

Inference and Representation

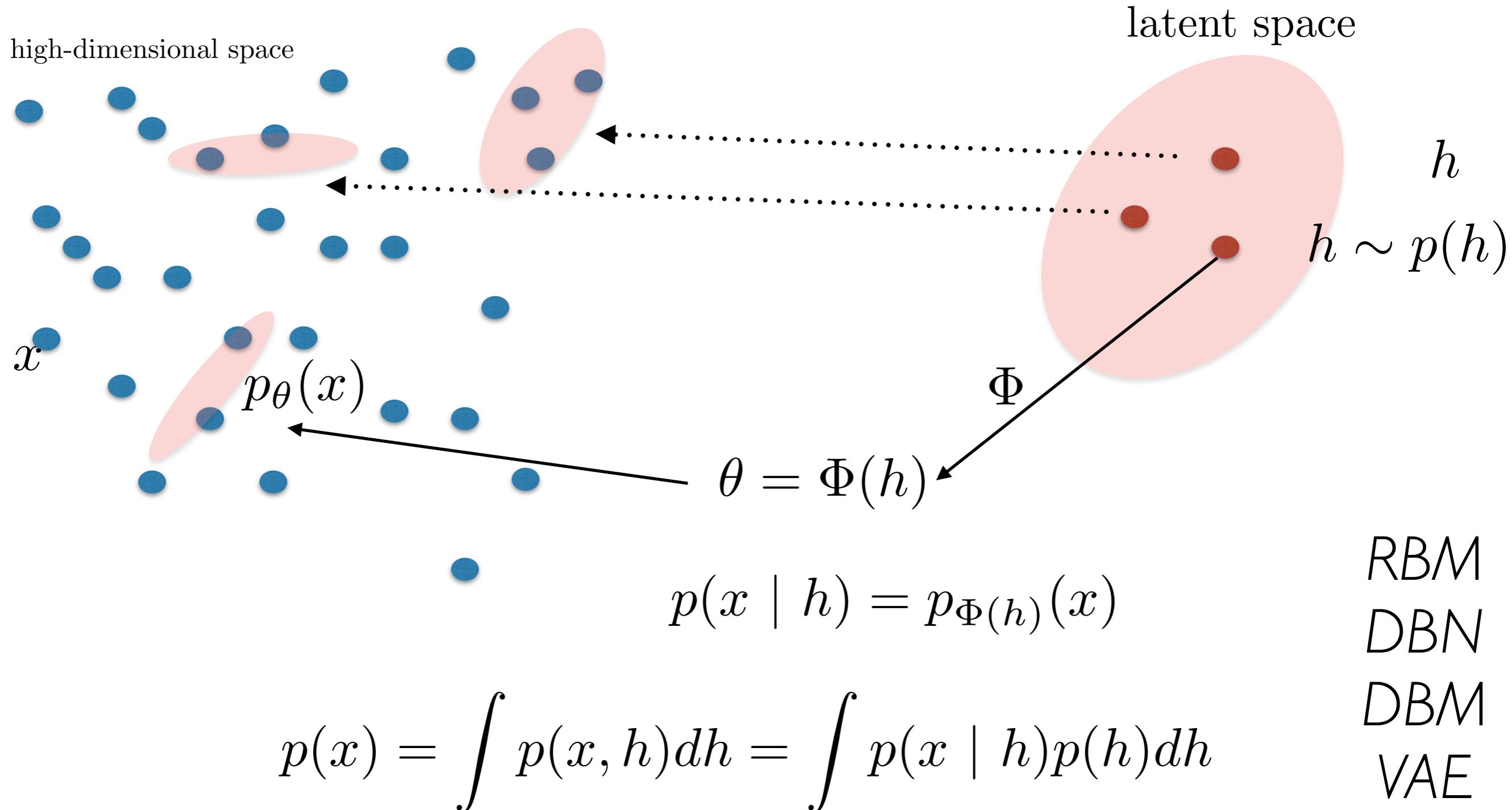
Lecture 11

Joan Bruna
Courant Institute, NYU



Latent Graphical Models

- Latent Graphical Models or *Mixtures*.



Model: additive combination of simple parametric models

Objectives lectures 11 and 12

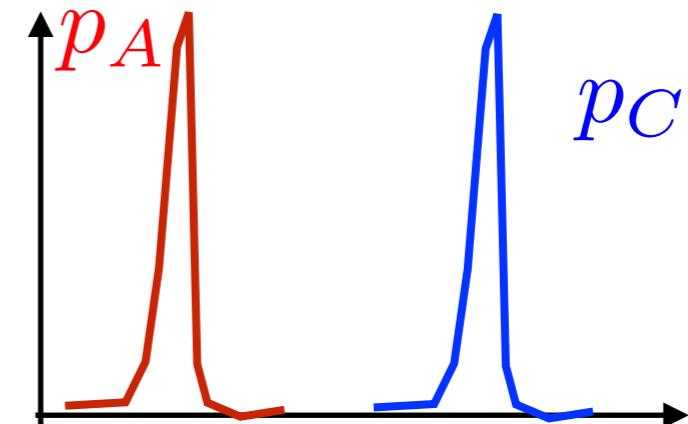
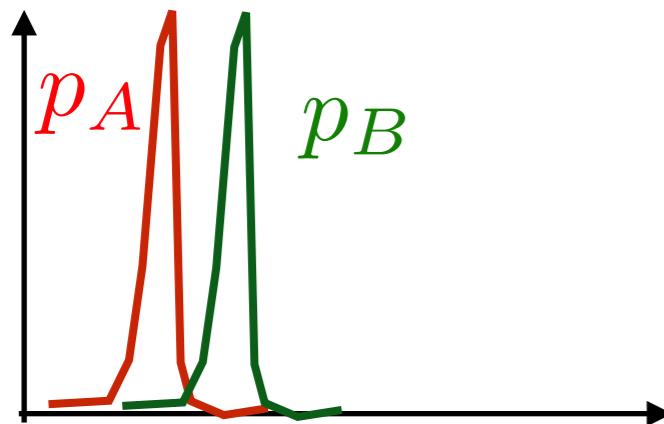
- Variational Flows
- Generative Adversarial Networks
- Autoregressive Models
- Open Research

Limits of Mixture Models

- Inference is 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 suffer from regression to the mean.

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.



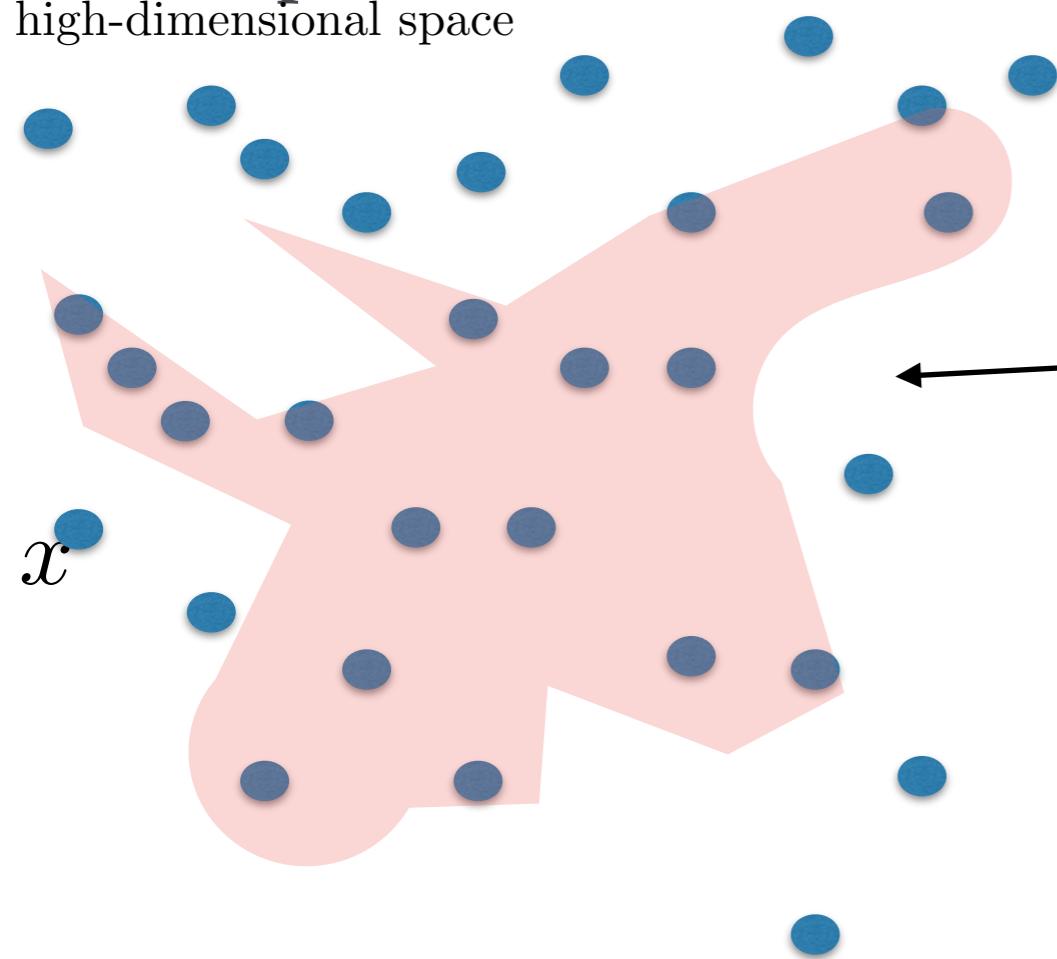
$$\text{MSE}(p_A, p_B) \approx \text{MSE}(p_A, p_C) = O(1)$$

- In particular, stable to deformations?

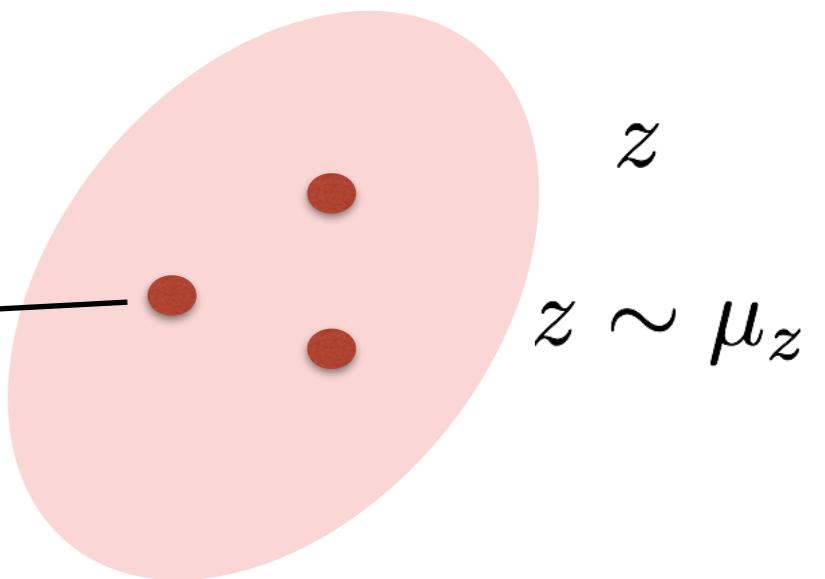
GENERATIVE MODELS OF COMPLEX DATA

► Implicit Models

high-dimensional space



latent space



GAN
NormFlow
...

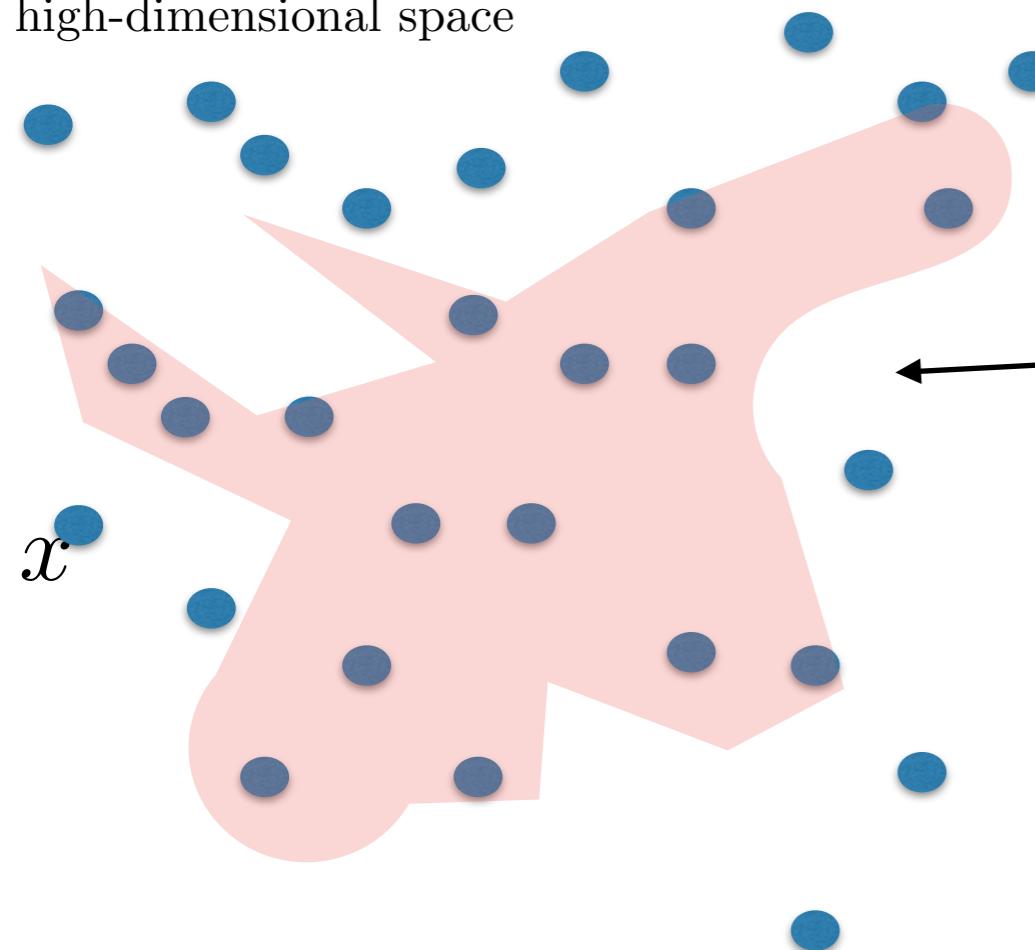
Pushforward measure: $P_\theta := G_\theta \# \mu_z$

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

GENERATIVE MODELS OF COMPLEX DATA

► Implicit Models

high-dimensional space



latent space

z

$z \sim \mu_z$

G_θ

$p(x)$ defined implicitly with

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

...

GAN
NormFlow

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]
 - and weaker distances [Wasserstein GANs].

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

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

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

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

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

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

- 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].
- Special case: *Inverse Autoregressive Flows* (i.e. Jacobian triangular) explored in "Variational Inference with Inverse Autoregressive Flows", by [Kingma, Salimans & Welling, NIPS'16].

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

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

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

- 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].
- Special case: *Inverse Autoregressive Flows* (i.e. Jacobian triangular) explored in "Variational Inference with Inverse Autoregressive Flows", by [Kingma, Salimans & Welling, NIPS'16].
- More recently: RealNVP [Dinh et al'16], Glow [Kingma et al.'18].

