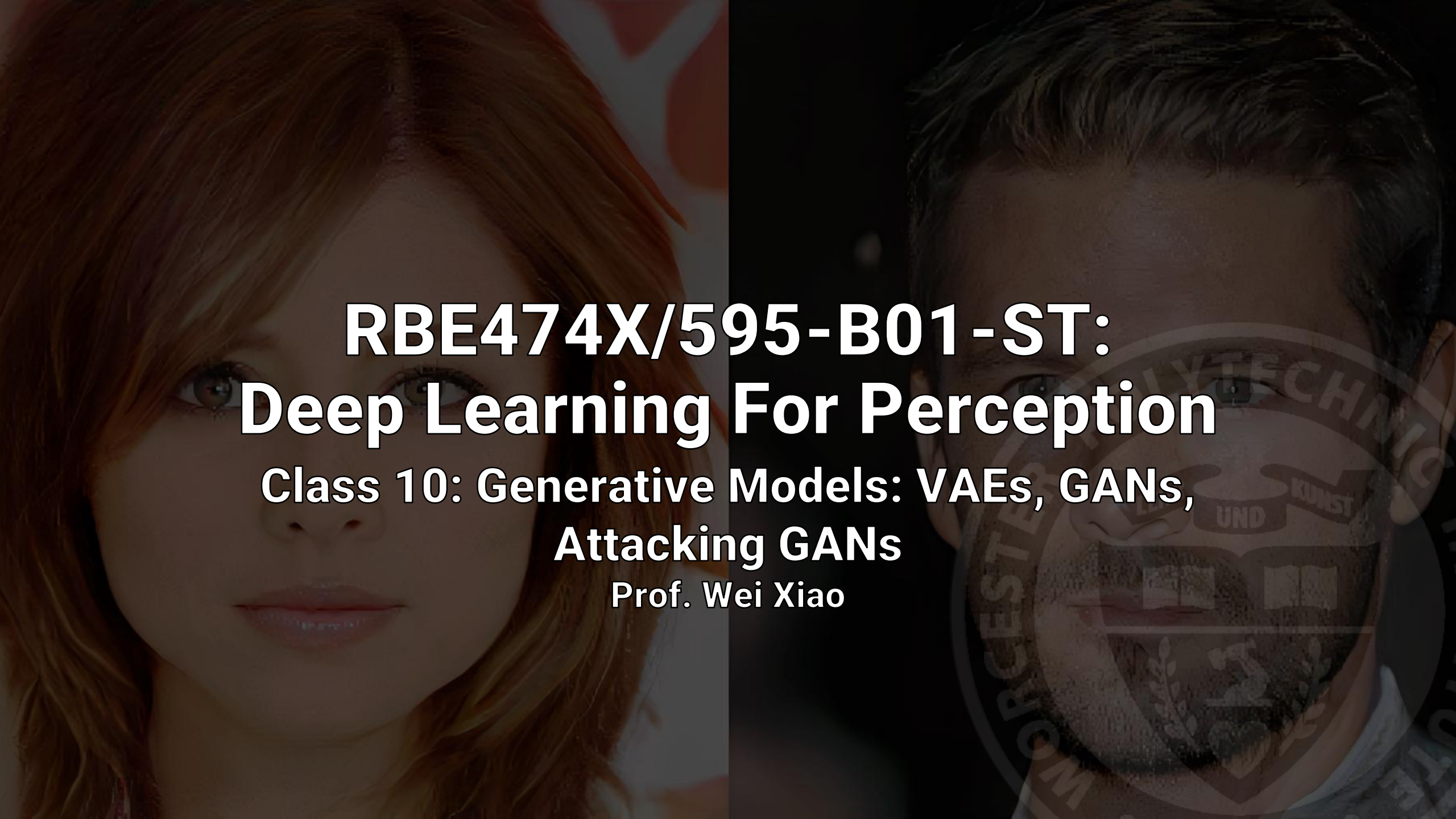


RBE474X/595-B01-ST: Deep Learning For Perception

Class 10: Generative Models: VAEs, GANs, Attacking GANs

Prof. Wei Xiao

Image Compression

1MB

Image Compression

1MB

0.1MB

Image Compression

1MB

0.1MB

We Don’t Want To Lose Details

M

N

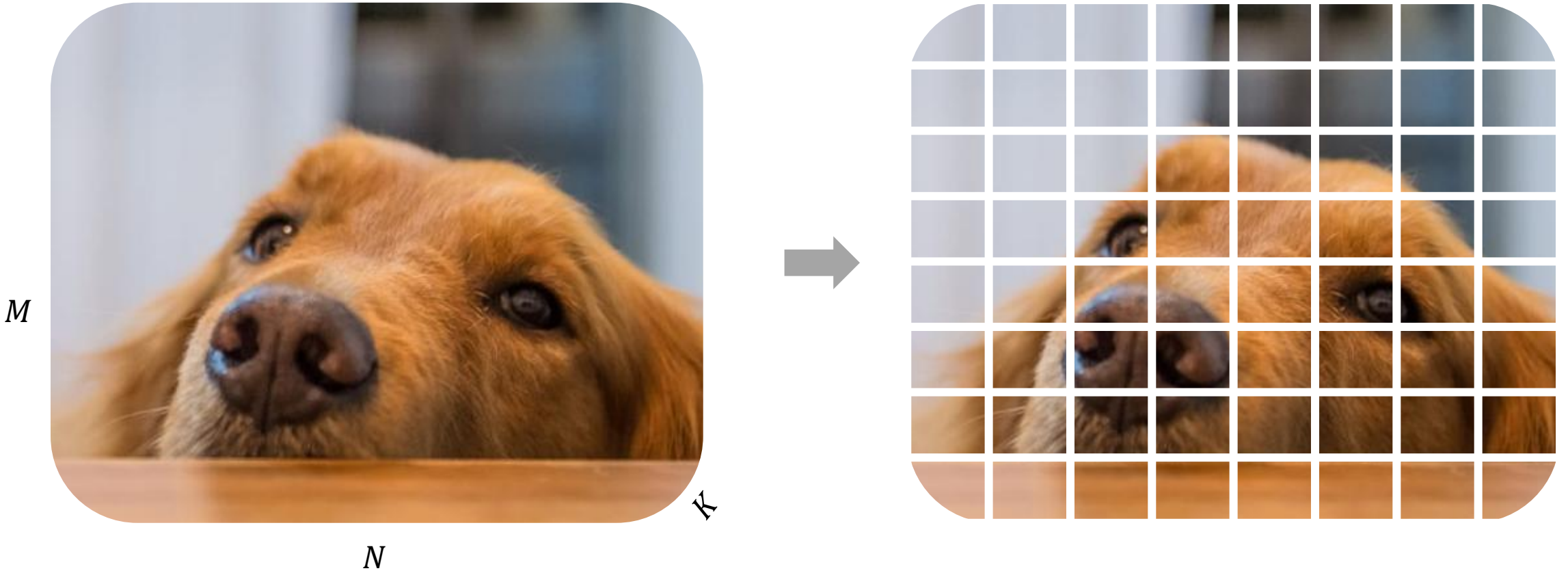
K

$$\begin{bmatrix} 0.37 \\ -40 \\ 250 \\ 0.74 \\ 0.11 \\ -0.35 \\ \vdots \\ 0.01 \end{bmatrix}$$

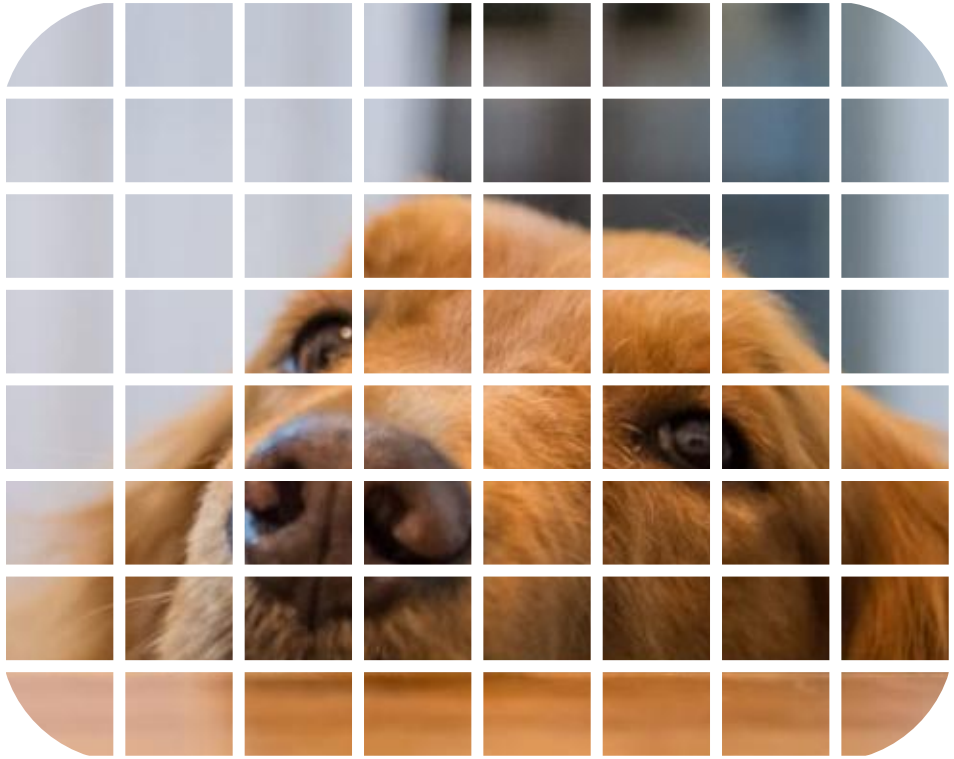
$\mathbb{R}^{D\times 1}$

$D \ll MNK$

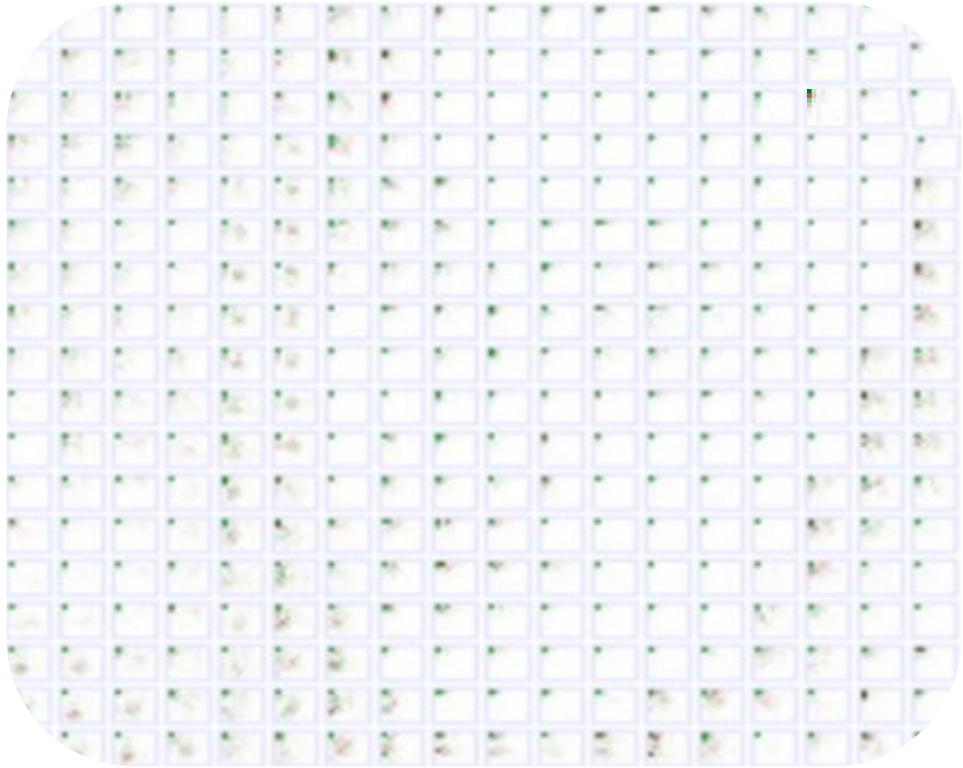
We Don’t Want To Lose Details



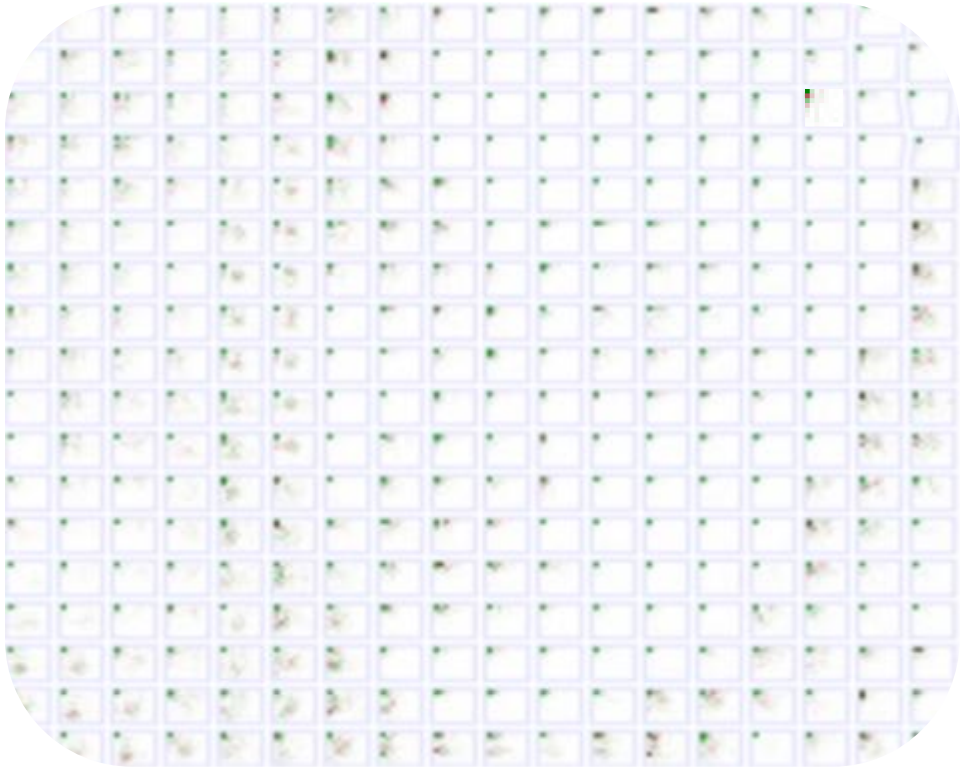
We Don’t Want To Lose Details



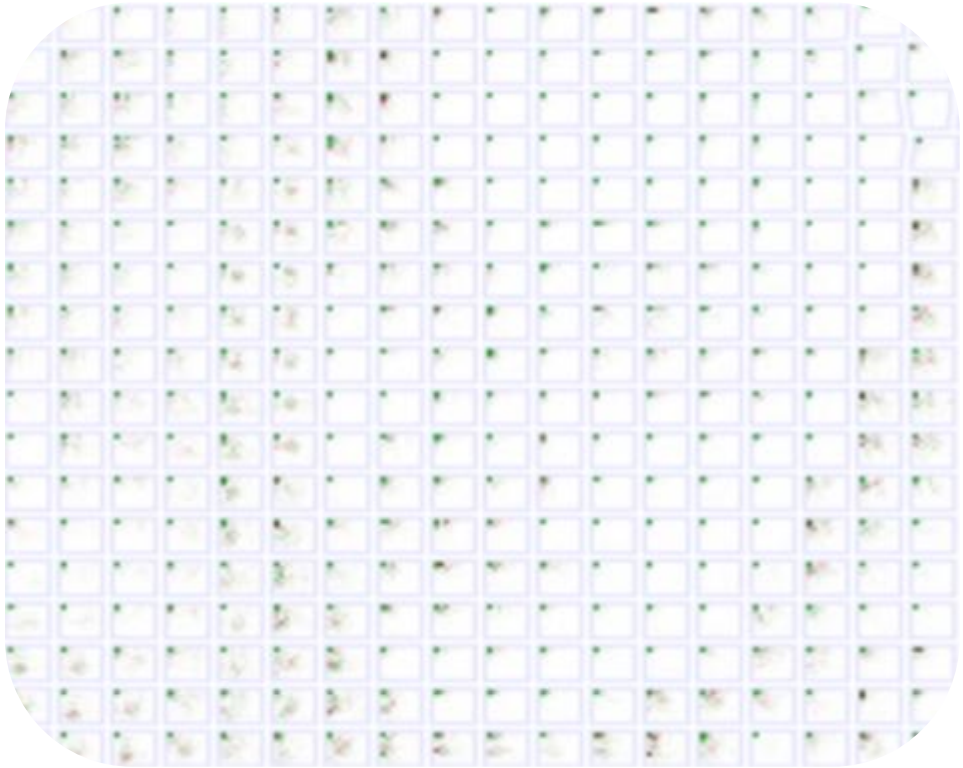
DCT(discrete cosine transform)

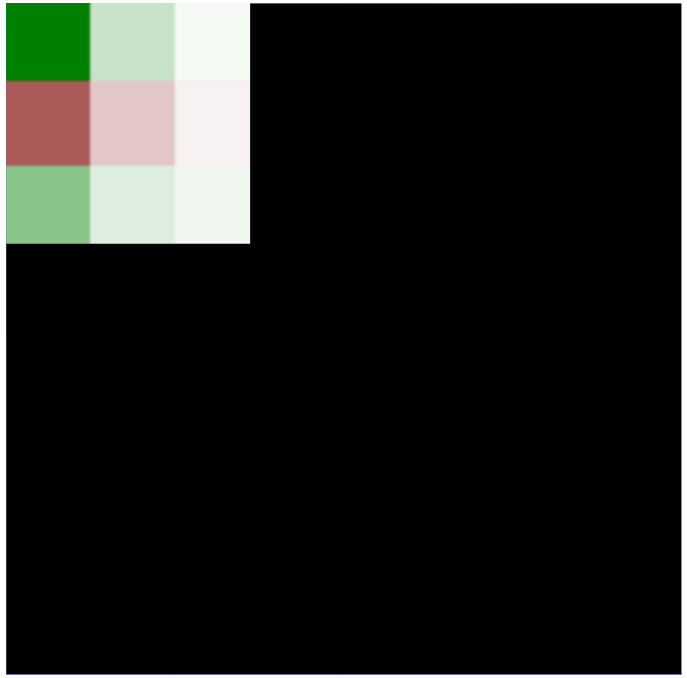


We Don’t Want To Lose Details



We Don’t Want To Lose Details



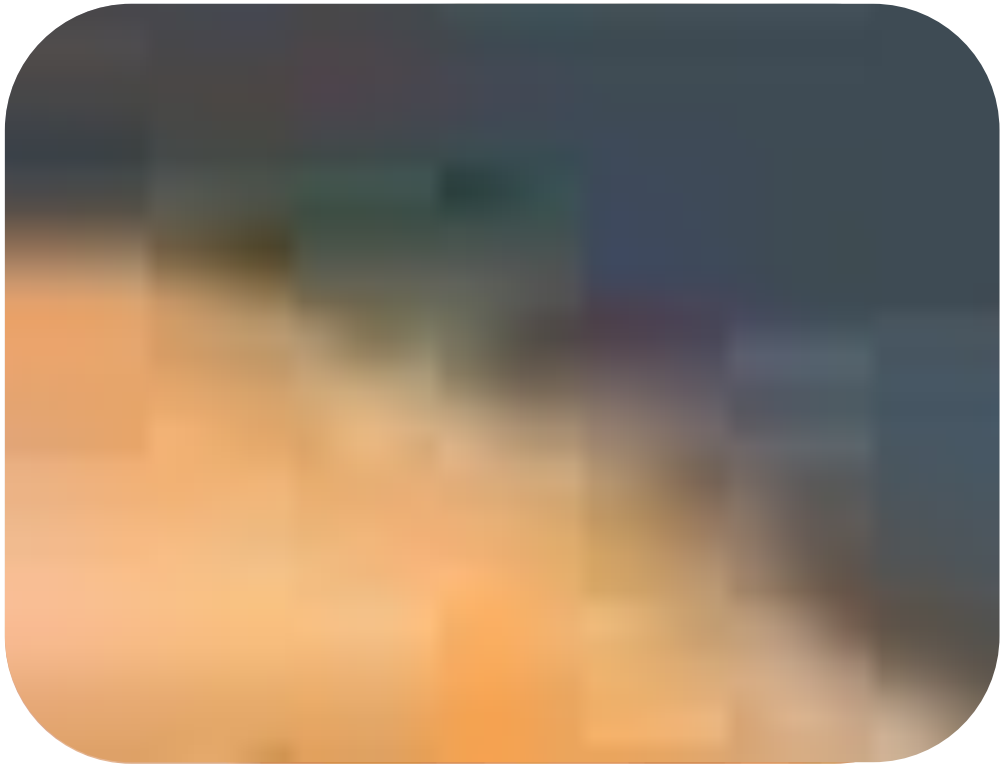


$$\begin{bmatrix} 0.37 \\ -40 \\ 250 \\ 0.74 \\ 0.11 \\ -0.35 \\ \vdots \\ 0.01 \end{bmatrix}$$

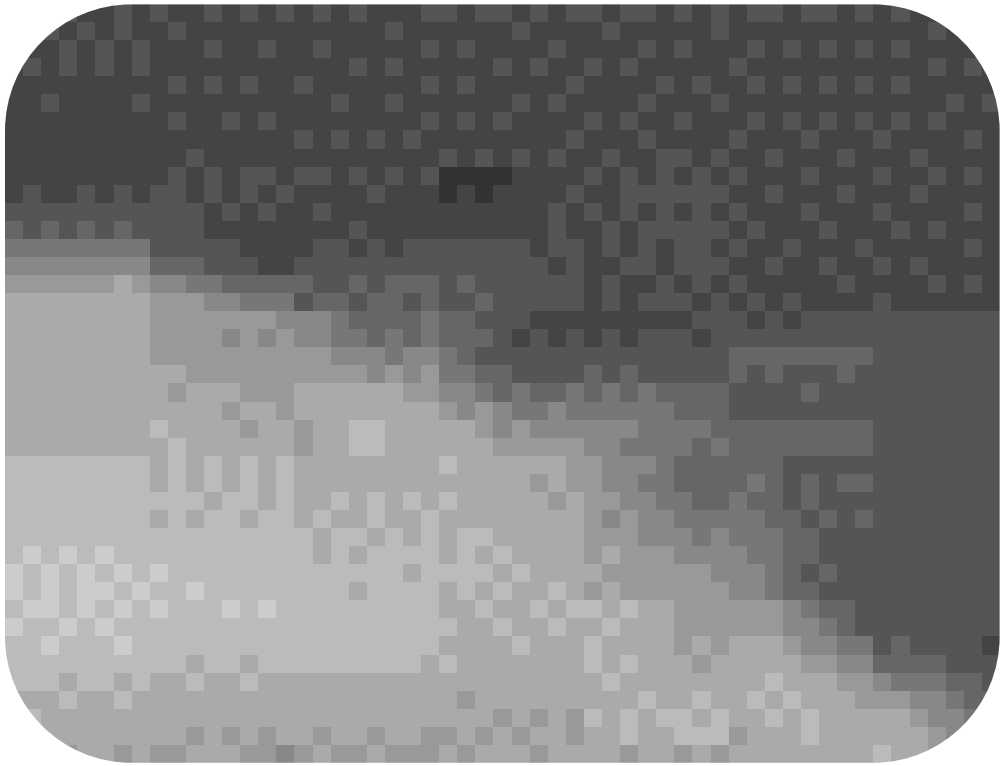
We Don’t Want To Lose Details

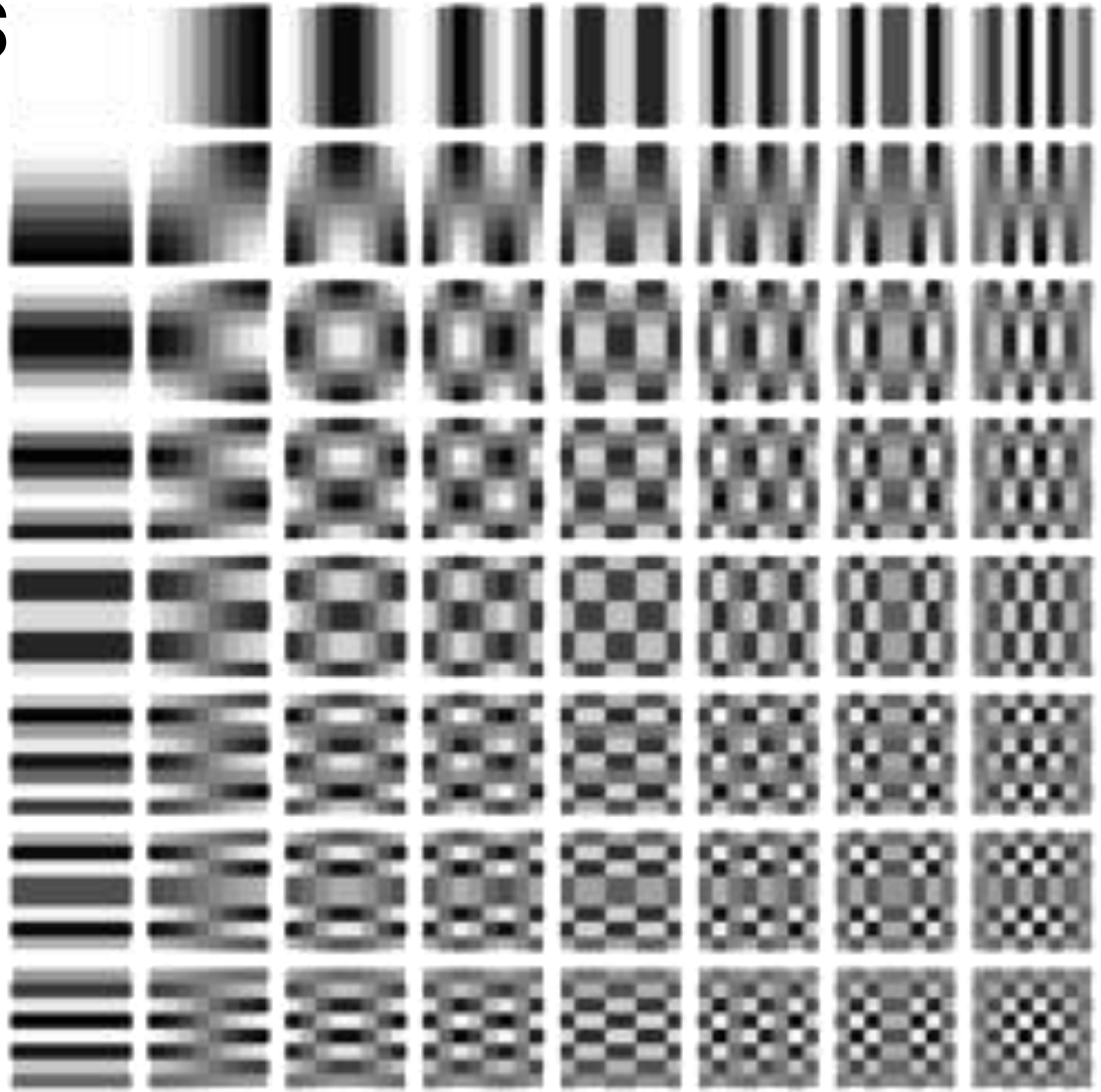
68× compression!

