MATHEMATICS OF DEEP LEARNING

JOAN BRUNA , CIMS + CDS, NYU, SPRING'18

Lecture 6: Unsupervised Learning with Geometric Priors

UNSUPERVISED LEARNING AS DATA-GENERATING MODELING

- We view the data as a sample from an underlying (and unknown) probability distribution Q defined over a Polish space $\mathcal{X}$ (complete and separable, whose topology comes from a distance function).
  - We denote by $\mathcal{P}_{\mathcal{X}}$ the space of probability measures μ defined on $(\mathcal{X}, \mathcal{U})$, $\mathcal{U}$: Borel σ-algebra generated by open sets of $\mathcal{X}$.

[Bottou et al.17]

UNSUPERVISED LEARNING AS DATA-GENERATING MODELING

- We view the data as a sample from an underlying (and unknown) probability distribution Q defined over a Polish space $\mathcal{X}$ (complete and separable, whose topology comes from a distance function).
  - We denote by $\mathcal{P}_{\mathcal{X}}$ the space of probability measures μ defined on $(\mathcal{X}, \mathcal{U})$, $\mathcal{U}$: Borel σ-algebra generated by open sets of $\mathcal{X}$.
- We need a measure to compare elements of $\mathcal{P}_{\mathcal{X}}$:

$$(P, Q) \mapsto D(P, Q) \in [0, \infty)$$

  - D can be a distance, but also a *pseudo-distance* (does not satisfy separation) or a *divergence* (not symmetric or no triangle inequality).

[Bottou et al.17]

UNSUPERVISED LEARNING AS DATA-GENERATING MODEL

- Given a parametric family of distributions $P_\theta \in \mathcal{P}_\mathcal{X}$, goal of data modeling is

$$\min_\theta L(\theta) = D(Q, P_\theta)$$

- Similarly as in supervised learning, we need to generalize from an empirical loss:

$$\min_\theta \hat{L}(\theta) = D(\hat{Q}, P_\theta) + \mathcal{R}(\theta) \ .$$

UNSUPERVISED LEARNING AS DATA-GENERATING MODEL

- Given a parametric family of distributions $P_\theta \in \mathcal{P}_\mathcal{X}$, goal of data modeling is

$$\min_\theta L(\theta) = D(Q, P_\theta)$$

- Similarly as in supervised learning, we need to generalize from an empirical loss:

$$\min_\theta \hat{L}(\theta) = D(\hat{Q}, P_\theta) + \mathcal{R}(\theta) \ .$$

- Which criteria D?
- Which model P_θ?

LATENT GRAPHICAL MODELS

- We assume first that both distributions admit a density with respect to a base measure μ :

$$Q(A) = \int_A q(x) d\mu(x) \ , \ P_\theta(A) = \int_A p_\theta(x) d\mu(x) \ .$$

LATENT GRAPHICAL MODELS

- We assume first that both distributions admit a density with respect to a base measure μ :

$$Q(A) = \int_A q(x) d\mu(x) \ , \ P_\theta(A) = \int_A p_\theta(x) d\mu(x) \ .$$

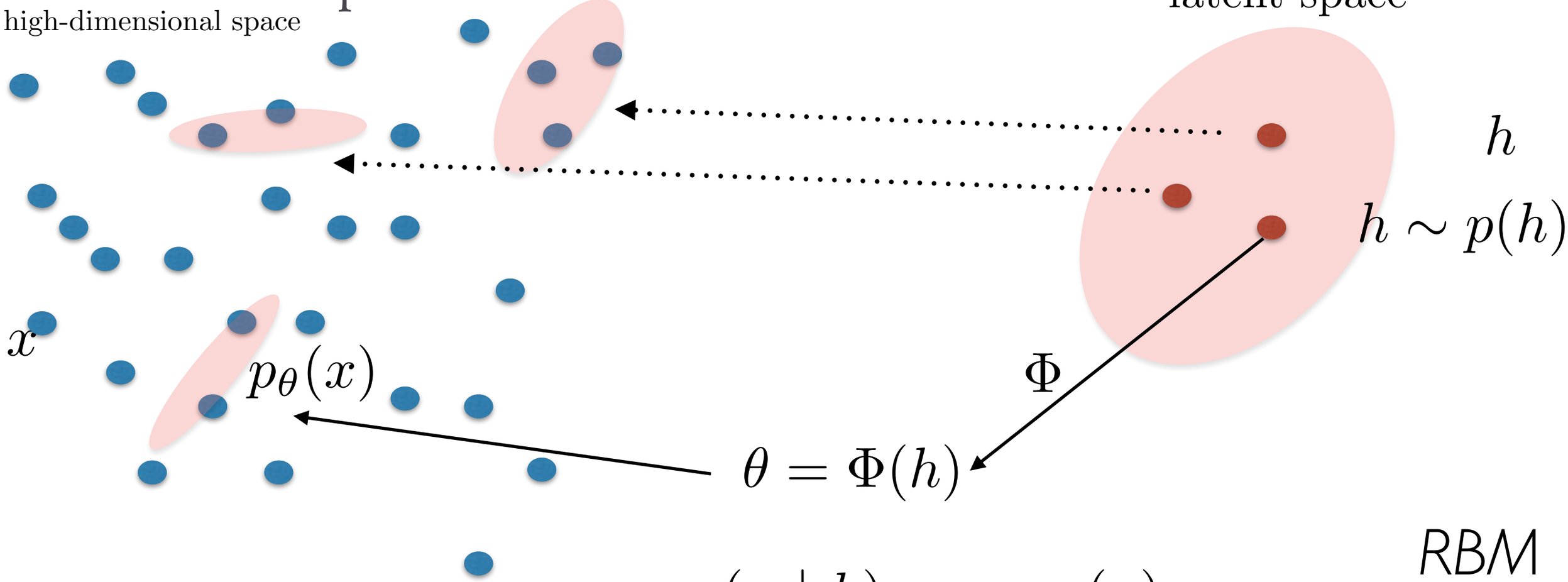
- A popular choice of criteria is the *Kullback-Leibler divergence*:

$$\begin{aligned} D_{KL}(Q, P_\theta) &= \mathbb{E}_{P_\theta} \left\{ f[q(x) p_\theta^{-1}(x)] \right\} \ , \ f(t) = t \log t \\ &= \int q(x) \log \left(\frac{q(x)}{p_\theta(x)} \right) d\mu(x) \ . \end{aligned}$$

- Learning is equivalent to $\max_\theta \mathbb{E}_Q \log p_\theta(x)$

LATENT GRAPHICAL MODELS

➤ Latent Graphical Models or *Mixtures*.



$$p(x \mid h) = p_{\Phi(h)}(x)$$

$$p(x) = \int p(x, h)dh = \int p(x \mid h)p(h)dh$$

RBM
DBN
DBM
VAE
...

Model: additive combination of simple parametric models

THE EM ALGORITHM

➤It is designed to find MLE solutions of latent variable models.

➤In general, we have log-likelihoods of the form

$$\log p(X \mid \theta) = \log \left(\sum_{Z} p(X, Z \mid \theta) \right) \quad , \quad \begin{array}{l} \theta = \text{model parameters} \ . \\ Z = \text{latent variables} \end{array}$$

THE EM ALGORITHM

➤It is designed to find MLE solutions of latent variable models.

➤In general, we have log-likelihoods of the form

$$\log p(X \mid \theta) = \log \left(\sum_Z p(X, Z \mid \theta) \right) \quad , \quad \begin{array}{l} \theta = \text{model parameters} \\ Z = \text{latent variables} \end{array} .$$

➤Using current parameters θ_{old}, we compute the expected total likelihood of the model (E-step):

$$\mathcal{Q}(\theta, \theta_{old}) = \mathbb{E}_{Z \sim p(Z \mid X, \theta_{old})} \log p(X, Z \mid \theta)$$

➤Then we update the parameters to maximize this likelihood:

$$\theta_{new} \in \arg\max_{\theta} \mathcal{Q}(\theta, \theta_{old}) \ .$$

EM AND VARIATIONAL BOUND

➤Q: Does this algorithm monotonically improve the likelihood?

➤Assume for now that latent variables are discrete.

EM AND VARIATIONAL BOUND

➤Q: Does this algorithm monotonically improve the likelihood?

➤Assume for now that latent variables are discrete.

➤For any positive density $q(Z)$over latent variables, we have

$$\log p(X \mid \theta) = \log \left(\sum_Z p(X, Z \mid \theta) \right) = \log \left(\sum_Z q(Z) \frac{p(X, Z \mid \theta)}{q(Z)} \right)$$

$$\geq \sum_Z q(Z) \log \left(\frac{p(X, Z \mid \theta)}{q(Z)} \right) = \mathcal{L}(q, \theta) \ .$$

VARIATIONAL BOUND

➤We can express the variational lower bound as

$$\begin{aligned}\mathcal{L}(q,\theta) &= \mathbb{E}_{q(Z)}\left[\log p(X,Z \mid \theta)\right] - \mathbb{E}_{q(Z)} \log q(Z) \\ &= \mathbb{E}_{q(Z)}\left[\log p(X,Z \mid \theta)\right] + H(q)\ .\end{aligned}$$

$H(q)$: Entropy of $q(Z)$.

VARIATIONAL BOUND

➤We can express the variational lower bound as

$$\begin{aligned}\mathcal{L}(q,\theta) &= \mathbb{E}_{q(Z)}\left[\log p(X,Z \mid \theta)\right] - \mathbb{E}_{q(Z)} \log q(Z) \\ &= \mathbb{E}_{q(Z)}\left[\log p(X,Z \mid \theta)\right] + H(q)\ .\end{aligned}$$

$H(q)$: Entropy of $q(Z)$.

➤Also, we have

$$\log p(X \mid \theta) = \mathcal{L}(q,\theta) + KL(q(z)||p(z \mid x,\theta))\ ,\ \text{where}$$

$$KL(q||p) = -\sum_z q(z) \log\left(\frac{p(z)}{q(z)}\right)$$

is the Kullback-Leibler divergence.

VARIATIONAL BOUND

➤Thus, the divergence $KL(q||p)$ measures how far our variational approximation $q(z)$ is from the true posterior, and directly controls the bound on the log-likelihood.

➤Using $\log p(X \mid \theta) = \mathcal{L}(q, \theta) + KL(q(z)||p(z \mid x, \theta))$

➤E-step: maximize lower bound $\mathcal{L}(q, \theta)$ with respect to q, holding parameters fixed.

➤M-step: maximize lower bound $\mathcal{L}(q, \theta)$ with respect to parameters, holding q fixed.

CORRECTNESS OF EM

- Suppose current parameter value is $\theta^{(n)}$.
- Consider the variational bound by picking $q(z) = p(z|x, \theta^{(n)})$.

$$\begin{aligned}
\log p(X|\theta) - \log p(X|\theta^{(n)}) &\geq \mathcal{L}(p(z|x, \theta^{(n)}), \theta) \\
&= \sum_z p(z|x, \theta^{(n)}) \log \left(\frac{p(x|z, \theta) p(z|\theta)}{p(z|x, \theta^{(n)}) p(x|\theta^{(n)}} \right) \\
&= \Delta(\theta|\theta^{(n)}).
\end{aligned}$$

with $\Delta(\theta^{(n)}|\theta^{(n)}) = 0$.

CORRECTNESS OF EM

- Thus $\theta^{(n+1)} = \arg\max_{\theta} \Delta(\theta|\theta^{(n)})$

$$
\begin{aligned}
\theta^{(n+1)} &= \arg\max_{\theta} \Delta(\theta|\theta^{(n)}) \\
&= \arg\max_{\theta} \sum_{z} p(z|x, \theta^{(n)}) \log\{(p(x|z, \theta)p(z|\theta)\} \\
&= \arg\max_{\theta} \sum_{z} p(z|x, \theta^{(n)}) \log\{(p(x, z|\theta)\}\} \\
&= \arg\max_{\theta} \mathbb{E}_{z\sim p(z|x,\theta^{(n)})} \log p(X, Z|\theta) \ .
\end{aligned}
$$

APPROXIMATE POSTERIOR INFERENCE

➤For most models, the posterior is analytically intractable:

$$p(z \mid x) = \frac{p(x \mid z)p(z)}{\int p(x \mid z')p(z')dz'}$$

➤***Variational Bayesian Inference:*** consider a parametric family of approximations $q(z \mid \beta)$ and optimize variational lower bound with respect to the variational parameters β .

MEAN FIELD VARIATIONAL BAYES

➤Joint likelihood of observed and latent variables:

$$p(X, Z \mid \theta) \qquad \theta\text{: generative model parameters}$$

➤Let us consider a posterior approximation $q(z|\beta)$ of the form

$$q(z \mid \beta) = \prod_i q_i(z_i \mid \beta_i) \qquad \beta\text{: Variational parameters}$$

- Mean-field approximation: we model hidden variables as being independent.

MEAN FIELD VARIATIONAL BAYES

➤Joint likelihood of observed and latent variables:

$$p(X, Z \mid \theta) \qquad \theta\text{: generative model parameters}$$

➤Let us consider a posterior approximation $q(z|\beta)$ of the form

$$q(z \mid \beta) = \prod_i q_i(z_i \mid \beta_i) \qquad \beta\text{: Variational parameters}$$

- Mean-field approximation: we model hidden variables as being independent.

➤Corresponding lower-bound is given by

$$\log p(X \mid \theta) \geq \int q(z \mid \beta) \log \frac{p(x, z \mid \theta)}{q(z \mid \beta)} dz = \mathbb{E}_{q(z|\beta)}\{\log(p(X, Z \mid \theta)\} + H(q(z \mid \beta))$$

MEAN FIELD VARIATIONAL BAYES

➤**Goal**: optimize lower-bound with respect to variational parameters.

MEAN FIELD VARIATIONAL BAYES

➤**Goal**: optimize lower-bound with respect to variational parameters.

➤As we have seen, this is equivalent to minimizing the divergence between true and approximate posterior:

$$\log p(X \mid \theta) = \mathcal{L}(\theta, \beta) + D_{KL}(q_\beta(z) || p(z|x, \theta))$$

MEAN FIELD VARIATIONAL BAYES

➤**Goal**: optimize lower-bound with respect to variational parameters.

