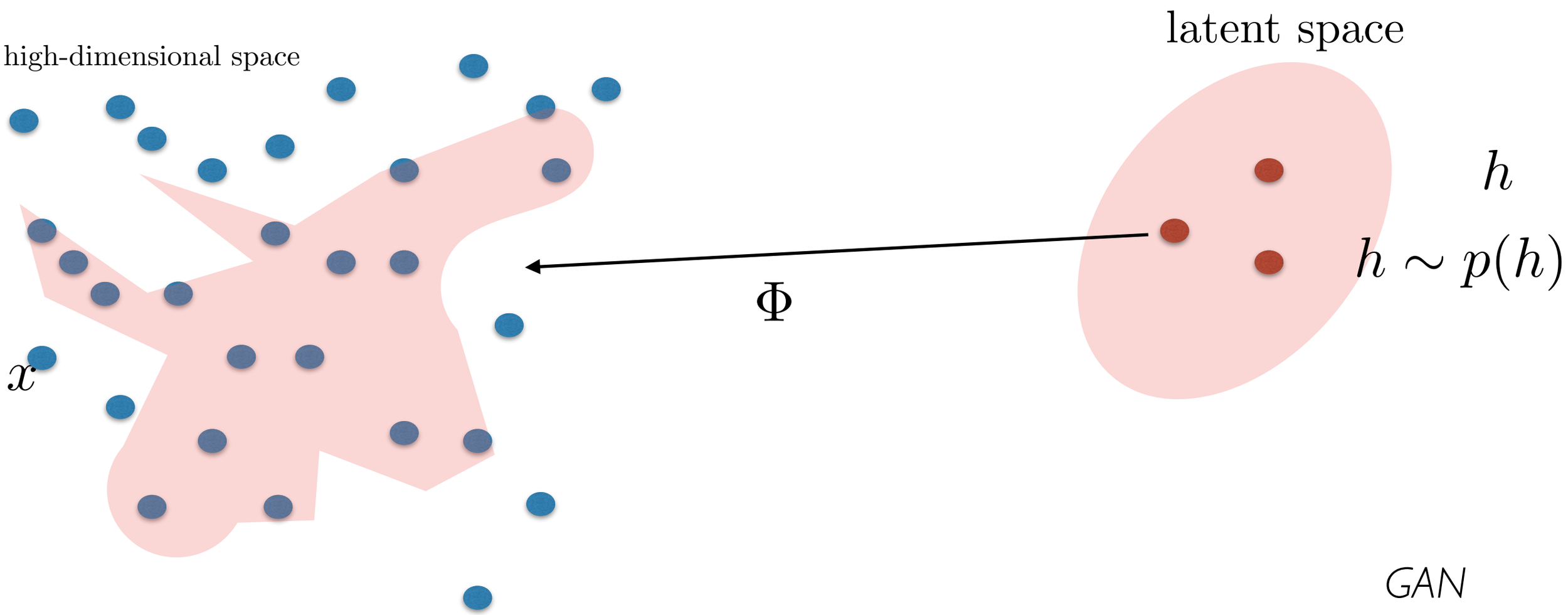
Inference and Representation
Lecture 12

Joan Bruna
Courant Institute, NYU

Generative Models of Complex data

- Flows or Transports of Measure



GAN
NormFlow
…

$p(x)$ defined implicitly with

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

Normalizing Flows

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

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

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

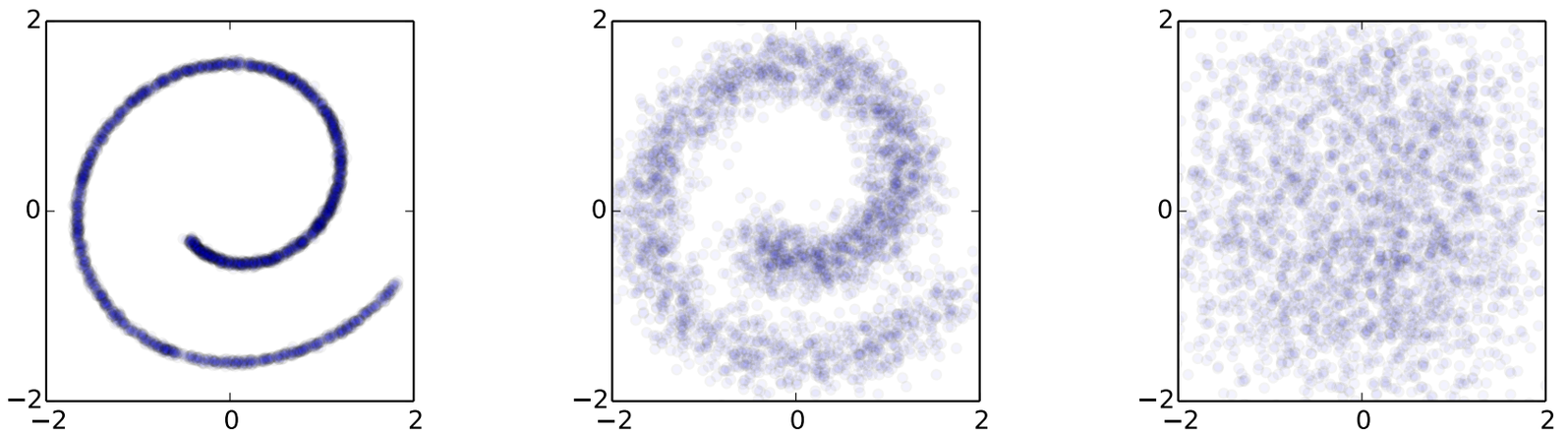
- One can parametrize invertible flows and use them within the variational inference to improve the variational approximation. [Rezende et al.'15]
- Also considered in ["NICE", Dinh et al'15].
- Special case: *Inverse Autoregressive Flows* (i.e. Jacobian triangular) explored in "Variational Inference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

Diffusion and Non-equilibrium Thermodynamics

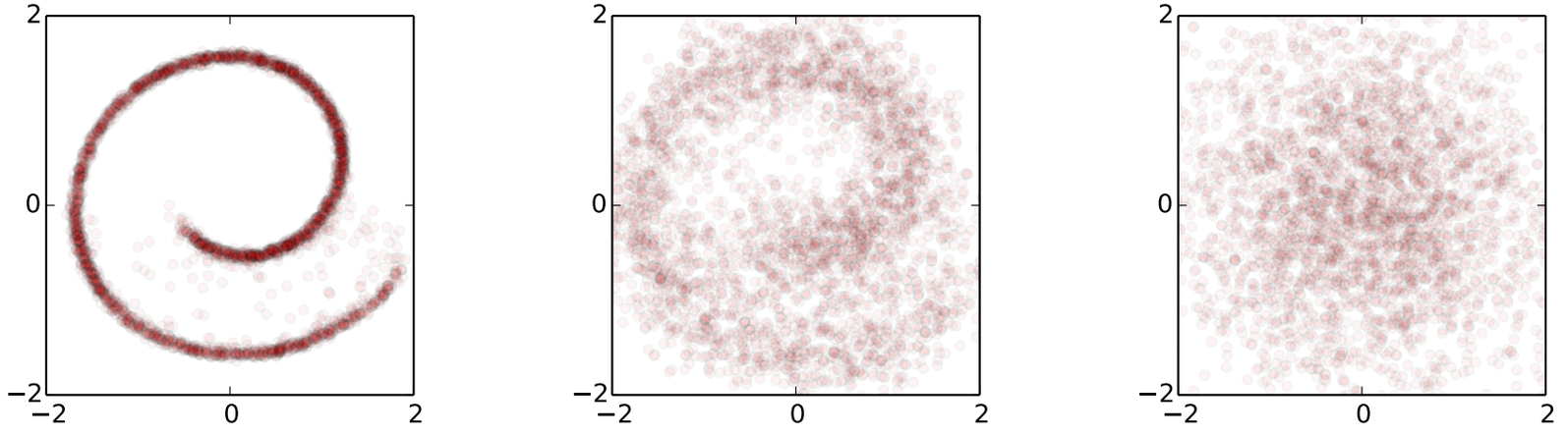
[Sohl-Dickstein et al.'15]