JPEG



DCT Basis Functions

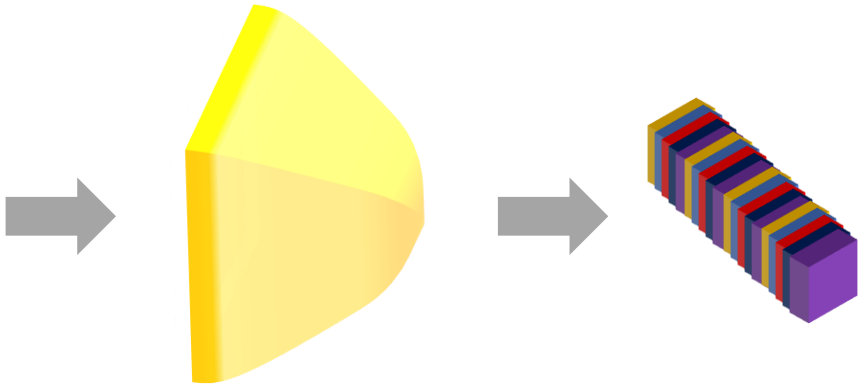




A **discrete cosine transform** (**DCT**) expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies

Era Of Deep Learning

Let's Do This With A Neural Network

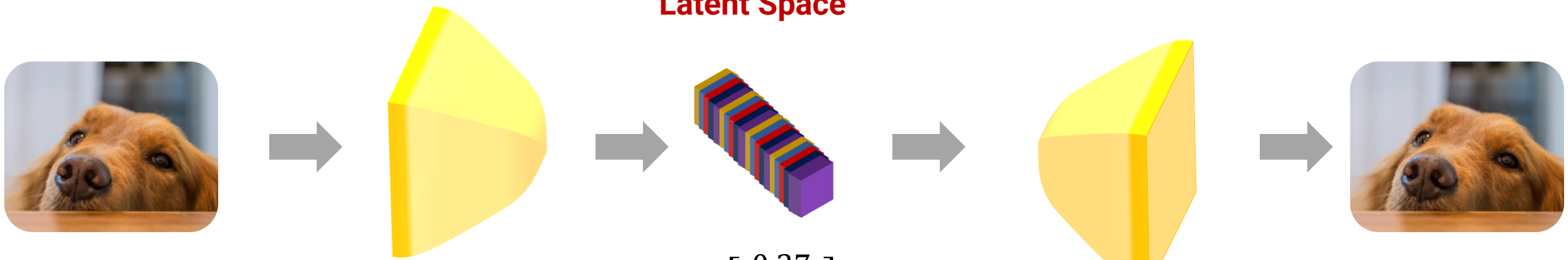


$$\begin{bmatrix} 0.37 \\ -40 \\ 250 \\ 0.74 \\ 0.11 \\ -0.35 \\ \vdots \\ 0.01 \end{bmatrix}$$

Era Of Deep Learning

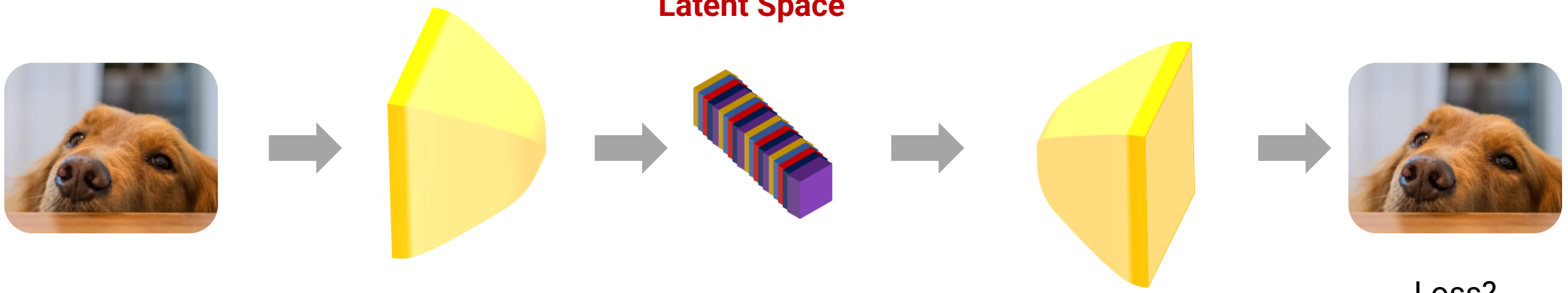
Let's Do This With A Neural Network

Lower-dimensional
Latent Space

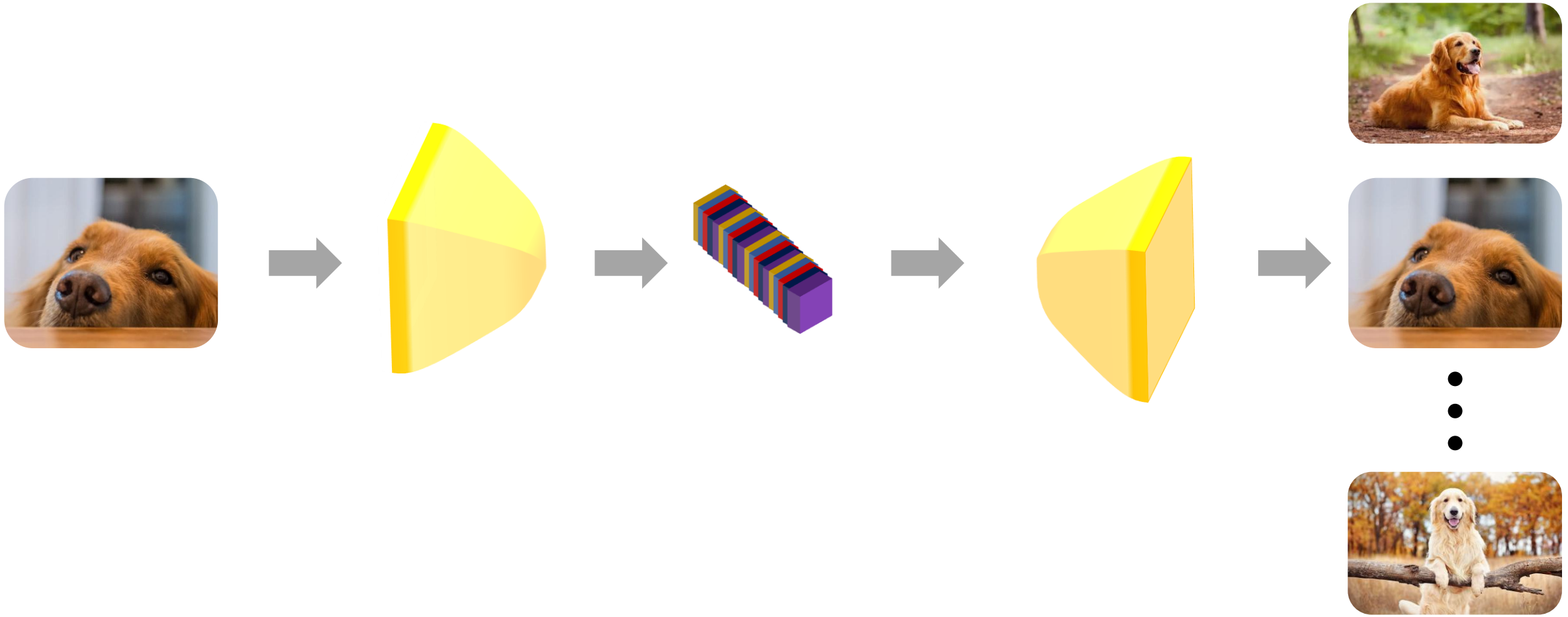


$$\begin{bmatrix} 0.37 \\ -40 \\ 250 \\ 0.74 \\ 0.11 \\ -0.35 \\ \vdots \\ 0.01 \end{bmatrix}$$

Autoencoder

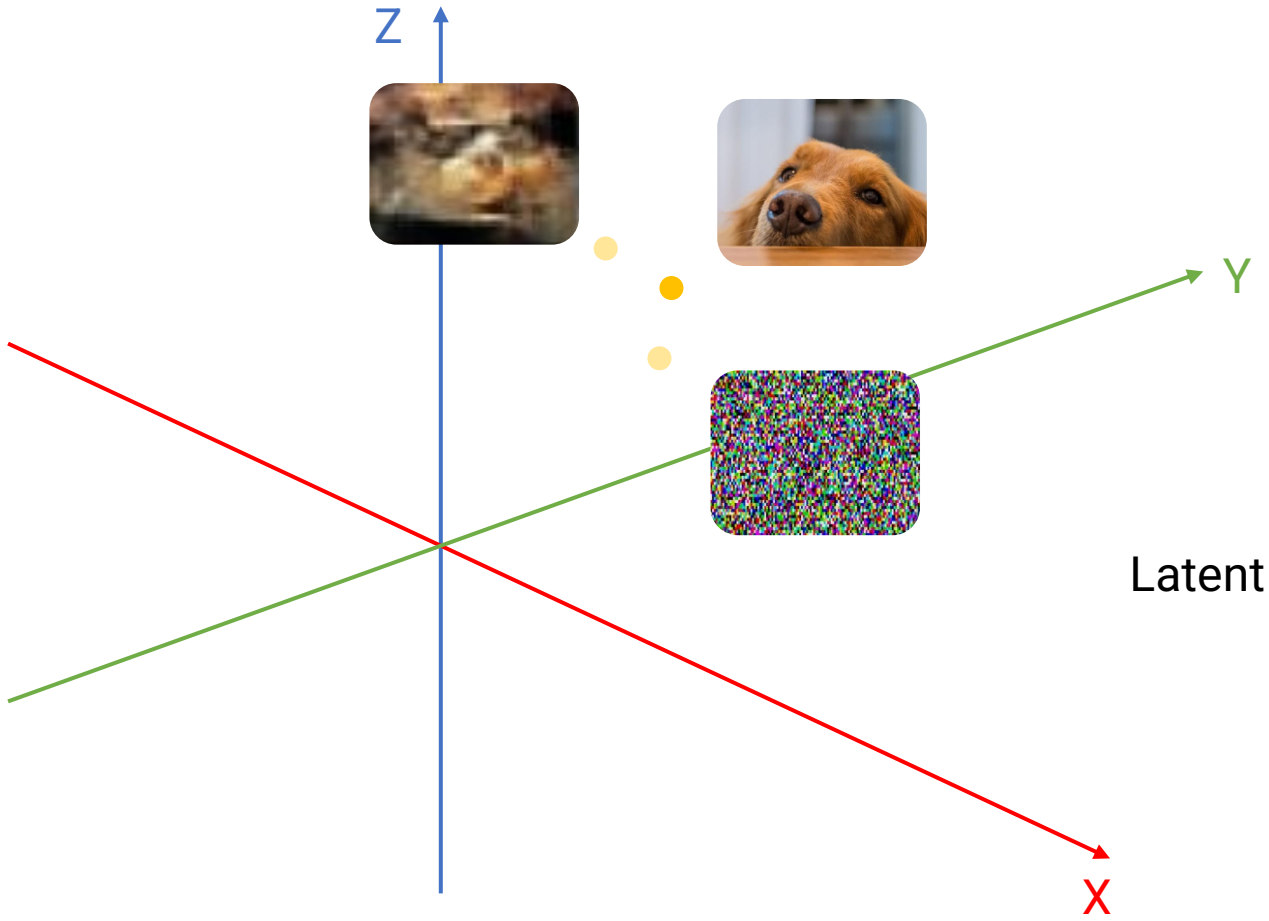


I Want All Doggo Images!



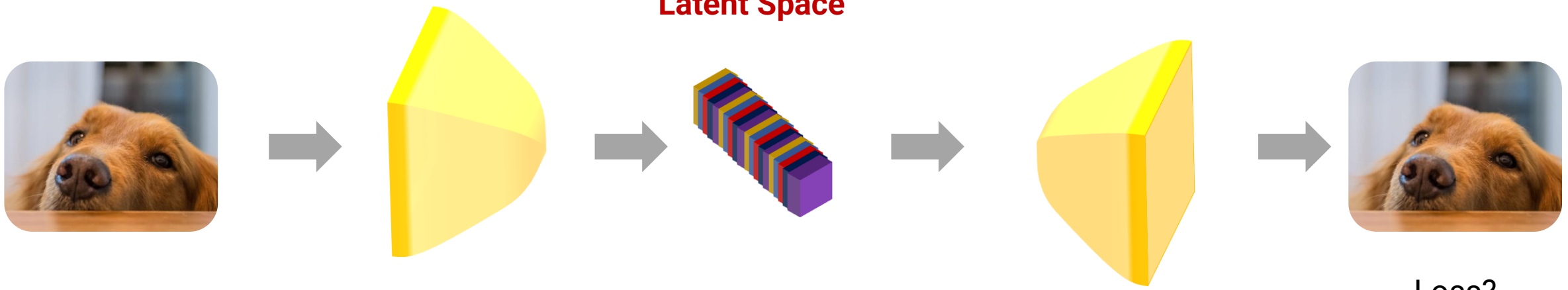
Can’t I Use Latent Space Tricks?

From Before?



Latent Space Is Not Structured! ☹

Autoencoder



Loss?
Simple pixel-wise loss!

Can only **generate** one image!
So, one NN per image ☹


Data Is The Key!

Moore’s Law for Machine Learning

“The amount of training data doubles every eighteen months.”

- Danny Lange, Unity


IMPLAUSIBLE

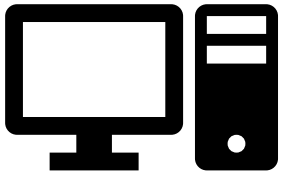

EXPENSIVE


DESTRUCTIVE

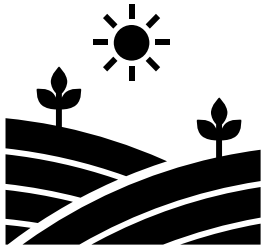

ARDUOUS

Ways To Increase Data

Augmentation



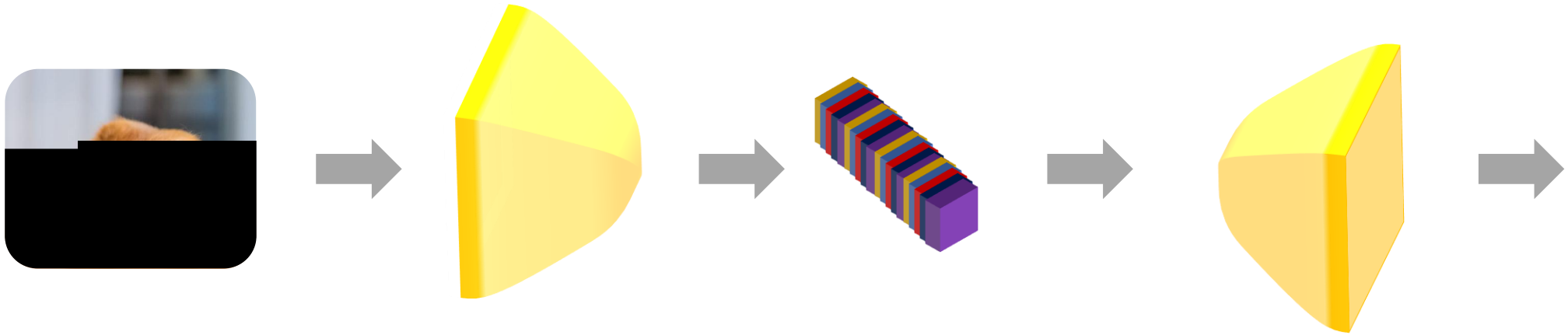
Simulation



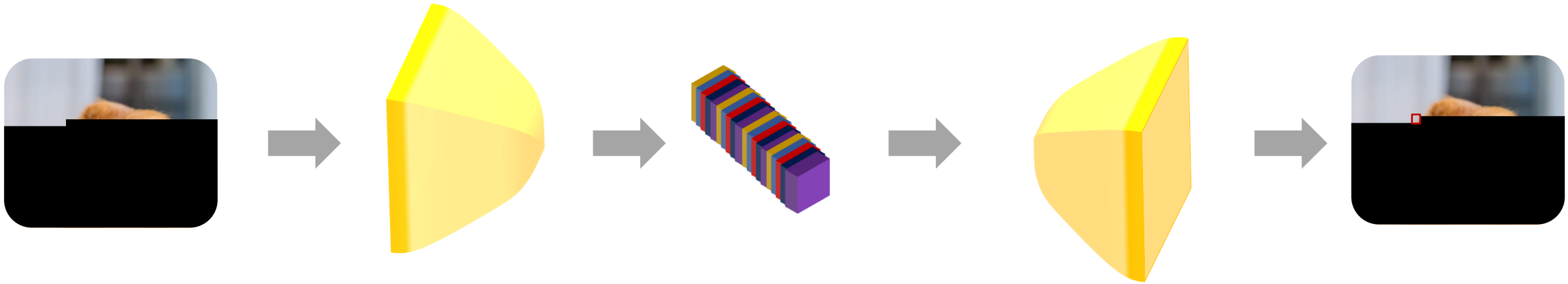
Field Collection

Crowd-source Data

I Want All Doggo Images!



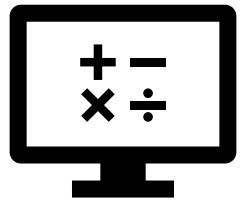
I Want All Doggo Images!



Make up an image **pixel-by-pixel**

Van den Oord, Aaron, et al. "Conditional image generation with pixelcnn decoders." Advances in neural information processing systems 29 (2016).

PixelRNN



Simplified Mathematical Model

Called a Fully Visible Belief Network

$$p(\boldsymbol{x}) = \prod_{i=1}^{n} p(\boldsymbol{x_i} | \boldsymbol{x_1}, \ldots, \boldsymbol{x_{i-1}})$$

Likelihood of image $\boldsymbol{x}$

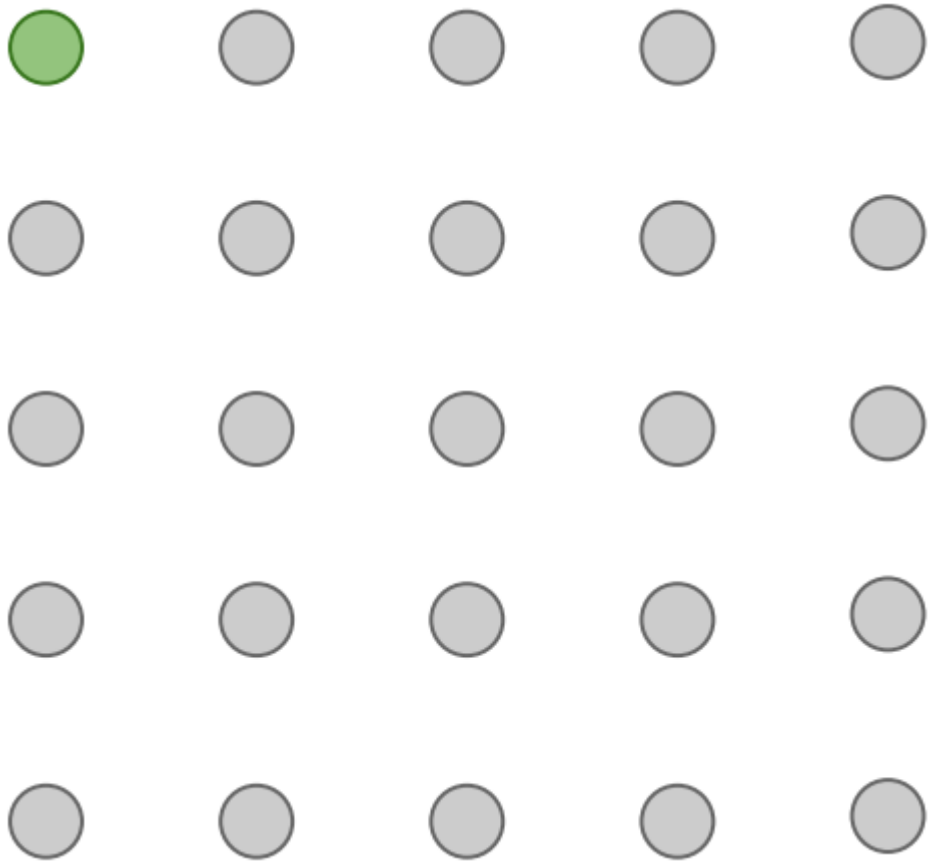
Probability of i^{th} pixel value given all previous values

Maximize the likelihood of training data!