➤As we have seen, this is equivalent to minimizing the divergence between true and approximate posterior:

$$\log p(X \mid \theta) = \mathcal{L}(\theta, \beta) + D_{KL}(q_\beta(z)||p(z|x, \theta))$$

➤If $q(z \mid \beta)$ is a factorial distribution, the entropy term is tractable:

$$H(q(z|\beta)) = \sum_i H(q_i(z_i|\beta_i))$$

➤Problematic term: $\nabla_\beta \mathbb{E}_{q(z|\beta)} \log p(X, Z|\theta)$

MEAN FIELD VARIATIONAL BAYES

[Paiskey, Blei, Jordan,'12]

➤Denote $f(Z) = \log p(X, Z|\theta)$

➤Then

$$\begin{aligned}
\nabla_\beta \mathbb{E}_{q(z|\beta)} f(Z) &= \nabla_\beta \int f(z) q(z|\beta) dz \\
&= \int f(z) \nabla_\beta q(z|\beta) dz \\
&= \int f(z) q(z|\beta) \nabla_\beta \log q(z|\beta) dz \\
&= \mathbb{E}_q \{ f(Z) \nabla_\beta \log q(z|\beta) \}
\end{aligned}$$

MEAN FIELD VARIATIONAL BAYES

[Paiskey, Blei, Jordan,'12]

➤Denote $f(Z) = \log p(X, Z|\theta)$

➤Then

$$\begin{aligned}
\nabla_\beta \mathbb{E}_{q(z|\beta)} f(Z) &= \nabla_\beta \int f(z) q(z|\beta) dz \\
&= \int f(z) \nabla_\beta q(z|\beta) dz \\
&= \int f(z) q(z|\beta) \nabla_\beta \log q(z|\beta) dz \\
&= \mathbb{E}_q \{ f(Z) \nabla_\beta \log q(z|\beta) \}
\end{aligned}$$

➤Stochastic approximation of $\nabla_\beta \mathbb{E}_{q(z|\beta)} f(Z)$

$$\approx \frac{1}{S} \sum_{s \le S, z^{(s)} \sim q(z|\beta)} \begin{matrix} \nabla_\beta \mathbb{E}_{q(z|\beta)} f(Z) \\ \vdots \\ f(z^{(s)}) \nabla_\beta \log q(z^{(s)}|\beta) \end{matrix}$$

VARIATIONAL AUTOENCODERS

[Kingma & Welling'14, Rezende et al.'14]

➤Recall the variational lower bound:

$$\log p(X \mid \theta) = \mathbb{E}_{q(z|\beta)}\{\log(p(X, Z \mid \theta)\} + H(q(z \mid \beta)) + D_{KL}(q(z|\beta)||p(z|x, \theta)$$

$$\log p(X \mid \theta) = \mathcal{L}(\theta, \beta, X) + D_{KL}(q(z|\beta)||p(z|X, \theta))$$

VARIATIONAL AUTOENCODERS

[Kingma & Welling'14, Rezende et al.'14]

➤Recall the variational lower bound:

$$\log p(X \mid \theta) = \mathbb{E}_{q(z|\beta)}\{\log(p(X, Z \mid \theta)\} + H(q(z \mid \beta)) + D_{KL}(q(z|\beta)||p(z|x, \theta)$$

$$\log p(X \mid \theta) = \mathcal{L}(\theta, \beta, X) + D_{KL}(q(z|\beta)||p(z|X, \theta))$$

➤Can we optimize jointly both generative and variational parameters efficiently?

VARIATIONAL AUTOENCODERS

[Kingma & Welling'14, Rezende et al.'14]

➤Recall the variational lower bound:

$$\log p(X \mid \theta) = \mathbb{E}_{q(z|\beta)}\{\log(p(X, Z \mid \theta)\} + H(q(z \mid \beta)) + D_{KL}(q(z|\beta)||p(z|x, \theta))$$

$$\log p(X \mid \theta) = \mathcal{L}(\theta, \beta, X) + D_{KL}(q(z|\beta)||p(z|X, \theta))$$

➤Can we optimize jointly both generative and variational parameters efficiently?

➤For appropriate posterior approximations, we can reparametrize samples as

$$Z \sim q(z|x, \beta) \Rightarrow Z \stackrel{d}{=} g_\beta(\epsilon, x) \ , \ \epsilon \sim p_0$$

$$\left(\text{e.g. } q(z|x, \beta) = \mathcal{N}(z; \mu(x), \Sigma(x)) \leftrightarrow z = \mu(x) + \Sigma(x)^{1/2}\epsilon \ , \ \epsilon \sim \mathcal{N}(0, \mathbf{1})\right)$$

VARIATIONAL AUTOENCODERS

➤It results that

$$\mathcal{L}(\theta, \beta, X) = -D_{KL}(q_\beta(z|X)||p_\theta(z)) + \mathbb{E}_{q_\beta(z|X)}\{\log p(X|z, \theta)\}$$

can be estimated via Monte-Carlo by

$$\widehat{\mathcal{L}(\theta, \beta, X)} = -D_{KL}(q_\beta(z|X)||p_\theta(z)) + \frac{1}{S}\sum_{s \leq S} \log p(X|z^{(s)}, \theta)$$

$$z^{(s)} = g_\beta(X, \epsilon^{(s)}) \text{ and } \epsilon^{(s)} \sim p_0 \ .$$

VARIATIONAL AUTOENCODERS

➤It results that

$$\mathcal{L}(\theta, \beta, X) = -D_{KL}(q_\beta(z|X)||p_\theta(z)) + \mathbb{E}_{q_\beta(z|X)}\{\log p(X|z, \theta)\}$$

can be estimated via Monte-Carlo by

$$\widehat{\mathcal{L}(\theta, \beta, X)} = -D_{KL}(q_\beta(z|X)||p_\theta(z)) + \frac{1}{S}\sum_{s \leq S} \log p(X|z^{(s)}, \theta)$$

$$z^{(s)} = g_\beta(X, \epsilon^{(s)}) \text{ and } \epsilon^{(s)} \sim p_0 \ .$$

➤First term acts as a regularizer: limits the capacity of the encoder

➤Second term is a reconstruction error.

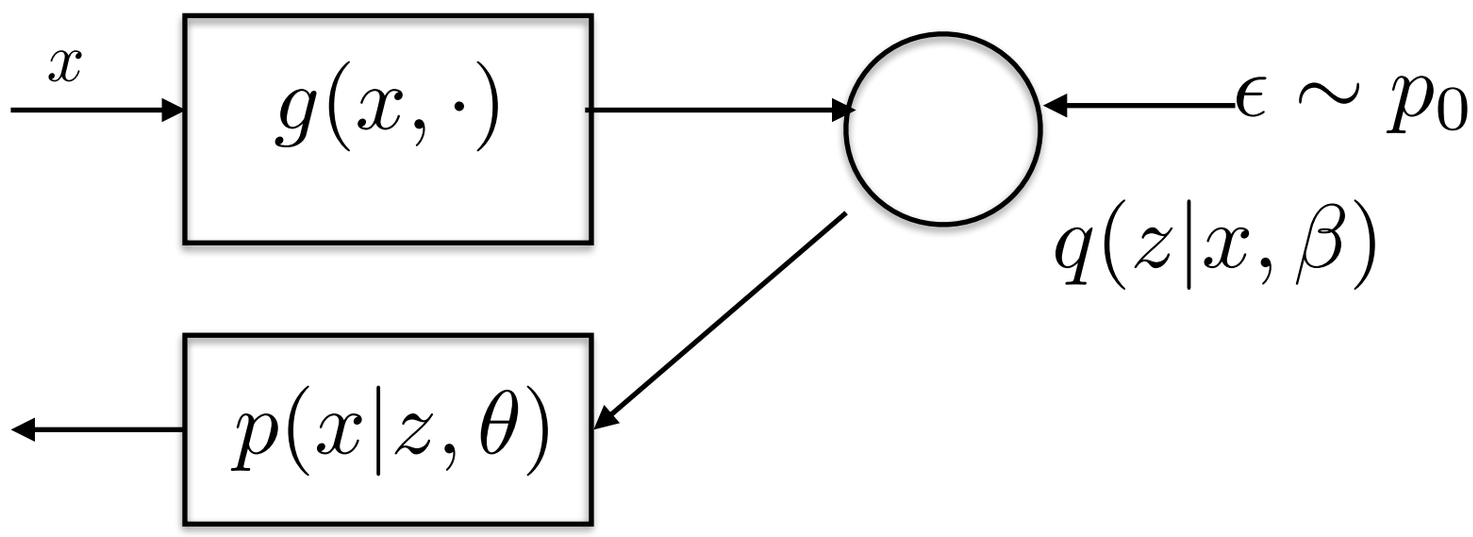
VARIATIONAL AUTOENCODERS

➤How to model $x \mapsto g_\beta(x, \cdot)$ and $z \mapsto p_\theta(\cdot, z)$?

VARIATIONAL AUTOENCODERS

➤How to model $x \mapsto g_\beta(x, \cdot)$ and $z \mapsto p_\theta(\cdot, z)$?

➤VAE idea: use neural networks to approximate variational and generative parameters.



VARIATIONAL AUTOENCODER

➤Example: Let the prior over latent variables be Gaussian isotropic: $$p(z) = \mathcal{N}(z; 0, \mathbf{I})$$

VARIATIONAL AUTOENCODER

➤Example: Let the prior over latent variables be Gaussian isotropic: $p(z) = \mathcal{N}(z; 0, \mathbf{I})$

➤Let the conditional likelihood be also Gaussian:

$p(x|z) = (x; \mu(z), \Sigma(z))$ $\mu(z), \Sigma(z)$: Neural networks

VARIATIONAL AUTOENCODER

➤Example: Let the prior over latent variables be Gaussian isotropic: $p(z) = \mathcal{N}(z; 0, \mathbf{I})$

➤Let the conditional likelihood be also Gaussian:

$$p(x|z) = (x; \mu(z), \Sigma(z)) \quad \mu(z), \Sigma(z) \ : \ \text{Neural networks}$$

➤Variational approximate posterior also Gaussian:

$$q_\beta(z|x) = \mathcal{N}(z; \overline{\mu}(x), \overline{\Sigma}(x))$$

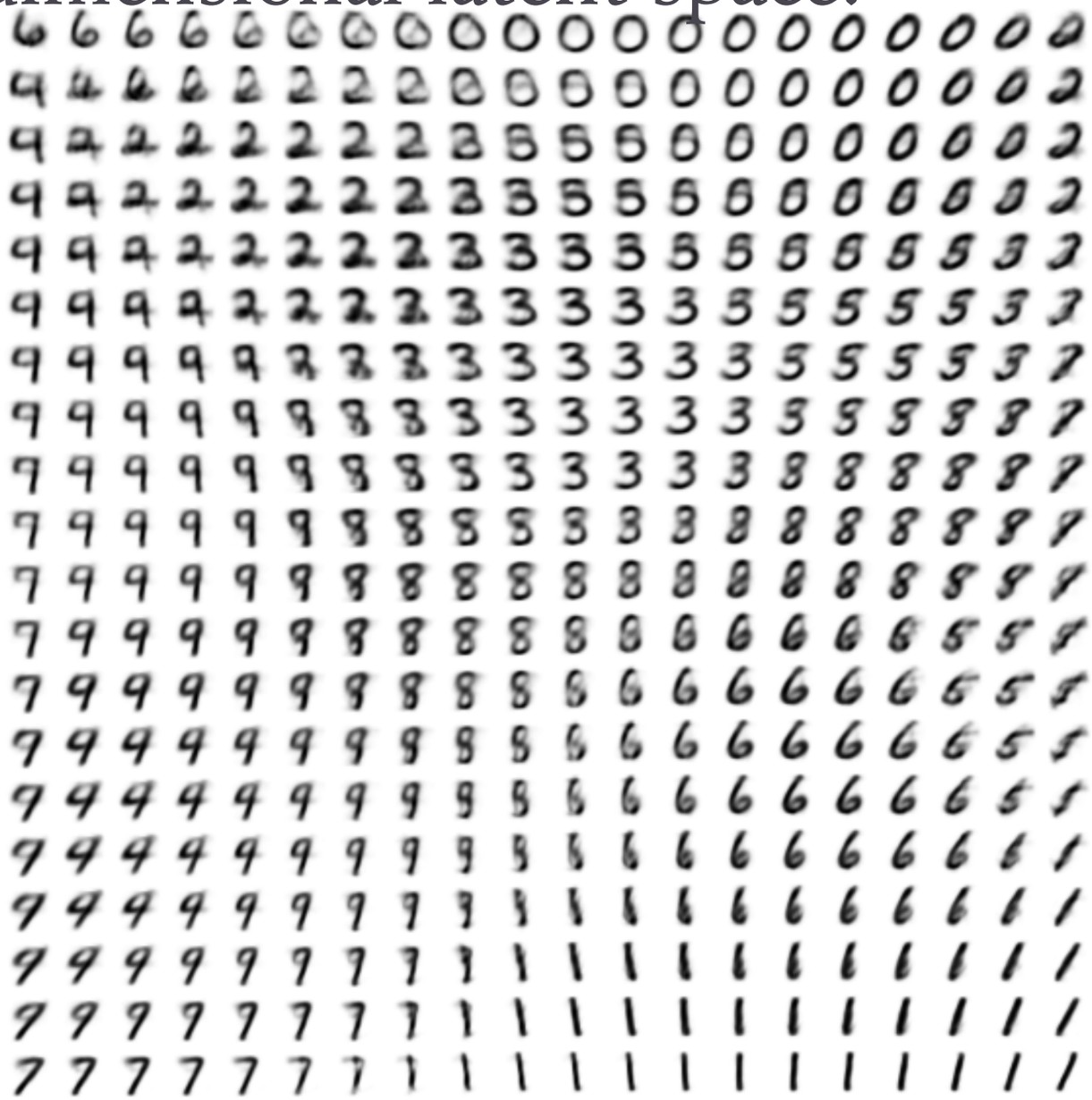
$$\overline{\mu}(z), \overline{\Sigma}(z) \ : \ \text{Neural networks}, (\overline{\Sigma} \text{ diagonal})$$

$$Z \sim q_\beta(z|x) \Leftrightarrow Z = \overline{\mu}(x) + \overline{\Sigma}(x)^{1/2}\epsilon \ , \ \epsilon \sim \mathcal{N}(0, 1)$$

VARIATIONAL AUTOENCODER

➤Examples using a two-dimensional latent space:

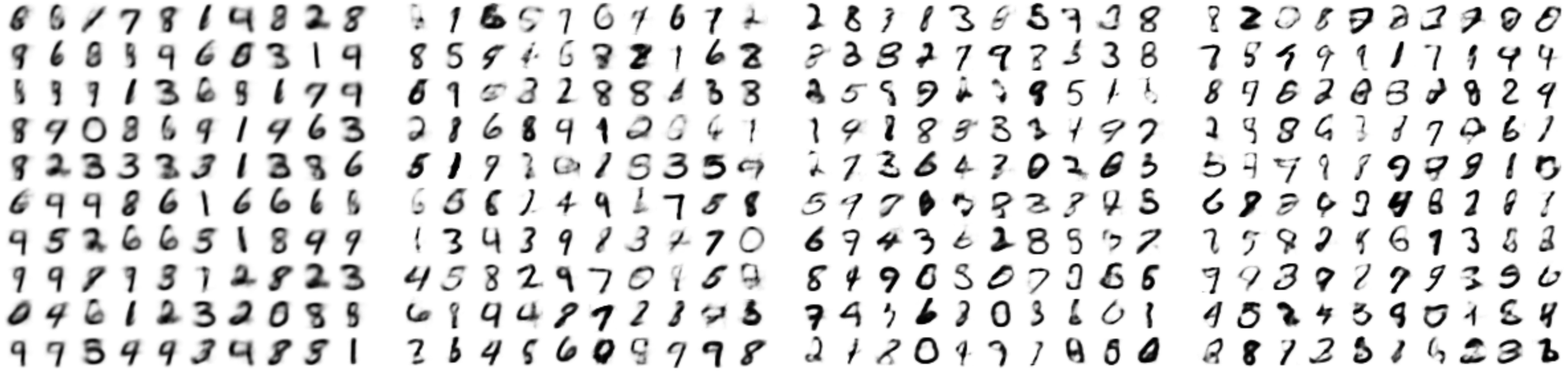
(a) Learned Frey Face manifold



(b) Learned MNIST manifold

EXAMPLES

➤Increasing latent dimensionality:



(a) 2-D latent space (b) 5-D latent space (c) 10-D latent space (d) 20-D latent space

EXTENSIONS TO SEMI-SUPERVISED LEARNING

➤Semi-supervised learning:

We observe $\{x_i\}_{i \leq L_1}$ and $\{x_j, y_j\}_{j \leq L_2}$, with $x_i \sim p(x)$, $x_j \sim p(x)$.