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

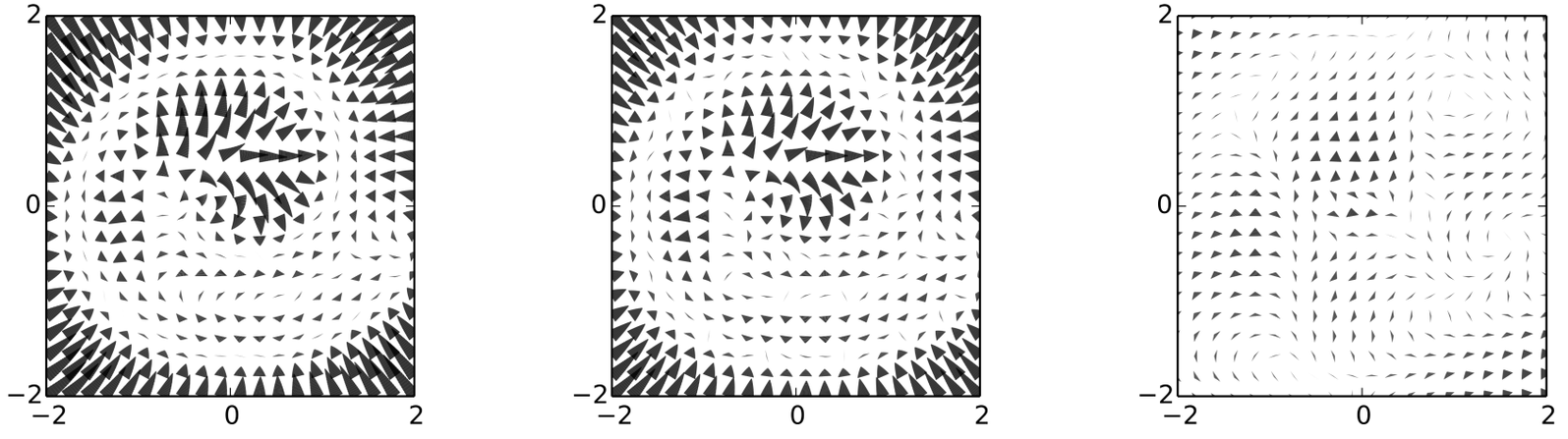
$q\left(\mathbf{x}^{(0\cdots T)}\right)$



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



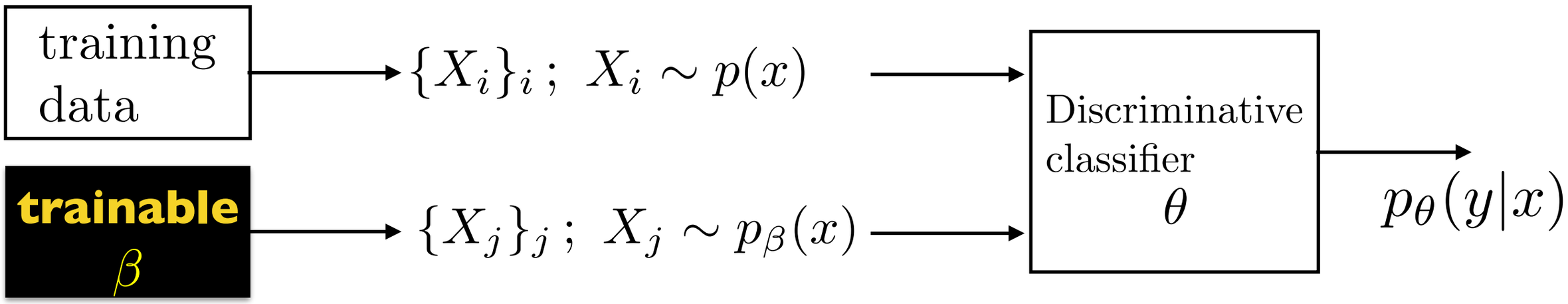
$\mathbf{f}_\mu\left(\mathbf{x}^{(t)}, t\right) - \mathbf{x}^{(t)}$



Generative Adversarial Networks

[Goodfellow et al., '14]

- Train generator and discriminator in a minimax setting:



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

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

Generative Adversarial Training

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

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

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

- Indeed,

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

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

- Numerical approach: alternate *k* steps of discriminator update with 1 step of generator update.
- Also, heuristic uses different false positive and false negative losses to improve numerical gradient computations.

LAPGAN

[Denton, Chintala et al.'15]