Slide idea from Stanford's CS231n

PixelRNN

Image generation starts from corner

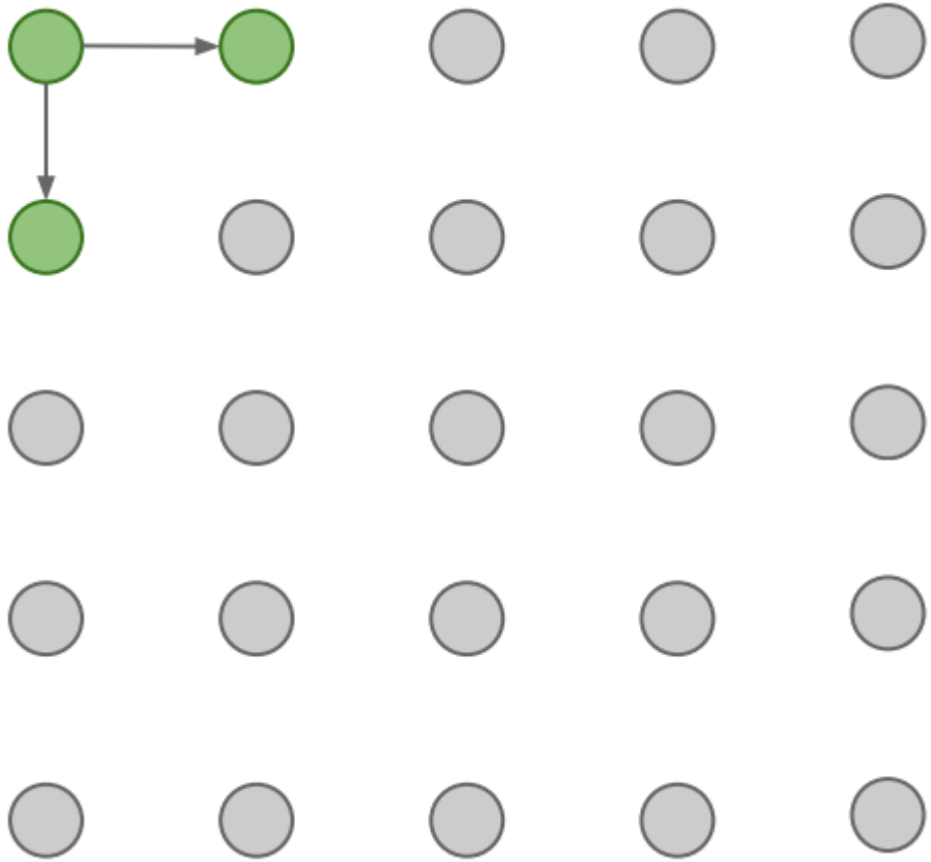


cs231n_2019_lecture11.pdf (stanford.edu)

PixelRNN

Image generation starts from corner

Dependency on previous pixels modeled using an RNN (LSTM)

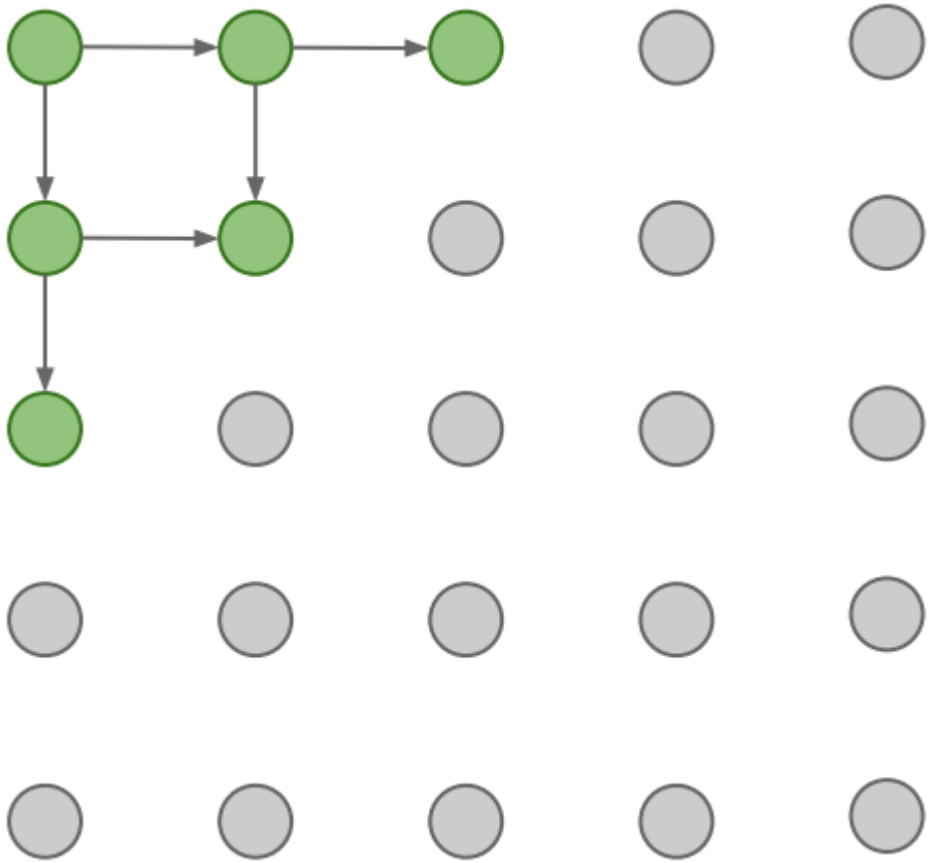


cs231n_2019_lecture11.pdf (stanford.edu)

PixelRNN

Image generation starts from corner

Dependency on previous pixels modeled using an RNN (LSTM)



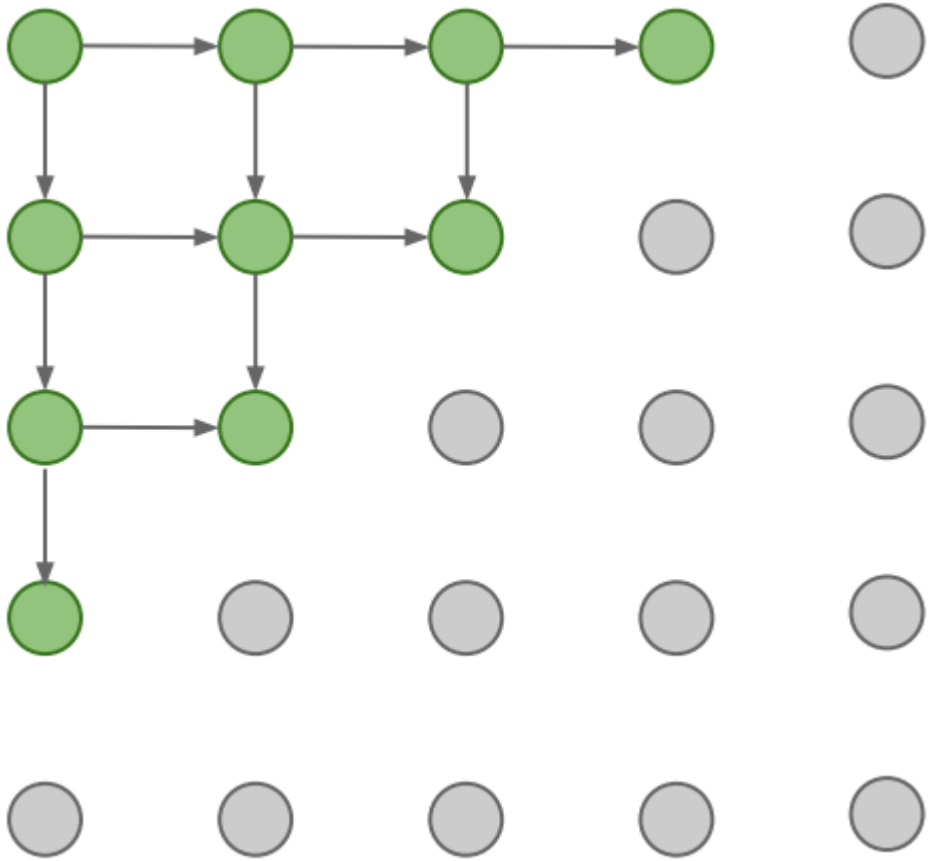
cs231n_2019_lecture11.pdf (stanford.edu)

PixelRNN

Image generation starts from corner

Dependency on previous pixels
modeled using an RNN (LSTM)

Issue?
Very Very Slow ☹



cs231n_2019_lecture11.pdf (stanford.edu)

PixelCNN

Still generate image pixels starting from corner

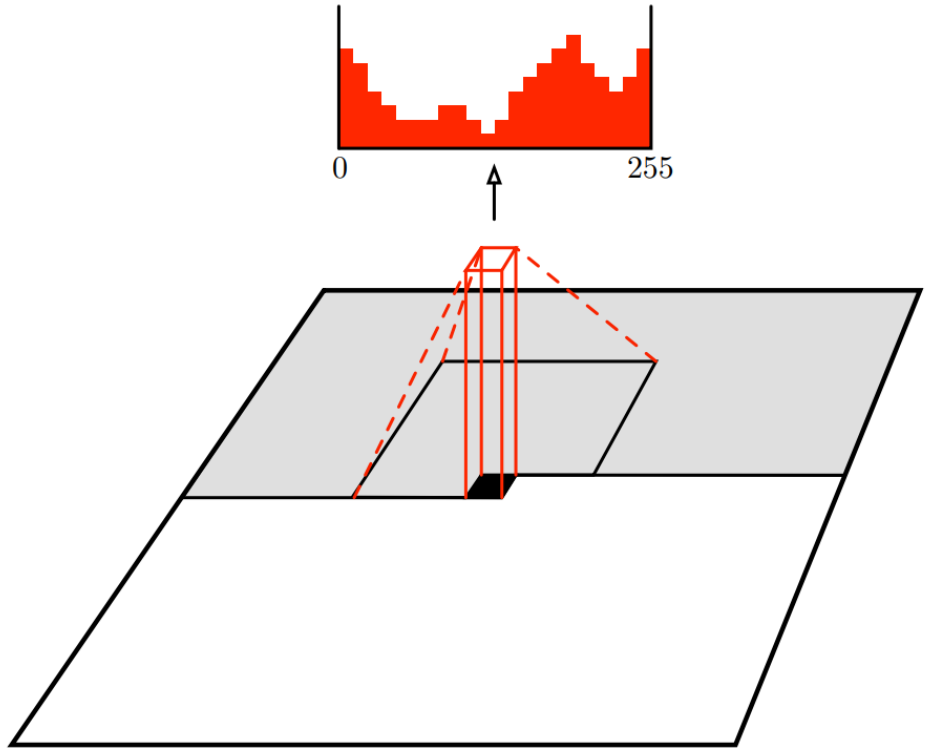
Dependency on previous pixels now modeled using a CNN over context region

Trained using Maximum Likelihood

$$p(\boldsymbol{x}) = \prod_{i=1}^{n} p(\boldsymbol{x_i} | \boldsymbol{x_1}, \ldots, \boldsymbol{x_{i-1}})$$

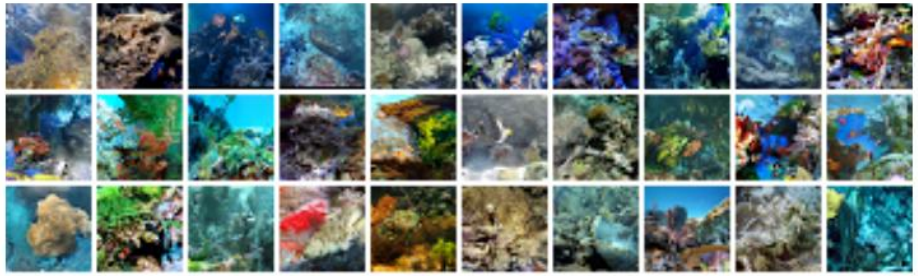
Much faster than PixelRNN: Can parallelize

Issue?
Still pixel by pixel, so slow ☹



Works?

African elephant



Coral Reef

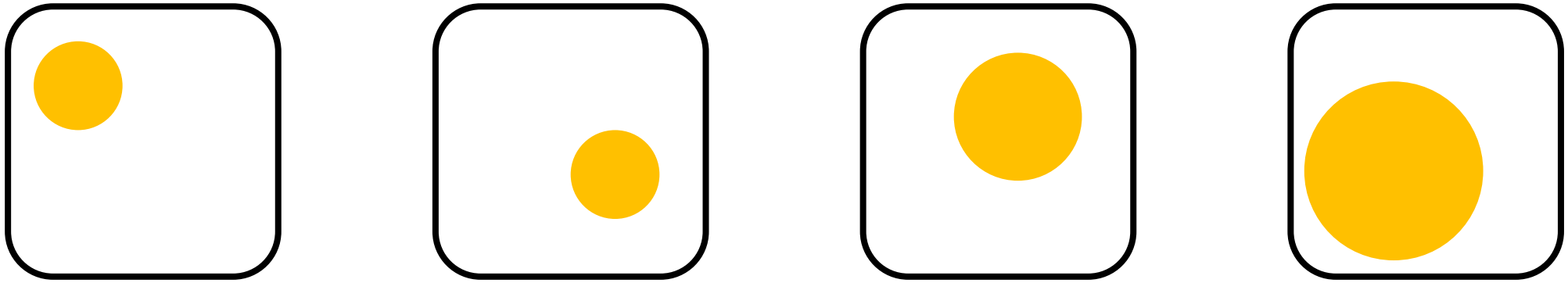
Sandbar

Sorrel horse

Lhasa Apso (dog)

Lawn mower

Toy Example



Circles!

- Same color
- Always fully enclosed
- Only variables:
  - Size
  - Location

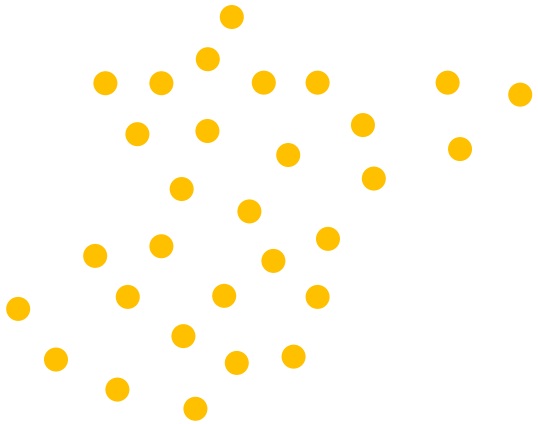
Slides inspired from: Understanding Variational Autoencoders (VAEs) | Deep Learning (youtube.com)

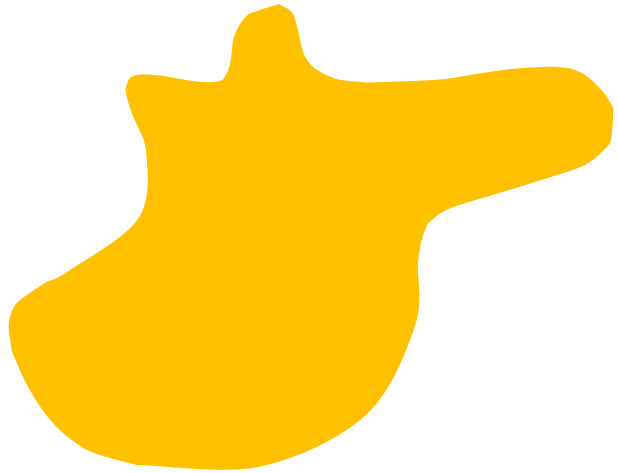
What Do We Actually Have?

$p^*(x)$

$x \sim D$

$p_\theta(x)$





$p^*(x) \approx p_\theta(x)$

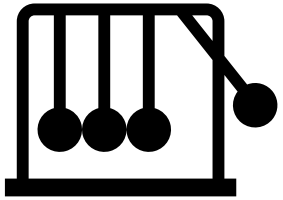
Likelihood

Find θ that maximizes the likelihood of data observed!

Maximum Likelihood Estimation!

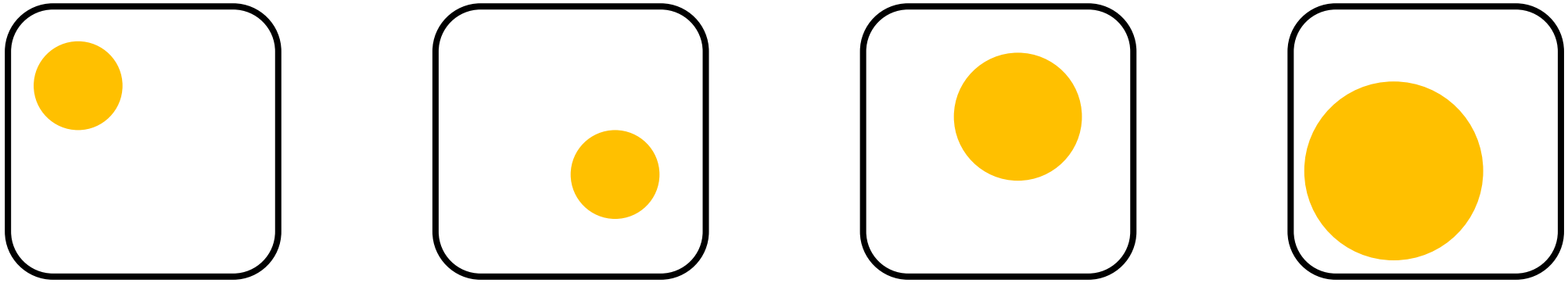
Toy Example

How would you get more data?



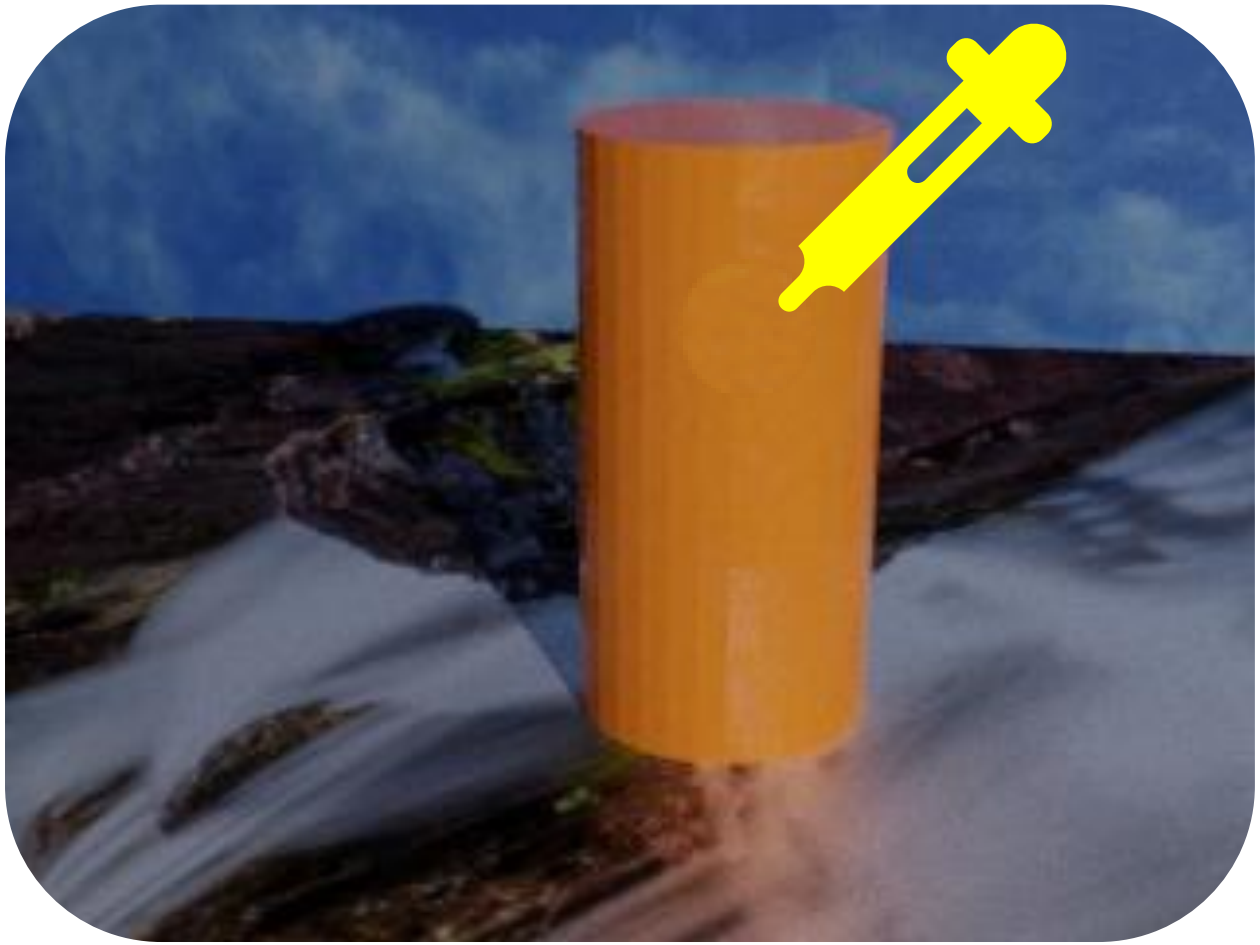
Simplified Mathematical/Physical Model

Data x

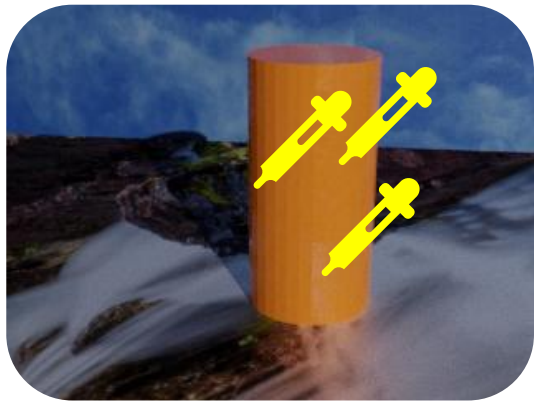


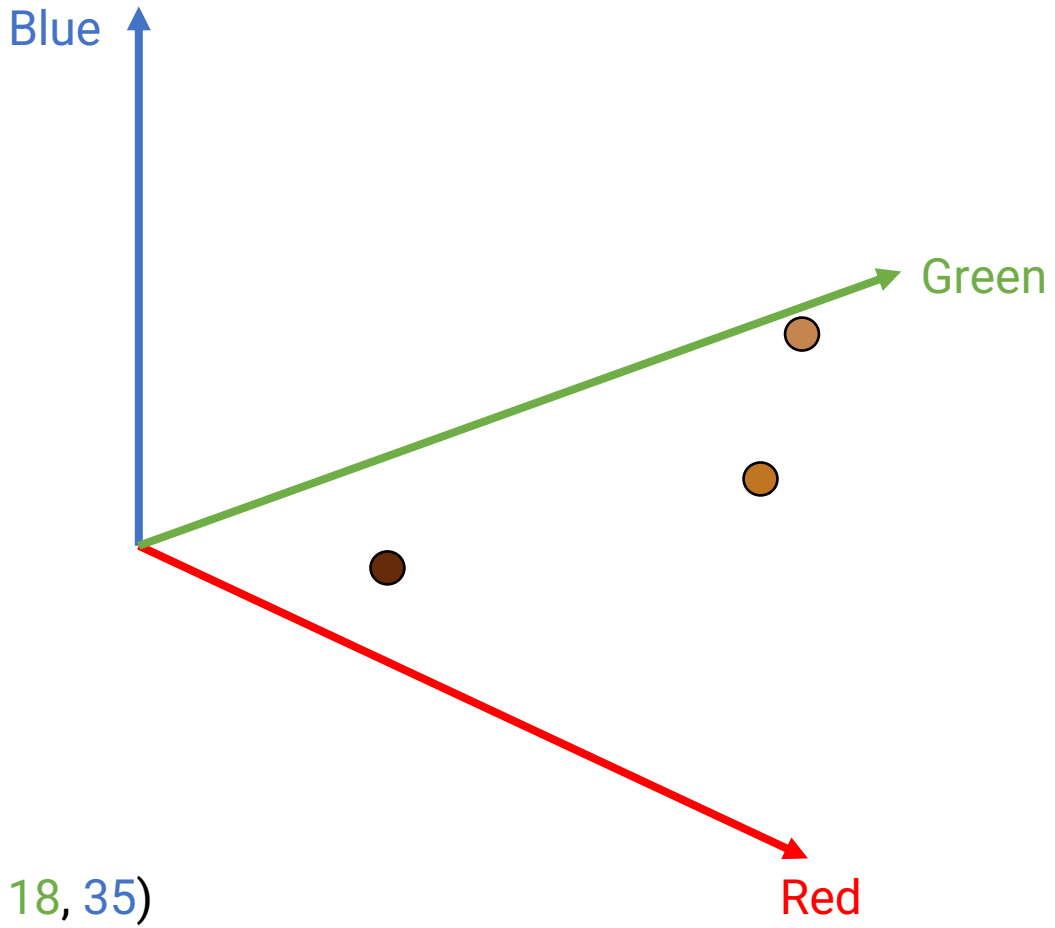
Latent Variables $z = [r, p]^T$

Is The Shape Actually Complex?



