

Stat 212b: Topics in Deep Learning

Lecture 16

Joan Bruna
UC Berkeley



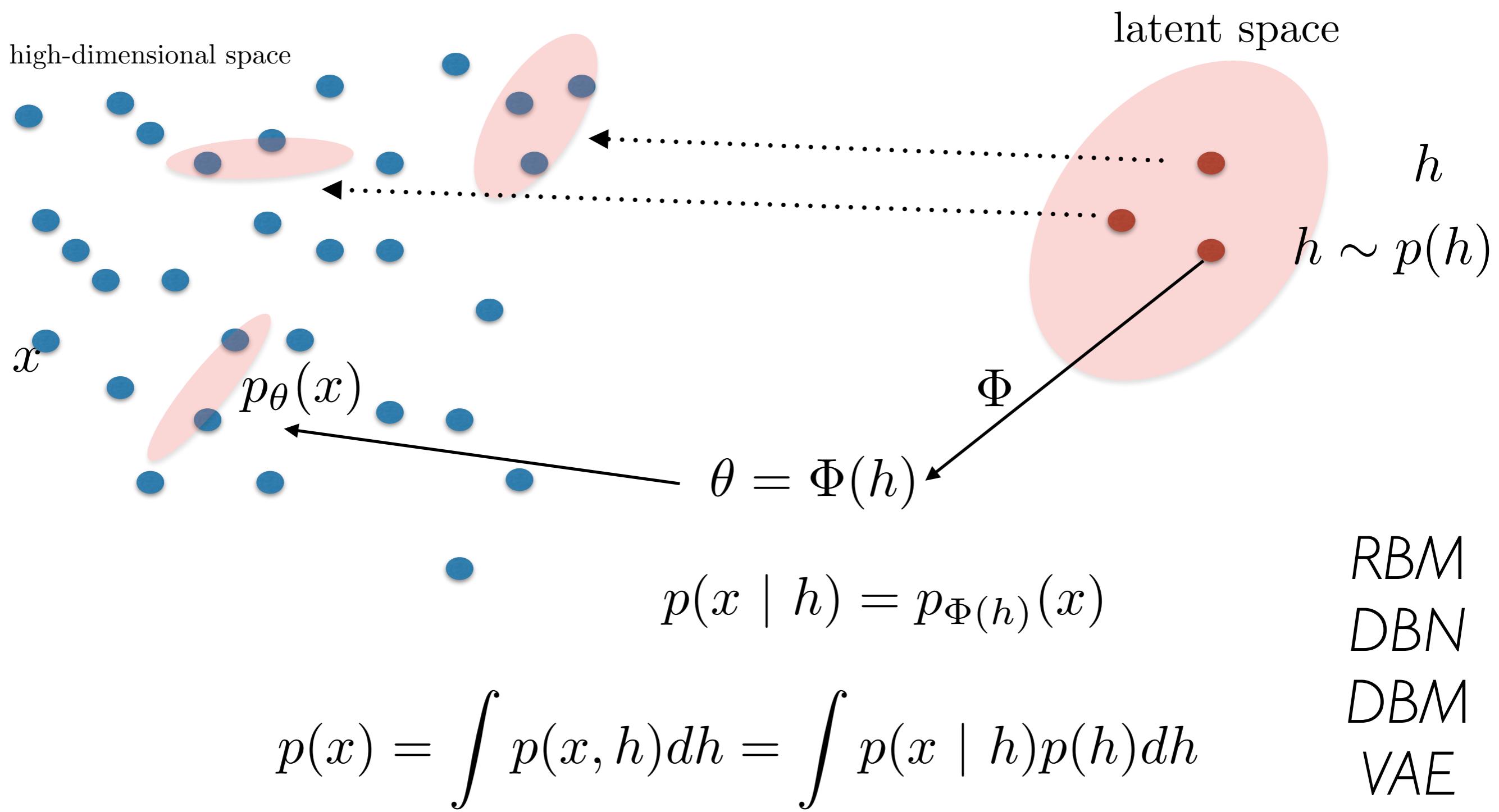
Today

- AlphaGo 2 - Lee Sedol 0



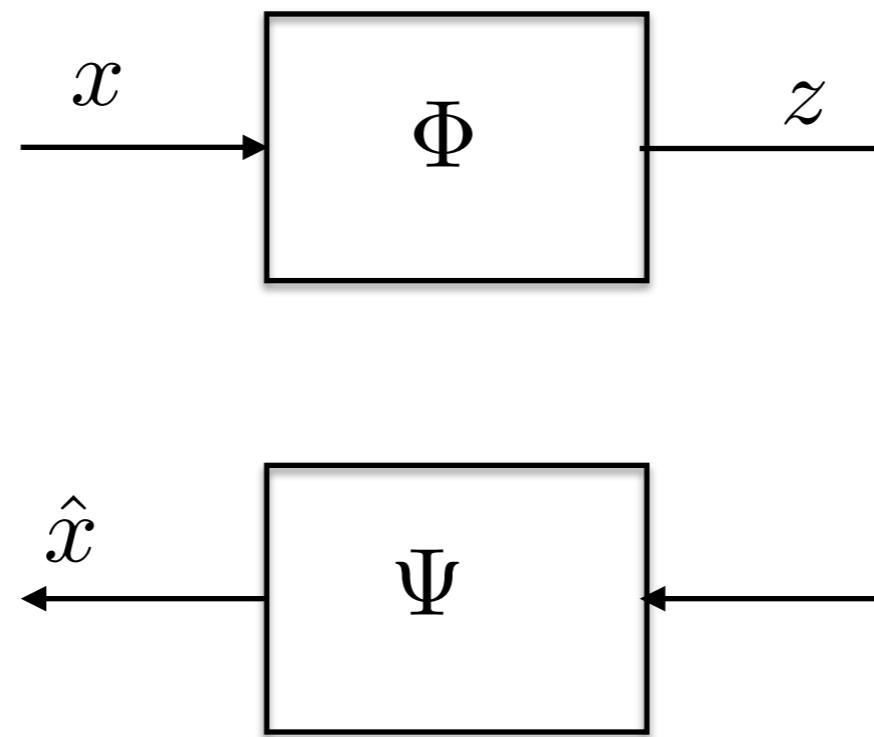
Review: Latent Graphical Models

- Latent Graphical Models or Mixtures.



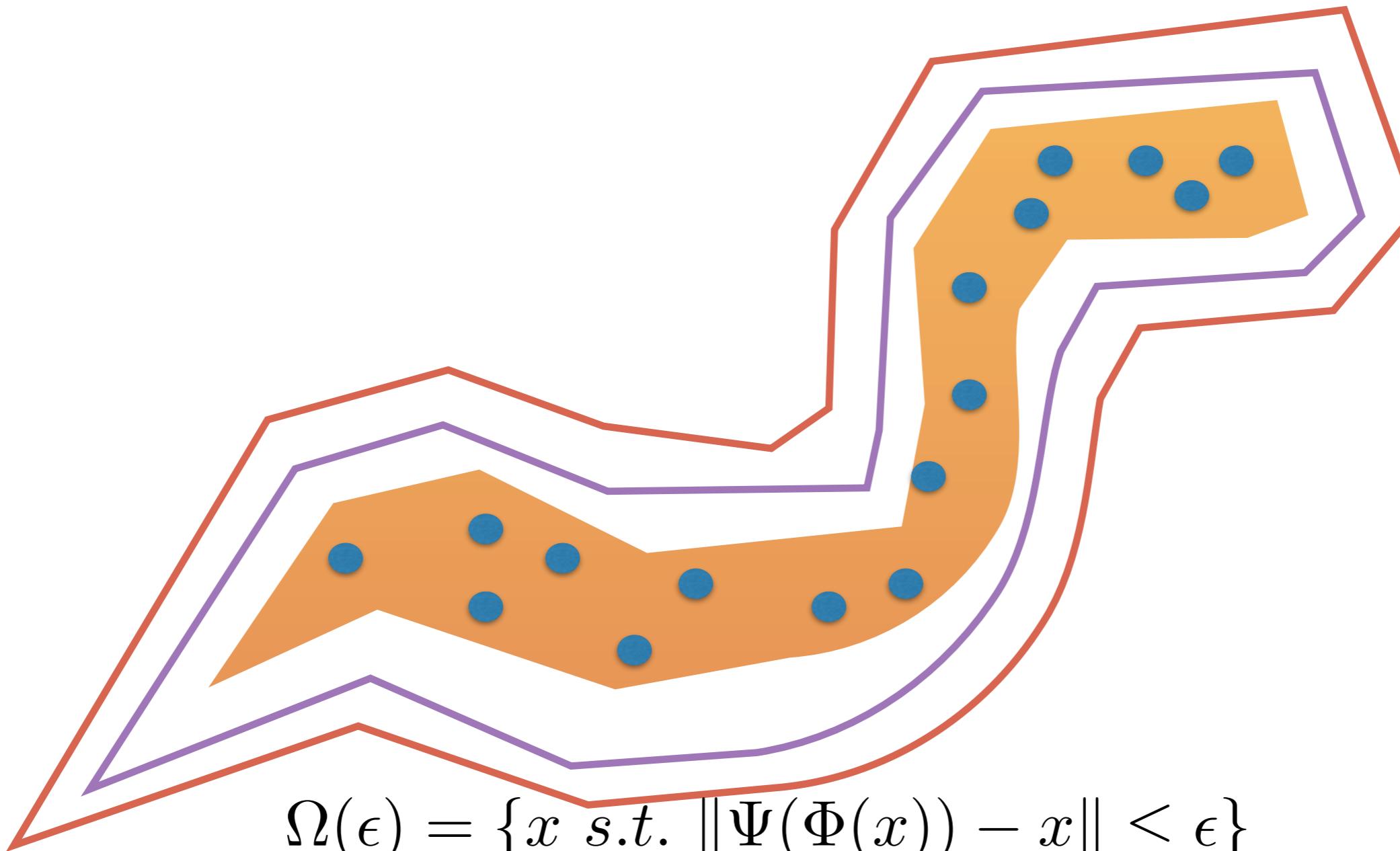
Review: Auto encoders

- Goal: given data $X = \{x_i\}$, learn a reparametrization $z_i = \Phi(x_i)$ that approximates X well with minimal capacity.



- The model contains an encoder Φ and a decoder Ψ .
- It introduces an *information bottleneck* to characterize input data from ambient space.

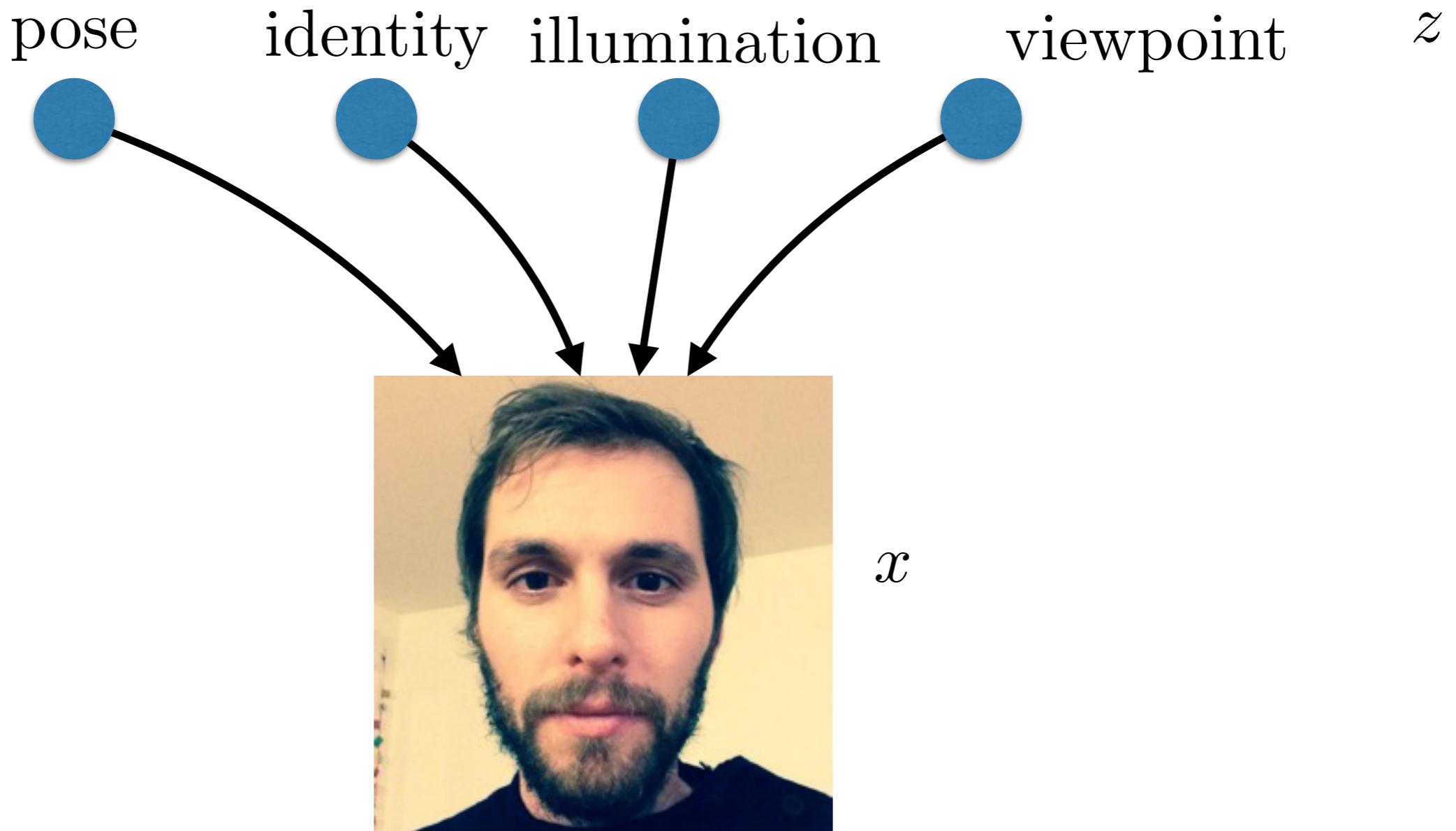
Review: Auto encoders Geometric Interpretation



- The reconstruction error approximates a distance to a covering manifold of X .
- Intrinsic manifold coordinates “disentangle” factors.