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

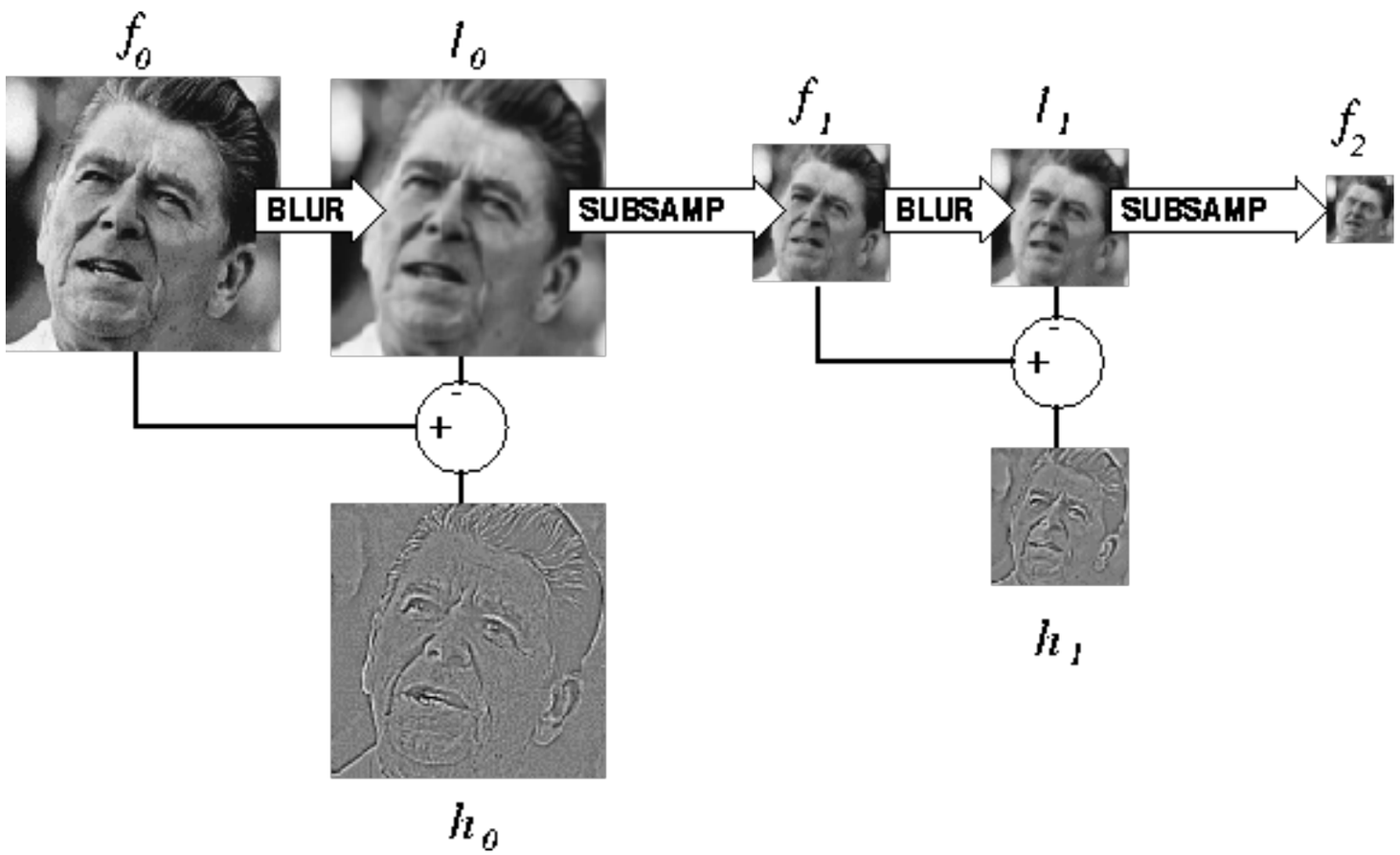
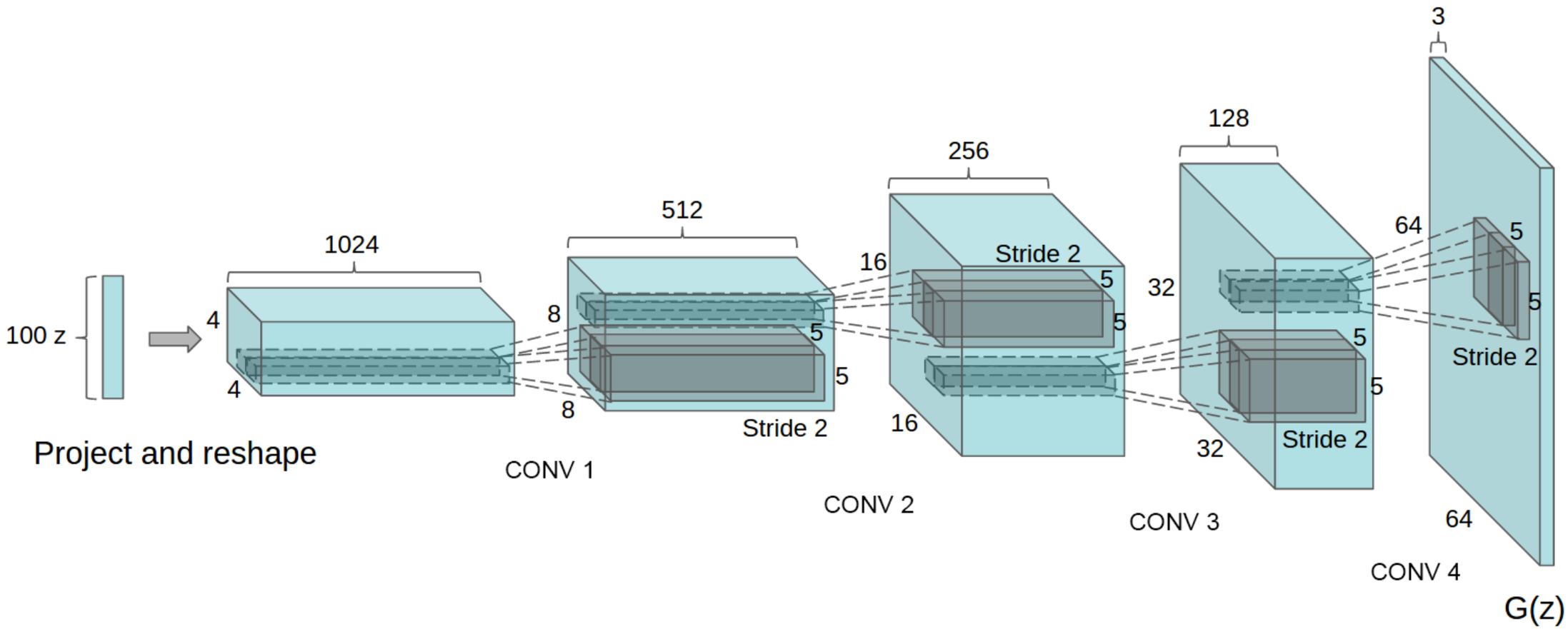


figure source: http://sepwww.stanford.edu

DC-GAN

[Radford et al.'16]

- Improved multi-scale architecture and Batch-Normalization:



GANs and Optimal Transport

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

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

- 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

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

- Estimated from finite data sample with

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

Wasserstein GAN [Arjovsky et al]

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

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

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

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* $\hat{p}_{r,L}$:

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

- Triangle Inequality:

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

Statistical Error — Modeling Error

Limitations of GAN Modeling

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

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

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

  – so we need a number of samples exponential in the dimension to make sampling error disappear.

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.

