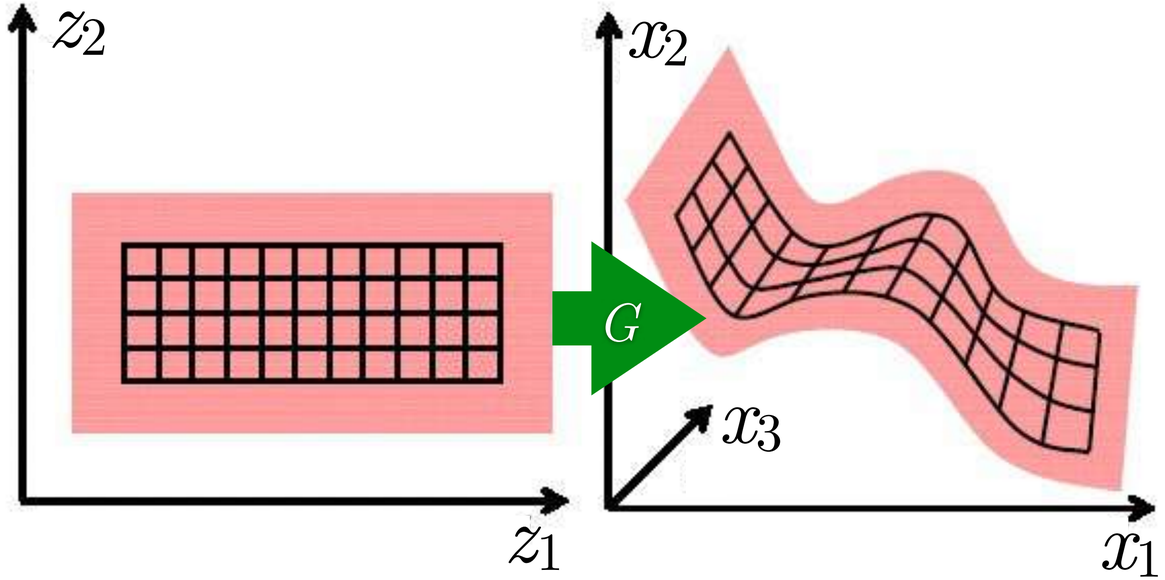
Generative Adversarial Networks

Aaron Courville
Université de Montréal

Some Slides are taken from Ian Goodfellow's NIPS 2016 Tutorial on GANs

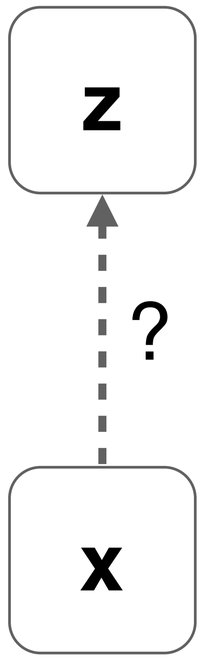
Another way to train a latent variable model

Latent variables

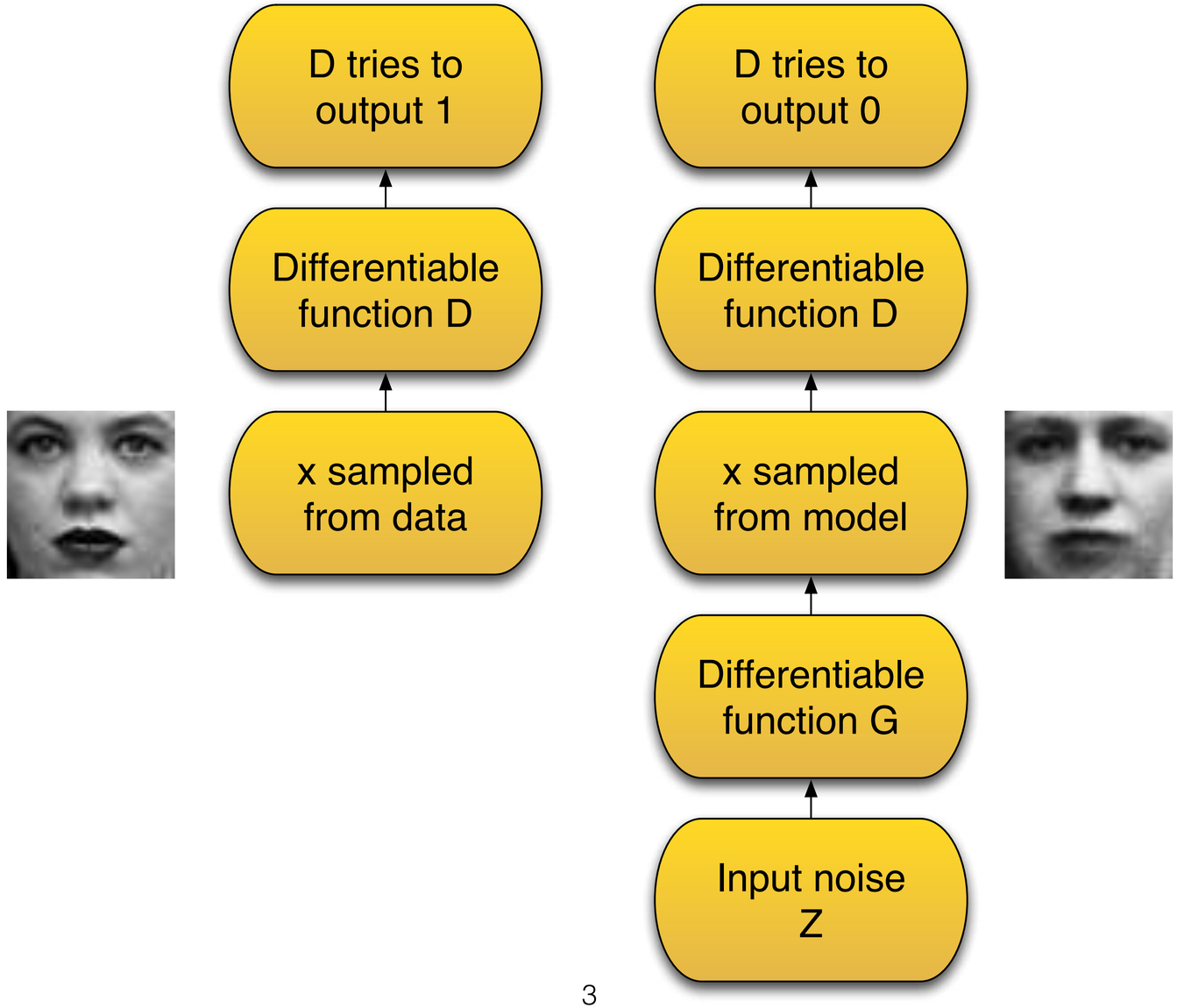


Observed variables

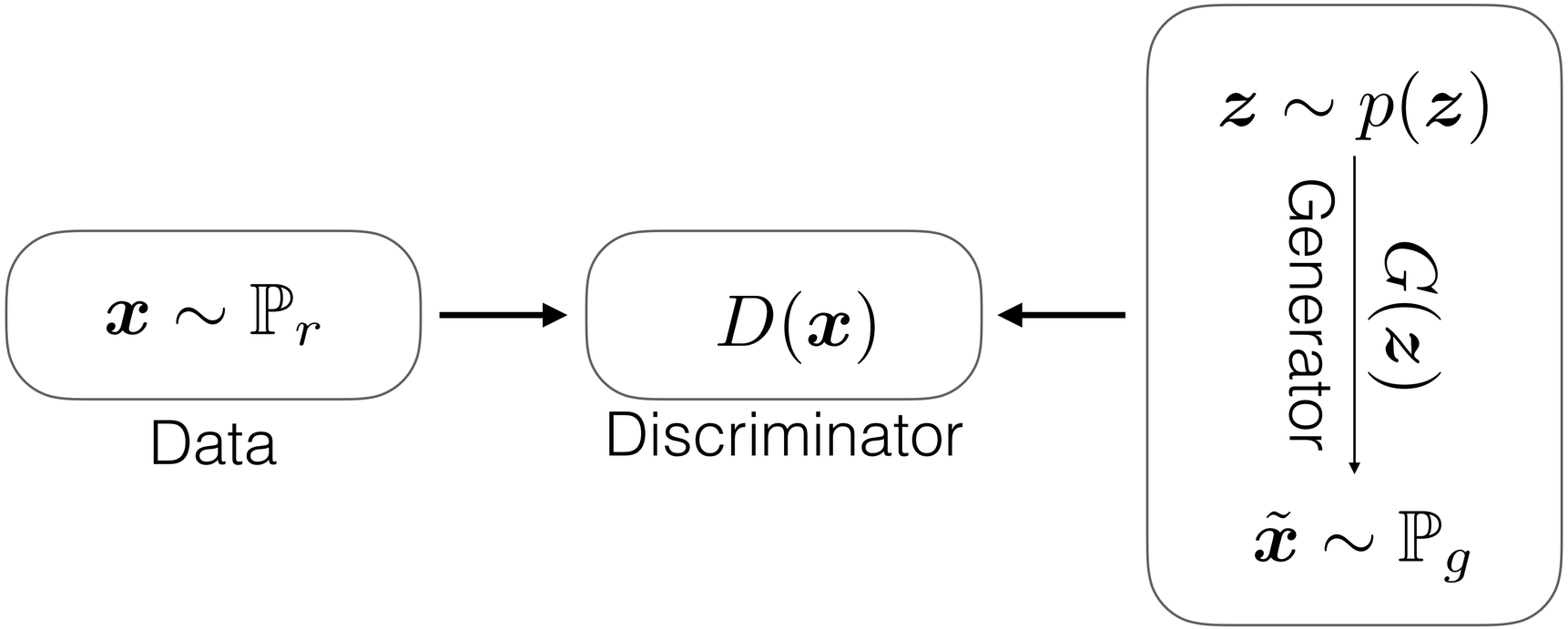
inference



Generative Adversarial Networks



Generative Adversarial Networks



GAN Objective

- Formally, express the game between discriminator D and generator G with the minimax objective:

$$\min_G \max_D \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathbb{P}_r} [\log(D(\boldsymbol{x}))] + \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g} [\log(1 - D(\tilde{\boldsymbol{x}}))].$$

  where:

  - $\mathbb{P}_r$ is the data distribution
  - $\mathbb{P}_g$ is the model distribution implicitly defined by:

$$\tilde{\boldsymbol{x}} = G(\boldsymbol{z}), \quad \boldsymbol{z} \sim p(\boldsymbol{z})$$

  - the generator input $\boldsymbol{z}$ is sampled from some simple noise distribution, (e.g. uniform or Gaussian).

GAN Theory

- Optimal (nonparametric) discriminator:

$$D^*(\boldsymbol{x}) = \frac{p_r(\boldsymbol{x})}{p_r(\boldsymbol{x}) + p_g(\boldsymbol{x})}$$

- Under an ideal discriminator, the generator minimizes the Jensen-Shannon divergence between $\mathbb{P}_r$ and $\mathbb{P}_g$.

$$\mathrm{JS}(\mathbb{P}_r \| \mathbb{P}_g) = \mathrm{KL}\left(\mathbb{P}_r \middle\| \frac{\mathbb{P}_r + \mathbb{P}_g}{2}\right) + \mathrm{KL}\left(\mathbb{P}_g \middle\| \frac{\mathbb{P}_r + \mathbb{P}_g}{2}\right)$$

$$\text{where } \mathrm{KL}(\mathbb{P}_r \| \mathbb{P}_g) = \int \log\left(\frac{p_r(x)}{p_g(x)}\right) p_r(x) d\mu(x)$$

GAN Theory … in practice

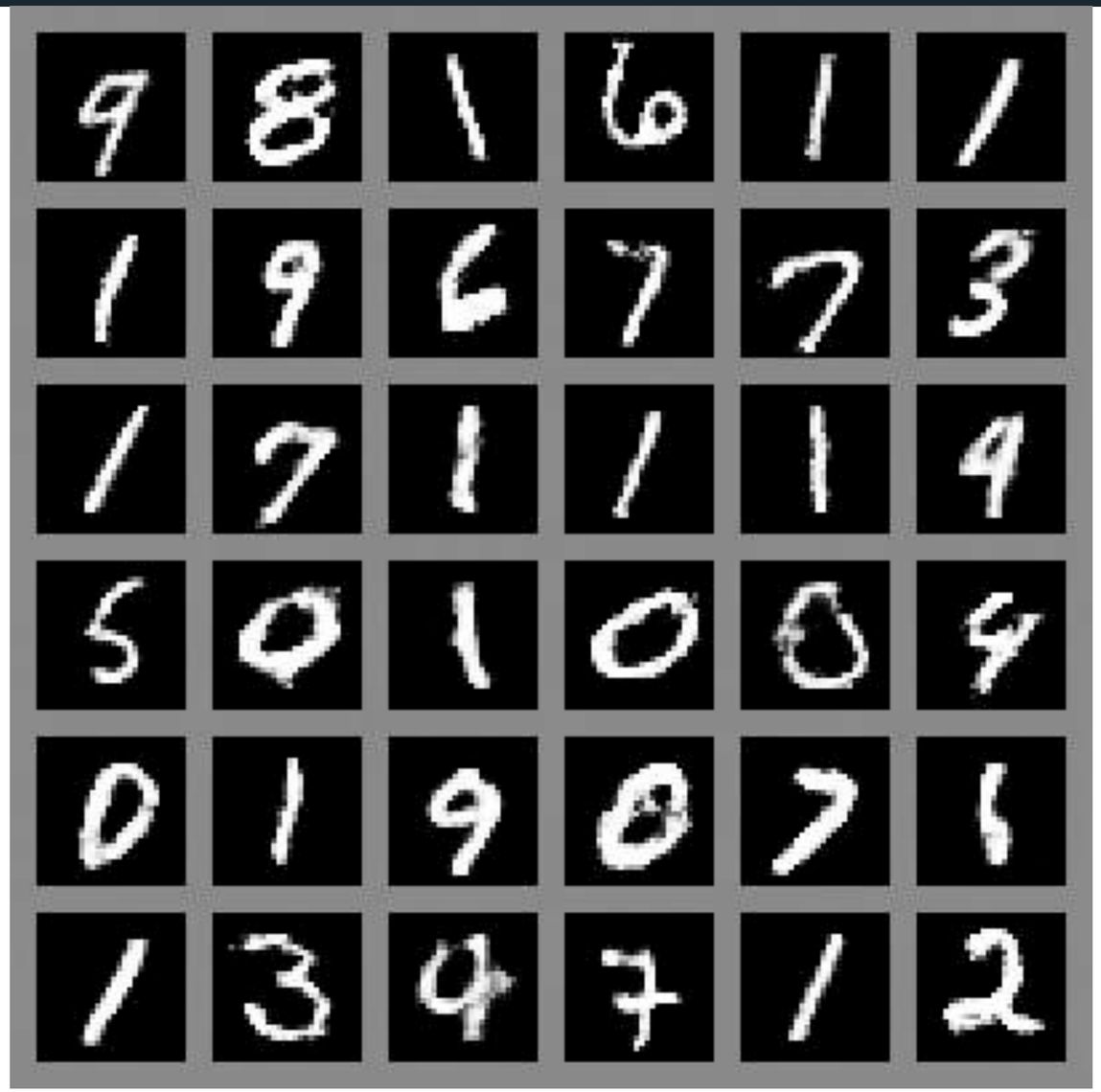
- The minimax objective leads to vanishing gradients as the discriminator saturates.
- In practice, Goodfellow et al (2014) advocate the heuristic training objective:

$$\max_D \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathbb{P}_r}[\log(D(\boldsymbol{x}))] + \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}[\log(1-D(\tilde{\boldsymbol{x}}))].$$

$$\max_G \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}[\log(D(\tilde{\boldsymbol{x}}))].$$

- However, this modified loss function can still misbehave in the presence of a good discriminator.

GAN samples



MNIST

CIFAR-10

Least-Squares GAN

Xudong Mao, Qing Li†, Haoran Xie, Raymond Y.K. Lau and Zhen Wang, ArXiv, Feb. 2017

MILA

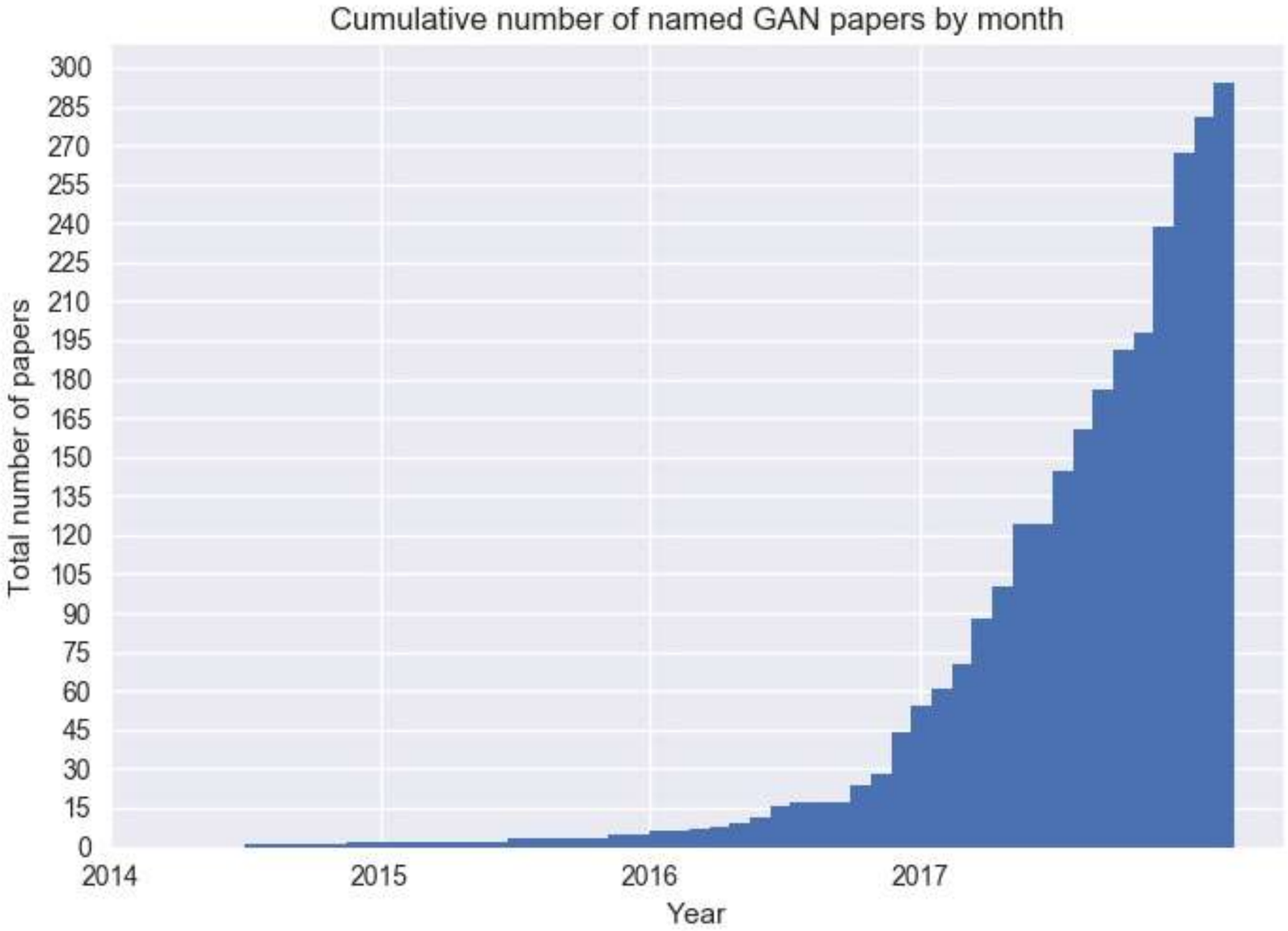
128x128 LSUN bedroom scenes

GAN Zoo

Deep Hunt, blog by Avinash Hindupur

GAN — Generative Adversarial Networks
3D-GAN — Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling
acGAN — Face Aging With Conditional Generative Adversarial Networks
AC-GAN — Conditional Image Synthesis With Auxiliary Classifier GANs
AdaGAN — AdaGAN: Boosting Generative Models
AEGAN — Learning Inverse Mapping by Autoencoder based Generative Adversarial Nets
AffGAN — Amortised MAP Inference for Image Super-resolution
AL-CGAN — Learning to Generate Images of Outdoor Scenes from Attributes and Semantic Layouts
ALI — Adversarially Learned Inference
AMGAN — Generative Adversarial Nets with Labeled Data by Activation Maximization
AnoGAN — Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery
ArtGAN — ArtGAN: Artwork Synthesis with Conditional Categorial GANs
b-GAN — b-GAN: Unified Framework of Generative Adversarial Networks
Bayesian GAN — Deep and Hierarchical Implicit Models
BEGAN — BEGAN: Boundary Equilibrium Generative Adversarial Networks
BiGAN — Adversarial Feature Learning
BS-GAN — Boundary-Seeking Generative Adversarial Networks
CGAN — Conditional Generative Adversarial Nets
CCGAN — Semi-Supervised Learning with Context-Conditional Generative Adversarial Networks
CatGAN — Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks
CoGAN — Coupled Generative Adversarial Networks
Context-RNN-GAN — Contextual RNN-GANs for Abstract Reasoning Diagram Generation
C-RNN-GAN — C-RNN-GAN: Continuous recurrent neural networks with adversarial training
CS-GAN — Improving Neural Machine Translation with Conditional Sequence Generative Adversarial Nets
CVAE-GAN — CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training
CycleGAN — Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks
DTN — Unsupervised Cross-Domain Image Generation
DCGAN — Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks
DiscoGAN — Learning to Discover Cross-Domain Relations with Generative Adversarial Networks
DR-GAN — Disentangled Representation Learning GAN for Pose-Invariant Face Recognition
DualGAN — DualGAN: Unsupervised Dual Learning for Image-to-Image Translation
EBGAN — Energy-based Generative Adversarial Network
f-GAN — f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization
GAWWN — Learning What and Where to Draw
GoGAN — Gang of GANs: Generative Adversarial Networks with Maximum Margin Ranking
GP-GAN — GP-GAN: Towards Realistic High-Resolution Image Blending
IAN — Neural Photo Editing with Introspective Adversarial Networks
iGAN — Generative Visual Manipulation on the Natural Image Manifold
IcGAN — Invertible Conditional GANs for image editing
ID-CGAN- Image De-raining Using a Conditional Generative Adversarial Network
Improved GAN — Improved Techniques for Training GANs
InfoGAN — InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets
LAGAN — Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis
LAPGAN — Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks
LR-GAN — LR-GAN: Layered Recursive Generative Adversarial Networks for Image Generation
LSGAN — Least Squares Generative Adversarial Networks
LS-GAN — Loss-Sensitive Generative Adversarial Networks on Lipschitz Densities
MGAN — Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks
MAGAN — MAGAN: Margin Adaptation for Generative Adversarial Networks
MAD-GAN — Multi-Agent Diverse Generative Adversarial Networks
MalGAN — Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN
MaliGAN — Maximum-Likelihood Augmented Discrete Generative Adversarial Networks
MARTA-GAN — Deep Unsupervised Representation Learning for Remote Sensing Images
McGAN — McGan: Mean and Covariance Feature Matching GAN
MDGAN — Mode Regularized Generative Adversarial Networks
MedGAN — Generating Multi-label Discrete Electronic Health Records using Generative Adversarial Networks
MIX+GAN — Generalization and Equilibrium in Generative Adversarial Nets (GANs)
MPM-GAN — Message Passing Multi-Agent GANs
MV-BiGAN — Multi-view Generative Adversarial Networks
pix2pix — Image-to-Image Translation with Conditional Adversarial Networks
PPGN — Plug & Play Generative Networks: Conditional Iterative Generation of Images in Latent Space
PrGAN — 3D Shape Induction from 2D Views of Multiple Objects
RenderGAN — RenderGAN: Generating Realistic Labeled Data
RTT-GAN — Recurrent Topic-Transition GAN for Visual Paragraph Generation
SGAN — Stacked Generative Adversarial Networks
SGAN — Texture Synthesis with Spatial Generative Adversarial Networks
SAD-GAN — SAD-GAN: Synthetic Autonomous Driving using Generative Adversarial Networks
SalGAN — SalGAN: Visual Saliency Prediction with Generative Adversarial Networks
SEGAN — SEGAN: Speech Enhancement Generative Adversarial Network
SeGAN — SeGAN: Segmenting and Generating the Invisible
SeqGAN — SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient
SimGAN — Learning from Simulated and Unsupervised Images through Adversarial Training
SketchGAN — Adversarial Training For Sketch Retrieval
SL-GAN — Semi-Latent GAN: Learning to generate and modify facial images from attributes
Softmax-GAN — Softmax GAN
SRGAN — Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network
S^2GAN — Generative Image Modeling using Style and Structure Adversarial Networks
SSL-GAN — Semi-Supervised Learning with Context-Conditional Generative Adversarial Networks
StackGAN — StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks
TGAN — Temporal Generative Adversarial Nets
TAC-GAN — TAC-GAN — Text Conditioned Auxiliary Classifier Generative Adversarial Network
TP-GAN — Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis
Triple-GAN — Triple Generative Adversarial Nets
Unrolled GAN — Unrolled Generative Adversarial Networks
VGAN — Generating Videos with Scene Dynamics
VGAN — Generative Adversarial Networks as Variational Training of Energy Based Models
VAE-GAN — Autoencoding beyond pixels using a learned similarity metric
VariGAN — Multi-View Image Generation from a Single-View
ViGAN — Image Generation and Editing with Variational Info Generative AdversarialNetworks
WGAN — Wasserstein GAN
WGAN-GP — Improved Training of Wasserstein GANs
WaterGAN — WaterGAN: Unsupervised Generative Network to Enable Real-time Color Correction of Monocular Underwater Images

An explo-GAN of papers



Explosive growth—All the named GAN variants cumulatively since 2014.