Review: Approximate Posterior Inference

- In latent graphical models, we can interpret latent variables as factors:



- How to infer z given x ?

Objectives

- Variational Inference Review
- Variational Autoencoders (contd)
- Examples and extensions
- Generative Adversarial Networks

The EM algorithm

- It is designed to find MLE solutions of latent variable models.
- In general, we have log-likelihoods of the form

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

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

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

- Then we update the parameters to maximize this likelihood: $\theta_{new} = \arg \max_{\theta} Q(\theta, \theta_{old})$.

EM and Variational Bound

- Q: Does this algorithm monotonically improve the likelihood?
- Assume for now that latent variables are discrete.
- For any distribution $q(Z)$ over latent variables, we have

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

(Jensen's Inequality: $\mathbb{E}(f(X)) \geq f(\mathbb{E}(X))$ if f is convex)

Variational Bound

- We can express the variational lower bound as

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

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

Variational Bound

- We can express the variational lower bound as

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

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

- Also, we have

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

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

is the Kullback-Leibler divergence.

Exponential Families

- The exponential family associated with ϕ is defined as the parametric family

$$p_\theta(x) = \exp\{\langle \theta, \phi(x) \rangle - A(\theta)\} , \text{ with}$$

$$A(\theta) = \log \int \exp\{\langle \theta, \phi(x) \rangle\} dx \quad \text{log-partition function}$$

- It is well defined for the family of parameters

$$\Omega = \{\theta ; A(\theta) < \infty\}$$

Conjugate Duality

- Conjugate duality representation of convex functions:

$$A^*(\mu) = \sup_{\theta \in \Omega} \{ \langle \mu, \theta \rangle - A(\theta) \}$$

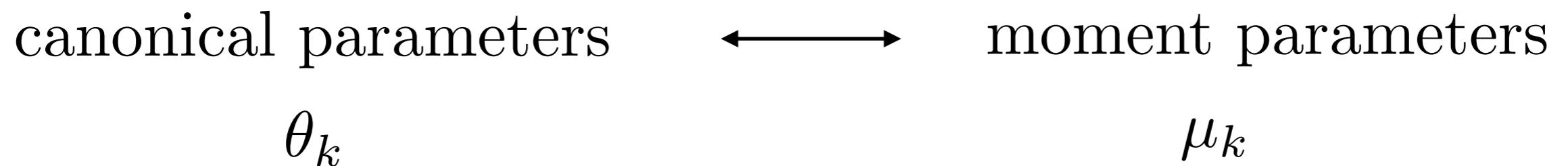
canonical parameters \longleftrightarrow moment parameters

$$\theta_k \qquad \qquad \mu_k$$

Conjugate Duality

- Conjugate duality representation of convex functions:

$$A^*(\mu) = \sup_{\theta \in \Omega} \{ \langle \mu, \theta \rangle - A(\theta) \}$$



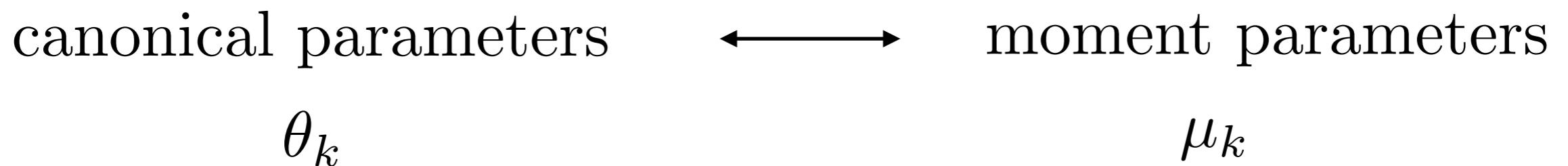
- Q: How to interpret the dual conjugate?

$A^*(\mu)$: Negative entropy of $p_{\theta(\mu)}$, where $p_{\theta(\mu)}$ is the exponential family distribution such that $\mathbb{E}_{\theta(\mu)} \phi(X) = \mu$.

Conjugate Duality

- Conjugate duality representation of convex functions:

$$A^*(\mu) = \sup_{\theta \in \Omega} \{ \langle \mu, \theta \rangle - A(\theta) \}$$



- Q: How to interpret the dual conjugate?

$A^*(\mu)$: Negative entropy of $p_{\theta(\mu)}$, where $p_{\theta(\mu)}$ is the exponential family distribution such that $\mathbb{E}_{\theta(\mu)} \phi(X) = \mu$.

- Variational representation: $A(\theta) = \sup_{\mu} \{ \langle \theta, \mu \rangle - A^*(\mu) \}$

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:

$$p(X, Z \mid \theta)$$

θ : generative model parameters

$$q(z|\beta)$$

Mean Field Variational Bayes

- Joint likelihood of observed and latent variables:

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

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

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

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

Mean Field Variational Bayes

- Joint likelihood of observed and latent variables:

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

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

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

- Mean-field approximation: we model hidden variables as being independent.
- Corresponding lower-bound is given by

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

Mean Field Variational Bayes

- **Goal:** optimize lower-bound with respect to variational parameters.
- As we have seen, this is equivalent to minimizing the divergence between true and approximate posterior:

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

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:

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

- 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

- Denote $f(Z) = \log p(X, Z|\theta)$

[Paiskey, Blei, Jordan, '12]

- 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

- Denote $f(Z) = \log p(X, Z|\theta)$

[Paiskey, Blei, Jordan, '12]

- 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 $\nabla_{\beta} \mathbb{E}_{q(z|\beta)} f(Z)$:

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

Mean Field Variational Bayes

- The estimator of the gradient is unbiased, but it may suffer from large variance.
- We may need a large number S of samples to stabilize the descent.
- Faster alternative?

Variational Autoencoders

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

- Recall the variational lower bound:

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

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

Variational Autoencoders

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

- Recall the variational lower bound:

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



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

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

Variational Autoencoders

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

- Recall the variational lower bound:

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



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

- Can we optimize jointly both generative and variational parameters efficiently?
- For appropriate posterior approximations, we can reparametrize samples as

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

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

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

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

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

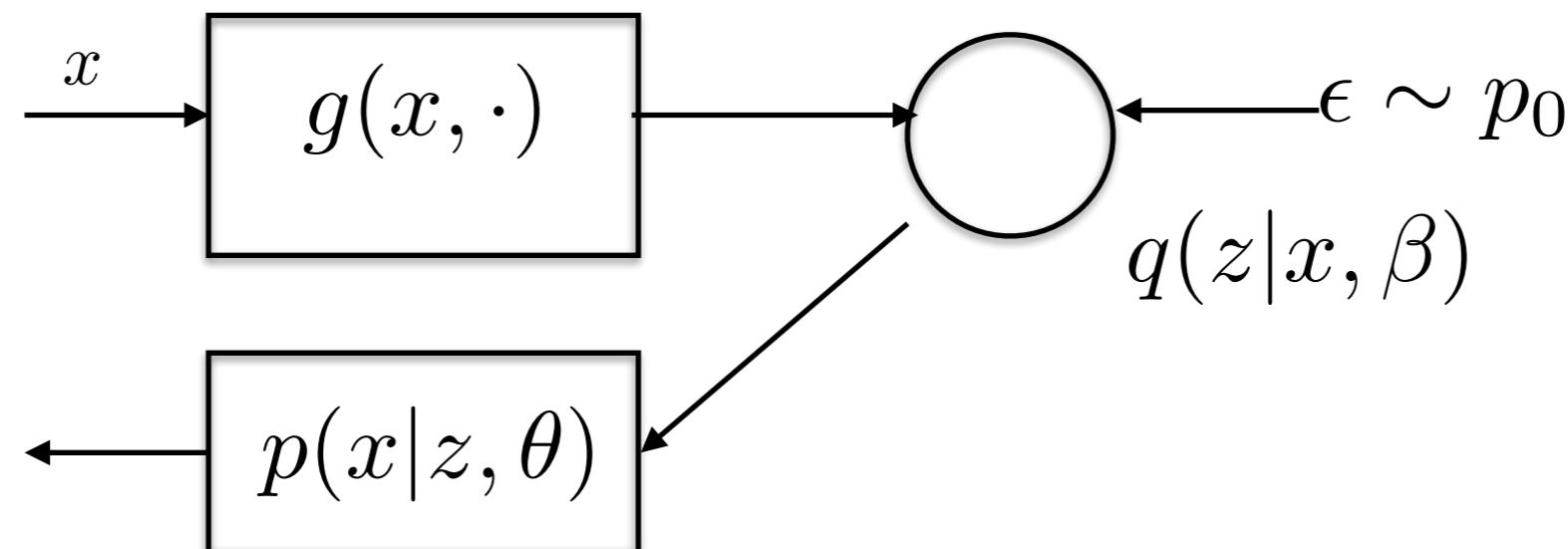
- First term acts as a *regularizer*: limits the capacity of the encoder
- Second term is a *reconstruction error*.

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:

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

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:

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

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

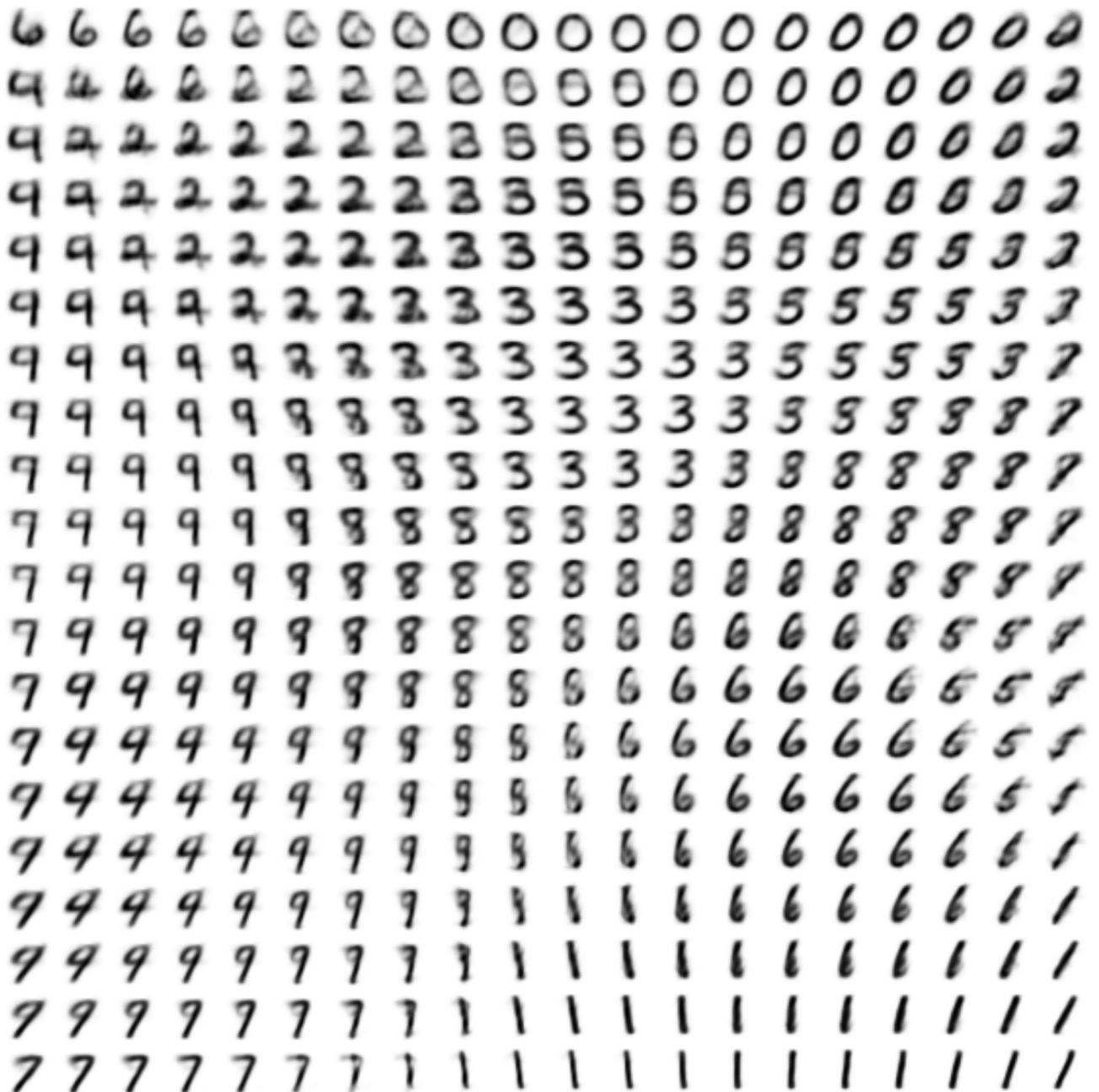
$$Z \sim q_\beta(z|x) \Leftrightarrow Z = \bar{\mu}(x) + \bar{\Sigma}(x)^{1/2}\epsilon, \quad \epsilon \sim \mathcal{N}(0, 1)$$

Variational Autoencoder

- Examples using a two-dimensional latent space:



(a) Learned Frey Face manifold



(b) Learned MNIST manifold

Examples

- Increasing latent dimensionality:

8 6 1 7 8 1 9 8 2 8	1 1 6 5 1 0 1 6 7 2	2 8 1 1 3 0 5 7 3 8	8 2 0 8 0 2 3 9 0 0
9 6 8 3 9 6 0 3 1 9	8 5 9 4 6 8 2 1 6 8	8 3 8 2 1 9 3 3 3 8	7 5 1 9 1 1 7 1 9 4
3 3 9 1 3 6 9 1 7 9	6 1 0 3 2 8 8 4 3 3	3 5 9 9 4 3 9 5 1 6	8 7 6 2 0 8 0 8 2 9
8 9 0 8 6 9 1 9 6 3	2 8 6 8 9 1 0 0 4 1	1 9 8 8 9 3 3 4 9 7	2 9 8 6 3 8 7 0 6 1
8 2 3 3 3 3 1 3 8 6	5 1 9 3 0 1 5 3 5 9	2 7 3 6 4 3 0 2 6 3	5 7 7 9 8 9 8 9 1 0
6 9 9 8 6 1 6 6 6 3	6 6 6 1 4 9 1 7 5 8	5 9 7 0 5 9 3 8 7 5	6 8 0 6 3 4 8 2 8 1
9 5 2 6 6 5 1 8 9 9	1 3 4 3 9 8 3 4 7 0	6 9 4 3 6 2 8 5 5 2	7 5 8 2 5 6 1 3 8 2
9 9 8 9 3 1 2 8 2 3	4 5 8 2 9 7 0 9 5 9	8 4 9 0 5 0 7 0 5 6 6	7 9 3 9 2 7 9 3 9 0
0 4 6 1 2 3 2 0 8 8	6 9 9 4 9 7 2 8 9 3	7 4 5 6 3 0 3 6 0 1	4 5 2 4 3 9 0 1 8 4
9 7 5 4 9 3 4 8 5 1	2 6 4 5 6 0 9 7 9 8	2 1 2 0 4 7 1 0 0 0	8 8 7 2 3 1 6 2 3 6

(a) 2-D latent space

(b) 5-D latent space

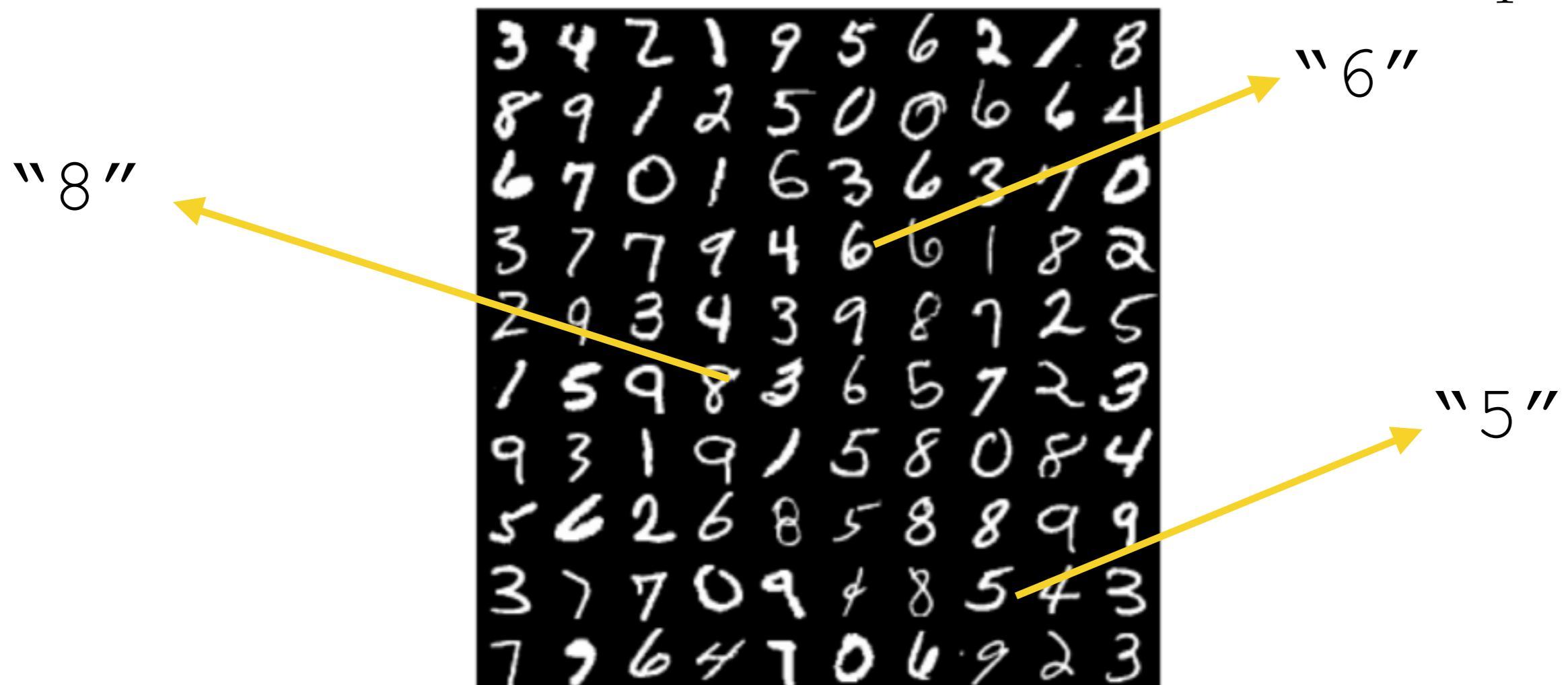
(c) 10-D latent space

(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 $x_i \sim p(x)$, $x_j \sim p(x)$.
 $L_1 \gg L_2$



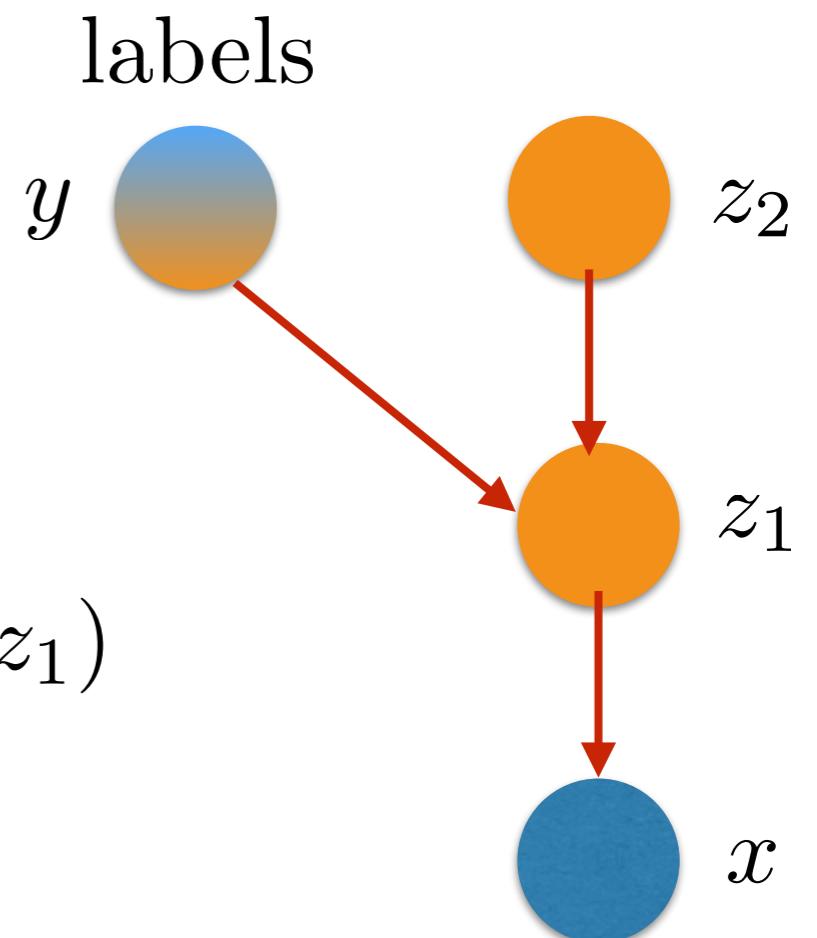
3	4	2	1	9	5	6	2	1	8
8	9	1	2	5	0	0	6	6	4
6	7	0	1	6	3	6	3	1	0
3	7	7	9	4	6	6	1	8	2
2	9	3	4	3	9	8	7	2	5
1	5	9	8	3	6	5	7	2	3
9	3	1	9	1	5	8	0	8	4
5	6	2	6	8	5	8	8	9	9
3	7	7	0	9	4	8	5	4	3
7	9	6	4	7	0	6	9	2	3

The grid contains 100 handwritten digits. Three specific digits are highlighted with yellow arrows pointing to them:

- An arrow labeled "8" points to the digit "8" in the first row, third column.
- An arrow labeled "6" points to the digit "6" in the third row, fourth column.
- An arrow labeled "5" points to the digit "5" in the fifth row, second column.

Extension 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:

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

- For datapoint with no labels:

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

Extension to Semi-Supervised Learning

- “Semi-supervised Learning with Deep Generative Networks”, Kingma et al,’14.
- Classification results on MNIST:

Table 1: Benchmark results of semi-supervised classification on MNIST with few labels.

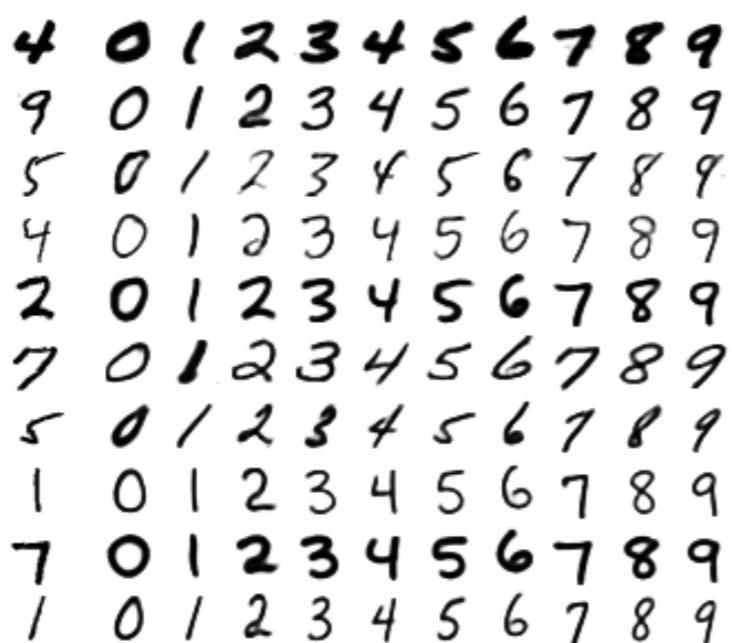
N	NN	CNN	TSVM	CAE	MTC	AtlasRBF	M1+TSVM	M2	M1+M2
100	25.81	22.98	16.81	13.47	12.03	8.10 (± 0.95)	11.82 (± 0.25)	11.97 (± 1.71)	3.33 (± 0.14)
600	11.44	7.68	6.16	6.3	5.13	–	5.72 (± 0.049)	4.94 (± 0.13)	2.59 (± 0.05)
1000	10.7	6.45	5.38	4.77	3.64	3.68 (± 0.12)	4.24 (± 0.07)	3.60 (± 0.56)	2.40 (± 0.02)
3000	6.04	3.35	3.45	3.22	2.57	–	3.49 (± 0.04)	3.92 (± 0.63)	2.18 (± 0.04)

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 \mathbf{z}



(b) MNIST analogies



(c) SVHN analogies

Other extensions

- Incorporate MCMC steps into the variational approximation:

“Markov Chain Monte Carlo and Variational Inference: Bridging the Gap”, Salimans et al’15

- Incorporate Importance Sampling to improve the variational lower bound:
“Importance Weighted Autoencoder”, Burda et al’16

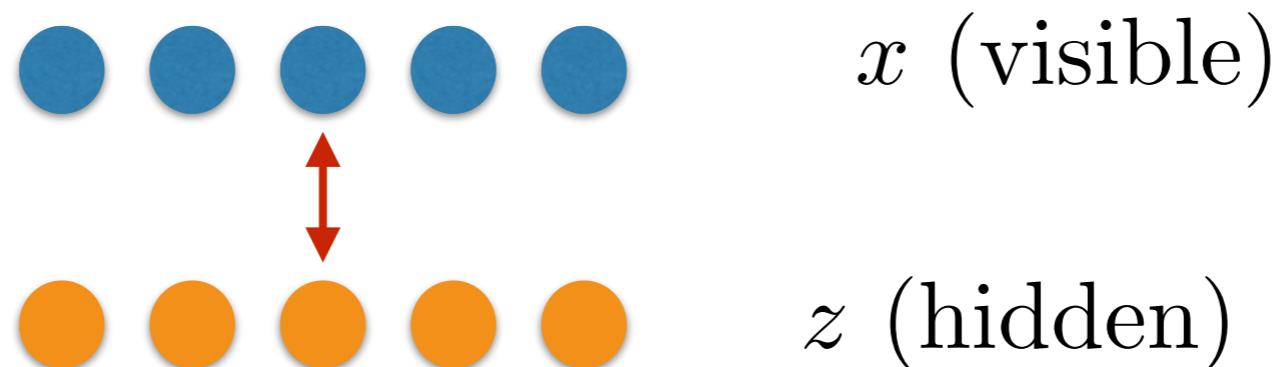
$$\mathcal{L}_k(x) = \mathbb{E}_{z_1, \dots, z_k \sim q(z|x)} \left[\log \frac{1}{k} \sum_{i=1}^k \frac{p(x, z_i)}{q(z_i|x)} \right].$$