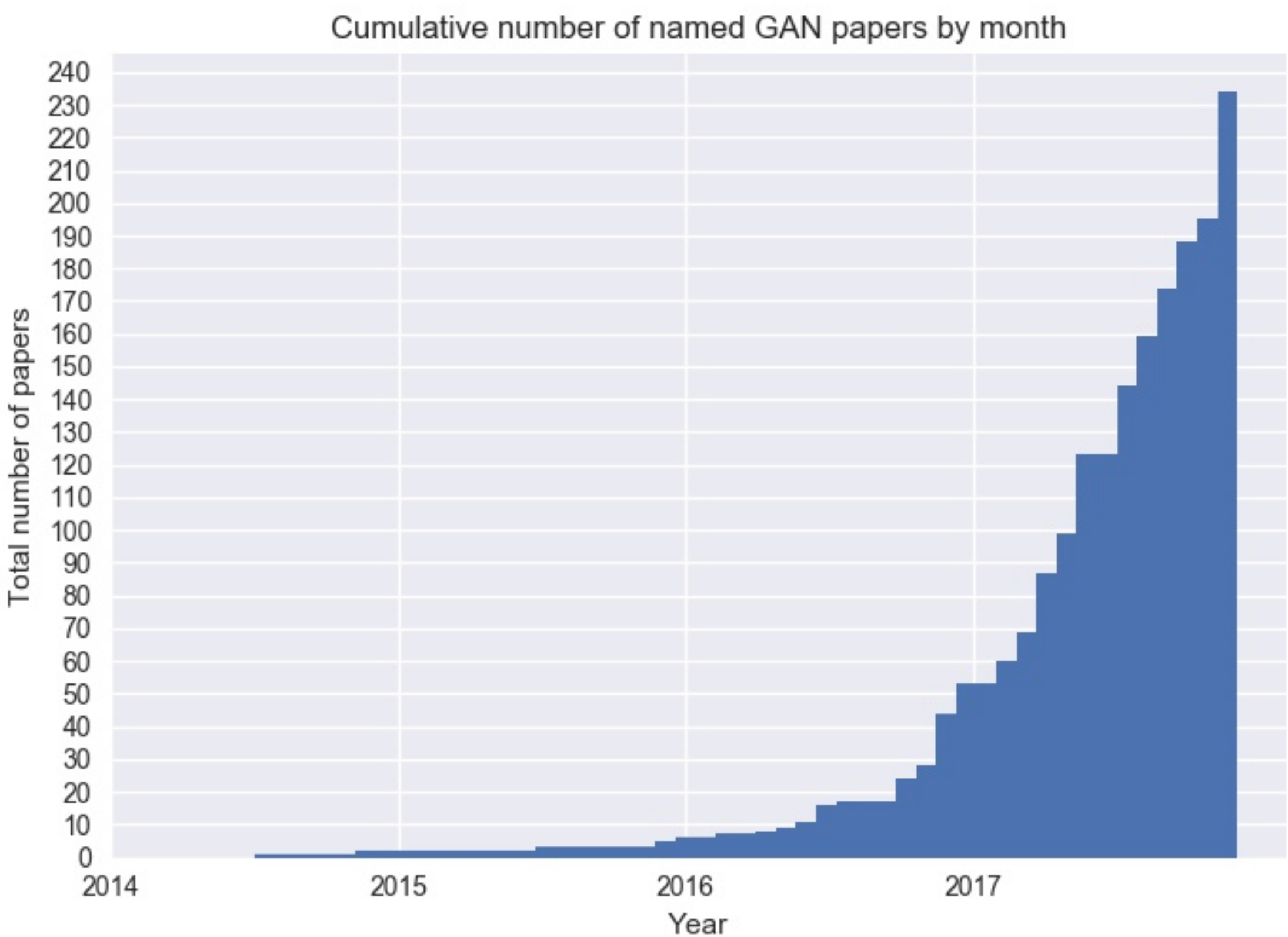
- In the case of Wasserstein distance, we have the curse of dimensionality: if input space is $\mathcal{X} = \mathbb{R}^d$, then

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

  – so we need a number of samples exponential in the dimension to make sampling error disappear.

- Intrinsic hardness of modeling densities in high dimensions.
- Entropic regularization: enforce p_m to be smooth by enforcing large $H(p_m)$ (computationally hard in general).

GANs are popular!



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

Some Recent Extensions

- *InfoGAN* [Chen & Rocky Duan et al., NIPS'16]

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

- CycleGAN [Zhu, Park, Isola, Efros]

- Progressive GAN [Karras et al]

InfoGAN [Chen, Duan et al, '16]

- Recall the notion of smoothness in generative models:
  - **Maximum Entropy**
- If ***x*** is a random vector of the form $z \sim F_0$, $x \mid z = G(z, \varphi)$, how to compute its entropy?

InfoGAN [Chen, Duan et al, '16]

- Recall the notion of smoothness in generative models:
  - **Maximum Entropy**
- If $\boldsymbol{x}$ is a random vector of the form $z \sim F_0$, $x \mid z = G(z, \varphi)$, how to compute its entropy?
- If $z \mapsto G(z, \varphi)$ is injective, then

$$p_G(x) = F_0(G^{-1}(x)) \cdot |DG^{-1}(x)| \ , \text{ and}$$

$$\mathbb{E}_{z \sim F_0} f(G(z)) = \mathbb{E}_{x \sim p_G} f(x) \quad \forall \ f \text{ measurable} \ .$$

InfoGAN [Chen, Duan et al, '16]

- Recall the notion of smoothness in generative models:
  - **Maximum Entropy**
- If $\boldsymbol{x}$ is a random vector of the form $z \sim F_0$, $x \mid z = G(z, \varphi)$, how to compute its entropy?
- If $z \mapsto G(z, \varphi)$ is injective, then

$$p_G(x) = F_0(G^{-1}(x)) \cdot |DG^{-1}(x)| \ , \text{ and}$$
$$\mathbb{E}_{z \sim F_0} f(G(z)) = \mathbb{E}_{x \sim p_G} f(x) \ \ \forall \ f \text{ measurable} \ .$$

- Since $H(p) = -\mathbb{E}_{x \sim p}[\log p(x)]$, we have

$$\begin{aligned} H(p_G) &= -\mathbb{E}_{x \sim p_G} \log p_G(x) \\ &= -\mathbb{E}_{x \sim p_G}[\log F_0 G^{-1} x + \log |DG^{-1}(x)|] \\ &= -\mathbb{E}_{z \sim F_0}[\log F_0(z) - \log |DG(z)|] \\ &= H(F_0) + \mathbb{E}_{z \sim F_0} \log |DG(z)| \ . \end{aligned}$$

InfoGAN [Chen, Duan et al, '16]

- Recall the notion of smoothness in generative models:
  - **Maximum Entropy**
- If $\boldsymbol{x}$ is a random vector of the form $z \sim F_0$, $x \mid z = G(z, \varphi)$, how to compute its entropy?
- If $z \mapsto G(z, \varphi)$ is injective, then

$$p_G(x) = F_0(G^{-1}(x)) \cdot |DG^{-1}(x)| \ , \text{ and}$$
$$\mathbb{E}_{z \sim F_0} f(G(z)) = \mathbb{E}_{x \sim p_G} f(x) \ \ \forall \ f \text{ measurable} \ .$$

- Since $H(p) = -\mathbb{E}_{x \sim p}[\log p(x)]$, we have

$$\begin{aligned} H(p_G) &= -\mathbb{E}_{x \sim p_G} \log p_G(x) \\ &= -\mathbb{E}_{x \sim p_G}[\log F_0 G^{-1} x + \log |DG^{-1}(x)|] \\ &= -\mathbb{E}_{z \sim F_0}[\log F_0(z) - \log |DG(z)|] \\ &= H(F_0) + \mathbb{E}_{z \sim F_0} \log |DG(z)| \ . \end{aligned}$$

- Challenge: computing $\mathbb{E}_{z \sim F_0} \log |DG(z)|$ is hard!

InfoGAN

- In some problems, we have prior information on the latent structure of variability, e.g.
  - Digits have a latent discrete (10 class) structure, plus continuous low-dim structure encoding style.
  - Faces have illumination and pose, identity, muscular movement, etc.

InfoGAN

- In some problems, we have prior information on the latent structure of variability, e.g.
  - Digits have a latent discrete (10 class) structure, plus continuous low-dim structure encoding style.
  - Faces have illumination and pose, identity, muscular movement, etc.
- Q: Can we combine max-entropy regularization with more control on the latent structure?