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]

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

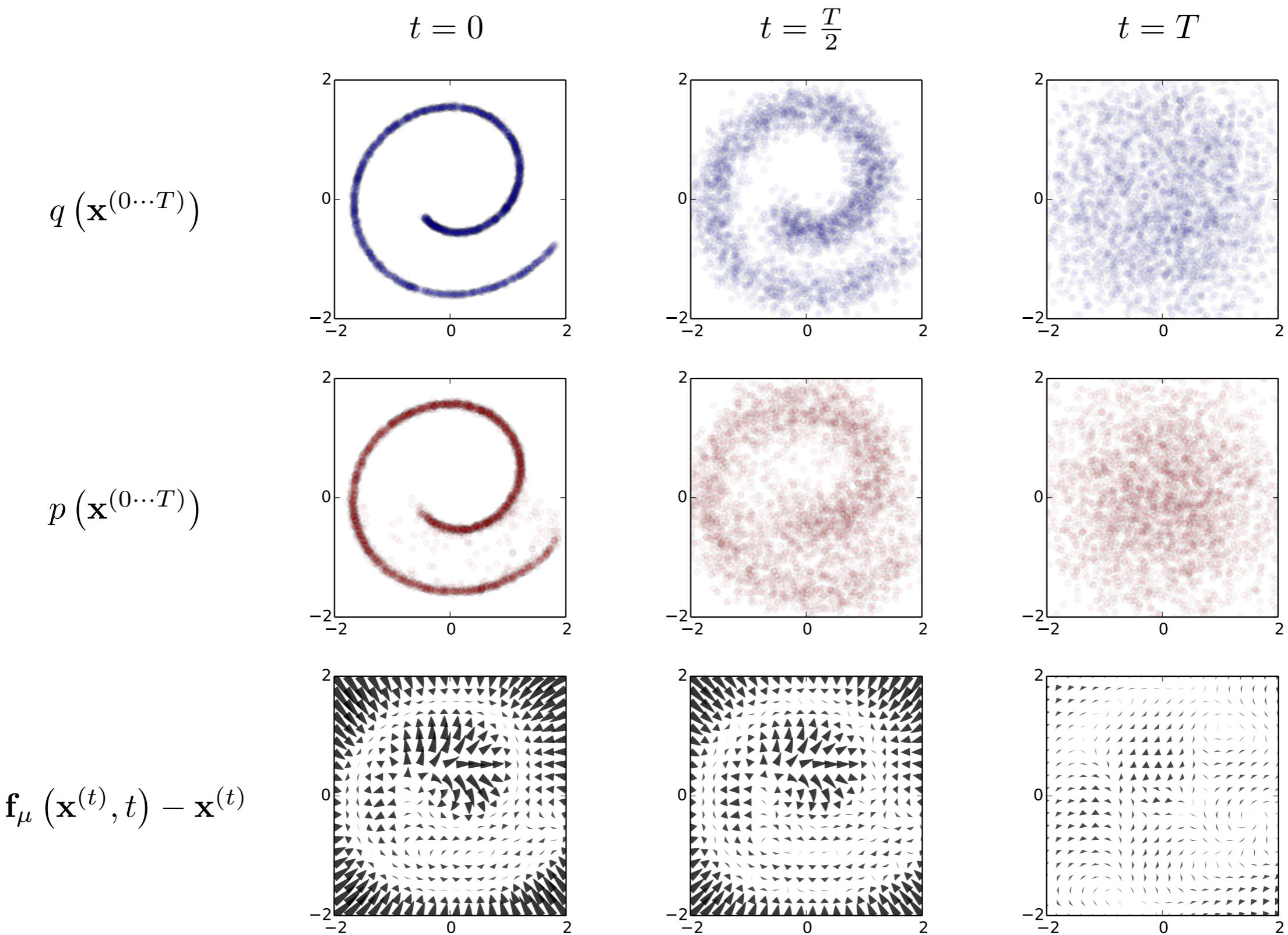
- The generative model learns how to reverse the diffusion:

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

- 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.
- Issue: cursed by the dimensionality

Diffusion and Non-equilibrium Thermodynamics

[Sohl-Dickstein et al.'15]



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).

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:

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

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

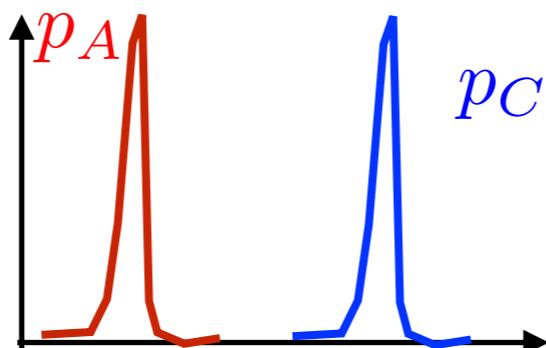
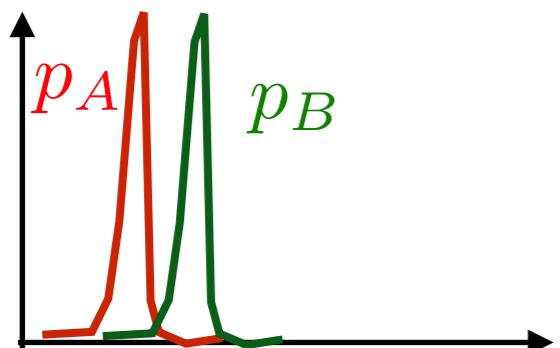
$$D_f(Q, P) = \sup_{|g|<\infty, g(\mathcal{X}) \subseteq \text{dom}(f^*)} \mathbb{E}_Q[g(x)] - \mathbb{E}_P[f^*(g(x))] .$$

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 .



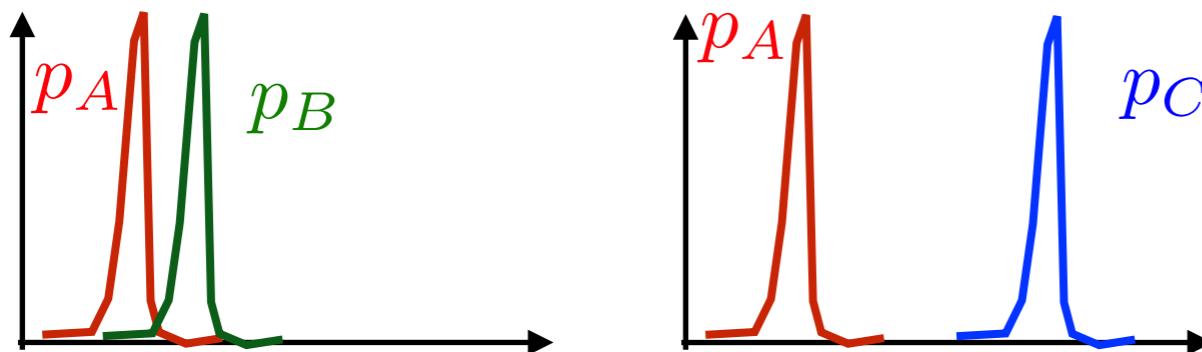
$$W(p_A, p_B) \ll W(p_A, p_C)$$

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

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

ENERGY DISTANCES

- The (Euclidean) Energy Distance is defined as

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

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* (MMD):

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

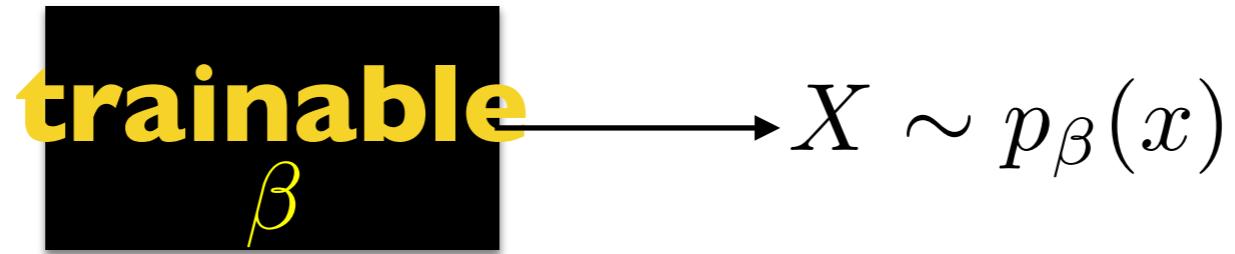
- \mathcal{H} is a Reproducing Kernel Hilbert Space associated with the so-called *triangular gap* Kernel associated with d :

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

Generative Adversarial Networks

[Goodfellow et al., '14]

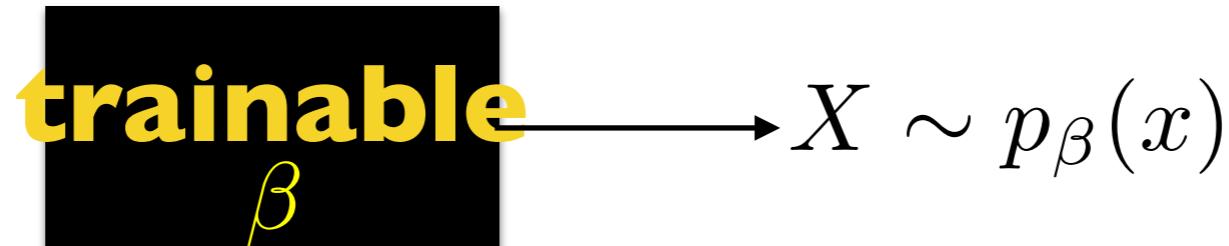
- Suppose we have a *trainable* black box generator:



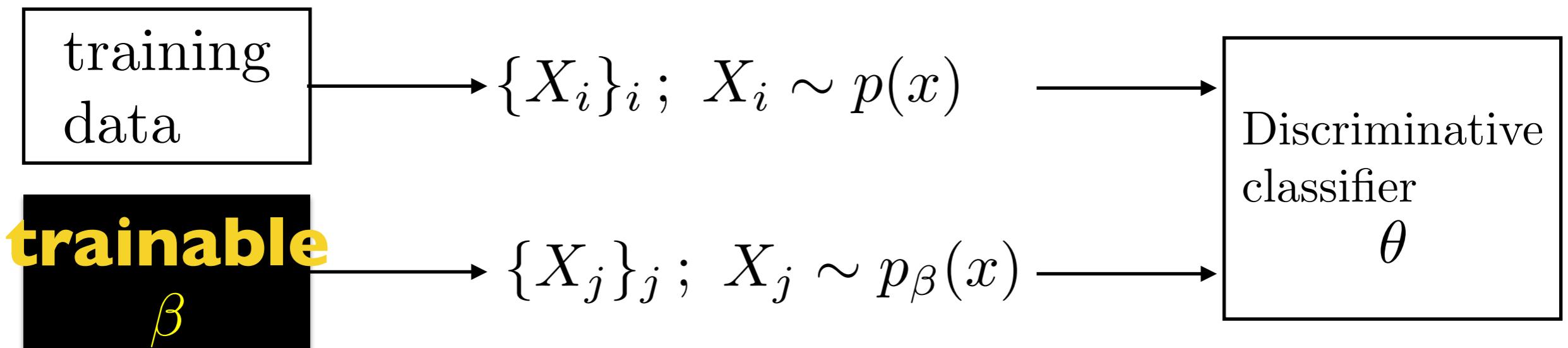
Generative Adversarial Networks

[Goodfellow et al., '14]

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



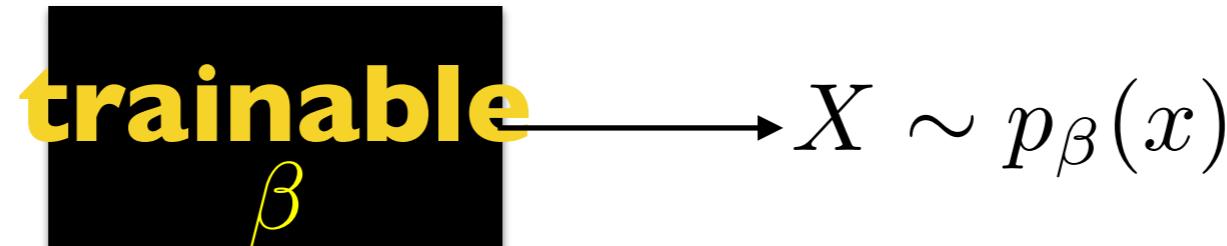
- Given observed data $\{X_i\}_i$; $X_i \sim p(x)$ how to force our generator to produce samples from $p(x)$?



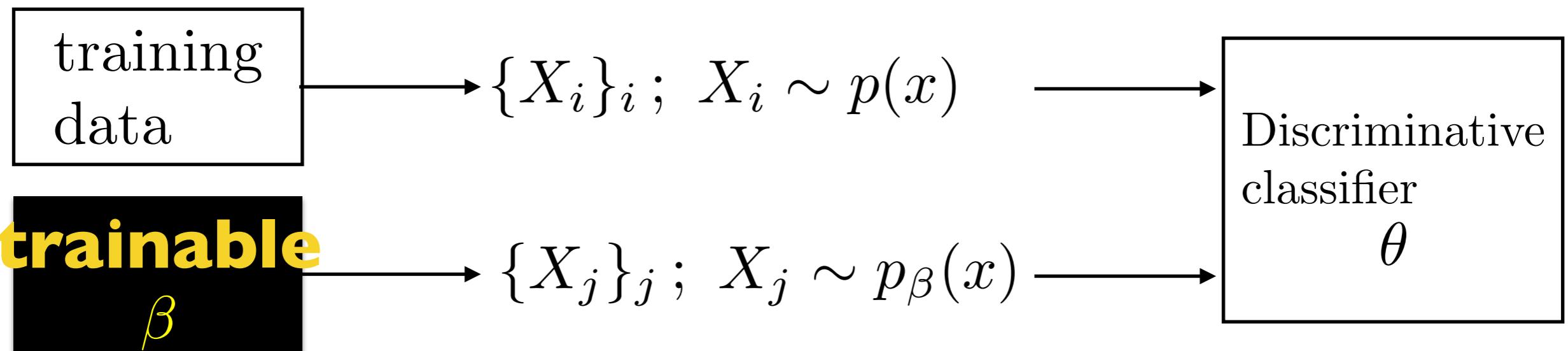
Generative Adversarial Networks

[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 $p(x)$?

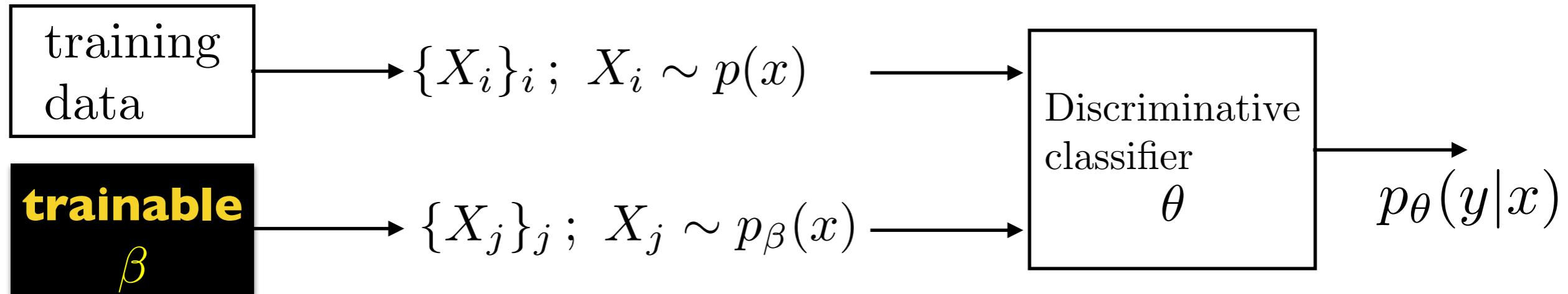


- The generator should make the classification task as hard as possible for any *discriminator*.

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

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

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 ,

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

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

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 follows 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
 - No guarantees of consistency
 - No guarantees of convergence (it is not gradient descent).