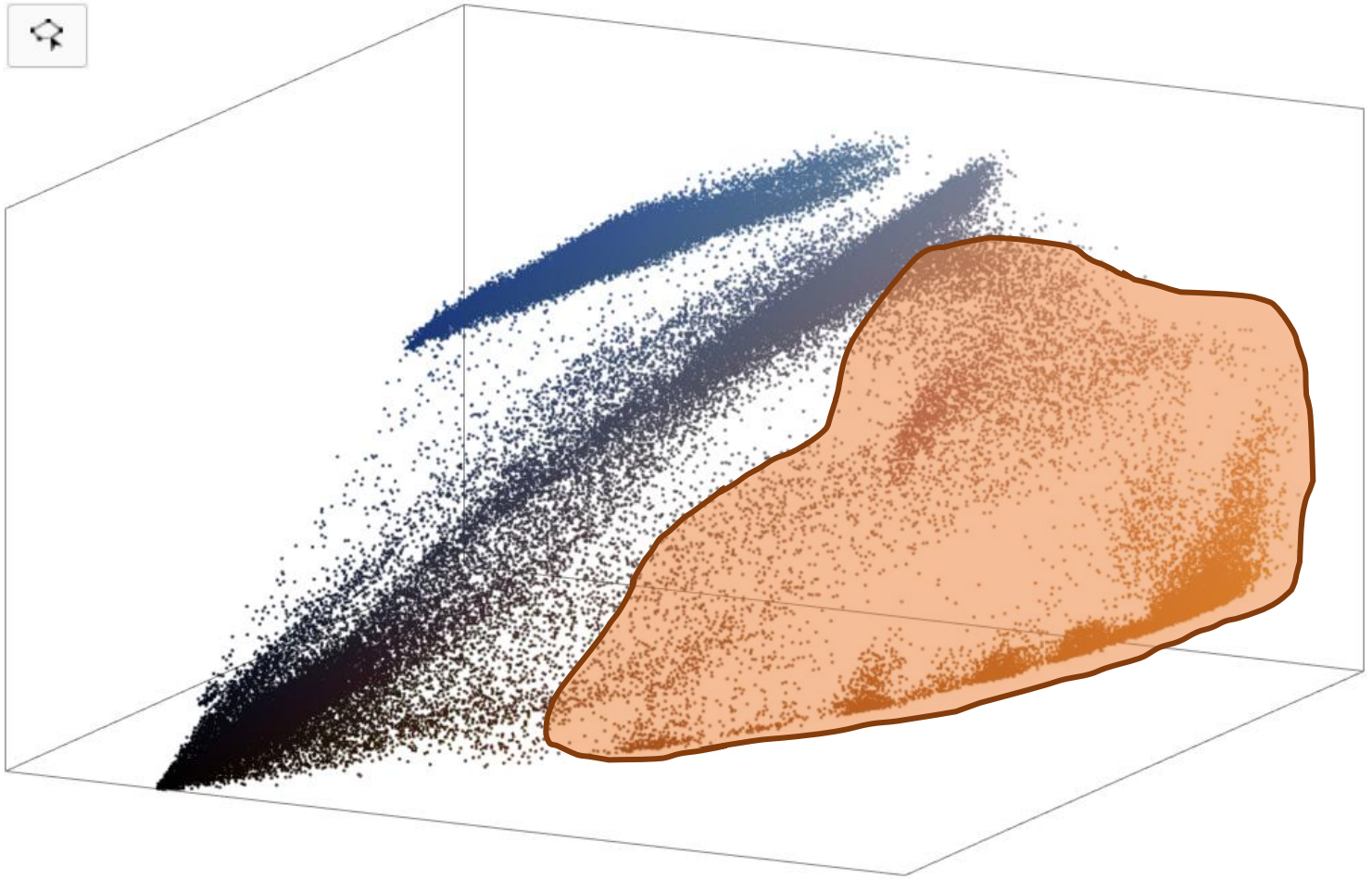
Is The Shape Actually Complex?



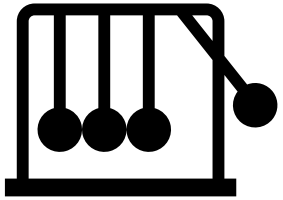


(192, 118, 35)

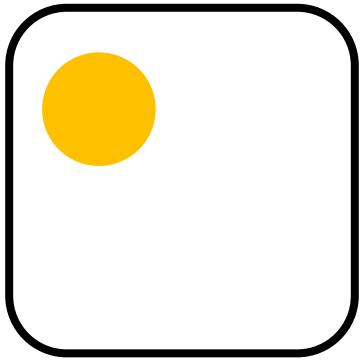
Is The Shape Actually Complex?

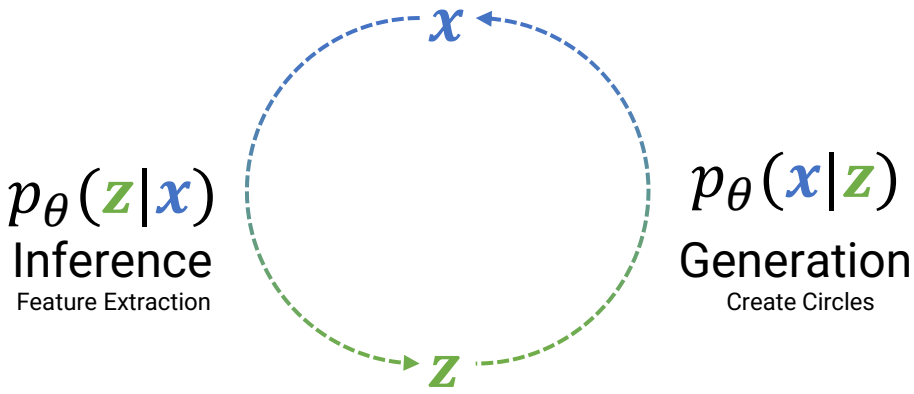


We Can Do Cool Stuff!

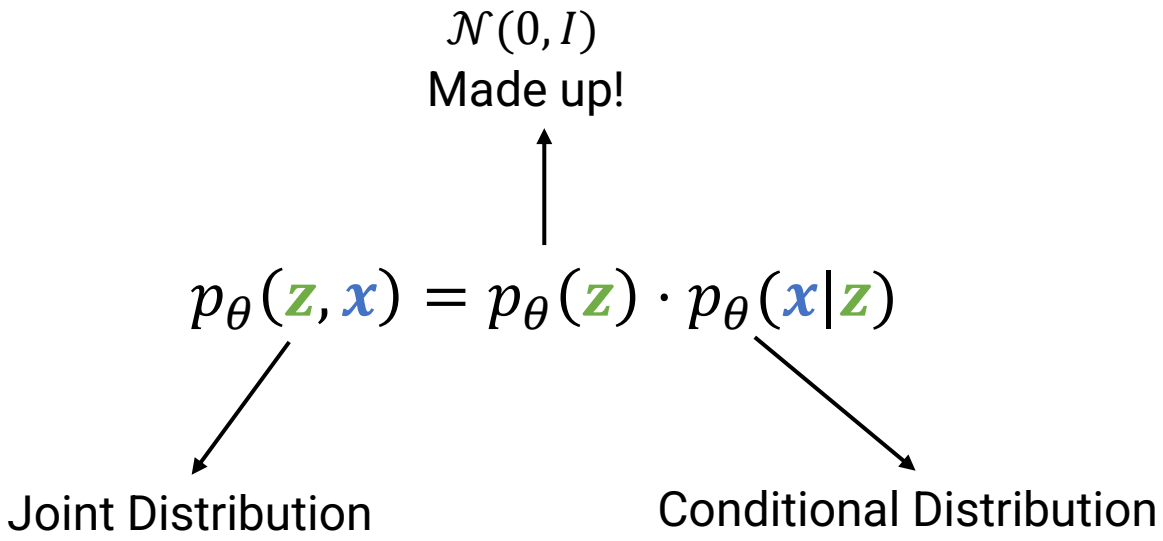


Simplified Mathematical/Physical Model



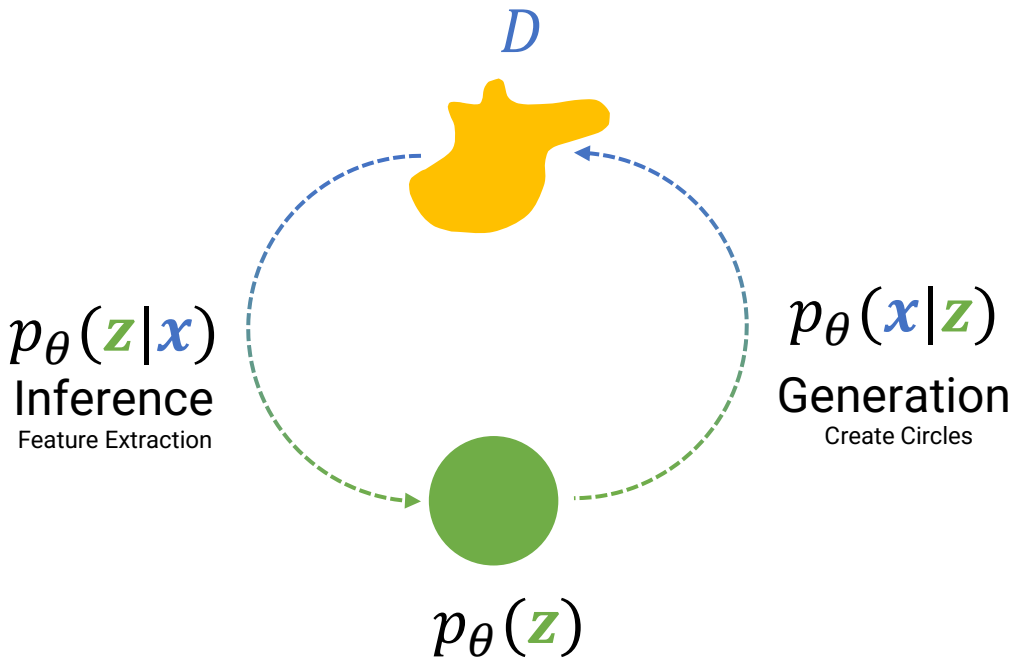


Time To Warm Up Probabilities!



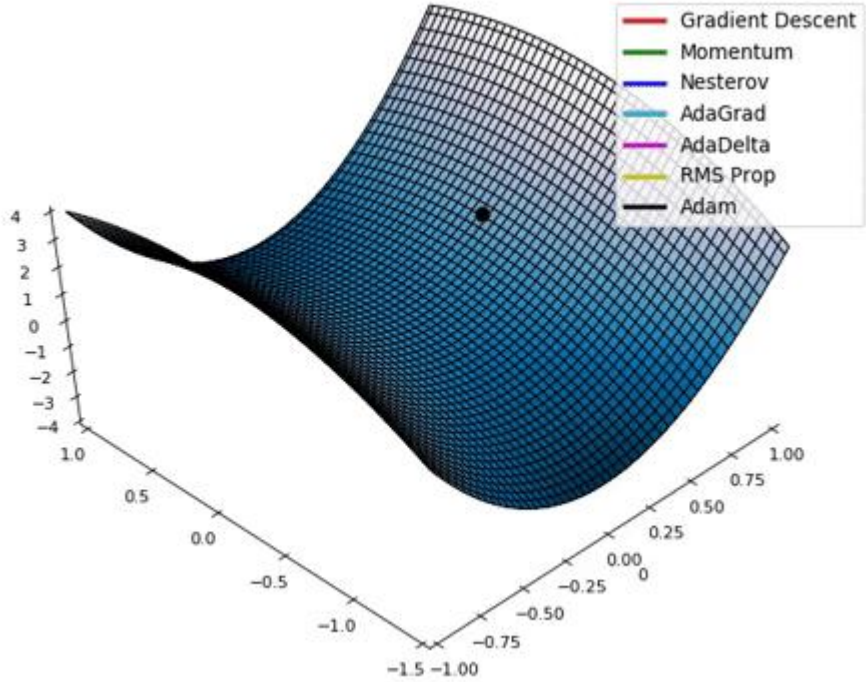
Variational Autoencoders | Generative AI Animated (youtube.com)

Esentially


D
$p_\theta(z|x)$
Inference
Feature Extraction
$p_\theta(x|z)$
Generation
Create Circles
$p_\theta(z)$

Story Time

Diederik P. Kingma



ADAM optimizer

Kingma, Diederik P. "Auto-encoding variational bayes." arXiv preprint arXiv:1312.6114 (2013).

Marginal Likelihood

Marginal Likelihood

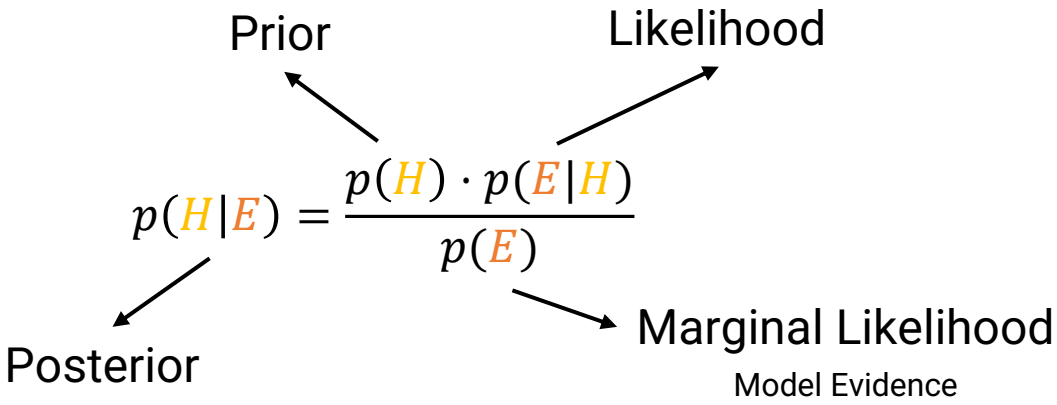
$$p_\theta(\boldsymbol{x}) = \int_{\boldsymbol{z}} p_\theta(\boldsymbol{z}, \boldsymbol{x})$$

Want it high for data point $\boldsymbol{x} \in D$

$$= \int_{\boldsymbol{z}} p_\theta(\boldsymbol{z}) \cdot p_\theta(\boldsymbol{x}|\boldsymbol{z}) d\boldsymbol{z}$$

Intractable

Bayes' Rule



$$p(z|x) = \frac{p(z) \cdot p(x|z)}{p(x)}$$

We want to find θ to maximize:

$$p_\theta(x) = \frac{p_\theta(z, x)}{p(z|x)}$$

Intractable ☹

Variational Inference

$$q_\phi(\boldsymbol{z}|\boldsymbol{x}) \approx p_\theta(\boldsymbol{z}|\boldsymbol{x})$$

Approximation Target

If p is Bayesian ⇒ Variational Bayes

For same ϕ over all data ⇒ Amortized VI

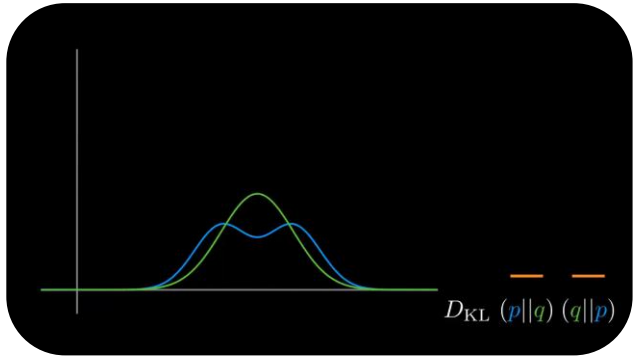
We want to push the two distributions to be similar!

Kullback-Leibler (KL) Divergence

We need to measure how "close" two distributions are

$$D_{KL}(Q\|P) = \int_{z} Q(z)\log\left(\frac{Q(z)}{P(z)}\right) dz \geq 0$$

Expected Information Loss due to approximation



$$D_{KL}(q_\phi\|p_\theta) = \log p_\theta\,(x) - \mathbb{E}_{q_\phi}[\log p_\theta\,(z,x) - \log q_\phi]$$

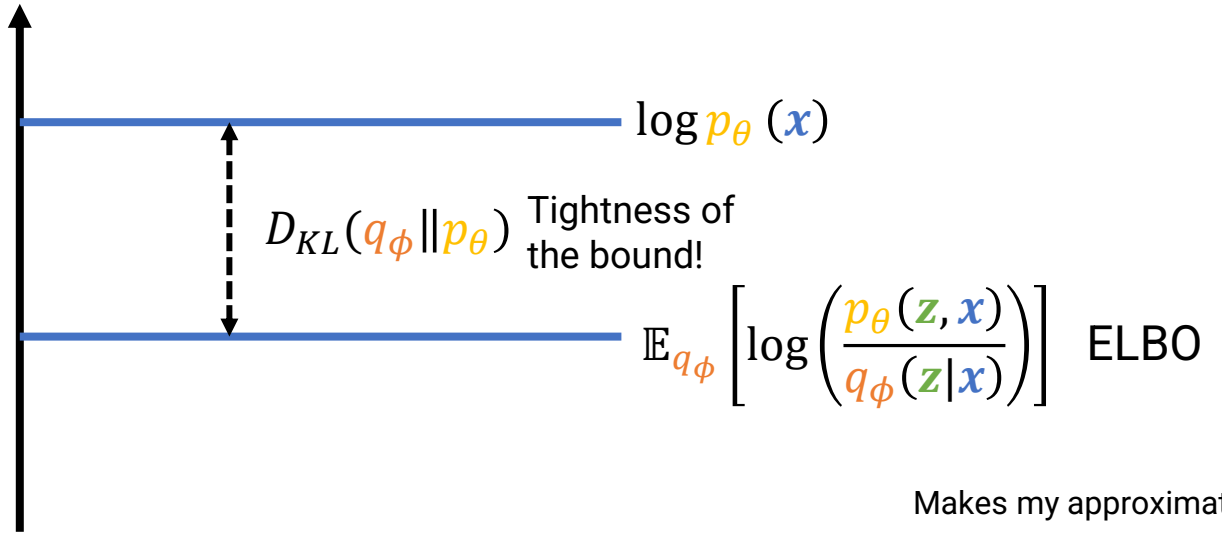
$$\log p_\theta\,(x) = D_{KL}(q_\phi\|p_\theta) + \mathbb{E}_{q_\phi}[\log p_\theta\,(z,x) - \log q_\phi]$$

Understanding Variational Autoencoders (VAEs) | Deep Learning (youtube.com)

ELBO

$$\log p_\theta(x) = D_{KL}(q_\phi \| p_\theta) + \mathbb{E}_{q_\phi}[\log p_\theta(z, x) - \log q_\phi]$$

$$\log p_\theta(x) \geq \mathbb{E}_{q_\phi}[\log p_\theta(z, x) - \log q_\phi]$$ **Evidence Lower Bound (ELBO)**



Makes my approximation better!

ELBO↑ $\log p_\theta(x) \uparrow$ $D_{KL}(q_\phi \| p_\theta) \downarrow$

Makes my model more correct!

Maximize ELBO!

Should be easy right?
No ☹

We can't compute gradients easily!
No $\nabla_\phi q_\phi$ ☹

What can we do?
Approximate!

Reparametrization Trick

$z = g(\phi, x, \epsilon)$

$\epsilon \sim p(\epsilon)$

Trick is to make all stochasticity to come from ϵ
And maintain everything else to be deterministic ☺

$$\mathcal{L}(x) = \mathbb{E}_{p(\epsilon)}\left[\log p_\theta(z, x) - \log q_\phi(z|x)\right]$$

$$\nabla_{\theta,\phi}\mathcal{L}(x) \approx -\frac{1}{L}\sum_{i=1}^{L} \nabla_{\theta,\phi}\left[\log p_\theta(z, x) - \log q_\phi(z|x)\right]$$ Monte Carlo Sampling

Stochastic Gradient Variational Bayes (SGVB)

$$\text{ELBO} = \mathbb{E}_{p(\epsilon)}\left[\log\left(\frac{p_\theta(z, x)}{q_\phi(z|x)}\right)\right]$$

$$\text{ELBO} \approx \left[\frac{1}{L}\sum_{i=1}^{L} \log p_\theta(x|z)\right] - D_{KL}(q_\phi(z|x)\|p_\theta(z))$$ Slow to compute! ☹

All Hail The Gaussian!

$$q_\phi(\boldsymbol{z}|\boldsymbol{x}) = g(\phi, \boldsymbol{x}, \epsilon) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$$

$$\epsilon \sim p(\epsilon) \qquad \text{Sampling}$$

$$p_\theta(\boldsymbol{z}) = \mathcal{N}(0, \boldsymbol{I})$$

$$\boldsymbol{\mu}, \boldsymbol{\sigma} = f(\boldsymbol{x}) \qquad \text{Deterministic}$$

$$\text{ELBO} \approx \left[\frac{1}{L}\sum_{i=1}^{L} \log p_\theta(\boldsymbol{x}|\boldsymbol{z})\right] - D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \| p_\theta(\boldsymbol{z}))$$

$$\approx \left[\frac{1}{L}\sum_{i=1}^{L} \log p_\theta(\boldsymbol{x}|\boldsymbol{z})\right] + \frac{1}{2}[\log \boldsymbol{\sigma}^2 - \boldsymbol{\mu}^2 - \boldsymbol{\sigma}^2 + 1]$$

Done over Mini batch
Called Single Sample Monte Carlo

How Do You Compute $p_\theta(x|z)$?

$$y \sim p_\theta(x|z)$$

V7 Chairs (Real-valued)

$p_\theta(x|z) = \mathcal{N}(y_\mu, y_\sigma)$

Gaussian!

Binarized MNIST (Binary-valued)

$p_\theta(x|z) = \text{Bernoulli}(y)$

Bernoulli!