Credit: Bruno Gavranović

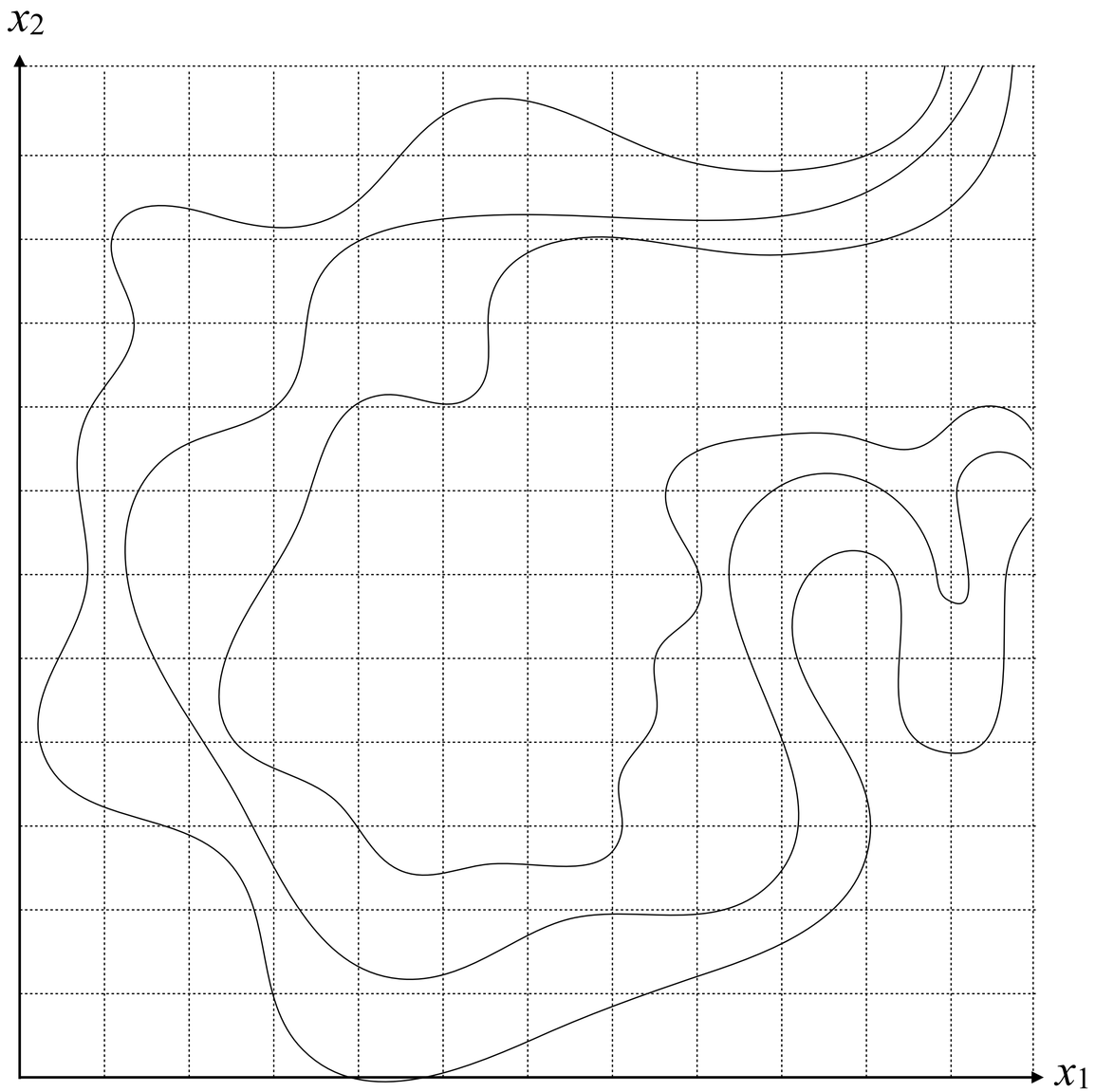
(from Deep Hunt, blog by Avinash Hindupur)

DCGAN samples (Radford, Metz and Chintala; 2016)

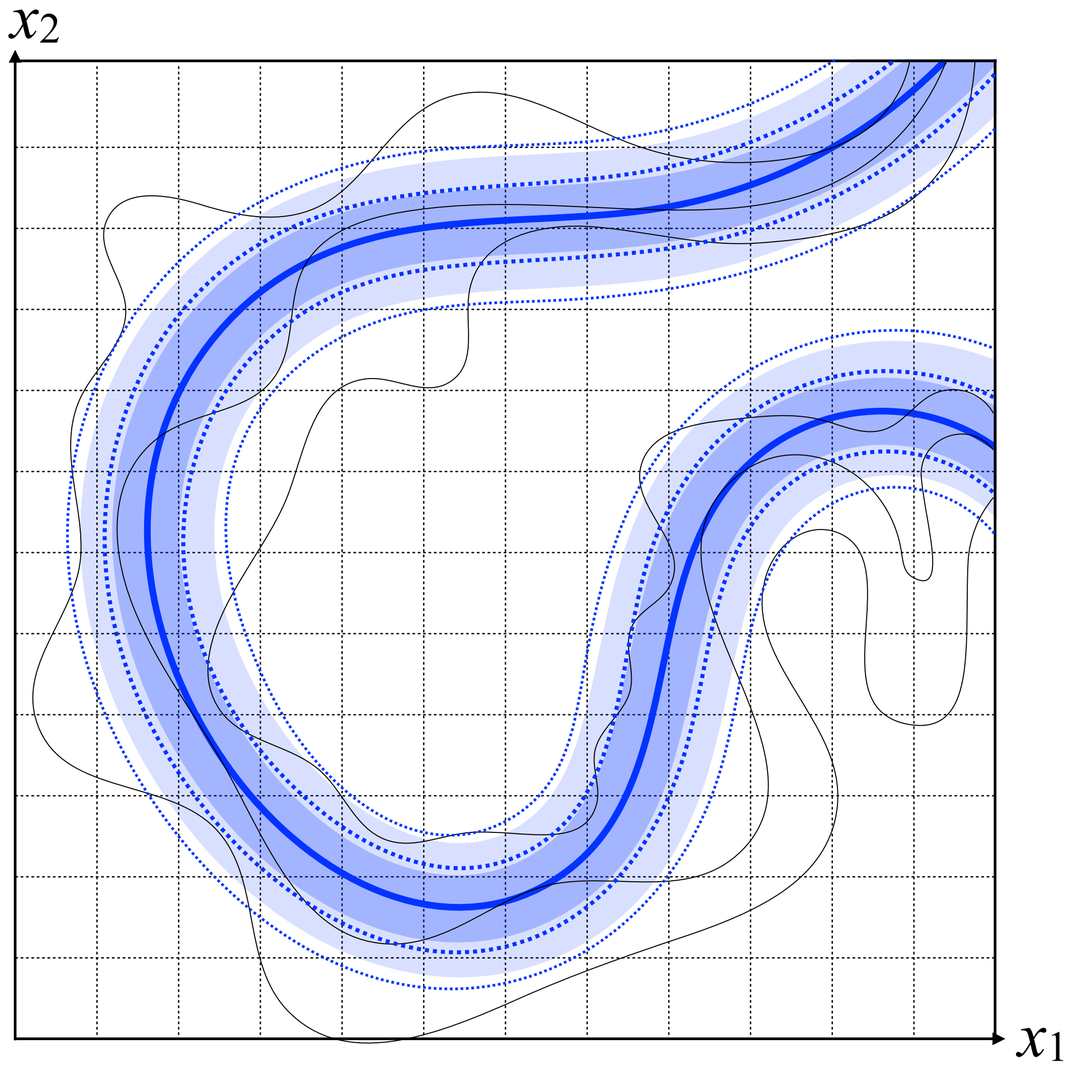
Z-space interpolations

LSUN bedroom scenes

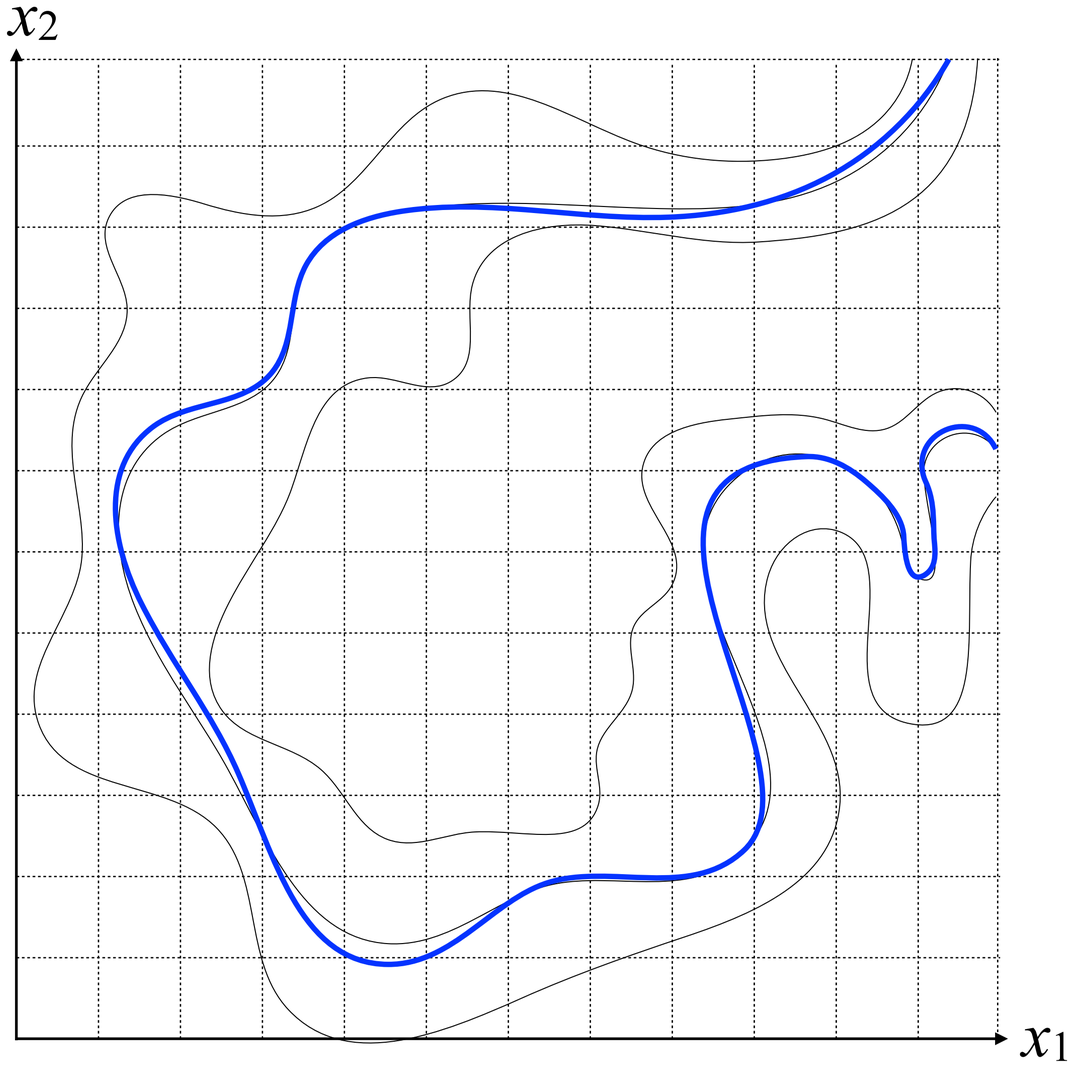
Cartoon of the Image manifold



What makes GANs special?



more traditional max-likelihood approach



GAN

f-GAN:Variational Divergence Minimization

Nowozin et al (2016)

- The f-divergence family: (generalizes KL divergence)

$$D_f(P\|Q) = \int_{\mathcal{X}} q(x) f\left(\frac{p(x)}{q(x)}\right) \mathrm{d}x,$$

Name	$D_f(P\|Q)$	Generator $f(u)$	$T^*(x)$
Kullback-Leibler	$\int p(x) \log \frac{p(x)}{q(x)} \mathrm{d}x$	$u \log u$	$1 + \log \frac{p(x)}{q(x)}$
Reverse KL	$\int q(x) \log \frac{q(x)}{p(x)} \mathrm{d}x$	$-\log u$	$-\frac{q(x)}{p(x)}$
Pearson χ^2	$\int \frac{(q(x)-p(x))^2}{p(x)} \mathrm{d}x$	$(u-1)^2$	$2(\frac{p(x)}{q(x)} - 1)$
Squared Hellinger	$\int \left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 \mathrm{d}x$	$\left(\sqrt{u}-1\right)^2$	$(\sqrt{\frac{p(x)}{q(x)}} - 1) \cdot \sqrt{\frac{q(x)}{p(x)}}$
Jensen-Shannon	$\frac{1}{2}\int p(x) \log \frac{2p(x)}{p(x)+q(x)} + q(x) \log \frac{2q(x)}{p(x)+q(x)} \mathrm{d}x$	$-(u+1)\log \frac{1+u}{2} + u \log u$	$\log \frac{2p(x)}{p(x)+q(x)}$
GAN	$\int p(x) \log \frac{2p(x)}{p(x)+q(x)} + q(x) \log \frac{2q(x)}{p(x)+q(x)} \mathrm{d}x - \log(4)$	$u \log u - (u+1)\log(u+1)$	$\log \frac{p(x)}{p(x)+q(x)}$

Table 1: List of f-divergences $D_f(P\|Q)$ together with generator functions. Part of the list of divergences and their generators is based on [26]. For all divergences we have $f : \mathrm{dom}_f \to \mathbb{R} \cup \{+\infty\}$, where f is convex and lower-semicontinuous. Also we have $f(1) = 0$ which ensures that $D_f(P\|P) = 0$ for any distribution P. As shown by [10] GAN is related to the Jensen-Shannon divergence through $D_{\mathrm{GAN}} = 2D_{\mathrm{JS}} - \log(4)$.

f-GAN:Variational Divergence Minimization

Nowozin et al (2016)

- The f-divergence family: (generalizes KL divergence)

$$D_f(P\|Q) = \int_{\mathcal{X}} q(x) f\left(\frac{p(x)}{q(x)}\right) \mathrm{d}x,$$

- Nowozin et al propose to make use of the *Fenchel conjugate* to redefine the divergence:

$$f^*(t) = \sup_{u \in \mathrm{dom}_f} \{ut - f(u)\}.$$

f-GAN:Variational Divergence Minimization

Nowozin et al (2016)

- The f-divergence family: (generalizes KL divergence)

$$
\begin{aligned}
D_f(P\|Q) &= \int_{\mathcal{X}} q(x) \sup_{t \in \mathrm{dom}_{f^*}} \left\{ t\frac{p(x)}{q(x)} - f^*(t) \right\} \mathrm{d}x \\
&\geq \sup_{T \in \mathcal{T}} \left(\int_{\mathcal{X}} p(x)\, T(x)\, \mathrm{d}x - \int_{\mathcal{X}} q(x)\, f^*(T(x))\, \mathrm{d}x \right) \\
&= \sup_{T \in \mathcal{T}} \left(\mathbb{E}_{x \sim P}\left[T(x) \right] - \mathbb{E}_{x \sim Q}\left[f^*(T(x)) \right] \right),
\end{aligned}
$$

Looks like the GAN objective

f-GAN:Variational Divergence Minimization

Nowozin et al (2016)

- The f-divergence family: (expressed as a GAN)

$$D_f(P\|Q) \geq \sup_{T\in\mathcal{T}} \left(\mathbb{E}_{x\sim P}\left[T(x)\right] - \mathbb{E}_{x\sim Q}\left[f^*(T(x))\right]\right),$$

Name	Output activation g_f	dom_{f^*}	Conjugate $f^*(t)$	$f'(1)$
Kullback-Leibler (KL)	v	$\mathbb{R}$	$\exp(t-1)$	1
Reverse KL	$-\exp(-v)$	$\mathbb{R}_-$	$-1-\log(-t)$	-1
Pearson χ^2	v	$\mathbb{R}$	$\frac{1}{4}t^2+t$	0
Squared Hellinger	$1-\exp(-v)$	$t<1$	$\frac{t}{1-t}$	0
Jensen-Shannon	$\log(2)-\log(1+\exp(-v))$	$t<\log(2)$	$-\log(2-\exp(t))$	0
GAN	$-\log(1+\exp(-v))$	$\mathbb{R}_-$	$-\log(1-\exp(t))$	$-\log(2)$

Table 2: Recommended final layer activation functions and critical variational function level defined by $f'(1)$. The critical value $f'(1)$ can be interpreted as a classification threshold applied to $T(x)$ to distinguish between true and generated samples.

GAN Evaluation

- Quantitatively evaluating GANs is not straightforward:
  - Max Likelihood is a poor indication of sample quality.
  - continues to be a topic of research.

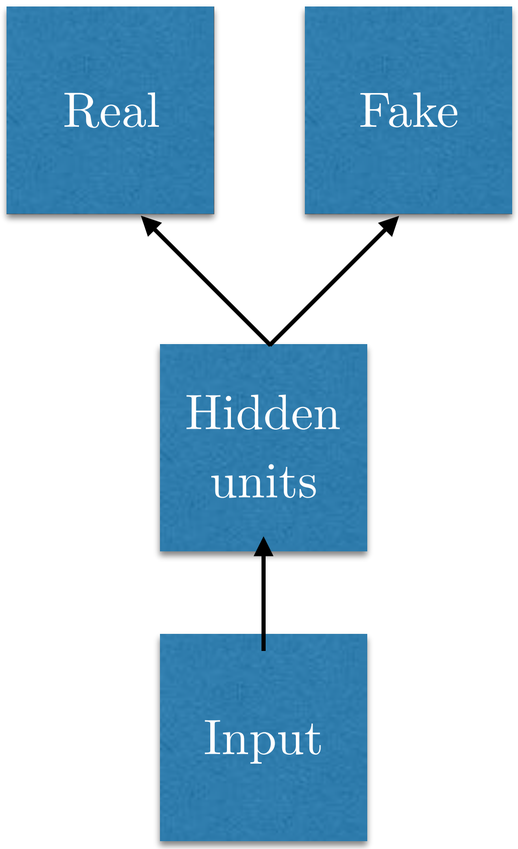
- Evaluation metrics (selected)
  - **Inception Score**: (y = labels given gen. image.)