Generative Adversarial Training

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

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

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

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

Generative Adversarial Training

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

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

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

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

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

- 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:

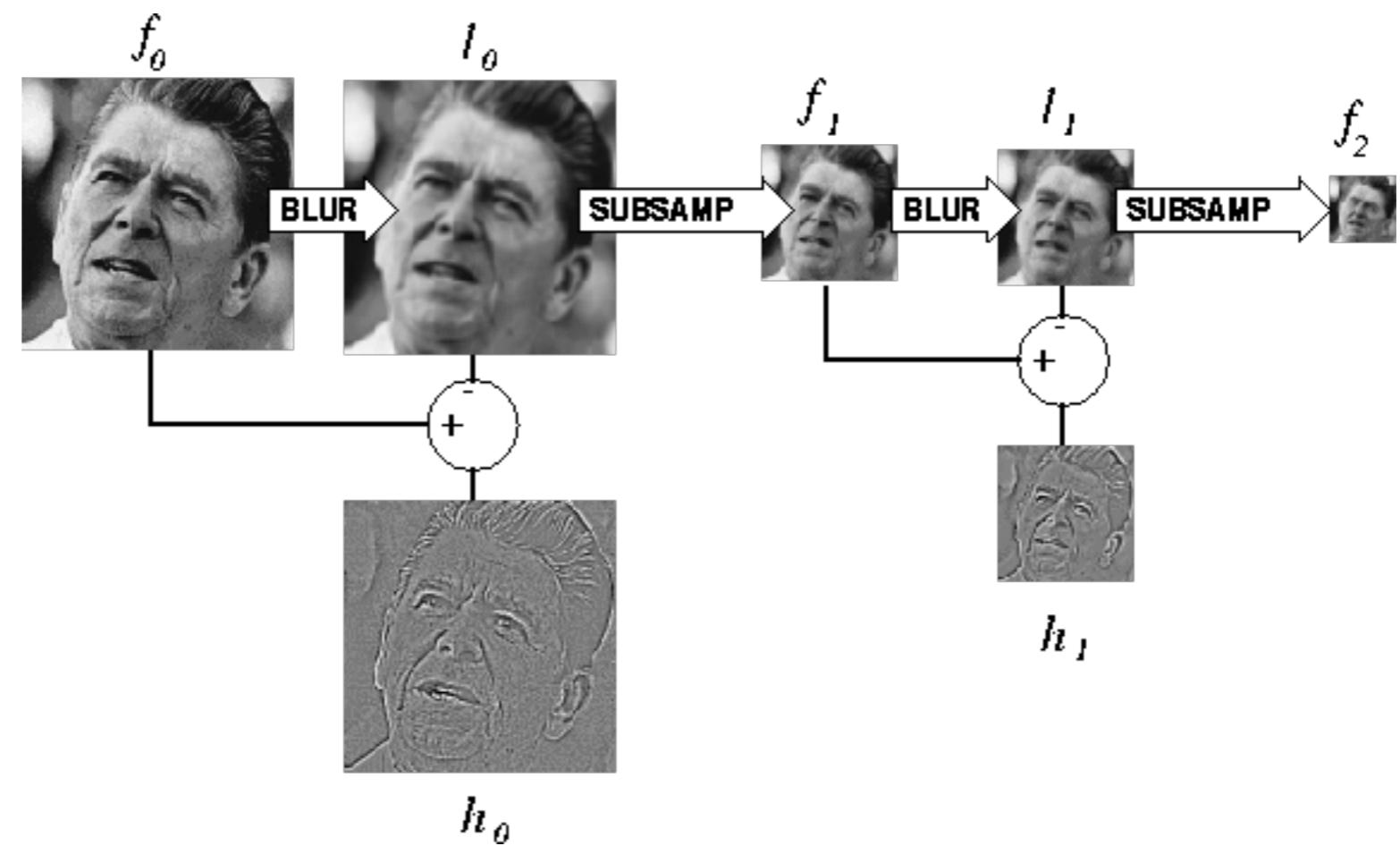
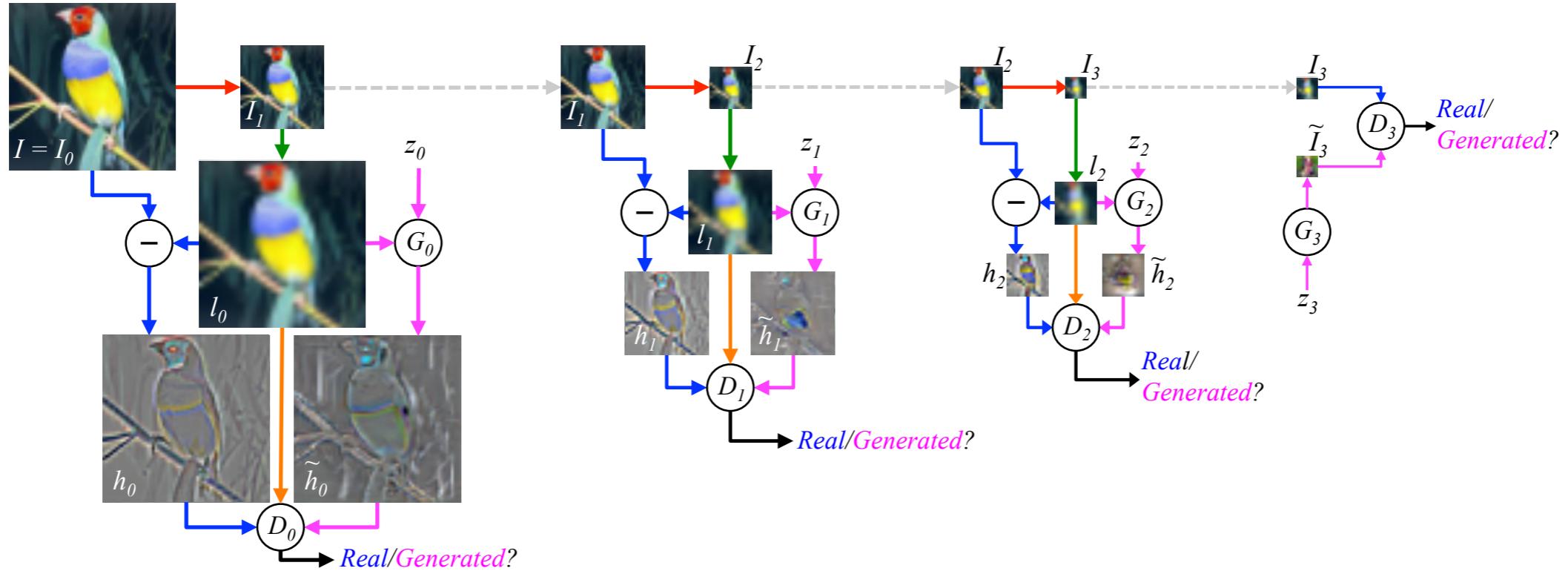


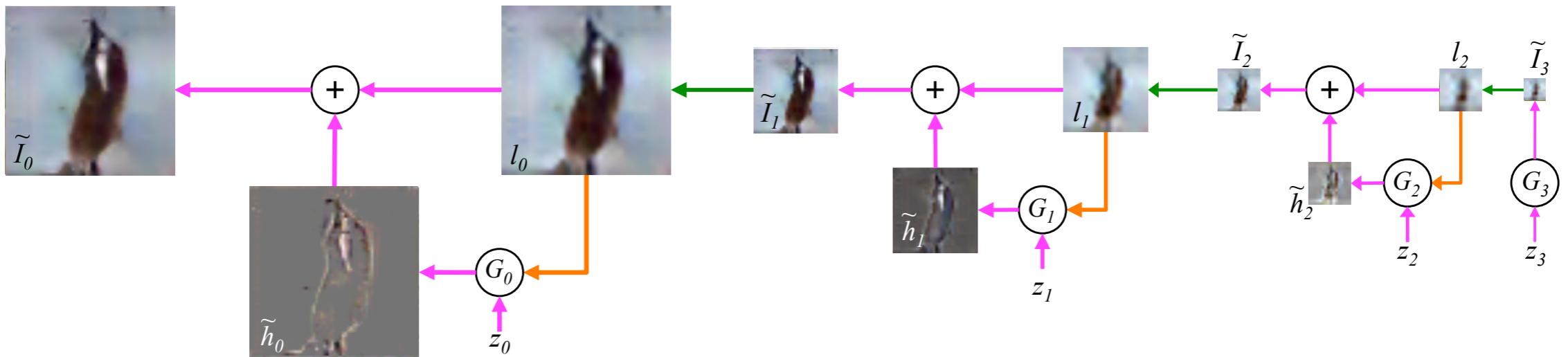
figure source: <http://sepwww.stanford.edu>

LAPGAN

- Training procedure:

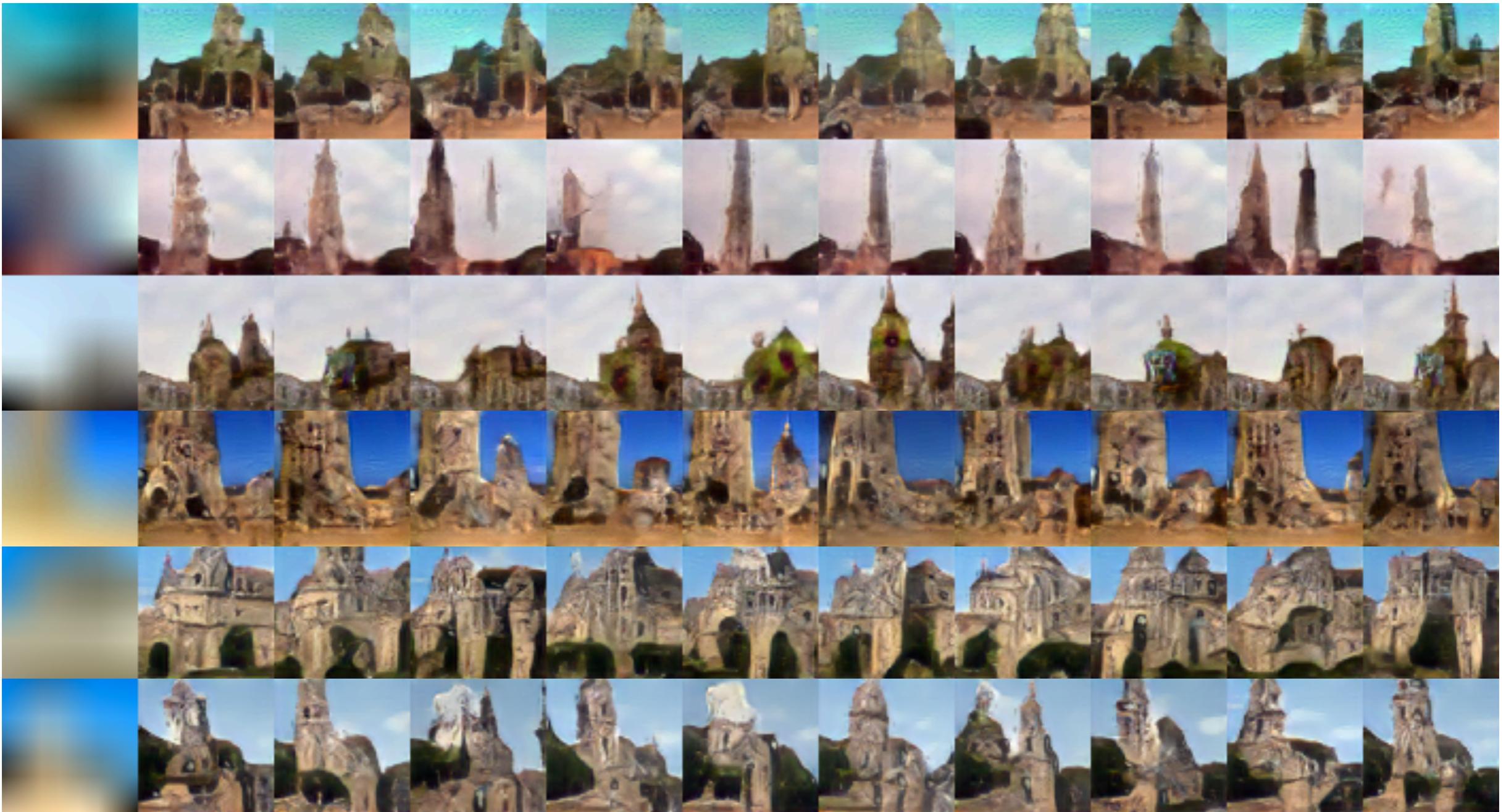


- Sampling procedure:



LAPGAN

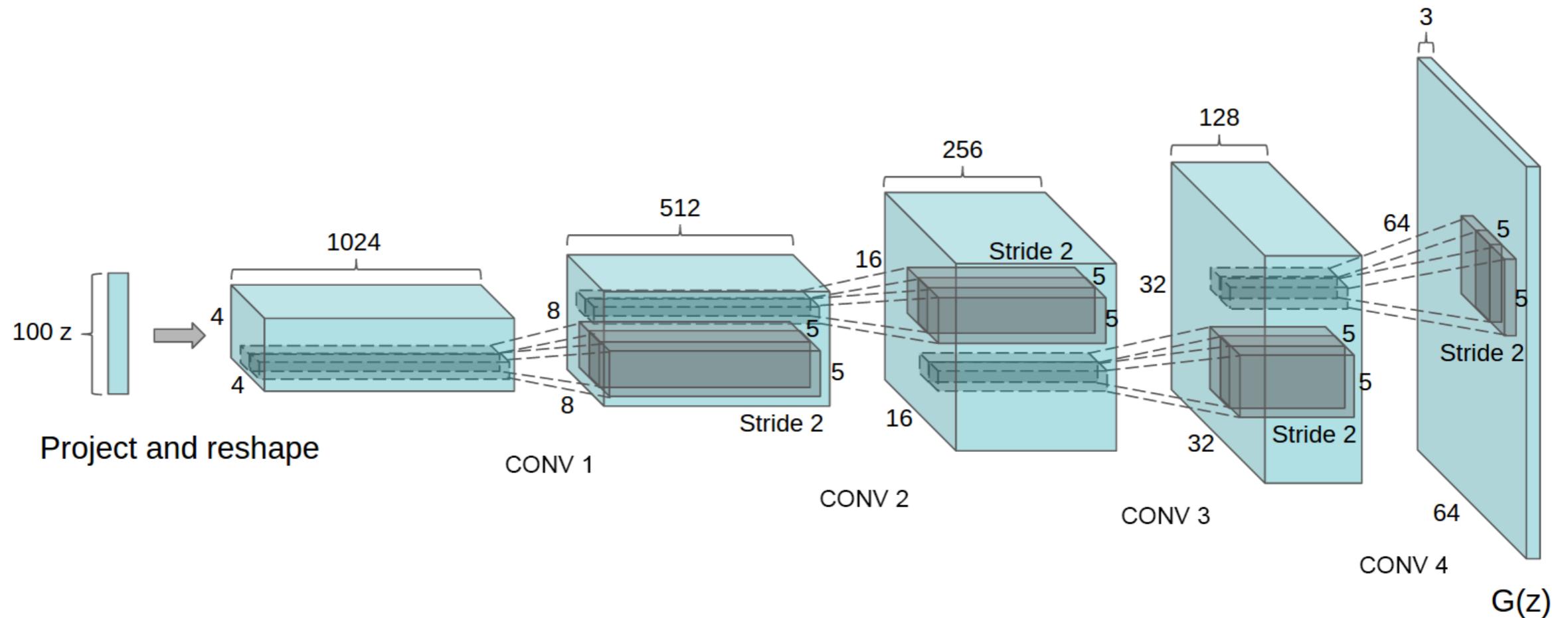
- Samples generated from the model:



DC-GAN

[Radford et al.'16]