Practically

$$\log p_\theta(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(y_\mu, y_\sigma)$$

$$\log p_\theta(\boldsymbol{x}|\boldsymbol{z}) = -\log y_\sigma - \log\sqrt{2\pi} - \frac{1}{2y_\sigma^2}(\boldsymbol{x} - y_\mu)^2$$

$\log p_\theta(\boldsymbol{x}|\boldsymbol{z}) \uparrow$ $(\boldsymbol{x} - y_\mu) \downarrow$ $y_\sigma \downarrow$ Numerical Instability

Focus on some data variables

Only work well on some pixels

Set $y_\sigma = 1$

Becomes MSE ☺

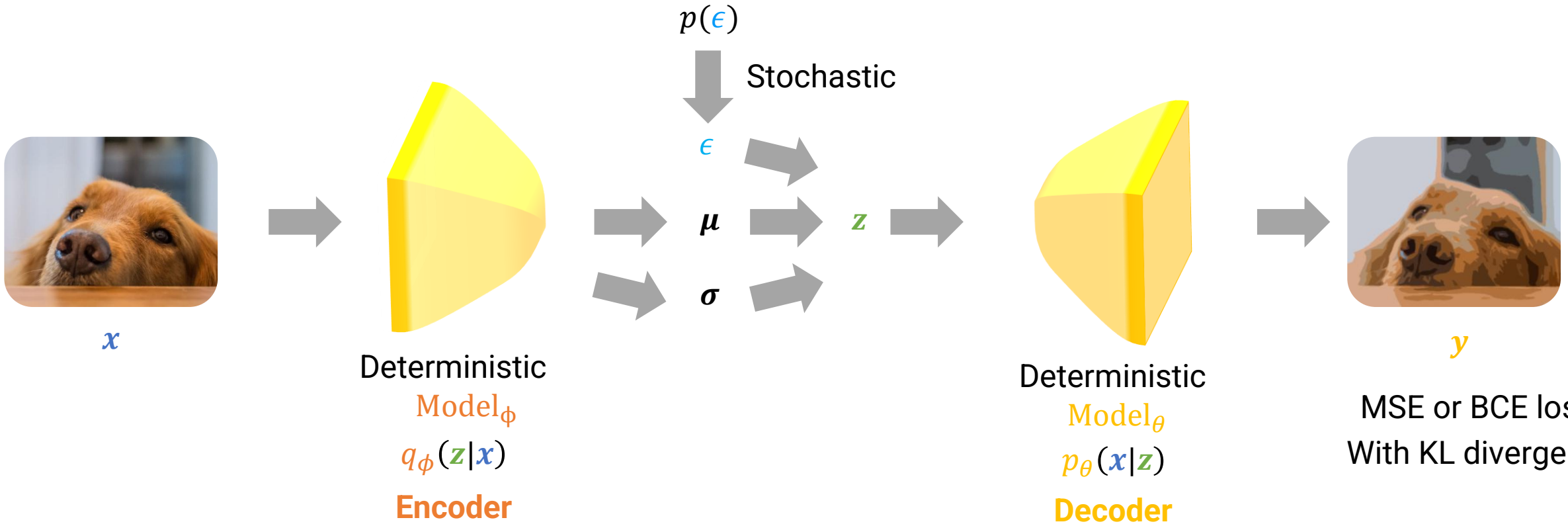
$$\log p_\theta(\boldsymbol{x}|\boldsymbol{z}) = \text{Bernoulli}(y)$$

$$\text{Bernoulli}(x; p) = p^x(1-p)^{1-x}; x \in \{0,1\}$$

$$\log p_\theta(\boldsymbol{x}|\boldsymbol{z}) = \sum_{i=1}^{D} \boldsymbol{x}_i \log \boldsymbol{y}_i + (1 - \boldsymbol{x}_i)\log(1 - \boldsymbol{y}_i)$$

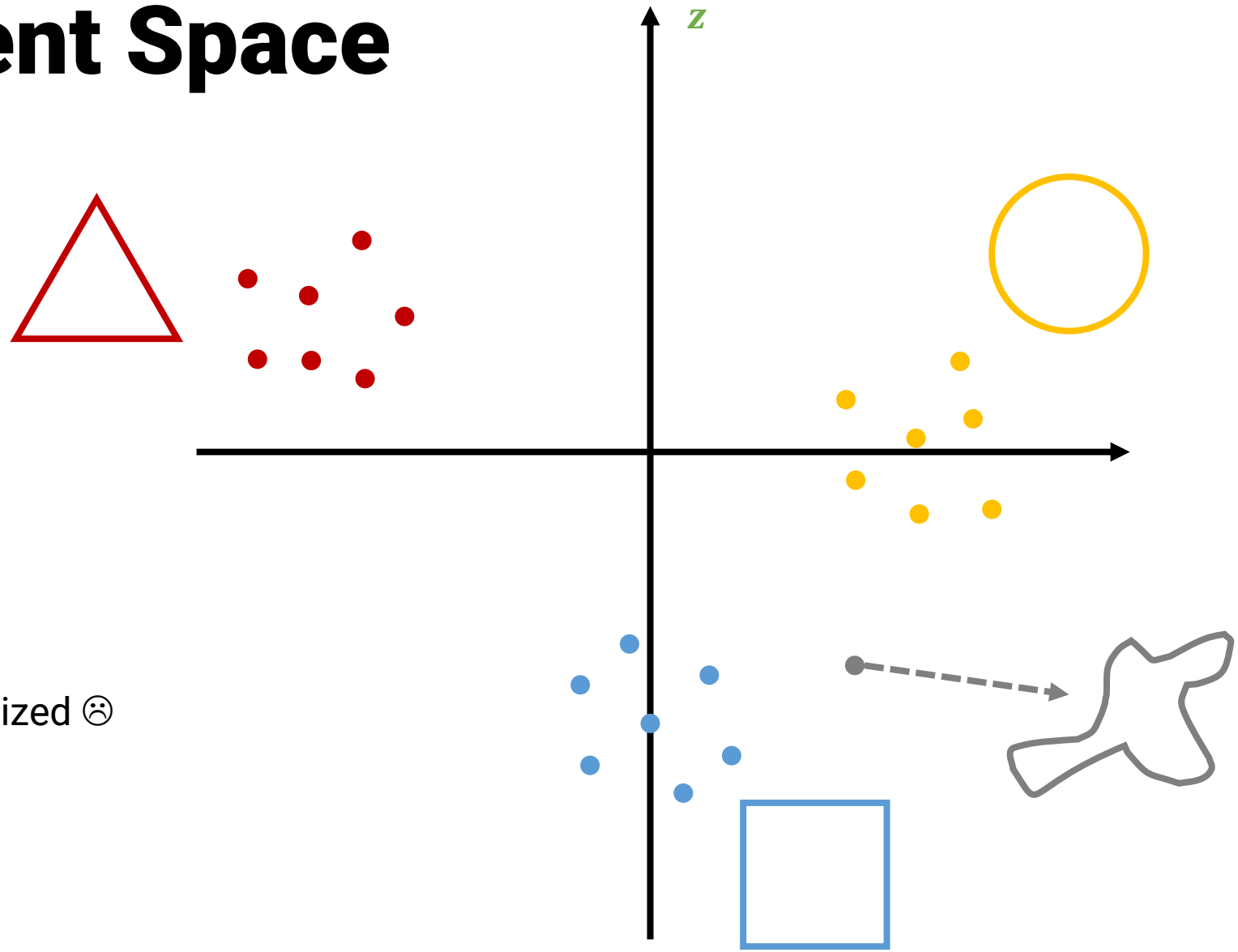
Becomes BCE (Binary Cross Entropy) ☺

Variational Auto Encoder



Kingma, Diederik P. "Auto-encoding variational bayes." arXiv preprint arXiv:1312.6114 (2013).

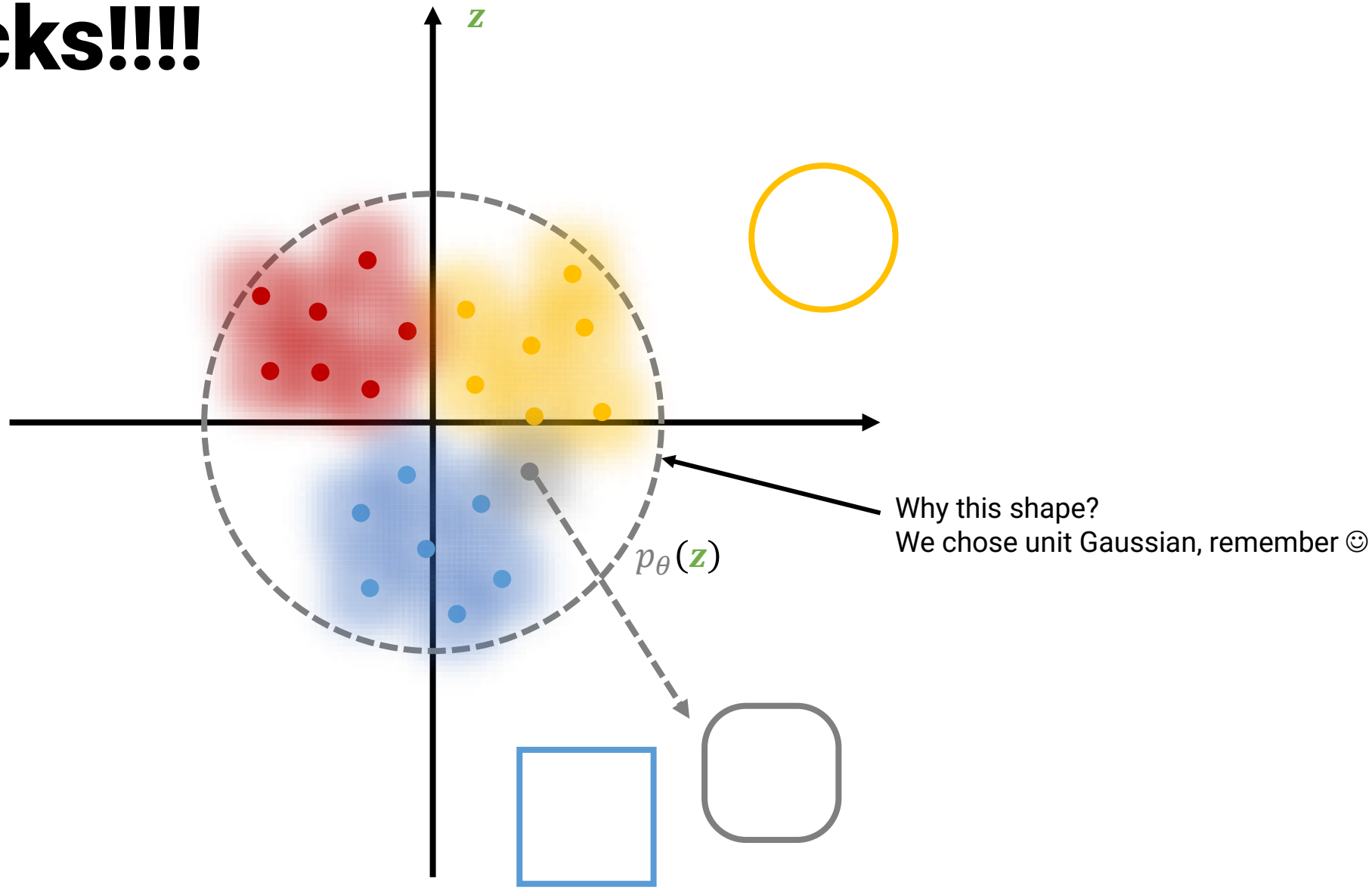
Latent Space



Not organized ☹

VAE Rocks!!!!

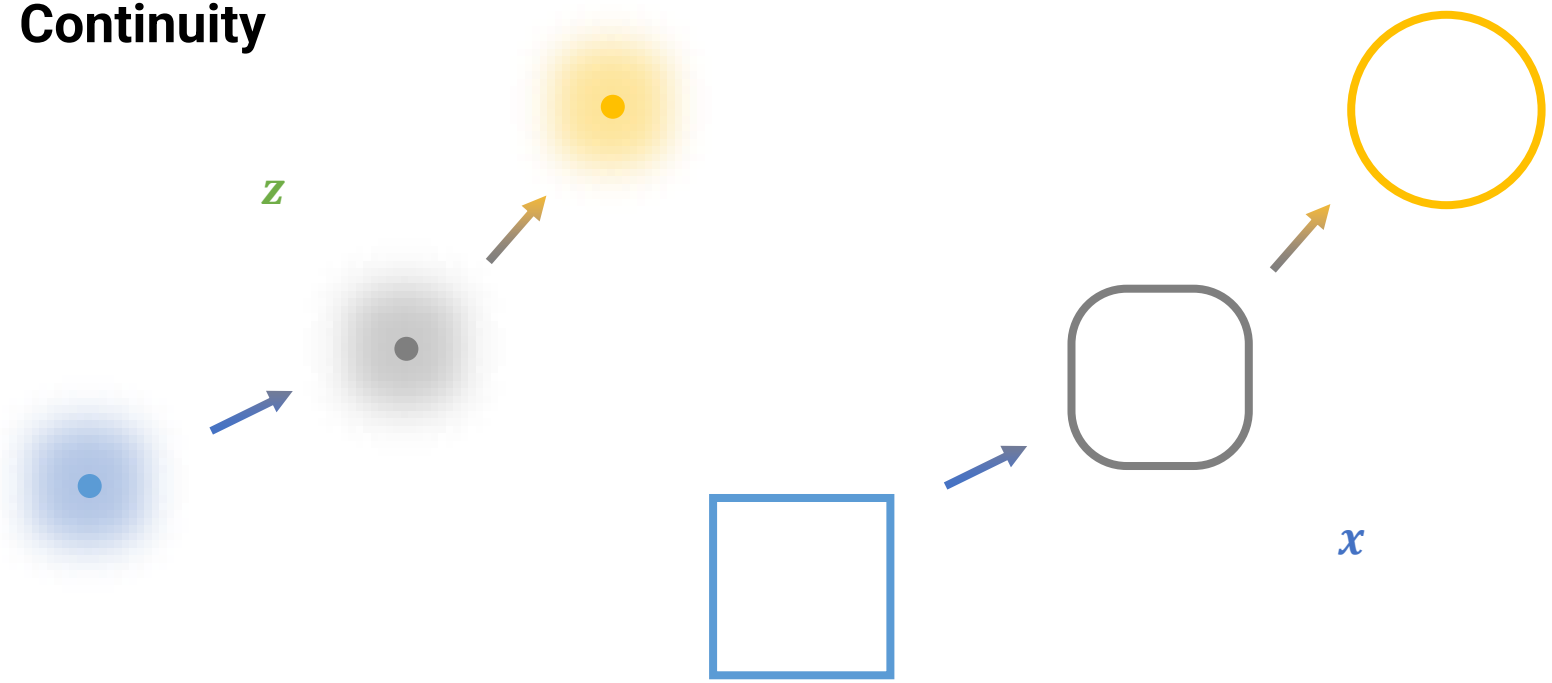
$D_{KL}(q_\phi(z|x)\|p_\theta(z))$



Well organized ☺

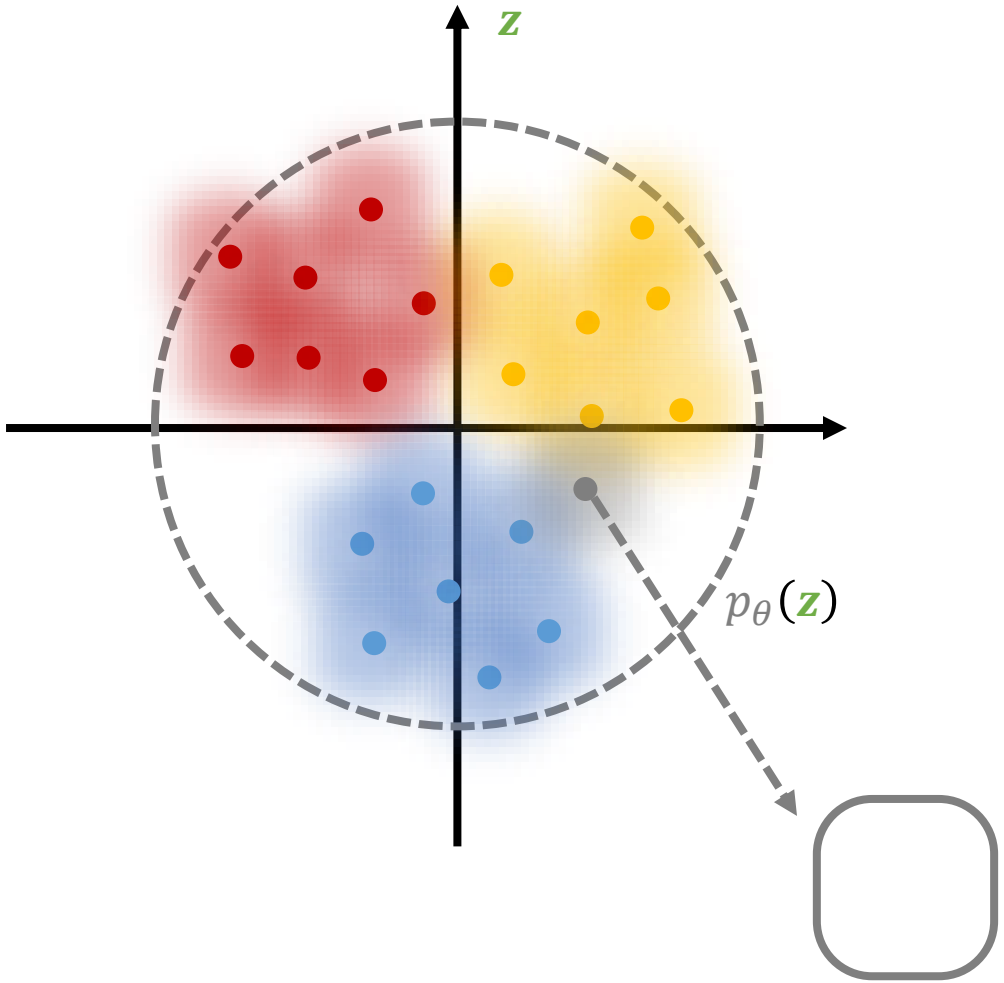
Awesome Latent Space!

Continuity



Awesome Latent Space!

Completeness

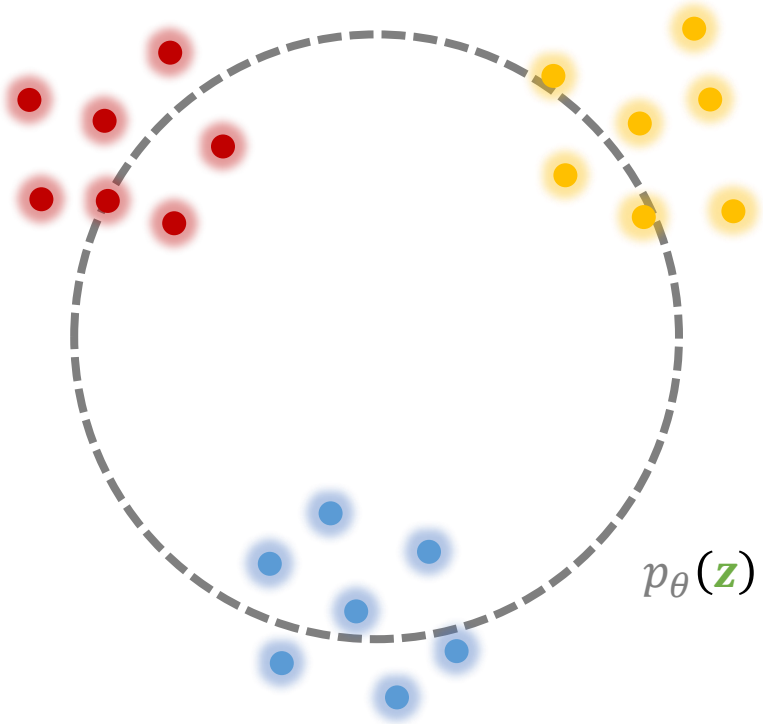


Beta VAE

VAE $\mathcal{L} = \mathcal{L}_{\text{recons}} + \mathcal{L}_{\text{KL}}$

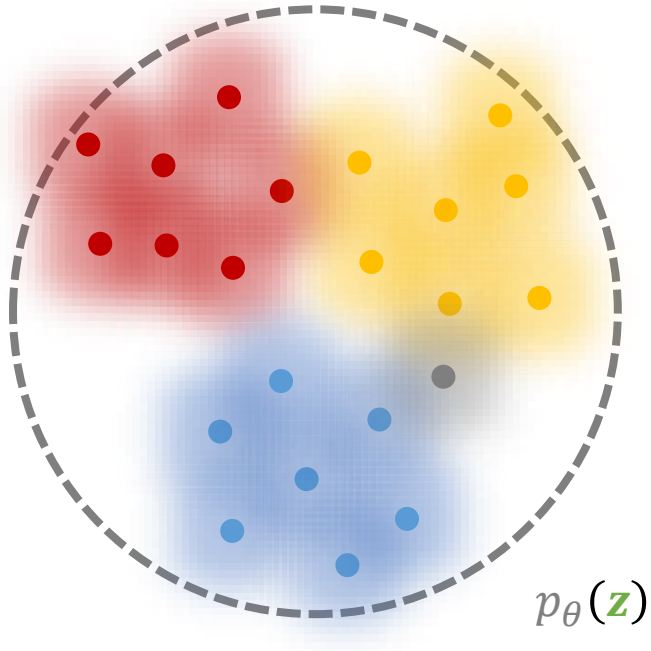
Weighted Loss $\mathcal{L} = \alpha\mathcal{L}_{\text{recons}} + \beta\mathcal{L}_{\text{KL}}$

$\alpha \gg \beta$



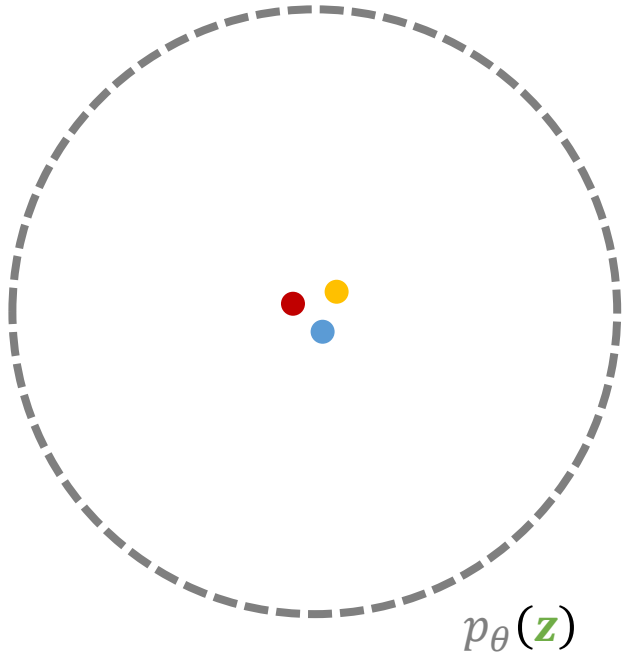
Discontinuity like Autoencoder

Doesn't mean $\beta = \alpha$



Balanced

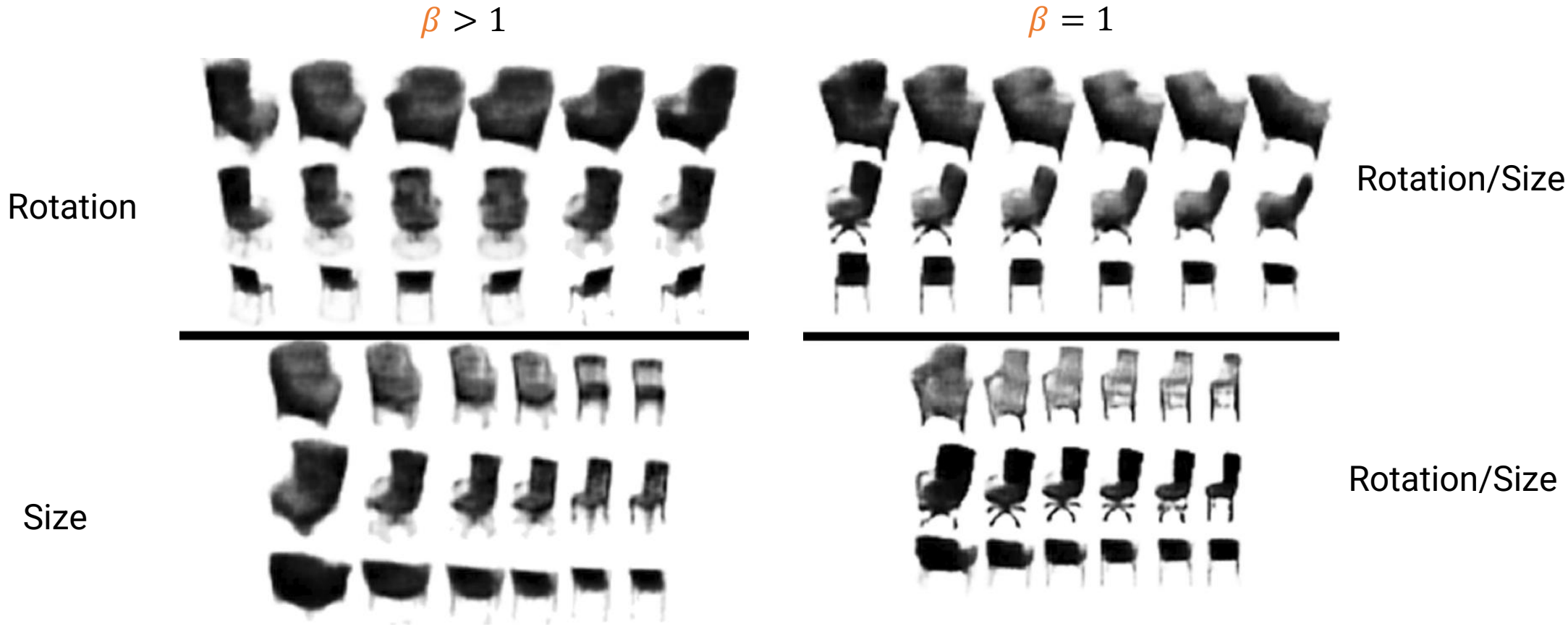
$\beta \gg \alpha$



No reconstructive capacity

Beta VAE

$\mathcal{L} = \mathcal{L}_{\text{recons}} + \beta\mathcal{L}_{\text{KL}}$



VAE Recap

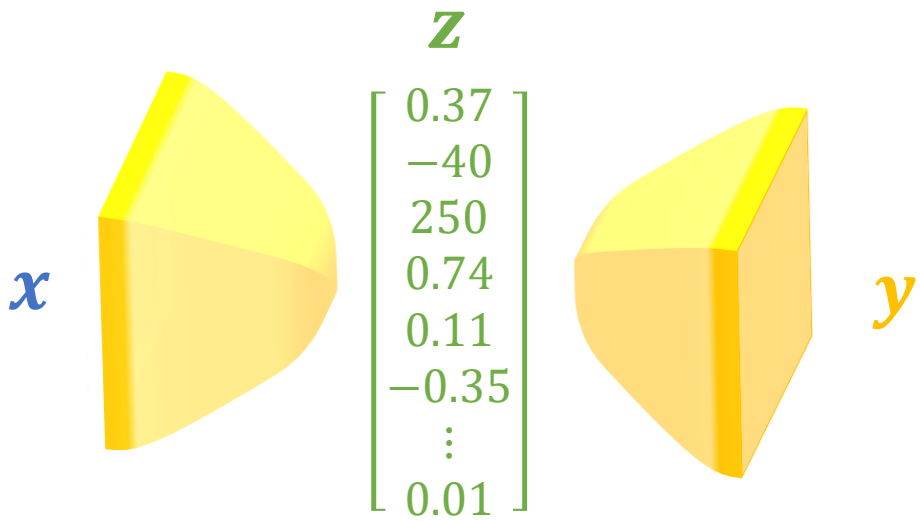
Latent Space Regularization
Distribution (KL) Loss

$$\mathcal{L}(\theta, \phi, \boldsymbol{x}, \boldsymbol{z}) = \underbrace{-\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})]}_{\text{Reconstruction Loss}} + \overbrace{D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \| p_\theta(\boldsymbol{z}))}$$