InfoGAN

- In some problems, we have prior information on the latent structure of variability, e.g.
  - Digits have a latent discrete (10 class) structure, plus continuous low-dim structure encoding style.
  - Faces have illumination and pose, identity, muscular movement, etc.
- Q: Can we combine max-entropy regularization with more control on the latent structure?
- An alternative is to use mutual information:

$$I(X, Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X) \ .$$

  - i.e. how much uncertainty remains in X after observing Y (and vice versa).
- It can be plugged-in the GAN objective as follows

$$\min_G \max_D \{\mathbb{E}_{x \sim \hat{P}}[\log D(x)] + \mathbb{E}_{z \sim F_0}[\log(1 - D(G(z))]\} - \lambda I(c; G(z, c)) \ ,$$

$$c \subset z$$

InfoGAN

- Computing the mutual information requires access to the posterior $p(c \mid x)$
- Although in the GAN we don't have this term, we have something that can approximate it: the discriminator network!

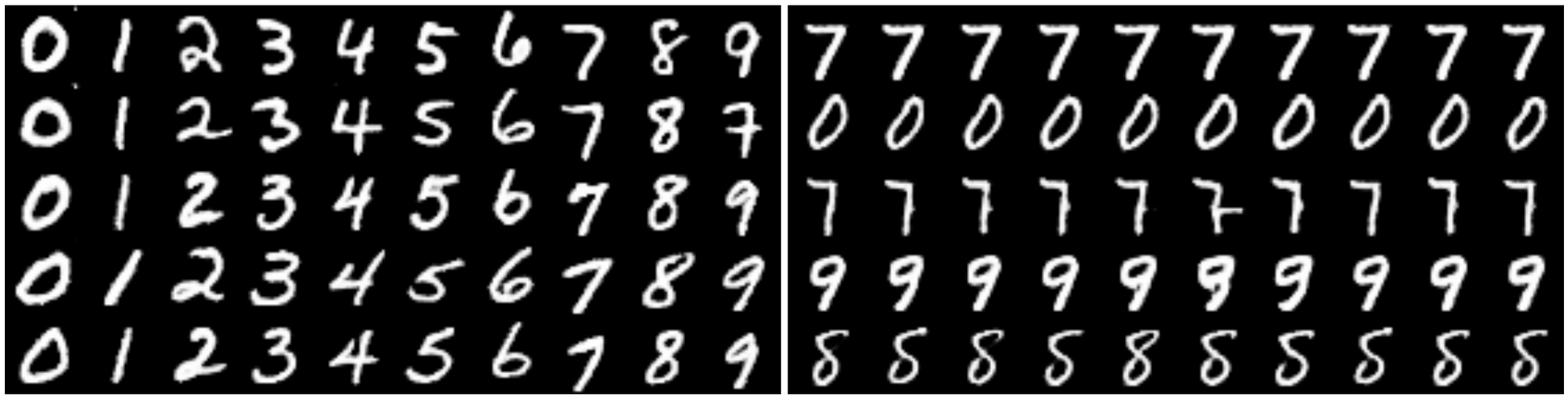
InfoGAN

- Computing the mutual information requires access to the posterior $p(c \mid x)$
- Although in the GAN we don't have this term, we have something that can approximate it: the discriminator network!
- Variational Information Maximization:

$$
\begin{aligned}
I(c, G(z, c)) &= \mathbb{E}_{x \sim G(z,c)} \mathbb{E}_{c' \sim p(c|x)} [\log p(c'|x)] \; + \; H(c) \\
&= \mathbb{E}_{x \sim G(z,c)} \left(D_{KL}(p(c'|x) || q(c'|x)) + \mathbb{E}_{c' \sim p(c|x)} \log q(c'|x) \right) + H(c) \\
&\geq \mathbb{E}_{x \sim G(z,c)} \mathbb{E}_{c' \sim p(c|x)} \log q(c'|x) + H(c) \\
&= \mathbb{E}_{(z,c) \sim P(c,z)} \log q(c'|G(z, c)) + H(c) \; .
\end{aligned}
$$

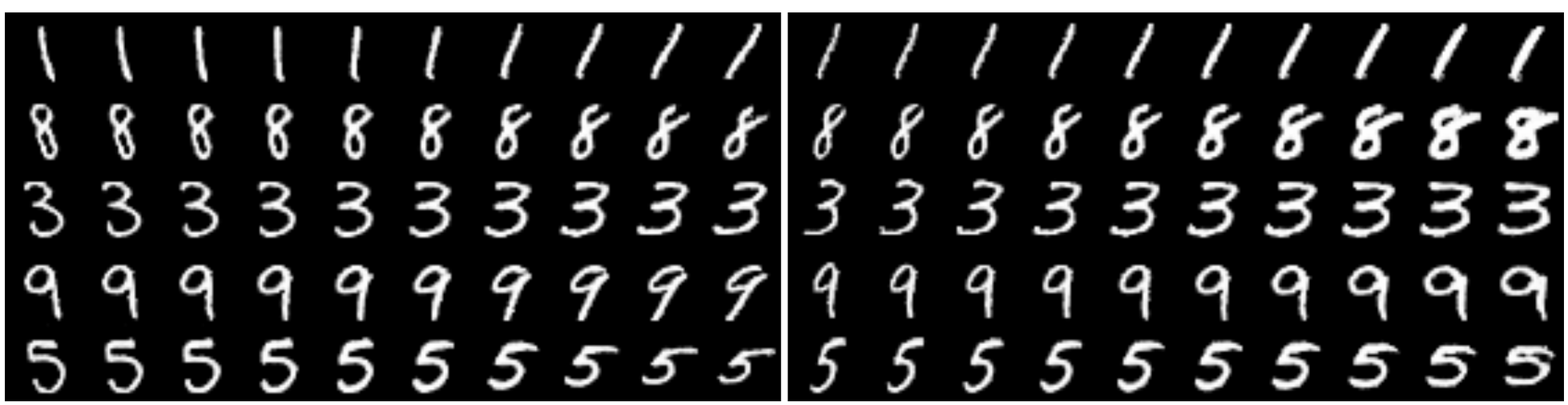
InfoGAN

- Unsupervised "distentanglement" in low-dimensional data.
- It also alleviates the tendency of GAN to overfit the training data.
- WARNING: singular distributions have no well-defined entropy!



(a) Varying c_1 on InfoGAN (Digit type)

(b) Varying c_1 on regular GAN (No clear meaning)



(c) Varying c_2 from -2 to 2 on InfoGAN (Rotation)

(d) Varying c_3 from -2 to 2 on InfoGAN (Width)

InfoGAN

- Unsupervised "distentanglement" in low-dimensional data.
- It also alleviates the tendency of GAN to overfit the training data.
- WARNING: singular distributions have no well-defined entropy!

(a) Rotation (b) Width

Conditional GANs

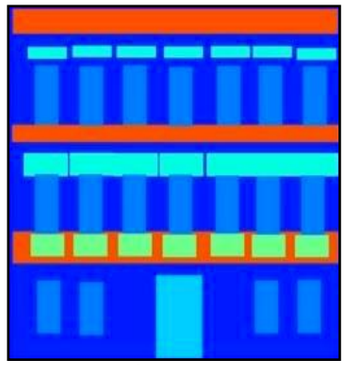
- Conditional GANS for computer vision tasks

Labels to Street Scene

input

output

Labels to Facade



input

output

BW to Color

input

output

Aerial to Map

input

output

Day to Night

input

output

Edges to Photo

input

output

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

Cycle GAN [Zhu, Park, Isola, Efros]

- Image-to-Image translation without aligned datasets.

