Stat 212b: Topics in Deep Learning
Lecture 17

Joan Bruna
UC Berkeley

Reminders & Announcements

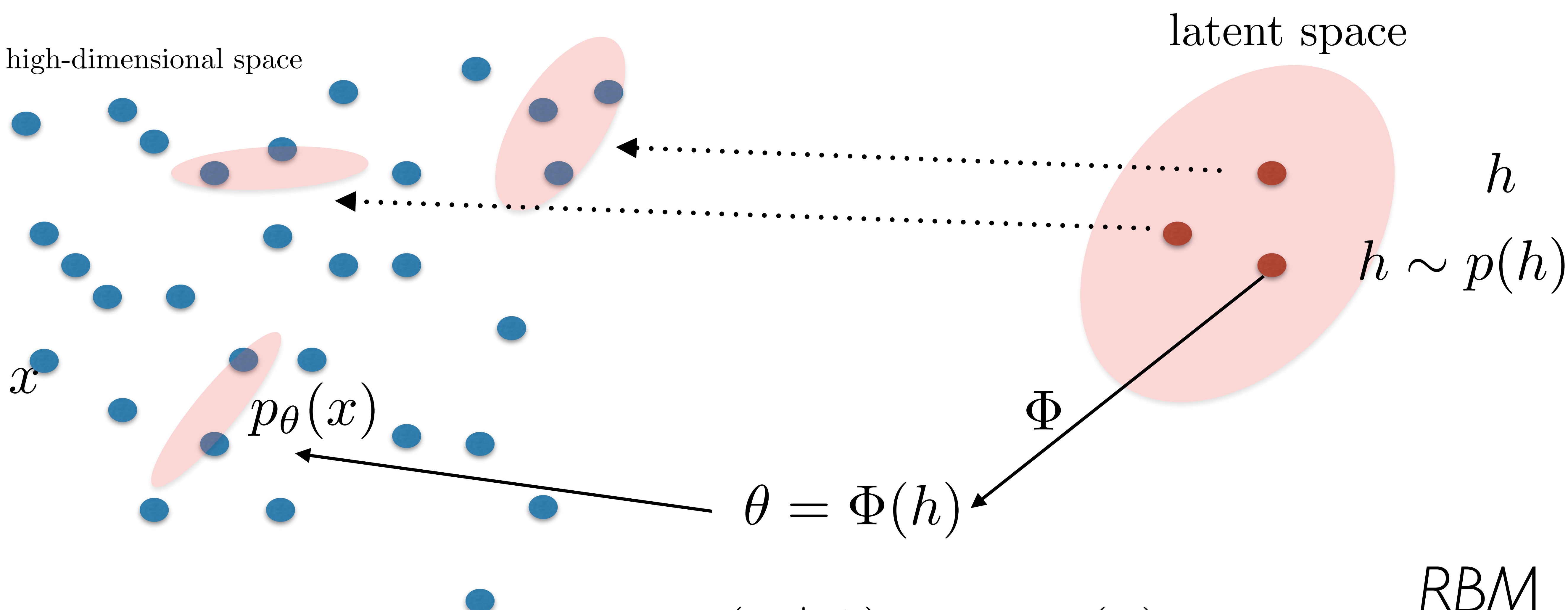
- Deadline for Reviews: **Friday Apr 1st**
  - Submit via BCourses assignment or email with [stat212b] in subject line.
- Deadline for Final Project Proposal: **Friday Apr 8th**
  - A short description of what you plan to do.
  - Can be either a *software implementation*, an *oral presentation* and/or a *tiny research project*.
- Ian Goodfellow's guest lecture is cancelled
  - replacement TBD.

Objectives

- (long) Review of previous lecture.
- Generative Adversarial Networks
  - applications
- Maximum Entropy Distributions
  - examples
  - MCMC
- Self-Supervised learning
  - word2vec
  - slow feature analysis

Review: Latent Graphical Models

- Latent Graphical Models or *Mixtures.*



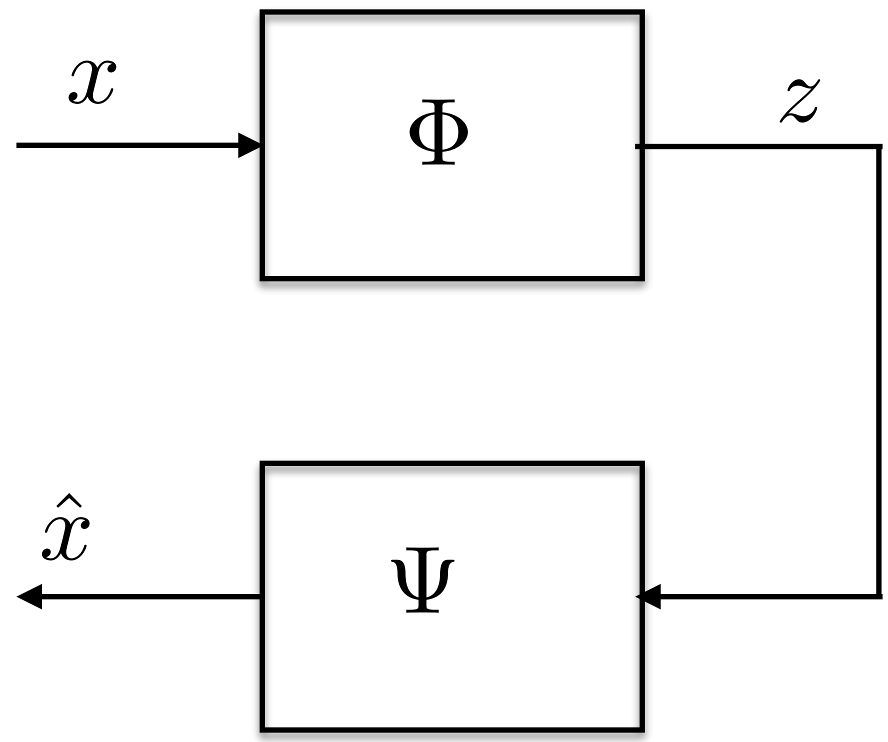
$$p(x \mid h) = p_{\Phi(h)}(x)$$

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

RBM
DBN
DBM
VAE
...

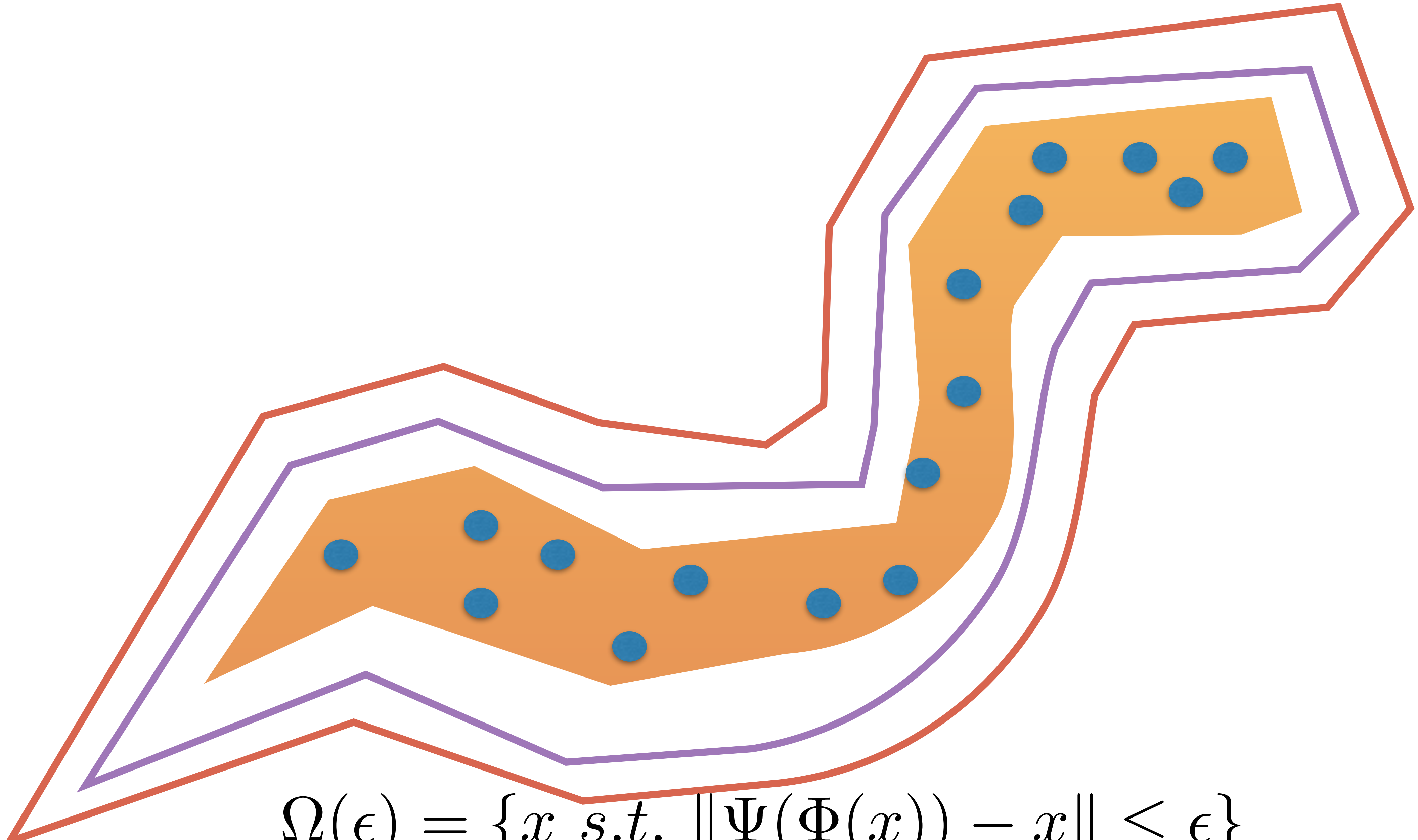
Review: Auto encoders

- *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



$$\Omega(\epsilon) = \{x \;\; s.t. \;\; \|\Psi(\Phi(x)) - x\| \leq \epsilon\}$$

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

Review: EM and Variational Bound

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

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

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

(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

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

$H(q)$: 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.