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

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

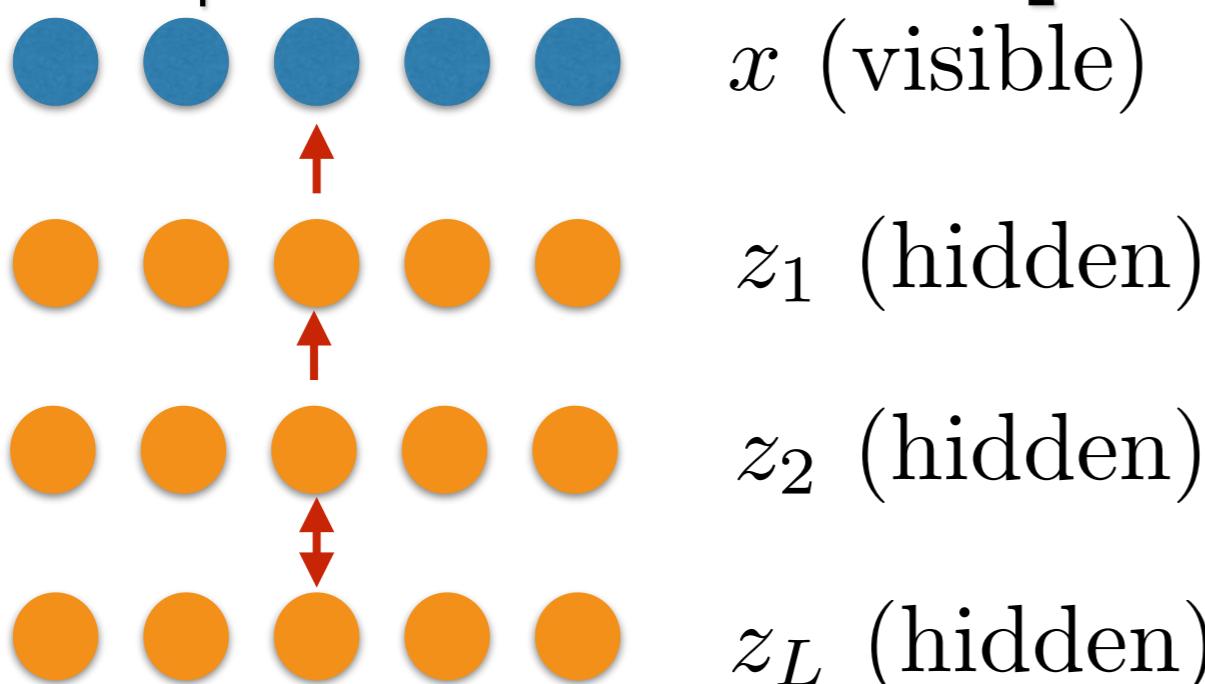
Other directed models

- Restricted Boltzmann Machines [Smolenski'86,Hinton,'02] are undirected graphical models with binary variables



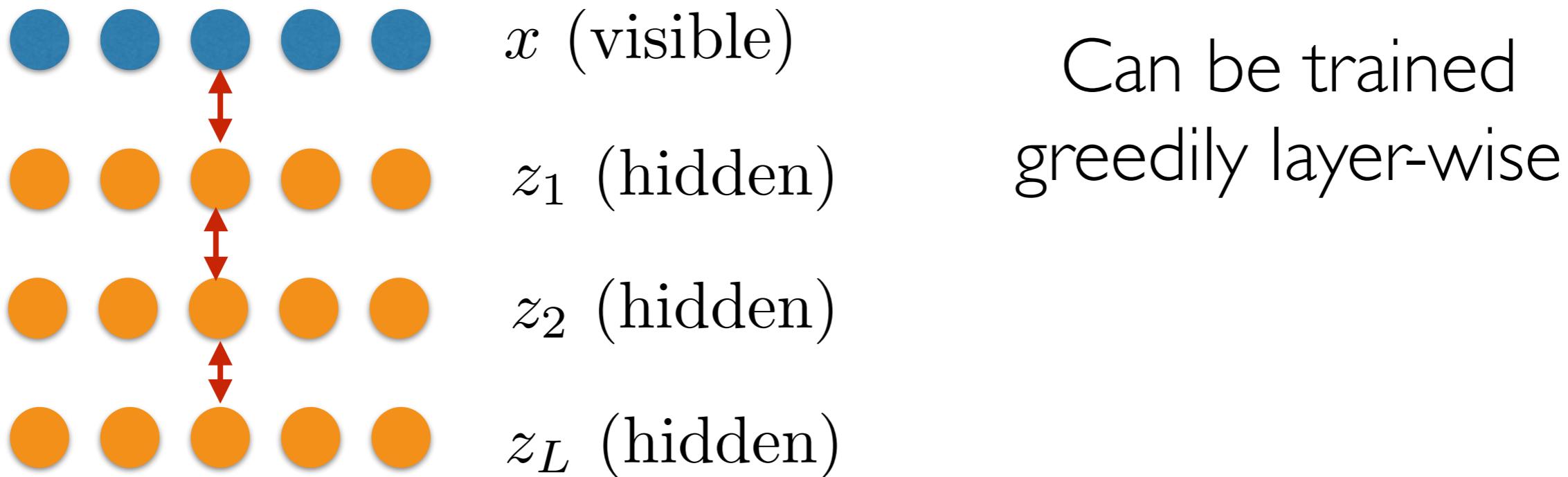
$$p(x, z) = \exp \left(\langle \theta_1, xz^T \rangle + \langle \theta_2, x \rangle + \langle \theta_3, z \rangle - \log A(\theta) \right)$$

- Deep Belief Networks [Hinton et al'02]



Other directed models

- Deep Boltzmann Machines [Saladutnikov & Hinton,'09]



- See also:
 - Wake-Sleep [Hinton et al'95]
 - Generative Stochastic Networks [Bengio,'13].
 - ...

Limits of Mixture Models

- Inference can be computationally expensive for large models.
- The modeling $p(x)$ is reduced to the task of modeling $p(x|z)$
- Q: How to account for image variability?

- $p(x|z) = \mathcal{N}(\Phi(z), \Sigma(z))$ corresponds to a model of *additive variability*:

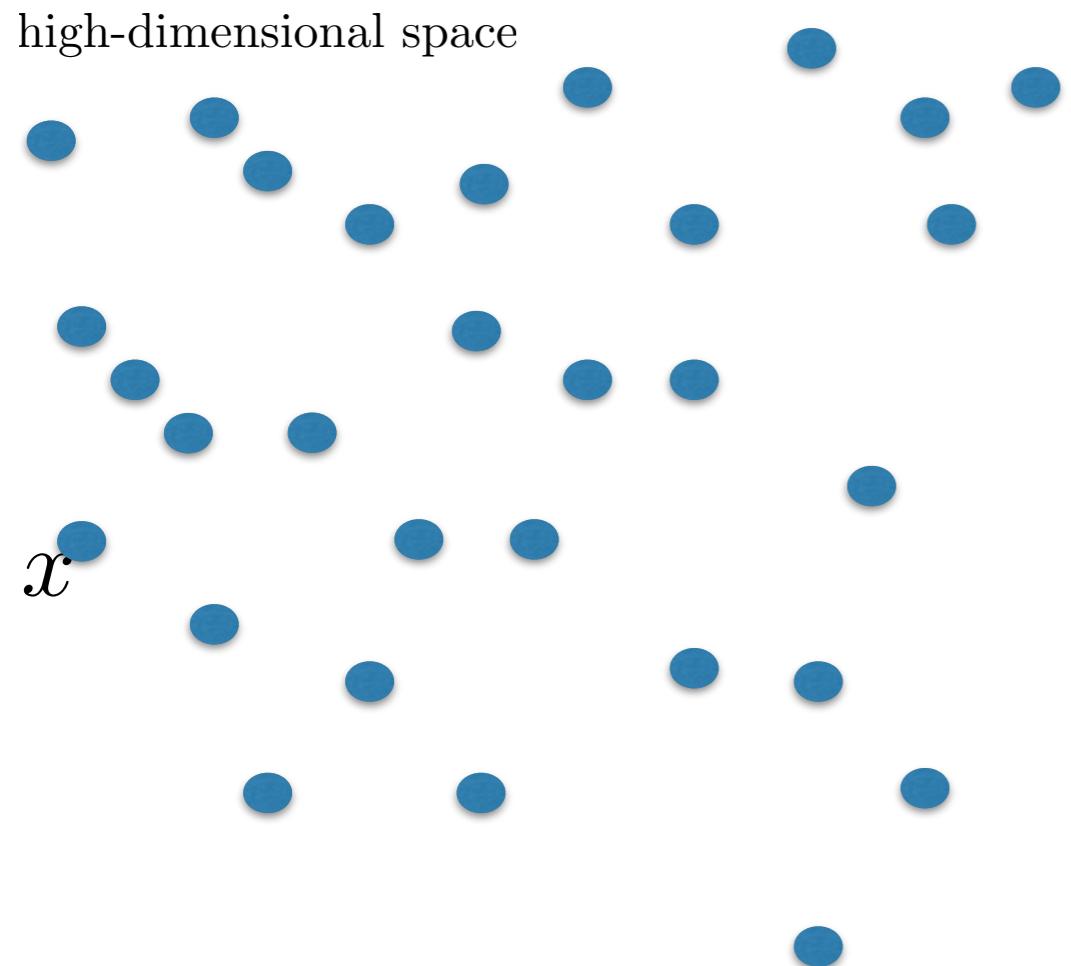
$$x = \Phi(z) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \Sigma(z))$$

$$-\log p(x|z) \propto \|\Sigma(z)^{-1/2}(x - \Phi(z))\|^2$$

- In particular, can we guarantee that $|p(x_\tau) - p(x)| \lesssim \|\tau\|$ with a mixture model?
- Gaussian likelihoods tend to suffer from regression to the mean.

Generative Models of Complex data

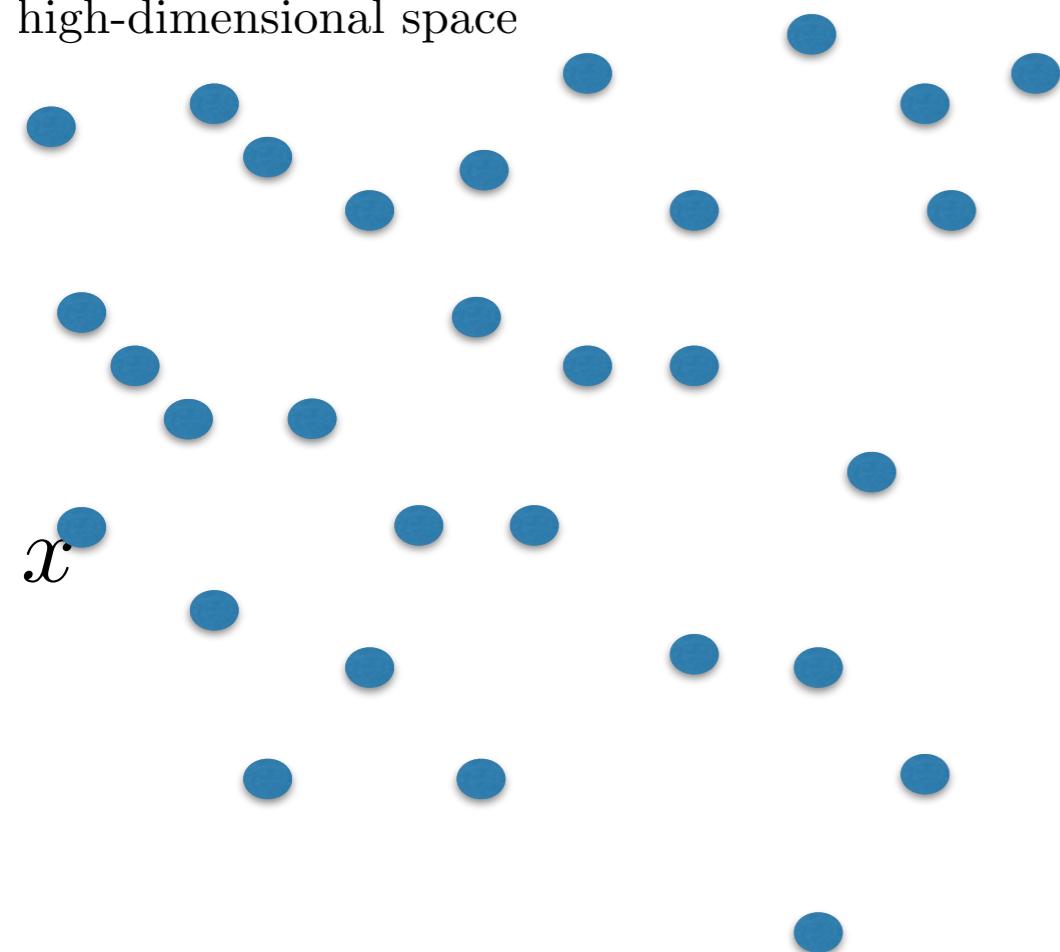
- Flows or Transports of Measure:



Generative Models of Complex data

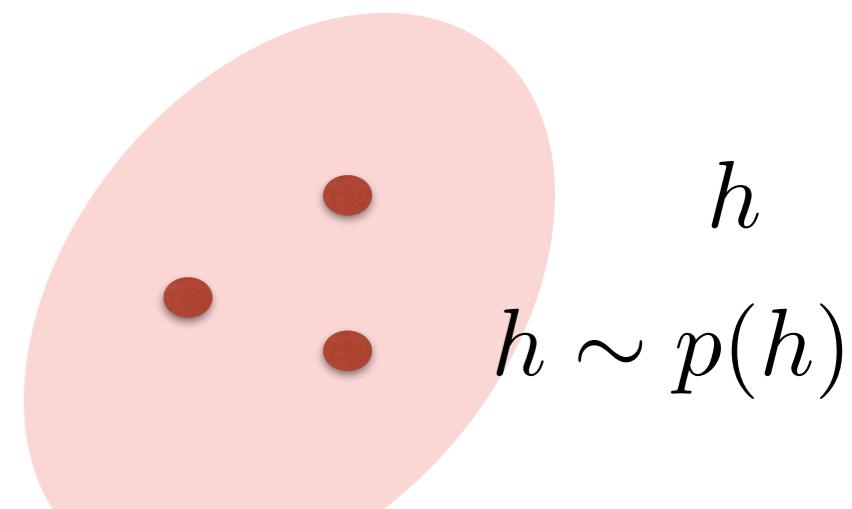
- Flows or Transports of Measure:

high-dimensional space



x

latent space



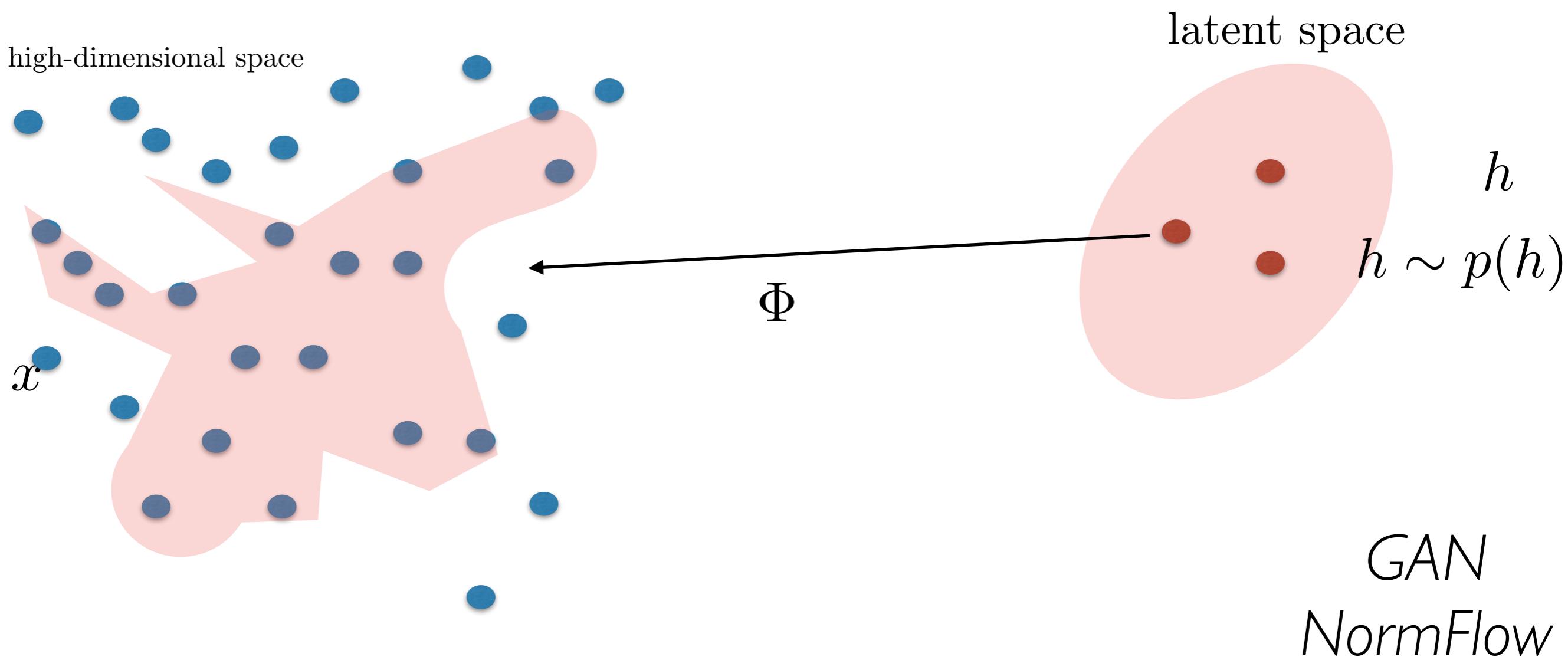
GAN

NormFlow

...

Generative Models of Complex data

- Flows or Transports of Measure



$p(x)$ defined implicitly with

$$\int f(x)p(x)dx = \int f(\Phi(h))p(h)dh , \quad \forall f \text{ measurable}$$

Measure Transports

- How to train the transport Φ ?
- We will see two methods:
 - Directly by optimizing data log-likelihood [Normalizing Flows]
 - Using a Discriminative Model [Generative Adversarial Networks]

Normalizing Flows

[Variational Inference with Normalizing Flows, Rezende & Mohamed'15]

- Consider a diffeomorphism $\Phi : \mathbb{R}^N \rightarrow \mathbb{R}^N$. [Tabak et al.'10]
- If $z \in \mathbb{R}^N$ is a random variable with density $q(z)$, what is the density of $z' = \Phi(z)$?
- We have, for any measurable f ,

$$\begin{aligned}\mathbb{E}_{z \sim q}(f(z')) &= \int f(z')q(z)dz \\ &= \int f(\Phi(z))q(z)dz = \int f(z)q(\Phi^{-1}(z))|\det(\nabla\Phi^{-1}(z))|dz \\ &= \int f(z)\tilde{q}(z)dz = \mathbb{E}_{z' \sim \tilde{q}}(f(z')) , \text{ with}\end{aligned}$$

$$\tilde{q}(z') = q(z) |\det \nabla\Phi(z)|^{-1} , \quad z = \Phi^{-1}(z') .$$

Normalizing Flows

- The density $q_K(z)$ obtained by transporting a base measure q_0 through a cascade of K diffeomorphisms Φ_1, \dots, Φ_K is

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

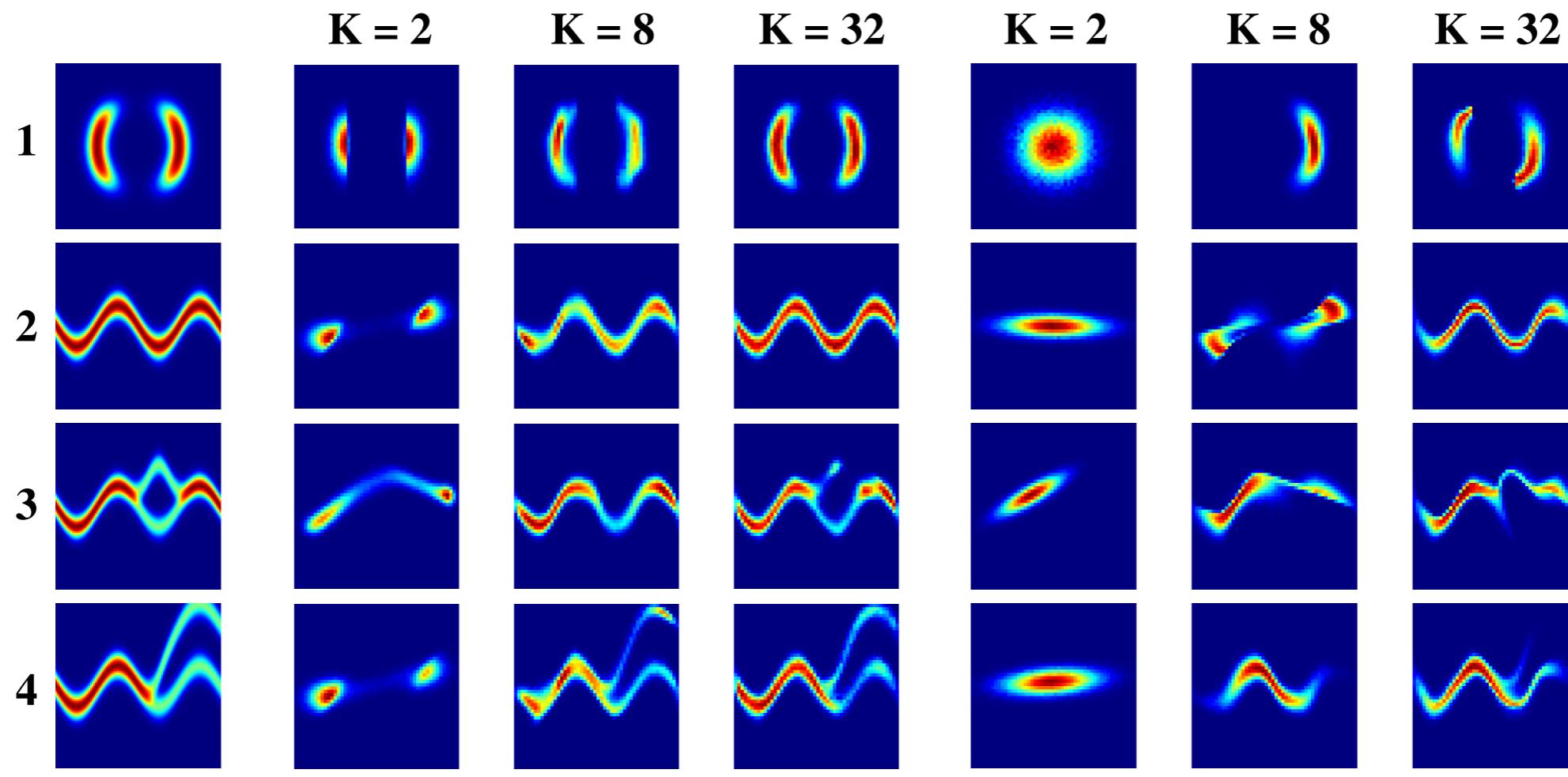
$$\log q_K(z) = \log q_0(z_0) - \sum_{k=1}^K \log |\det \nabla_{z_k} \Phi_k| .$$

- 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].

Normalizing Flows

- Some low-dimensional transport results:

[Rezende et al.'15]



(a)

(b) Norm. Flow

(c) NICE

Diffusion and Non-equilibrium Thermodynamics

[Sohl-Dickstein et al.'15]

- We can also consider *infinitesimal* flows:

$$\frac{\partial q_t(z)}{\partial t} = \mathcal{F}(q_t(z)) , \quad q_0(z) = p_0(z) .$$

\mathcal{F} describes the dynamics.

Diffusion and Non-equilibrium Thermodynamics

[Sohl-Dickstein et al.'15]

- We can also consider *infinitesimal* flows:

$$\frac{\partial q_t(z)}{\partial t} = \mathcal{F}(q_t(z)) , \quad q_0(z) = p_0(z) .$$

\mathcal{F} describes the dynamics.

- For $\mathcal{F} = -\Delta$ we have Gaussian diffusion.

It defines a Markov diffusion kernel that successively transforms data distribution $p_0(x)$ into a tractable distribution $\pi(x)$:

$$\pi(x) = \int T_\pi(x|x')\pi(x')dx'$$

$$q(x^{(t+1)}|x^{(t)}) = T_\pi(x^{(t+1)}|x^{(t)}, \beta_t) \quad \beta_t: \text{diffusion rate.}$$

Diffusion and Non-equilibrium Thermodynamics

[Sohl-Dickstein et al.'15]

- The “forward” trajectory diffuses the data distribution into a tractable distribution, eg Gaussian.

Diffusion and Non-equilibrium Thermodynamics

[Sohl-Dickstein et al.'15]

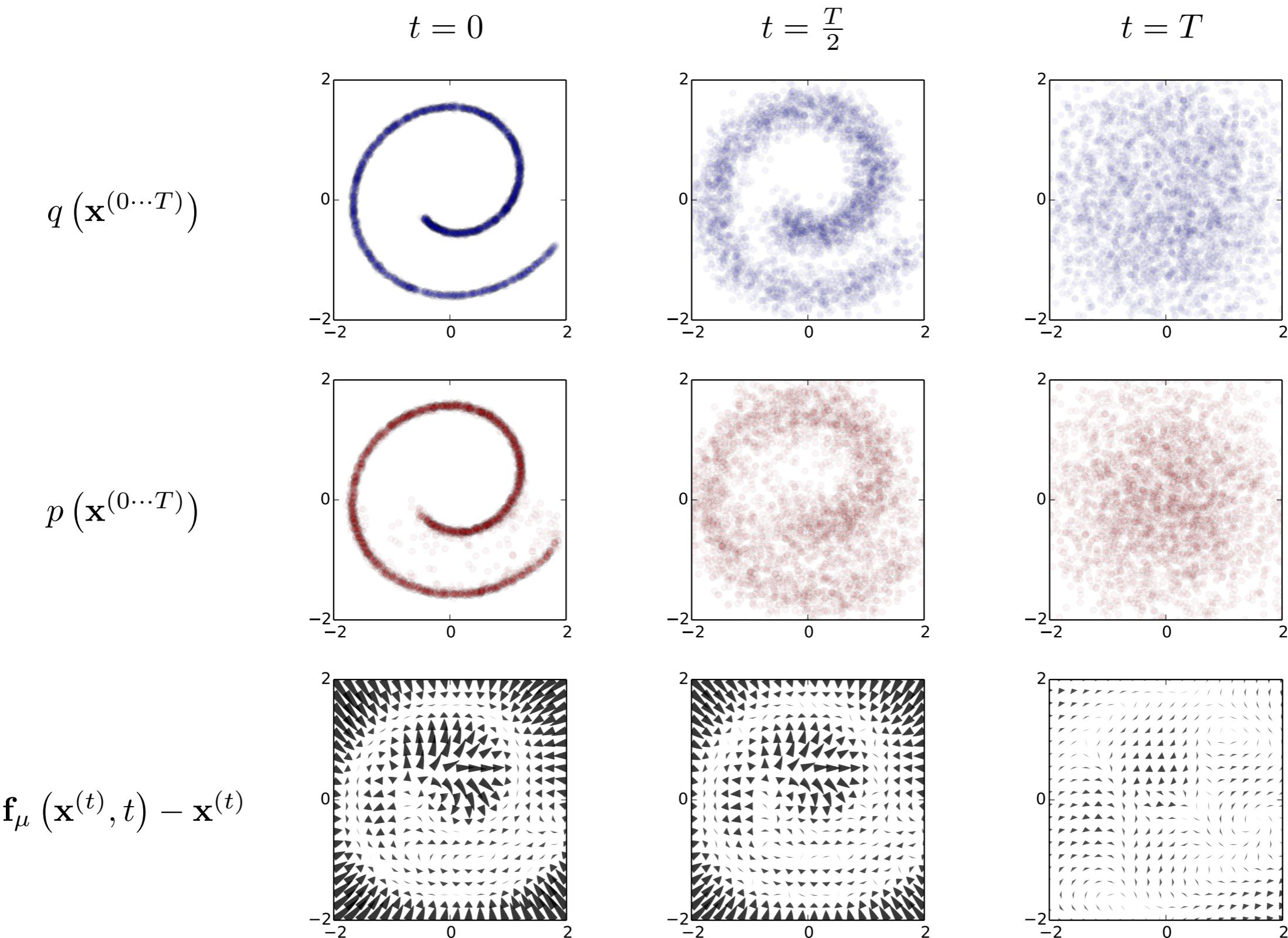
- The “forward” trajectory diffuses the data distribution into a tractable distribution, eg Gaussian.
- The generative model learns how to reverse the diffusion:

$$p(x^{(0\dots T)}) = p(x^{(T)}) \prod_{t \leq T} p(x^{(t-1)} | x^{(t)}) .$$

- in the limit of infinitesimal diffusion, the forward and backward kernel have the same functional form (Gaussian).
- The parameters of the model are $\{\mu(x^{(t)}, t), \Sigma(x^{(t)}, t)\}_{t \leq T}$.
- The data likelihood admits lower bound that can be evaluated efficiently using annealed importance sampling.