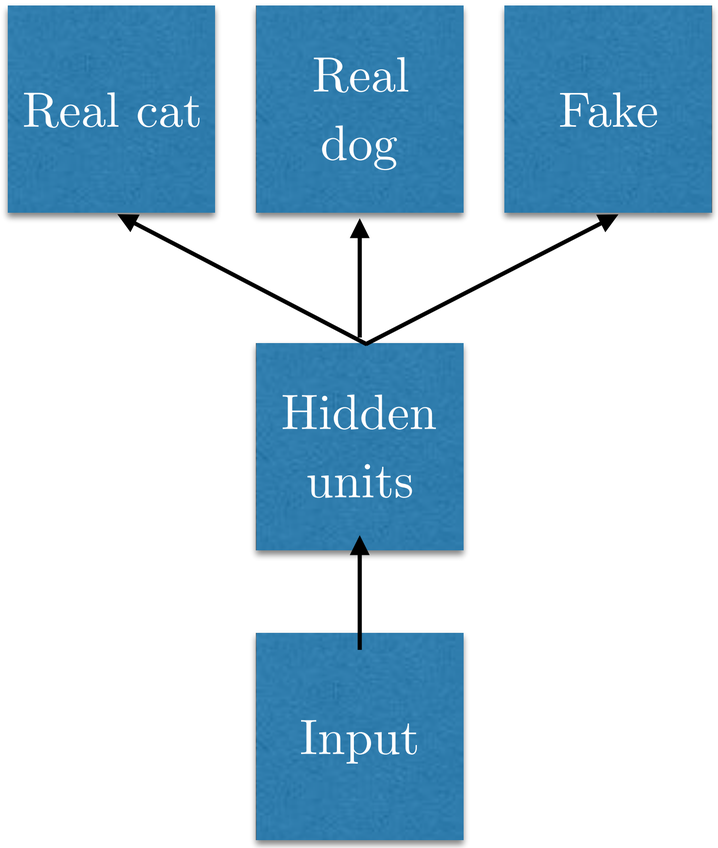
$$\mathrm{IS}(\mathbb{P}_g) = e^{\mathbb{E}_{\mathbf{x}\sim\mathbb{P}_g}[KL(p_{\mathcal{M}}(y|\mathbf{x})||p_{\mathcal{M}}(y))]}$$

  - **Mode Score**: improved version of the Inception Score
    (taking the difference in marginal label distribution into account.)
  - **Kernel MMD** (Maximum Mean Discrepancy):

$$\mathrm{MMD}(\mathbb{P}_r, \mathbb{P}_g) = \left(\mathbb{E}_{\substack{\mathbf{x}_r, \mathbf{x}'_r \sim \mathbb{P}_r, \\ \mathbf{x}_g, \mathbf{x}'_g \sim \mathbb{P}_g}} \left[k(\mathbf{x}_r, \mathbf{x}'_r) - 2k(\mathbf{x}_r, \mathbf{x}_g) + k(\mathbf{x}_g, \mathbf{x}'_g) \right] \right)^{\frac{1}{2}}$$

Supervised Discriminator





(Odena 2016, Salimans et al 2016)

Semi-Supervised Classification

MILA

MNIST (Permutation Invariant)

Model	Number of incorrectly predicted test examples for a given number of labeled samples			
	20	50	100	200
DGN [21]			333 ± 14	
Virtual Adversarial [22]			212	
CatGAN [14]			191 ± 10	
Skip Deep Generative Model [23]			132 ± 7	
Ladder network [24]			106 ± 37	
Auxiliary Deep Generative Model [23]			96 ± 2	
Our model	1677 ± 452	221 ± 136	93 ± 6.5	90 ± 4.2
Ensemble of 10 of our models	1134 ± 445	142 ± 96	86 ± 5.6	81 ± 4.3

(Salimans et al 2016)

Semi-Supervised Classification

CIFAR-10

Model	Test error rate for a given number of labeled samples			
	1000	2000	4000	8000
Ladder network [24]			20.40±0.47	
CatGAN [14]			19.58±0.46	
Our model	21.83±2.01	19.61±2.09	18.63±2.32	17.72±1.82
Ensemble of 10 of our models	19.22±0.54	17.25±0.66	15.59±0.47	14.87±0.89

SVHN

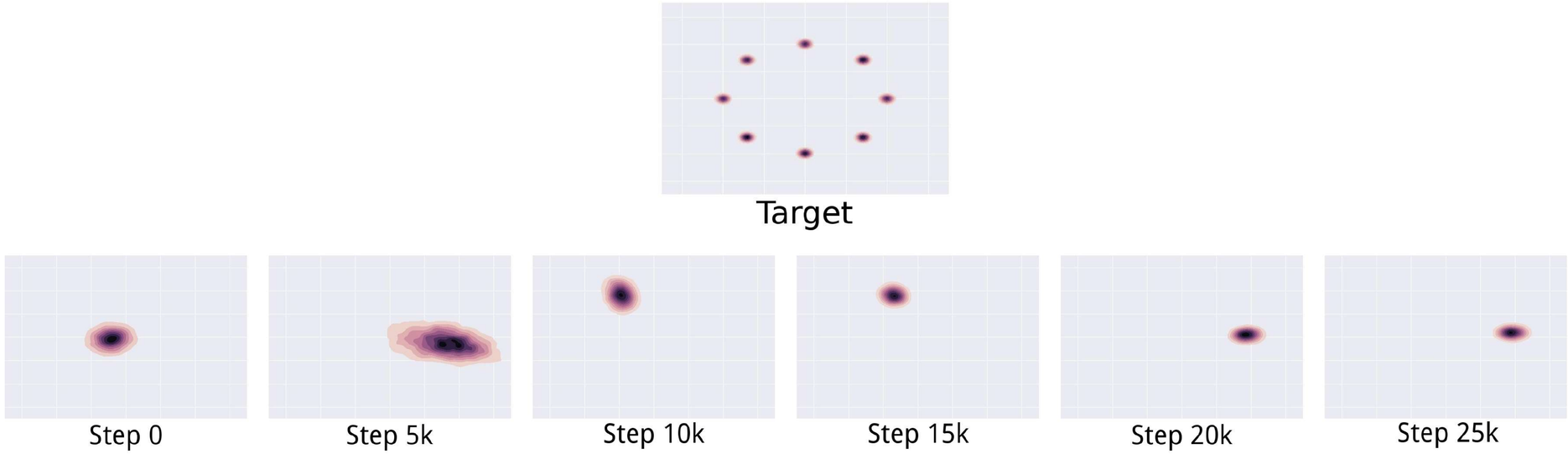
Model	Percentage of incorrectly predicted test examples for a given number of labeled samples		
	500	1000	2000
DGN [21]		36.02±0.10	
Virtual Adversarial [22]		24.63	
Auxiliary Deep Generative Model [23]		22.86	
Skip Deep Generative Model [23]		16.61±0.24	
Our model	18.44 ± 4.8	8.11 ± 1.3	6.16 ± 0.58
Ensemble of 10 of our models		5.88 ± 1.0	

(Salimans et al 2016)

GAN Failures: Mode Collapse

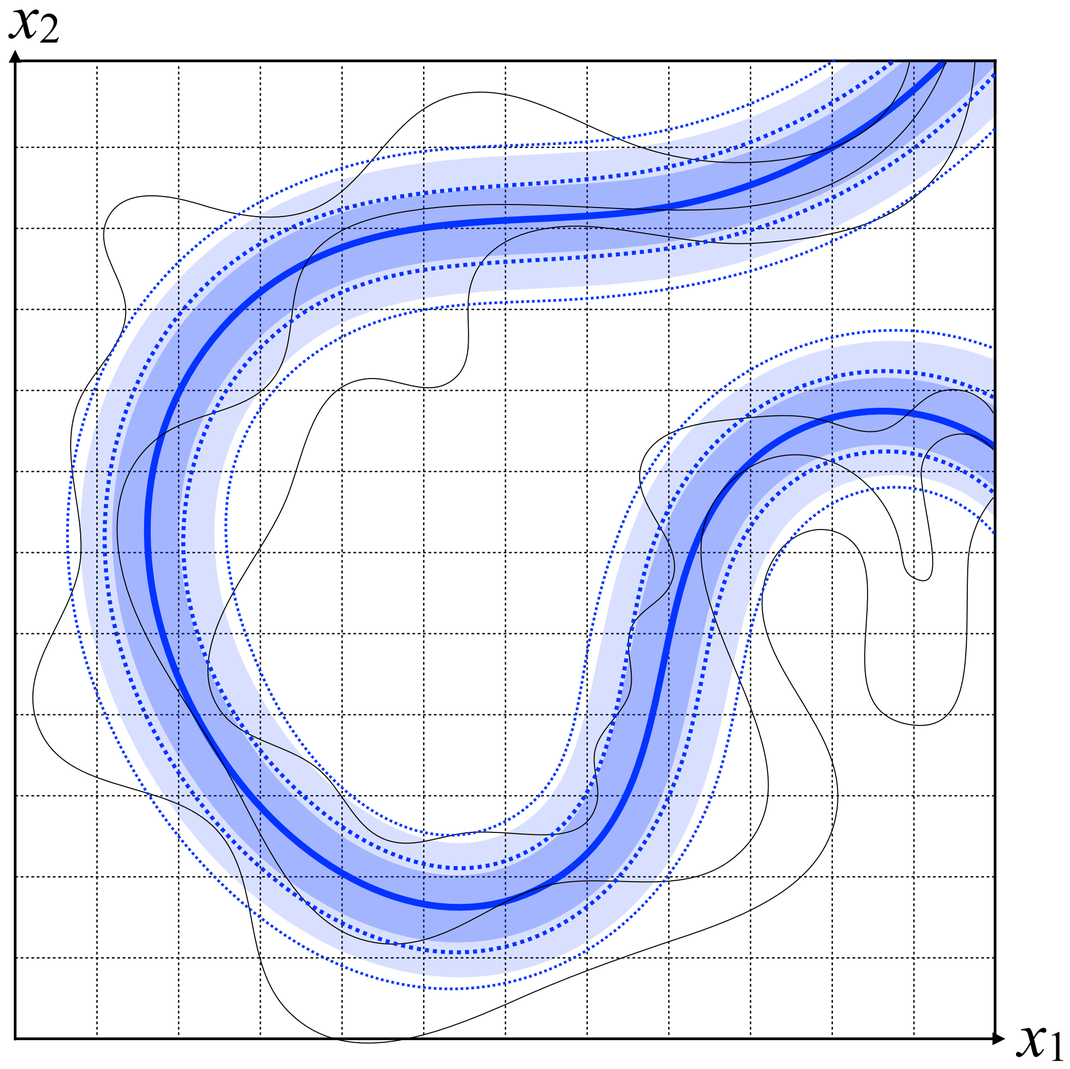
$$\min_G \max_D V(G, D) \neq \max_D \min_G V(G, D)$$

- D in inner loop: convergence to correct distribution
- G in inner loop: place all mass on most likely point

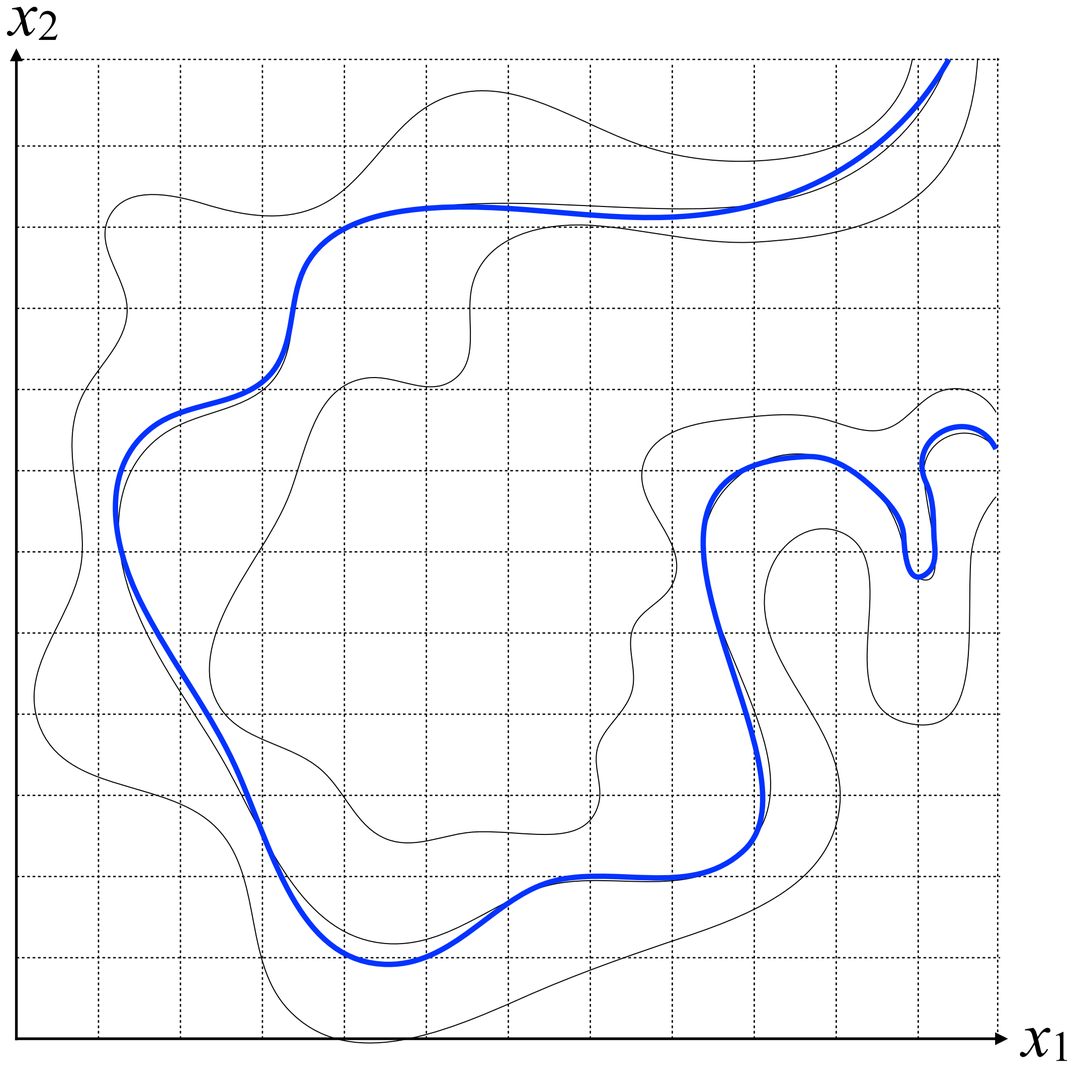


(Metz et al 2016)

What makes GANs special?



more traditional max-likelihood approach



GAN

Mode Collapse: Measure

Sanjeev Arora and Yi Zhang (2017)
Do GANs actually learn the distribution? An empirical study

- Presents an estimate of support size based on the Birthday Paradox:

  - Suppose there are k people in a room. How large must k be before we have a high likelihood of having two people with the same birthday? Clearly, if we want 100% probability, then k > 366 suffices. But assuming people's birthdays are iid draws from some distribution on [1, 366] it can be checked that the probability exceeds 50% even when k is as small as 23.

  - Suppose a distribution has support N. The birthday paradox says that a sample of size about $N^{1/2}$ would be quite likely to have a duplicate.

Mode Collapse: Measure

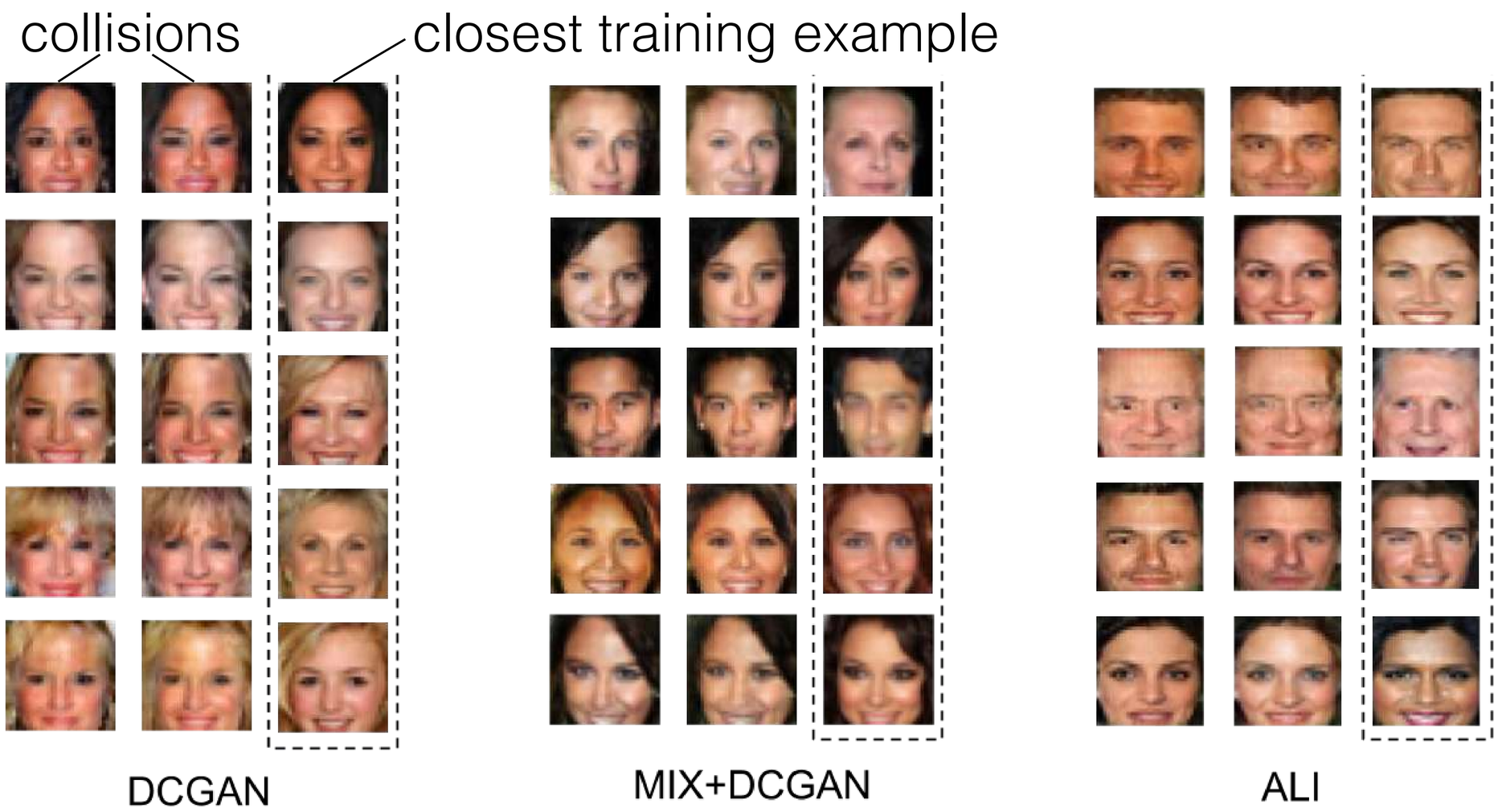
Sanjeev Arora and Yi Zhang (2017)
Do GANs actually learn the distribution? An empirical study

- Presents an estimate of support size based on the Birthday Paradox:
  (a) Pick a sample of size s from the generated distribution.
  (b) Use an automated measure of image similarity to flag the 20 (say) most similar pairs in the sample.
  (c) Visually inspect the flagged pairs and check for duplicates.
  (d) Repeat.

- If this test reveals that samples of size s have duplicate images with good probability, then suspect that the distribution has support size about s^2.

Mode Collapse: Measure

Sanjeev Arora and Yi Zhang (2017)



Mode Collapse: Measure

Sanjeev Arora and Yi Zhang (2017): Show a dependence on Discriminator size

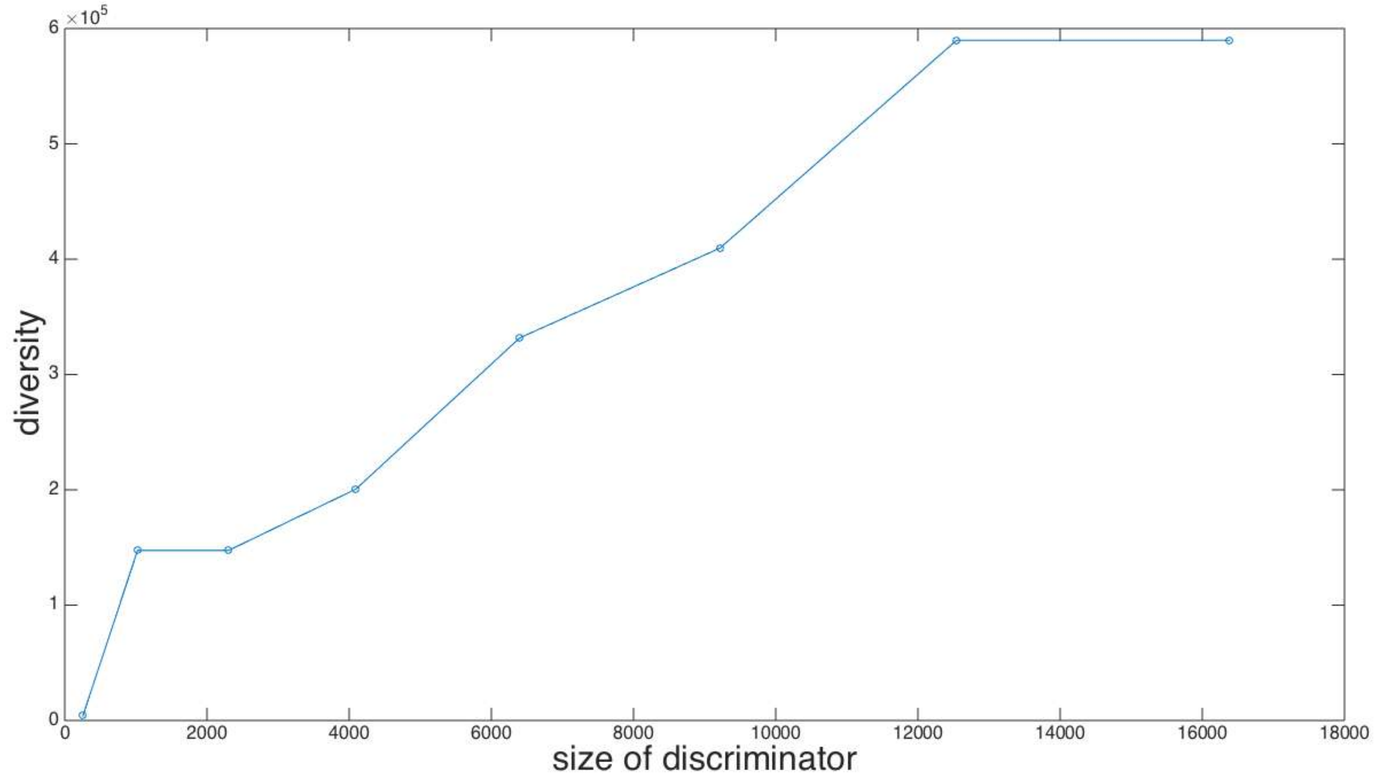
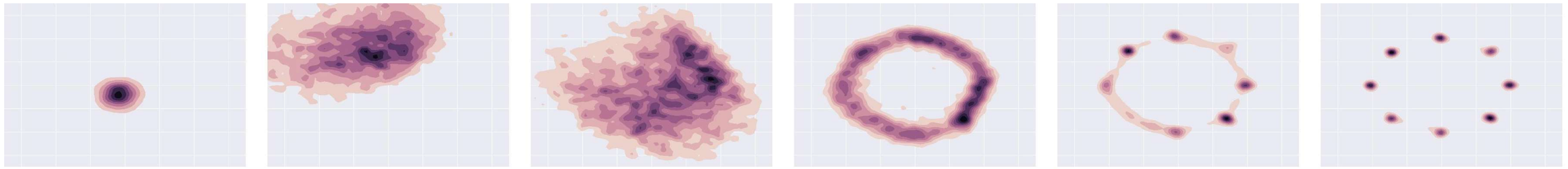


Figure 2: Diversity's dependence on discriminator size. The diversity is measured as the square of the batch size needed to encounter collision w.p. $\geq 50\%$ v.s. size of discriminator. The discriminator size is explained in the main article.

Mode Collapse: Solutions

- **Unrolled GANs** (Metz et al 2016): Prevents mode collapse by backproping through a set of (k) updates of the discriminator to update generator parameters.

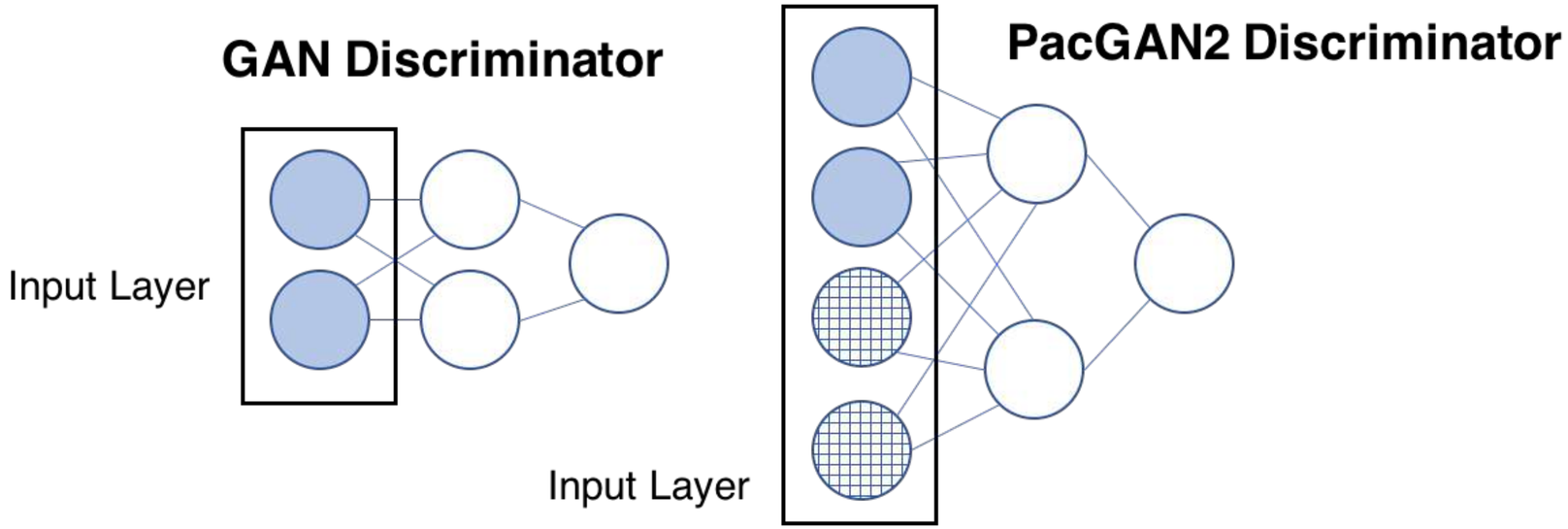


- **VEEGAN** (Srivastava et al 2017): Introduce a reconstructor network which is learned both to map the true data distribution p(x) to a Gaussian and to approximately invert the generator network.