Approximate Posterior Inference

- For most models, the posterior is analytically intractable:

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

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:

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

Mean Field Variational Bayes

- Joint likelihood of observed and latent variables:

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

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

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

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

Mean Field Variational Bayes

- Joint likelihood of observed and latent variables:

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

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

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

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

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

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:

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

- 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

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

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

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

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:

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

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

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 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

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

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

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.

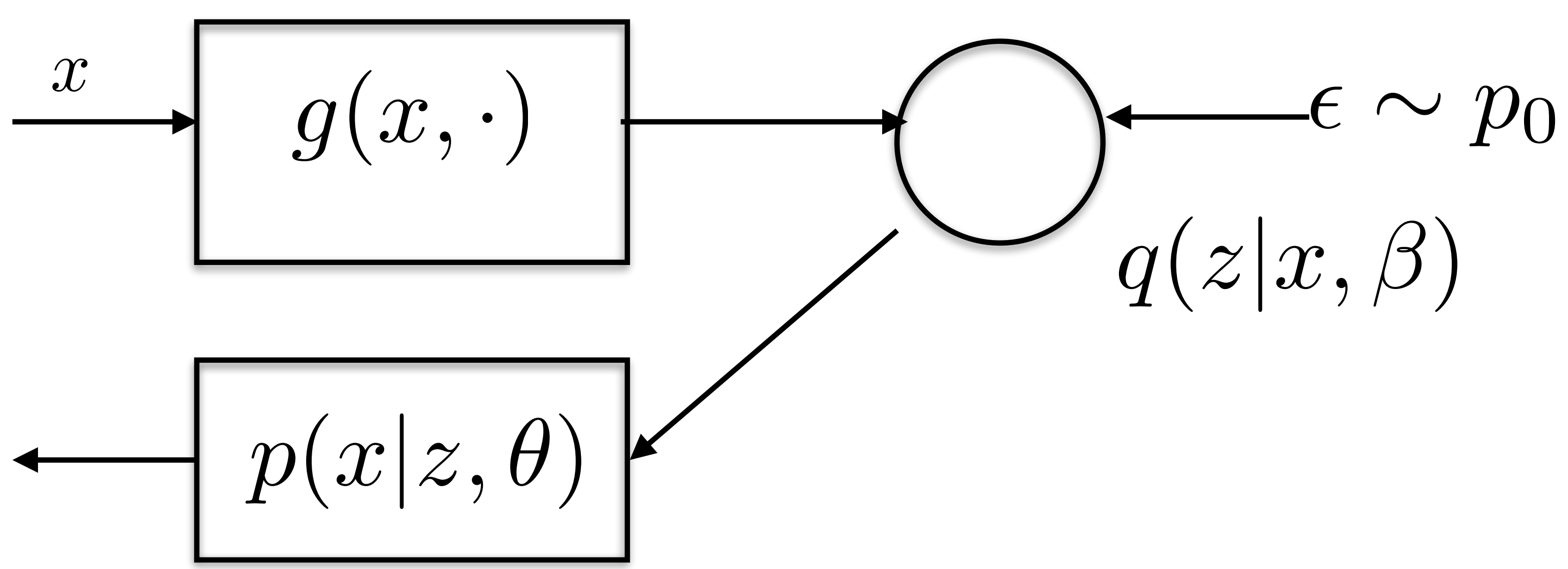
Variational Autoencoders

- 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:

$$p(z) = \mathcal{N}(z; 0, \mathbf{I})$$

- Let the conditional likelihood be also Gaussian:

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

- Variational approximate posterior also Gaussian:

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

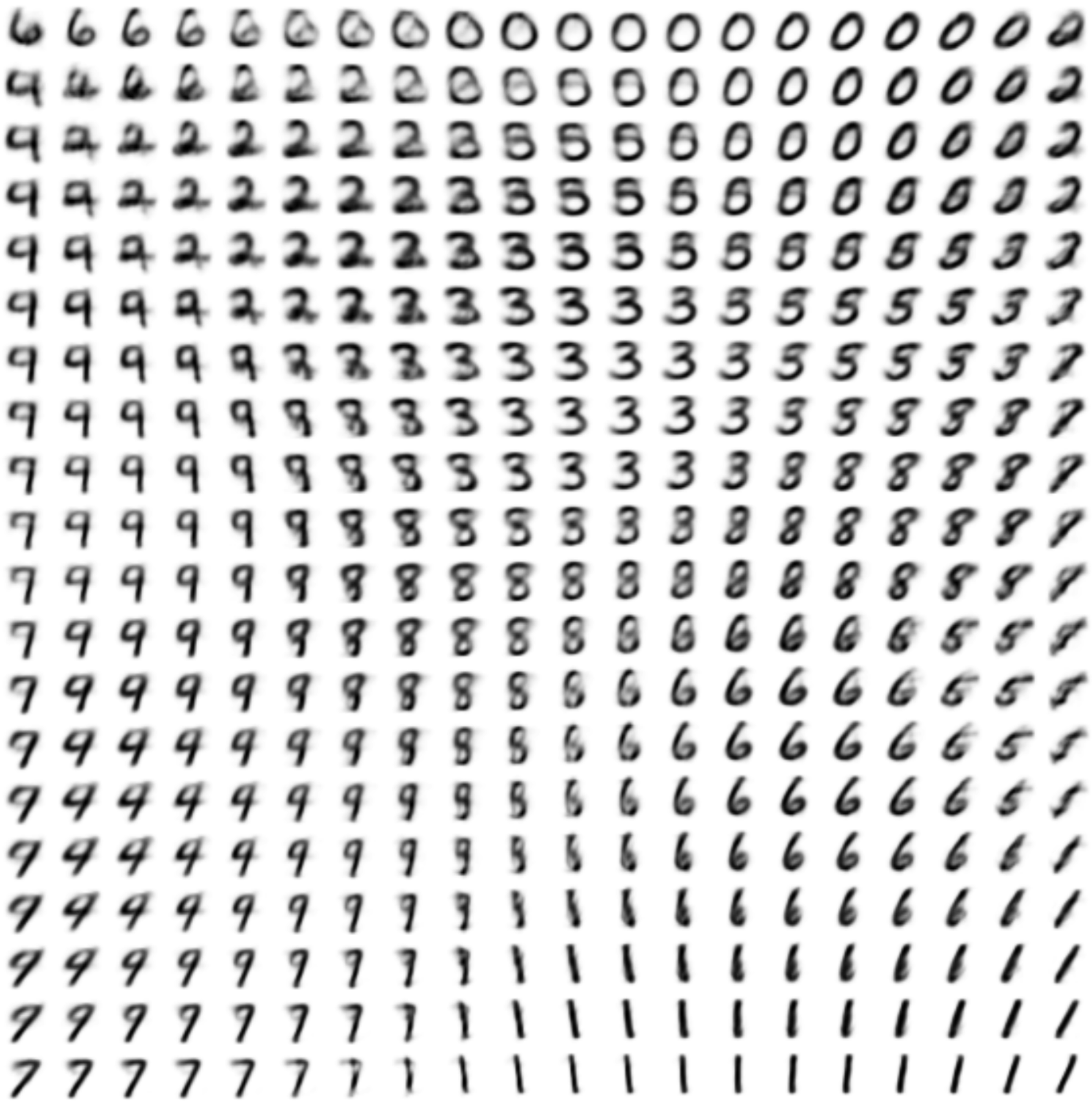
$$\overline{\mu}(z), \overline{\Sigma}(z) \;\; : \;\; \text{Neural networks, } (\overline{\Sigma} \text{ diagonal})$$

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

Variational Autoencoder

- Examples using a two-dimensional latent space:

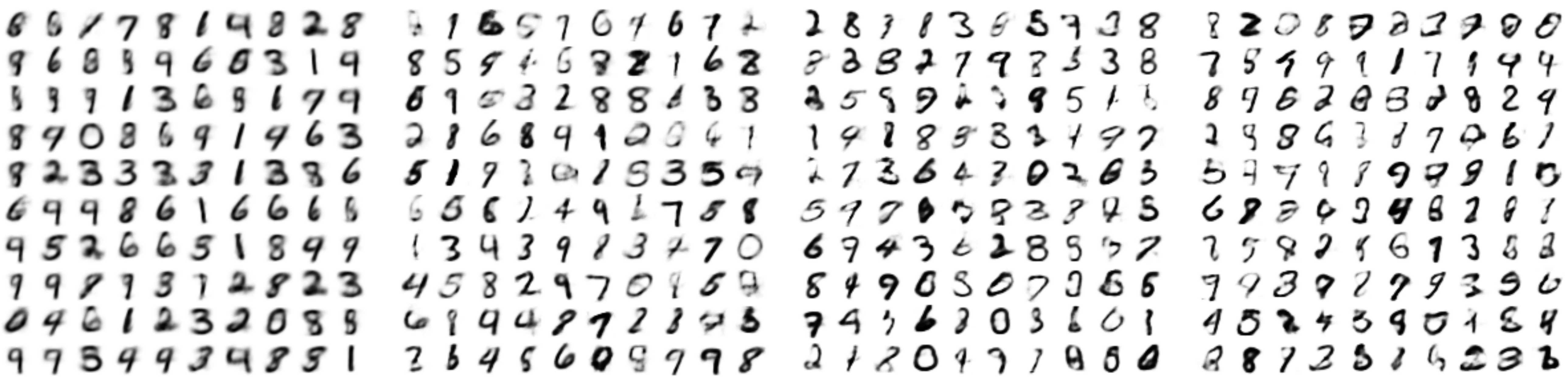
(a) Learned Frey Face manifold



(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:

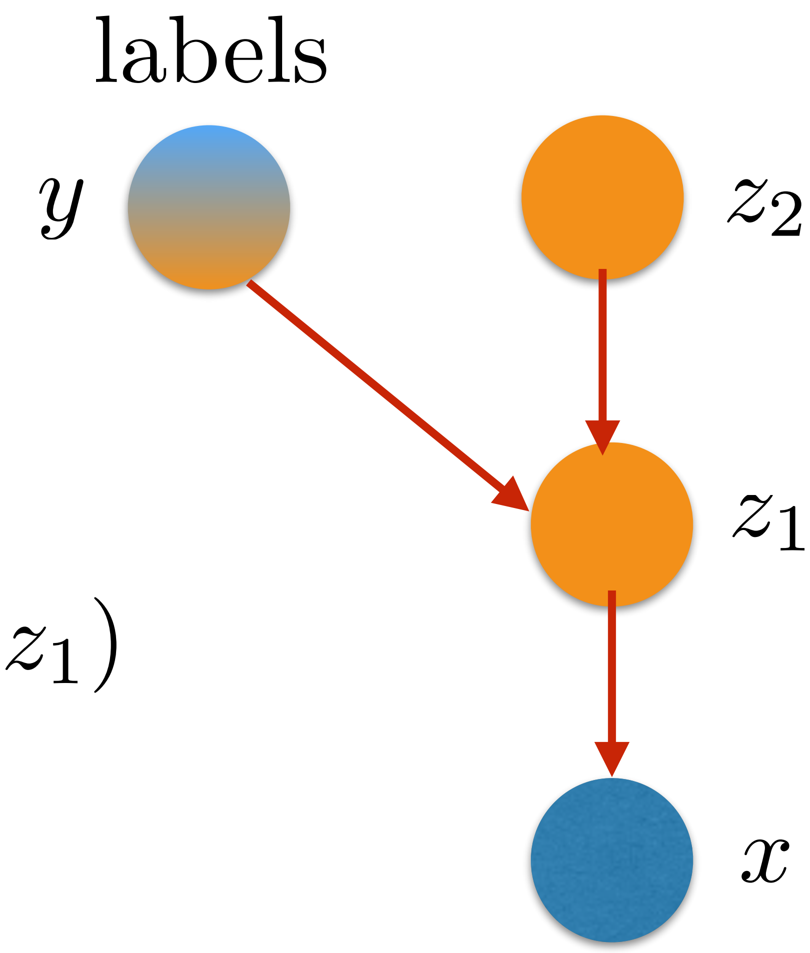
We observe $\{x_i\}_{i \leq L_1}$ and $\{x_j, y_j\}_{j \leq L_2}$, with $x_i \sim p(x)$, $x_j \sim p(x)$.

$$L_1 \gg L_2$$

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:

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

- 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.*
- Classification results on MNIST:

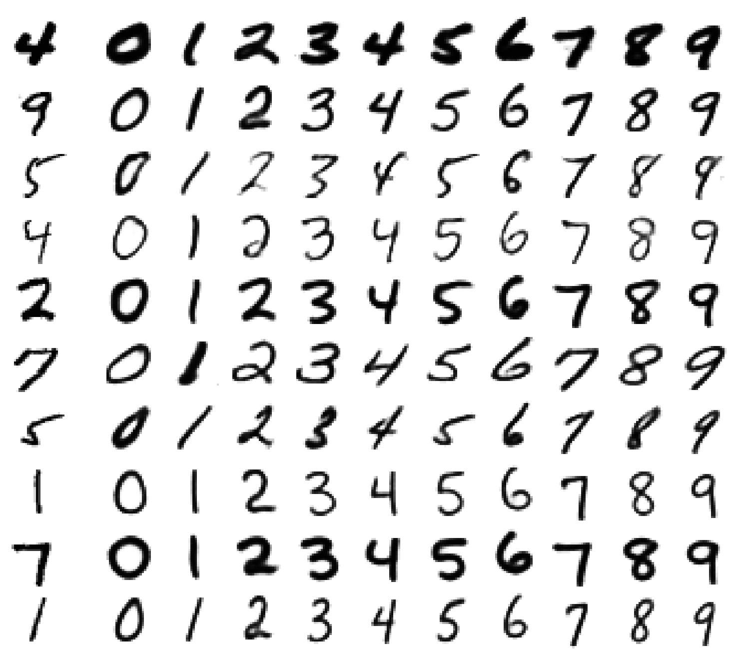
Table 1: Benchmark results of semi-supervised classification on MNIST with few labels.

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 ($\pm$ 0.95)	11.82 ($\pm$ 0.25)	11.97 ($\pm$ 1.71)	**3.33** ($\pm$ 0.14)
600	11.44	7.68	6.16	6.3	5.13	–	5.72 ($\pm$ 0.049)	4.94 ($\pm$ 0.13)	**2.59** ($\pm$ 0.05)
1000	10.7	6.45	5.38	4.77	3.64	3.68 ($\pm$ 0.12)	4.24 ($\pm$ 0.07)	3.60 ($\pm$ 0.56)	**2.40** ($\pm$ 0.02)
3000	6.04	3.35	3.45	3.22	2.57	–	3.49 ($\pm$ 0.04)	3.92 ($\pm$ 0.63)	**2.18** ($\pm$ 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 **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 Autoencoders" Burda et al'16

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

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

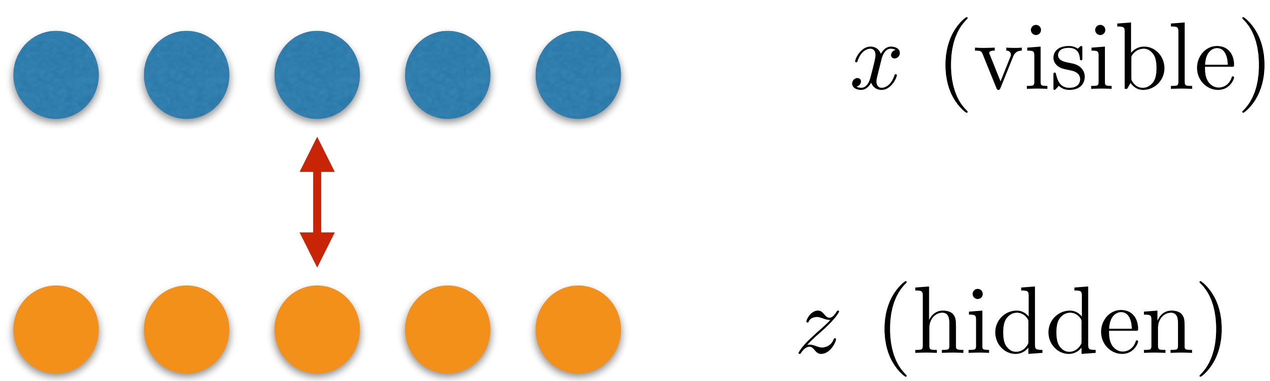
$$\lim_{k \to \infty} \mathcal{L}_k(x) = \log p(x) \text{ if } \frac{p(x, z)}{q(z|x)} \text{ is bounded} .$$

Other extensions

- Make posterior inference more flexible at small computational cost using an inverse autoregressive Gaussian model [Kingma, Salimans, Welling, '16].