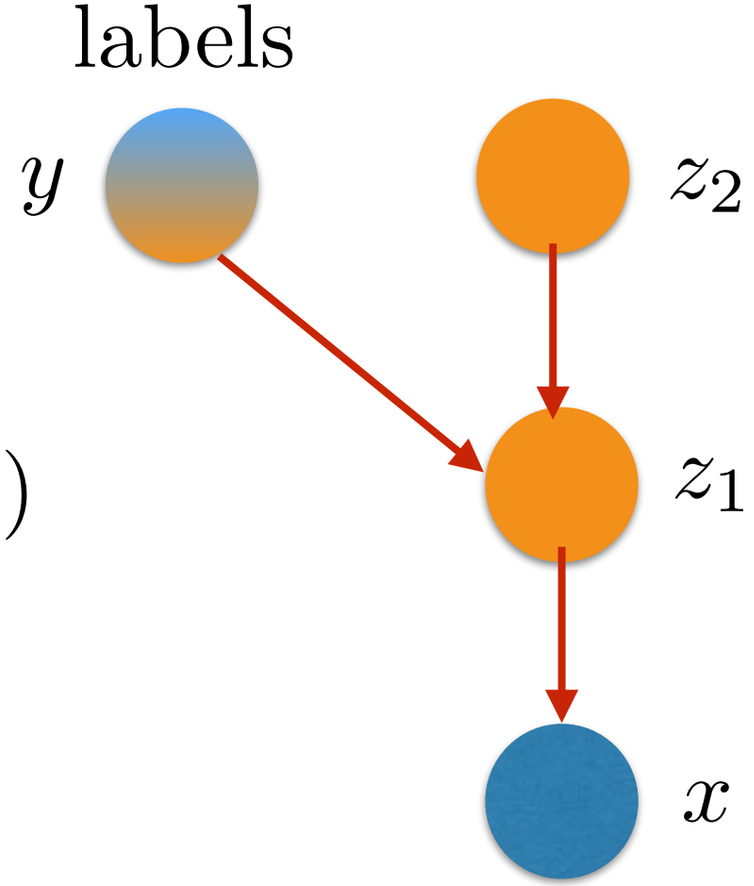
$L_1 \gg L_2$

EXTENSIONS TO SEMI-SUPERVISED LEARNING

➤"Semi-supervised Learning with Deep Generative Networks", Kingma et al,'14.

➤Labels are treated as either observed or hidden.

$$p(x, y, z_1, z_2) = p(y)p(z_2)p(z_1|z_2, y)p(x|z_1)$$



EXTENSION TO SEMI-SUPERVISED LEARNING

➤"Semi-supervised Learning with Deep Generative Networks", Kingma et al,'14.

➤For datapoint with labels:

$$\log p_\theta(x, y) \geq \mathbb{E}_{q_\beta(z|x,y)} \left(\log p_\theta(x|y, z) + \log p_\theta(y) + \log p(z) - \log q_\beta(z|x, y)\right)$$

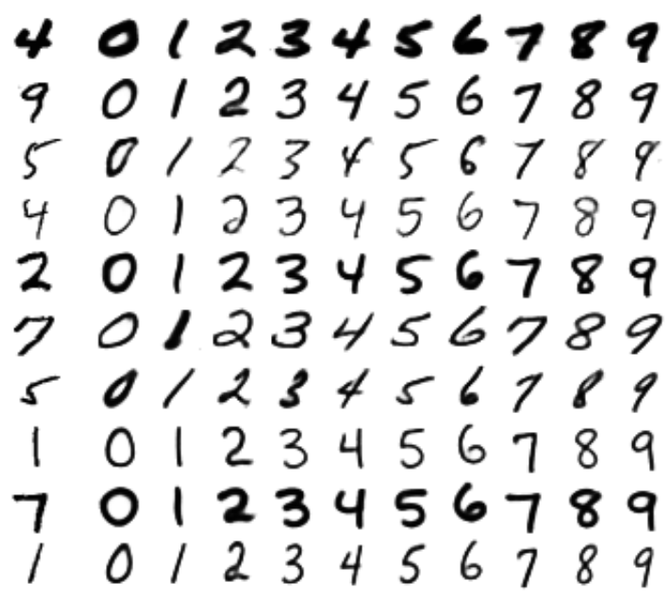
➤For datapoint with no labels:

$$\log p_\theta(x) \geq \mathbb{E}_{q_\beta(y,z|x)} \left(\log p_\theta(x|y, z) + \log p_\theta(y) + \log p(z) - \log q_\beta(z, y|x)\right)$$

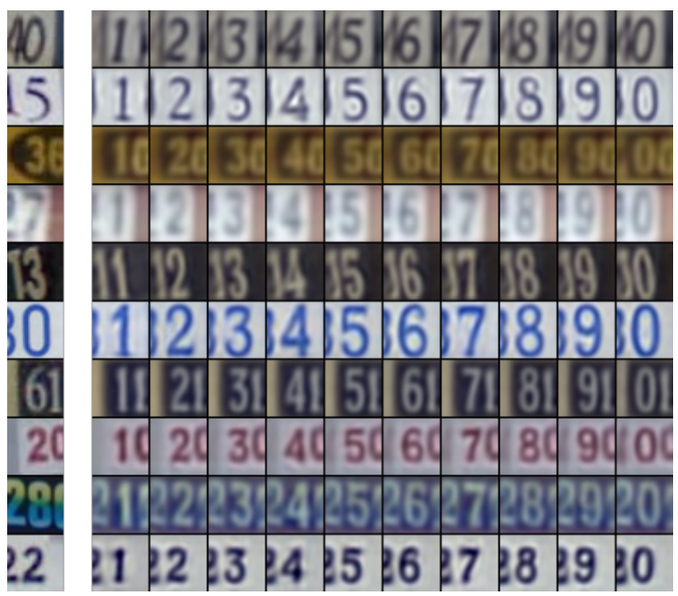
EXTENSION TO SEMI-SUPERVISED LEARNING

- *"Semi-supervised Learning with Deep Generative Networks", Kingma et al,'14.*
- Disentangling label and "style":

(a) Handwriting styles for MNIST obtained by fixing the class label and varying the 2D latent variable **z**



(b) MNIST analogies



(c) SVHN analogies

INCORPORATE MCMC TO POSTERIOR APPROX.

"Markov Chain Monte Carlo and Variational Inference: Bridging the Gap", Salimans et al'15

➤Markov Chains are another powerful tool to approximate intractable posteriors.

$$p(z \mid x) \stackrel{d}{=} \lim_{T \to \infty} q_0(z_0 \mid x) \prod_{t \leq T} q(z_t \mid z_{t-1}, x) \ .$$

INCORPORATE MCMC TO POSTERIOR APPROX.

"Markov Chain Monte Carlo and Variational Inference: Bridging the Gap", Salimans et al'15

➤Markov Chains are another powerful tool to approximate intractable posteriors.

$$p(z \mid x) \stackrel{d}{=} \lim_{T \to \infty} q_0(z_0 \mid x) \prod_{t \leq T} q(z_t \mid z_{t-1}, x) \ .$$

➤For fixed T, this can be seen as another variational approximation, by considering $y = z_1, \ldots, z_{T-1}$ as extra hidden variables.

INCORPORATE MCMC TO POSTERIOR APPROX.

"Markov Chain Monte Carlo and Variational Inference: Bridging the Gap", Salimans et al'15

➤We saw in Lecture 7 how to use Markov Chains to approximate intractable posteriors.

$$p(z \mid x) \stackrel{d}{=} \lim_{T \to \infty} q_0(z_0 \mid x) \prod_{t \leq T} q(z_t \mid z_{t-1}, x) \ .$$

➤For fixed T, this can be seen as another variational approximation, by considering $y = z_1, \ldots, z_{T-1}$ as extra hidden variables.

$r(y|x, z_T)$: auxiliary variational approximation

➤The resulting Variational Lower bound becomes

$$\begin{aligned} \mathcal{L}_{MCMC} &= \mathcal{L} - \mathbb{E}_{q(z_T|x)}\{D_{KL}(r(y|z_T, x)||q(y \mid z_T, x)\} \\ &\leq \mathcal{L} \leq \log p(x) \ . \end{aligned}$$

INCORPORATE MCMC TO POSTERIOR APPROX.

"Markov Chain Monte Carlo and Variational Inference: Bridging the Gap", Salimans et al'15

$$\mathcal{L}_{aux} = \mathbb{E}_q \left\{ \log p(x, z_T) - \log q(z_0|x) \right\} + \sum_{t=1}^{T} \left(\log r_t(z_{t-1}|x, z_t) - \log q_t(z_t|x, z_{t-1}) \right)$$

➤The authors consider Hamilton Monte-Carlo as MCMC choice, resulting in Hamiltonian Variational Inference.

➤It provides a flexible (albeit more computationally demanding) variational approximation that can be adjusted with the number *T* of MCMC steps.

VARIATIONAL INFERENCE WITH IMPORTANCE SAMPLING

➤Another mechanism to improve the variational lower bound is to use importance sampling.

"Importance Weighted Autoencoders"
Burda et al'16

➤For each k, we define

$$\mathcal{L}_k(x) = \mathbb{E}_{z_1,\ldots,z_k \sim q(z|x)} \left[\log \frac{1}{k} \sum_{i=1}^{k} \frac{p(x, z_i)}{q(z_i|x)} \right] .$$

➤It results that

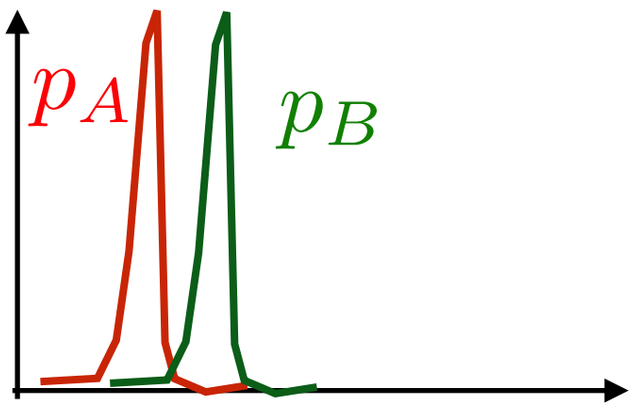
$$\forall\ k\ ,\ \ \log p(x) \geq \mathcal{L}_{k+1}(x) \geq \mathcal{L}_k(x)\ ,\quad \text{and}$$

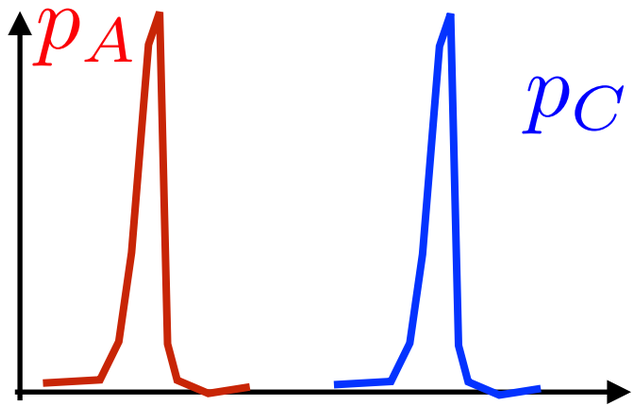
$$\lim_{k\to\infty} \mathcal{L}_k(x) = \log p(x) \text{ if } \frac{p(x,z)}{q(z|x)} \text{ is bounded .}$$

LIMITATIONS OF LIKELIHOOD-BASED LEARNING

- Singular measures do not have density with respect to Lebesgue.
  - Need to add "artificial" noise to make ML work, e.g. $X \mid \{Z = z\} \sim \mathcal{N}(\mu_z, \Sigma_z)$

- Topology is too strong: geometry of input space does not play any role.



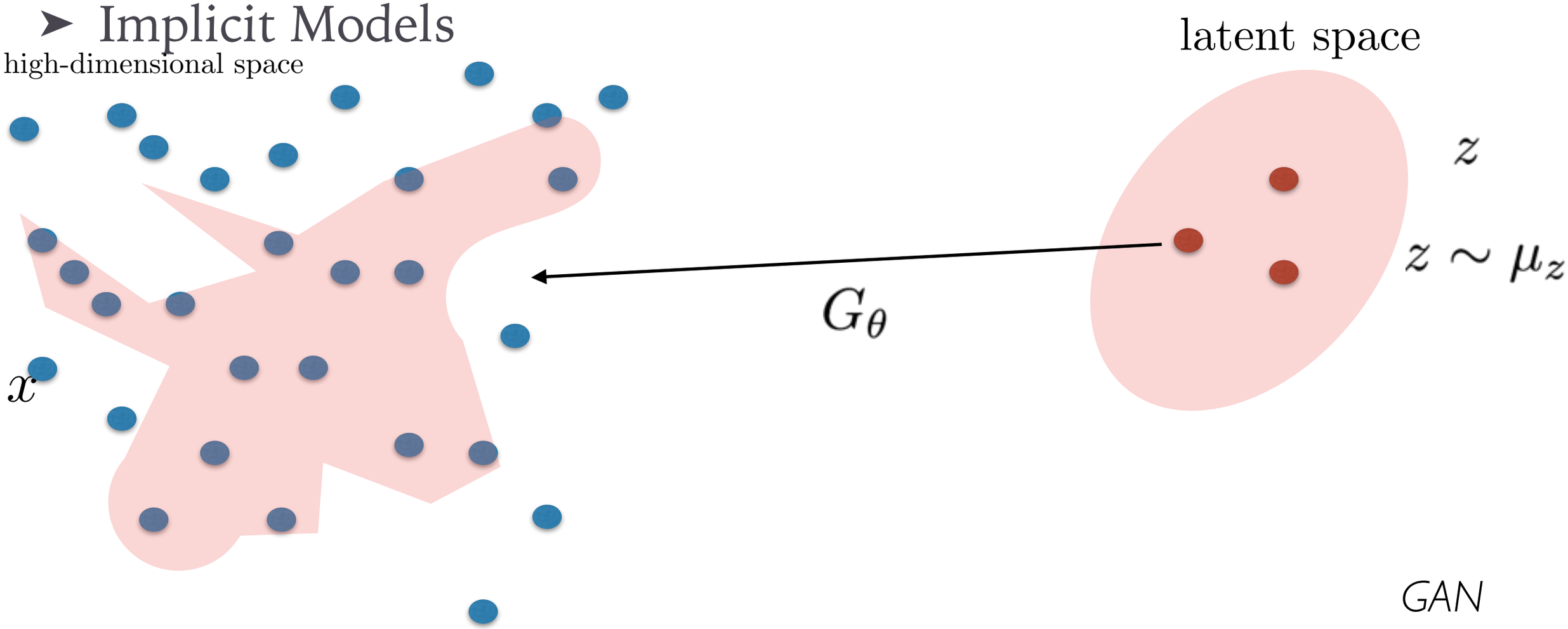


$$\mathrm{MSE}(p_A, p_B) \approx \mathrm{MSE}(p_A, p_C) = O(1)$$

- In particular, stable to deformations?

