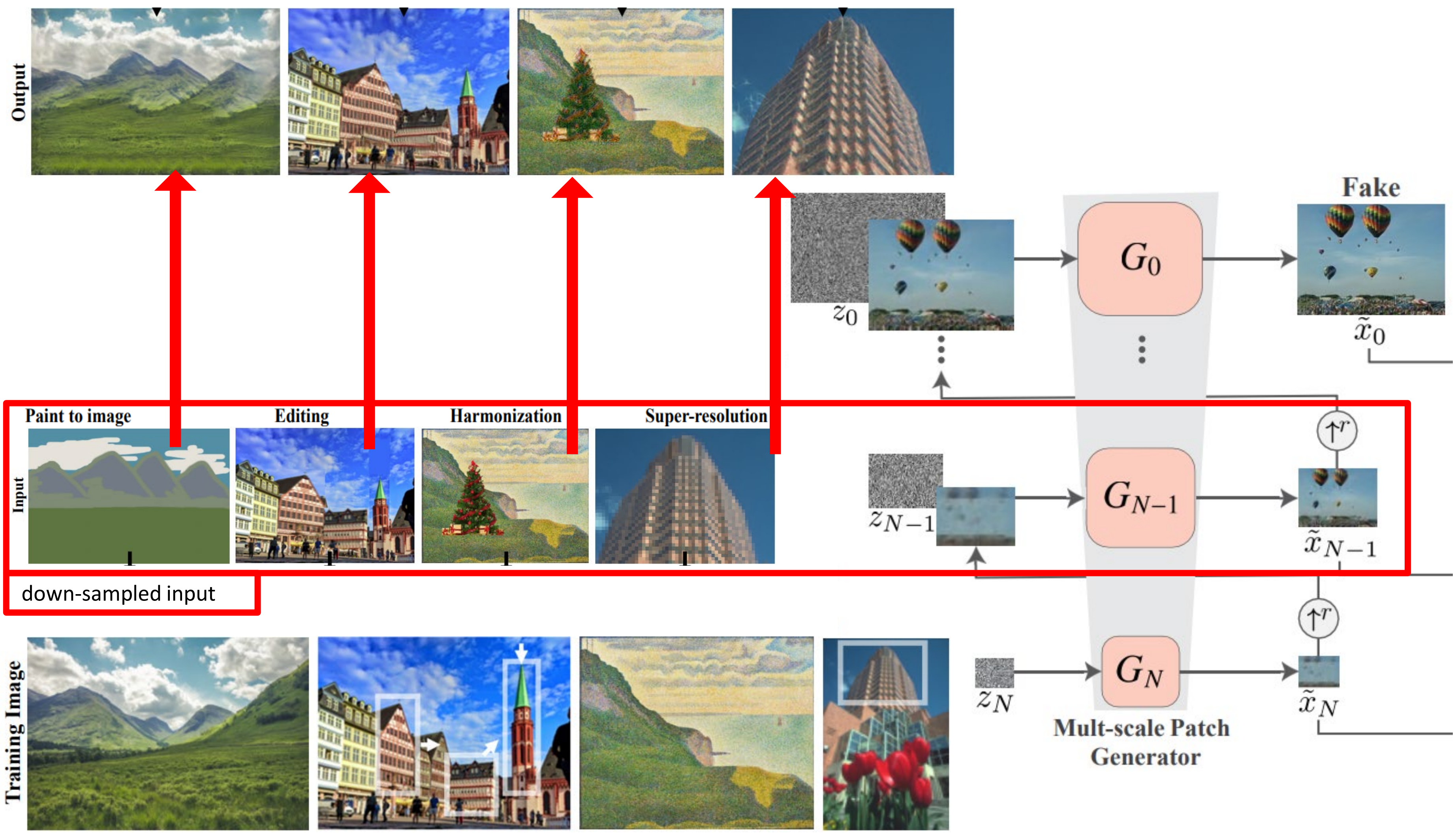




add **noise** before **Conv**:
ensure that GAN does not
disregard the noise

Shaham et al., SinGAN: Learning a Generative Model from a Single Natural Image, ICCV 2019