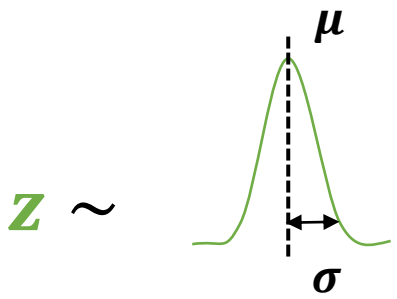
Reconstruction Loss
l_2 Loss

VAE Recap

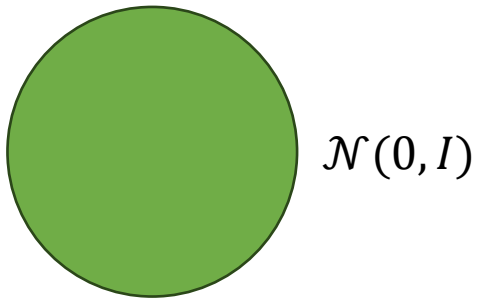
1. Continuous Latent Space



2. Gaussian $q_\phi(z|x)$

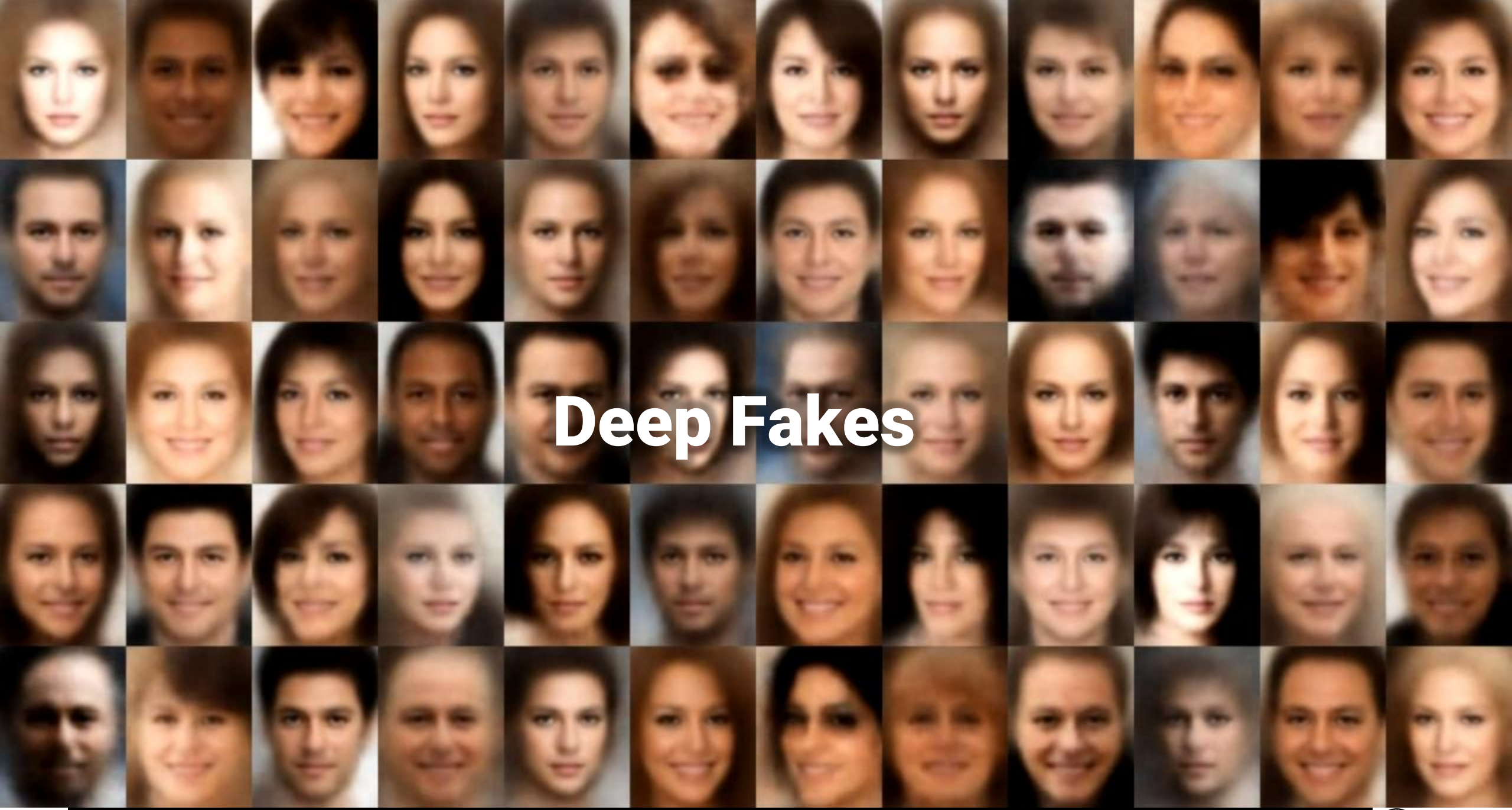


3. Pre-defined $p_\theta(z)$



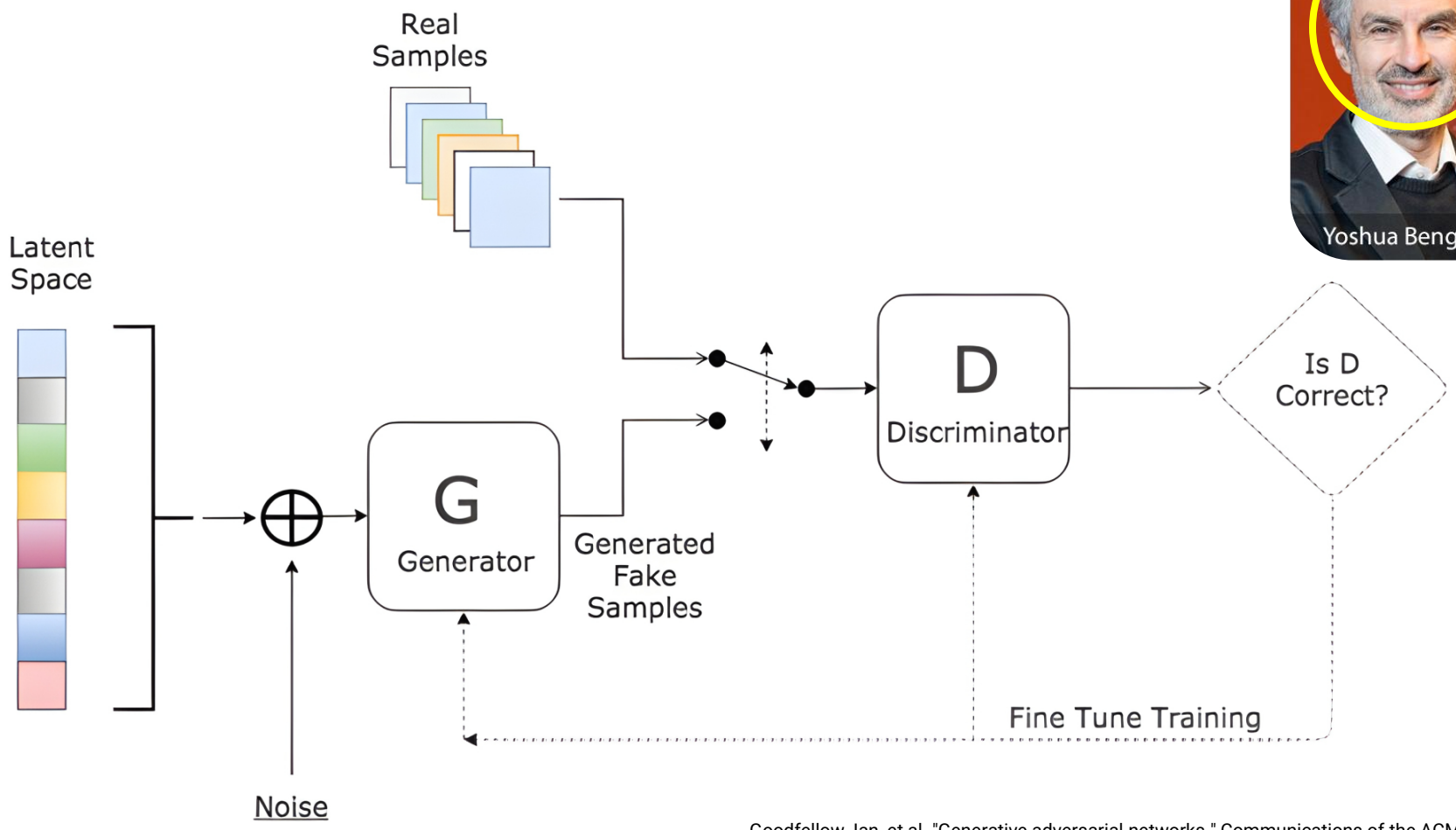
4. Loss function

$$\mathcal{L}(\theta, \phi, x, z) = -\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] + D_{KL}(q_\phi(z|x) \| p_\theta(z))$$


Deep Fakes

Generative Adversarial Networks

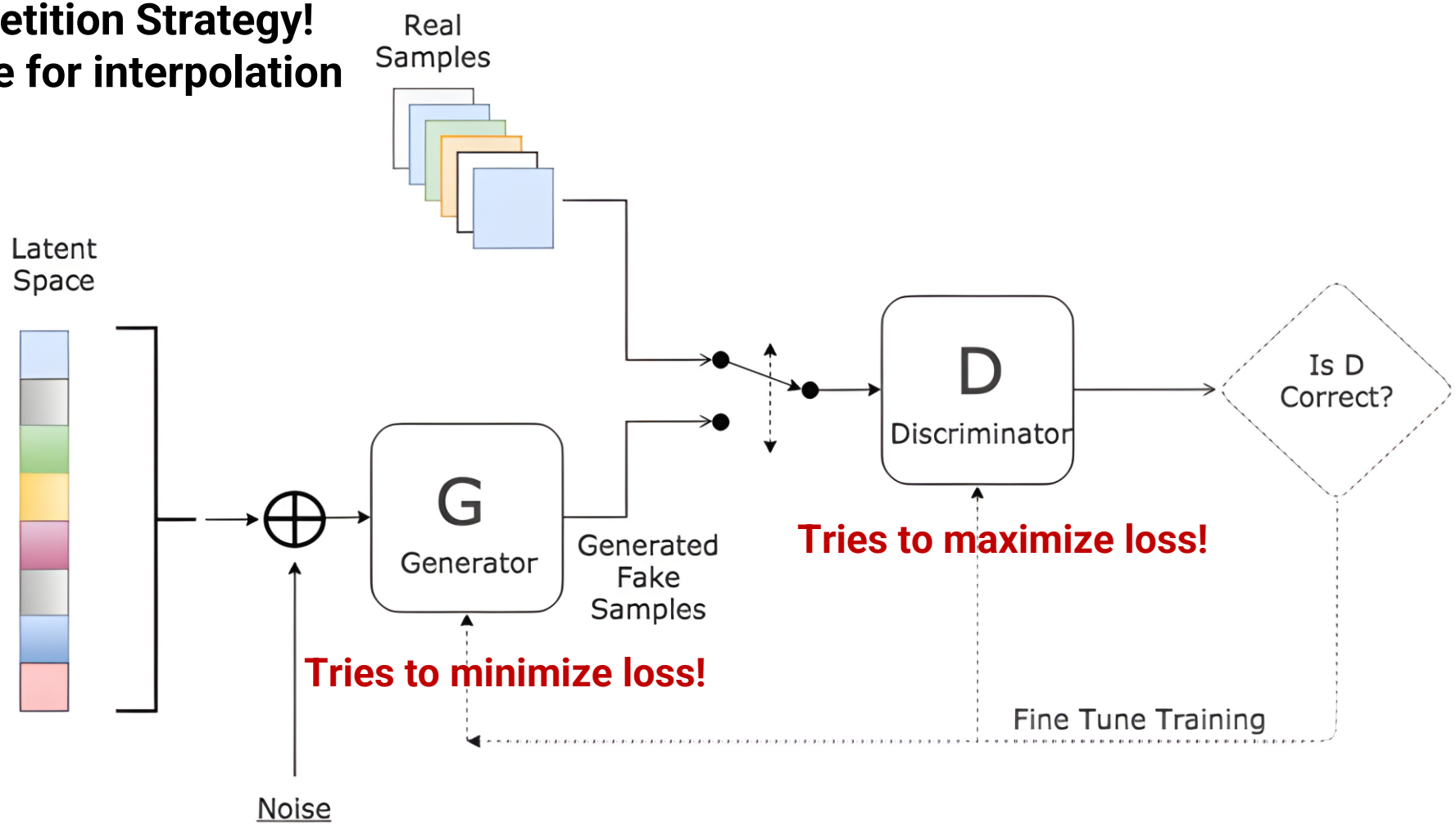
Let's Not Model Explicit Density Function



Goodfellow, Ian, et al. "Generative adversarial networks." Communications of the ACM 63.11 (2020): 139-144. CVPR 2018 Tutorial on GANs (google.com)

Generative Adversarial Networks

Use A Competition Strategy!
Latent Space for interpolation



Loss Function

Two-player min-max game

Discriminator predicts "realness" of data

Gradient Descent on G

Likelihood of generated data

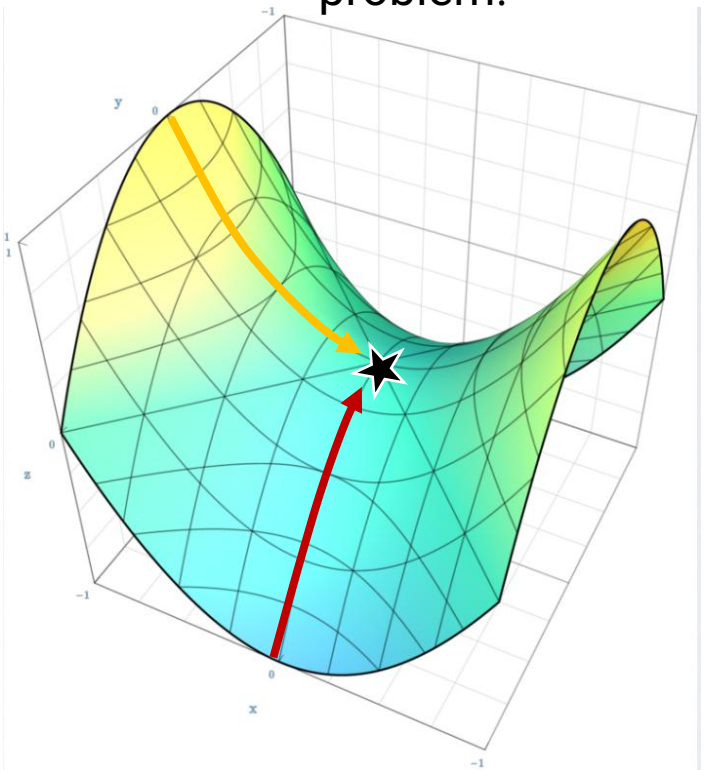
$$\underset{\theta_g}{\operatorname{argmin}}\ \underset{\theta_d}{\operatorname{argmax}}\ \mathbb{E}_{x\sim p_{\text{data}}}\left[\log D_{\theta_d}(x)\right] + \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1 - D_{\theta_d}(G_{\theta_g}(z))\right)\right]$$

Likelihood of true data

Gradient Ascent on D

Generator produces "real-looking" data

Saddle point problem!



GAN Gremlins!

Inherently unstable to train ☹

Vanishing Gradients
D becomes “too good” faster and G gradients become 0

Non-Convergence
D and G oscillate without reaching equilibrium

Mode Collapse
G will only generate a small set of examples (low diversity)

Mode Dropping
G does not cover the data distribution

GAN Gremlins!

Vanishing Gradients

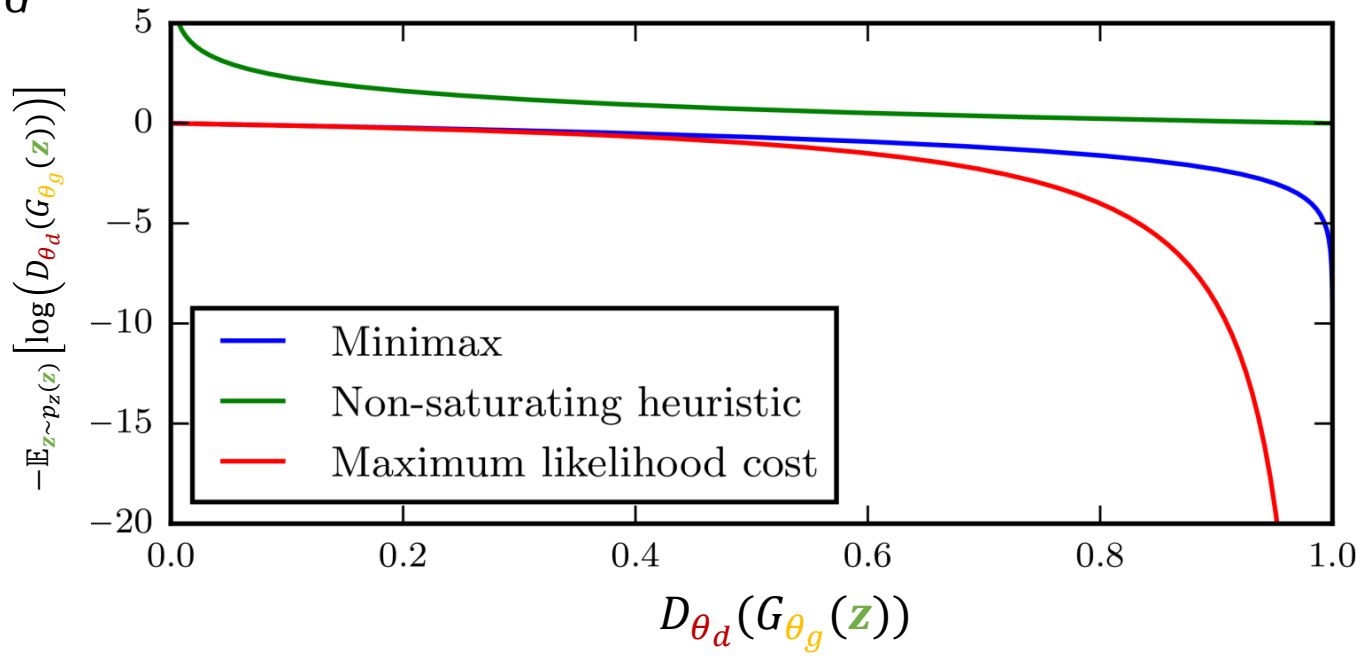
$$\underset{\theta_g}{\operatorname{argmin}} \underset{\theta_d}{\operatorname{argmax}} \mathbb{E}_{x \sim p_{\text{data}}} \left[\log D_{\theta_d}(x)\right] + \mathbb{E}_{z \sim p_z(z)} \left[\log\left(1 - D_{\theta_d}(G_{\theta_g}(z))\right)\right]$$

Use a non-saturating heuristic objective for G

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

Limit D capacity

Schedule learning between G and D



Generative Adversarial Networks (stanford.edu)

GAN Gremlins!

Non-Convergence

Generally, hacks with not much mathematical backing

Use soft and Noisy Labels

Use DCGAN and Hybrid Models

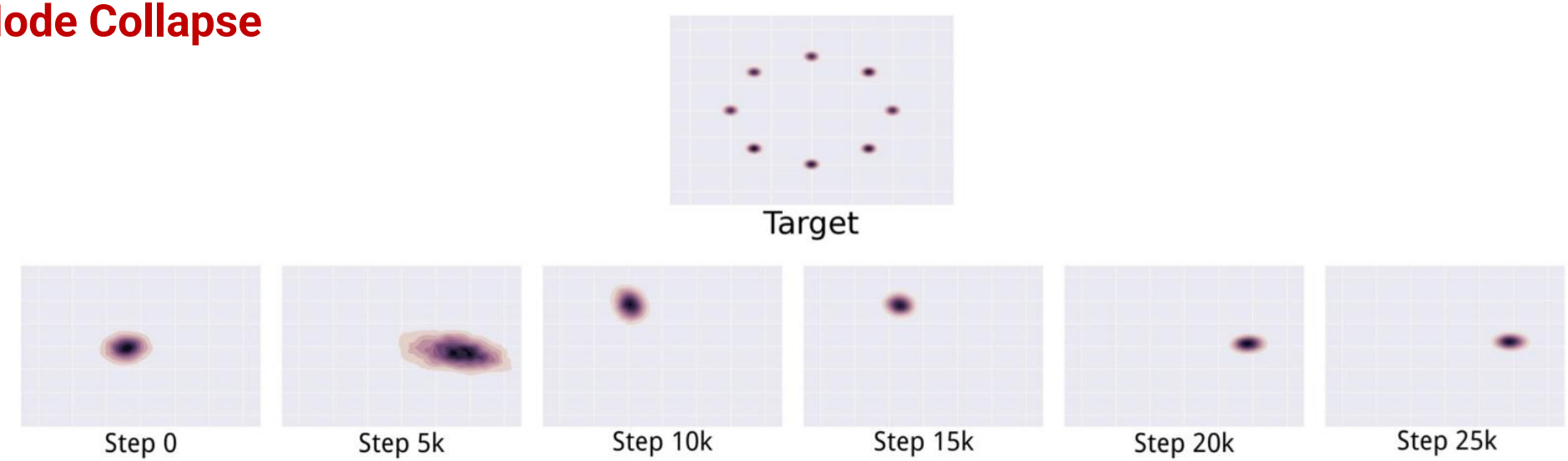
Use stability tricks from Reinforcement Learning

Use ADAM optimizer

soumith/ganhacks: starter from "How to Train a GAN?" at NIPS2016 (github.com)

GAN Gremlins!