Monet ⇄ Photos

Zebras ⇄ Horses

Monet → photo

zebra → horse

photo → Monet

Photograph

Monet

Van Gogh

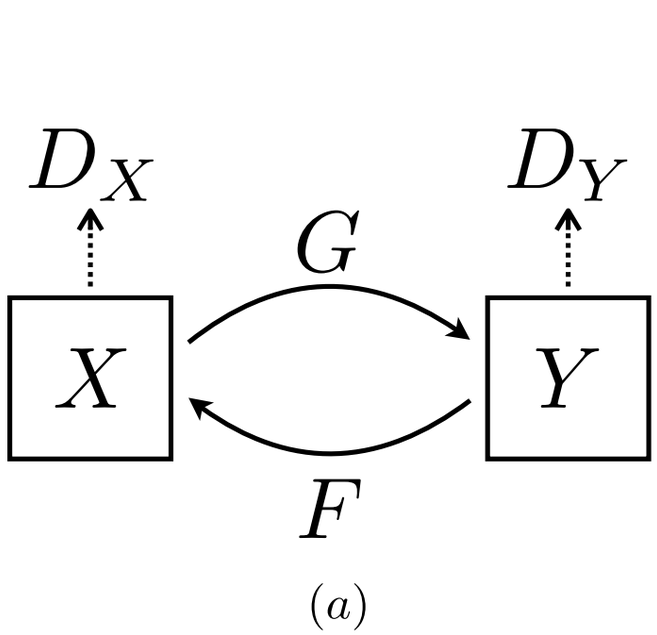
horse → zebra

Cezanne

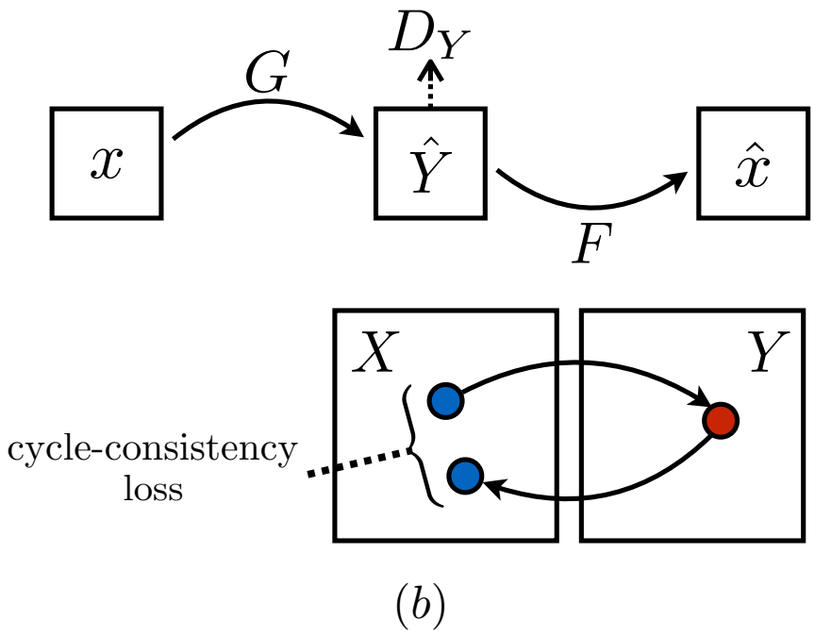
Ukiyo-e

Cycle GAN

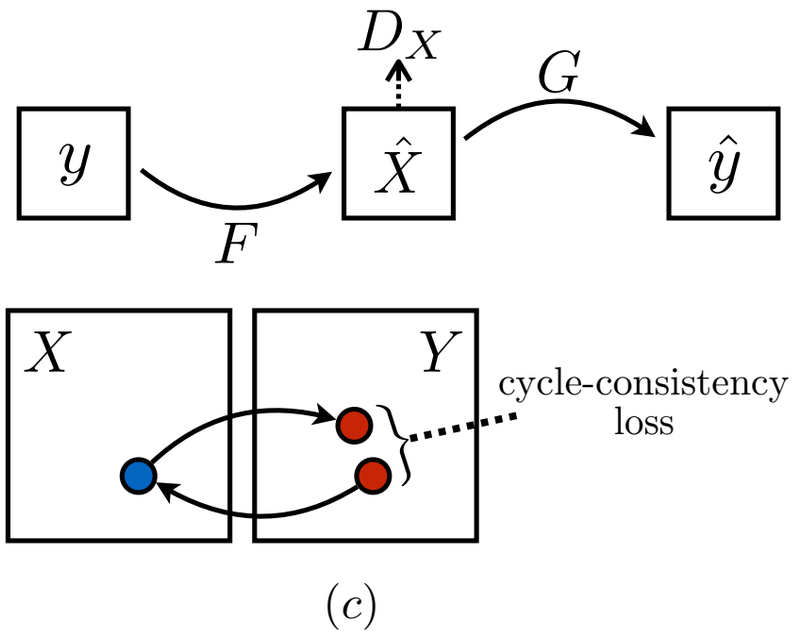
- Cycle Consistency Loss:



(a)



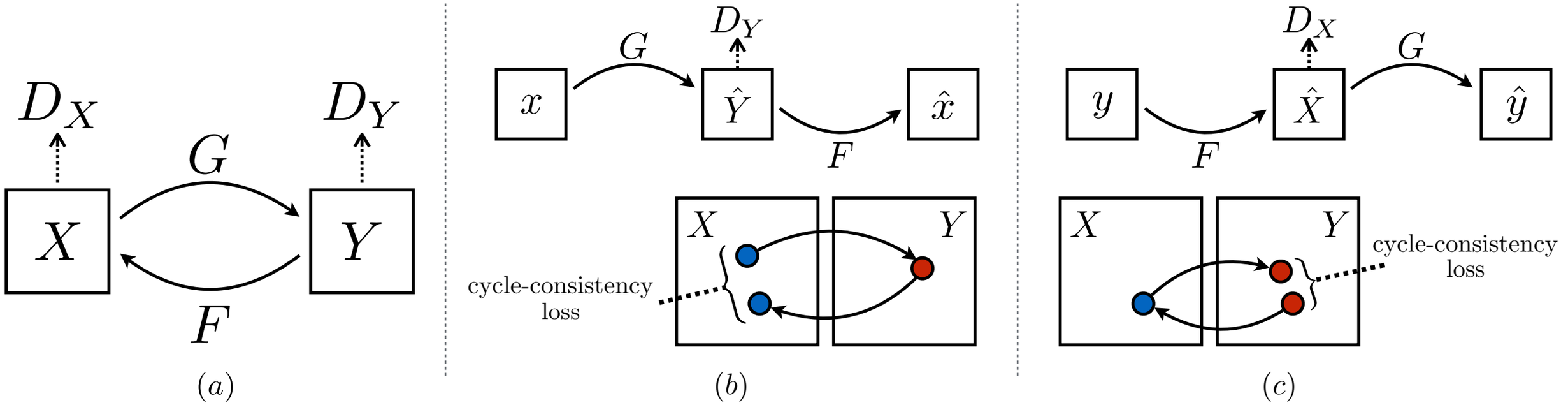
(b)



(c)

Cycle GAN

- Cycle Consistency Loss:



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

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

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

- Also includes a "feature-matching" term that stabilizes training.