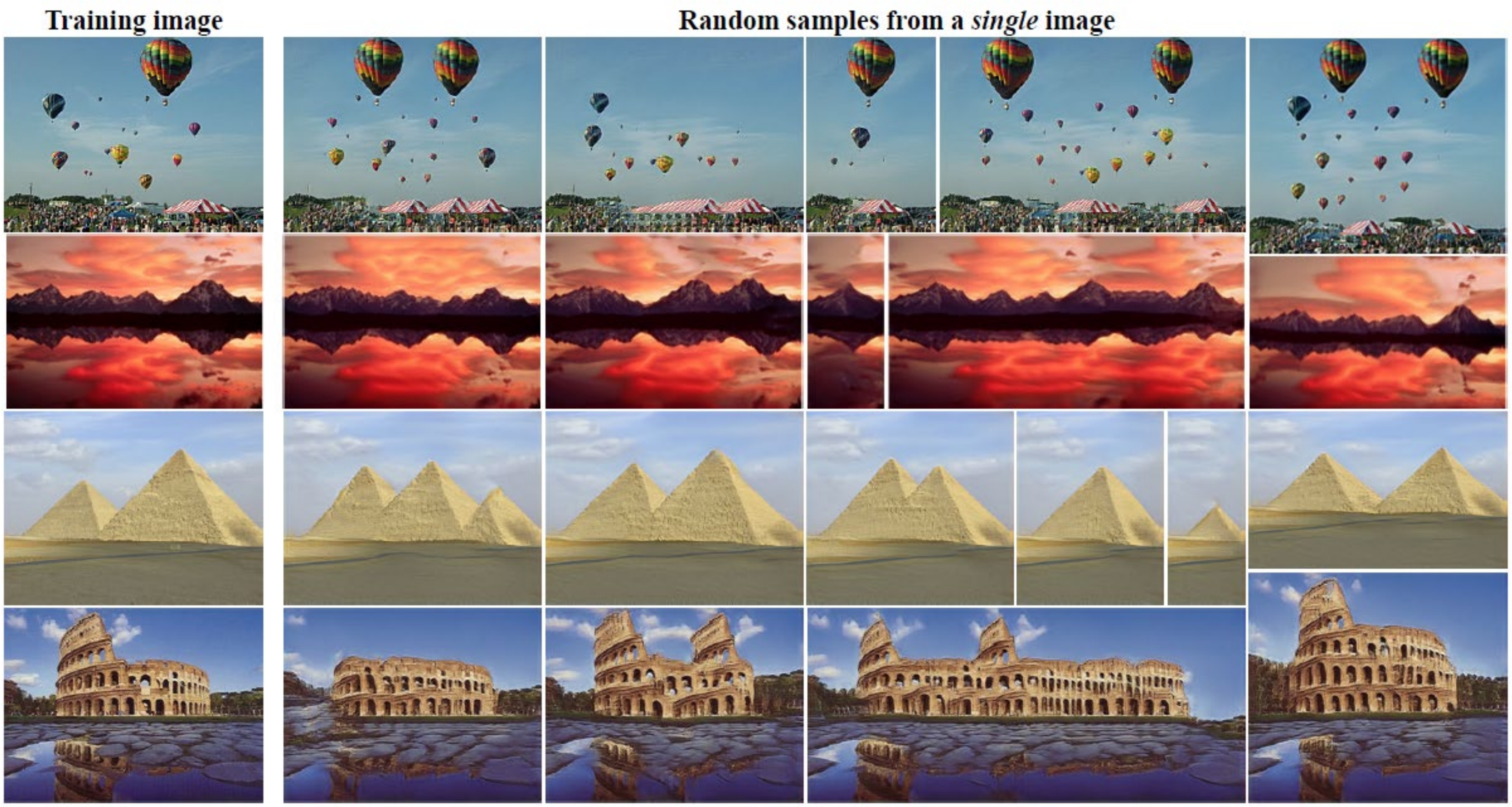
Inference Stage for SinGAN



Shaham et al., SinGAN: Learning a Generative Model from a Single Natural Image, ICCV 2019