Mode Collapse: Solutions

- **Minibatch Discrimination** (Salimans et al 2016): Add minibatch features that classify each example by comparing it to other members of the minibatch (Salimans et al 2016)

- ***PacGAN****: The power of two samples in generative adversarial networks* (Lin et al 2017): Also uses multisample discrimination.



Mode Collapse: Solutions

- ***PacGAN**: The power of two samples in generative adversarial networks* (Lin et al 2017)

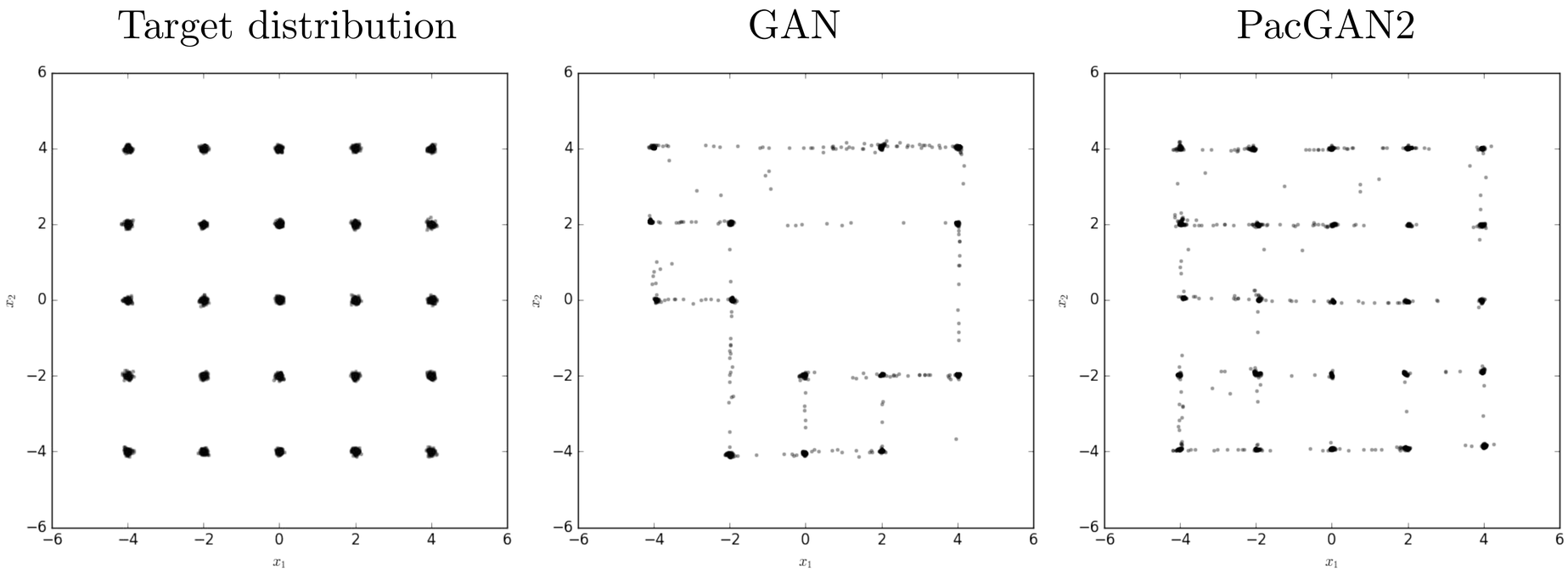
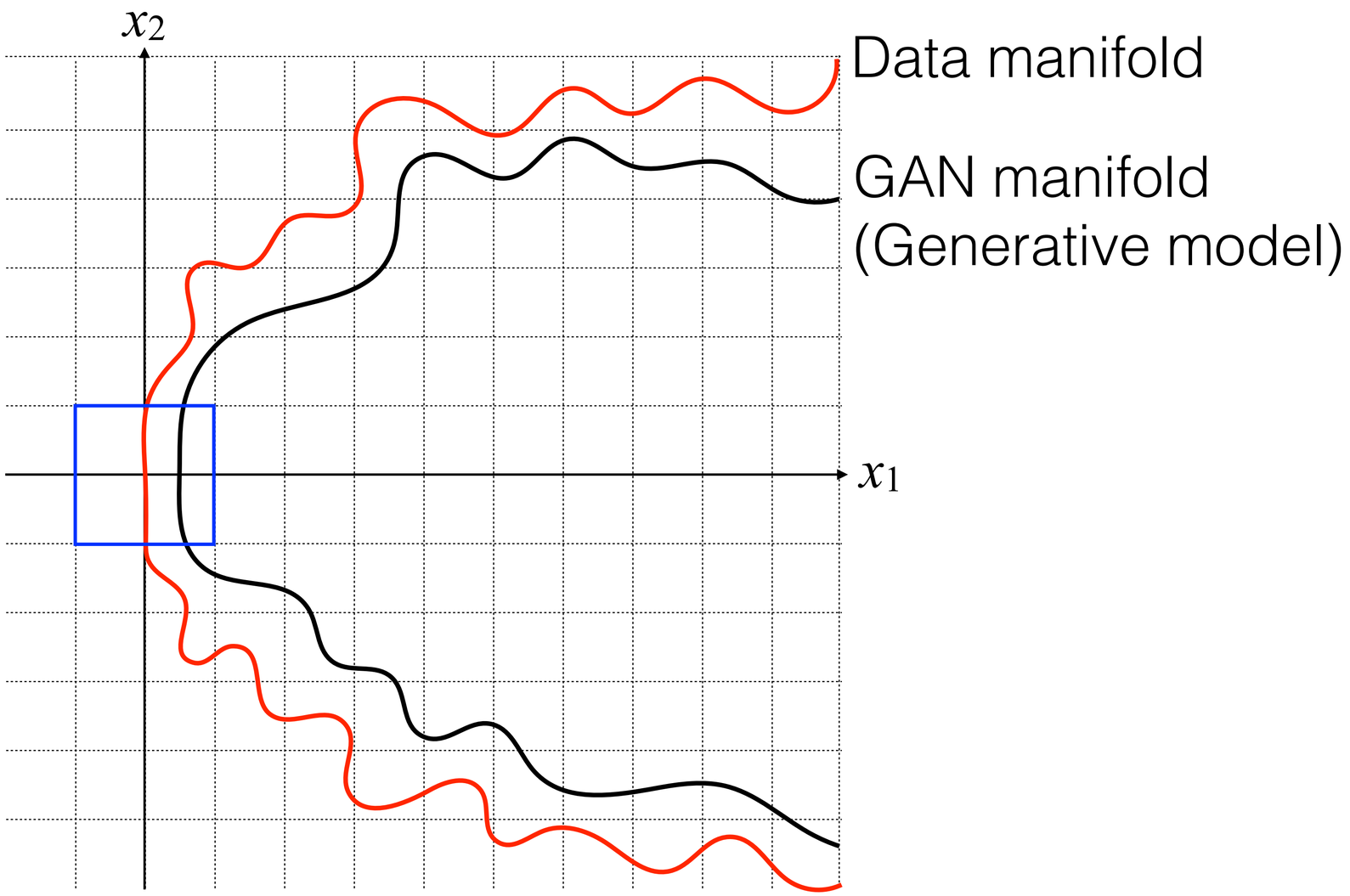


Figure 2: Scatter plot of the 2D samples from the true distribution (left) of 2D-grid and the learned generators using GAN (middle) and PacGAN2 (right). PacGAN2 captures all of the 25 modes.

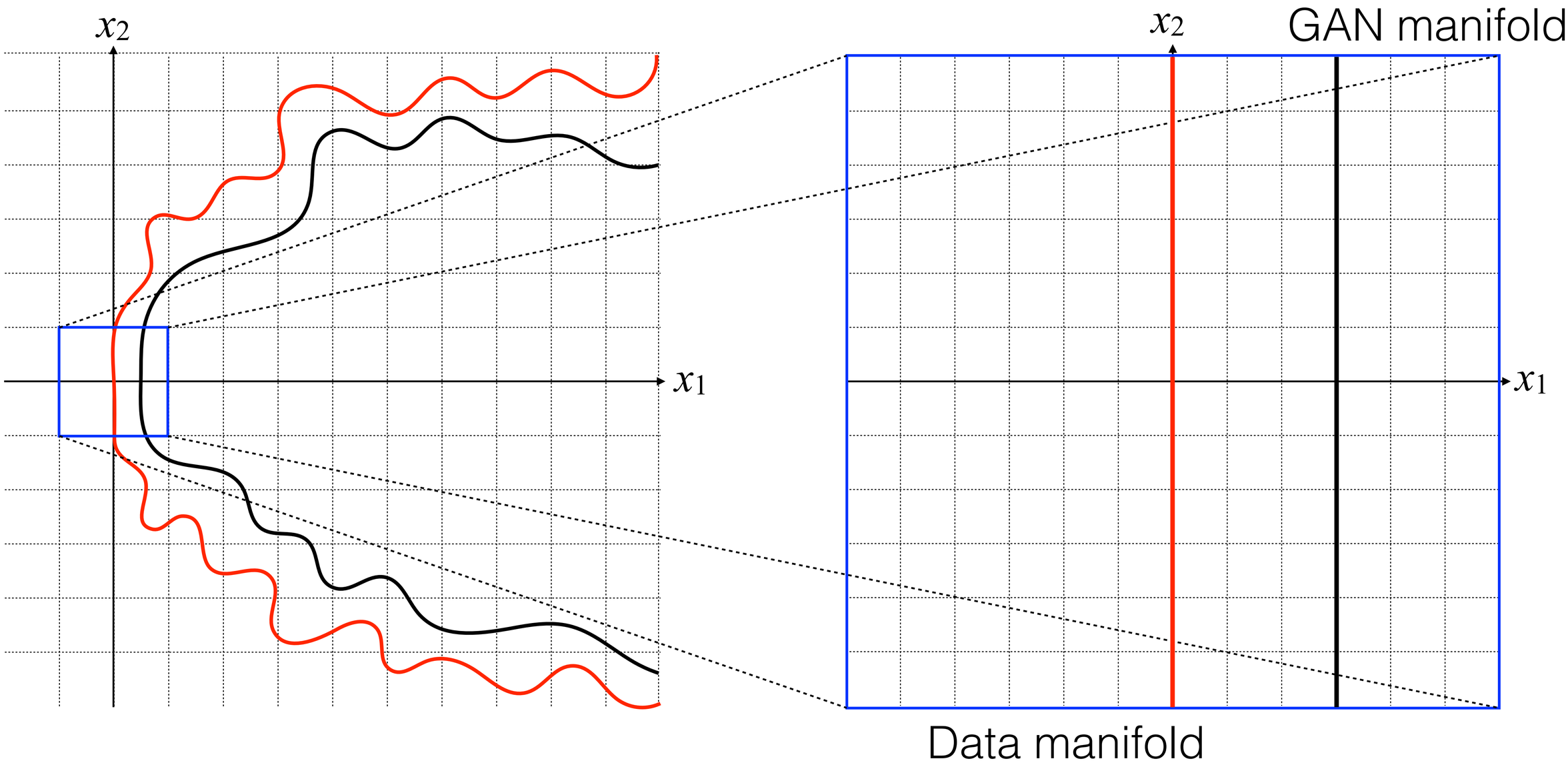
Training a GAN: Distances between Manifolds



Data manifold

GAN manifold
(Generative model)

Training a GAN: Distances between Manifolds

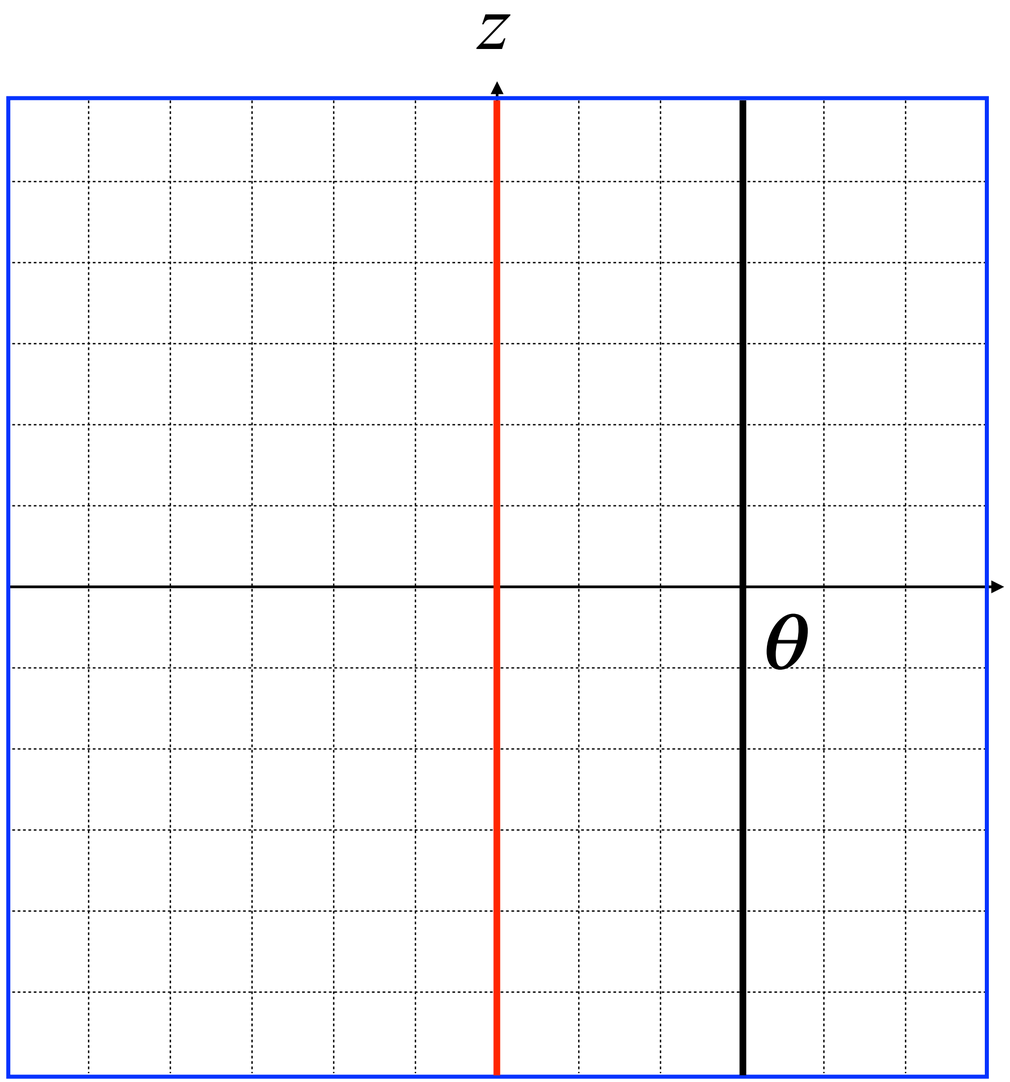


Jensen-Shannon Divergence

$$\mathrm{JS}(\mathbb{P}_r \| \mathbb{P}_g) = \mathrm{KL}\left(\mathbb{P}_r \middle\| \frac{\mathbb{P}_r + \mathbb{P}_g}{2}\right) + \mathrm{KL}\left(\mathbb{P}_g \middle\| \frac{\mathbb{P}_r + \mathbb{P}_g}{2}\right)$$

- What is the JS divergence in this simple case?

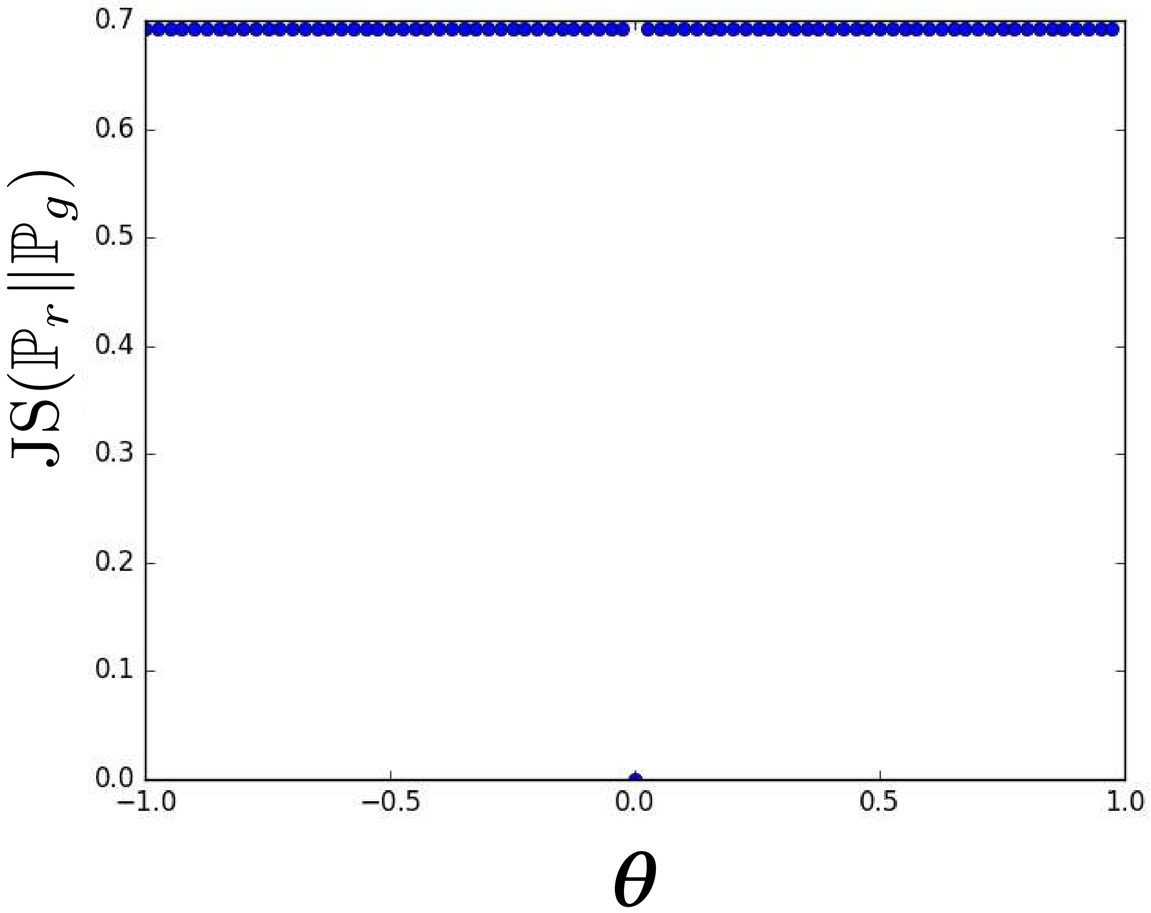
$$\mathrm{JS}(\mathbb{P}_r \| \mathbb{P}_g) = \begin{cases} \log 2 & \text{if } \theta \neq 0, \\ 0 & \text{if } \theta = 0, \end{cases}$$

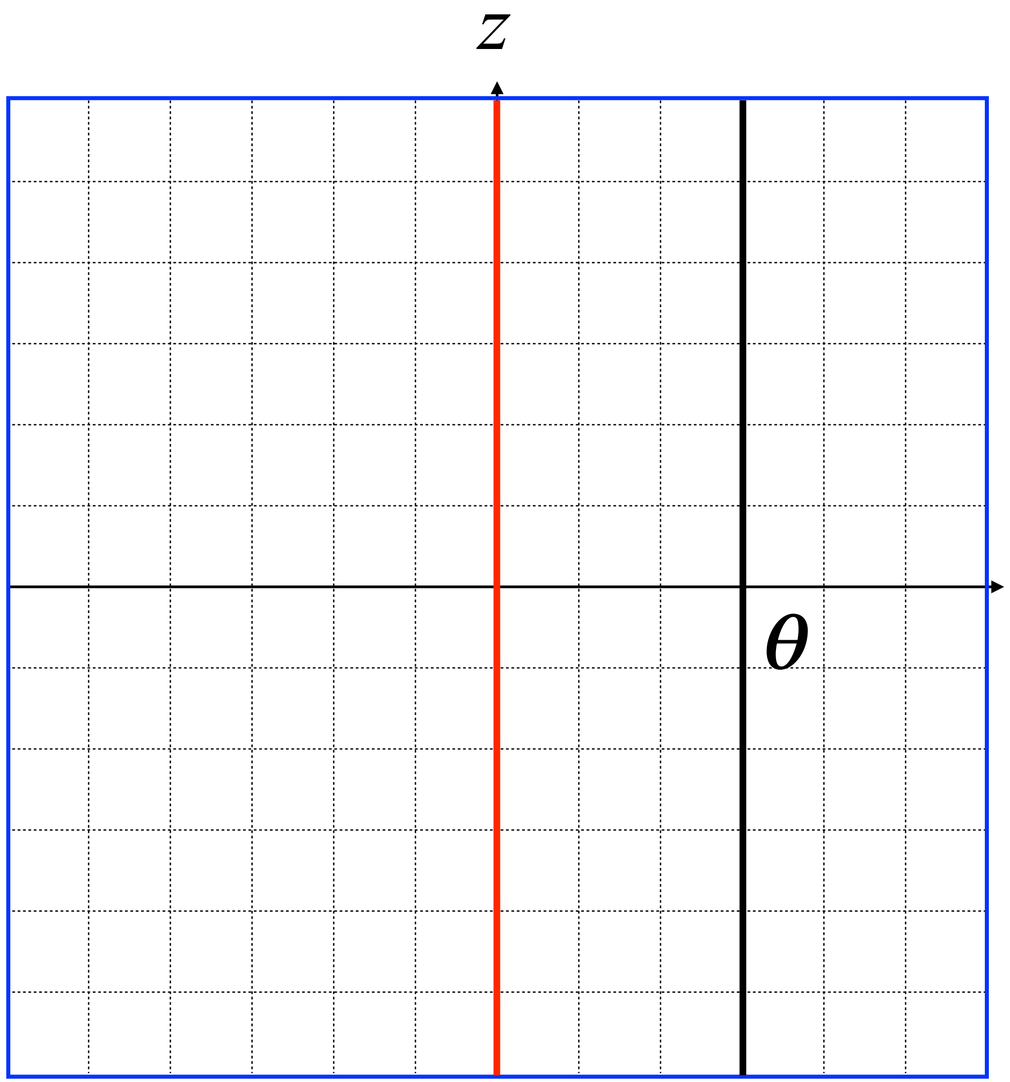


Example from Arjovsky et al. 2017

Jensen-Shannon Divergence

$$\mathrm{JS}(\mathbb{P}_r \| \mathbb{P}_g) = \begin{cases} \log 2 & \text{if } \theta \neq 0, \\ 0 & \text{if } \theta = 0, \end{cases}$$





Example from Arjovsky et al. 2017

Earth-Movers Distance

- JS divergence is not a useful learning signal to train GANs.
- An alternative: Earth-Mover (also called Wasserstein-1) distance.