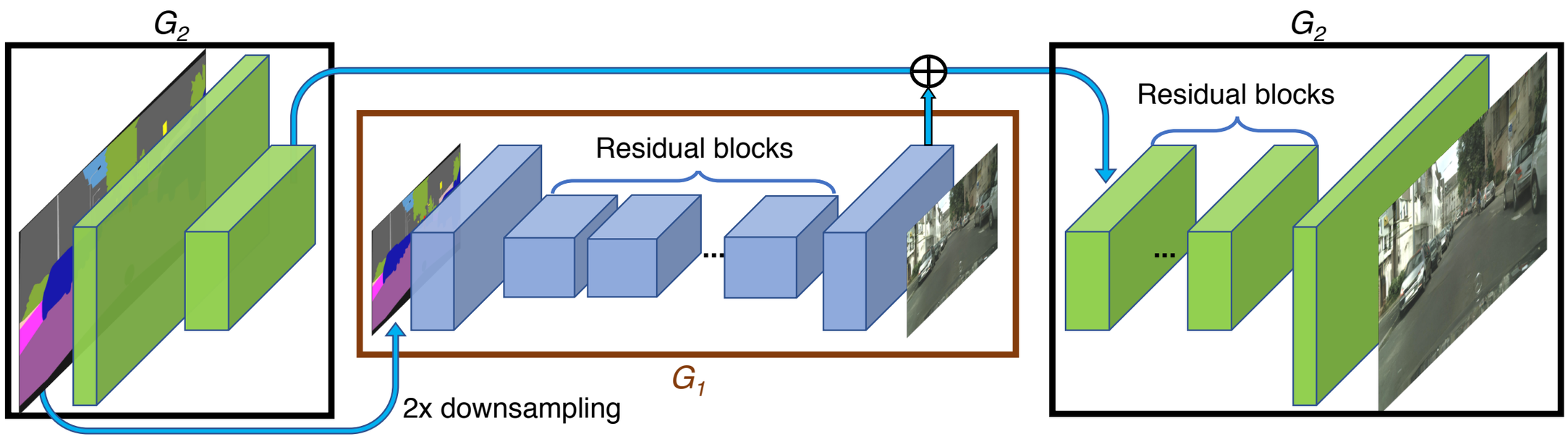
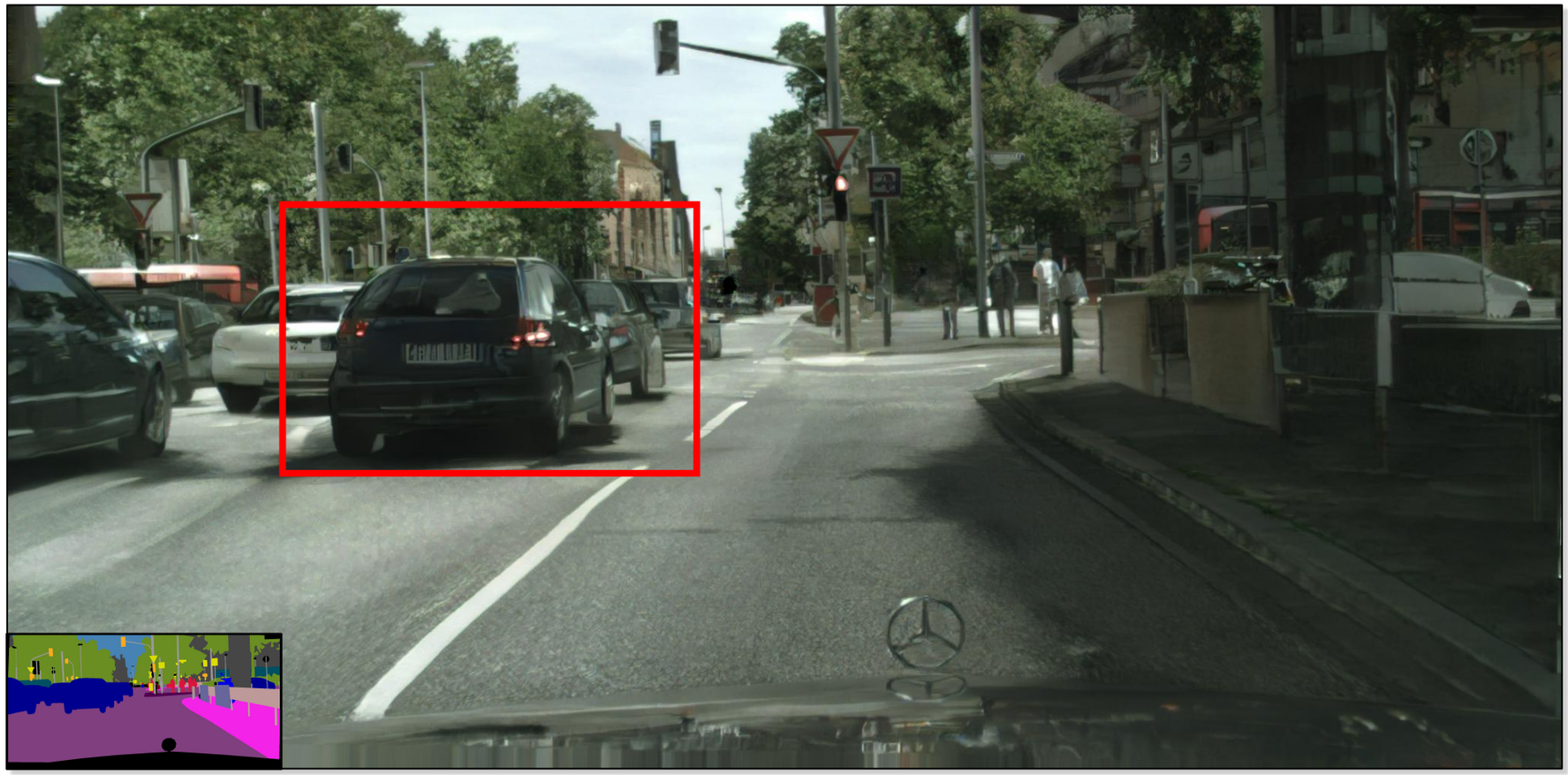


Photo Realistic GAN [Wang et al]