SinGAN:
Learning a Generative Model from a Single Natural Image

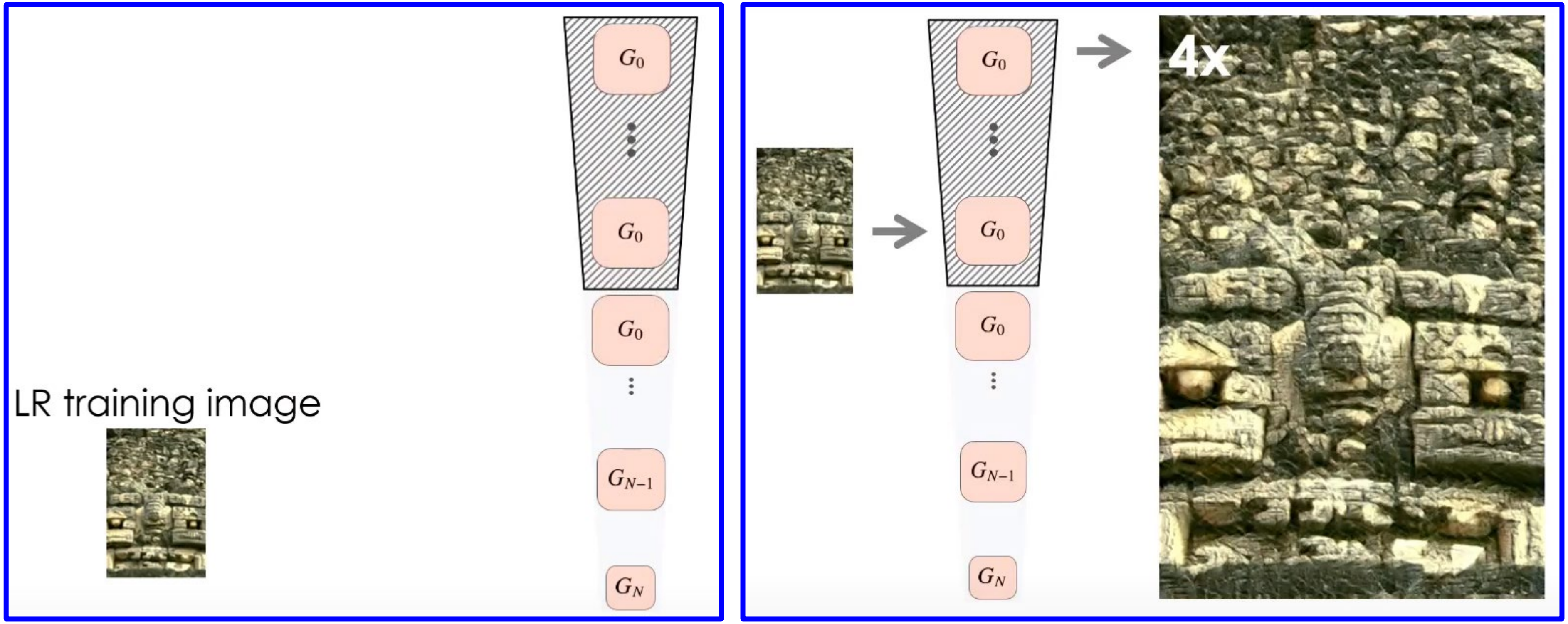
- Random image generation



Shaham et al., SinGAN: Learning a Generative Model from a Single Natural Image, ICCV 2019

SinGAN:
Learning a Generative Model from a Single Natural Image

- Super-Resolution



Shaham et al., SinGAN: Learning a Generative Model from a Single Natural Image, ICCV 2019

SinGAN:
Learning a Generative Model from a Single Natural Image

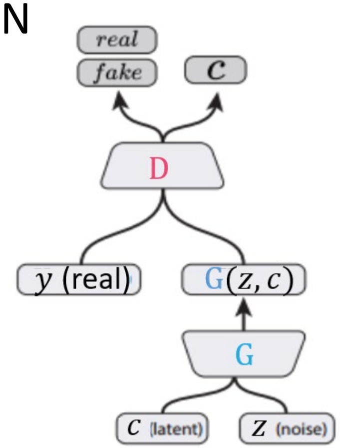
- Editing

Image	Injection scale	Total number of scales
Rock1	$n = 5$	$N = 7$
Rock2	$n = 5$	$N = 7$
Rock3 (also Fig. 12, main text)	$n = 5$	$N = 7$
Tree	$n = 7$	$N = 9$
Mountain	$n = 4$	$N = 8$
Red cliff	$n = 5$	$N = 9$
Hay	$n = 6$	$N = 9$

Shaham et al., SinGAN: Learning a Generative Model from a Single Natural Image, ICCV 2019