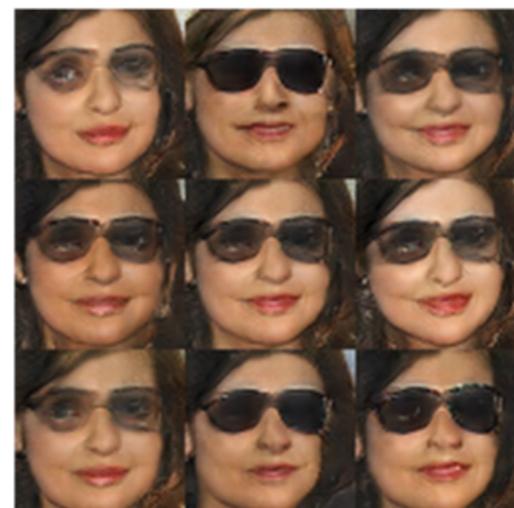
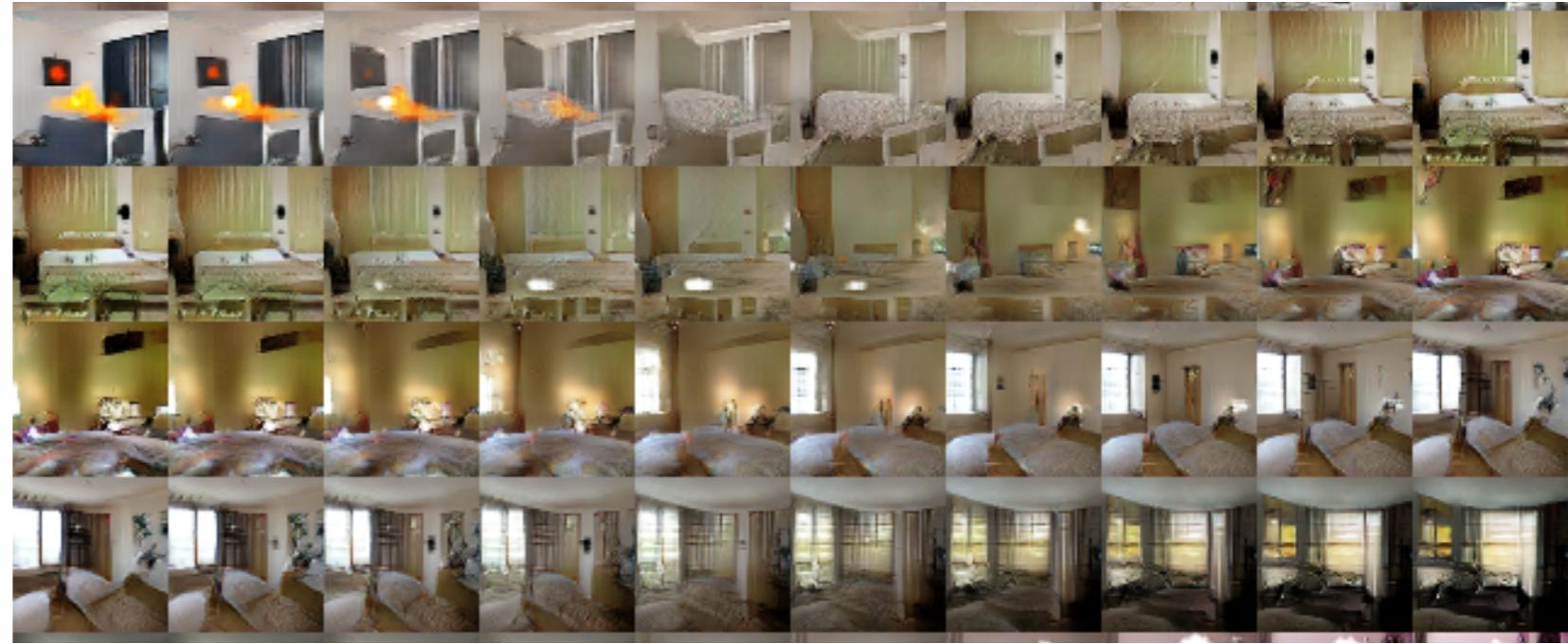
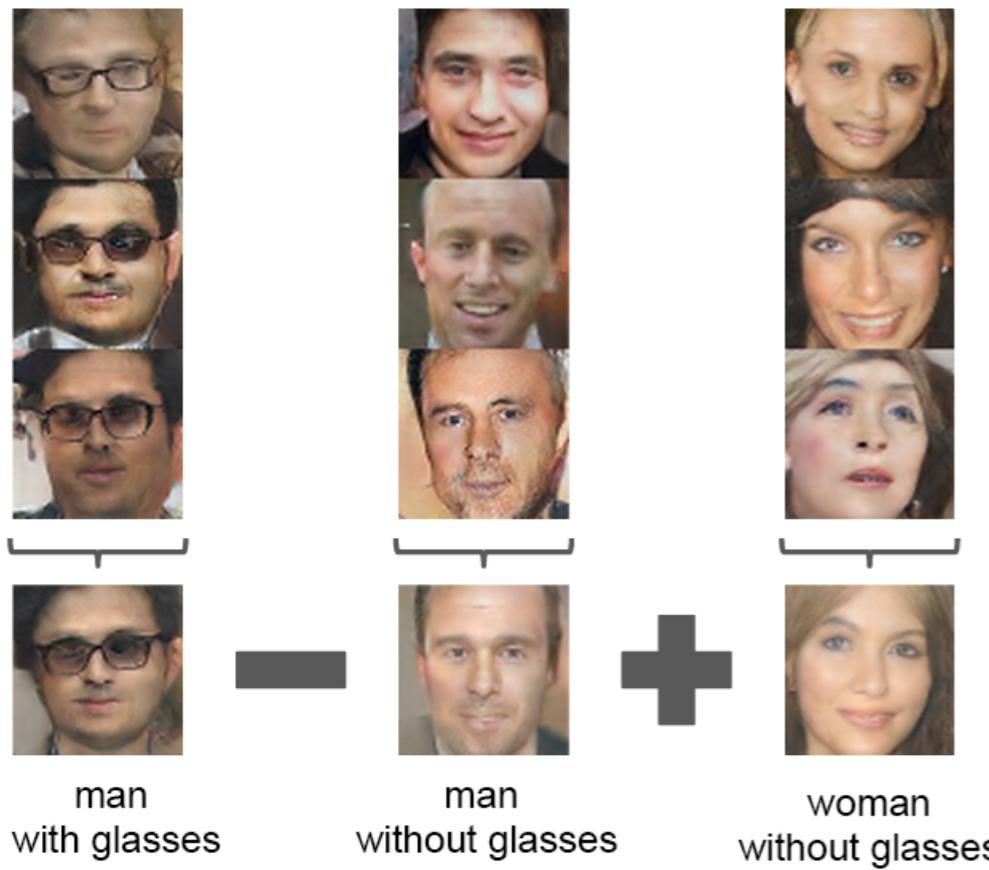
- Improved multi-scale architecture and Batch-Normalization:



DC-GAN

[Radford et al.'16]

- Improved multi-scale architecture and Batch-Normalization:



GANs and Optimal Transport

- So far, we have measured/trained density models using log-likelihood:

Model : $p(x; \theta)$

$$E(\theta) = \frac{1}{L} \sum_{l \leq L} \log p(x_l; \theta).$$

- This requires ability to measure model likelihoods.
 - (or at least a good lower bound as in variational autoencoders).

GANs and Optimal Transport

- So far, we have measured/trained density models using log-likelihood:

Model : $p(x; \theta)$

$$E(\theta) = \frac{1}{L} \sum_{l \leq L} \log p(x_l; \theta).$$

- This requires ability to measure model likelihoods.
 - (or at least a good lower bound as in variational autoencoders).
- The underlying “distance” in the space of distributions is the Kullback-Liebler divergence

$$KL(p_r \parallel p_m) = \int \log \left(\frac{p_r(x)}{p_m(x)} \right) p_r(x) d\mu(x) ,$$

- Estimated from finite data sample with

$$\widehat{KL}(p_r \parallel p_m) = \frac{1}{L} \sum_{l \leq L} \left(\frac{p_r(x_l)}{p_m(x_l; \theta)} \right) \propto -E(\theta) + C .$$

GANs and Optimal Transport

- Original GANs are associated with Jensen-Shannon divergence

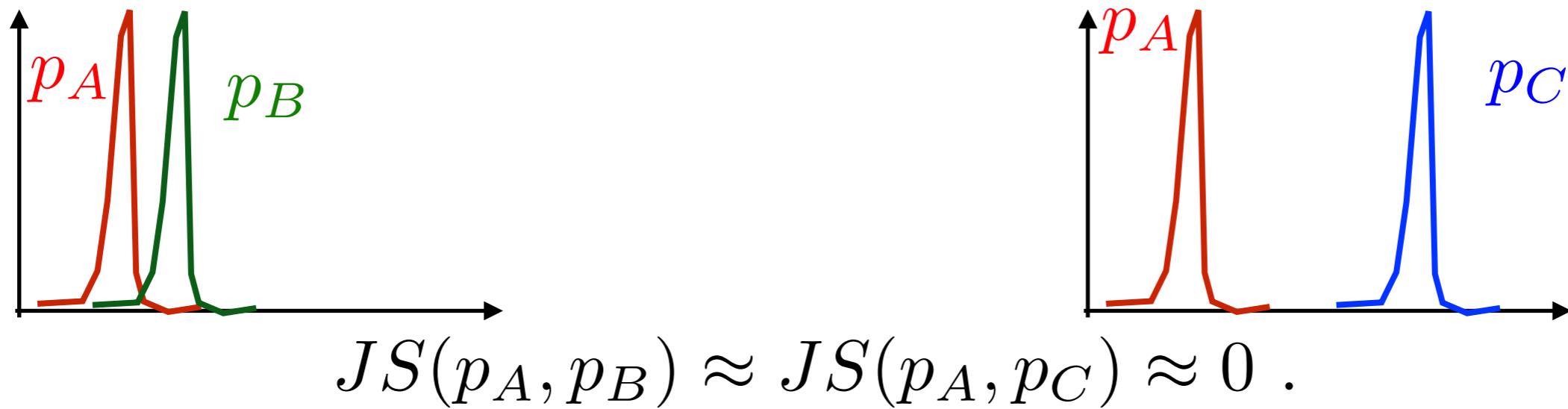
$$JS(P||Q) = \frac{1}{2} (D_{KL}(P||1/2(P + Q)) + D_{KL}(Q||1/2(P + Q))) .$$

GANs and Optimal Transport

- Original GANs are associated with Jensen-Shannon divergence

$$JS(P||Q) = \frac{1}{2} (D_{KL}(P||1/2(P+Q)) + D_{KL}(Q||1/2(P+Q))) .$$

- Limitations:
 - Expression of densities not always available.
 - These metrics require the distributions to be absolutely continuous with respect to each other, e.g. by having densities wrt Lebesgue.
 - May be too ‘strong’ a distance for many applications:



Earth-Mover Distance

- Alternatively, we can consider the Earth-Mover Distance (EMD)

$$W(P_r, P_m) = \inf_{\gamma \in \Pi(P_r, P_m)} \mathbb{E}_{(x,y) \sim \gamma} \{ \|x - y\| \},$$

$\Pi(P_r, P_m)$: Set of joint distributions in (x, y) whose marginals are P_r, P_m .

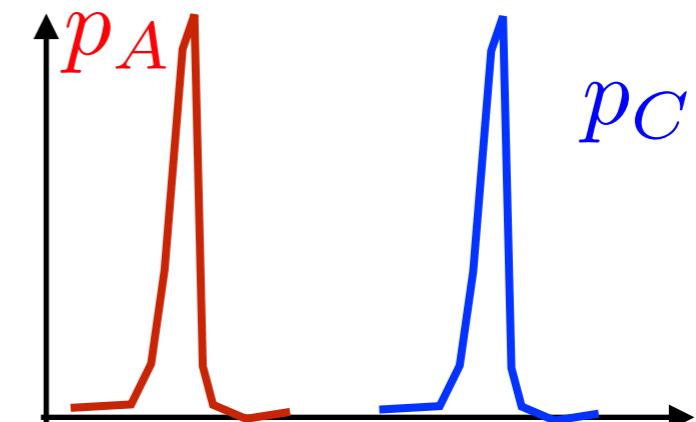
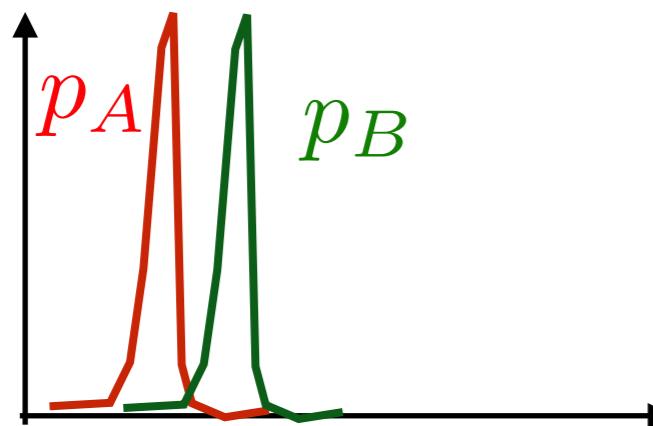
Earth-Mover Distance

- Alternatively, we can consider the Earth-Mover Distance (EMD)

$$W(P_r, P_m) = \inf_{\gamma \in \Pi(P_r, P_m)} \mathbb{E}_{(x,y) \sim \gamma} \{ \|x - y\| \},$$

$\Pi(P_r, P_m)$: Set of joint distributions in (x, y) whose marginals are P_r, P_m .

- Measures how much mass needs to be transported from P_r to P_m , measured with the Euclidean metric here (Wasserstein-1 distance).
- It defines a “weaker” topology in the space of probability distributions:



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

Kantorovich Duality

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

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

$f : \mathcal{X} \rightarrow \mathbb{R}$ continuous,

$\text{Lip}(f)$: Lipschitz constant of f :

$$\forall x, x' , |f(x) - f(x')| \leq \text{Lip}(f) \|x - x'\| .$$

Kantorovich Duality

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

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

$f : \mathcal{X} \rightarrow \mathbb{R}$ continuous,

$\text{Lip}(f)$: Lipschitz constant of f :

$$\forall x, x' , |f(x) - f(x')| \leq \text{Lip}(f) \|x - x'\| .$$

- 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 Lipschitz 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 criteria ρ :

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

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

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

- ## ► Triangle Inequality:

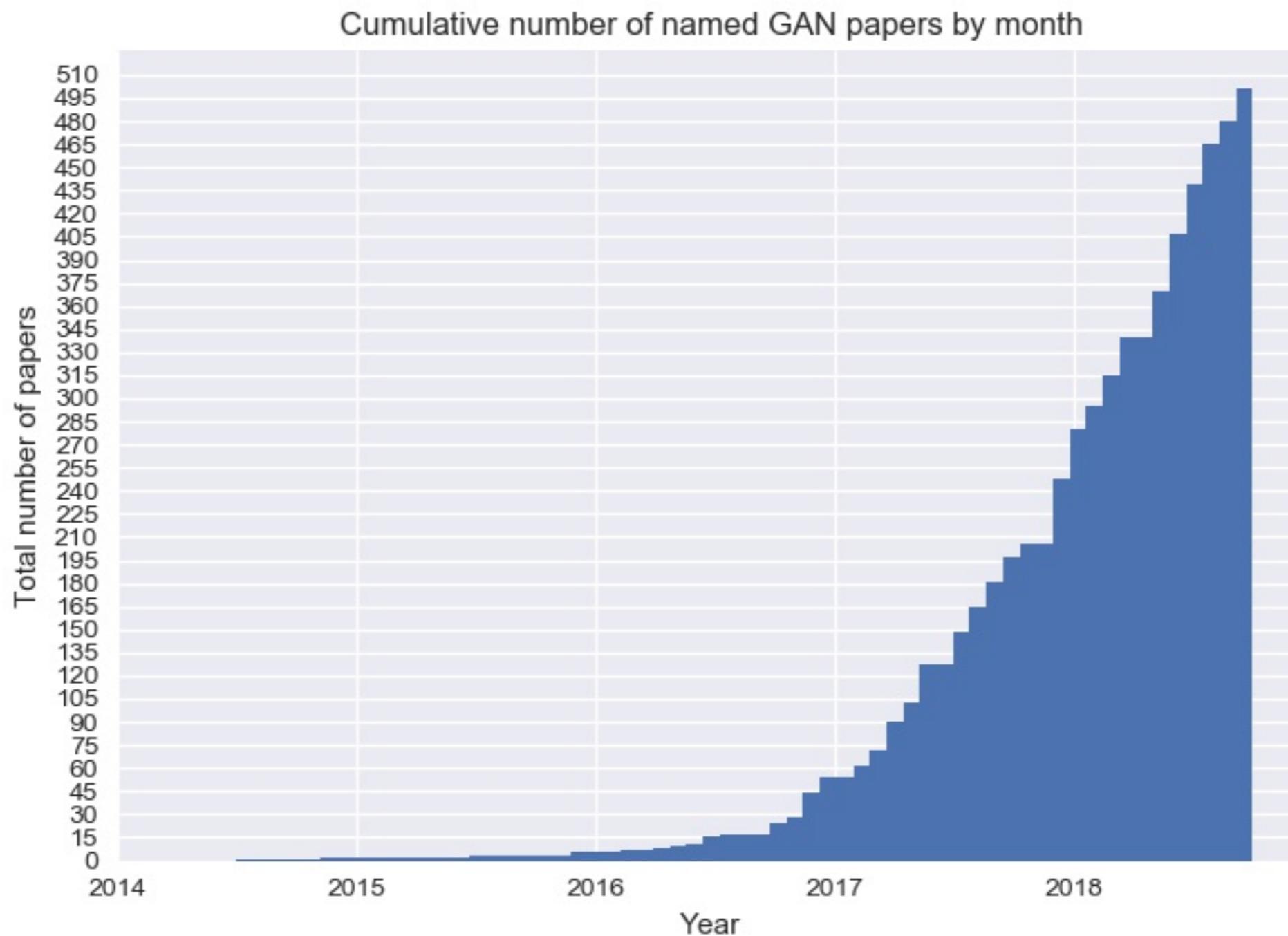
LIMITATIONS OF GAN MODELING

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

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 POPULAR!