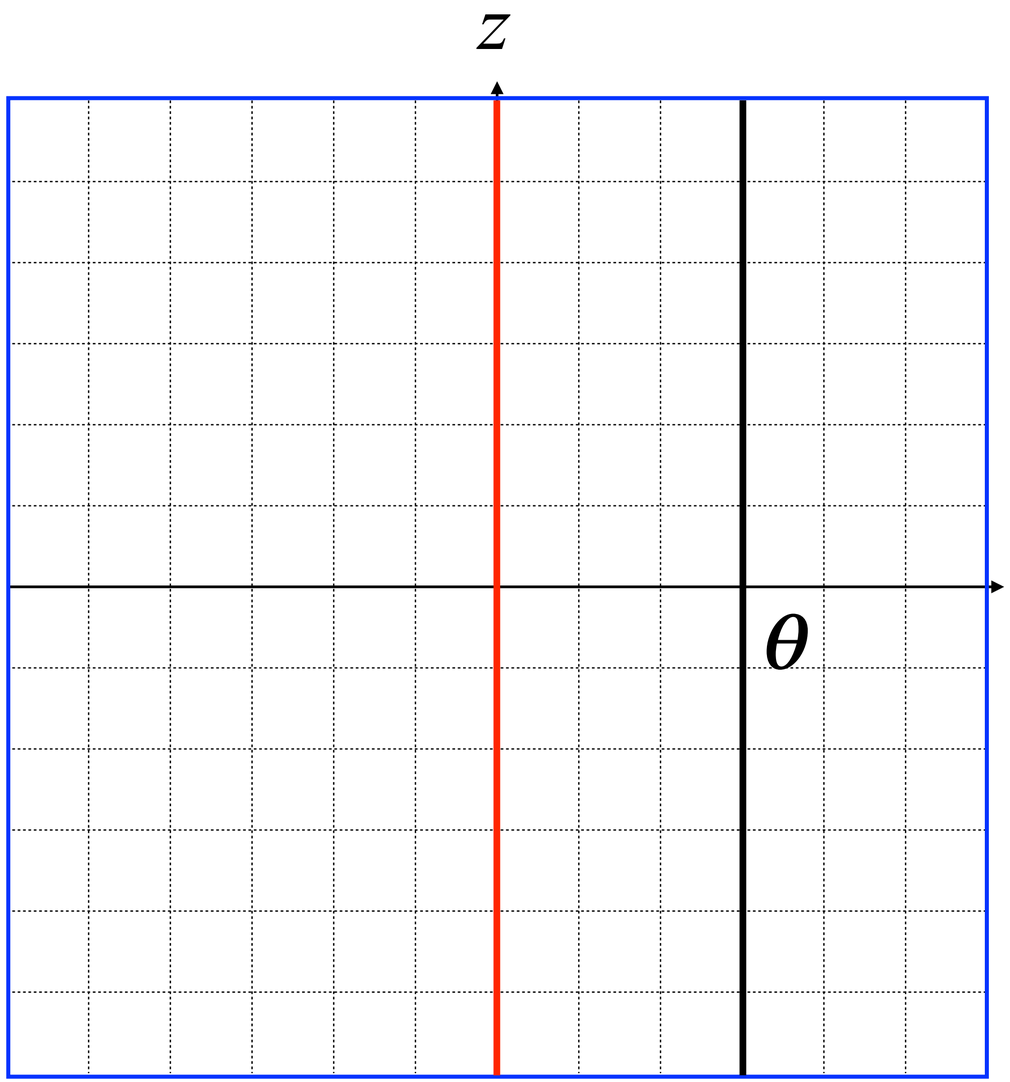
$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma} \left[\|x - y\| \right]$$

  - Minimum cost of transporting mass to transform the distribution $\mathbb{P}_r$ into the distribution $\mathbb{P}_g$.
  - The EM distance is continuous everywhere and differentiable almost everywhere (under mild assumptions).

Wasserstein Distance

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma} \left[\|x - y\| \right]$$

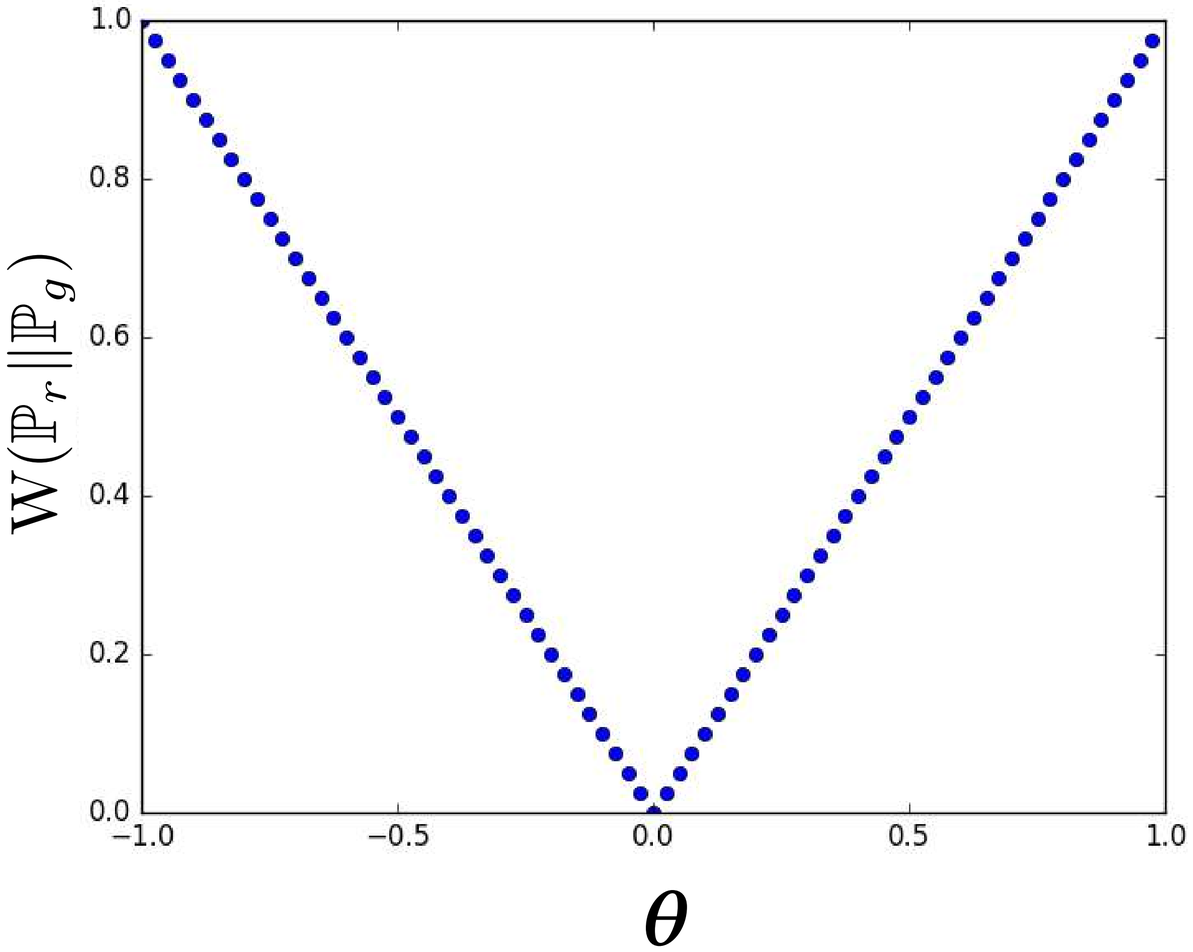
- What is the EM (or Wasserstein) distance in this simple case?

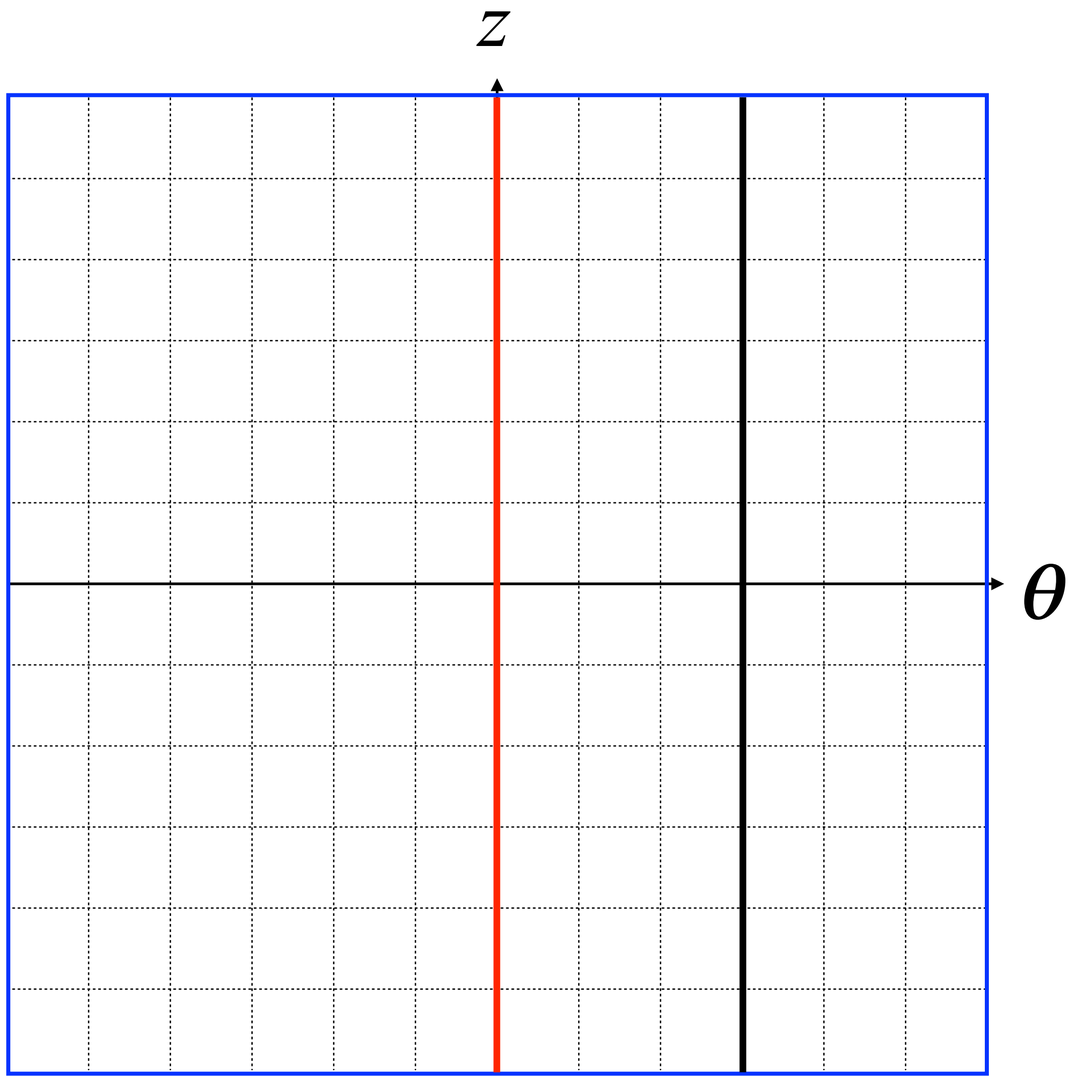


Example from Arjovsky et al. 2017

Wasserstein Distance

$$\mathrm{W}(\mathbb{P}_r \| \mathbb{P}_g) = |\theta|$$

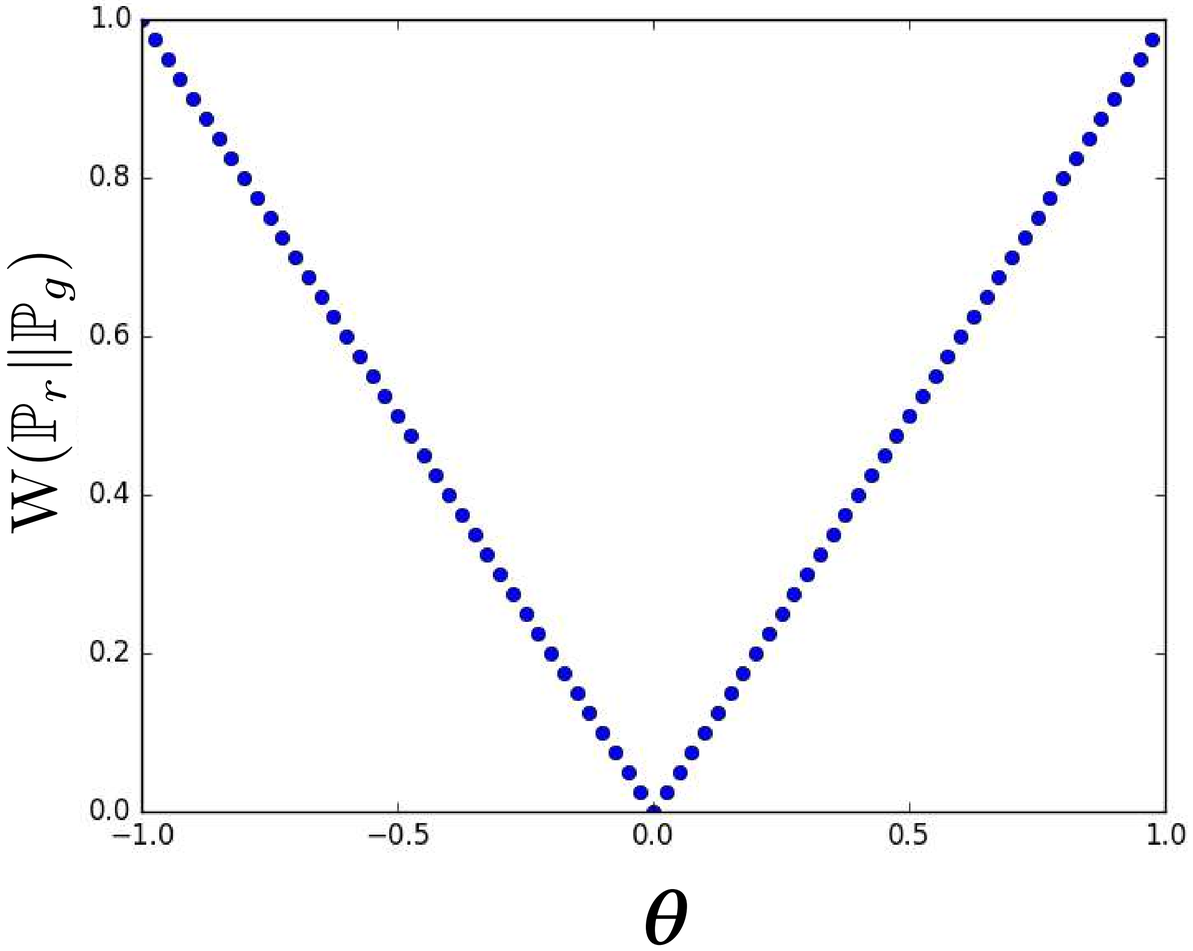


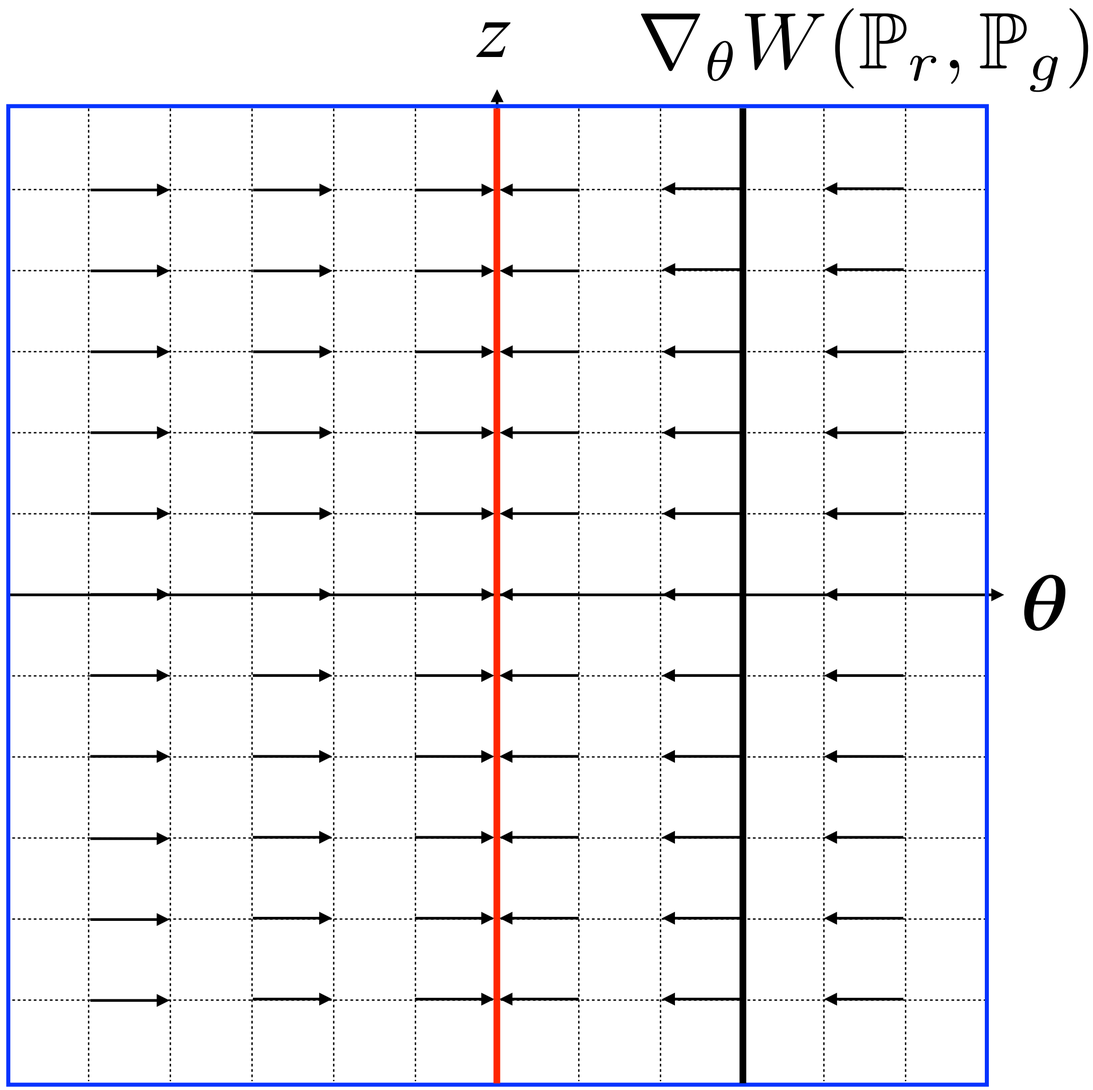


Example from Arjovsky et al. 2017

Wasserstein Distance

$$\mathrm{W}(\mathbb{P}_r \| \mathbb{P}_g) = |\theta|$$





Example from Arjovsky et al. 2017

Wasserstein GAN Arjovsky, Chintala, Bottou (2017)

- $W(\mathbb{P}_r, \mathbb{P}_g)$ might have nice properties compared to $\mathrm{JS}(\mathbb{P}_r, \mathbb{P}_g)$
- However, the infimum is intractable in:

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma} \left[\|x - y\| \right]$$

- Can exploit Kantorovich-Rubinstein duality:

$$W(\mathbb{P}_r, \mathbb{P}_g) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_g}[f(x)]$$

where the supremum is over all the 1-Lipschitz functions $f : \mathcal{X} \to \mathbb{R}$

Wasserstein GAN Arjovsky, Chintala, Bottou (2017)

MILA

- The WGAN Objective function:

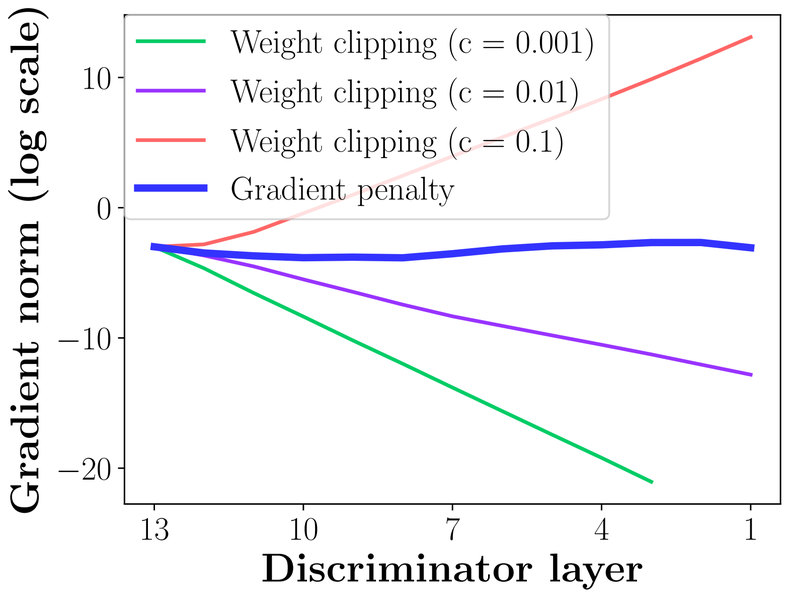
$$\min_{G} \max_{D \in \mathcal{D}} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathbb{P}_r} \left[D(\boldsymbol{x}) \right] - \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g} \left[D(\tilde{\boldsymbol{x}}) \right]$$

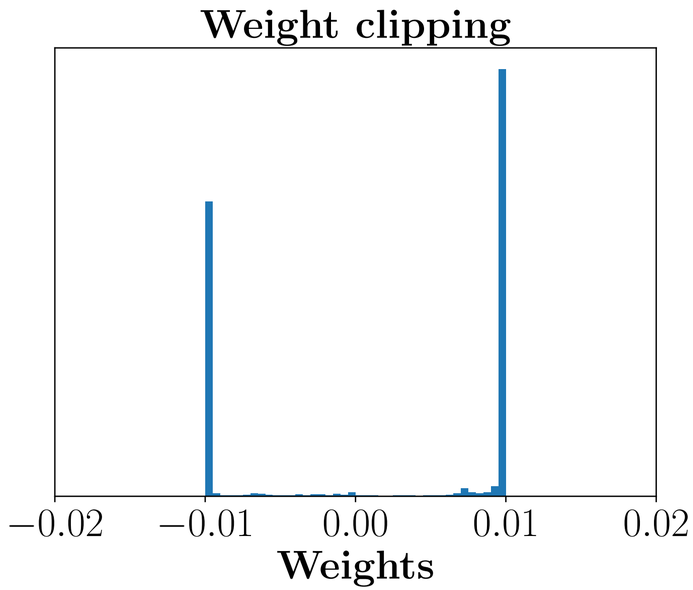
  where $\mathcal{D}$ is the set of 1-Lipschitz functions.