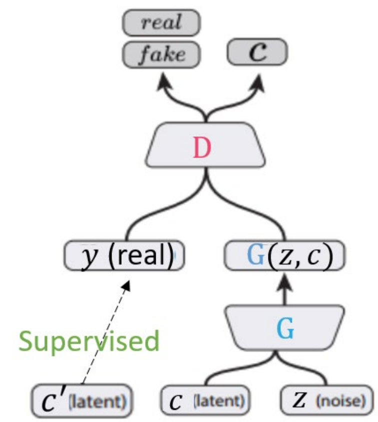
Representation Disentanglement: Conditional GAN

- Goal
  - Interpretable deep feature representation
  - Disentangle attribute of interest ***c*** from the derived latent representation ***z***
    - Unsupervised: InfoGAN
    - Supervised: AC-GAN



InfoGAN

Chen et al.
NIPS ’16



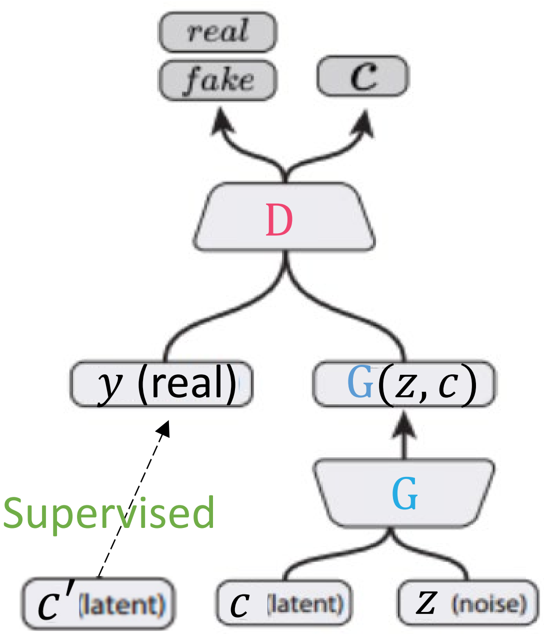
ACGAN

Odena et al.
ICML ’17

Chen et al., InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets., NIPS 2016.
Odena et al., Conditional image synthesis with auxiliary classifier GANs. ICML’17

AC-GAN

- Supervised Disentanglement



- **Learning**
  - **Overall objective function**

$$G^* = \arg\min_G \max_D \mathcal{L}_{GAN}(G, D) + \mathcal{L}_{cls}(G, D)$$

  - **Adversarial Loss**

$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}[\log(1 - D(G(z, c)))] + \mathbb{E}[\log D(y)]$$

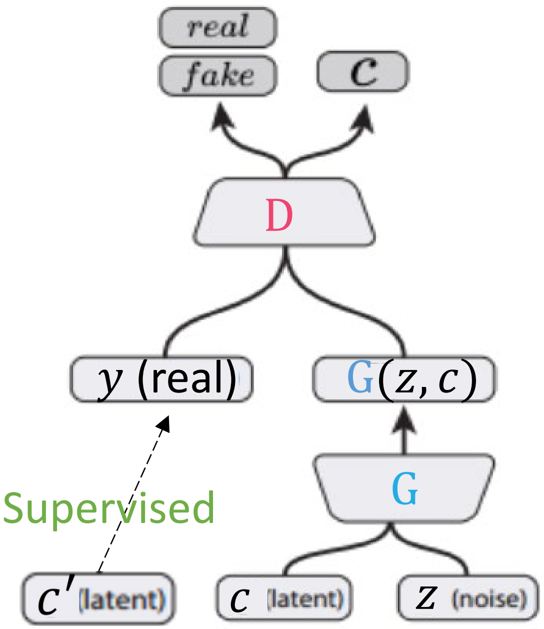
  - **Disentanglement loss**

$$\mathcal{L}_{cls}(G, D) = \underbrace{\mathbb{E}[-\log D_{cls}(c'|y)]}_{\text{Real data w.r.t. its domain label}} + \underbrace{\mathbb{E}[-\log D_{cls}(c|G(x, c))]}_{\text{Generated data w.r.t. assigned label}}$$

Odena et al., Conditional image synthesis with auxiliary classifier GANs. ICML'17