Mode Collapse



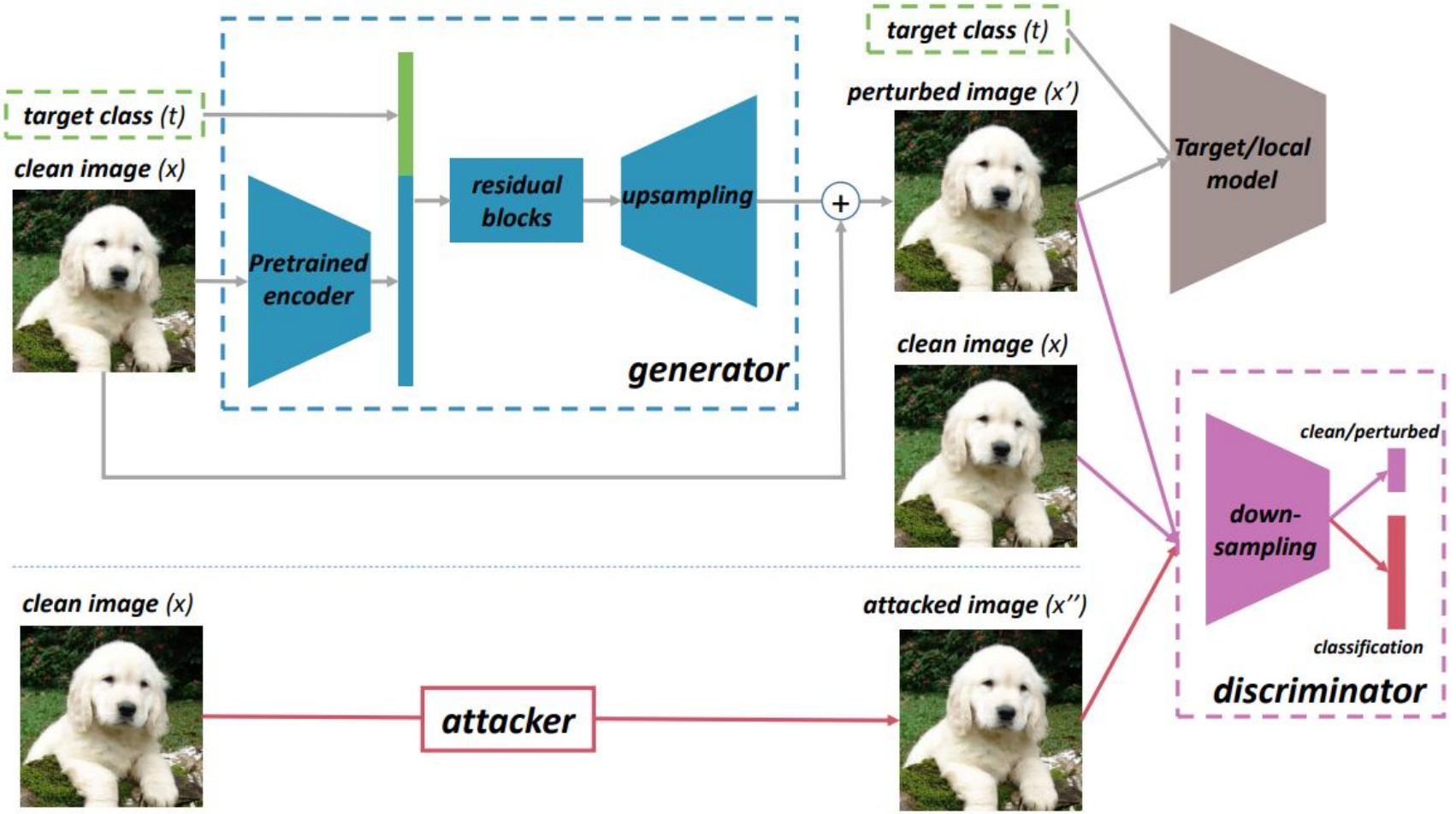
Use divergence measures like KL (Recall VAE) Check f-GAN

Use Integral Probability metrics like Earth Movers Distance Check Wasserstein GAN, Fisher GAN and Sobolev GAN

GANs Work Well….

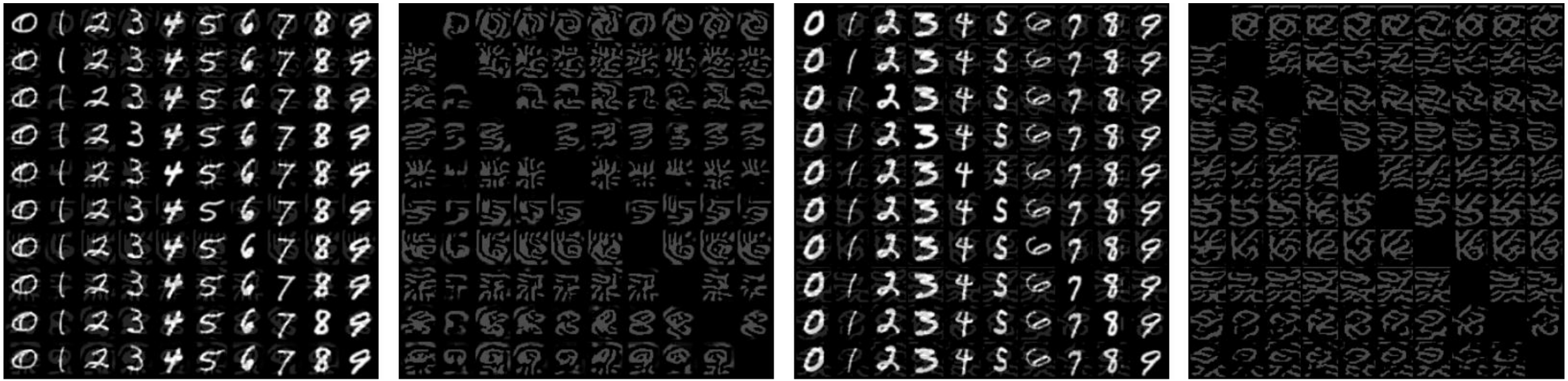
Well almost well….

AI-GAN



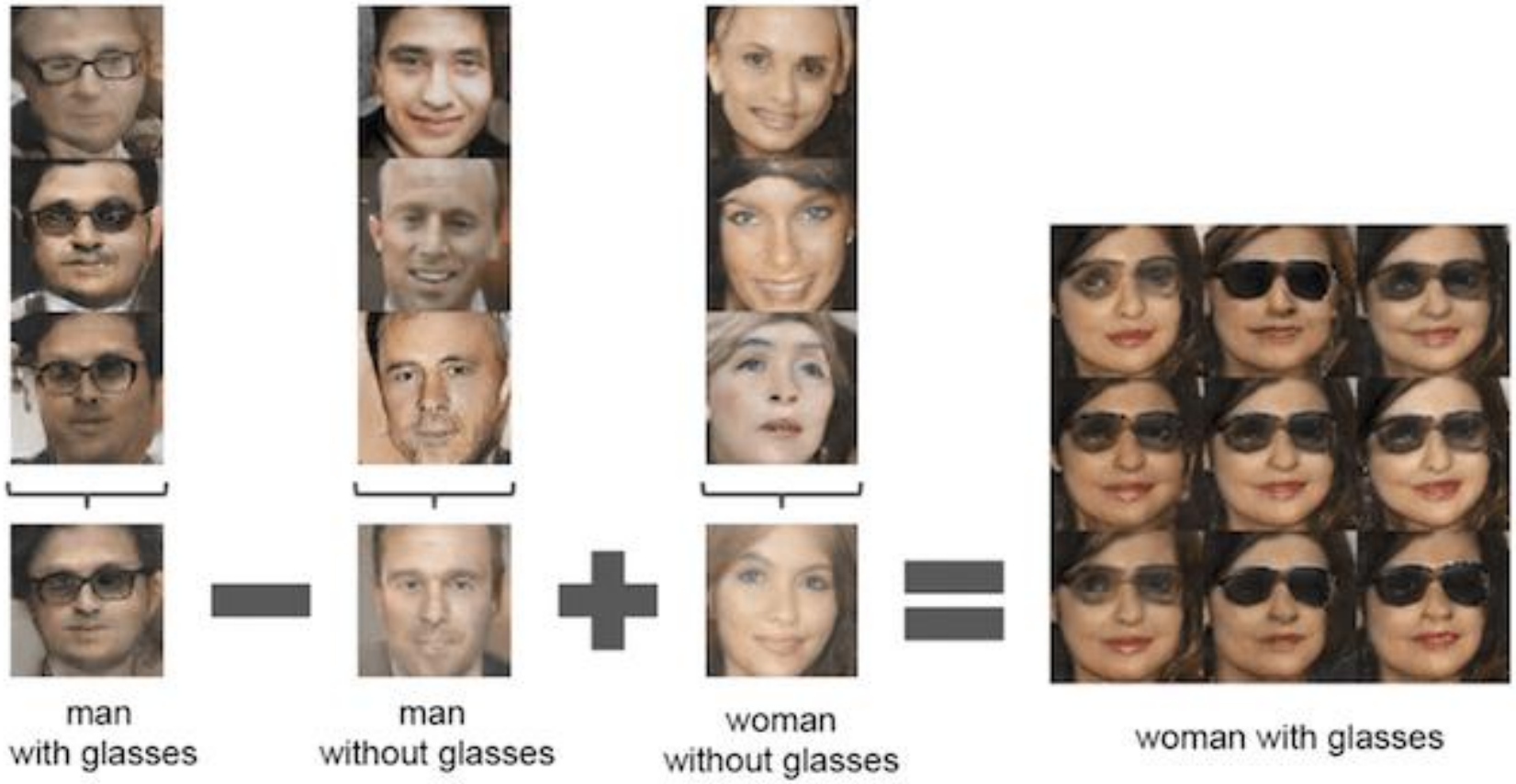
Bai, Tao, et al. "Ai-gan: Attack-inspired generation of adversarial examples." 2021 IEEE International Conference on Image Processing (ICIP). IEEE, 2021.

AI-GAN



Generative Models Can Do Cool Stuff!

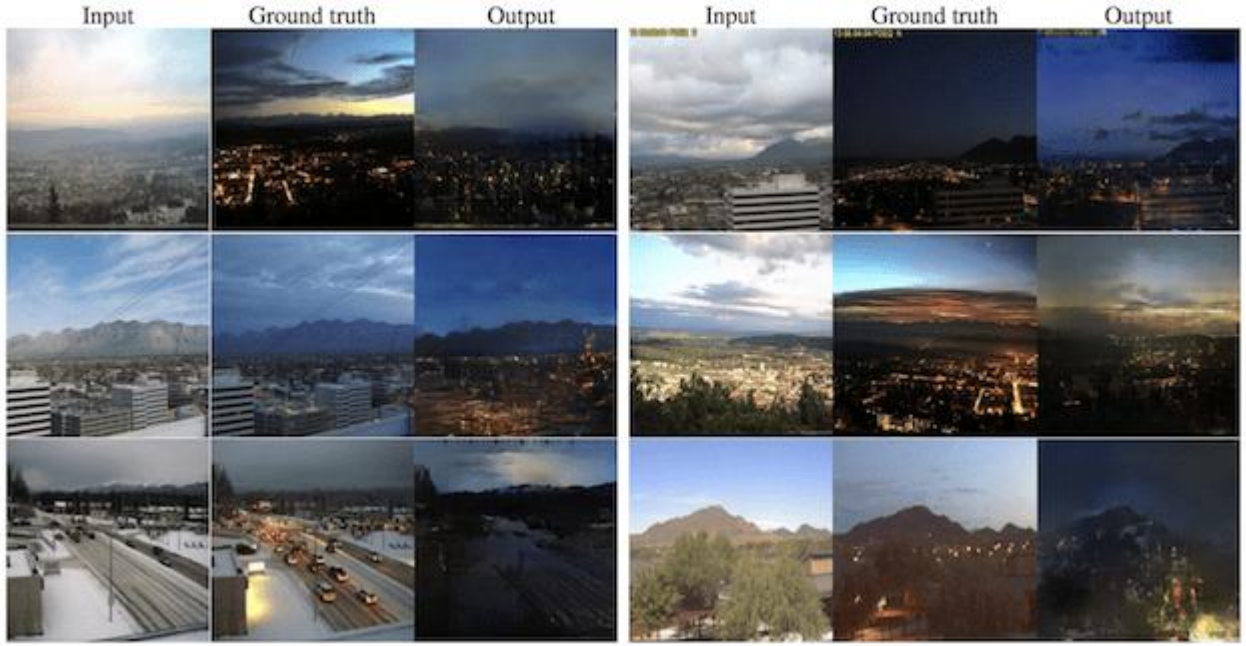
Generation with Latent Space Arithmetic



18 Impressive Applications of Generative Adversarial Networks (GANs) - MachineLearningMastery.com

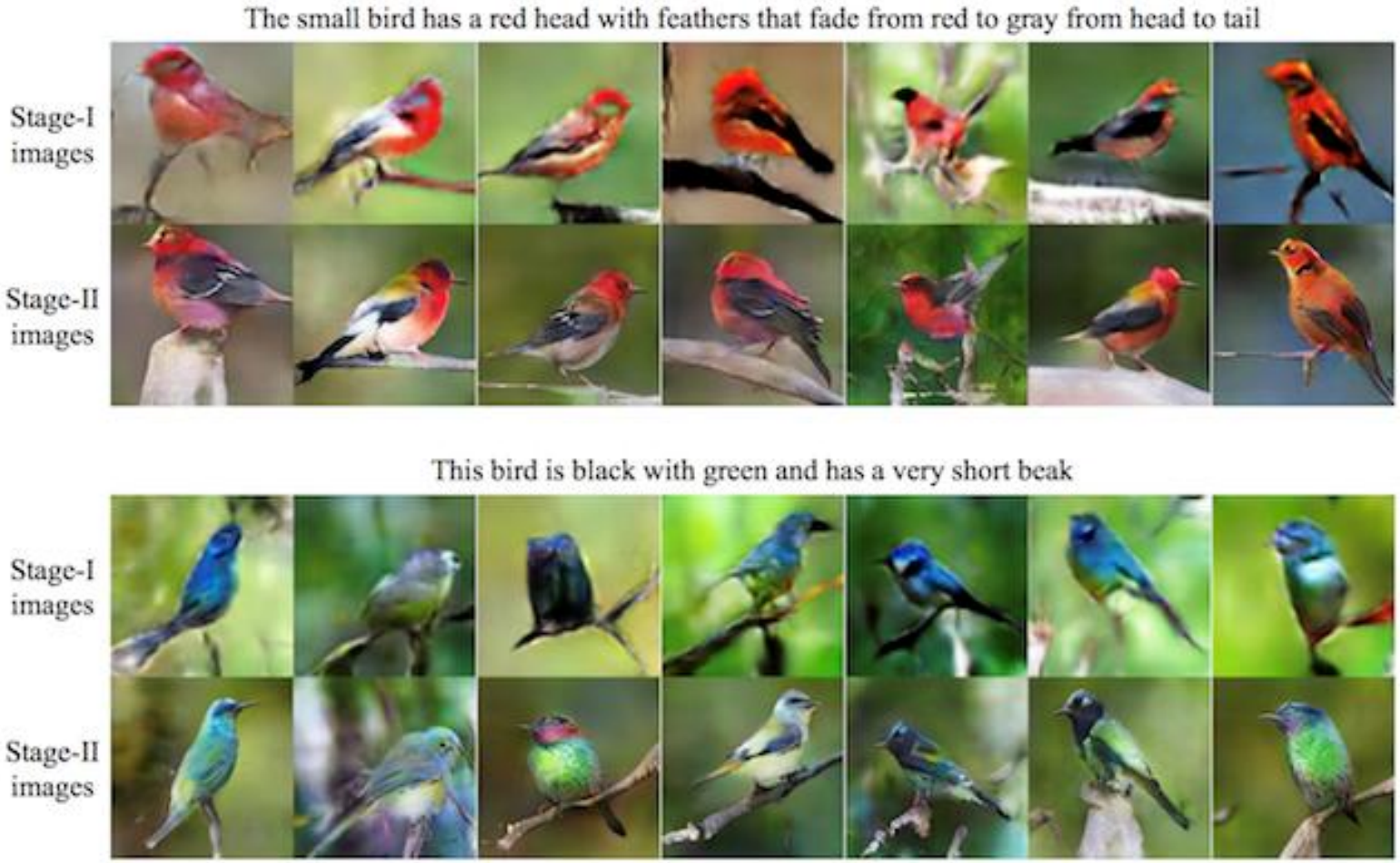
Generative Models Can Do Cool Stuff!

Image To Image Translation AKA pix2pix



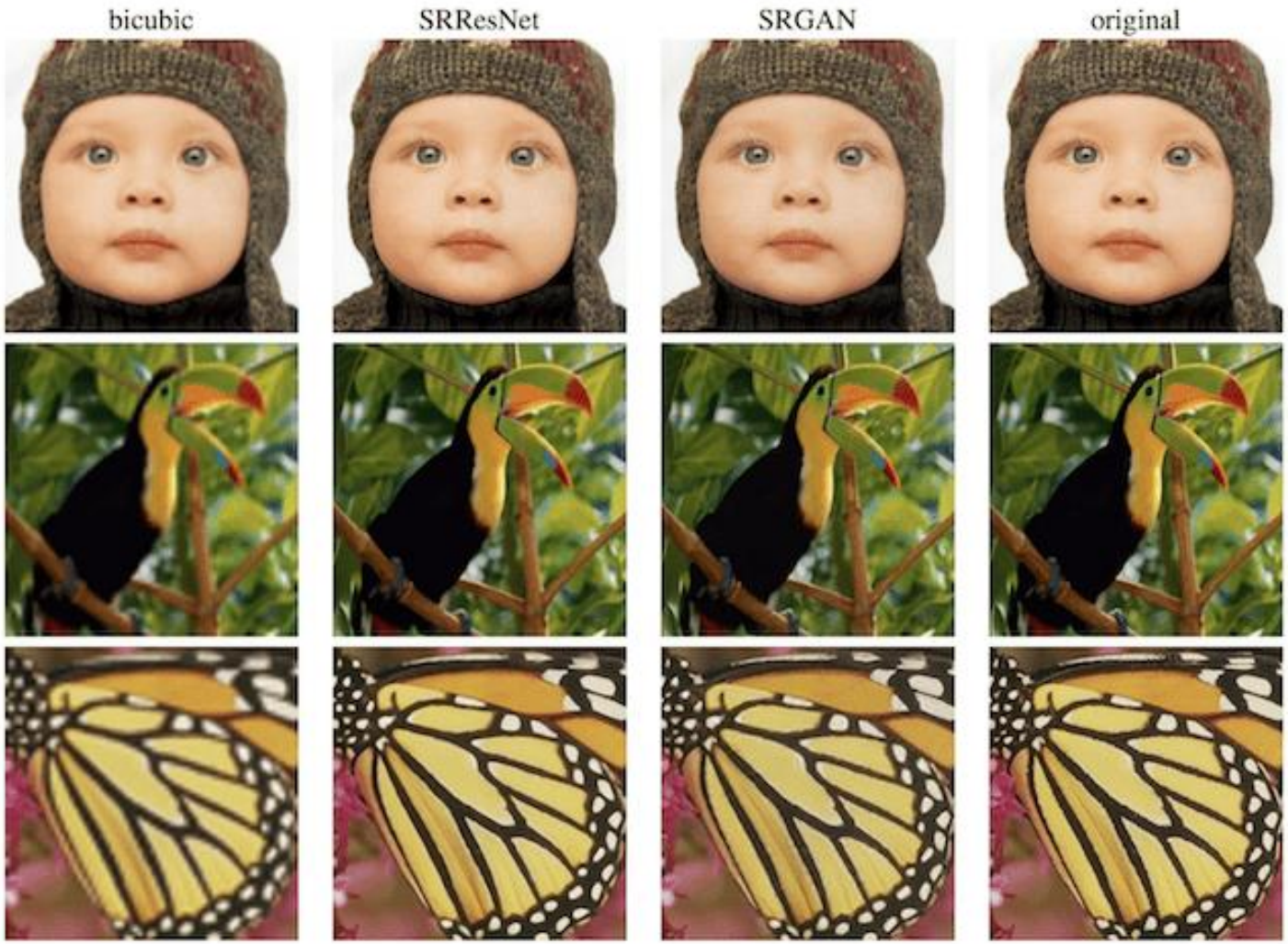
Generative Models Can Do Cool Stuff!

Text2Image



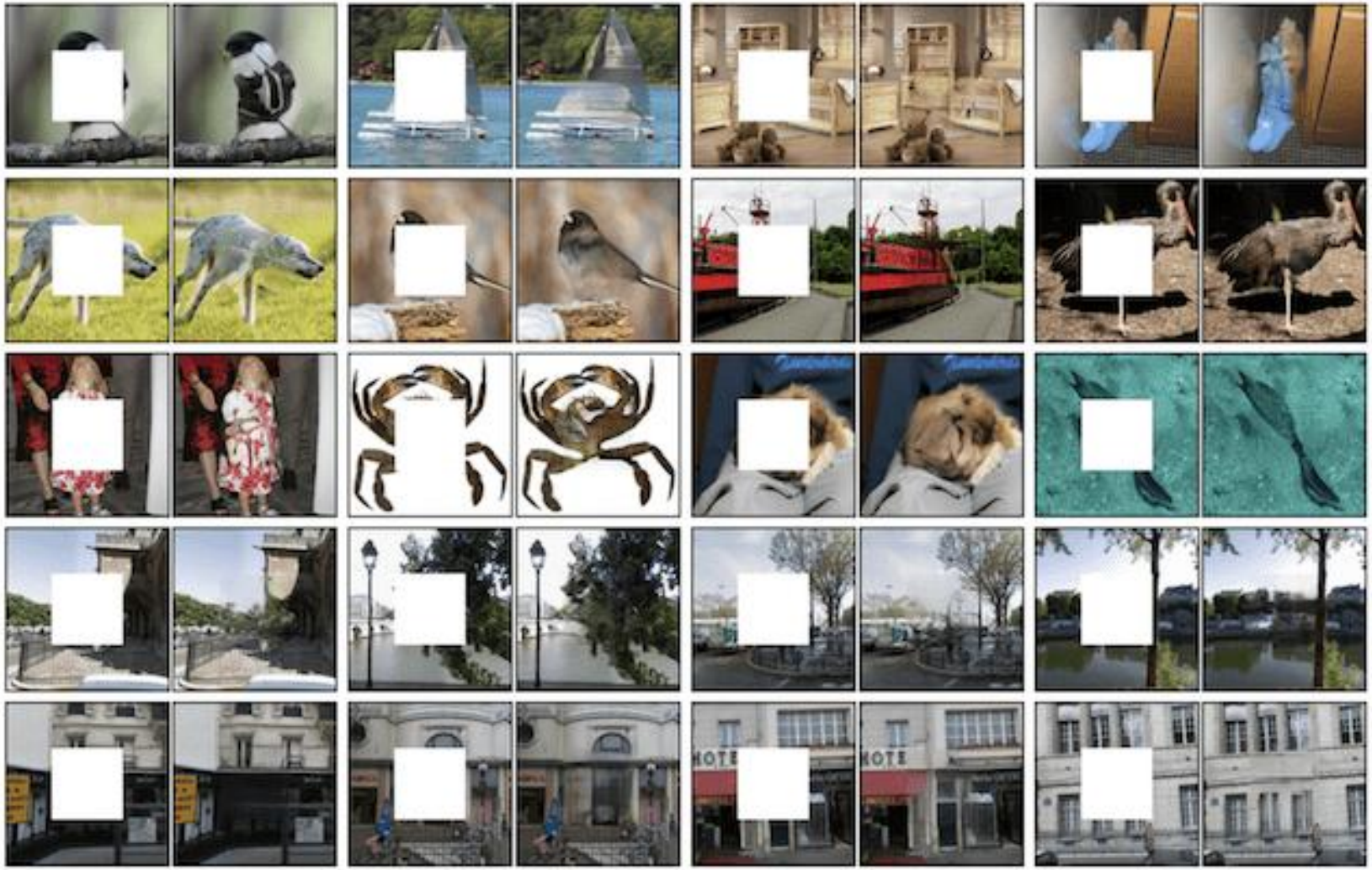
Generative Models Can Do Cool Stuff!