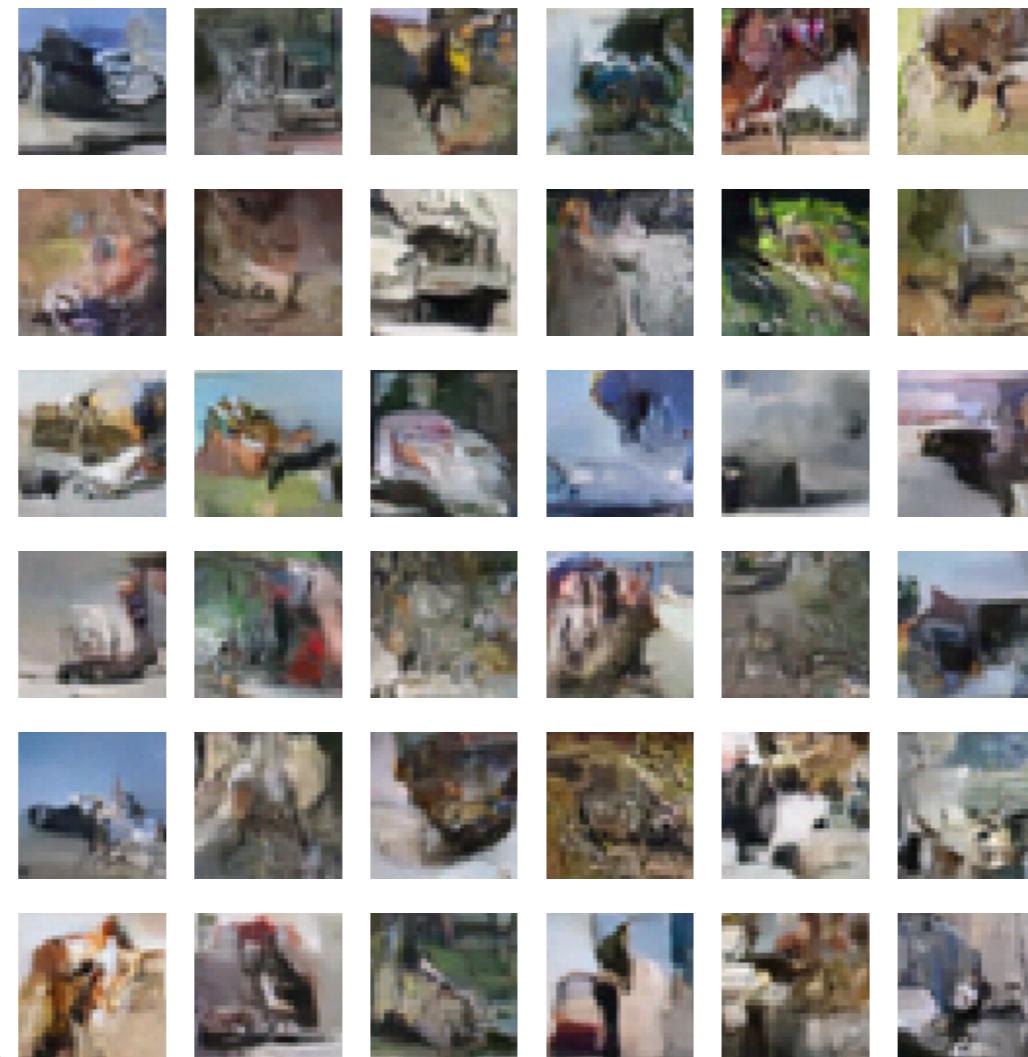
Diffusion and Non-equilibrium Thermodynamics

[Sohl-Dickstein et al.'15]

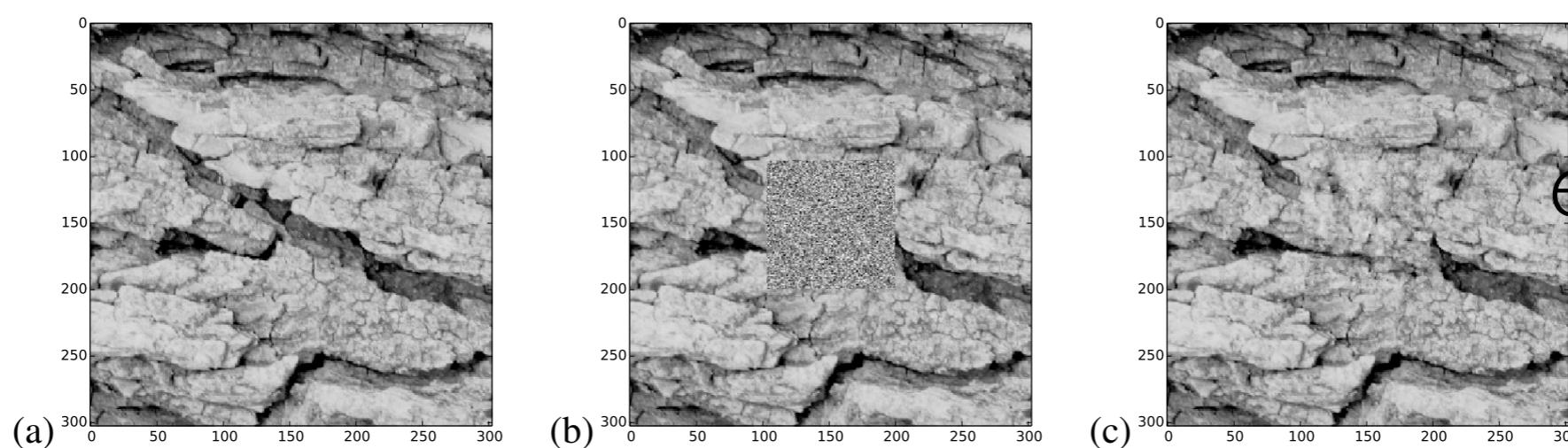


Diffusion and Non-equilibrium Thermodynamics

[Sohl-Dickstein et al.'15]



samples
from the model
trained on
CIFAR-10



inpainting
experiments

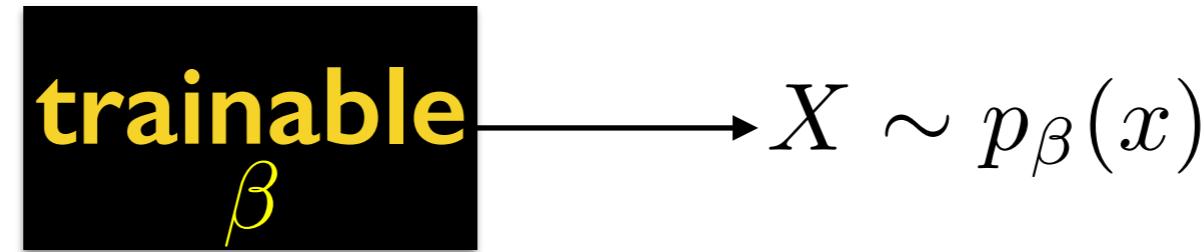
Generative Adversarial Networks

[Goodfellow et al., '14]

Generative Adversarial Networks

[Goodfellow et al., '14]

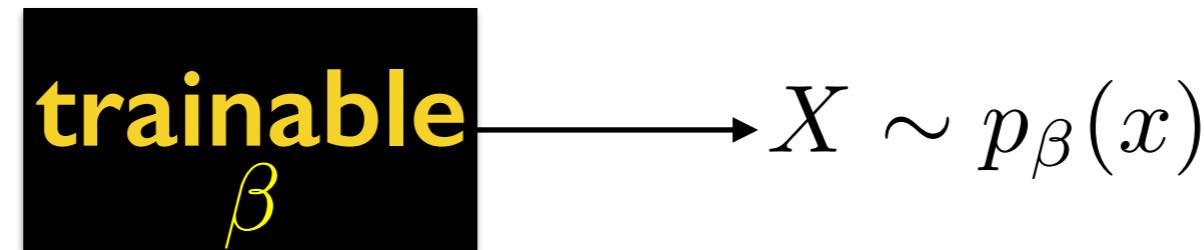
- Suppose we have a *trainable* black box generator:



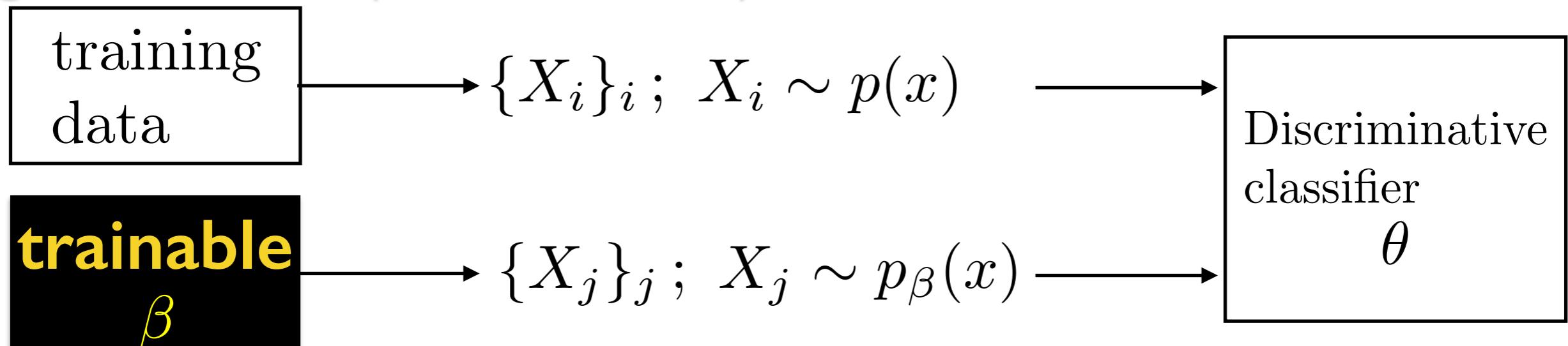
Generative Adversarial Networks

[Goodfellow et al., '14]

- Suppose we have a *trainable* black box generator:



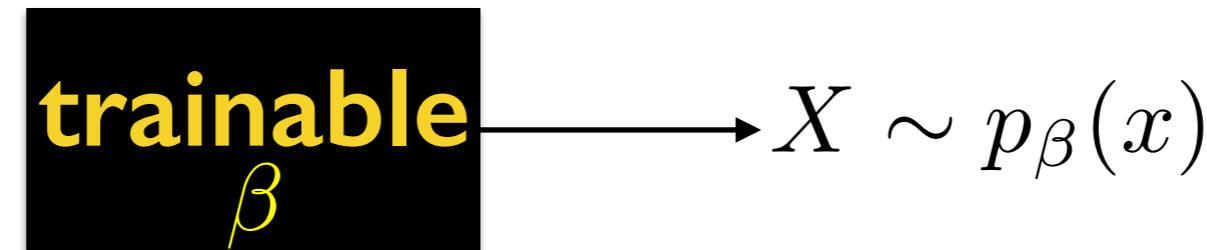
- Given observed data $\{X_i\}_i$; $X_i \sim p(x)$, how to force our generator to produce samples from $p(x)$?



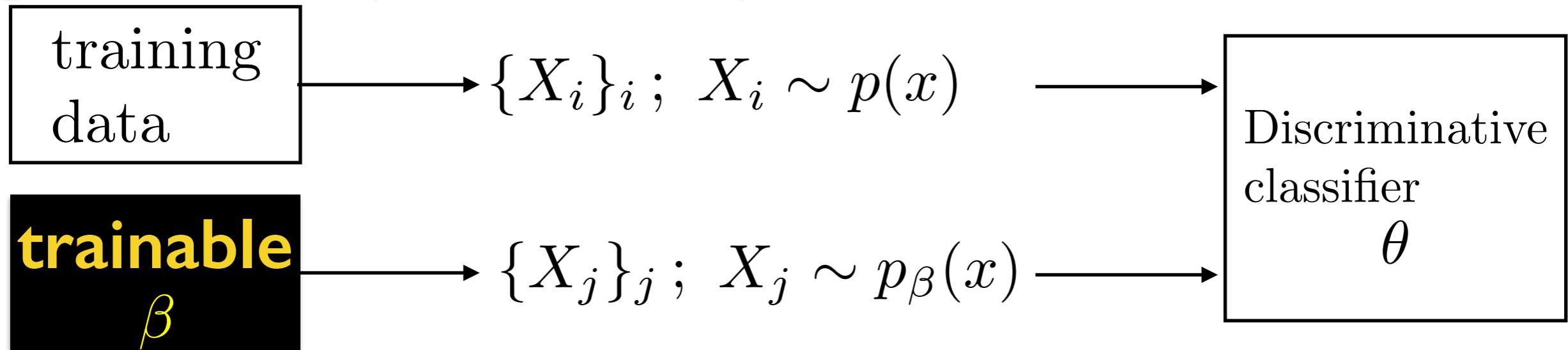
Generative Adversarial Networks

[Goodfellow et al., '14]

- Suppose we have a *trainable* black box generator:



- Given observed data $\{X_i\}_i$; $X_i \sim p(x)$, how to force our generator to produce samples from $p(x)$?