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]
- OT with mixture models: **Wasserstein Autoencoders**
[Tolstikhin, Bousquet, Gelly, Schoelkopf,'17].
- Large Scale Gan Training [anonymous, ICLR'19 subm]

CONDITIONAL GANS

Labels to Street Scene

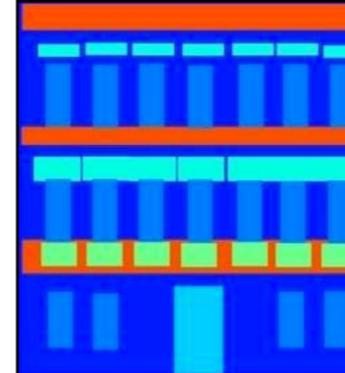


input



Aerial to Map

Labels to Facade



input



output

BW to Color



inpu



output

An aerial photograph of a residential area. The image shows several streets lined with houses, mostly single-family homes with dark roofs and light-colored exteriors. The houses are arranged in a grid pattern, with some larger lots and green spaces visible between the more densely packed areas. The streets are paved and appear relatively clean. There are a few trees scattered throughout the neighborhood, primarily along the sidewalks and in the larger yards. The overall impression is of a typical suburban or urban residential community.

input

O



input



output



input



output

“Image-to-Image translation with Conditional Adversarial Networks”, Isola et al,’16

CYCLE GAN [ZHU, PARK, ISOLA, EFROS]

Monet \leftrightarrow Photos



Monet \rightarrow photo

Zebras \leftrightarrow Horses



zebra \rightarrow horse

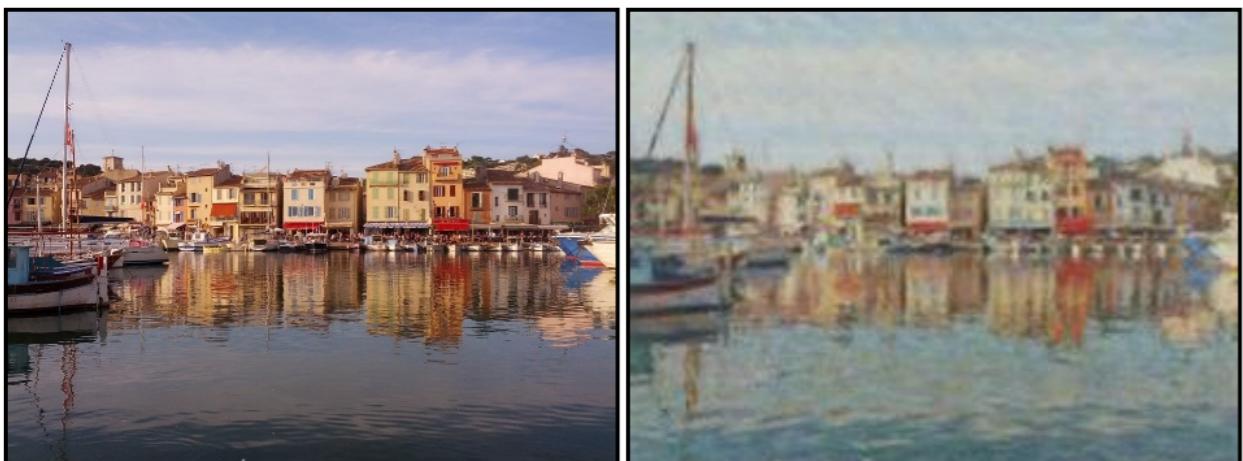
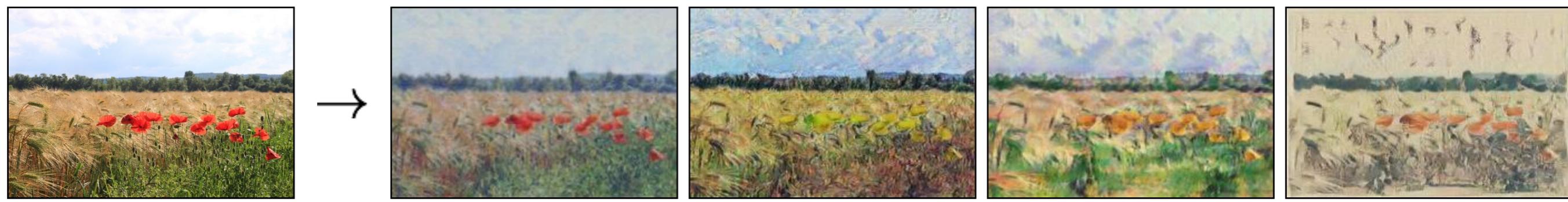


photo \rightarrow Monet



horse \rightarrow zebra



Photograph

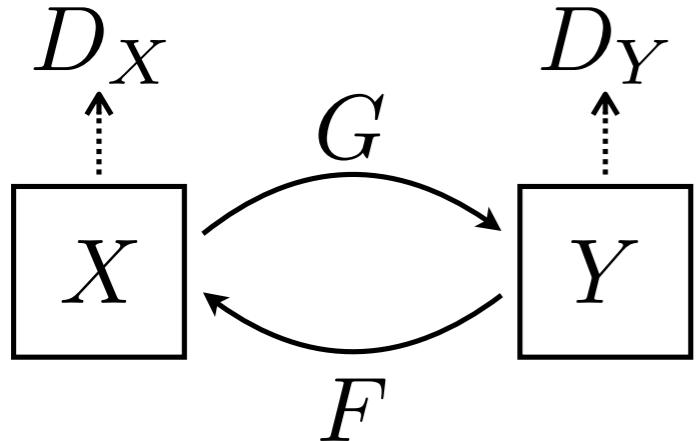
Monet

Van Gogh

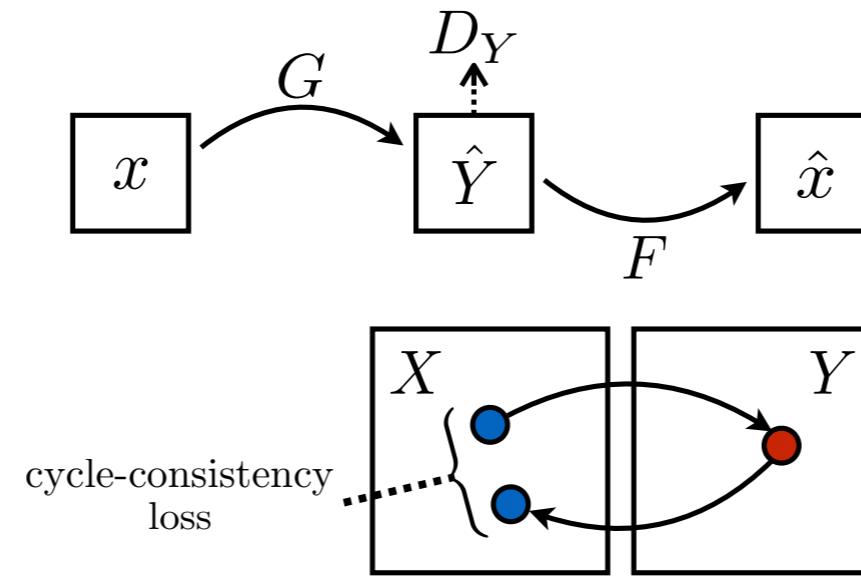
Cezanne

Ukiyo-e

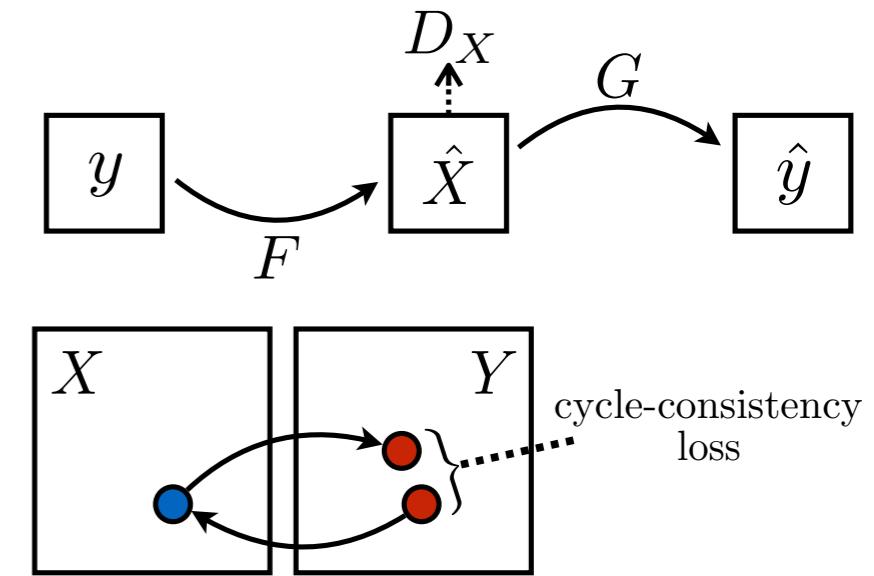
CYCLE GAN



(a)

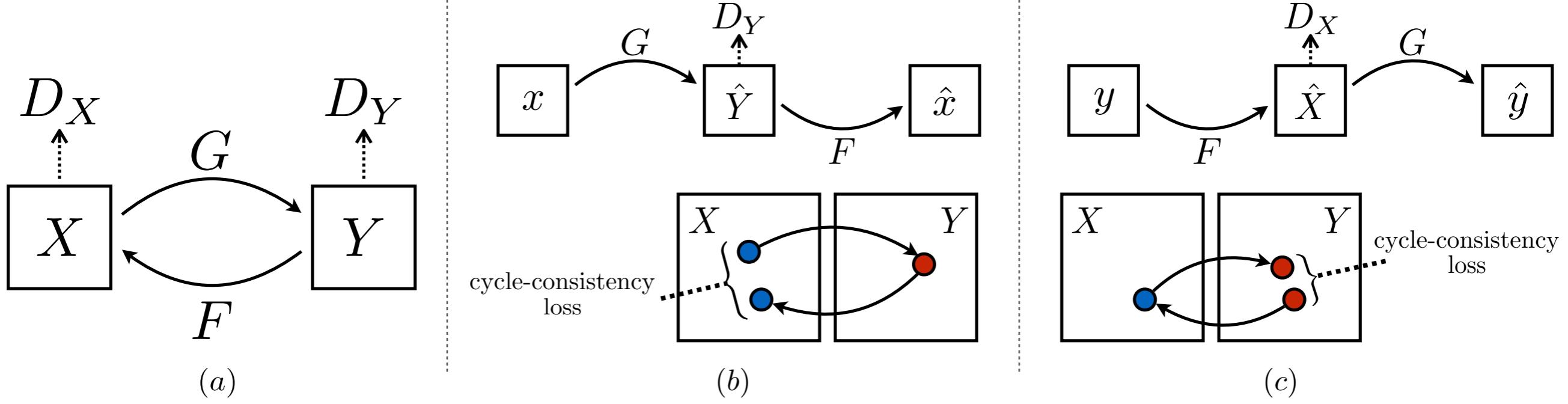


(b)



(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 training.

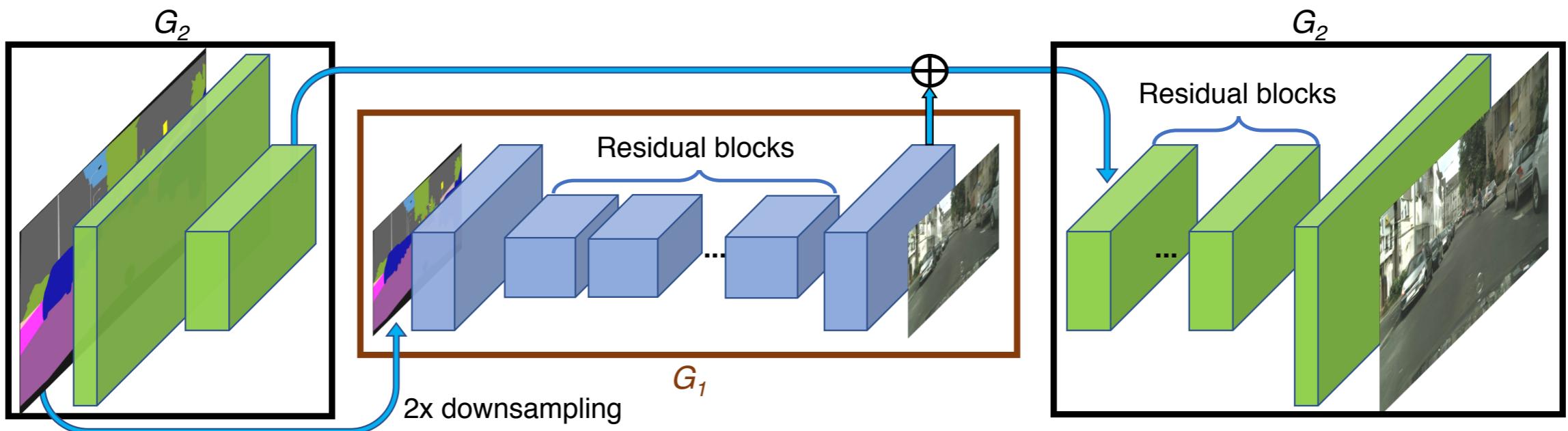


PHOTO REALISTIC GAN [WANG ET AL]