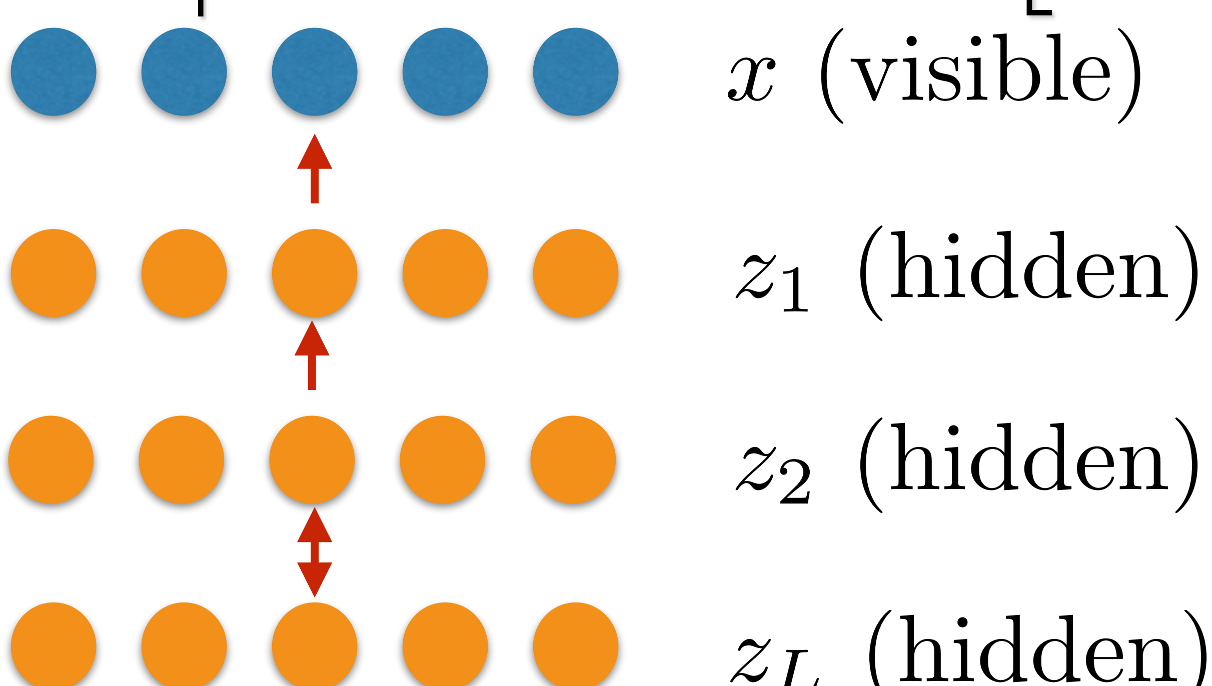
Other directed models

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



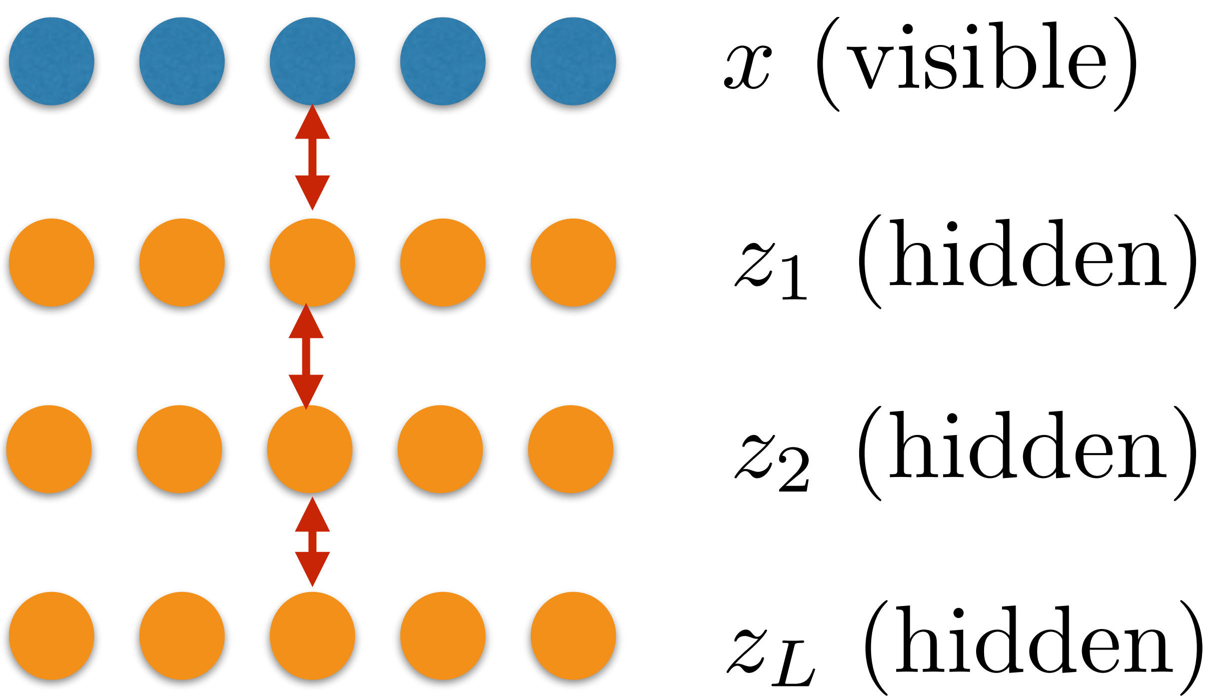
$$p(x,z) = \exp\left(\langle \theta_1, xz^T \rangle + \langle \theta_2, x \rangle + \langle \theta_3, z \rangle - \log A(\theta)\right)$$

- Deep Belief Networks [Hinton et al'02]



Other directed models

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



Can be trained greedily layer-wise

- 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*:

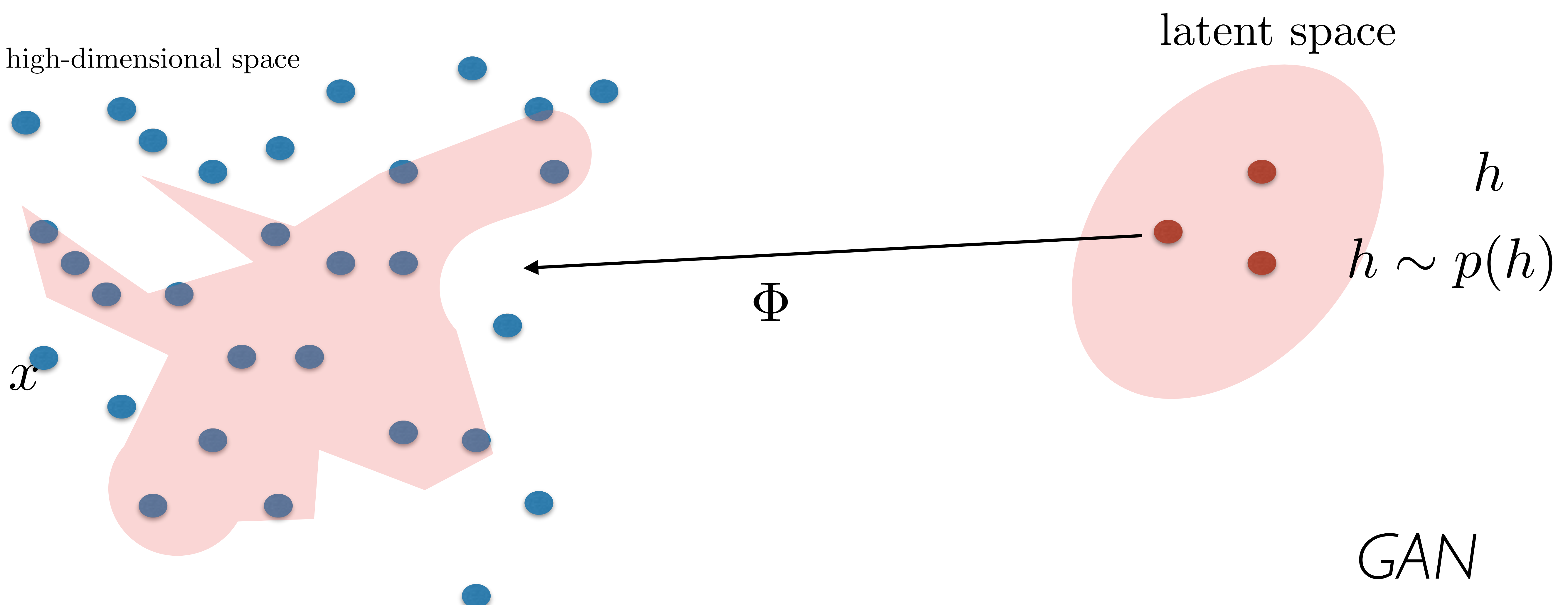
    $$x = \Phi(z) + \epsilon \ , \ \ \epsilon \sim \mathcal{N}(0, \Sigma(z))$$

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

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

Generative Models of Complex data

- Flows or Transports of Measure



GAN
NormFlow
$\ldots$

$p(x)$ defined implicitly with

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

Measure Transports

- How to train the transport Φ?

- We saw 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]
[Tabak et al.'10]

- Consider a diffeomorphism $\Phi : \mathbb{R}^N \to \mathbb{R}^N$.
- 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*,

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

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

Normalizing Flows

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

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

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

- 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].

Diffusion and Non-equilibrium Thermodynamics

[Sohl-Dickstein et al.'15]

- We can also consider *infinitesimal* flows:

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

$\mathcal{F}$ 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 $\pi(x)$:

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

$$q(x^{(t+1)}|x^{(t)}) = T_\pi(x^{(t+1)}|x^{(t)}, \beta_t)$$ β_t: 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]

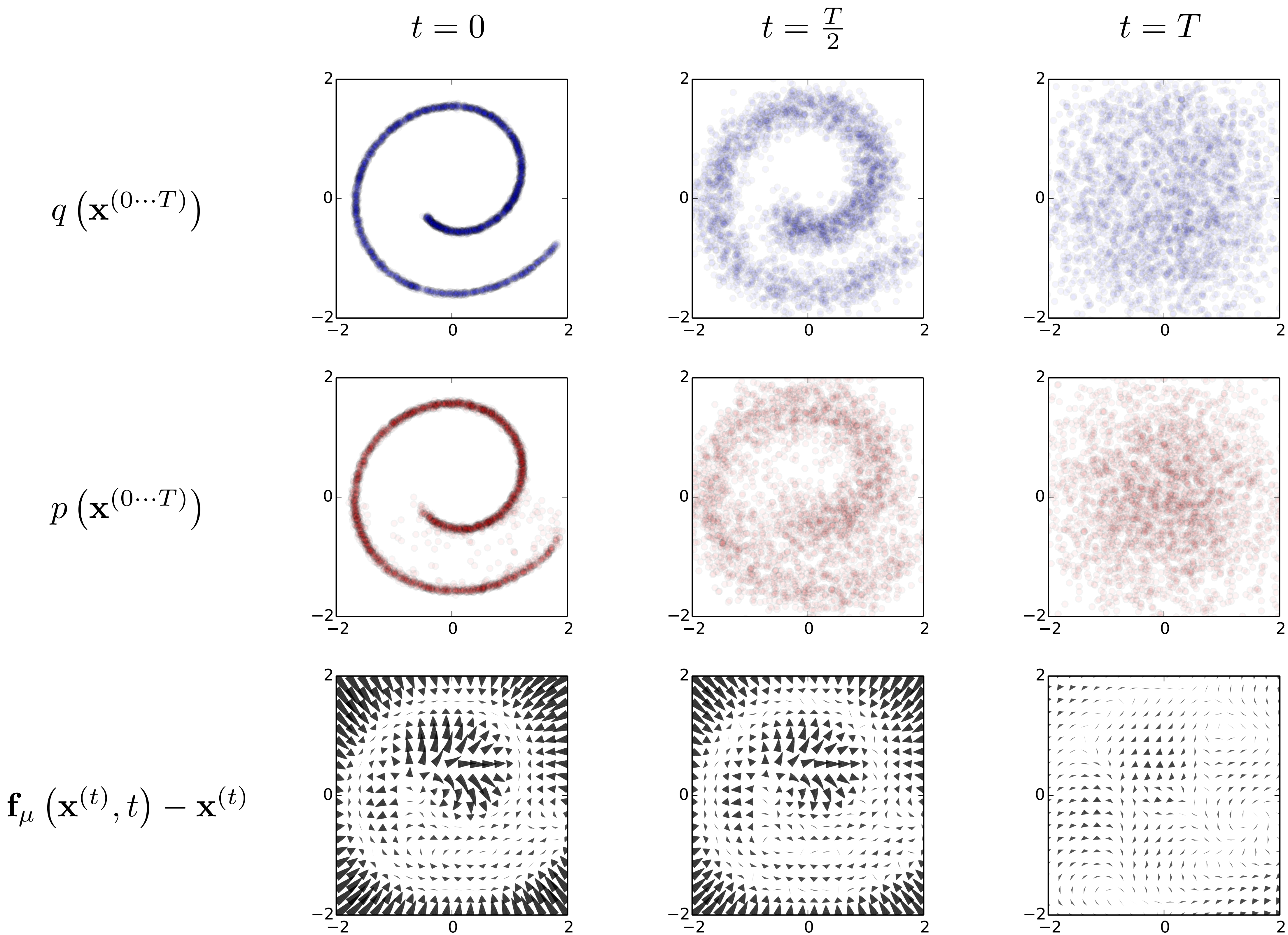
- The "forward" trajectory diffuses the data distribution into a tractable distribution, eg Gaussian.
- The generative model learns how to reverse the diffusion:

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

  – 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 a lower bound that can be evaluated efficiently using annealed importance sampling.