GENERATIVE MODELS OF COMPLEX DATA

➤ Implicit Models



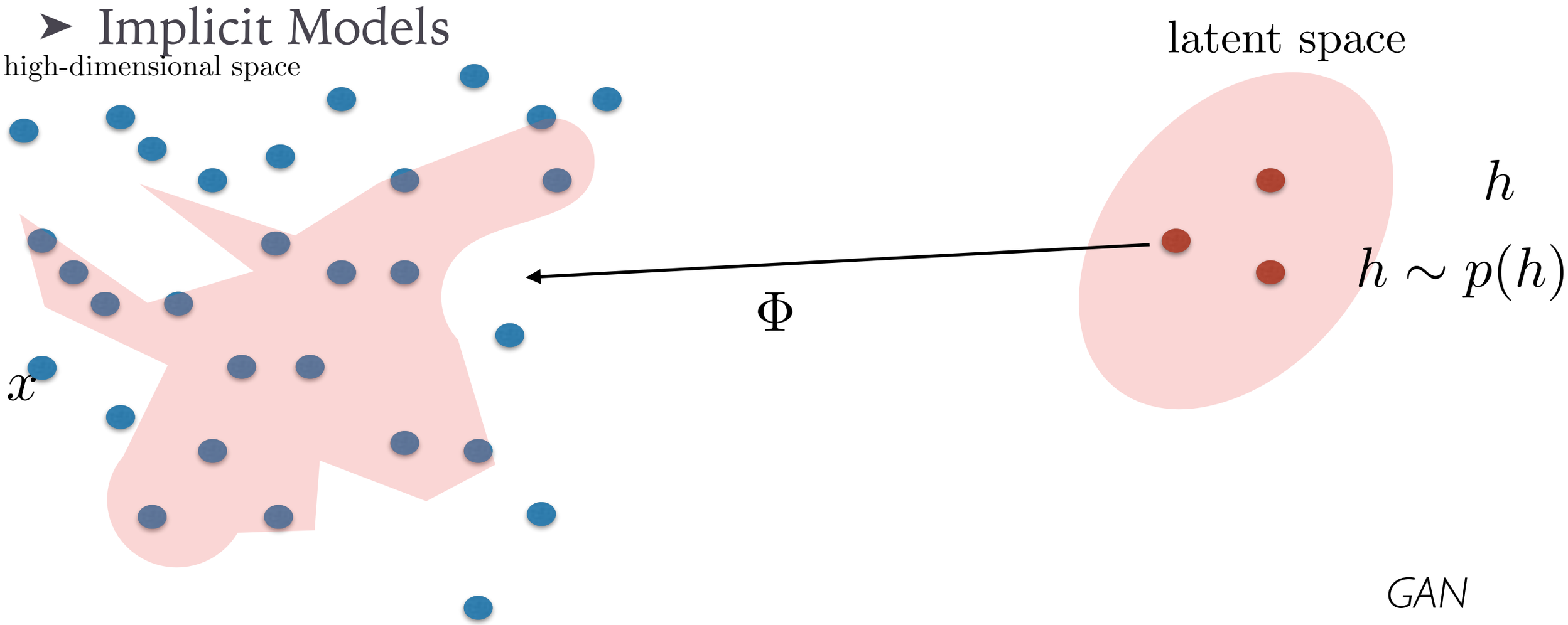
GAN
NormFlow
…

Pushforward measure: $P_\theta := G_\theta \# \mu_z$

$$P_\theta(A) = \mu_z(G_\theta^{-1}(A))$$

GENERATIVE MODELS OF COMPLEX DATA

➤ Implicit Models



GAN
NormFlow
…

$p(x)$ defined implicitly with

$$\int f(x)p(x)d\mu(x) = \int f(G_\theta(z))d\mu_z(z) \;\forall\, f \text{ measurable}$$

ADVERSARIAL TRAINING WITH IMPLICIT MODELS

- More generally, we consider criteria to compare distributions of the form

$$D(Q,P) = \sup_{(f_Q, f_P) \in \mathcal{S}} \mathbb{E}_Q[f_Q(X)] - \mathbb{E}_P[f_P(X)] \ .$$

  - The family $\mathcal{S}$ determines the metric over distributions.
  - Slight generalization of Integral Probability Metrics (IPMs).

- When $P = P_\theta$, this leads to the saddle-point or *adversarial* learning objective:

$$\min_\theta \left\{ C(\theta) := \sup_{(f_Q, f_P) \in \mathcal{S}} \mathbb{E}_Q[f_Q(X)] - \mathbb{E}_{z \sim \mu}[f_P(G_\theta(z))] \right\} \ .$$

DIFFERENT PROBABILITY METRICS

- This adversarial framework includes many existing probability metrics:
- Integral Probability Metrics: $\mathcal{S} = \{(f, f); f, -f \in \mathcal{R}\}$
  - ex the Total Variation distance:

$$D_{TV}(Q, P) := \sup_A |P(A) - Q(A)| = \sup_{f \in C(\mathcal{X}, [-1,1])} \mathbb{E}_Q[f(X)] - \mathbb{E}_P[f(X)]$$

- F-divergences: $D_f(Q, P) = \int f\left(\frac{q(x)}{p(x)}\right) p(x) d\mu(x)$
  - under appropriate regularity

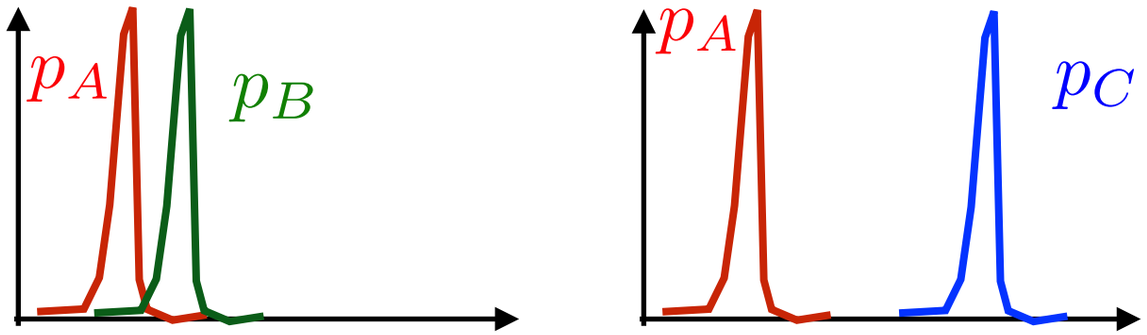
$$D_f(Q, P) = \sup_{|g| < \infty, g(\mathcal{X}) \subseteq \mathrm{dom}(f^*)} \mathbb{E}_Q[g(x)] - \mathbb{E}_P[f^*(g(x))] \ .$$

WASSERSTEIN DISTANCES

- For $p \geq 1$, the p-Wasserstein distance is

$$W_p(Q,P)^p := \inf_{\pi \in \Pi(Q,P)} \mathbb{E}_{(x,y)\sim\pi}[d(x,y)^p]$$

$\Pi(Q,P)$: measures on $\mathcal{X} \times \mathcal{X}$ with marginals Q and P.



$$W(p_A, p_B) \ll W(p_A, p_C)$$

- Variational form is given by Kantorovich duality:

$$W_p(Q,P)^p = \sup_{(f_Q, f_P) \in \mathcal{S}_c} \mathbb{E}_Q[f_Q(X)] - \mathbb{E}_P[f_P(X)]$$

  - p=1 simplifies to

$$W_1(Q,P) = \sup_{f \in \mathrm{Lip}_1} \mathbb{E}_Q[f(X)] - \mathbb{E}_P[f(X)]$$

ENERGY DISTANCES

- The (Euclidean) Energy Distance is defined as

$$\mathcal{E}(Q,P)^2 := 2\mathbb{E}_{X\sim Q,Y\sim P}[\|X-Y\|] - \mathbb{E}_{X\sim Q,X'\sim Q}[\|X-X'\|] - \mathbb{E}_{Y\sim P,Y'\sim P}[\|Y-Y'\|] \ .$$

- Its generalization replaces euclidean distance by a generic symmetric function d(x,y), leading to the Maximum Mean Discrepancies:

$$\mathcal{E}_d(Q,P) = \sup_{\|f\|_{\mathcal{H}}\leq 1} \mathbb{E}_P[f(X)] - \mathbb{E}_Q[f(X)] \ ,$$

- $\mathcal{H}$ is a Reproducing Kernel Hilbert Space associated with the so-called *triangular gap* Kernel associated with d:

$$K_d(x,y) := \frac{1}{2}\left(d(x,x_0) + d(y,x_0) - d(x,y)\right) \ .$$

MEASURE TRANSPORTS

- How to train the transport G_θ?

- We will see two methods:
  - Directly by optimizing data log-likelihood assuming measure admits a density [Normalizing Flows]
  - Using a Discriminative Model [Generative Adversarial Networks]

NORMALIZING FLOWS

➤ The density $q_K(z)$ obtained by transporting a base measure q_0 through a cascade of K diffeomorphisms $\Phi_1, \ldots, \Phi_K$ is

$$z_K = \Phi_K \circ \ldots \Phi_1(z_0) \ , \ \text{with } z_0 \sim q_0(z)$$

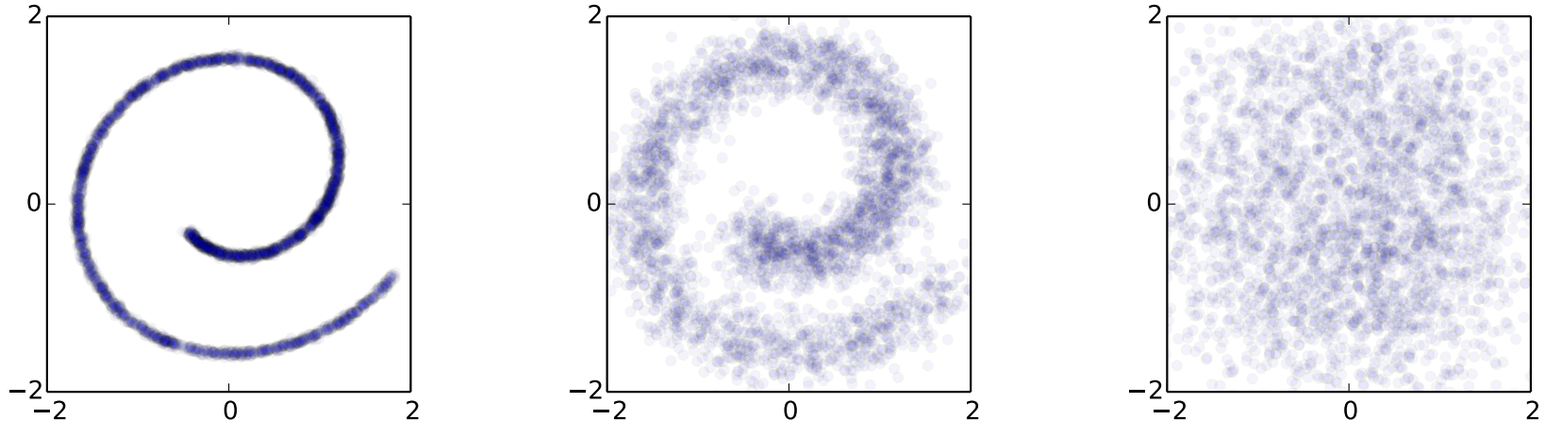
$$\log q_K(z) = \log q_0(z_0) - \sum_{k<K} \log \left| \det \nabla_{z_k} \Phi_k \right| \ .$$

- ➤ One can parametrize invertible flows and use them within the variational inference to improve the variational approximation. [Rezende et al.'15]
- ➤ Also considered in ["NICE", Dinh et al'15].
- ➤ Special case: *Inverse Autoregressive Flows* (i.e. Jacobian triangular) explored in "Variational Inference with Inverse Autoregressive Flows", by [Kingma, Salimans & Welling, NIPS'16].

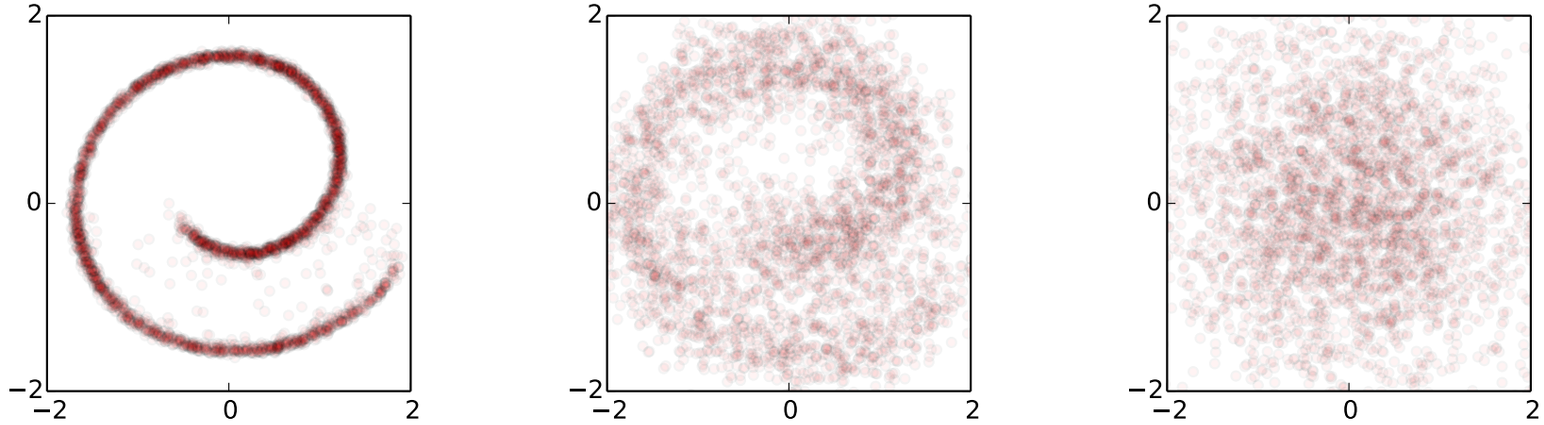
DIFFUSION AND NON-EQUILIBRIUM THERMODYNAMICS

$t = 0$ $t = \frac{T}{2}$ $t = T$

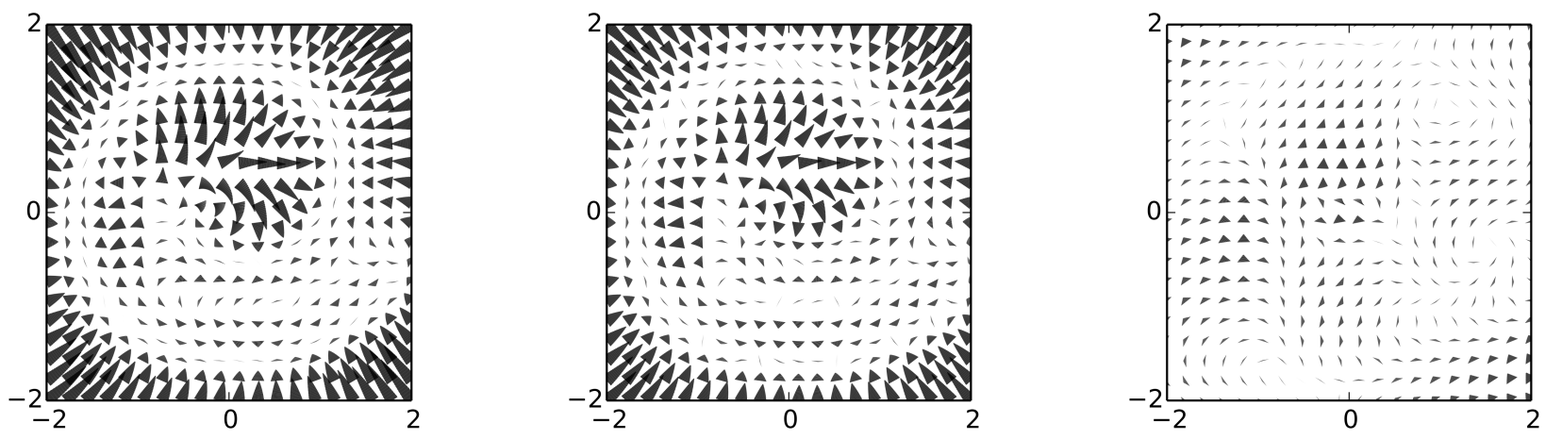
$q\left(\mathbf{x}^{(0\cdots T)}\right)$