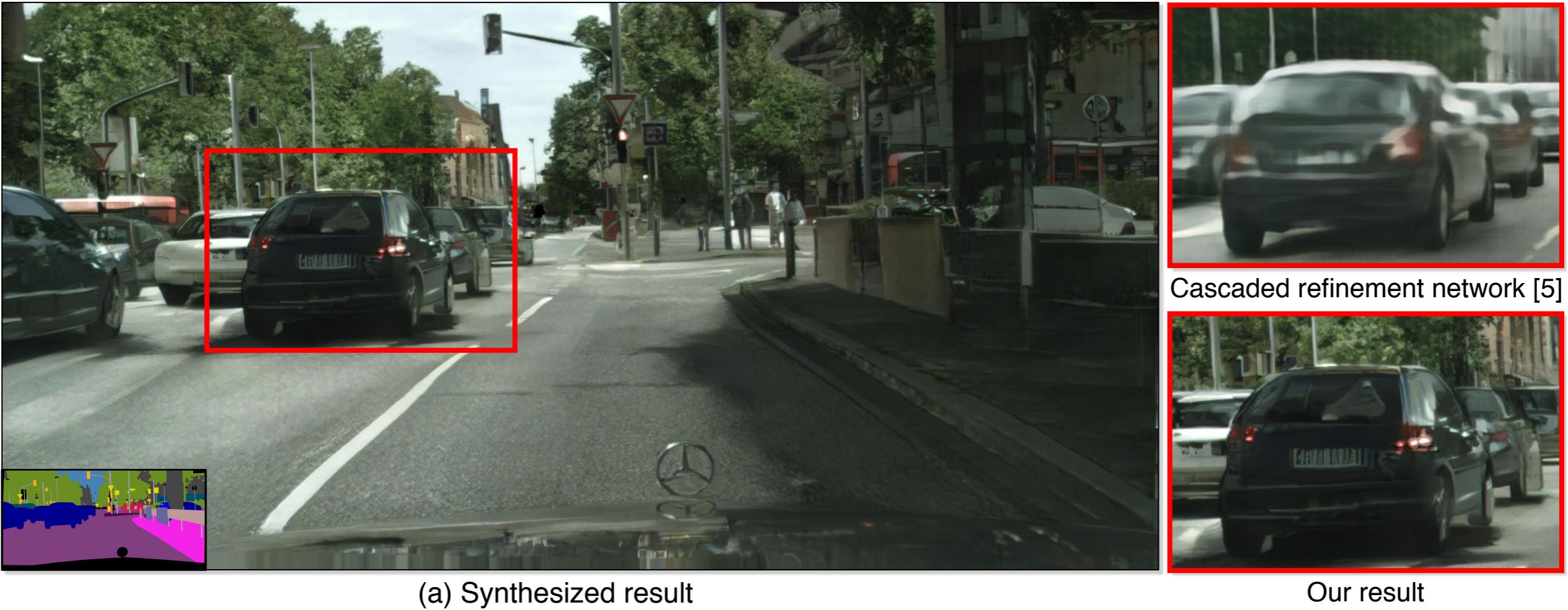
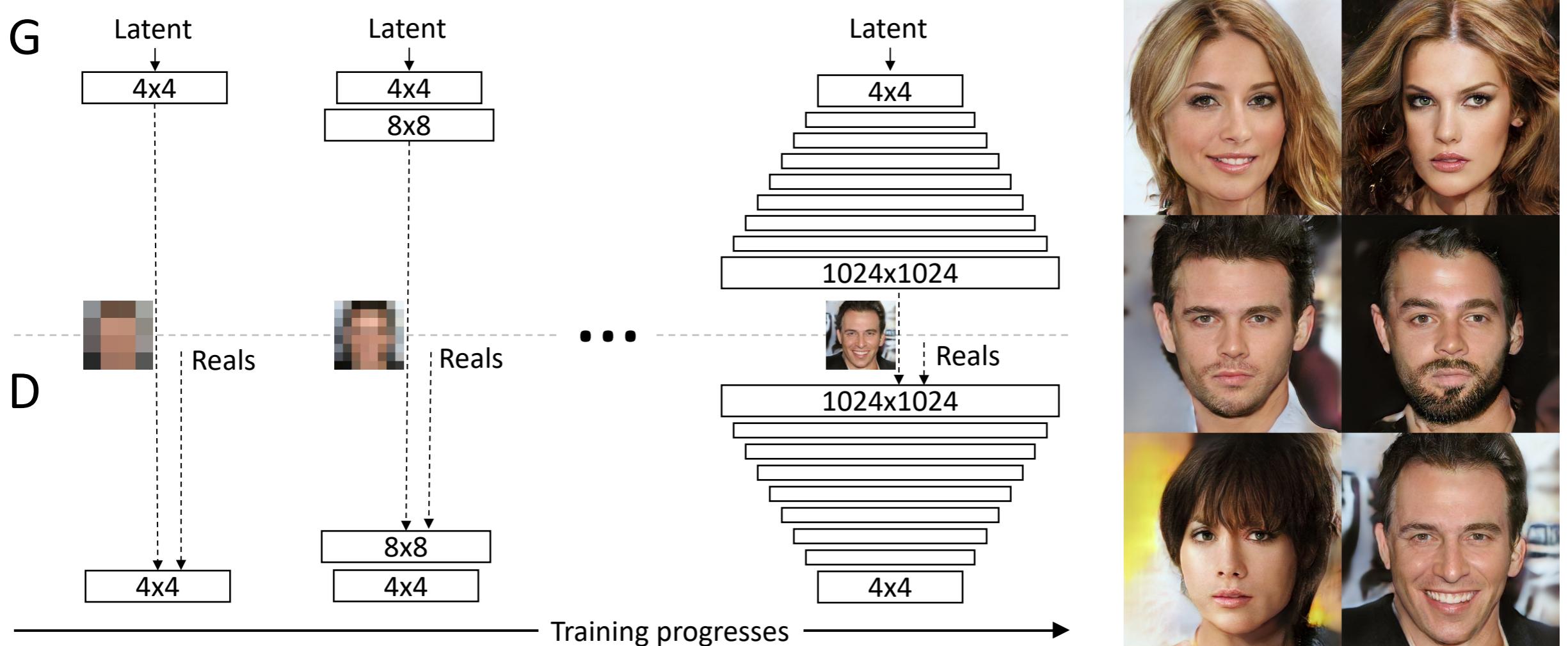


PHOTO-REALISTIC GAN



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

CHURCHOUDOOR

BICYCLE

TVMONITOR

GENERATIVE LATENT OPTIMIZATION (GLO) [BOJANOWSKY, SZLAM ET AL'18]

- GAN image models show high-quality reconstructions.
- Is this due to the inductive bias from the deep convolutional architecture? or from the adversarial training protocol?

GENERATIVE LATENT OPTIMIZATION (GLO) [BOJANOWSKY, SZLAM ET AL'18]

- GAN image models show high-quality reconstructions.
- Is this due to the inductive bias from the deep convolutional architecture? or from the adversarial training protocol?
- The authors consider the following problem:

$$\min_{\theta \in \Theta} \frac{1}{N} \sum_{i \leq N} \min_{z_i \in \mathcal{Z}} \ell(x_i, G_\theta(z_i)) .$$

$G_\theta(z)$: DC-GAN architecture.

- Non-linear Dictionary learning.
- Latent space has simple uniform prior over hypercube.

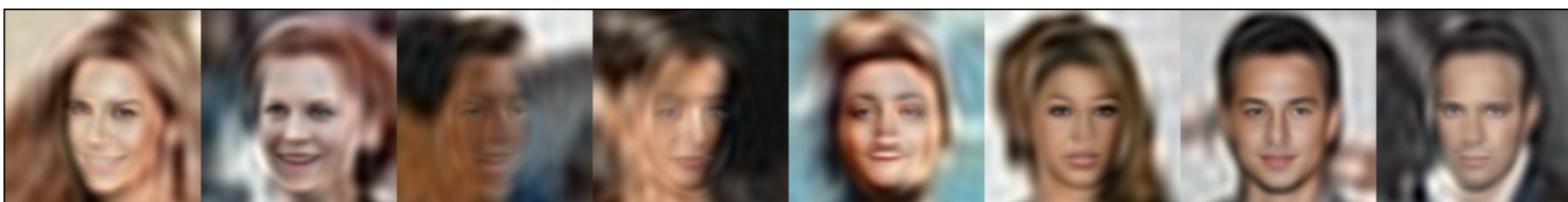
GENERATIVE LATENT OPTIMIZATION (GLO) [BOJANOWSKY, SZLAM ET AL'18]

- Some reconstruction results.

Original images



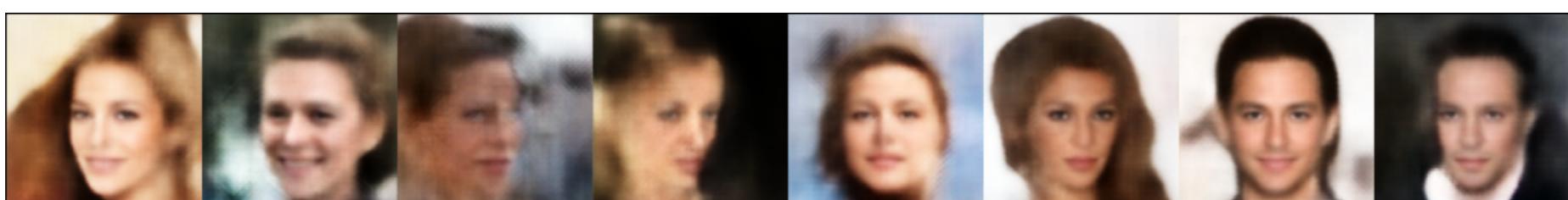
PCA



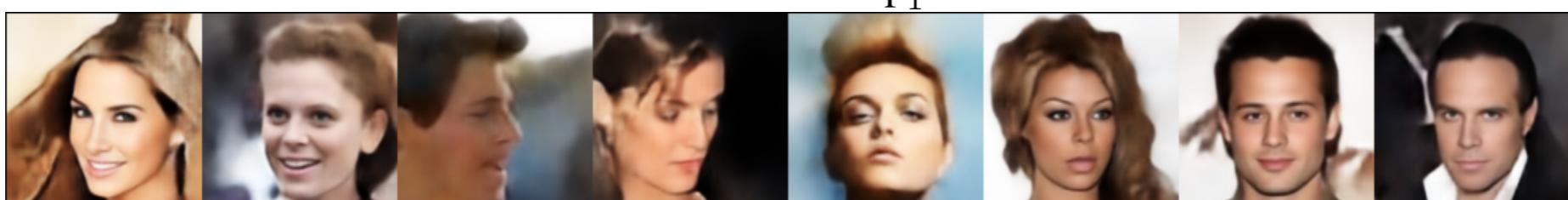
GAN with MSE



GLO with MSE



GLO with Lap₁



Autoregressive Models

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

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

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

Autoregressive Models

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

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

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

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

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

Autoregressive Models

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

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

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

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

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

- Q: In which situations is it better to use the factorized?

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.

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

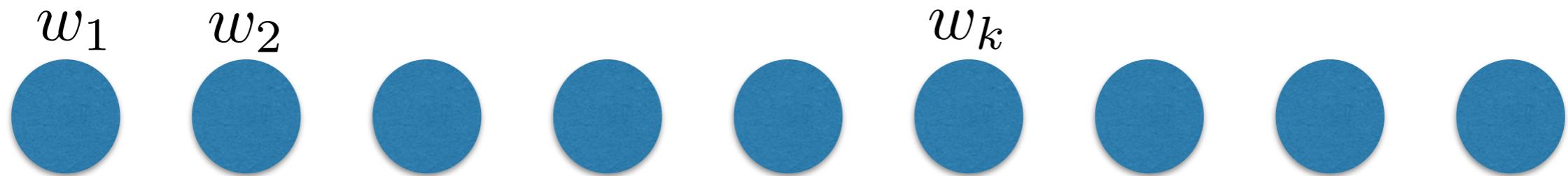
Word2vec [Mikolov et al.'13].

- Unsupervised learning “success story”.



Word2vec [Mikolov et al.'13].

- Unsupervised learning “success story”.

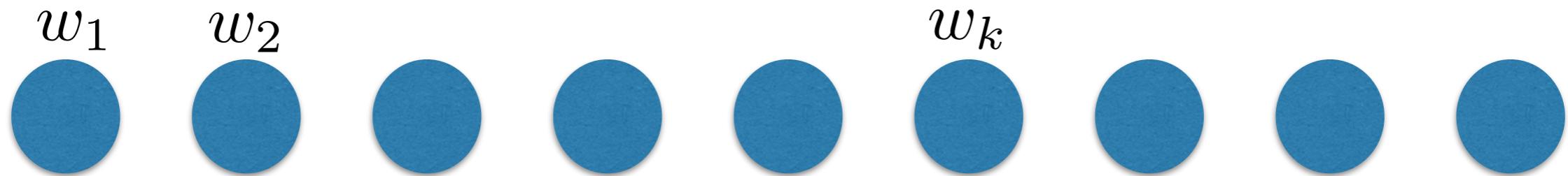


- Language creates a notion of similarity between words:

words w_1 , w_2 are similar if they are “exchangeable”
i.e., they appear often within the same context.

Word2vec [Mikolov et al.'13].

- Unsupervised learning “success story”.



- Language creates a notion of similarity between words:

words w_1 , w_2 are similar if they are “exchangeable”
i.e., they appear often within the same context.

- Goal: find a word representation $\Phi(w_i) \in \mathbb{R}^d$ that expresses this similarity as a dot product

$$\text{sim}(w_i, w_j) \approx \langle \Phi(w_i), \Phi(w_j) \rangle .$$

Word2vec [Mikolov et al.'13].

- Main idea: Skip-gram with negative sampling.
- Construct a “training set”
 - positive pairs $\mathcal{D} = \{(w_k, c_k)\}_k$ of (words, contexts) appearing in a huge language corpus.
 - negative pairs $\mathcal{D}' = \{(w_{k'}, c_{k'})\}_{k'}$ of (words, contexts) not appearing in the corpus.

Word2vec [Mikolov et al.'13].

- Main idea: Skip-gram with negative sampling.
- Construct a “training set”
 - positive pairs $\mathcal{D} = \{(w_k, c_k)\}_k$ of (words, contexts) appearing in a huge language corpus.
 - negative pairs $\mathcal{D}' = \{(w_{k'}, c_{k'})\}_{k'}$ of (words, contexts) not appearing in the corpus.
- Model the probability of a pair (w, c) being positive as

$$p(D = 1|c, w) = \sigma(\langle v_w, v_c \rangle), \quad v_w, v_c \in \mathbb{R}^d.$$
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Word2vec [Mikolov et al.'13].

- Main idea: Skip-gram with negative sampling.
- Construct a “training set”
 - positive pairs $\mathcal{D} = \{(w_k, c_k)\}_k$ of (words, contexts) appearing in a huge language corpus.
 - negative pairs $\mathcal{D}' = \{(w_{k'}, c_{k'})\}_{k'}$ of (words, contexts) not appearing in the corpus.
- Model the probability of a pair (w, c) being positive as

$$p(D = 1|c, w) = \sigma(\langle v_w, v_c \rangle), \quad v_w, v_c \in \mathbb{R}^d.$$
- Training with Maximum Likelihood:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

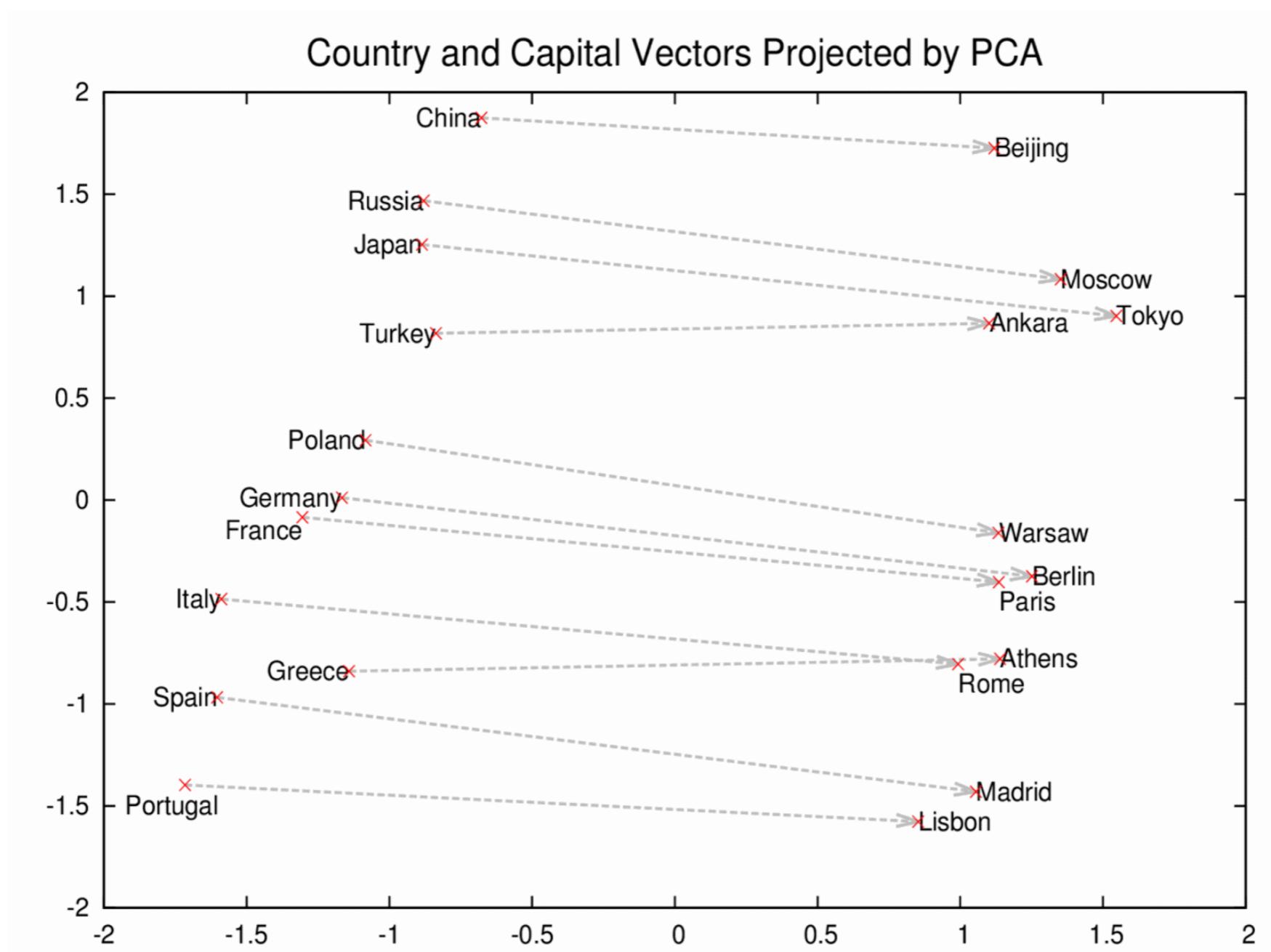
$$\arg \max_{\theta} \prod_{(w,c) \sim \mathcal{D}} p(D = 1|c, w, \theta) \prod_{(w,c) \sim \mathcal{D}'} p(D = 0|c, w, \theta)$$

$$\arg \max_{\theta} \sum_{(w,c) \sim \mathcal{D}} \log \sigma(\langle v_w, v_c \rangle) + \sum_{(w,c) \sim \mathcal{D}'} \log \sigma(-\langle v_w, v_c \rangle)$$

\mathcal{D} : positive contexts \mathcal{D}' : negative contexts

Word2vec [Mikolov et al.'13].