AC-GAN

- Supervised Disentanglement

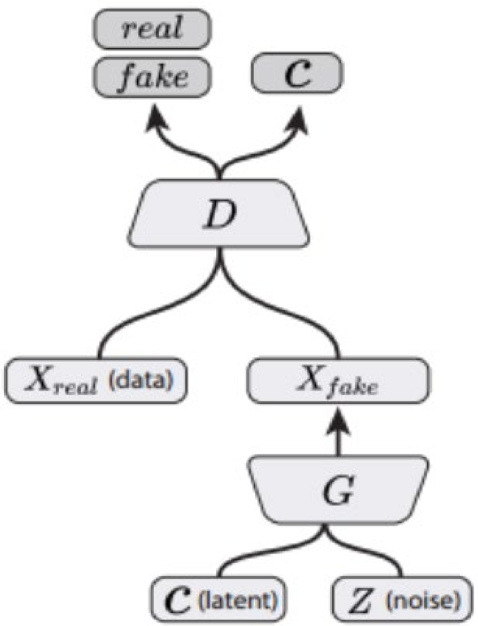


Different c values

Odena et al., Conditional image synthesis with auxiliary classifier GANs. ICML'17

InfoGAN

- Unsupervised Disentanglement



- Learning
  - **Overall objective function**

$$G^* = \arg\min_{G}\max_{D} \mathcal{L}_{GAN}(G, D) + \mathcal{L}_{cls}(G, D)$$

  - **Adversarial Loss**

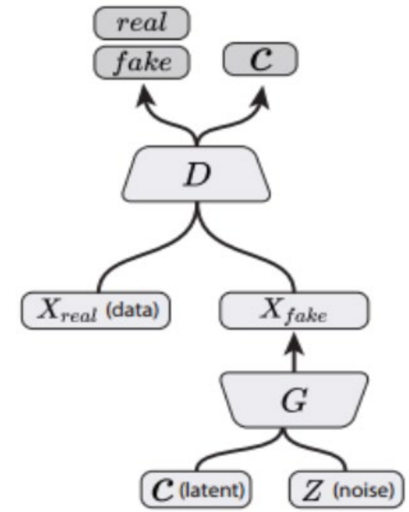
$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}[\log(1 - D(G(z, c)))] + \mathbb{E}[\log D(y)]$$

  - **Disentanglement loss**

$$\mathcal{L}_{cls}(G, D) = \underbrace{\mathbb{E}[-\log D_{\text{cls}}(c|G(x, c))]}_{\text{Generated data w.r.t. assigned label}}$$

Chen et al., InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets., NIPS 2016.

InfoGAN

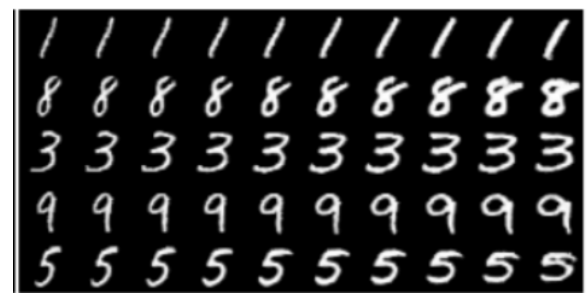


- Unsupervised Disentanglement
  - No guarantee in disentangling particular semantics
  - It can be viewed as…

Different c

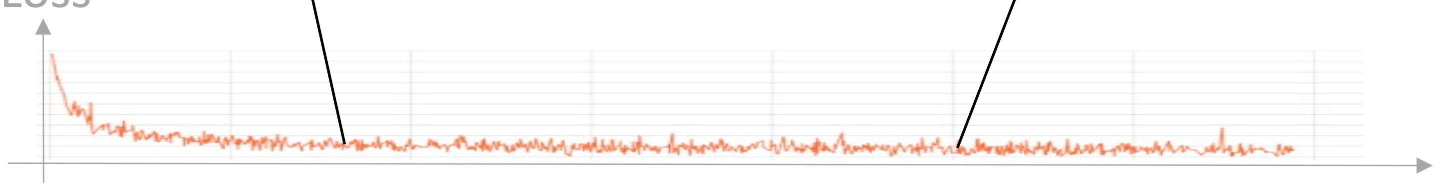
Rotation Angle

Different c



Width

Loss



Time

Training process

Chen et al., InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets., NIPS 2016.

What We've Covered Today…

- Generative Models
  - Diffusion Model
    - Conditional Diffusion Model
      - Classifier Guidance
      - Classifier-Free Guidance
      - Text/Image Guidance
    - Personalization via Diffusion Model
  - Generative Adversarial Network
  - **HW #2 is out! (due 10/29)**