$p\left(\mathbf{x}^{(0\cdots T)}\right)$



$\mathbf{f}_\mu\left(\mathbf{x}^{(t)}, t\right) - \mathbf{x}^{(t)}$

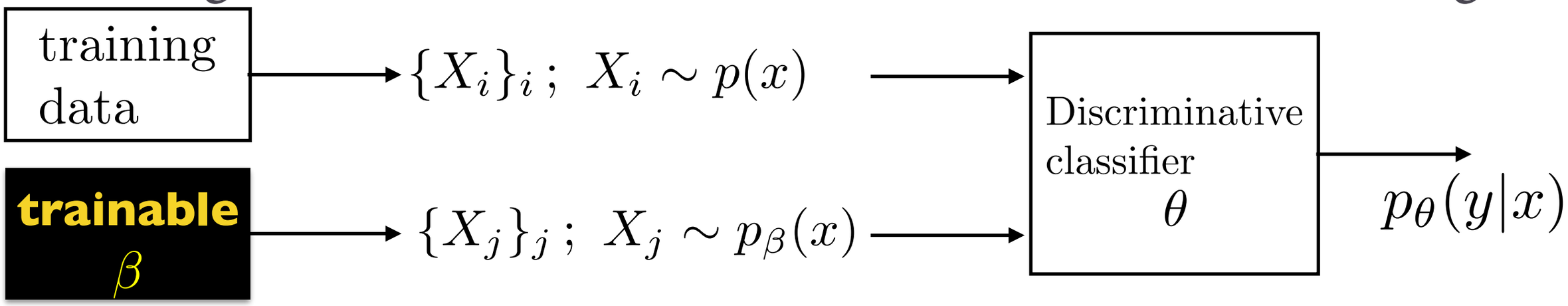


[Sohl-Dickstein et al.'15]

GENERATIVE ADVERSARIAL NETWORKS

[Goodfellow et al., '14]

➤Train generator and discriminator in a minimax setting:



$y = 1$: "real" samples
$y = 0$: "fake" samples

$$\min_{\beta} \max_{\theta} \left(\mathbb{E}_{x \sim p_{data}} \log p_\theta(y = 1|x) + \mathbb{E}_{x \sim p_\beta} \log p_\theta(y = 0|x) \right) \quad .$$

GENERATIVE ADVERSARIAL TRAINING

➤Challenge: it is unfeasible to optimize fully in the inner discriminator loop:

$$\min_{\beta} \max_{\theta} F(\beta, \theta)$$

$$F(\beta, \theta) = \left(\mathbb{E}_{x \sim p_{data}} \log p_\theta(y = 1|x) + \mathbb{E}_{x \sim p_\beta} \log p_\theta(y = 0|x) \right) \ .$$

➤Indeed,

$$\theta^*(\beta) = \arg\max_{\theta} F(\beta, \theta) \ . \qquad G(\beta) := F(\beta, \theta^*(\beta))$$

$$\frac{\partial G(\beta)}{\partial \beta} = 0 \quad w.h.p.$$

➤Numerical approach: alternate k steps of discriminator update with 1 step of generator update.

➤Also, heuristic uses different false positive and false negative losses to improve numerical gradient computations.

LAPGAN

[Denton, Chintala et al.'15]

- Initial GAN models were hard to scale to large input domains.
- Laplacian Pyramid of Adversarial Networks significantly improved quality by generating independently at each scale.
- Laplacian Pyramids are invertible linear multi-scale decompositions:

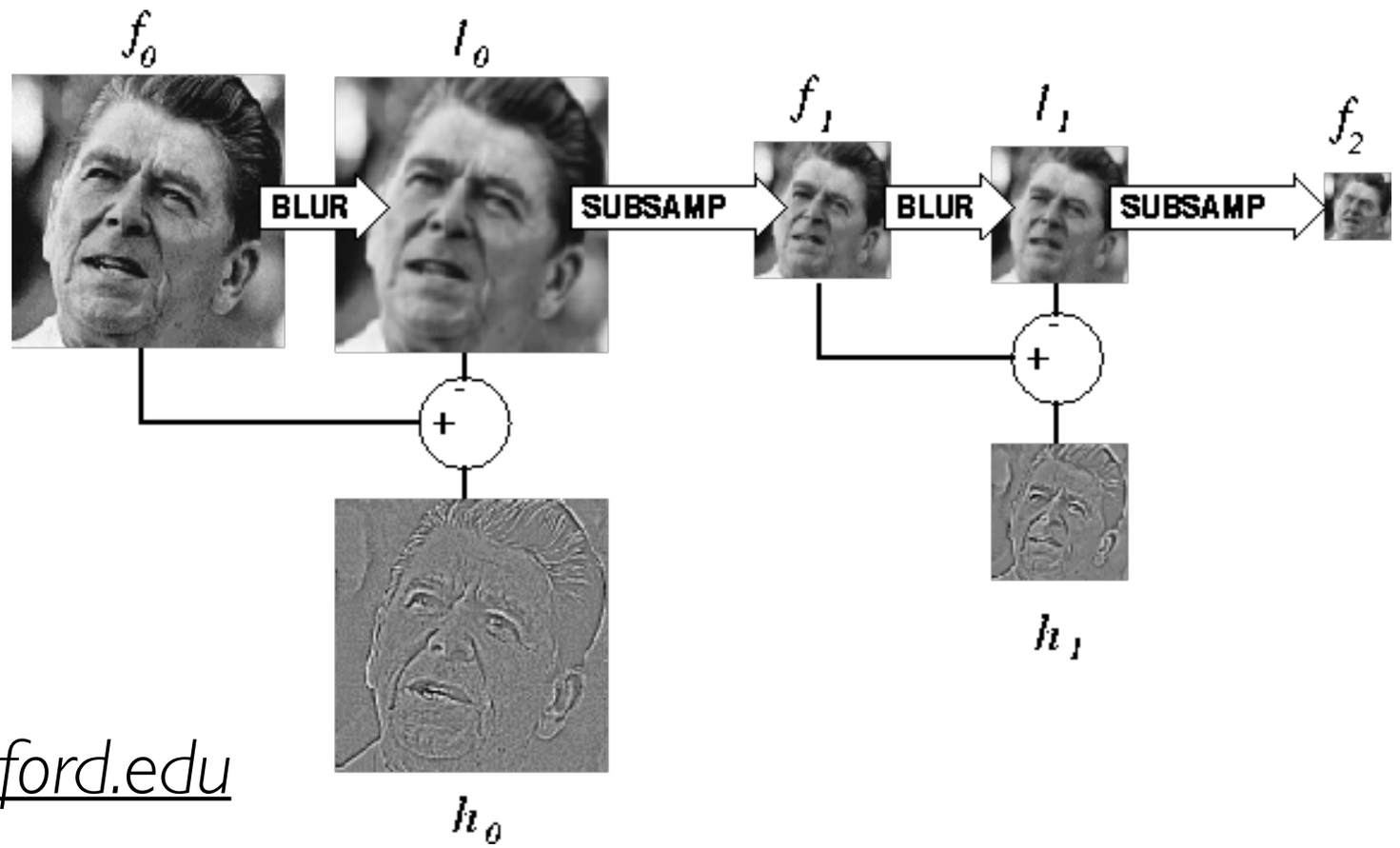
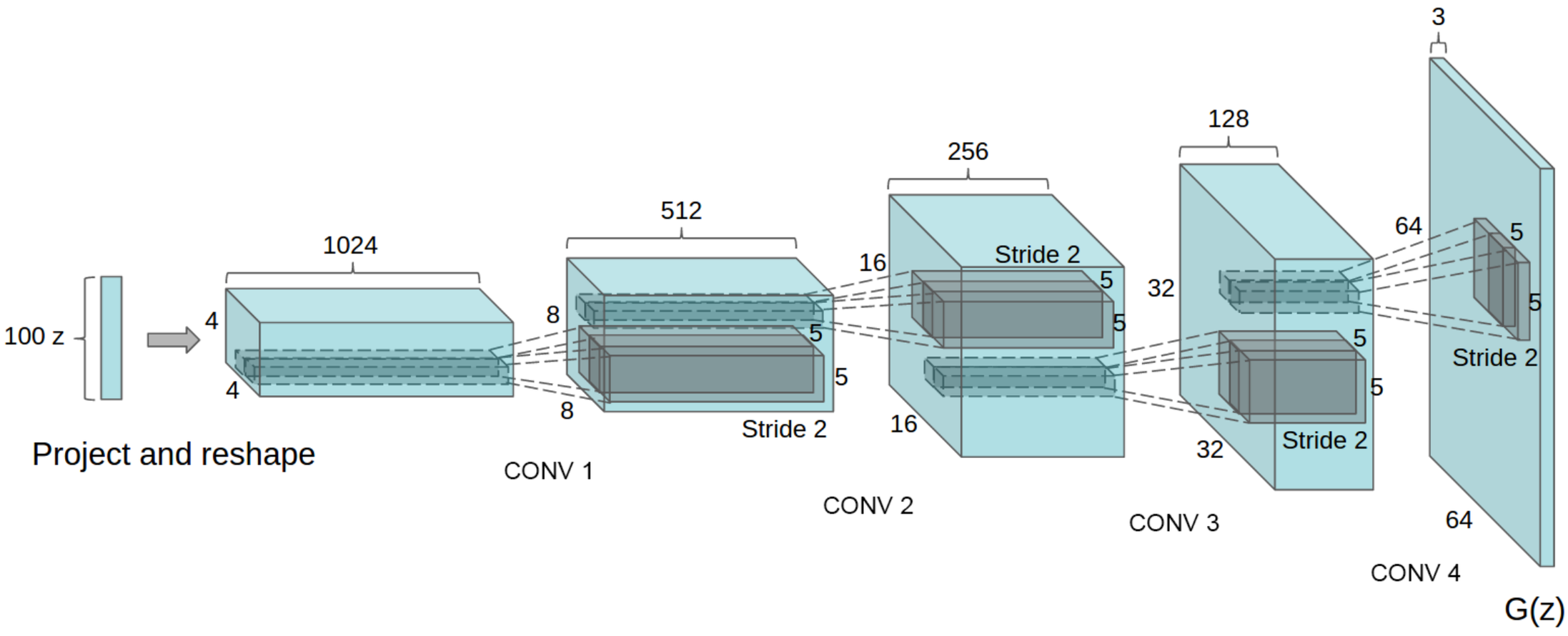


figure source: http://sepwww.stanford.edu

DC-GAN

[Radford et al.'16]

➤Improved multi-scale architecture and Batch-Normalization:

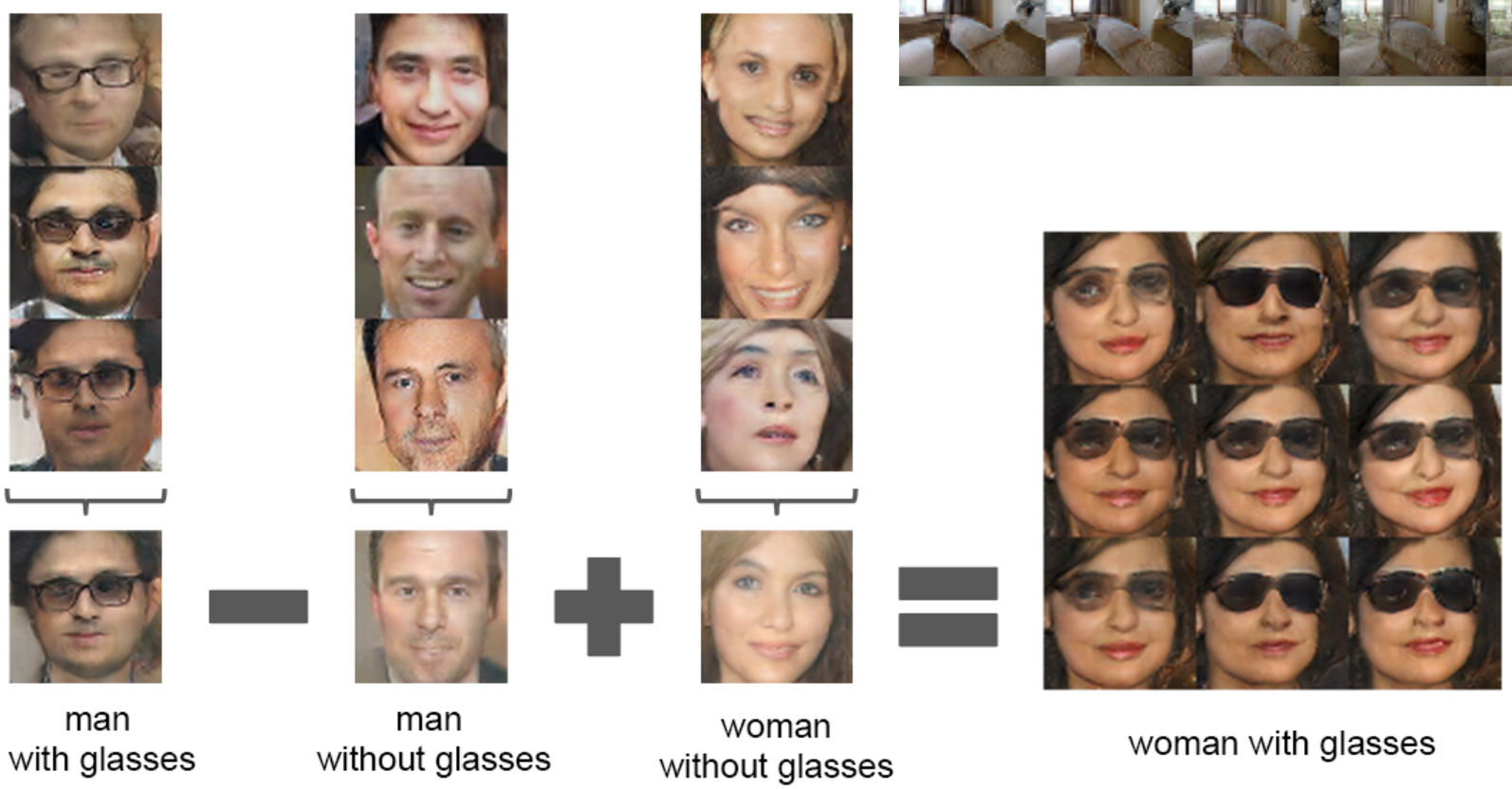


G(z)

DC-GAN

[Radford et al.'16]

➤Improved multi-scale architecture and Batch-Normalization:



KANTOROVICH DUALITY

- In general, computing such "couplings" is hard.
- However, in the Wasserstein-1 case, recall the dual representation:

$$W(p_r, p_m) = \sup_{\mathrm{Lip}(f)\leq 1} \mathbb{E}_{x\sim p_r}\{f(x)\} - \mathbb{E}_{x\sim p_m}\{f(x)\} \ .$$

$f : \mathcal{X} \to \mathbb{R}$ continuous,
$\mathrm{Lip}(f)$: Lipschitz constant of f:
$\forall\ x, x'\ ,\ |f(x) - f(x')| \leq \mathrm{Lip}(f)\|x - x'\|\ .$

KANTOROVICH DUALITY

- In general, computing such "couplings" is hard.
- However, in the Wasserstein-1 case, we have the following dual representation:

$$W(p_r, p_m) = \sup_{\mathrm{Lip}(f)\leq 1} \mathbb{E}_{x\sim p_r}\{f(x)\} - \mathbb{E}_{x\sim p_m}\{f(x)\} \ .$$

$f : \mathcal{X} \to \mathbb{R}$ continuous,
$\mathrm{Lip}(f)$: Lipschitz constant of f:
$\forall\ x, x'\ ,\ |f(x) - f(x')| \leq \mathrm{Lip}(f)\|x - x'\|\ .$

- Recall the discriminator loss of the "classic" GAN:

$$-\mathbb{E}_{x\sim p_r}\{\log D(x)\} + \mathbb{E}_{x\sim p_m}\{\log(1 - D(x))\}$$

WASSERSTEIN GAN [ARJOVSKY ET AL]

- In practice, we approximate the supremum over Lipschitz functions with a class of functions parametrized by a neural network:

$$W(p_r, p_m) = \sup_{\theta \in \mathcal{K}} \mathbb{E}_{x \sim p_r} \{f_\theta(x)\} - \mathbb{E}_{x \sim p_m} \{f_\theta(x)\} \ .$$

- Lipschitz bounds are enforced in [Arjovsky et al.] by simply clipping the weights ($\theta \in \mathcal{K}$).
- Better control of Lipchitz regularity in, e.g. [Gulrajani et al].