- Can be seen as an implicit matrix factorization using a mutual information criteria [Yoav & Goldberg,'14].
- Huge impact on Google's business bottom-line.



Video Prediction

- Rather than modeling the density of natural images

$$p(x) , \quad x \in \mathbb{R}^d$$

we may be also interested in modeling the conditional distributions

$$p(x_{t+1} | x_1, \dots, x_t)$$

where $(x_t)_t$ is temporally ordered data.

Video Prediction

- Rather than modeling the density of natural images

$$p(x) , \quad x \in \mathbb{R}^d$$

we may be also interested in modeling the conditional distributions
where $(x_t)_t$ is temporally ordered data. $p(x_{t+1}|x_1, \dots, x_t)$

- Similarly, can we find a signal representation $\Phi(x_t)$ that is consistent with the “video language” metric? i.e.

$$\langle \Phi(x_t), \Phi(x_s) \rangle \approx h(|t - s|)$$

- This is the objective of Slow Feature Analysis [Sejnowski et al'02, Cadieu & Olshausen'10 and many others].

Video Prediction

- [Mathieu, Couarie, LeCun, '16]: Conditional video prediction using CNNs and an adversarial cost



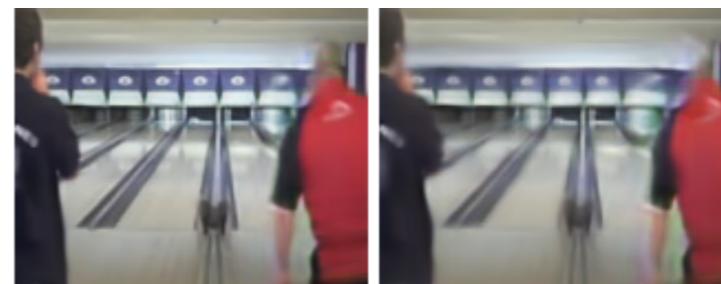
Ground truth



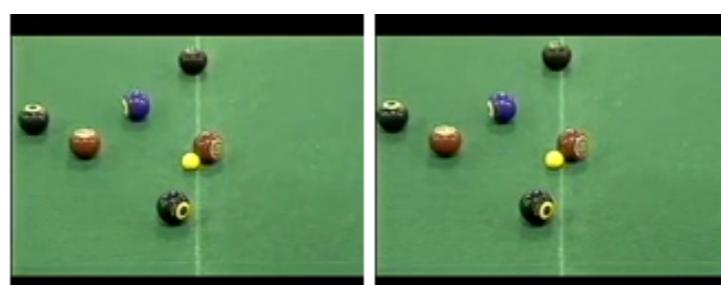
ℓ_2 result



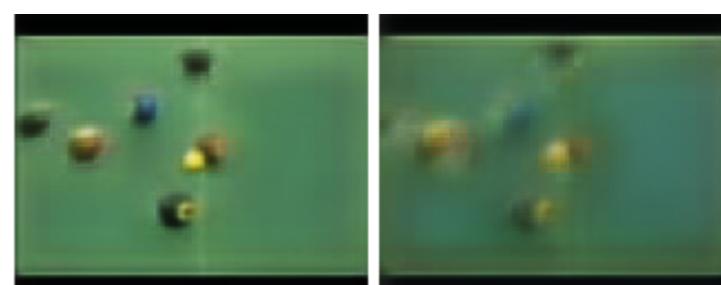
Adversarial result



Adversarial+GDL result



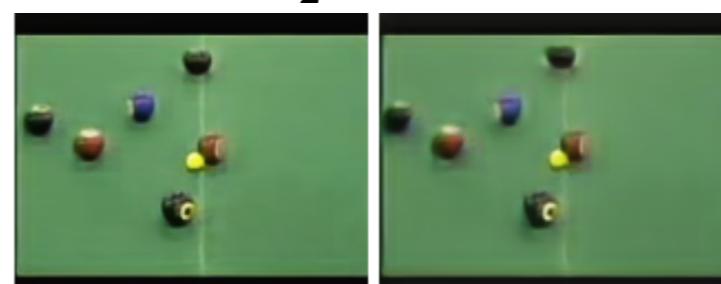
Ground truth



ℓ_2 result



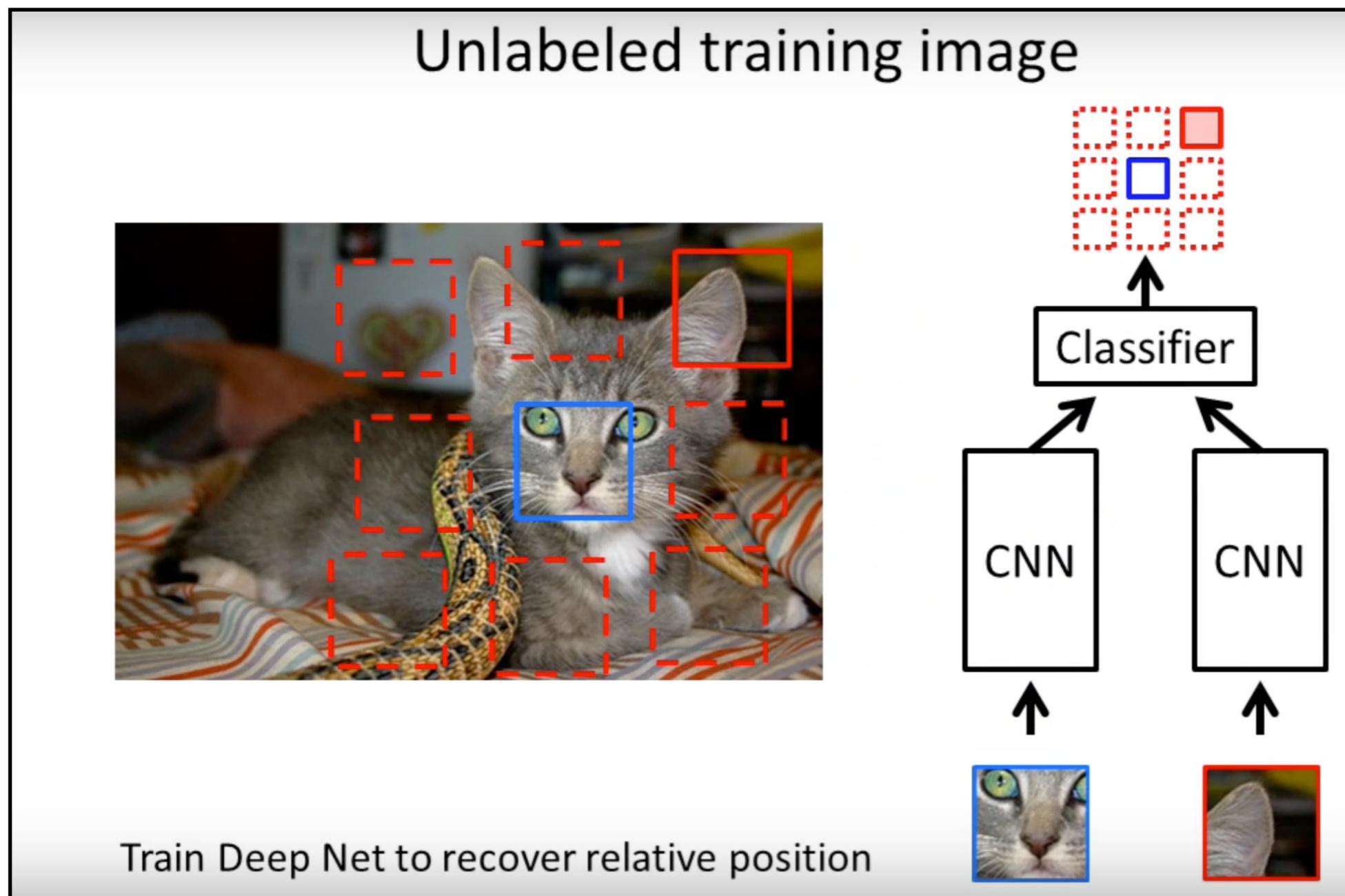
Adversarial result



Adversarial+GDL result

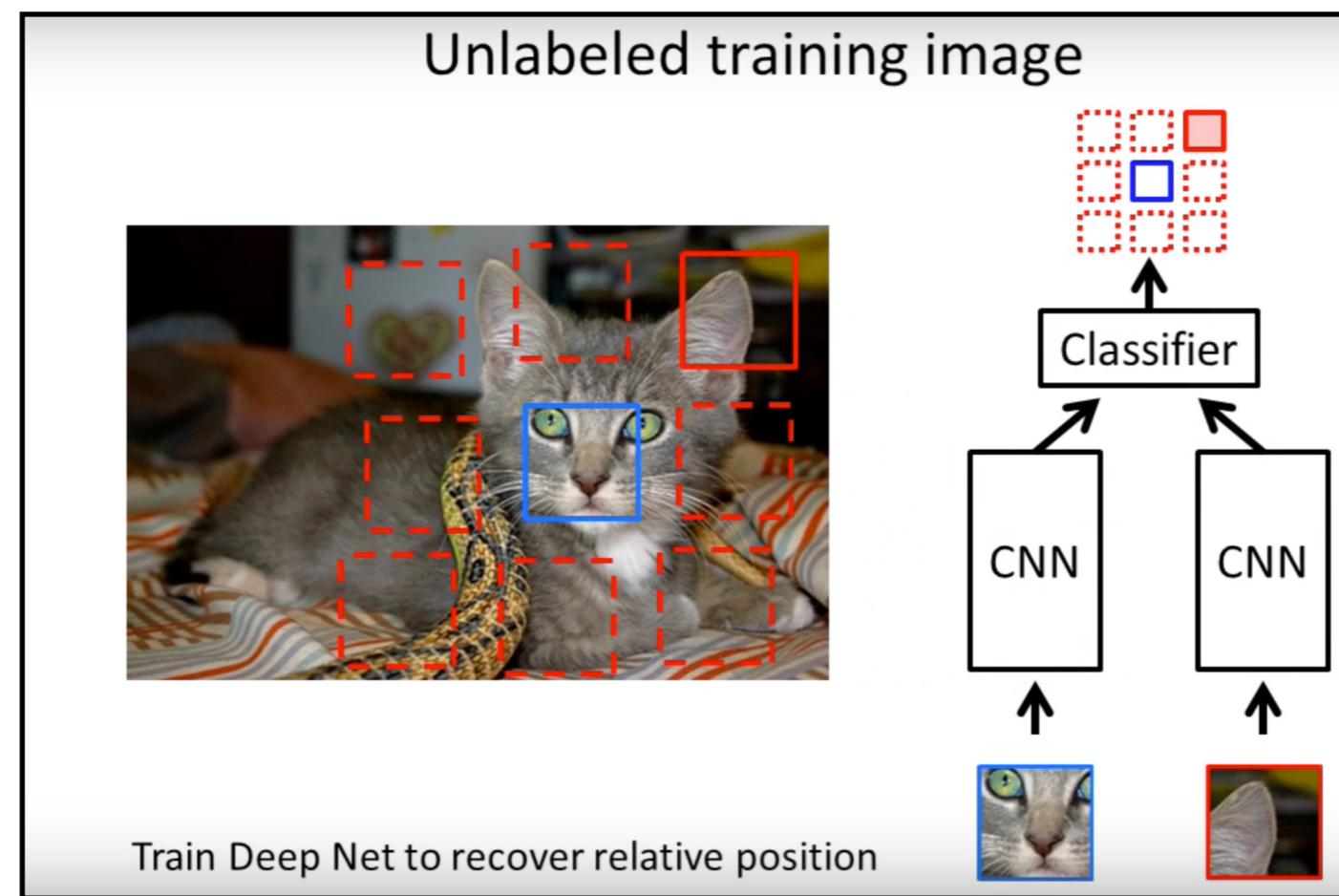
Patch Relative Configuration [Doerch et al.'15]

- Generalize the idea of positive, negative pairs to a multi-class classification problem about spatial configurations.



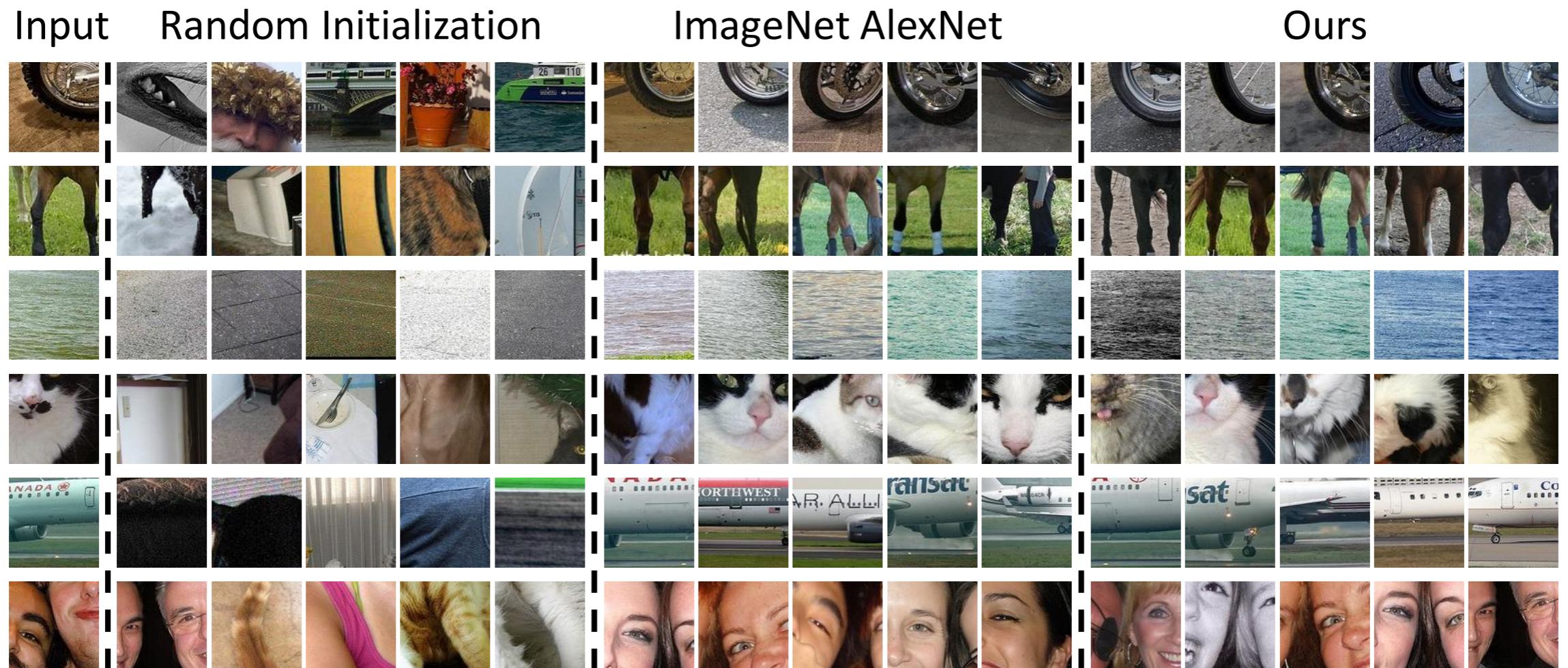
Patch Relative Configuration [Doerch et al.'15]

- Premise: A patch representation $\Phi(x)$ that does well in this task indirectly builds object priors.
- The criterion is not generative, but it retains enough information to generalize to other tasks



Patch Relative Configuration [Doerch et al.'15]

- Retrieval tasks:



- The representation captures visual similarity, leveraged in object detection, retrieval, etc.