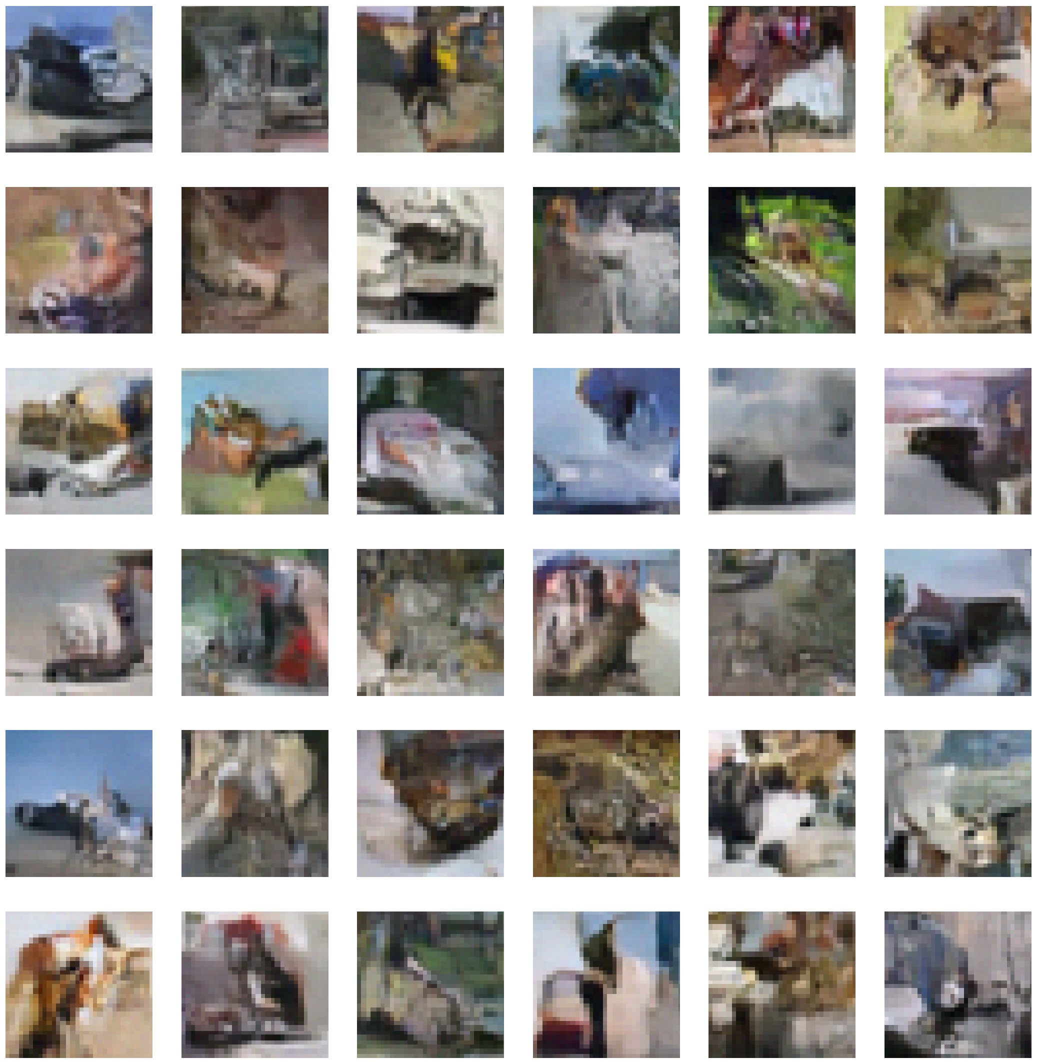
Diffusion and Non-equilibrium Thermodynamics

[Sohl-Dickstein et al.'15]

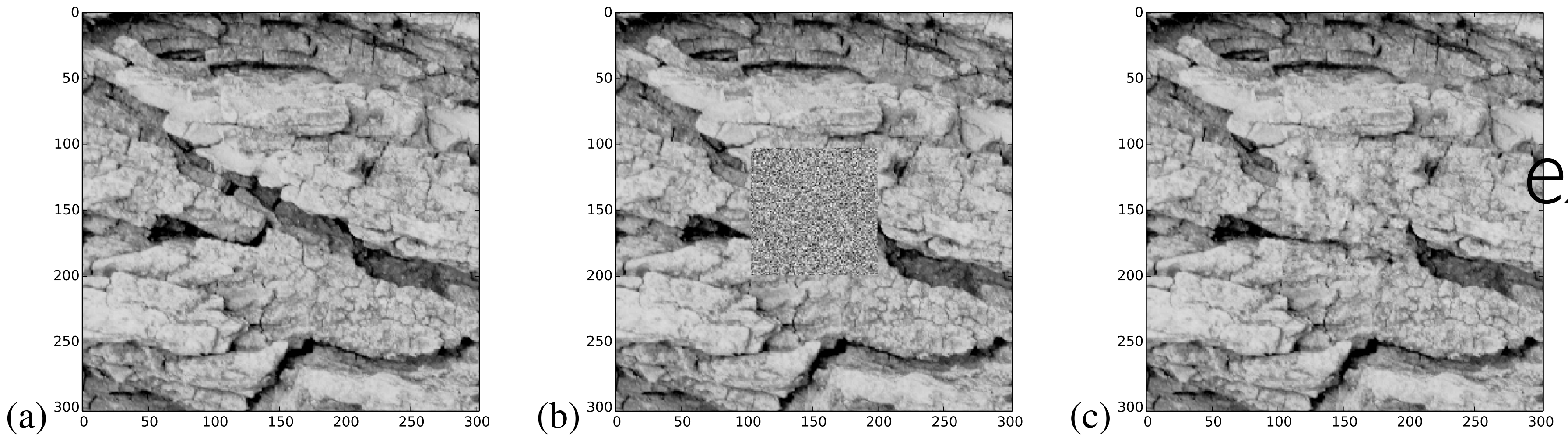


Diffusion and Non-equilibrium Thermodynamics

[Sohl-Dickstein et al.'15]



samples from the model trained on CIFAR-10



inpainting experiments

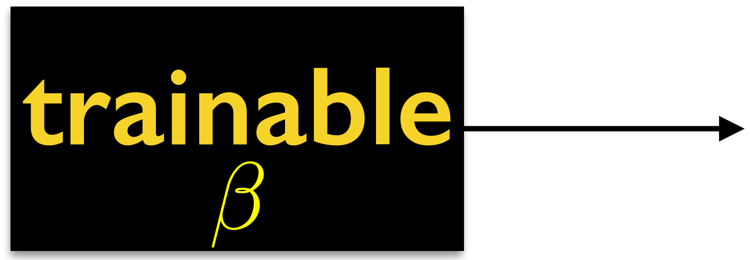
Generative Adversarial Networks

[Goodfellow et al., '14]

Generative Adversarial Networks

[Goodfellow et al., '14]

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



$$X \sim p_\beta(x)$$

Generative Adversarial Networks

[Goodfellow et al., '14]

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

trainable β → $X \sim p_\beta(x)$

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

training data → $\{X_i\}_i\,;\ X_i \sim p(x)$ → Discriminative classifier θ

trainable β → $\{X_j\}_j\,;\ X_j \sim p_\beta(x)$ → Discriminative classifier θ

Generative Adversarial Networks

[Goodfellow et al., '14]

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

$$\boxed{\textbf{trainable}\ \beta} \longrightarrow X \sim p_\beta(x)$$

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

$$\boxed{\text{training data}} \longrightarrow \{X_i\}_i\,;\ X_i \sim p(x) \longrightarrow \boxed{\text{Discriminative classifier } \theta}$$

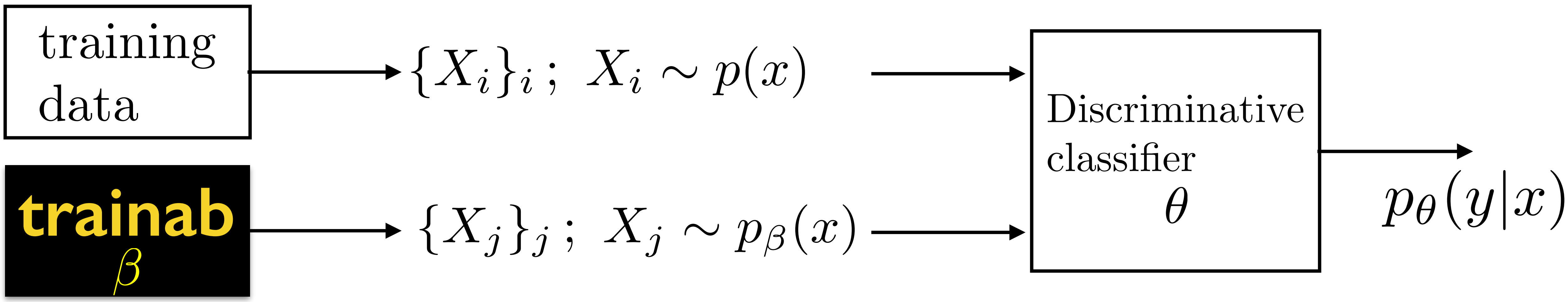
$$\boxed{\textbf{trainable}\ \beta} \longrightarrow \{X_j\}_j\,;\ X_j \sim p_\beta(x) \longrightarrow \boxed{\text{Discriminative classifier } \theta}$$

- 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,

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

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

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

$$\min -H(\bar{\alpha}) \text{ is attained when } \alpha = 1/2, \text{ thus}$$

$$p_\beta(x) = p_{data}(x)$$

- In practice, however, we parametrize both generator and discriminator using neural networks.