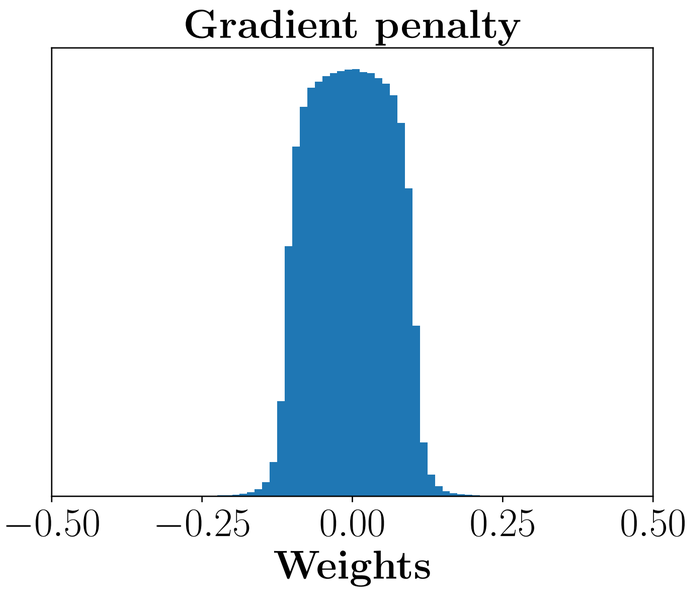
- Open question: how to effectively enforce the Lipschitz constraint on the critic D?
  - Arjovsky et al. (2017) propose to clip the weights of the critic to lie within a compact space [-c, c].
  - Results in a subset of the k-Lipschitz functions (k is a function of c).

Issues with Weight Clipping

1. Underuse capacity
2. Exploding and vanishing gradients







Gradient Penalty Approach

Gulrajani, Ahmed, Arjovsky, Dumoulin, Courville (2017)

- **A property of the optimal WGAN critic**: If $\tilde{\boldsymbol{x}} \sim \mathbb{P}_g$ then there is a point $\boldsymbol{x} \sim \mathbb{P}_r$, such that for all points $\boldsymbol{x}_t = t\boldsymbol{x} + (1-t)\tilde{\boldsymbol{x}}$ (on a straight line between $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$) then:

$$\nabla D^*(\boldsymbol{x}_t) = \frac{\boldsymbol{x} - \boldsymbol{x}_t}{\|\boldsymbol{x} - \boldsymbol{x}_t\|}$$

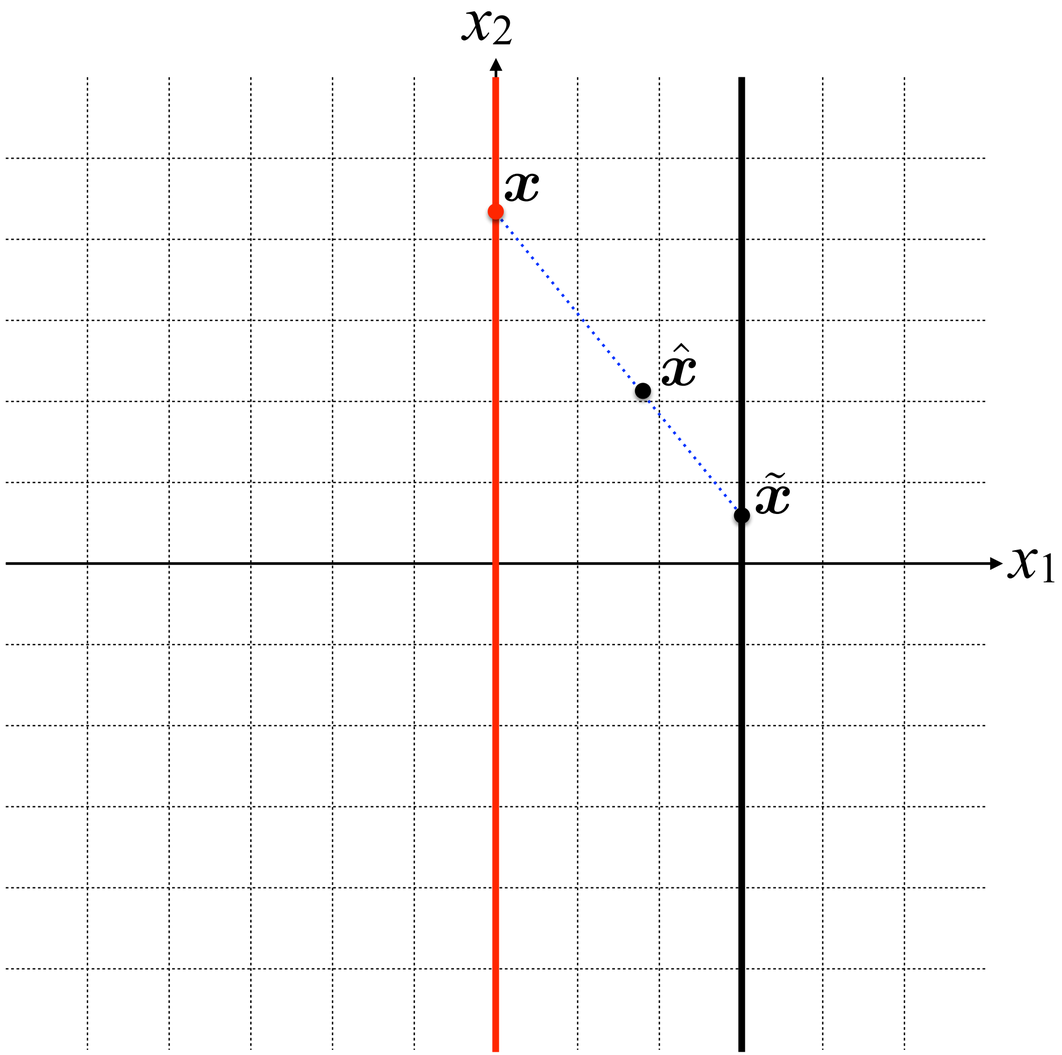
- This implies the optimal WGAN critic has gradient norm 1 at $\boldsymbol{x}_t$

- Gradient Penalty version of WGAN (i.e. the WGAN-GP) objective:

$$L = \underbrace{\mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g}[D(\tilde{\boldsymbol{x}})] - \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathbb{P}_r}[D(\boldsymbol{x})]}_{\text{Original critic loss}} + \underbrace{\lambda \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}} \sim \mathbb{P}_{\hat{\boldsymbol{x}}}}\left[(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1)^2\right]}_{\text{Our gradient penalty}}$$

Gradient Penalty Approach

Gulrajani, Ahmed, Arjovsky, Dumoulin, Courville (2017)



Gradient penalty:

$$\mathop{\mathbb{E}}_{\hat{\boldsymbol{x}} \sim \mathbb{P}_{\hat{\boldsymbol{x}}}} \left[(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1)^2 \right]$$

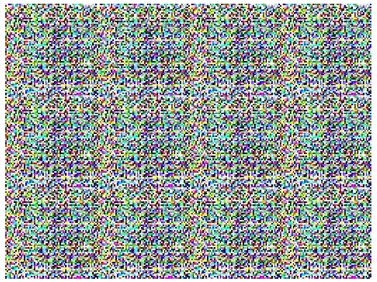
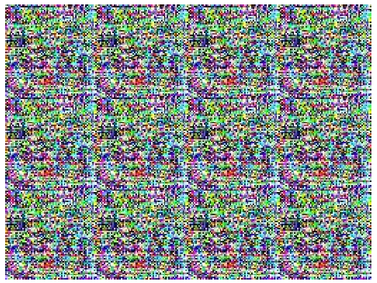
Sample along straight lines:

$$\epsilon \sim U[0,1], \boldsymbol{x} \sim \mathbb{P}_r, \tilde{\boldsymbol{x}} \sim \mathbb{P}_g$$

$$\hat{\boldsymbol{x}} = \epsilon \boldsymbol{x} + (1-\epsilon)\tilde{\boldsymbol{x}}$$

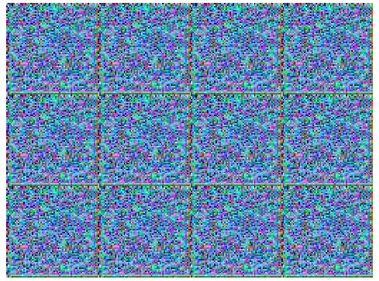
Comparison on difficult to train architectures

- Comparison based on recommended default parameter setting for each algorithm.
- WGAN-GP is more robust to variations in training setups.

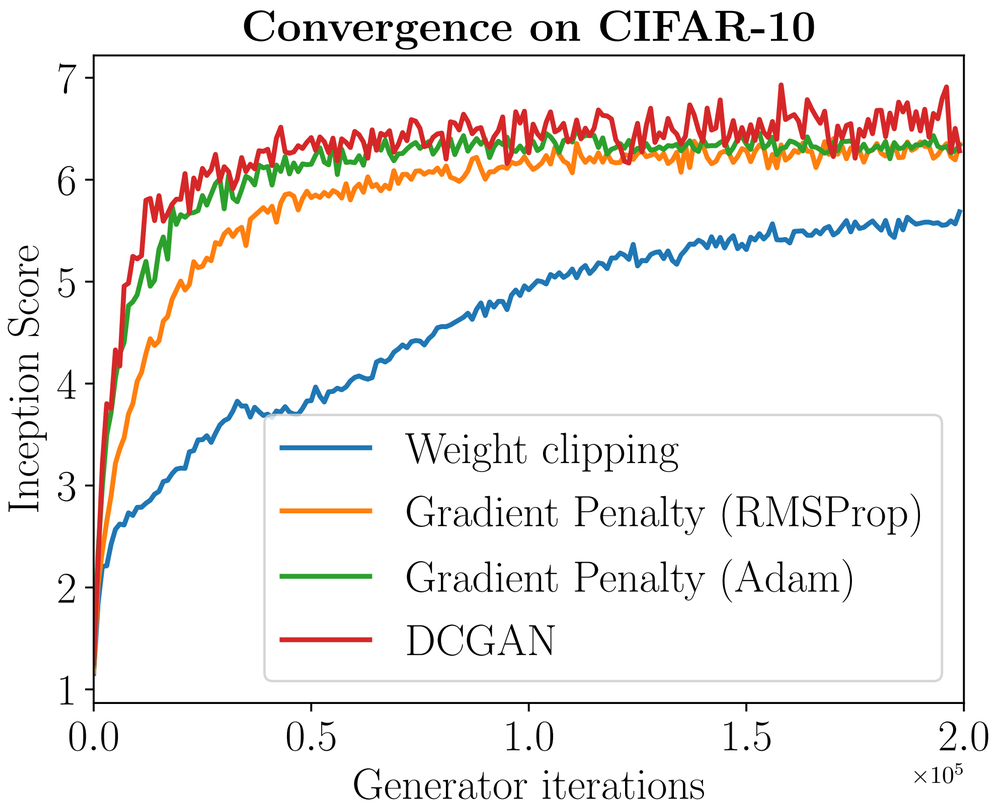
DCGAN	LSGAN	WGAN (clipping)	WGAN-GP (ours)
Baseline (G: DCGAN, D: DCGAN)			
			
G: No BN and a constant number of filters, D: DCGAN			
			
G: 4-layer 512-dim ReLU MLP, D: DCGAN			
			

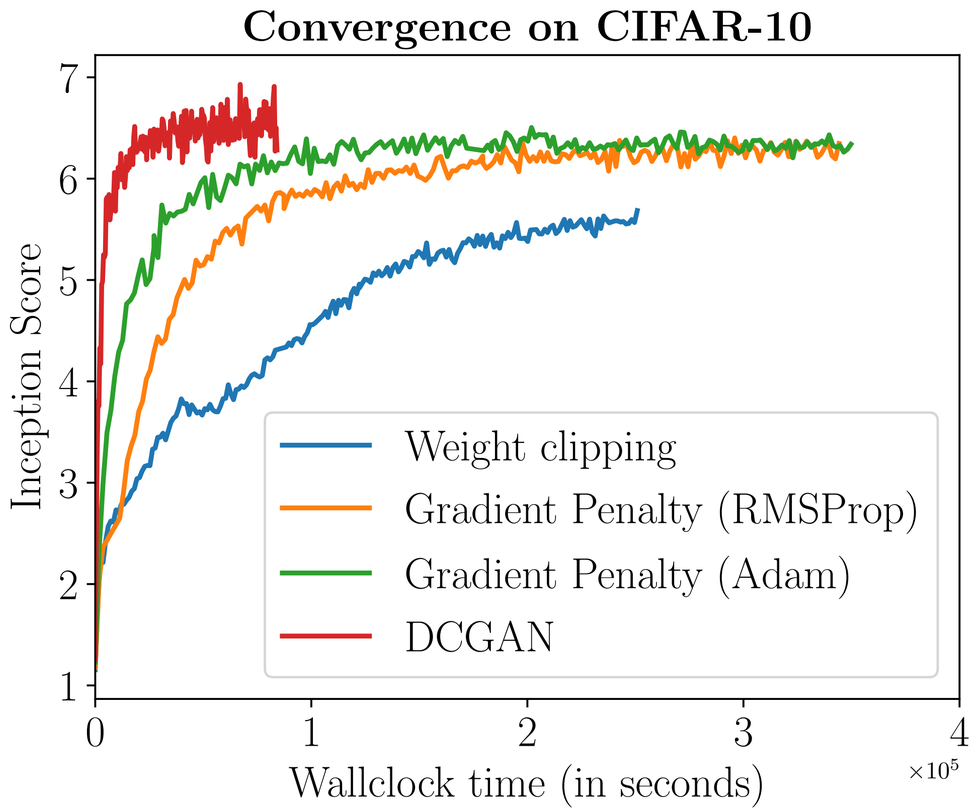
Comparison on difficult to train architectures

- Comparison based on recommended default parameter setting for each algorithm.
- WGAN-GP is more robust to variations in training setups.

DCGAN	LSGAN	WGAN (clipping)	WGAN-GP (ours)
Gated multiplicative nonlinearities everywhere in G and D			
			
$tanh$ nonlinearities everywhere in G and D			
			
101-layer ResNet G and D			
			

WGAN with Gradient Penalty





But what about inference…

- How can we use generative models?
    - GANs can generate content, but somethings you want to make inference about observed data.
- Can we incorporate an inference mechanism into GANs?
- Can we learn an inference mechanisms using an adversarial training paradigm?

Two papers, one model

- **ALI**: Vincent Dumoulin, Ishmael Belghazi, Olivier Mastropietro Ben Poole, Alex Lamb, Martin Arjovsky (2016) *ADVERSARIALLY LEARNED INFERENCE*, arXiv:1606.00704

- **BiGAN**: Donahue, Krähenbühl and Darrell (2016), *ADVERSARIAL FEATURE LEARNING*, arXiv:1605.09782

- But also showing results on Hierarchical ALI by Ishmael Belghazi, Sai Rajeshwar, Olivier Mastropietro and Negar Rostamzadeh

Adversarially learned inference: Main idea

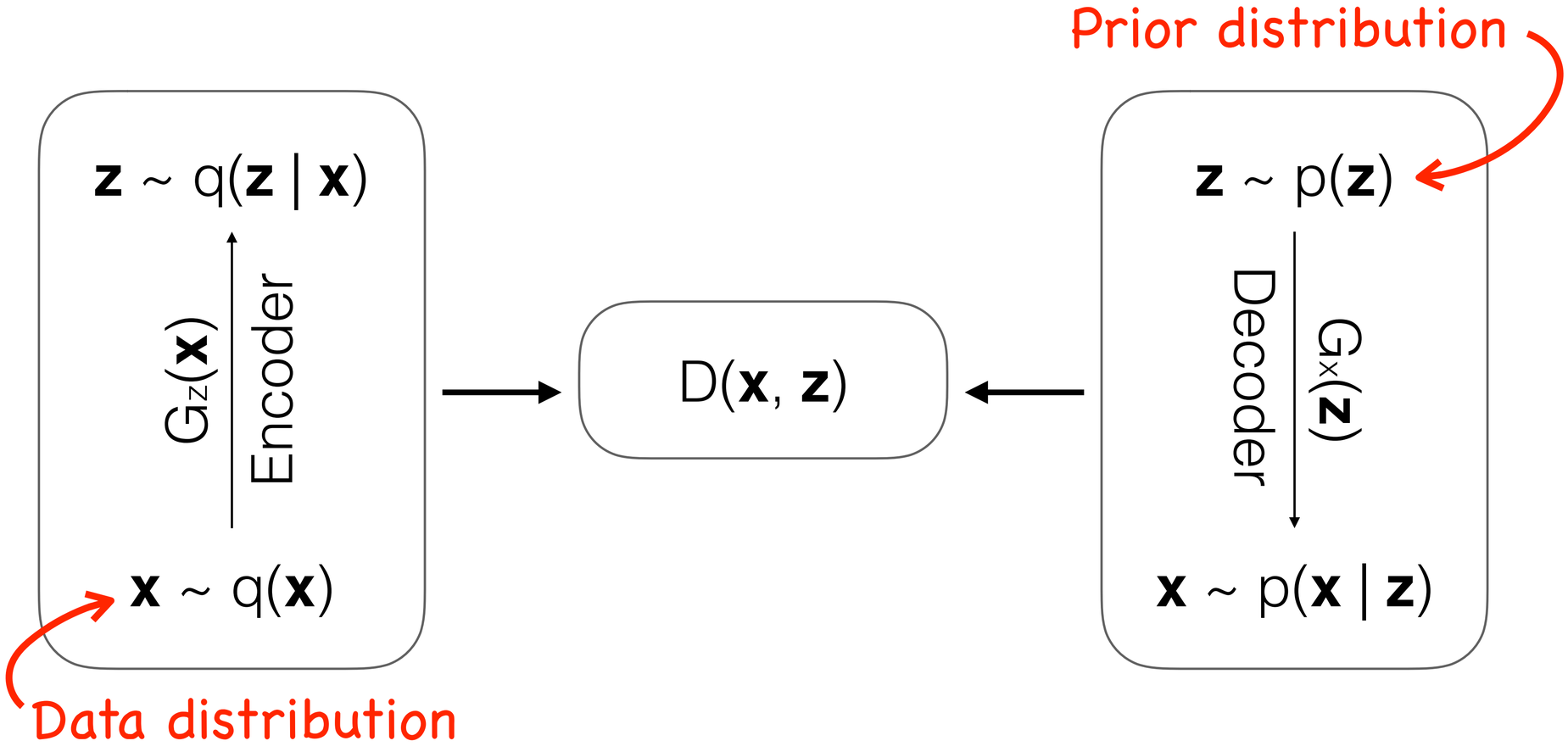
- Cast the learning of both an inference model (*encoder*) and a generative model (*decoder*) in a GAN-like adversarial framework.
- Discriminator is trained to discriminate between *joint* samples ($\mathbf{x}$, $\mathbf{z}$) from:
  - E*ncoder distribution* $q(\mathbf{x}, \mathbf{z}) = q(\mathbf{x})\, q(\mathbf{z} \mid \mathbf{x})$, or
  - D*ecoder distribution* $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})\, p(\mathbf{x} \mid \mathbf{z})$.

Data distribution

Prior distribution

- Generator learns conditionals $q(\mathbf{z} \mid \mathbf{x})$ and $p(\mathbf{x} \mid \mathbf{z})$ to fool the discriminator.

ALI: model diagram



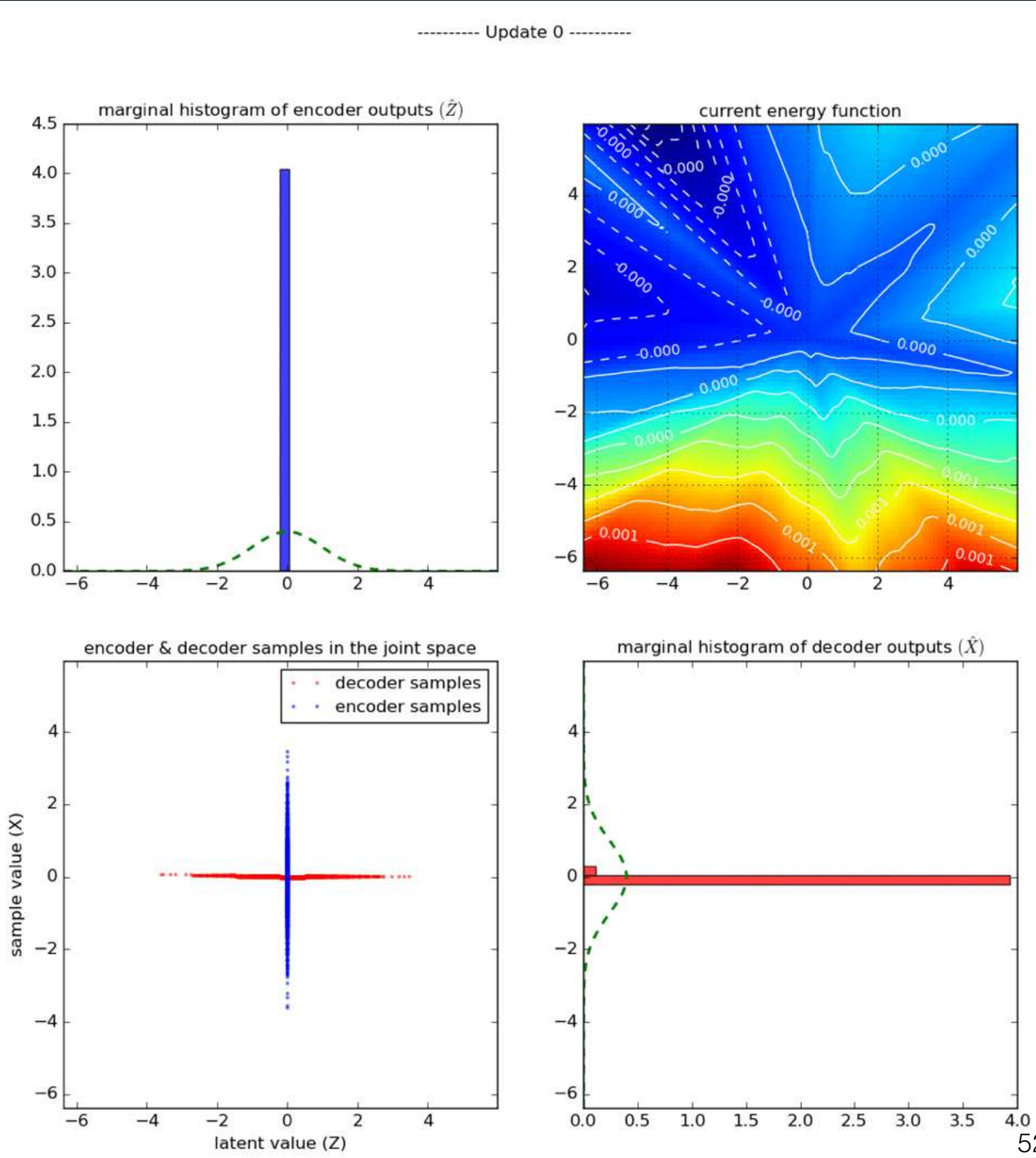
Toy Example

- Learning the Identity function:

  Encoder: $X \sim N(0,1)$

  Decoder: $Z \sim N(0,1)$

Zihang Dai



Theoretical properties

In analogy with GAN, under an ideal discriminator, the generator minimizes the Jensen-Shannon divergence between p(**x**, **z**) and q(**x**, **z**).