- Other works train directly the primal objective using the *Sinkhorn algorithm* [Genevay et al.'17, Bellemare et al'17, Salimans et al'17].

LIMITATIONS OF GAN MODELING

- We are attempting to fit a distribution p_m to the "real" distribution p_r using a distance/divergence criteria ρ :

$$\inf_{\theta} \rho(p_r, p_m(\theta)) \ .$$

LIMITATIONS OF GAN MODELING

- We are attempting to fit a distribution p_m to the "real" distribution p_r using a distance/divergence criteria ρ :

$$\inf_{\theta} \rho(p_r, p_m(\theta)) \ .$$

- However, we do not have access to p_r , only to the *empirical measure* $\hat{p}_{r,L}$:

$$\hat{p}_{r,L}(x) = \frac{1}{L} \sum_{l \leq L} \delta(x - x_l) \ .$$

LIMITATIONS OF GAN MODELING

- We are attempting to fit a distribution p_m to the "real" distribution p_r using a distance/divergence criteriaρ :

$$\inf_{\theta} \rho(p_r, p_m(\theta)) \ .$$

- However, we do not have access to p_r , only to the *empirical measure* $\hat{p}_{r,L}$*:*

$$\hat{p}_{r,L}(x) = \frac{1}{L} \sum_{l \leq L} \delta(x - x_l) \ .$$

- Triangle Inequality:

$$\rho(p_r, p_m(\theta)) \leq \rho(p_r, \hat{p}_{r,L}) + \rho(\hat{p}_{r,L}, p_m(\theta)) \ .$$

Statistical Error

Modeling Error

LIMITATIONS OF GAN MODELING

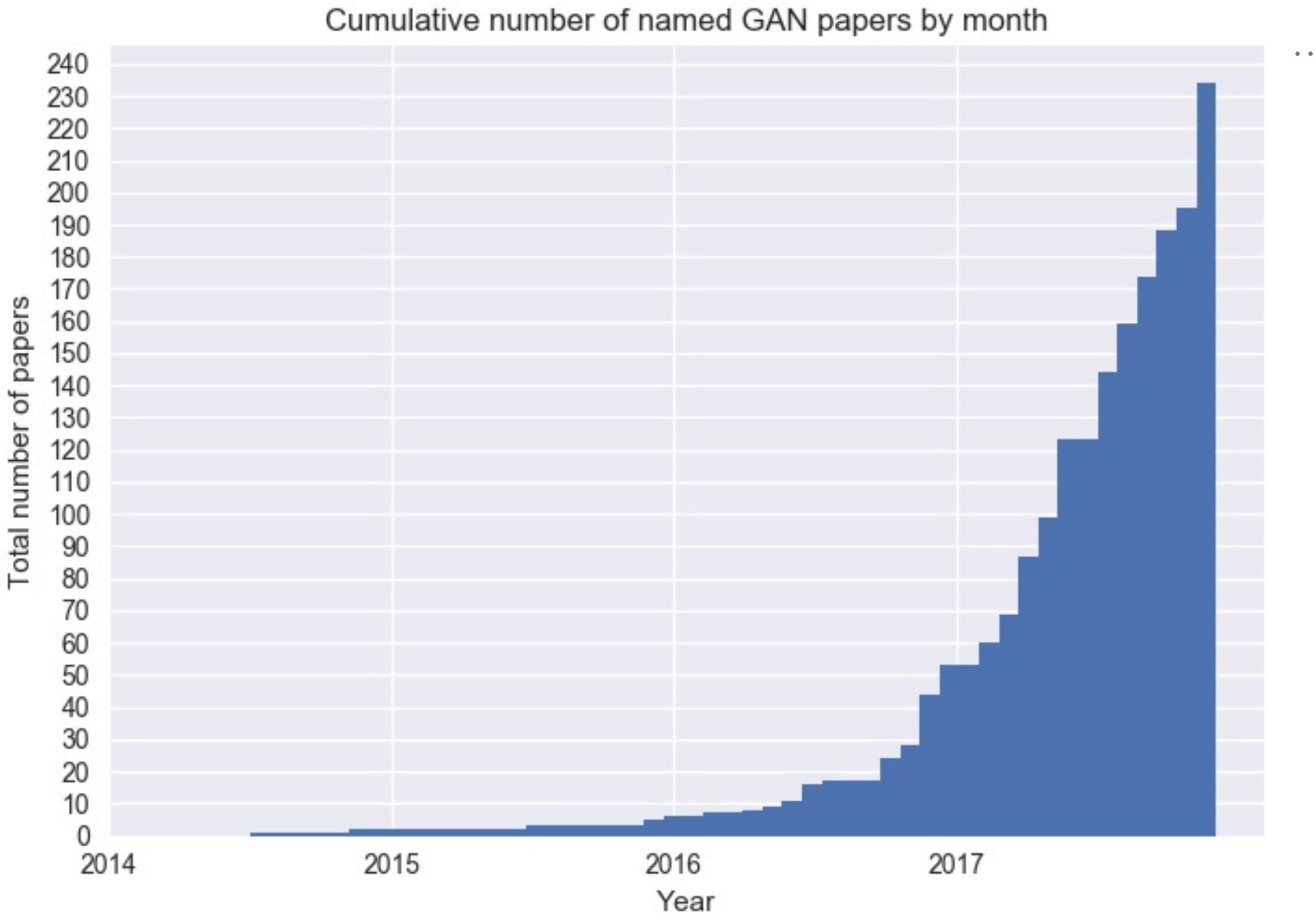
- Thus, we need to regularize the density estimation problem to avoid fitting the empirical measure instead of the underlying real distribution.

- In the case of Wasserstein distance, we have the curse of dimensionality: if input space is $\mathcal{X} = \mathbb{R}^d$, then

$$\mathbb{E}\{\rho(p_r, \hat{p}_{r,L})\} \simeq L^{-\frac{1}{d}}$$

- so we need a number of samples exponential in the dimension to make sampling error disappear.
- Energy distances do not have this curse: $\mathbb{E}\{\rho(p_r, \hat{p}_{r,L})\} \simeq L^{-1/2}$
- Open: why in practice W_1 is more efficient?

GANS ARE BROKEN ARI



source: https://github.com/hindupuravinash/the-gan-zoo

OPTIMAL TRANSPORT OVER GEOMETRIC DISTANCES

- So far, Wasserstein distances are defined over generic metric spaces.
- Consider now the setup where $\mathcal{X} = L^2(\Omega)$ (images)
- How to choose the base distance $d(x, y)$ so that the corresponding Wasserstein metric over is stable with respect to deformations?

OPTIMAL TRANSPORT OVER GEOMETRIC DISTANCES

- So far, Wasserstein distances are defined over generic metric spaces.
- Consider now the setup where $\mathcal{X} = L^2(\Omega)$ (images)
- How to choose the base distance $d(x, y)$ so that the corresponding Wasserstein metric over is stable with respect to deformations?
- Possible idea:

$$d(x, y) := \|\Phi(x) - \Phi(y)\|$$

Φ: Scattering or CNN representation

- Open: Study properties of induced W_Φ. Pseudodistance. When is it a distance?

SOME RECENT EXTENSIONS

- *Image-to-Image Translation with Conditional Adversarial Networks* [Isola et al., '16]

- CycleGAN [Zhu, Park, Isola, Efros]

- Progressive GAN [Karras et al]

- Also, **OT with mixture models: Wasserstein Autoencoders** [Tolstikhin, Bousquet, Gelly, Schoelkopf,'17].

CONDITIONAL GANS

Labels to Street Scene

input

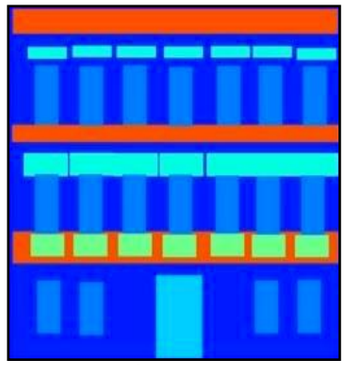
output

Aerial to Map

input

output

Labels to Facade



input

output

BW to Color

input

output

Day to Night

input

output

Edges to Photo

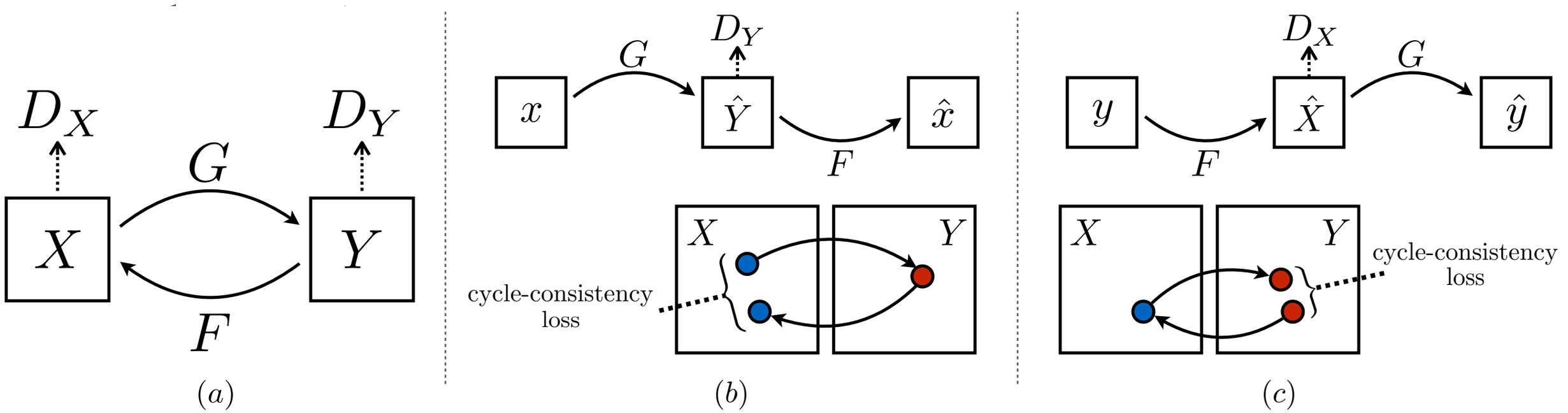
input

output

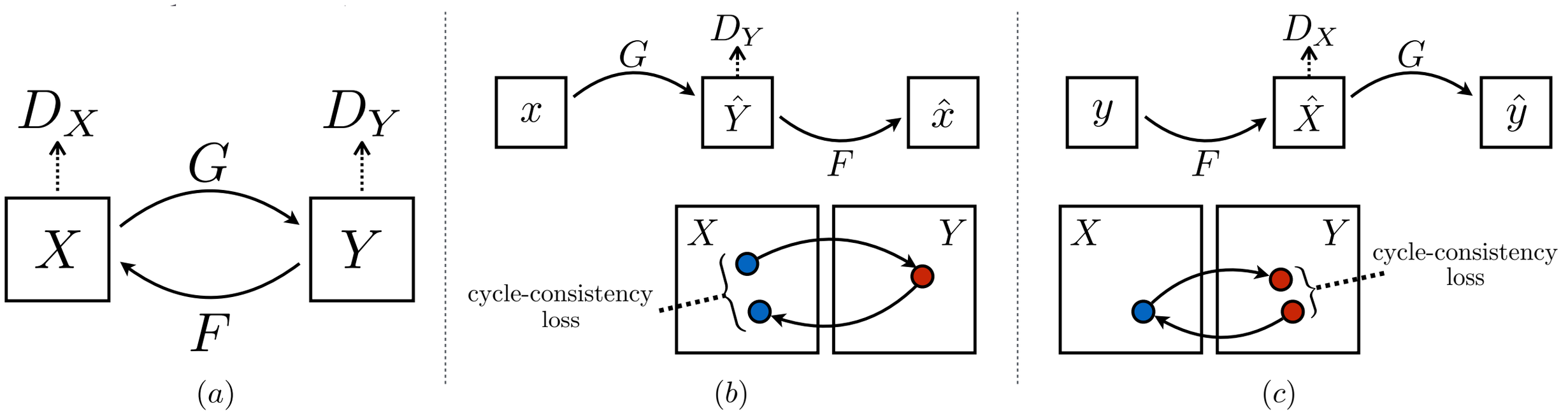
"Image-to-Image translation with Conditional Adversarial Networks", Isola et al,'16

CYCLE GAN [ZHU, PARK, ISOLA, EFROS]

CYCLE GAN


D_X
G
D_Y
X
Y
F
(a)
D_Y
G
x
$\hat{Y}$
$\hat{x}$
F
X
Y
cycle-consistency loss
(b)
D_X
G
y
$\hat{X}$
$\hat{y}$
F
X
Y
cycle-consistency loss
(c)

CYCLE GAN



$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_r(y)}\{\log D_Y(y)\} + \mathbb{E}_{x \sim p_r(x)}\{\log(1 - D_Y(G(x))\} \ .$$
$$\mathcal{L}_{\text{GAN}}(F, D_X, Y, X) = \mathbb{E}_{x \sim p_r(x)}\{\log D_X(x)\} + \mathbb{E}_{y \sim p_r(y)}\{\log(1 - D_X(F(y))\} \ .$$
$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_r(x)}\{\|F(G(x)) - x\|\} + \mathbb{E}_{y \sim p_r(y)}\{\|G(F(y)) - y\|\} \ .$$

Full objective function:

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) + \lambda \mathcal{L}_{\text{cyc}}(G, F) \ .$$

PHOTO REALISTIC GAN [WANG ET AL]

- Enhance the multiscale architecture with resolution-specific discriminators

$$\min_G \max_{D_1, D_2, D_3} \sum_{i=1}^{3} \mathcal{L}_{\text{GAN}}(G, D_i) \ ,$$