Super Resolution

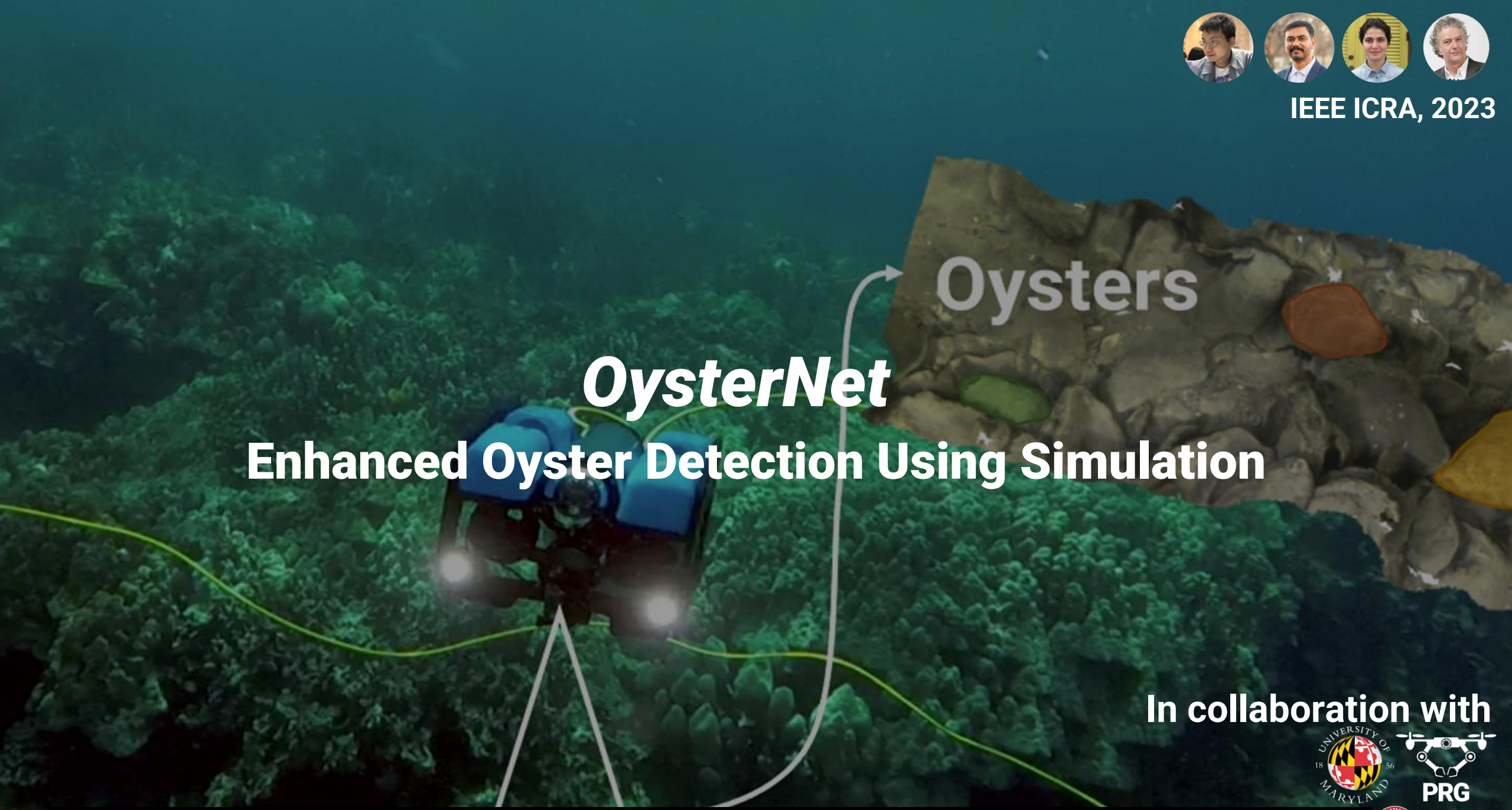


Generative Models Can Do Cool Stuff!

Inpainting




Generate A LOT Of Data



IEEE ICRA, 2023

OysterNet

Enhanced Oyster Detection Using Simulation

In collaboration with

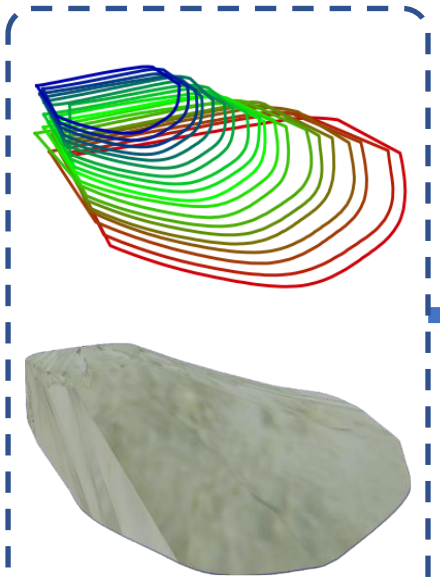
PRG

Real Oyster Data

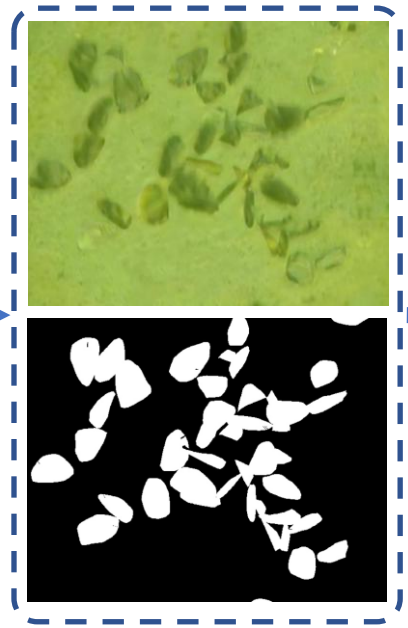
Oyster Modeling

Simulated Oyster Data

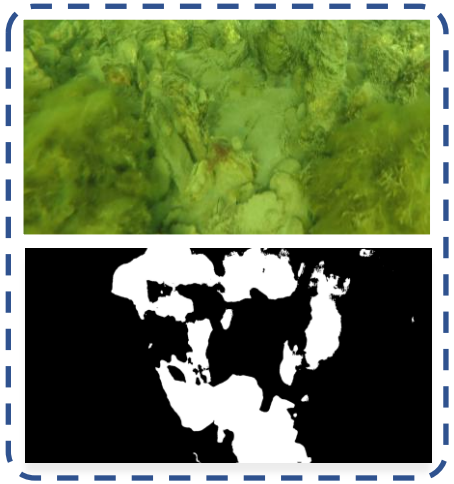
Synthetic Oyster Data



Sim to Real



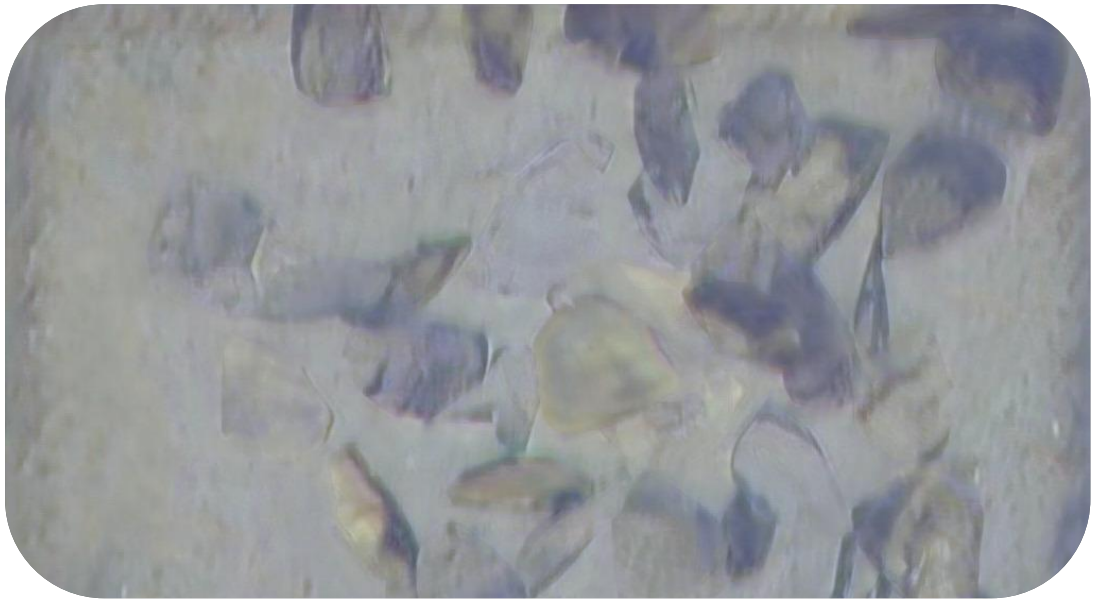
OysterNet

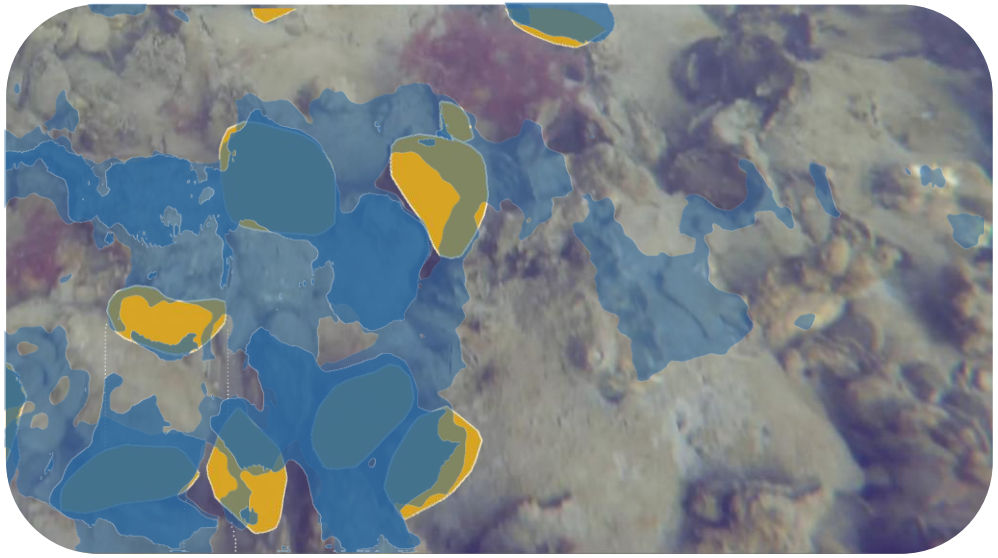


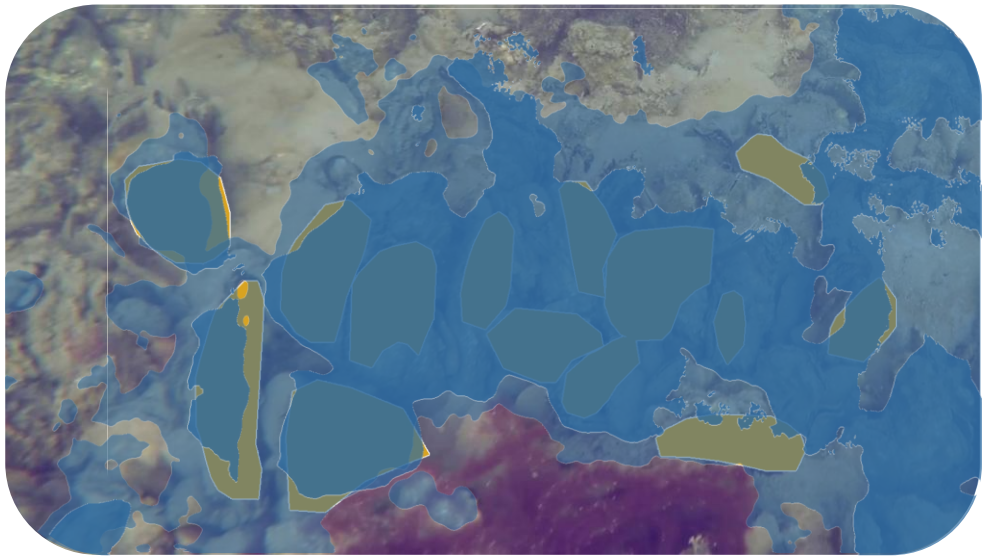
Oyster Detection

Photo-realistic image

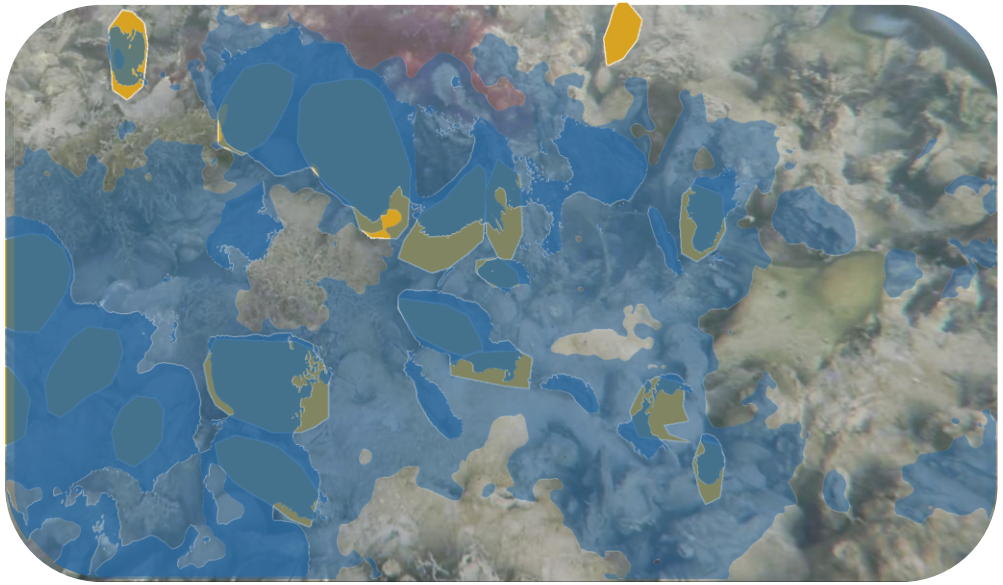
Simulation synthetic image

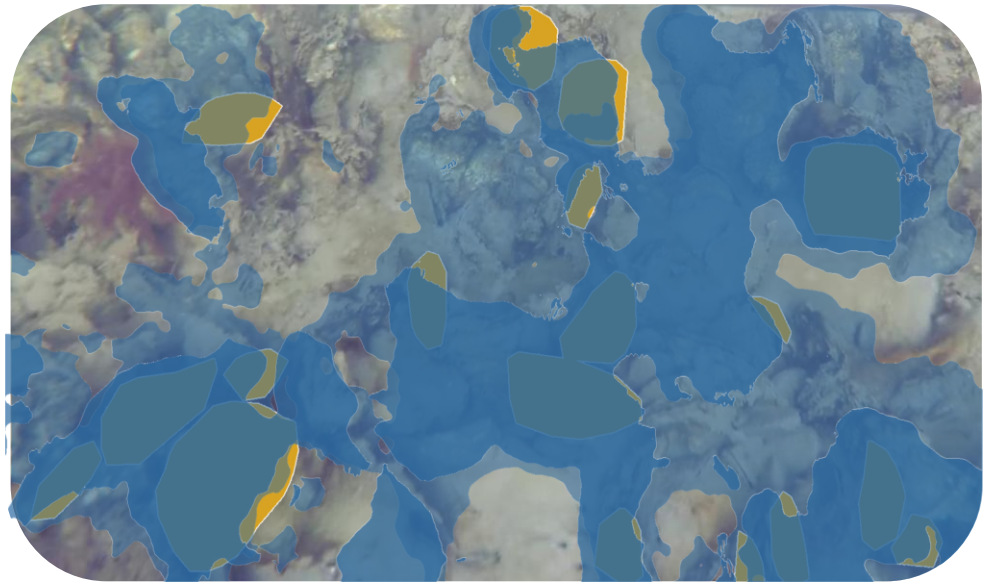






Detections with synthetic data





IEEE IROS, 2023

Olive the Above

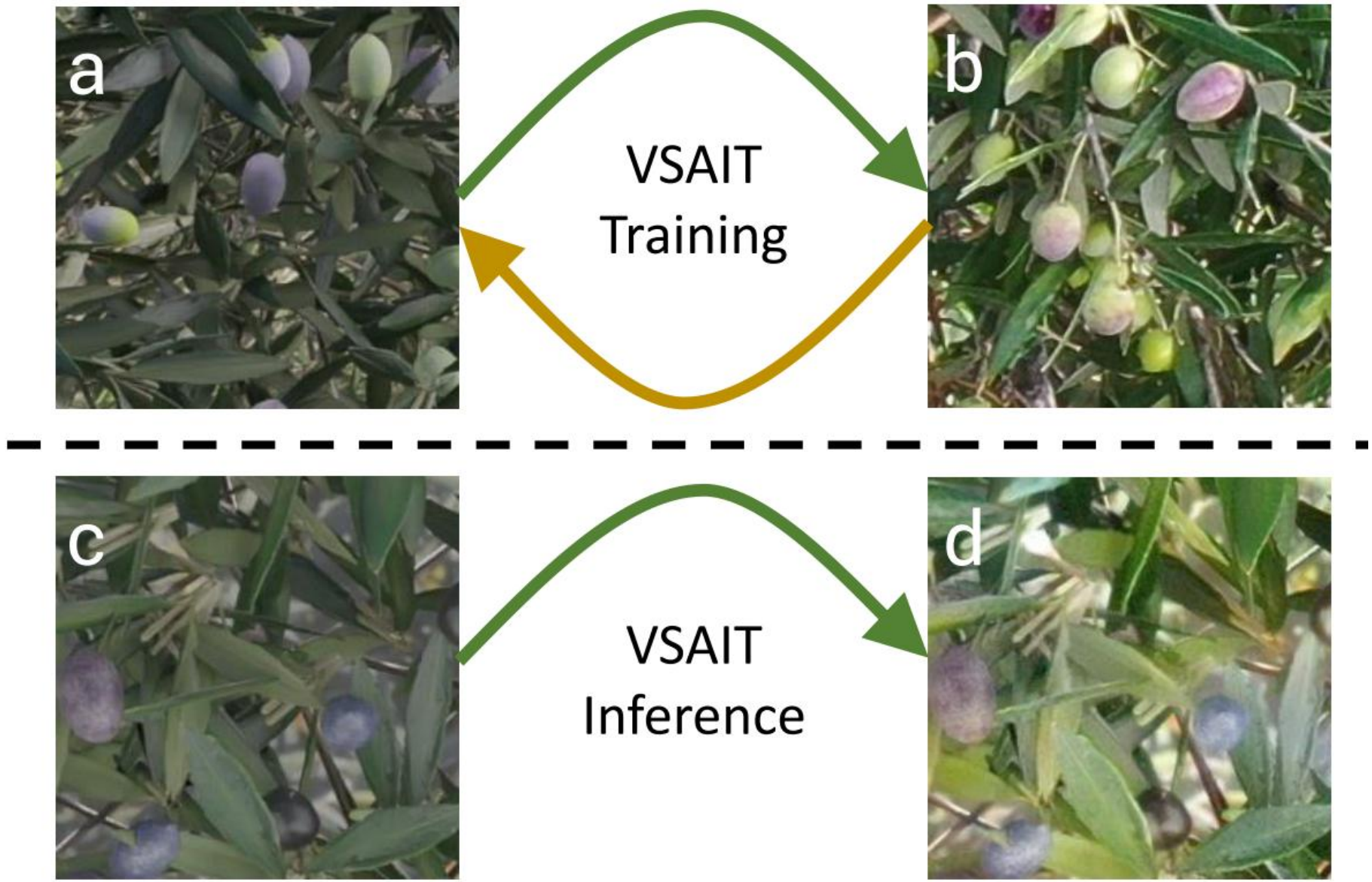
Detecting Olives with Synthetic or Real Data?

In collaboration with

TECHNICAL UNIVERSITY OF CRETE

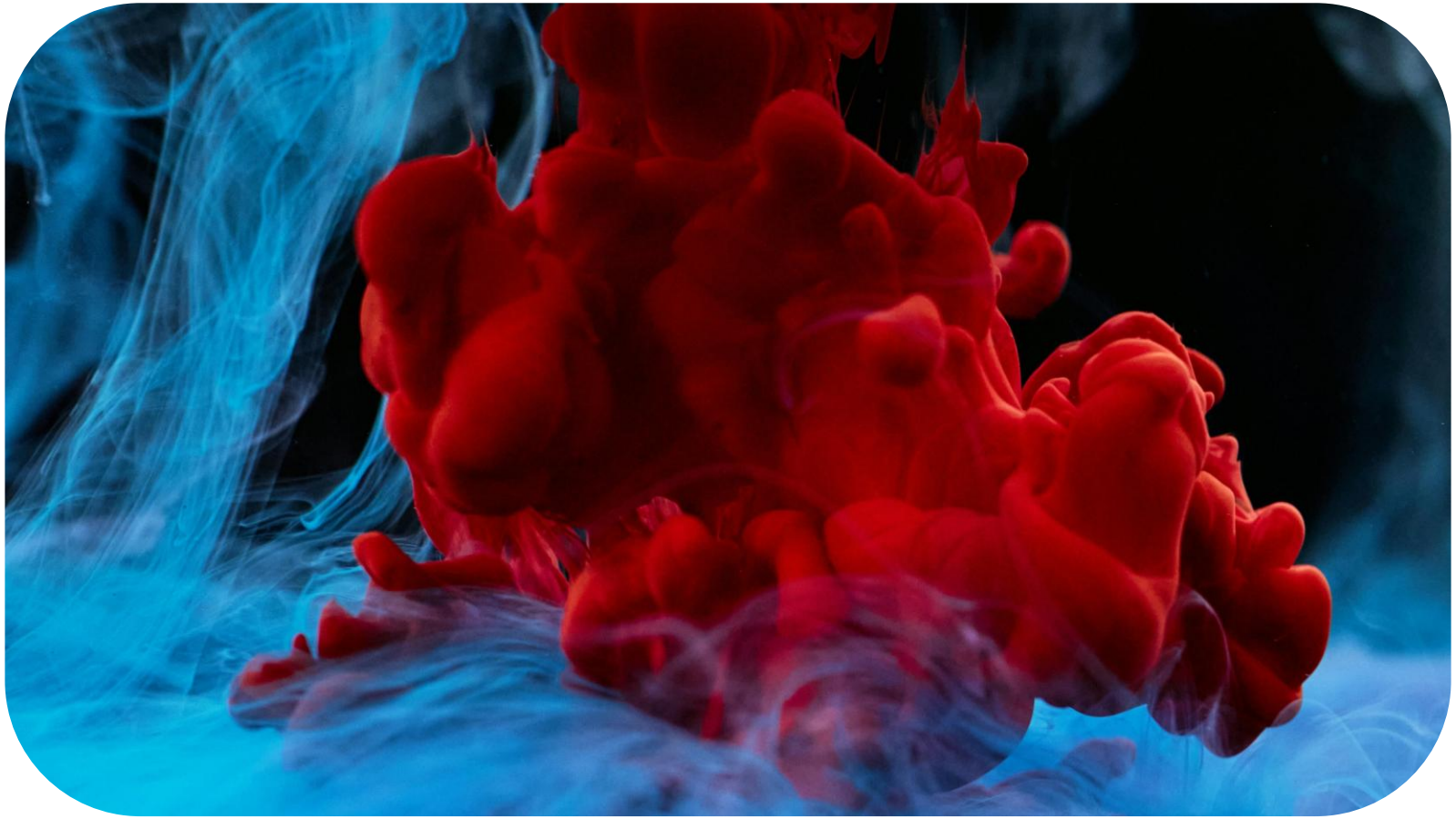
UNIVERSITY OF MARYLAND

PRG


a
b
VSAIT
Training
c
d
VSAIT
Inference


Better Deep Fakes
DEEP
FAKES

Next Class!



Advanced Generative Models: Diffusion Models