BiGAN: Encoder & Decoder are Inverses

Donahue, Krähenbühl and Darrell (2016)

- Donahue, Krähenbühl and Darrell (2016), *ADVERSARIAL FEATURE LEARNING*:
  - In the case of a deterministic encoder & decoder, in order to “fool” an ideal discriminator, the encoder and decoder must invert each other.

BiGAN: Encoder & Decoder are Inverses

Donahue, Krähenbühl and Darrell (2016)

- **Intuition**: Discriminator input pair $(\boldsymbol{x}, \boldsymbol{z})$ must satisfy at least one of the following two properties:

(a) $\boldsymbol{x} \in \text{supp}\left(p_{data}(\boldsymbol{x})\right) \ \wedge \ G_z(\boldsymbol{x}) = \boldsymbol{z}$

(b) $\boldsymbol{z} \in \text{supp}\left(p_{prior}(\boldsymbol{z})\right) \ \wedge \ G_x(\boldsymbol{z}) = \boldsymbol{x}$

- If only one of these properties is satisfied, a perfect discriminator can infer the source of $(\boldsymbol{x}, \boldsymbol{z})$ with certainty.

- Therefore, in order to fool an ideal discriminator, the encoder $G_z(\boldsymbol{x})$ and decoder $G_x(\boldsymbol{z})$ must satisfy both (a) and (b) at $(\boldsymbol{x}, \boldsymbol{z})$

SVHN

Samples

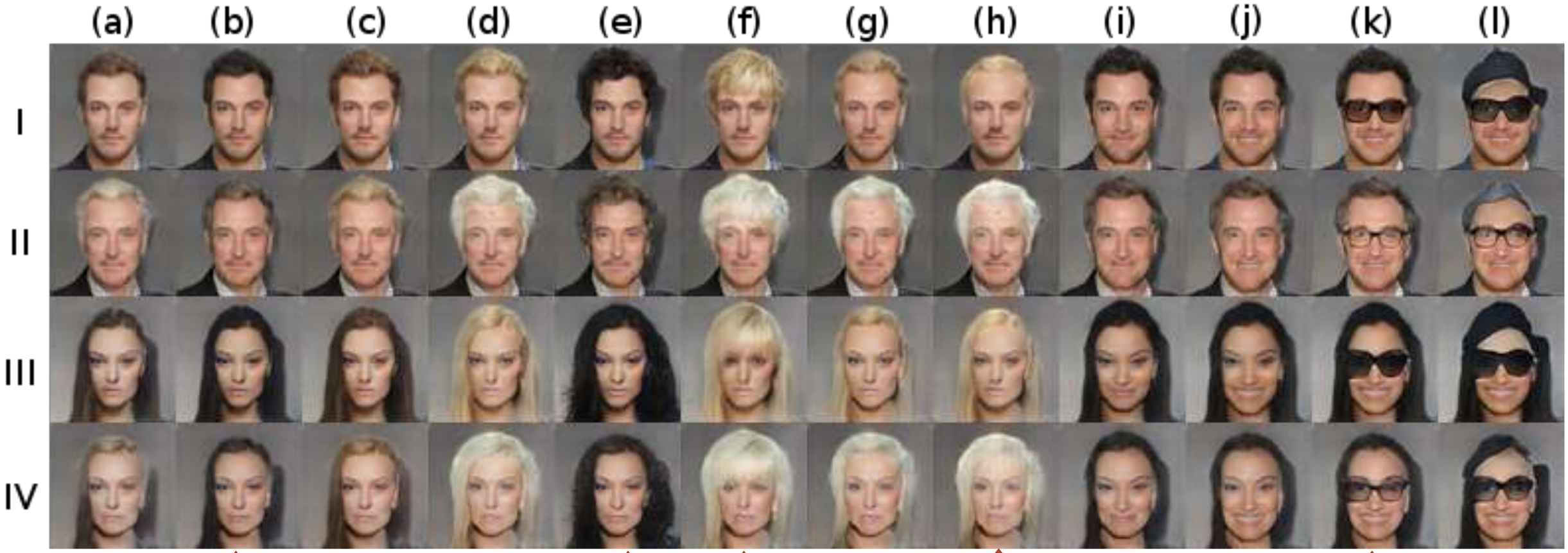
Reconstructions

CelebA face dataset

Samples

Reconstructions

Conditional generation: CelebA



Semi-supervised experiments

Table 1: SVHN test set missclassification rate

Model	Misclassification rate
VAE (M1 + M2) (Kingma et al., 2014)	36.02
SWWAE with dropout (Zhao et al., 2015)	23.56
DCGAN + L2-SVM (Radford et al., 2015)	22.18
SDGM (Maaløe et al., 2016)	16.61
GAN (feature matching) (Salimans et al., 2016)	$\mathbf{8.11 \pm 1.3}$
ALI (ours, L2-SVM)	19.14 ± 0.50
ALI (ours, no feature matching)	$\mathbf{7.42 \pm 0.65}$

Table 2: CIFAR10 test set missclassification rate for semi-supervised learning using different numbers of trained labeled examples. For ALI, error bars correspond to 3 times the standard deviation.

Number of labeled examples	1000	2000	4000	8000
Model	Misclassification rate			
Ladder network (Rasmus et al., 2015)			20.40	
CatGAN (Springenberg, 2015)			19.58	
GAN (feature matching) (Salimans et al., 2016)	$\mathbf{21.83 \pm 2.01}$	$\mathbf{19.61 \pm 2.09}$	$\mathbf{18.63 \pm 2.32}$	$\mathbf{17.72 \pm 1.82}$
ALI (ours, no feature matching)	$\mathbf{19.98 \pm 0.89}$	$\mathbf{19.09 \pm 0.44}$	$\mathbf{17.99 \pm 1.62}$	$\mathbf{17.05 \pm 1.49}$

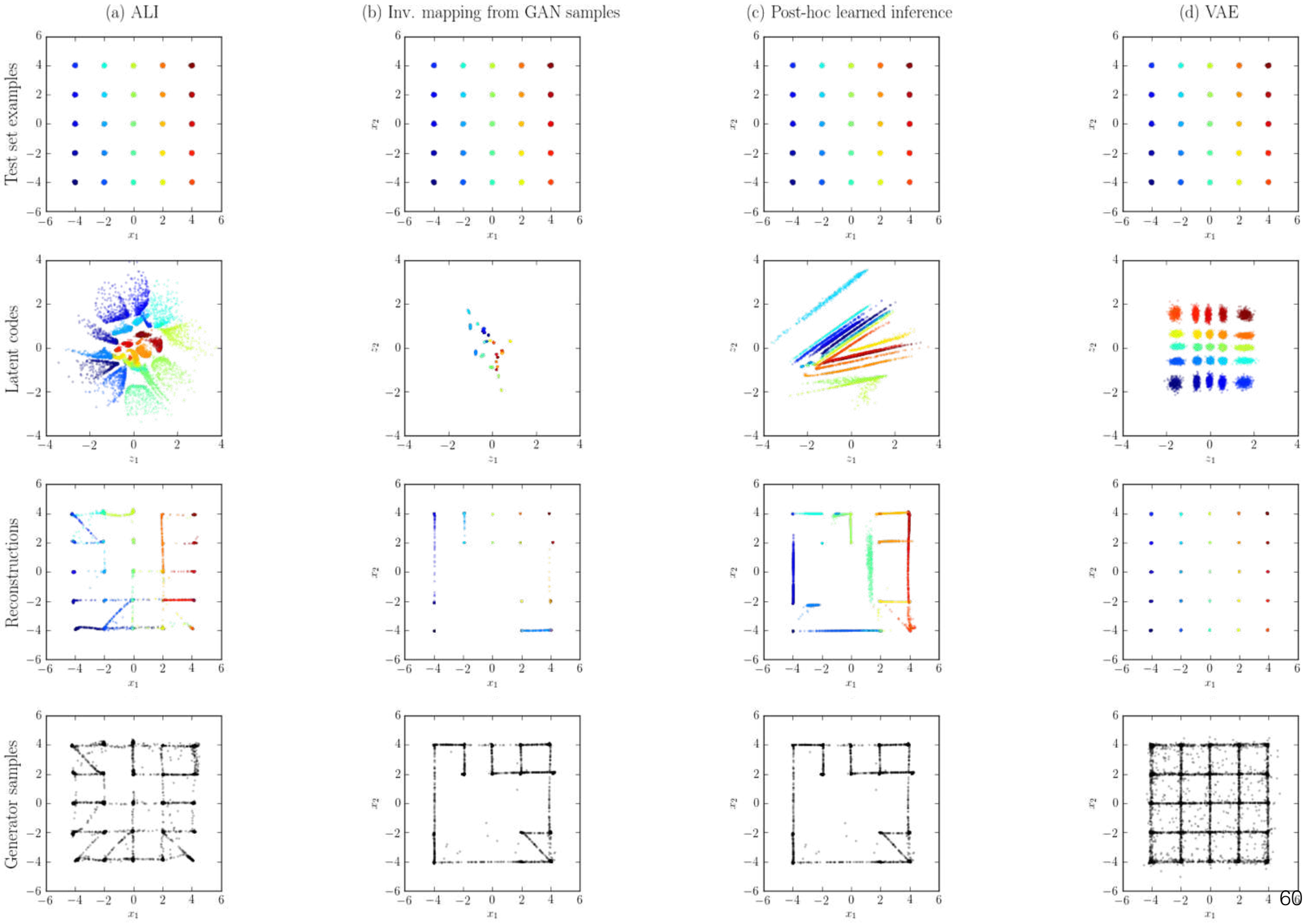
Alternative inference mechanisms:

(a) ALI (ours)

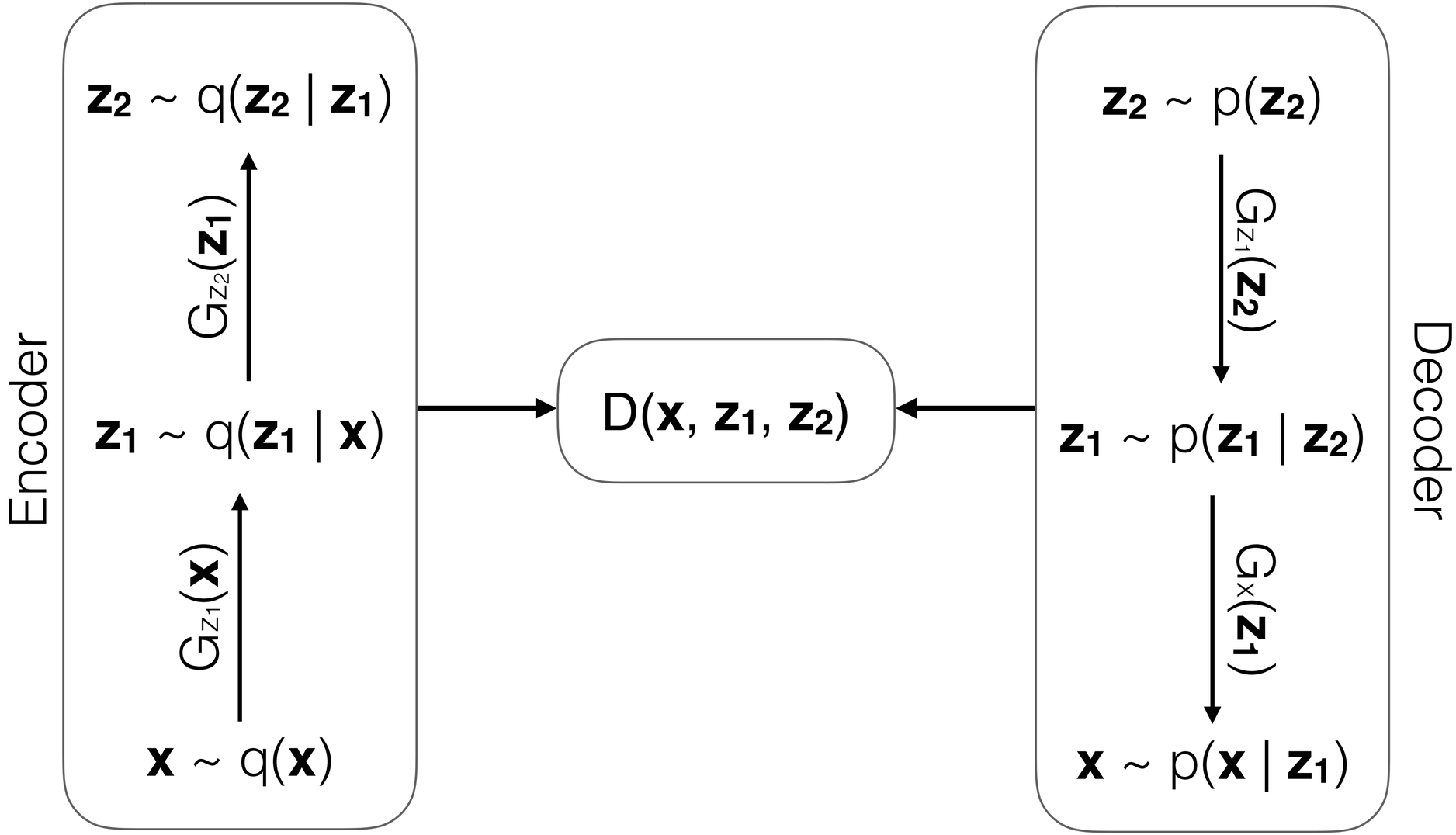
(b) Learn encoder via z reconstruction

(c) Post hoc encoder learning (ALI-style)

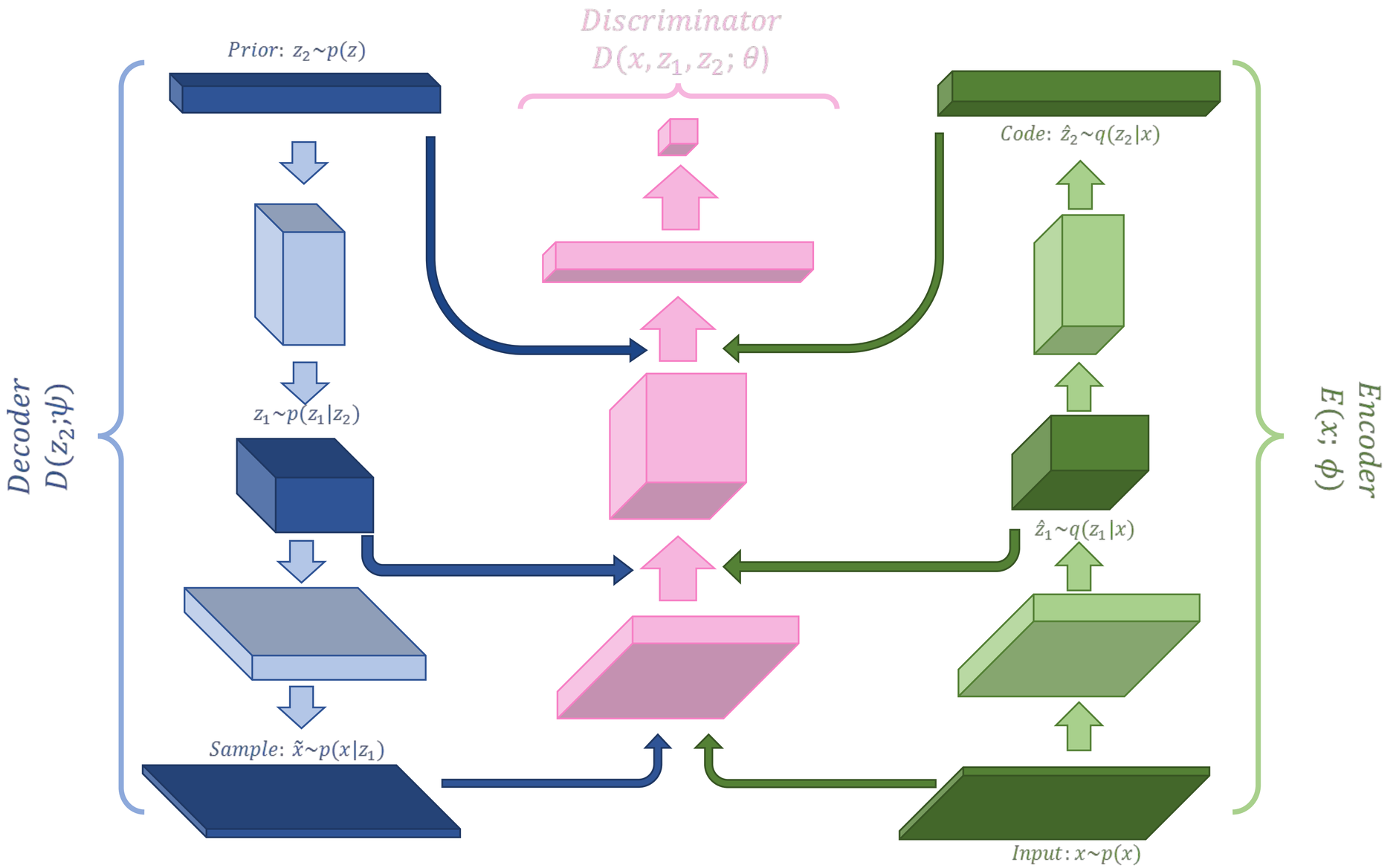
(d) Variational Autoencoder (VAE)