- Also includes a "feature matching" term that stabilizes t

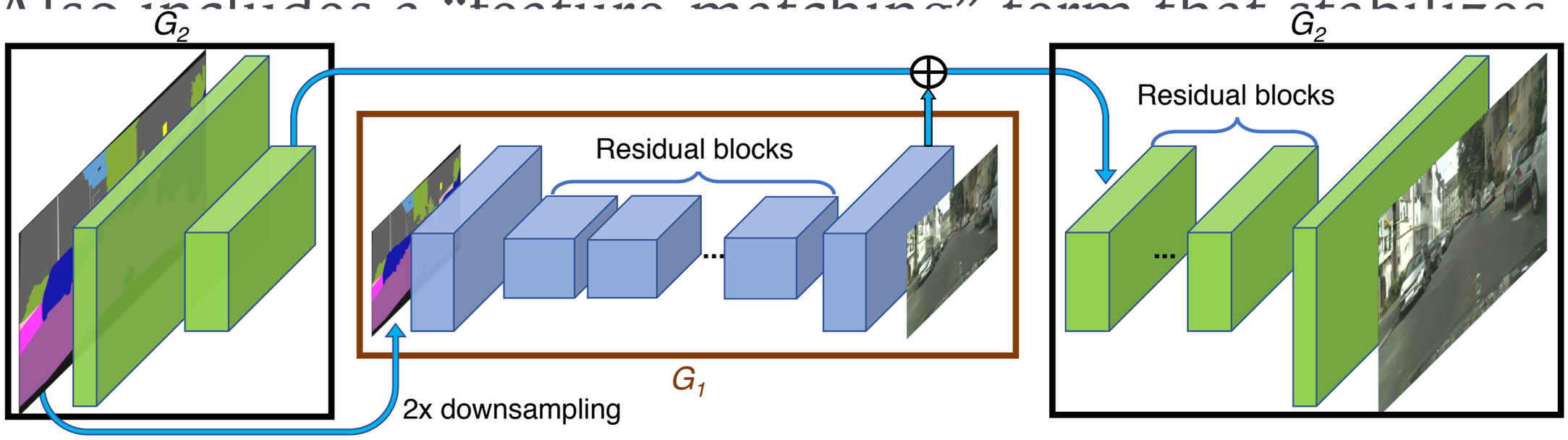
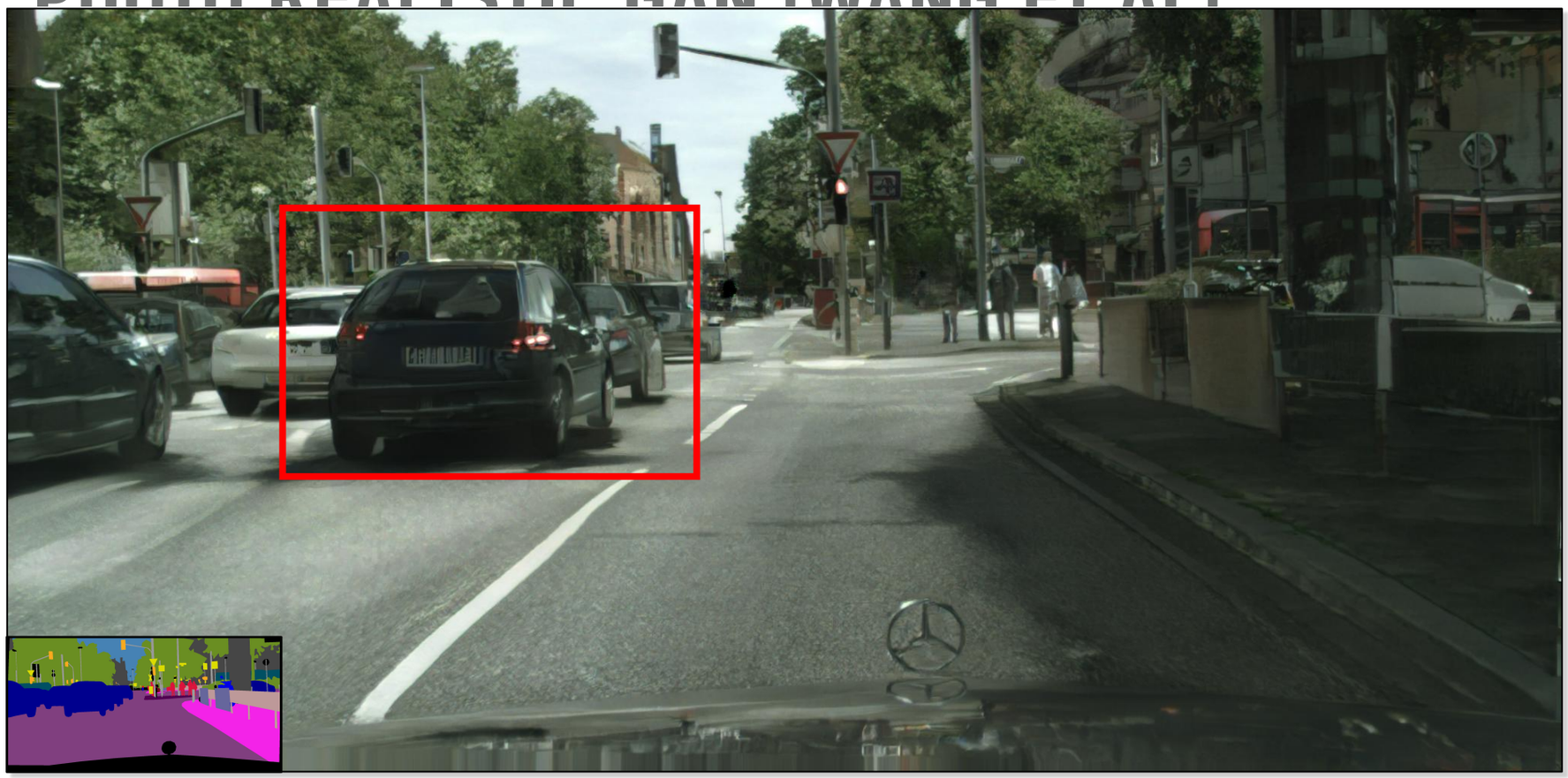


PHOTO REALISTIC GAN [WANG ET AL.]



(a) Synthesized result

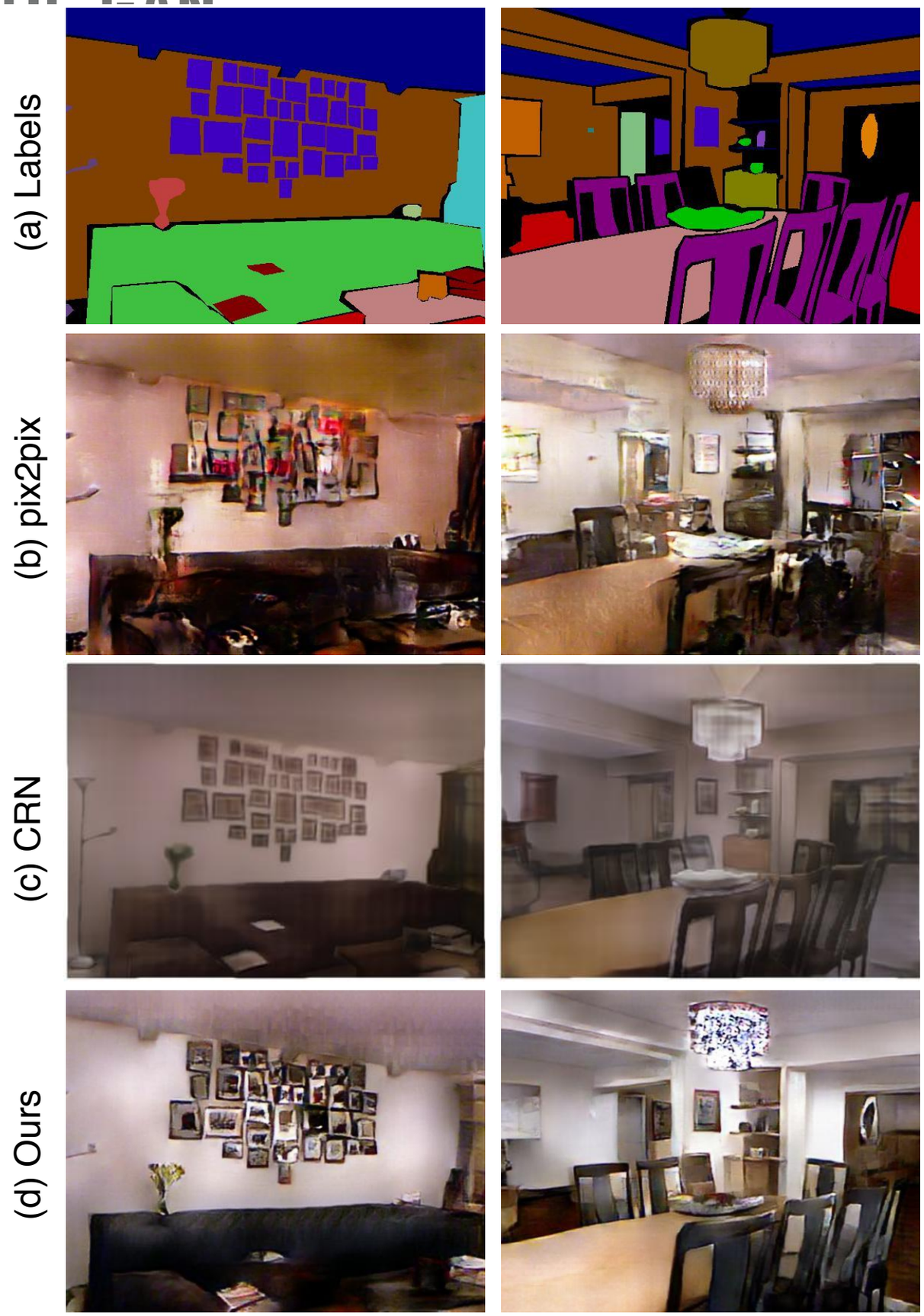
Cascaded refinement network [5]

Our result

(b) Application: Change label types

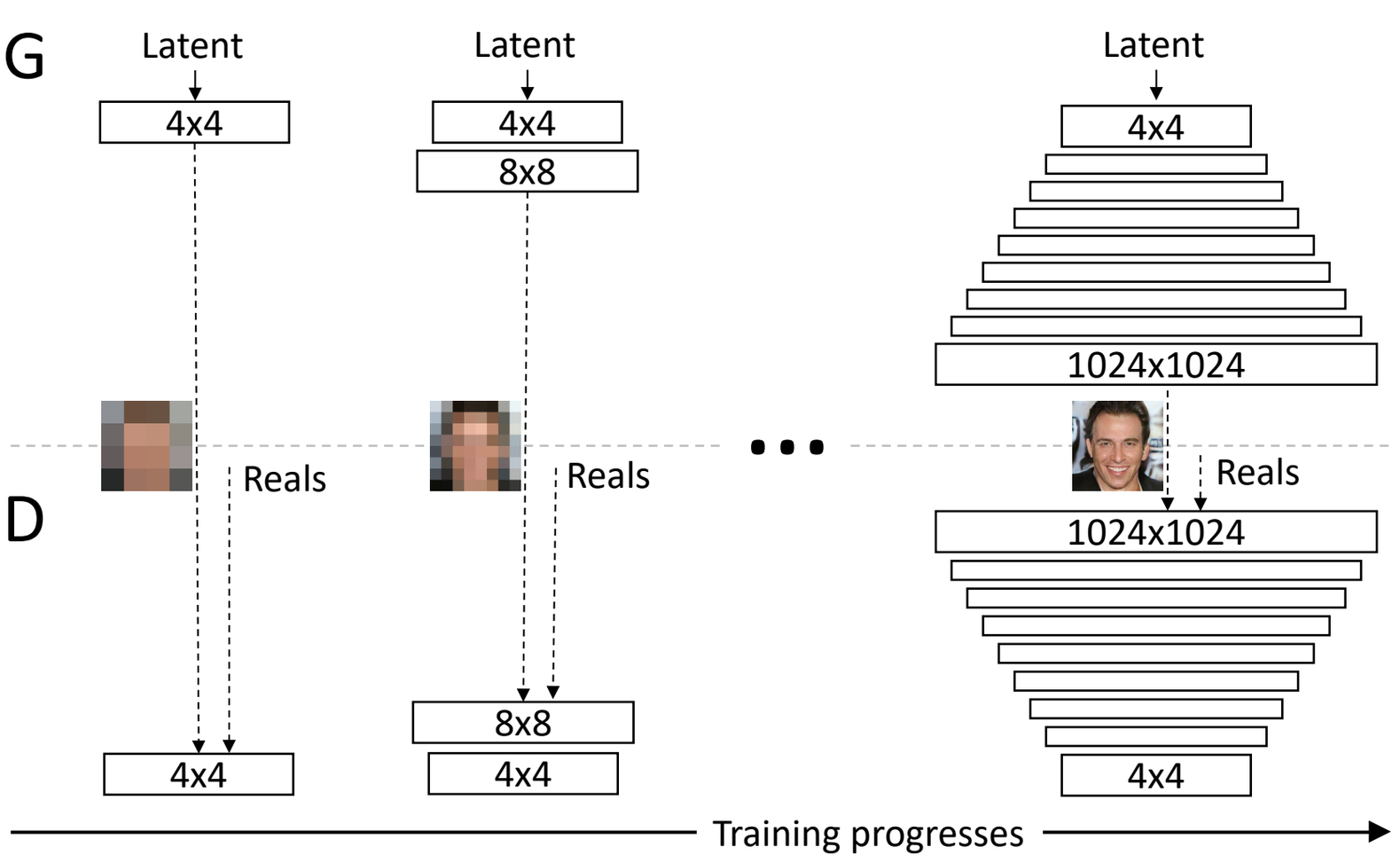
(c) Application: Edit object appearance

PHOTO-REALISTIC GAN



PROGRESSIVE GANS [KARRAS ET AL]

- Multiscale training: fine scales are included in the training progressively.



PROGRESSIVE GAN [KARRAS ET AL]

PROGRESSIVE GANS

POTTEDPLANT HORSE SOFA BUS CHURCHOUTDOOR BICYCLE TVMONITOR

LIMITS OF TRANSPORTATION MODELS

- Direct learning by Optimizing the flow requires back propagation through a term of the form

$$f(\Theta) = \log \det \nabla \Phi(x_i; \Theta)$$

  - Very expensive for generic transformations Φ
    - Highly specific flows affect the flexibility of the model.

- Indirect learning by the Discriminative Adversarial Training is implicit
  - No cheap way to evaluate the density $p(x)$
  - Also, no cheap way to do inference, e.g. $p(z|x)$

AUTOREGRESSIVE MODELS

➤So far, we have seen models that attempt to estimate a density of the input domain $x \in \mathbb{R}^n$

$$p(x) = \int p(h)p(x|h)dh \ , \ p(x|h) = \exp(\langle \theta_h, \Phi(x) \rangle - A(\theta_h))$$

$$p(x) = p_0(\Phi(x)) \cdot |\det \nabla \Phi(x)|^{-1}$$

AUTOREGRESSIVE MODELS

- So far, we have seen models that attempt to estimate a density of the input domain $x \in \mathbb{R}^n$

$$p(x) = \int p(h)p(x|h)dh \ , \ p(x|h) = \exp(\langle \theta_h, \Phi(x) \rangle - A(\theta_h))$$

$$p(x) = p_0(\Phi(x)) \cdot |\det \nabla \Phi(x)|^{-1}$$

- Chained Bayes Rule: for any ordering $(x_{\sigma(1)}, \ldots, x_{\sigma(n)})$ of the coordinates we have

$$p(x) = \prod_{i \leq n} p(x_{\sigma(i)} | x_{\sigma(1)} \ldots x_{\sigma(i-1)})$$

- In which situations it is an appropriate factorization?

AUTOREGRESSIVE MODELS

- Time Series
  - Speech, Music
  - Video
  - Language
  - Other time series (Weather, Finance, ...)
- Spatially ordered data, Multi-Resolution data
  - Images

- Learning is thus reduced to the problem of conditional prediction.

$$p(x) \rightarrow \{p(x_i|x_{N(i)})\}_i$$

CANONICAL AND MICROCANONICAL MODELS

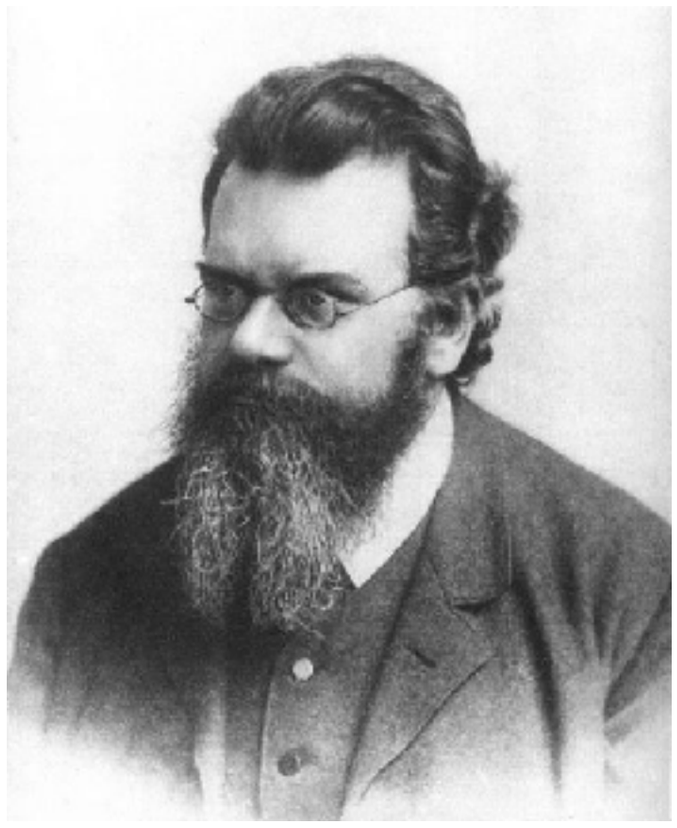
LEARNING WITH SUFFICIENT STATISTICS

- A feature representation $\Phi(x)$ defines a class of densities (relative to a base measure) via sufficient statistics:

$$p_\theta(x) = g_\theta(\Phi(x)) \ .$$

- $\Phi(x)$ may encode prior knowledge to break the curse of dimensionality (e.g if $\dim(\Phi(x)) \ll \dim(x)$)
- If $\Phi(x)$ is stable to transformations and g_θ is smooth, then p_θ is also stable.