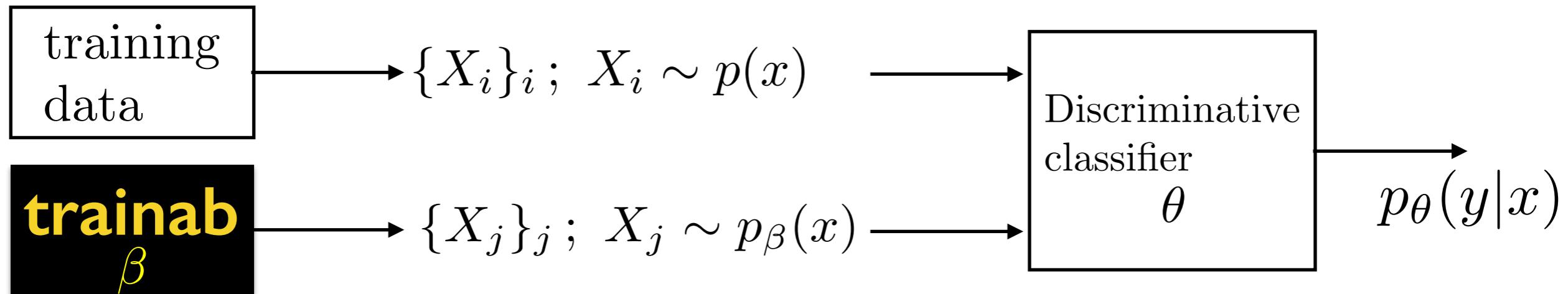


- The generator should make the classification task as hard as possible for any discriminator.

Generative Adversarial Networks

[Goodfellow et al., '14]

- Train generator and discriminator in a minimax setting:



$y = 1$: “real” samples

$y = 0$: “fake” samples

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

Generative Adversarial Networks

- Q: Do we have consistency? (in the limit of infinite capacity)

Generative Adversarial Networks

- Q: Do we have consistency? (in the limit of infinite capacity)

Given current p_β and p_{data} , the optimum discriminator is given by

$$D(x) = p(y = 1|x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_\beta(x)} .$$

For each x ,

$$p_{\text{data}}(x) \log D(x) + p_\beta(x) \log(1 - D(x)) = (p_{\text{data}}(x) + p_\beta(x)) (\alpha \log \gamma + (1 - \alpha) \log(1 - \gamma)) ,$$

$$\alpha = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_\beta(x)} , \quad \gamma = D(x) .$$

But

$$\alpha \log \gamma + (1 - \alpha) \log(1 - \gamma) = -H(\bar{\alpha}) - D_{KL}(\bar{\alpha}||p(y|x)) \leq -H(\bar{\alpha})$$

Generative Adversarial Networks

- It results that
 $\min -H(\bar{\alpha})$ is attained when $\alpha = 1/2$, thus
$$p_\beta(x) = p_{data}(x)$$
- In practice, however, we parametrize both generator and discriminator using neural networks.
- Optimize the cost using gradient descent.

Generative Adversarial Training

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

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

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

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

Generative Adversarial Training

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

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

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

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

- Indeed,
- Numerical approach: alternate k steps of discriminator update with 1 step of generator update.

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:

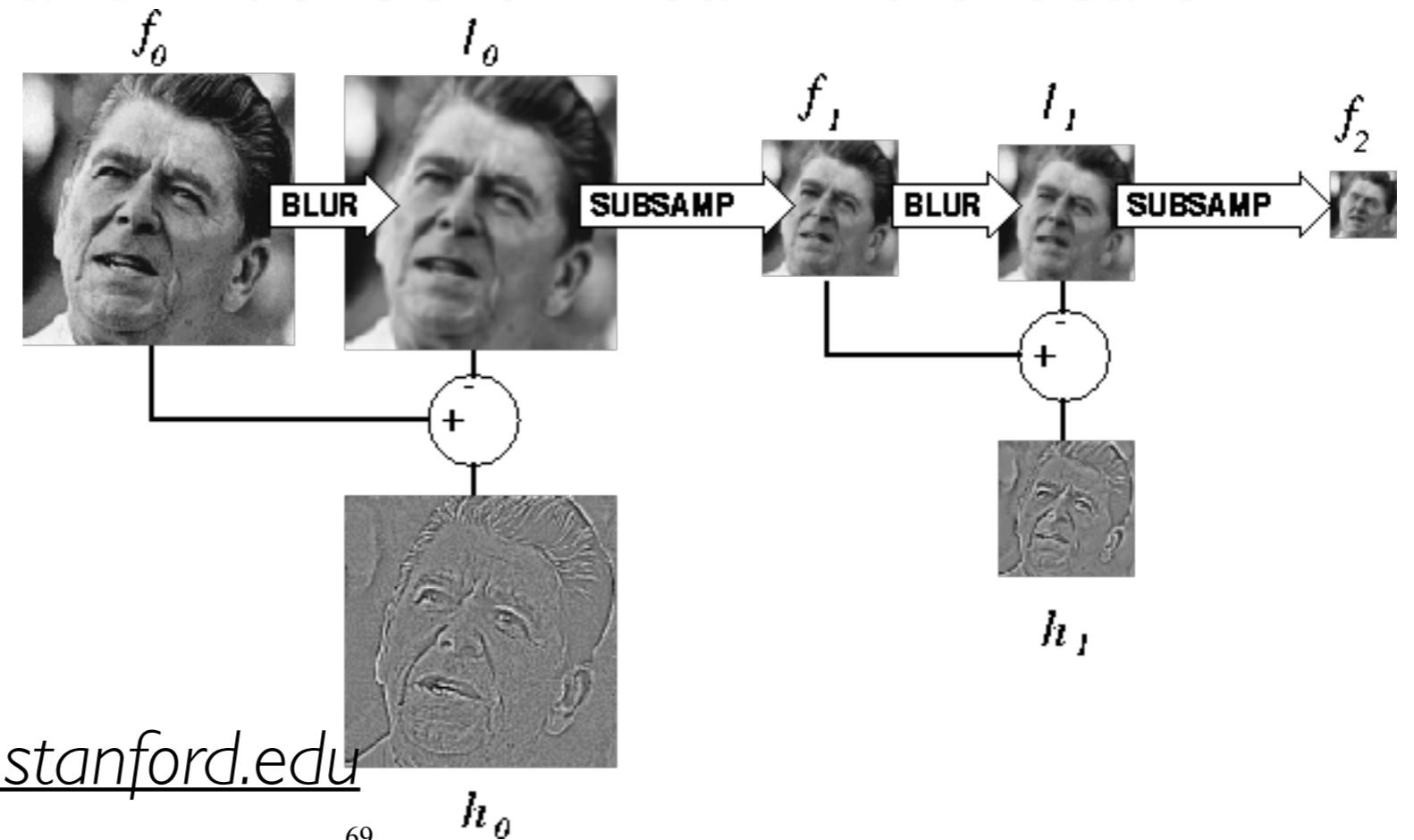
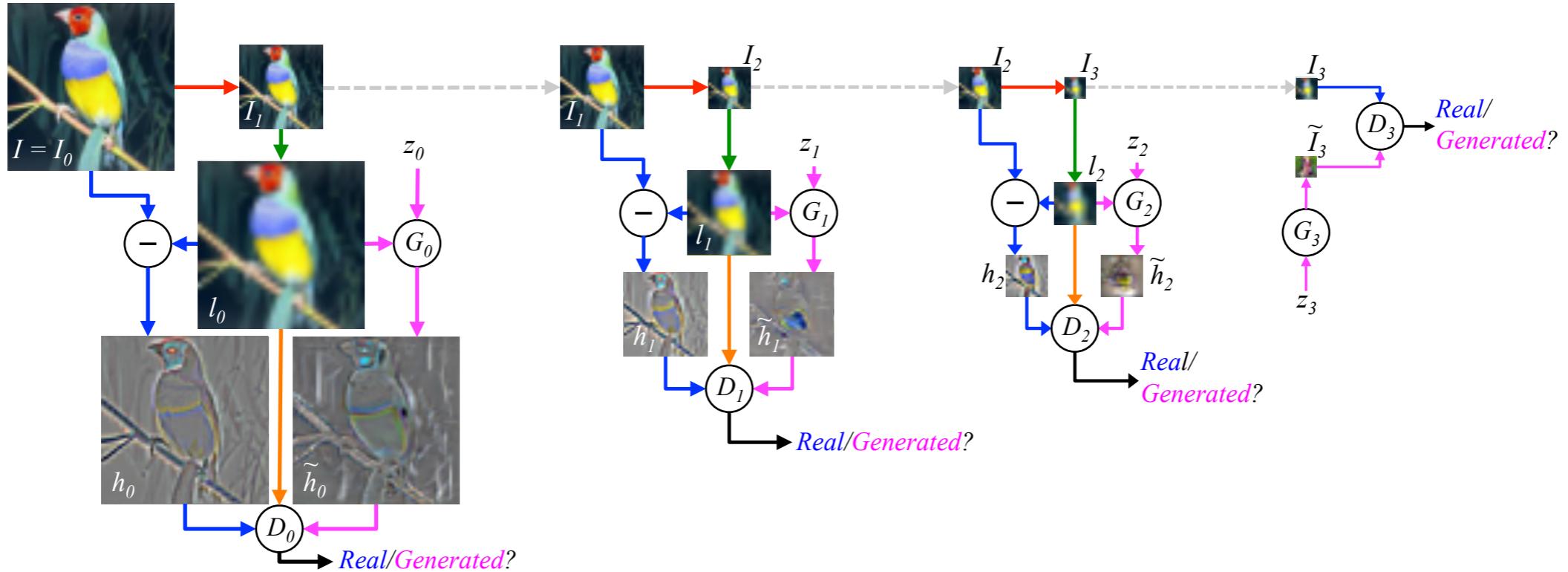


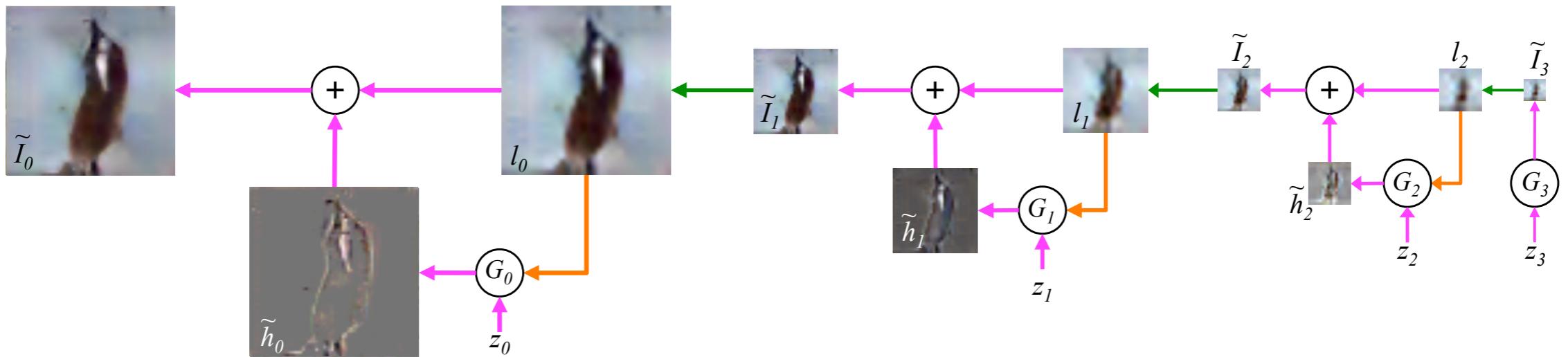
figure source: <http://sepwww.stanford.edu>

LAPGAN

- Training procedure:

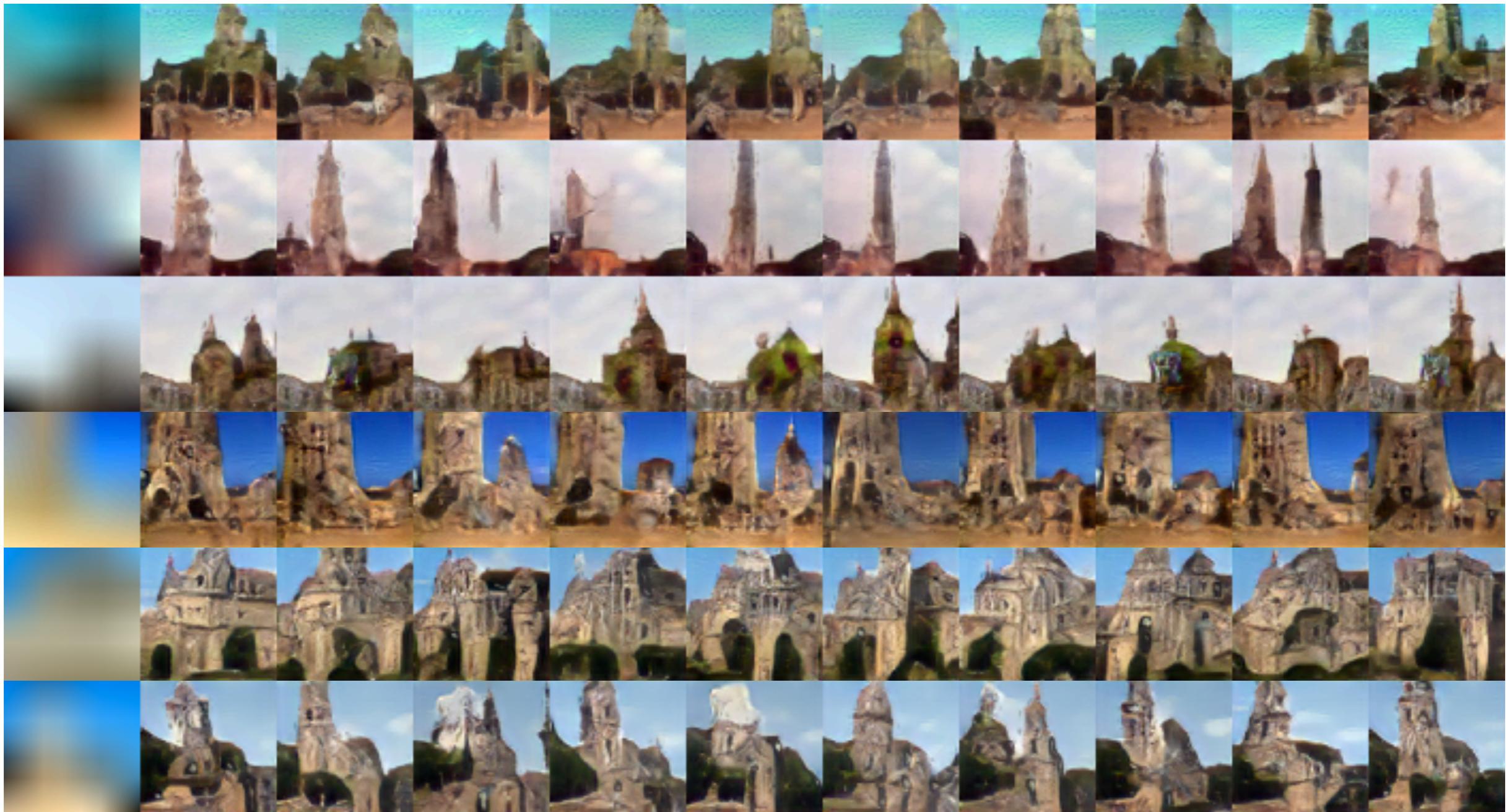


- Sampling procedure:



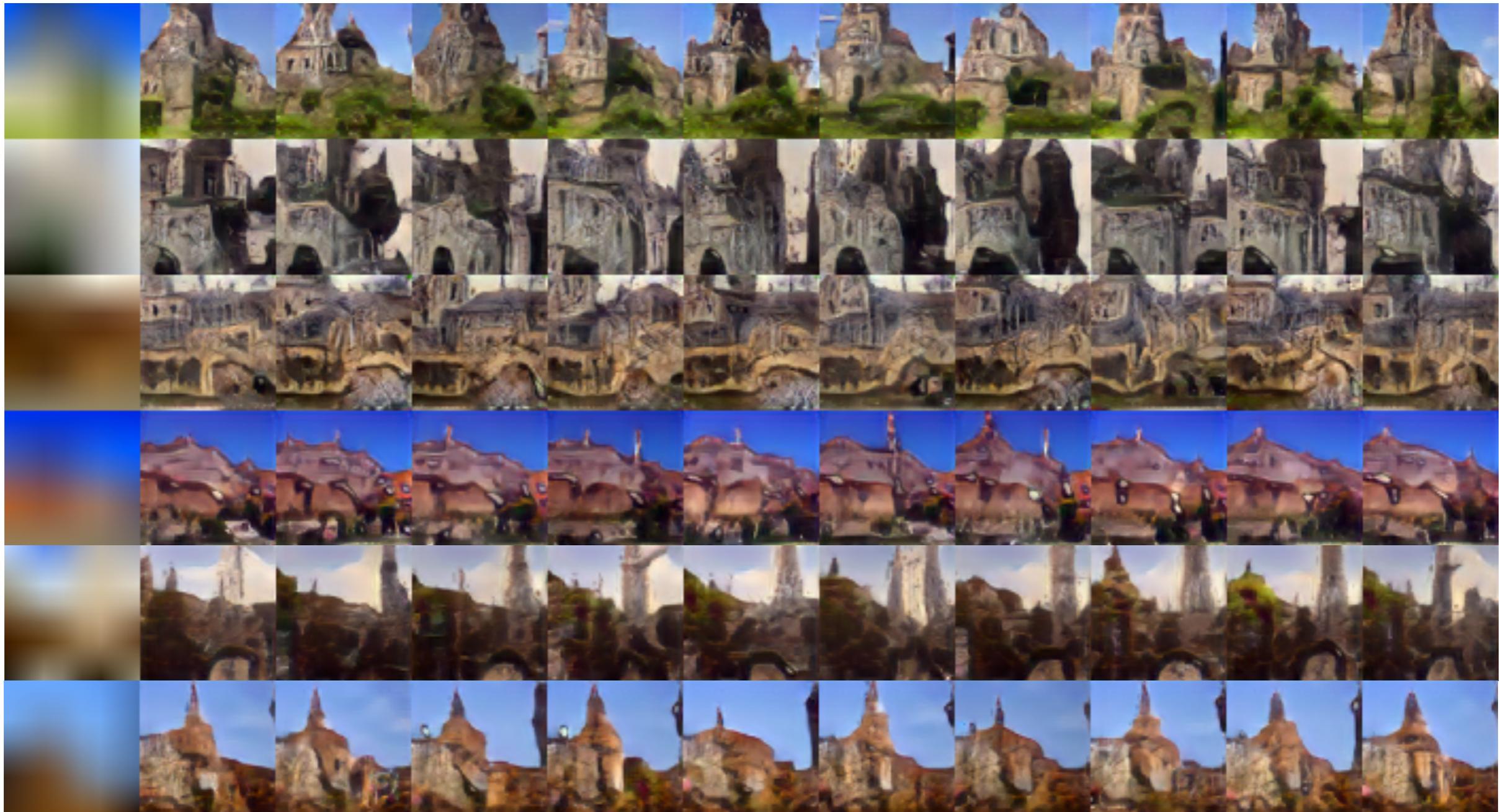
LAPGAN

- Samples generated from the model:



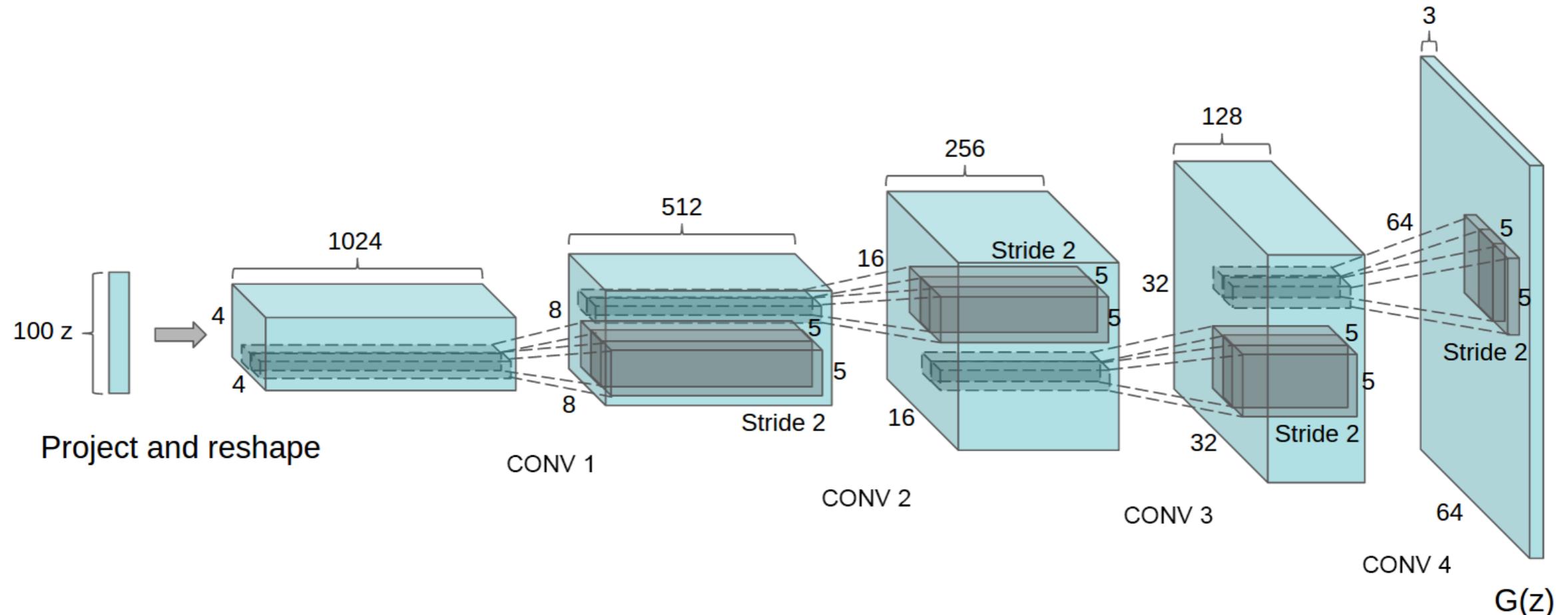
LAPGAN

- Samples generated from the model:



DC-GAN

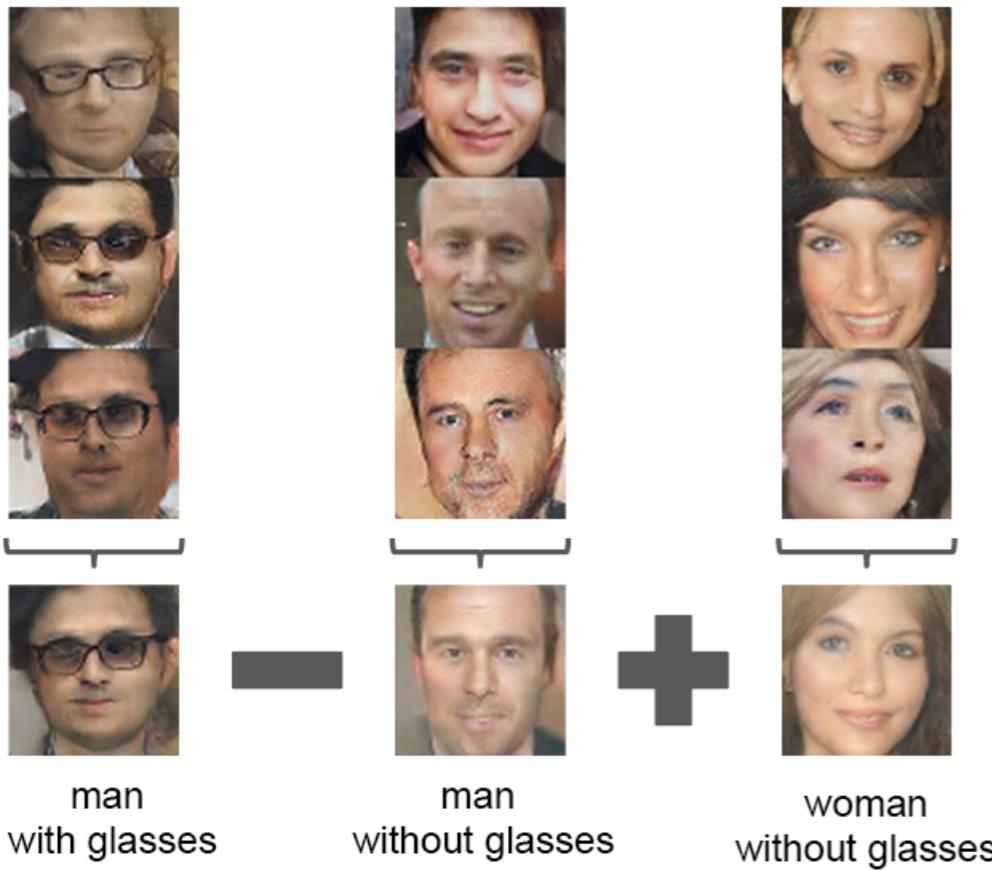
- Improved multi-scale architecture and Batch-Normalization: [Radford et al.'16]



DC-GAN

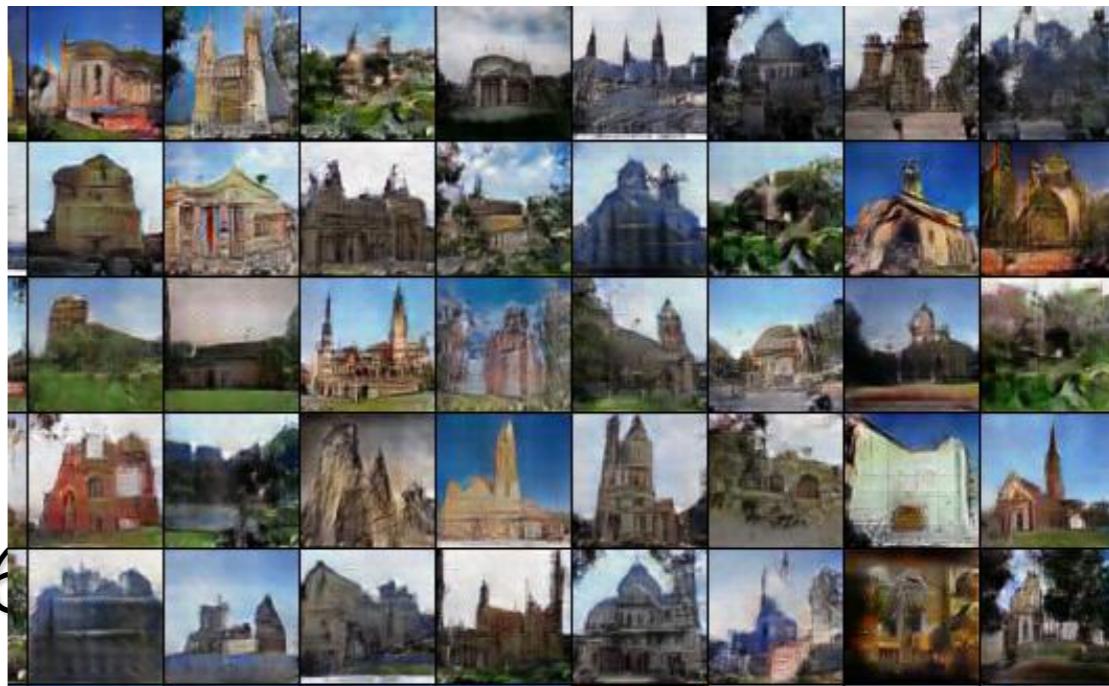
- Improved multi-scale architecture and Batch-Normalization:

[Radford et al.'16]



Generative Adversarial Networks

- GRAN [Generative Recurrent Adversarial Nets, Im et al.'16]
- Video Prediction [Mathieu et al.'16]
- CNN Reconstruction [Brox et al.'16]
- A very *hot* topic within the Deep Learning community



Generative Adversarial Networks

- Some open research directions:

1. **Optimization:**

1. How to ensure a correct algorithm?
2. Existence of a Lyapunov function?

2. **Statistics:**

1. How to determine the discriminator power (eg VC-dimension) to obtain consistent estimators?
2. Control of overfitting to the training distribution?

3. **Applications:**

- Language Modeling
- Reinforcement Learning
- Algorithmic Tasks
- Importance Sampling