Hierarchical ALI: model diagram

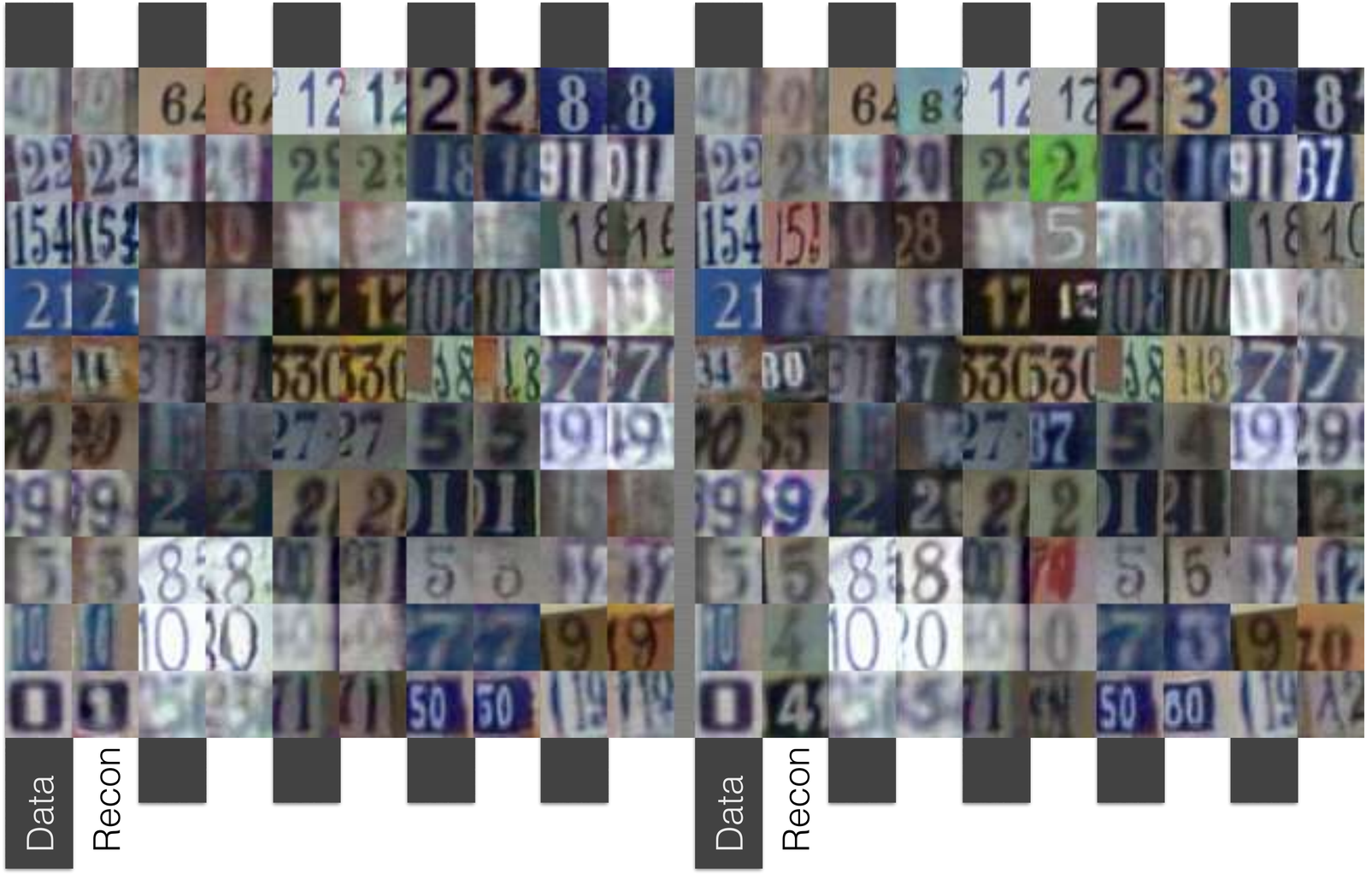


Hierarchical ALI: model diagram



Hierarchical ALI: SVHN

Model samples

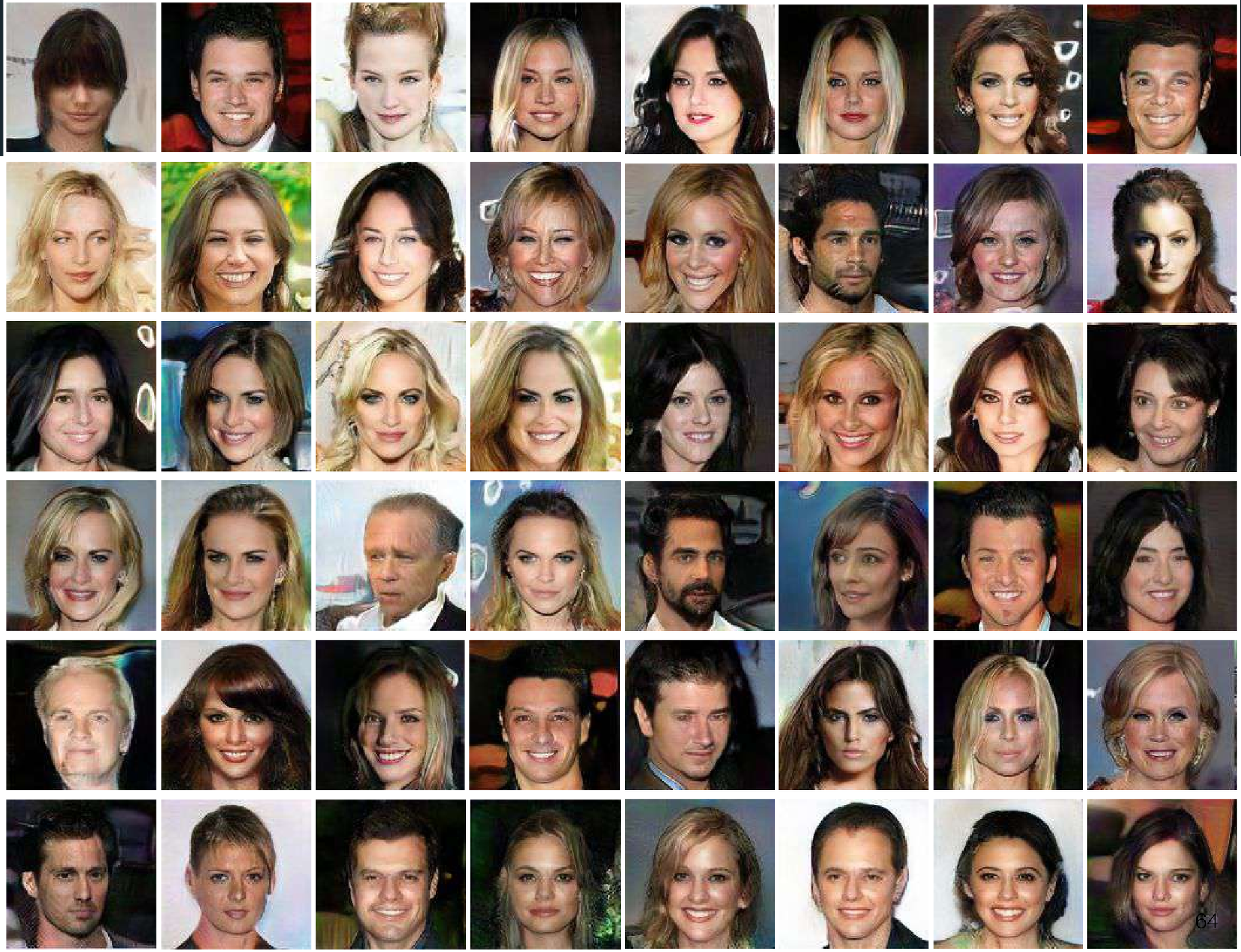


Reconstructions given z_1, z_2

Reconstructions given z_2

Hierarchical ALI

CelebA-128X128



Model samples

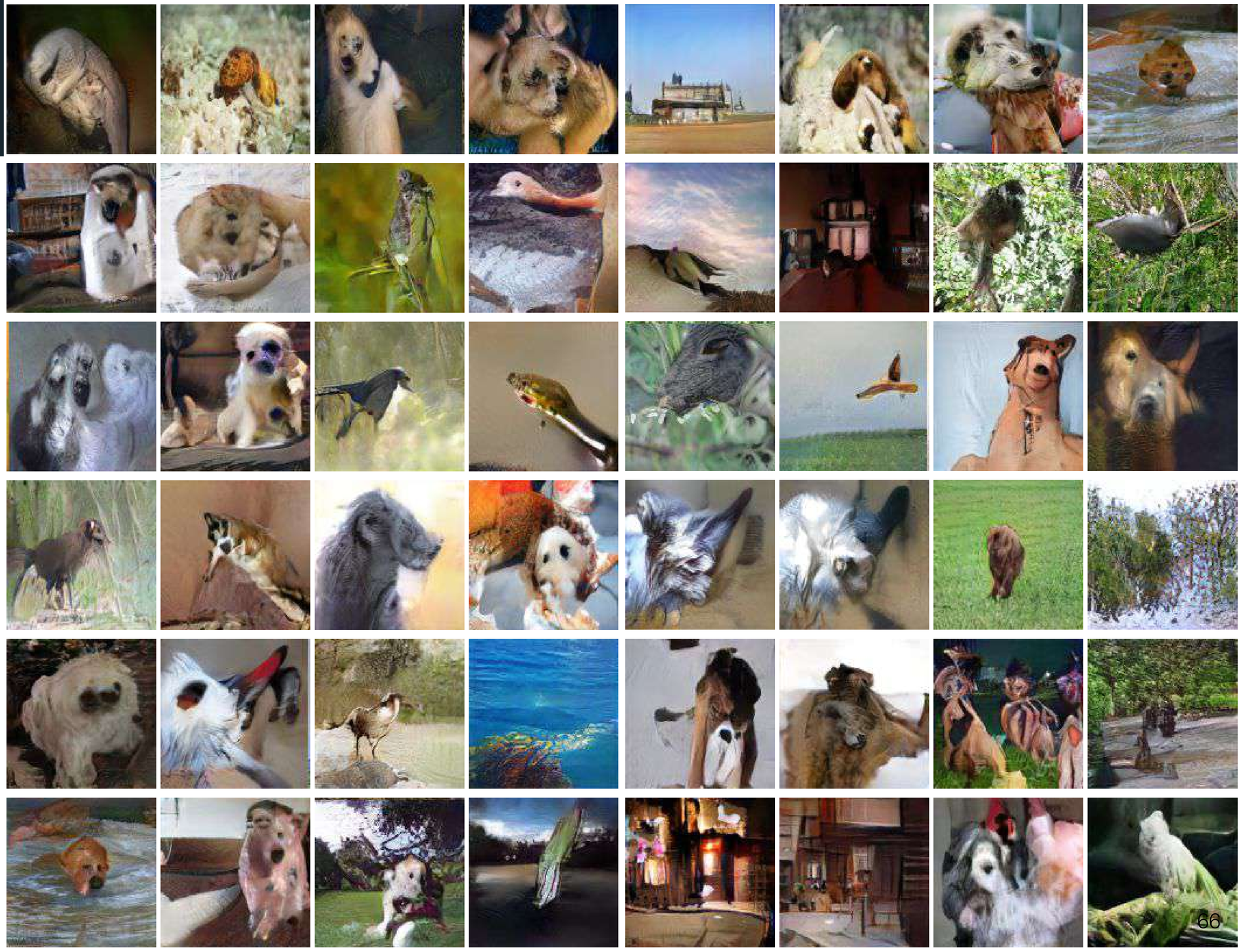
Hierarchical ALI: CelebA-128x128

Reconstructions given z_1, z_2

Reconstructions given z_2

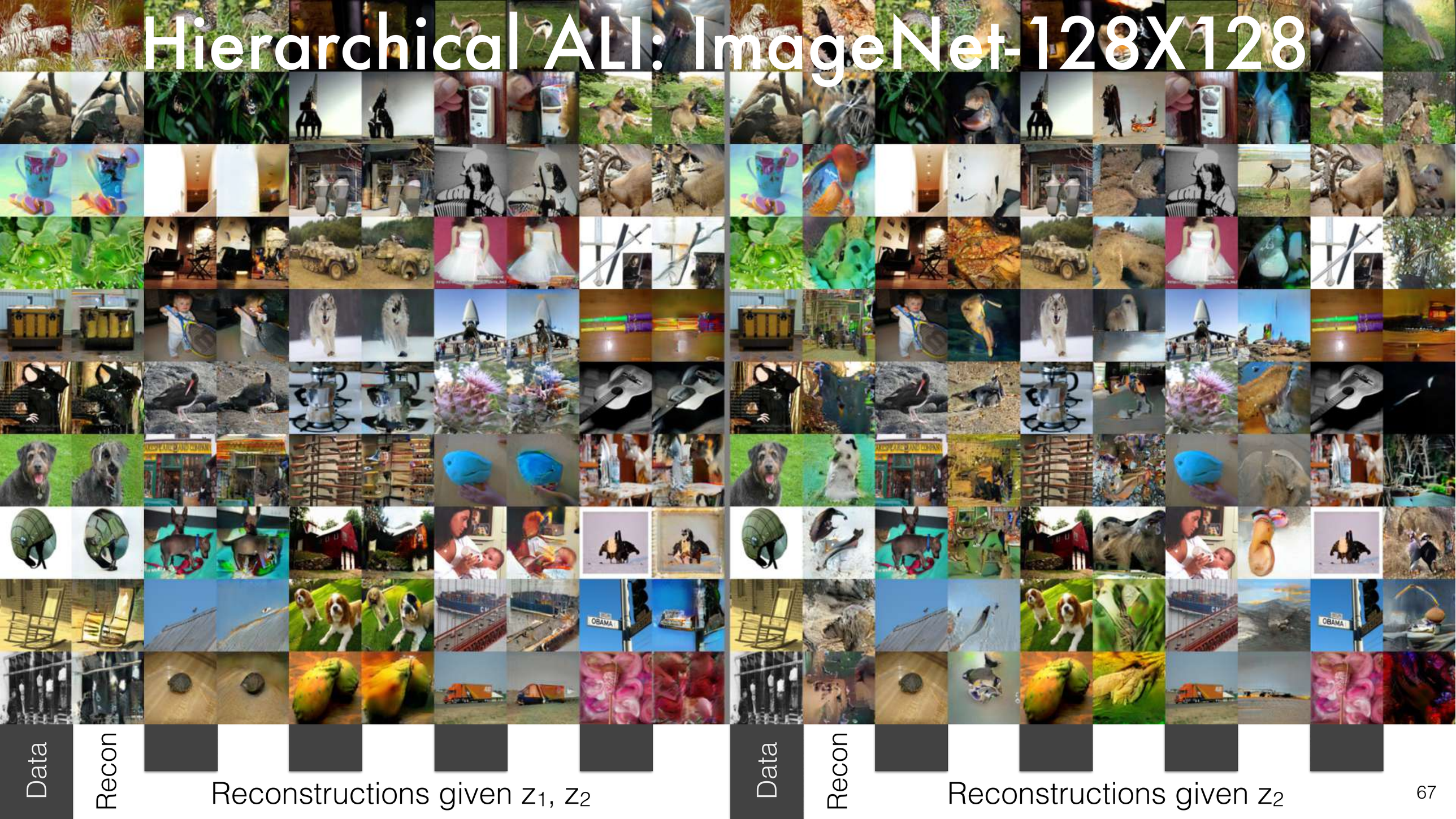
Hierarchical ALI

Unconditional ImageNet-128x128



Model samples

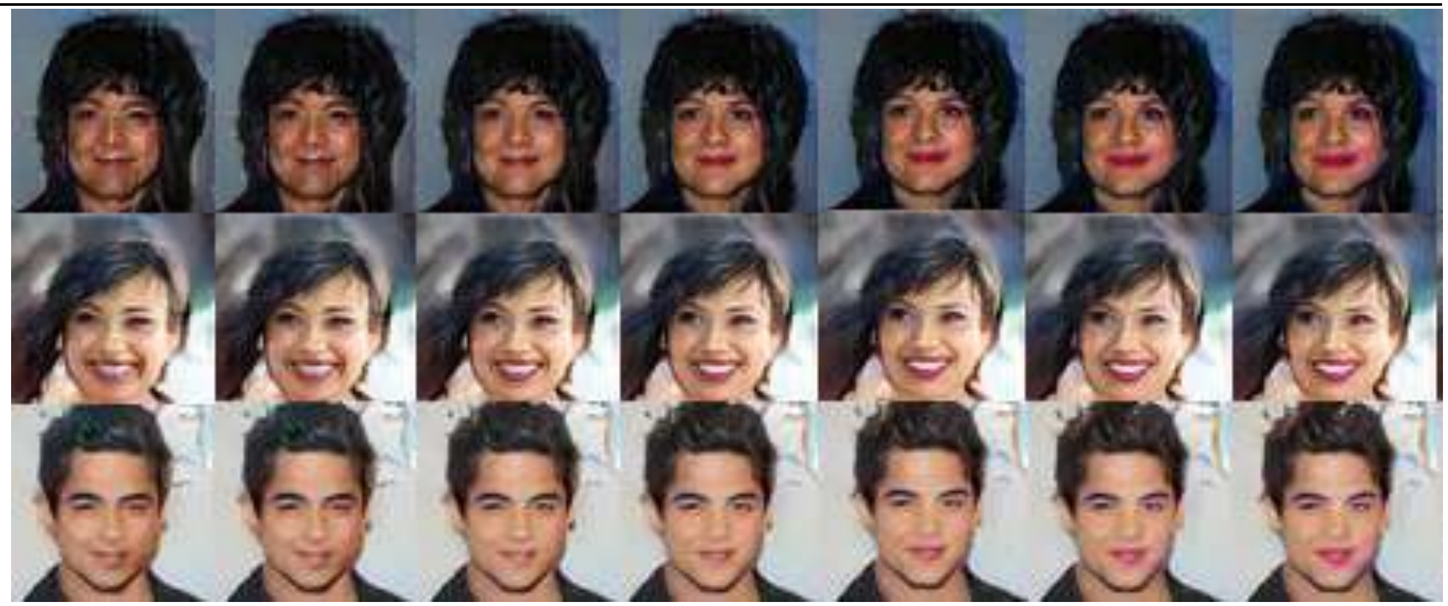
Hierarchical ALI: ImageNet-128X128



Low-level latent variable manipulation

Image saturation:

Lipstick:



High-level latent variable manipulation

Gender:

Age:

Orientation:

Interpolating from true images

True images from CelebA testset.

cycleGAN: Adversarial training of domain transformations

(Zhu et al. ICCV 2017)

MILA

- CycleGAN learns transformations across domains with unpaired data.
- Combines GAN loss with “cycle-consistency loss”: L1 reconstruction.

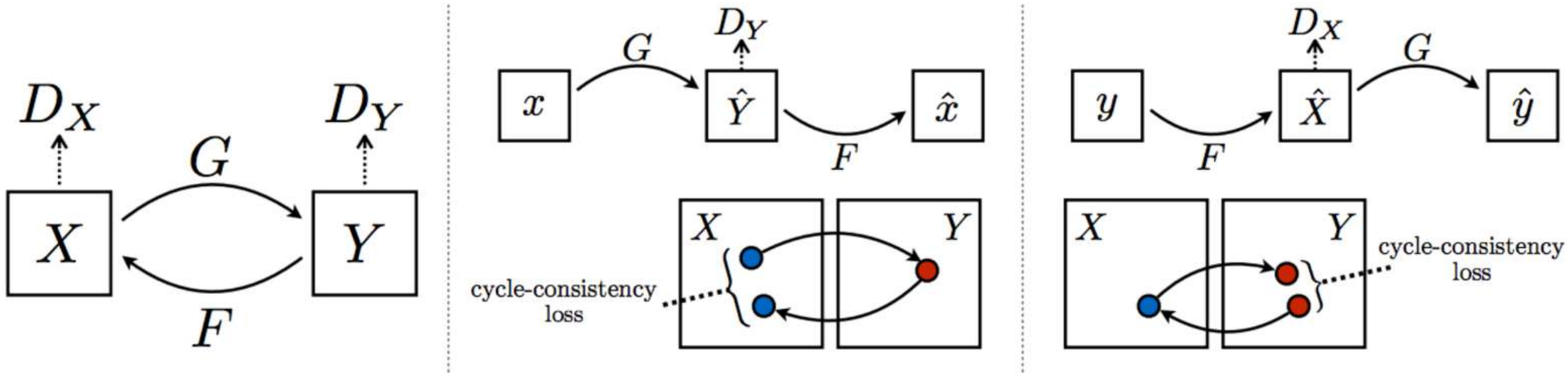


Image credits: Jun-Yan Zhu*, Taesung Park*, Phillip Isola, and Alexei A. Efros. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks", in IEEE International Conference on Computer Vision (ICCV), 2017.

CycleGAN for unpaired data

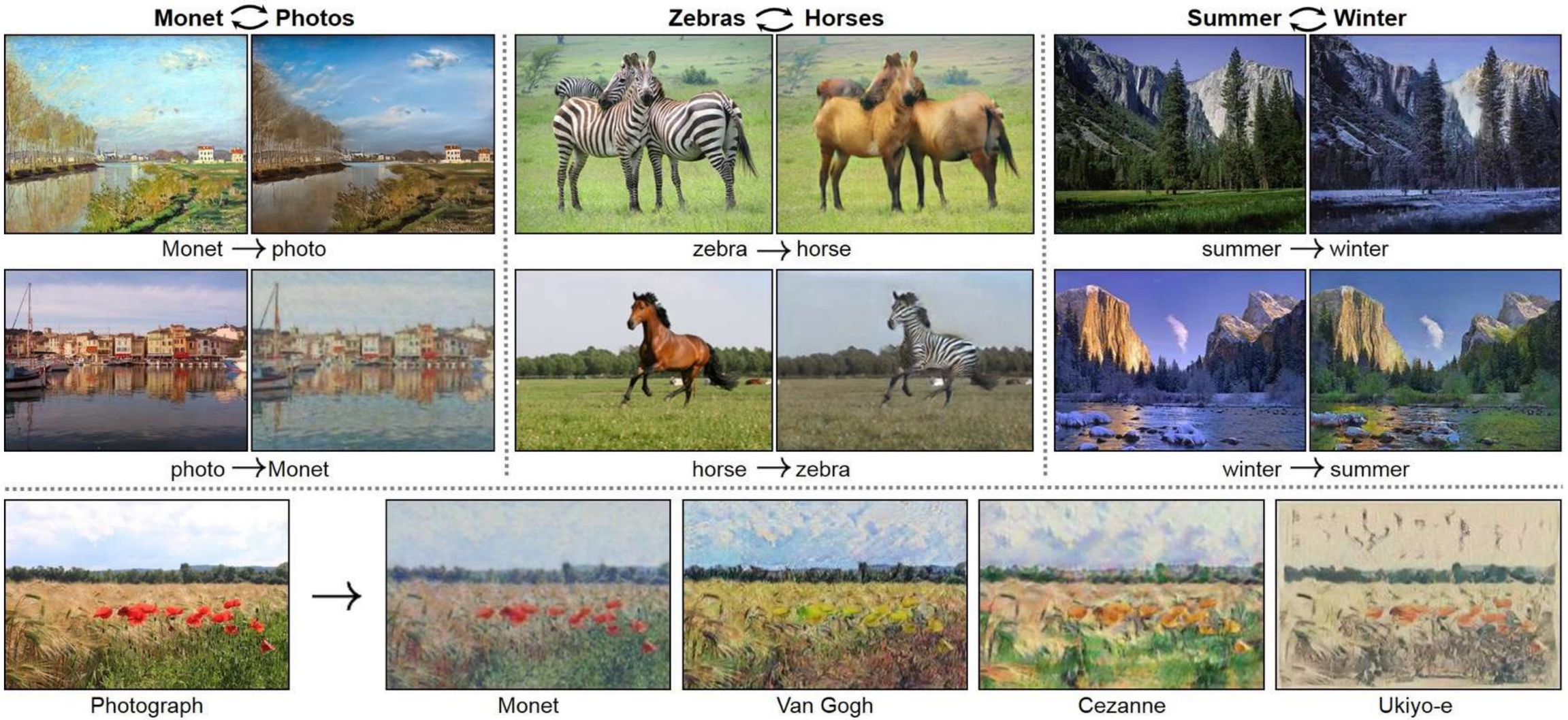


Image credits: Jun-Yan Zhu*, Taesung Park*, Phillip Isola, and Alexei A. Efros. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks", in IEEE International Conference on Computer Vision (ICCV), 2017.

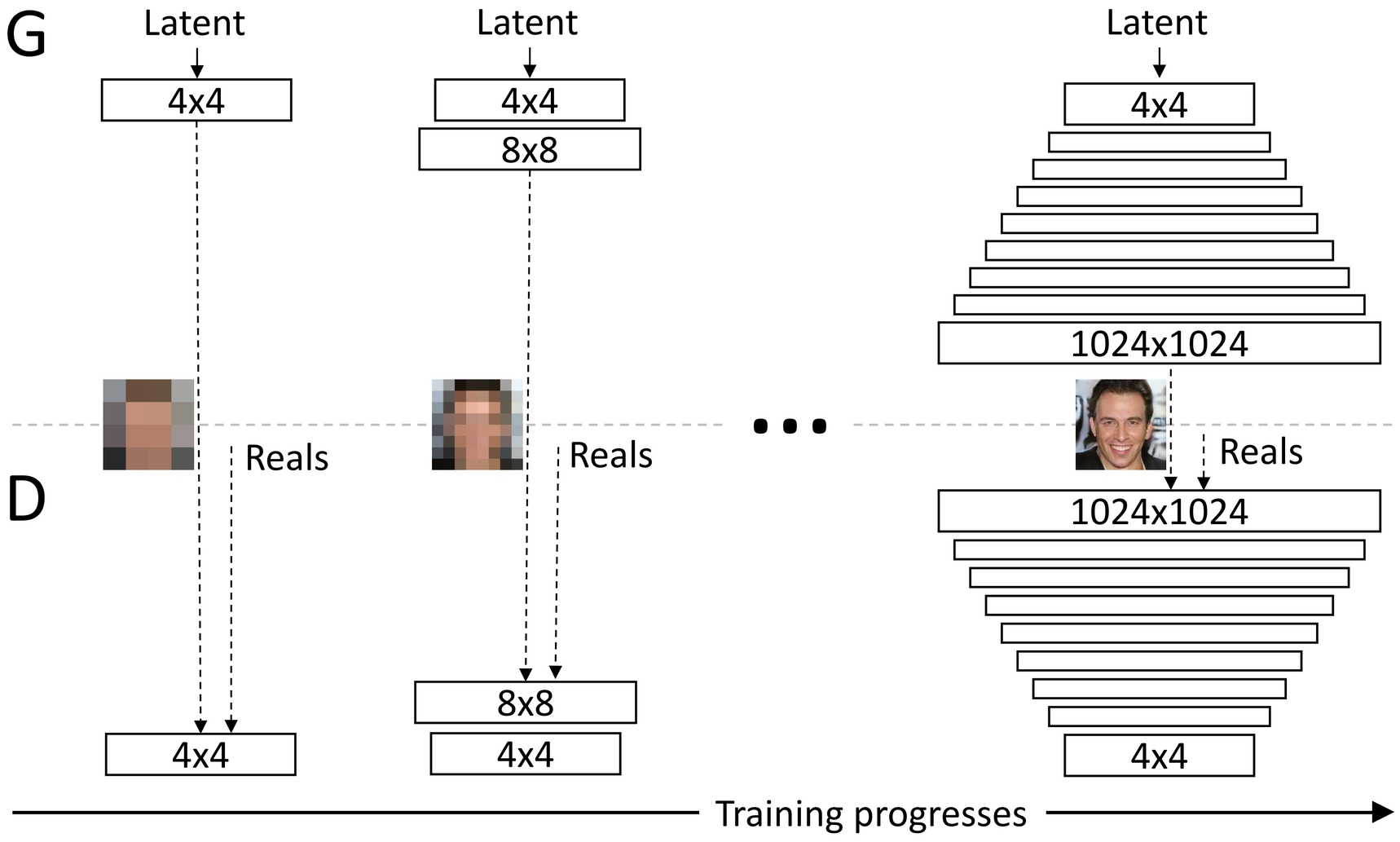
PROGRESSIVE GROWING OF GANS FOR IMPROVED QUALITY, STABILITY, AND VARIATION (Kerras et al. from NVIDIA, 2017)

MILA

- Recent work from NVIDIA.
- Improves image quality by growing the model size throughout training.
- Samples from a model trained on the CelebA face dataset.

1024x1024 model samples

PROGRESSIVE GROWING OF GANS FOR IMPROVED QUALITY, STABILITY, AND VARIATION (Kerras et al. from NVIDIA, 2017)



PROGRESSIVE GROWING OF GANS FOR IMPROVED QUALITY, STABILITY, AND VARIATION (Kerras et al. from NVIDIA, 2017)

MILA

- Recent work from NVIDIA.
- Improves image quality by growing the model size throughout training.
- Conditional samples from a model trained on the LSUN dataset

POTTEDPLANT HORSE SOFA BUS CHURCHOUTDOOR BICYCLE TVMONITOR