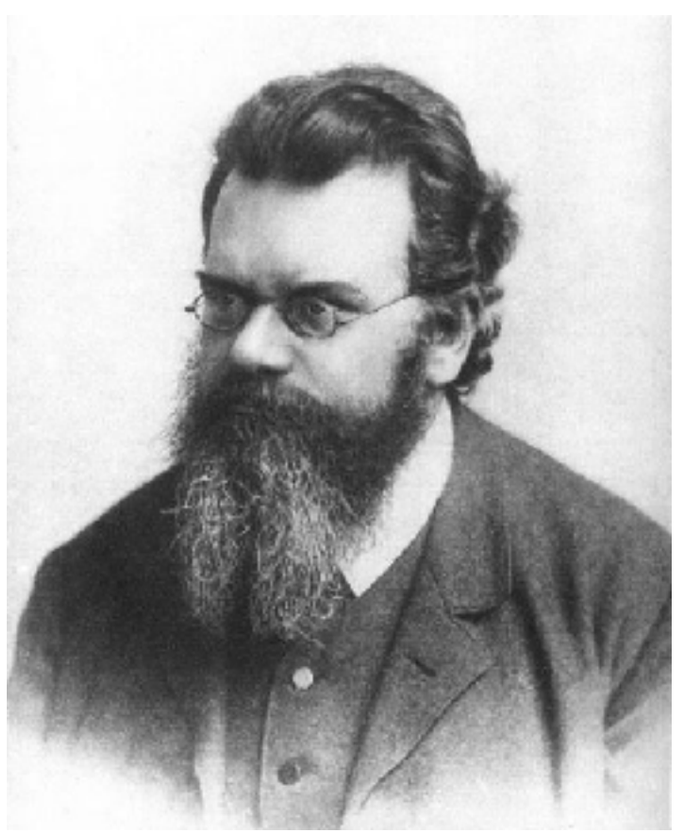
CANONICAL REPRESENTATION



CANONICAL REPRESENTATION

- The *canonical ensemble* representation is defined as the exponential family determined by $\Phi(x)$:

$$p_\theta(x) = \frac{\exp(\langle \theta, \Phi(x) \rangle)}{Z} \ .$$



L. Boltzmann

CANONICAL REPRESENTATION

- The *canonical ensemble* representation is defined as the exponential family determined by $\Phi(x)$:

$$p_\theta(x) = \frac{\exp(\langle \theta, \Phi(x) \rangle)}{Z} \ .$$

- θ are so-called *canonical* parameters.
- **Maximum entropy** model subject to the constraint that

$$\mathbb{E}_{x \sim p_\theta}(\Phi(x)) = \mathbb{E}_{x \sim p}(\Phi(x)) \coloneqq \mu \ .$$

CANONICAL REPRESENTATION

- The *canonical ensemble* representation is defined as the exponential family determined by $\Phi(x)$:

$$p_\theta(x) = \frac{\exp(\langle \theta, \Phi(x) \rangle)}{Z} \ .$$

  - θ are so-called *canonical* parameters.
  - **Maximum entropy** model subject to the constraint that

$$\mathbb{E}_{x \sim p_\theta}(\Phi(x)) = \mathbb{E}_{x \sim p}(\Phi(x)) \coloneqq \mu \ .$$

- θ and μ are related via a variational principle:

$$H(\mu) = \sup_\theta \left(\langle \mu, \theta \rangle - \log Z(\theta) \right) \ .$$

- Challenge: computationally intensive: need to rely to MCMC methods (e.g. Gibbs sampling).

FROM CANONICAL TO MICROCANONICAL

- Alternatively, we may search for features Φ such that $\Phi(X)\,, X \sim p$ **concentrates**, i.e. becomes Gaussian with $\|\Sigma\| \to 0$.

FROM CANONICAL TO MICROCANONICAL

- Alternatively, we may search for features Φ such that $\Phi(X), X \sim p$ **concentrates**, i.e. becomes Gaussian with $\|\Sigma\| \to 0$.
- Then, define p_θ as the maximum entropy distribution such that

$$\Phi(X) \stackrel{d}{=} \Phi(Y) \ , X \sim p, Y \sim p_\theta \ .$$

FROM CANONICAL TO MICROCANONICAL

- Alternatively, we may search for features Φ such that $\Phi(X), X \sim p$ concentrates, i.e. $\|\Sigma\| \to 0$.
- Then, define p_θ as the maximum entropy distribution such that

$$\Phi(X) \overset{d}{=} \Phi(Y) \ , X \sim p, Y \sim p_\theta \ .$$

- If $A\|x\| \leq \|\Phi(x)\| \leq B\|x\|$, then p_θ is constructed as
  1. sample $z \sim q$, $(\text{where } \Phi(X) \sim q)$
  2. sample $x \sim \text{Unif}(\{x\,; \Phi(x) = z\})$.
- Estimation: Finding the parameters of Gaussian model.
- Sampling by solving non-linear LS problems:

$$\min_x \|\Phi(x) - z\|^2 \ .$$

- This is a Mixture of Micro-canonical ensembles from statistical physics. Non-asymptotic model.

MICROCANONICAL LIPSCHITZ MODELS

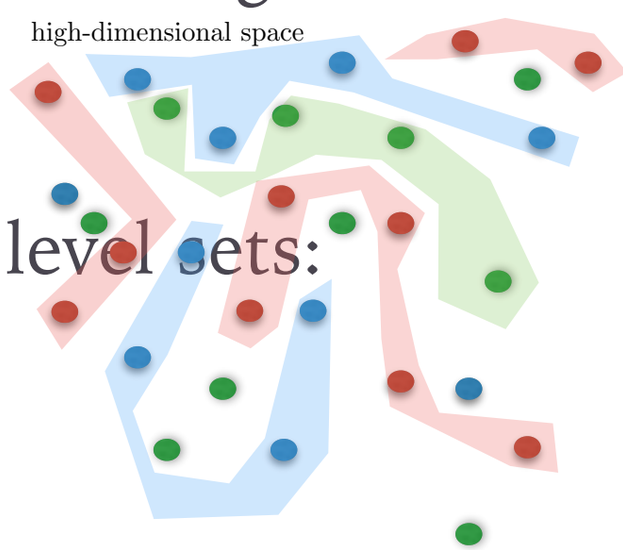
REPRESENTATION WISH LIST

Φ

- transforms unknown, complex distributions into a known model, e.g. Gaussian.
  - High-dimensional concentration mechanisms: **LLNs**, **CLT.**
  - Control of Variance $\mathbb{E}(\|\Phi(X) - \mathbb{E}\Phi(X)\|^2)$ is sufficient via Chebyshev ineq.

$p(x)$ Φ

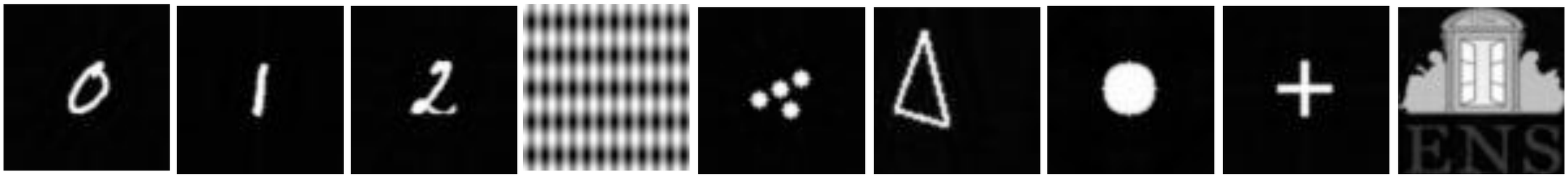
- Φ should be smooth within the Level Sets of .
  - must capture the invariances of the data, but nothing else!
  - Level sets “not too large”.
  - High-dimensional mechanism to control size of level sets: **Sparsity**



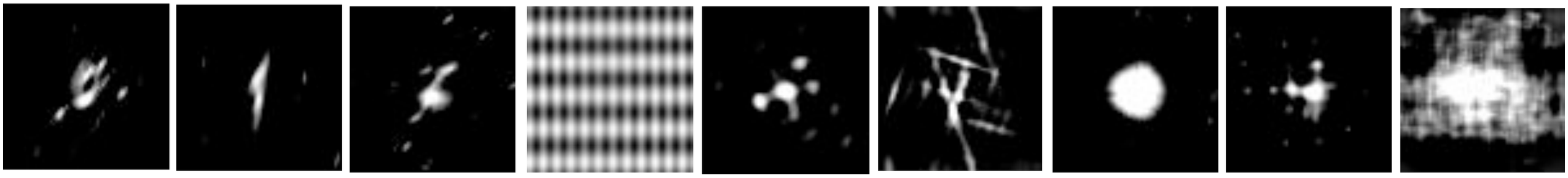
SCATTERING MICROCANONICAL MODELS

SPARSE SHAPE RECONSTRUCTIONS

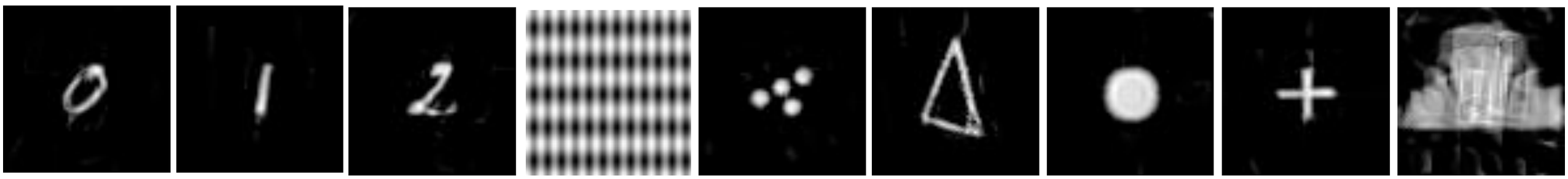
Original images of N^2 pixels:



$m = 1$, $2^J = N$: reconstruction from $O(\log_2 N)$ scattering coeff.

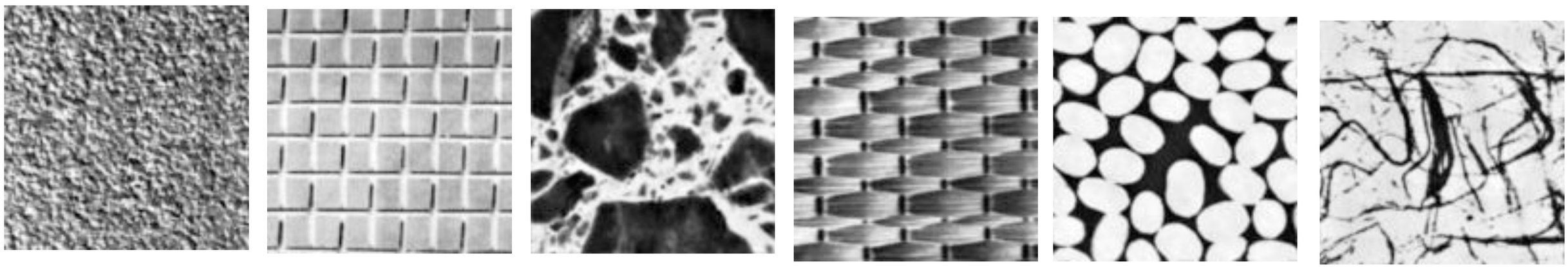


$m = 2$, $2^J = N$: reconstruction from $O(\log_2^2 N)$ scattering coeff.

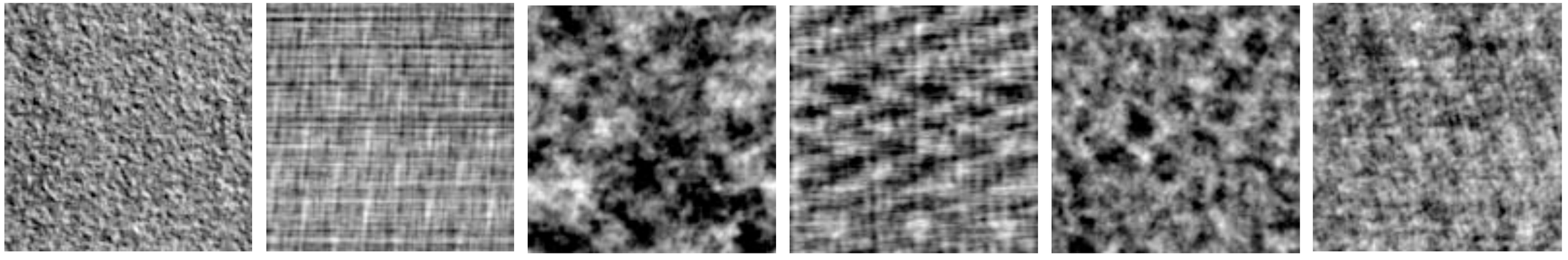


ERGODIC TEXTURE RECONSTRUCTION

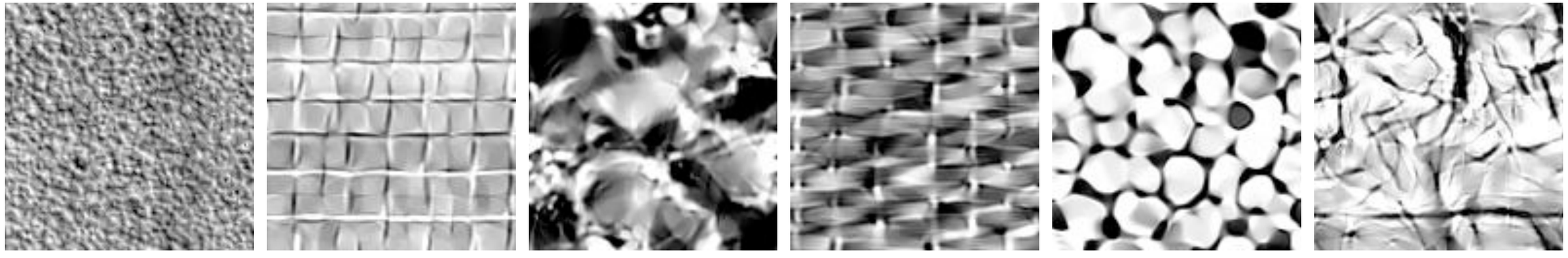
Original Textures



Gaussian process model with same second order moments



$m = 2$, $2^J = N$: reconstruction from $O(\log_2^2 N)$ scattering coeff.



ERGODIC TEXTURE RECONSTRUCTION USING CNNS

➤Results using a deep CNN network from [Gatys et al, $O((\log N)^2)$

➤

)