Pixel Recurrent Networks [v.d.Oord et al'16]

- Prediction tasks of the form $\hat{x}_{t+1} = F(x_1, \dots, x_t)$ require a loss or an associated likelihood
 - e.g. $\|\hat{x}_{t+1} - x_{t+1}\|^2 \Leftrightarrow p(x_{t+1}|x_1, \dots, x_t) = \mathcal{N}(F(x_1, \dots, x_t), I)$
- In discrete domains we simply use a multinomial loss, in continuous domains there is no principled choice.
- How about images?

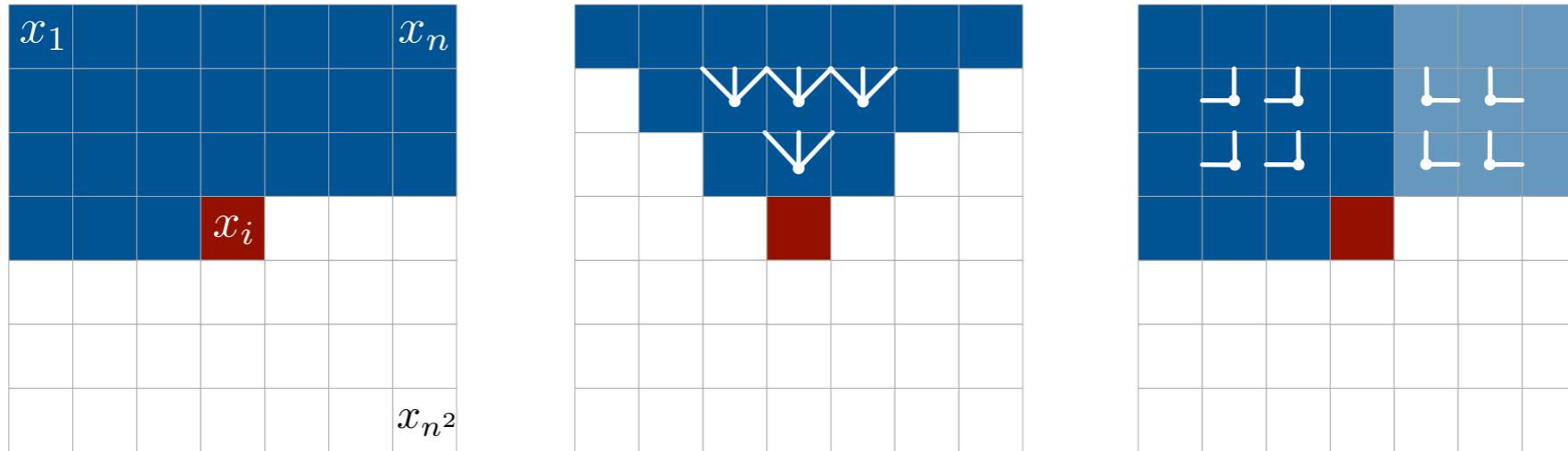
Pixel Recurrent Networks [v.d.Oord et al'16]

- Prediction tasks of the form $x_{t+1}^{\hat{}} = F(x_1^{\text{req}}, \dots, x_t^{\text{a}})$ loss or an associated likelihood
 - e.g. $\|\hat{x}_{t+1} - x_{t+1}\|^2 \Leftrightarrow p(x_{t+1}|x_1, \dots, x_t) = \mathcal{N}(F(x_1, \dots, x_t), I)$
- In discrete domains we simply use a multinomial loss, in continuous domains there is no principled choice.
- How about images?
 - We can treat them as discrete two-dimensional grids
 $x(u) \in \{0, 255\}$
 - Model each pixel from its “past” context:

$$p(x(u)|x(v); v \in \Omega(u)) = \text{softmax}(\Phi(x, \Omega(u)))$$

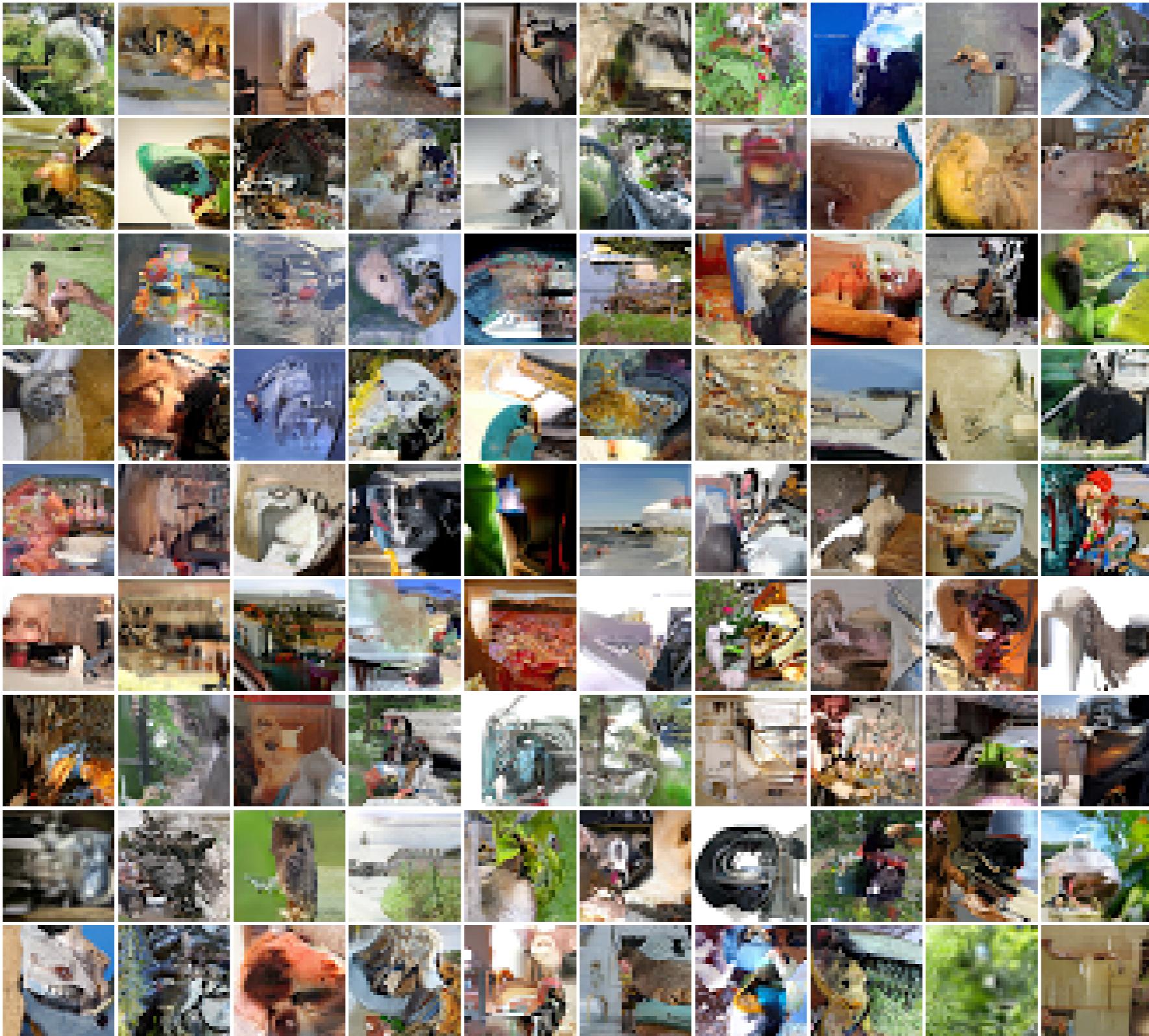
Pixel Recurrent Networks [v.d.Oord et al'16]

- Contexts are modeled using “diagonal BiLSTMS”.



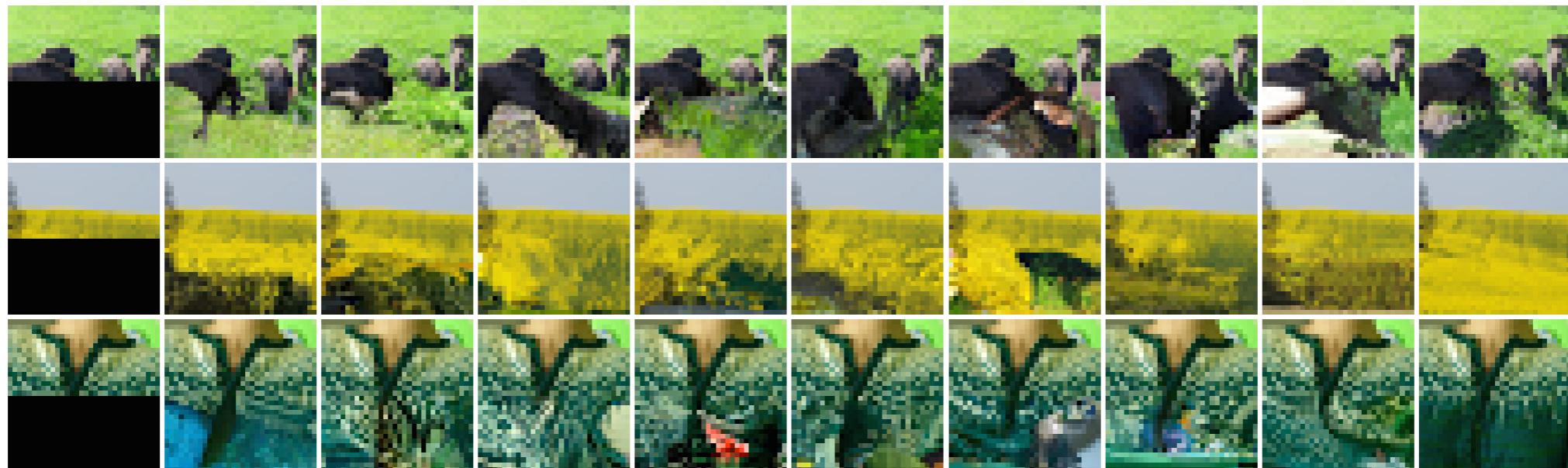
- Multi-Scale architecture conditions generations upon low-resolution samples (similarly as in LAPGANs).
- Very deep Recurrent Networks (> 10 layers).

- state-of-the-art image generation and modeling.



Pixel Recurrent Networks [v.d.Oord et al'16]

occluded



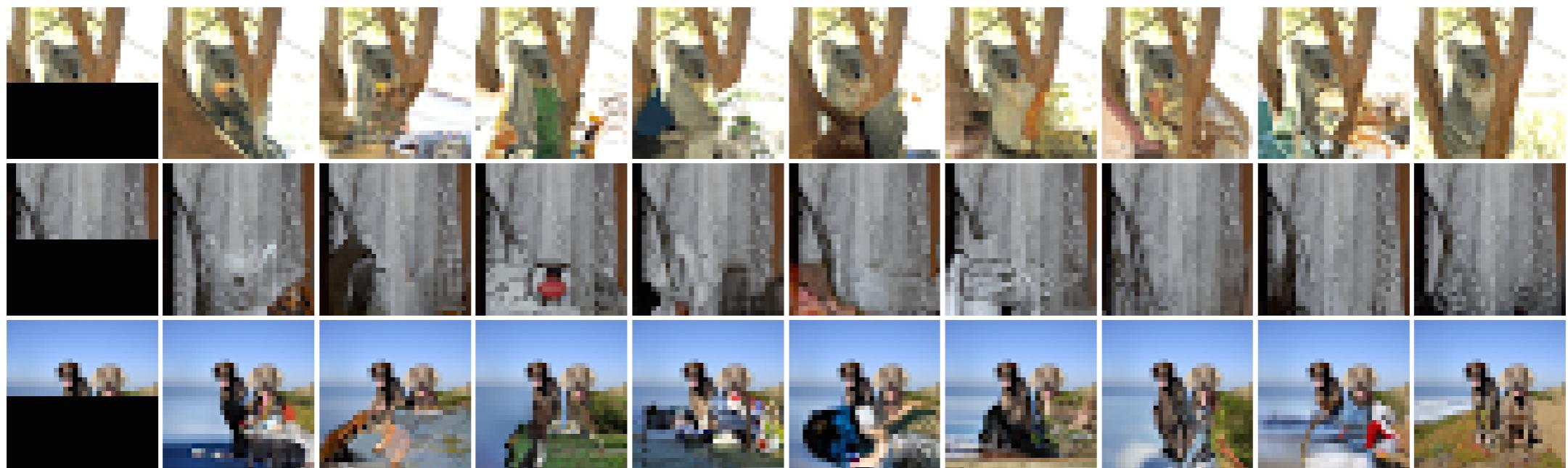
completions

original

occluded

completions

original



Pixel Recurrent Networks [v.d.Oord et al'16]

- MNIST and Cifar-10 log-likelihoods:

Model	NLL Test
DBM 2hl [1]:	≈ 84.62
DBN 2hl [2]:	≈ 84.55
NADE [3]:	88.33
EoNADE 2hl (128 orderings) [3]:	85.10
EoNADE-5 2hl (128 orderings) [4]:	84.68
DLGM [5]:	≈ 86.60
DLGM 8 leapfrog steps [6]:	≈ 85.51
DARN 1hl [7]:	≈ 84.13
MADE 2hl (32 masks) [8]:	86.64
DRAW [9]:	≤ 80.97
Diagonal BiLSTM (1 layer, $h = 32$):	80.75
Diagonal BiLSTM (7 layers, $h = 16$):	79.20

Table 4. Test set performance of different models on MNIST in *nats* (negative log-likelihood). Prior results taken from [1] (Salakhutdinov & Hinton, 2009), [2] (Murray & Salakhutdinov, 2009), [3] (Uria et al., 2014), [4] (Raiko et al., 2014), [5] (Rezende et al., 2014), [6] (Salimans et al., 2015), [7] (Gregor et al., 2014), [8] (Germain et al., 2015), [9] (Gregor et al., 2015).

Model	NLL Test (Train)
Uniform Distribution:	8.00
Multivariate Gaussian:	4.70
NICE [1]:	4.48
Deep Diffusion [2]:	4.20
Deep GMMs [3]:	4.00
RIDE [4]:	3.47
PixelCNN:	3.14 (3.08)
Row LSTM:	3.07 (3.00)
Diagonal BiLSTM:	3.00 (2.93)

Table 5. Test set performance of different models on CIFAR-10 in *bits/dim*. For our models we give training performance in brackets. [1] (Dinh et al., 2014), [2] (Sohl-Dickstein et al., 2015), [3] (van den Oord & Schrauwen, 2014a), [4] personal communication (Theis & Bethge, 2015).

- Recently, extension to video and speech (see deepmind.com for details).

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?

Open Challenges in Generative Modeling