- Optimize the cost using gradient descent.

Generative Adversarial Training

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

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

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

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

Generative Adversarial Training

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

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

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

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

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

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

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:

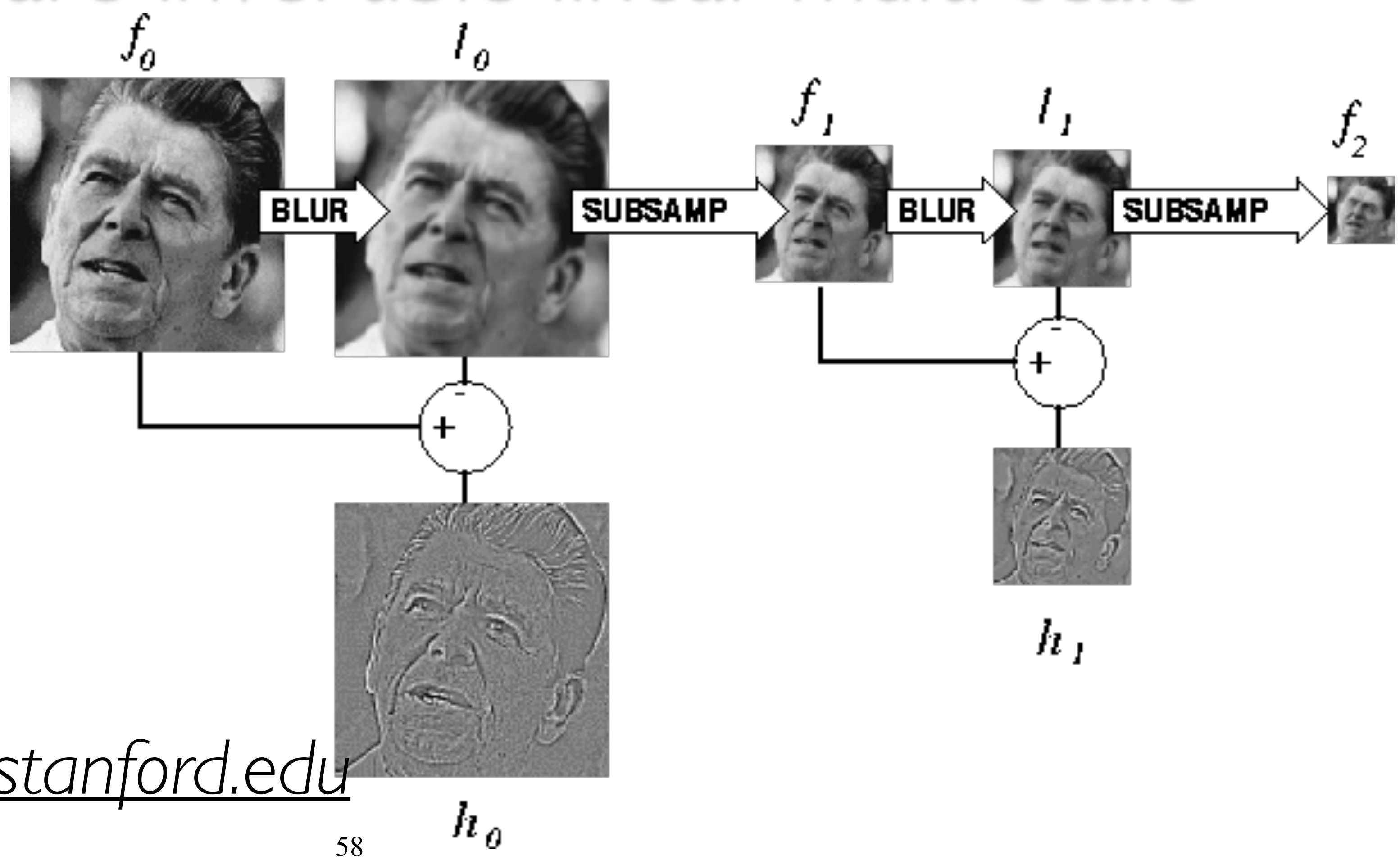
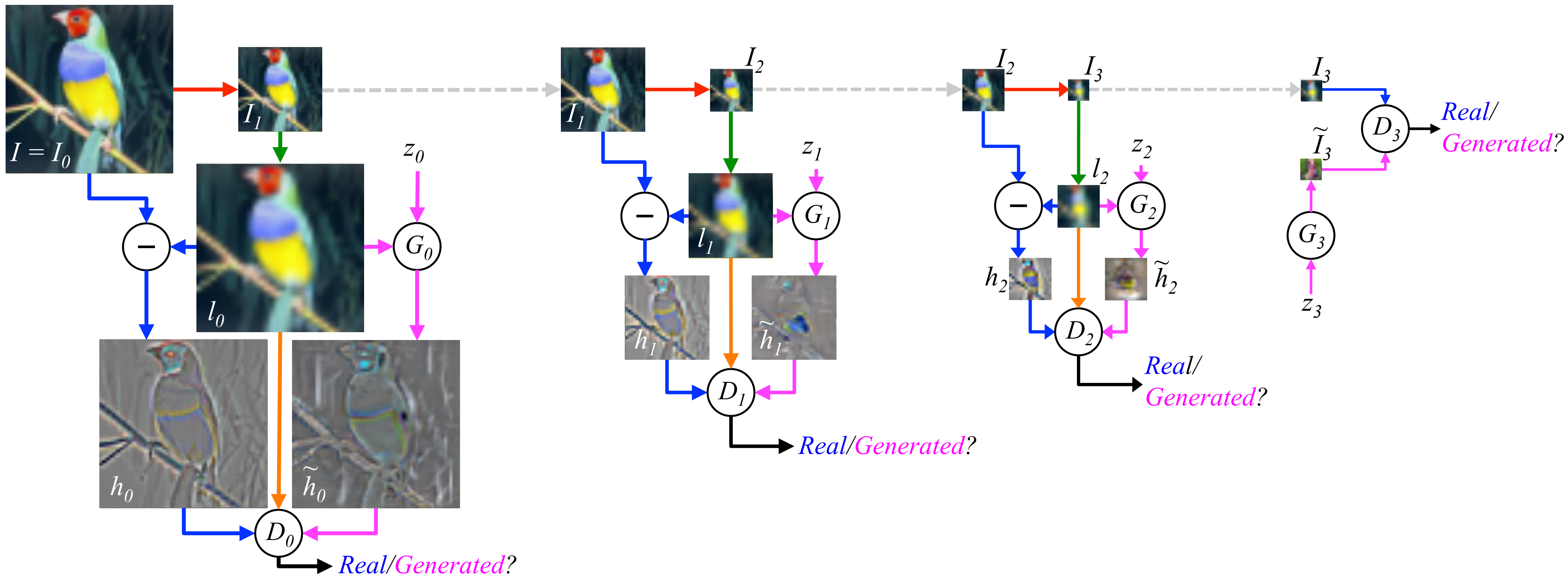


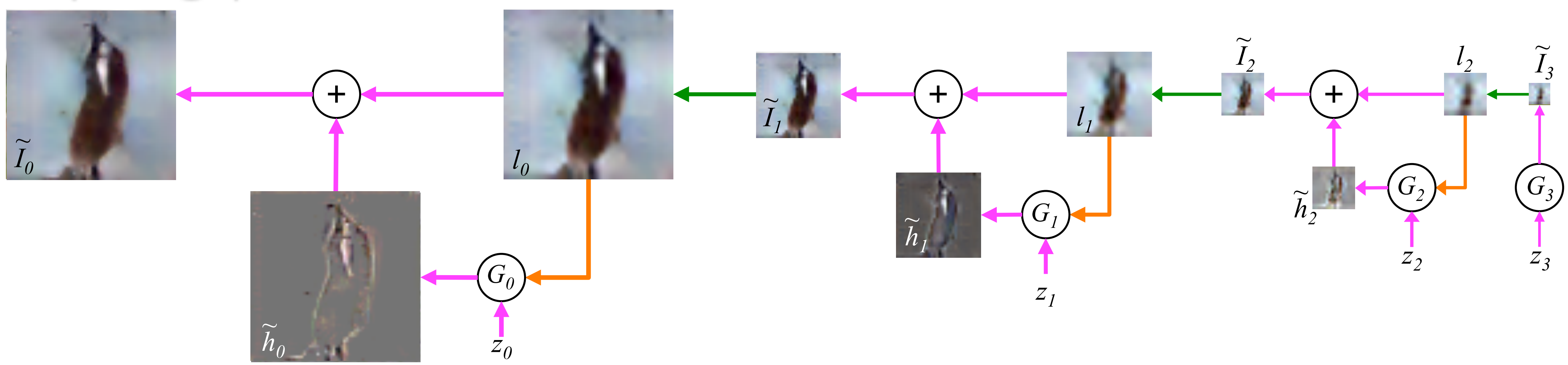
figure source: http://sepwww.stanford.edu

LAPGAN

- Training procedure:

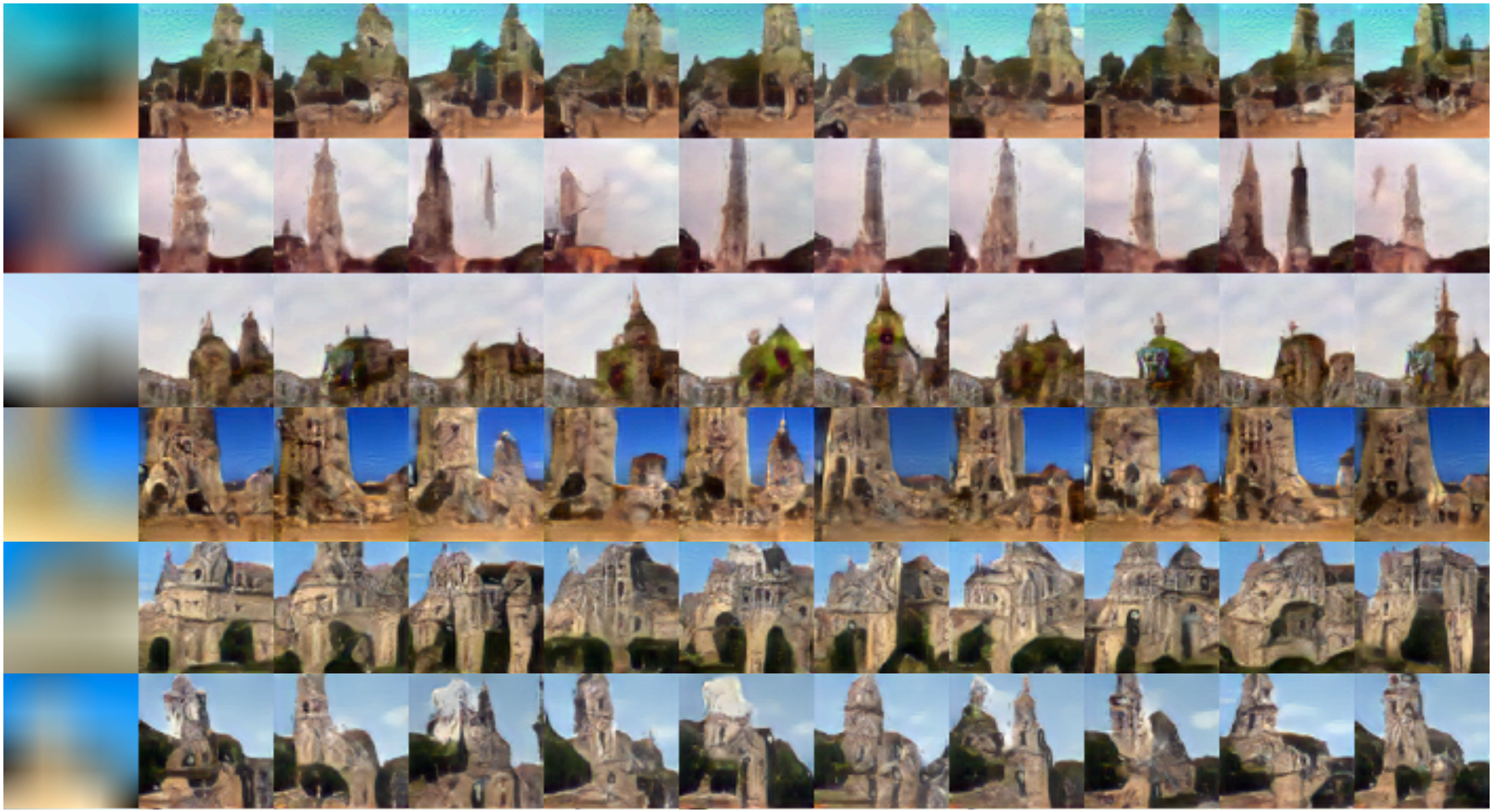


- Sampling procedure:



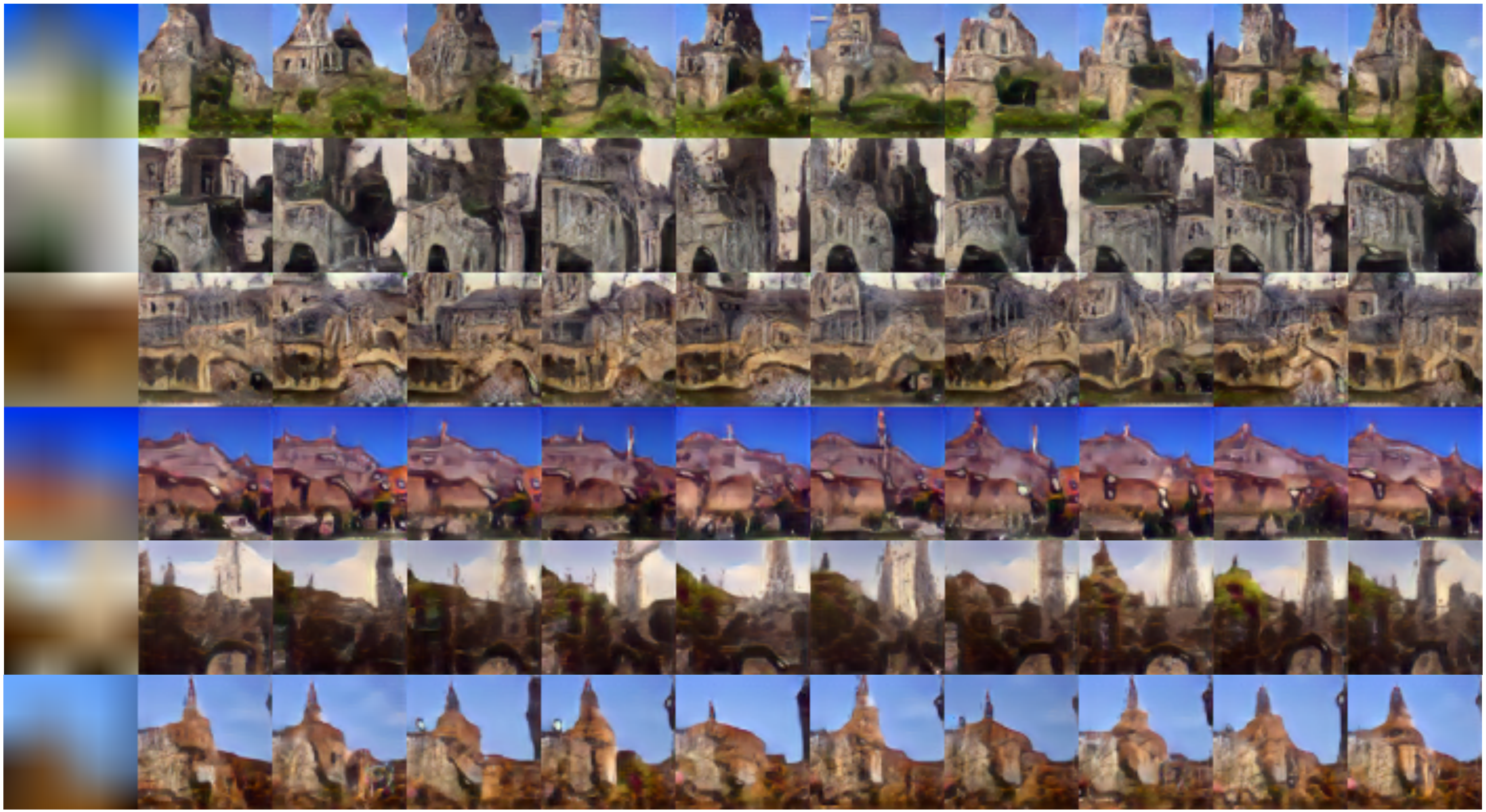
LAPGAN

- Samples generated from the model:



LAPGAN

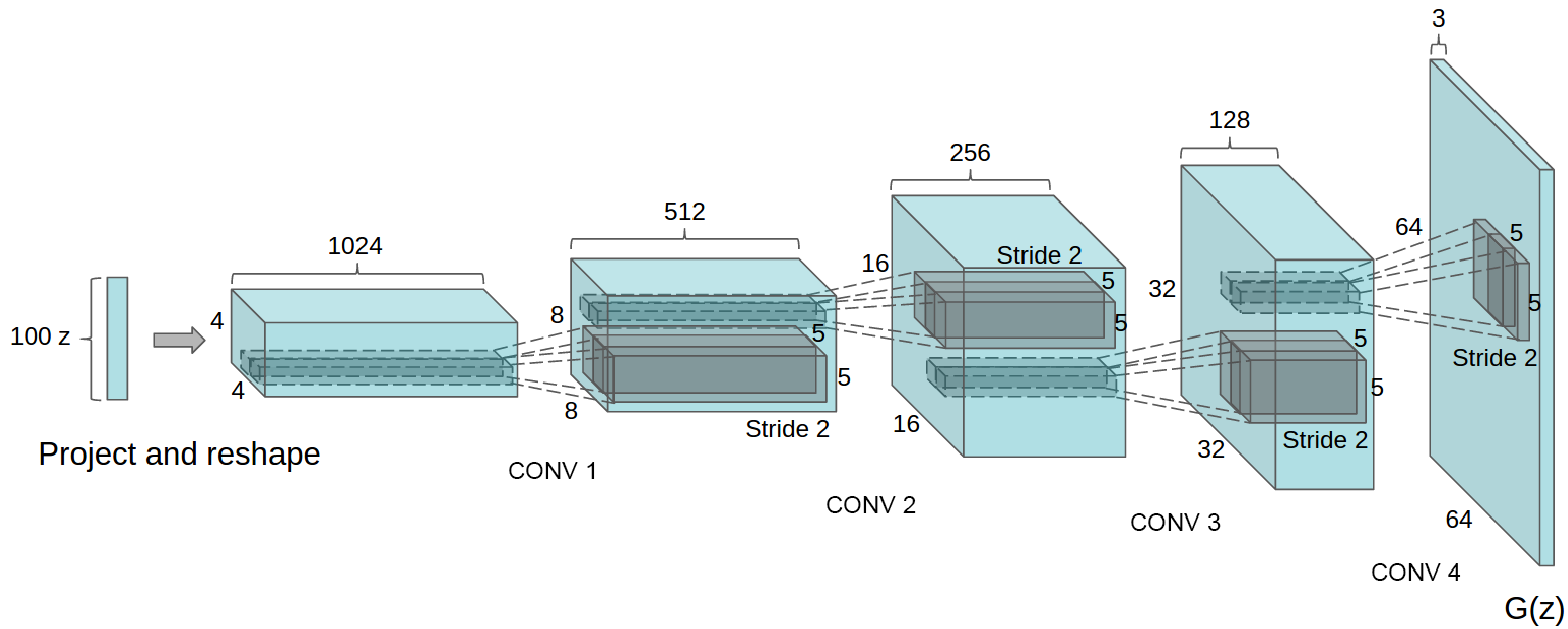
- Samples generated from the model:



DC-GAN

[Radford et al.'16]

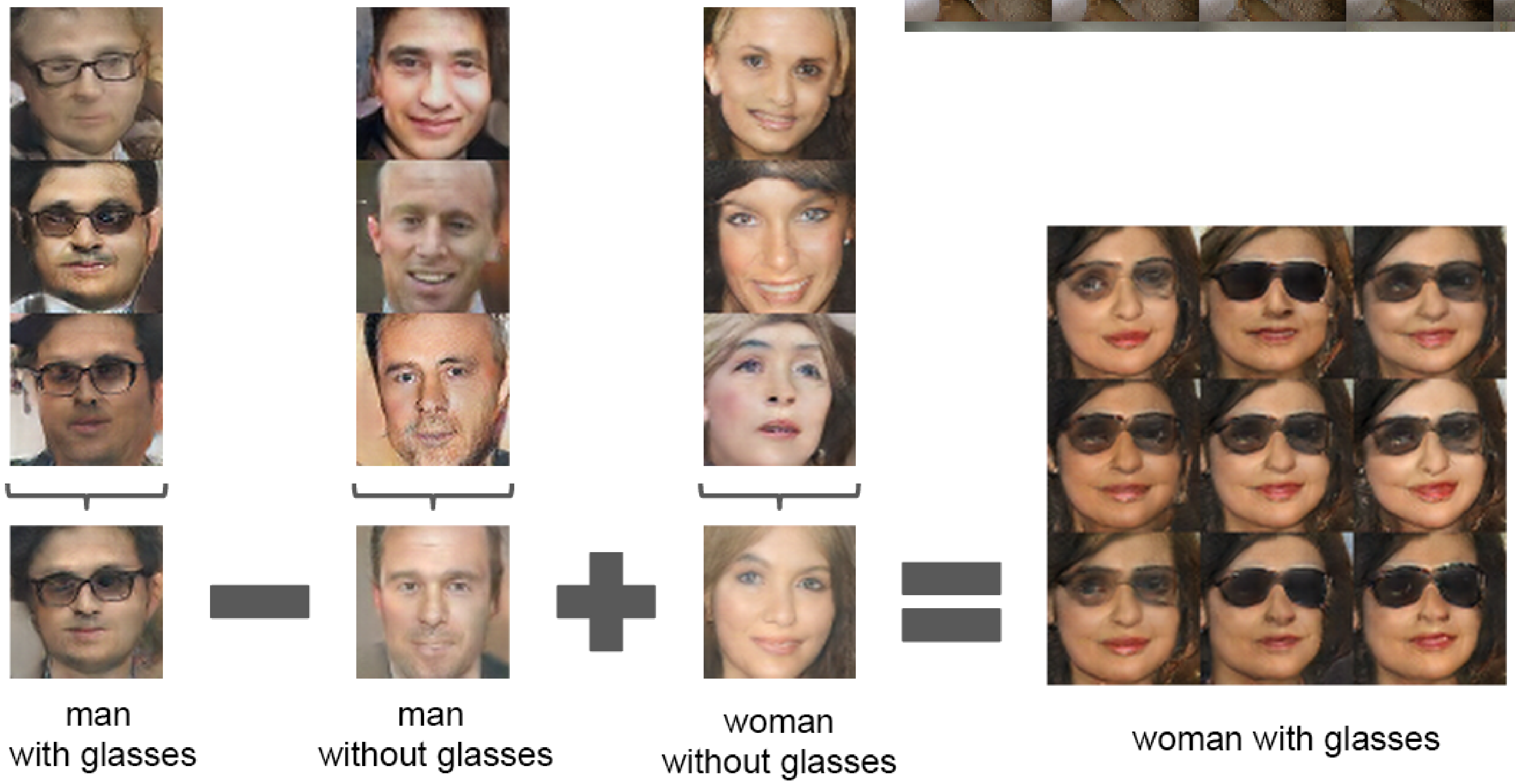
- Improved multi-scale architecture and Batch-Normalization:



DC-GAN

[Radford et al.'16]

- Improved multi-scale architecture and Batch-Normalization:



Generative Adversarial Networks

- GRAN [Generative Recurrent Adversarial Nets, Im et al.'16]

- Video Prediction [Mathieu et al.'1

- 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

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. $p(z|x)$
- How to regularize the density estimation?

Gibbs Models

- Motivation: Given a collection of discriminative measurements $\Phi(x) = \{\Phi_j(x)\}_j$, how can we build a generative model?

Gibbs Models

- Motivation: Given a collection of discriminative measurements $\Phi(x) = \{\Phi_j(x)\}_j$, how can we build a generative model?

- Supervised Learning Setup:

Empirical training distribution $\hat{p} : \{(x_i, y_i)\}$ $y_i \in \{1, K\}$

Empirical class-conditional moments:

$$\mu_k = \mathbb{E}_{(x,y)\sim\hat{p}}(\Phi(x)|y = k) \quad k = 1 \ldots K$$

Gibbs Models

- Motivation: Given a collection of discriminative measurements $\Phi(x) = \{\Phi_j(x)\}_j$, how can we build a generative model?

- Supervised Learning Setup:

Empirical training distribution $\hat{p} : \{(x_i, y_i)\} \quad y_i \in \{1, K\}$

Empirical class-conditional moments:

$$\mu_k = \mathbb{E}_{(x,y)\sim\hat{p}}(\Phi(x)|y = k) \quad k = 1 \ldots K$$

- Necessary condition:

Class-conditional models $p_k(x)$ satisfy

$$\forall\ k\ ,\ \mathbb{E}_{x\sim p_k}\Phi(x) = \mu_k$$

Q: Does this completely specify p_k?

Gibbs Models

- Motivation: Given a collection of discriminative measurements $\Phi(x) = \{\Phi_j(x)\}_j$, how can we build a generative model?

- Supervised Learning Setup:

Empirical training distribution $\hat{p} : \{(x_i, y_i)\} \quad y_i \in \{1, K\}$

Empirical class-conditional moments:

$$\mu_k = \mathbb{E}_{(x,y)\sim\hat{p}}(\Phi(x)|y = k) \quad k = 1 \ldots K$$

- Necessary condition:

Class-conditional models $p_k(x)$ satisfy

$$\forall\ k\ ,\ \mathbb{E}_{x\sim p_k}\Phi(x) = \mu_k$$

Q: Does this completely specify p_k?

Clearly not

Gibbs Models

- Thus, we need a regularization principle.
- A "good" norm for probability distributions is the entropy

$$H(p) = -\mathbb{E}[\log p] = -\int p(x) \log p(x) dx$$

- It captures a form of smoothness for probability distributions
  - On compact domains, the maximum entropy distribution is the uniform measure (maximally smooth)
  - On non-compact domains, the max-entropy distribution might not exist.
- In our problem, we can use it to select, under the constraints $\forall\ k\ ,\ \mathbb{E}_{x \sim p_k} \Phi(x) = \mu_k$, those with maximum uncertainty (maximum smoothness).

Gibbs Models and Maximum Entropy

- We are thus interested in the problem

$$\max_p \; H(p)$$
$$s.t. \; \mathbb{E}_{x\sim p}\Phi(x) = \mu \; . \in \mathbb{R}^d$$

- Constrained optimization that we approach using calculus of variations
- Lagrangian of the problem is

$$L(p, \lambda_1, \dots, \lambda_d) = H(p) + \sum_j \lambda_j (\mathbb{E}_{x\sim p}\Phi_j(x) - \mu_j) \; .$$

$$= -\int p(x)\log(p(x))dx + \sum_j \lambda_j \left(\int \Phi_j(x)p(x)dx - \mu_j \right)$$

Gibbs Models and Maximum Entropy

- Thus we have

$$\frac{\partial L}{\partial p(x)} = -\log p(x) - 1 + \sum_j \lambda_j \Phi_j(x) = 0$$

$$\Rightarrow \quad \log p(x) = \lambda_0 + \sum_j \lambda_j \Phi_j(x)$$

$$\Rightarrow \quad p(x) = \frac{\exp\left(\sum_j \lambda_j \Phi_j(x)\right)}{Z}$$

where

λ_j are Lagrange multipliers guaranteeing that $\mathbb{E}_{x\sim p}\Phi_j(x) = \mu_j$.

Z is a Lagrange multiplier guaranteeing that $p(x) = 1$

Gibbs Model

- Thus, given features $\Phi(x)$, maximum entropy distributions are in the exponential family given by

$$p(x) = \exp\left(\langle \lambda, \Phi(x) \rangle - A(\lambda)\right)$$

- In a discriminative setting, the final model is a mixture in this exponential family:

$$k \sim \mathrm{cat}\{1, K\}$$

$$x \sim p_k(x) = \exp(\langle \lambda_k, \Phi(x) \rangle - A(\lambda_k)) \ , \quad \mathbb{E}_{x \sim p_k} \Phi(x) = \mu_k \ .$$

- This model has many names:
  – Gibbs, Boltzmann, "Energy-based" Model, MaxEnt, …

Gibbs Learning

- Q: How to train this model?
  - i.e. how to adjust the Lagrange multipliers?

- Q: How to train such a model without labels/ discriminative features?
  - What criteria?
  - Learn the sufficient statistics