(a) Synthesized result

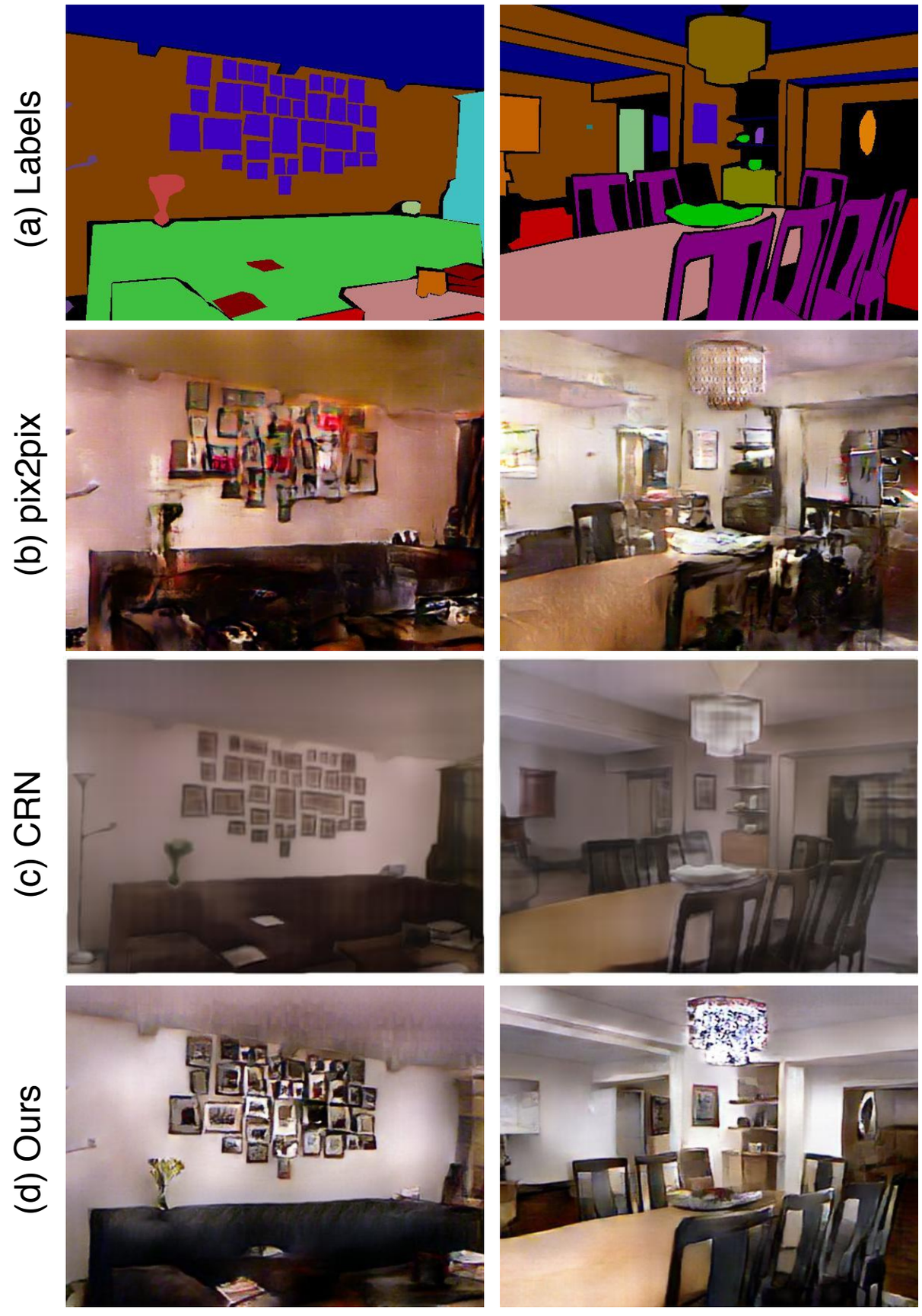
Cascaded refinement network [5]

Our result

(b) Application: Change label types

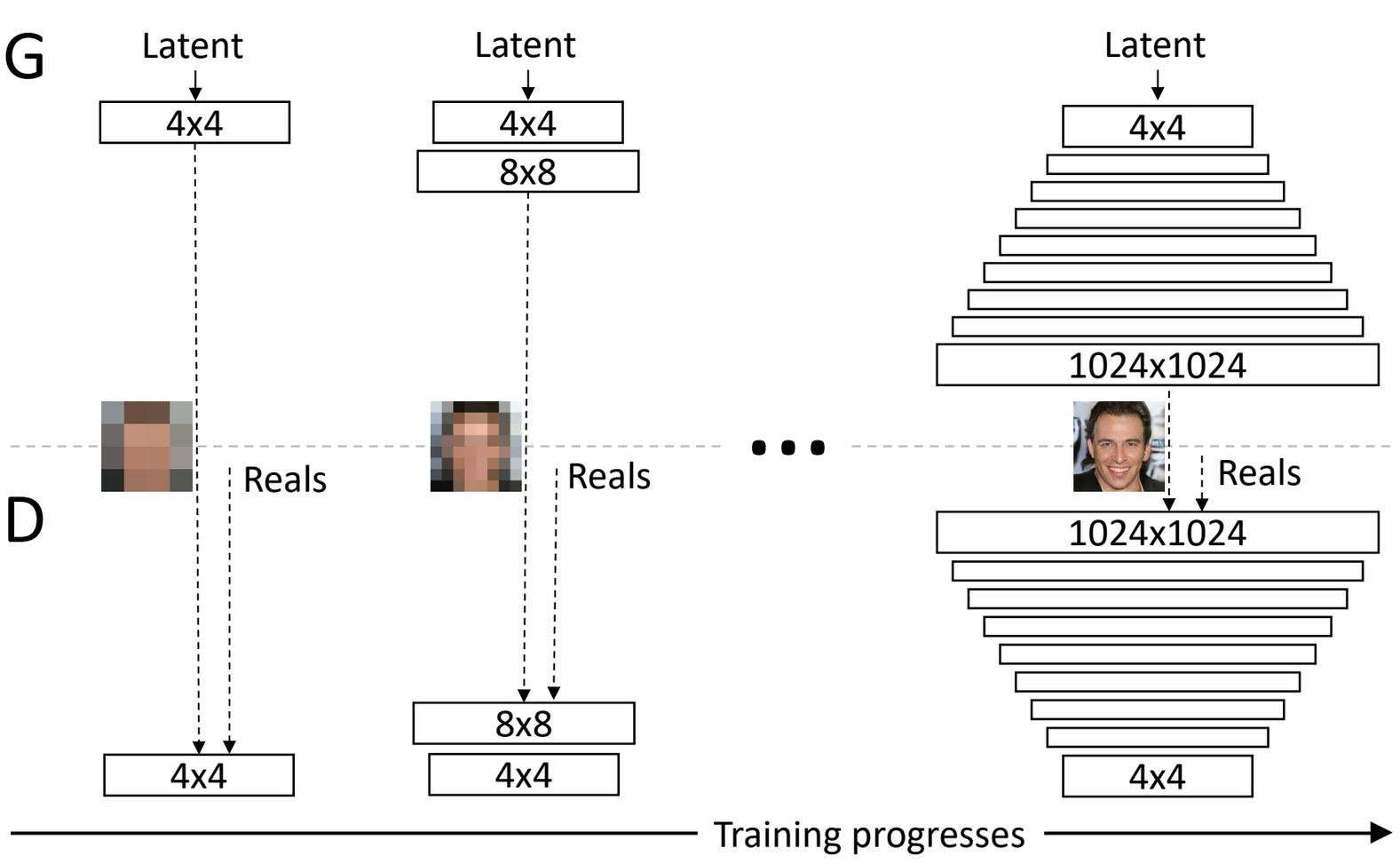
(c) Application: Edit object appearance

Photo-Realistic GAN



Progressive GANs [Karras et al]

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



Progressive GAN [Karras et al]

Progressive GANs

POTTEDPLANT HORSE SOFA BUS CHURCHOUTDOOR BICYCLE TVMONITOR

Limits of Transportation Models

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

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

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

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

Autoregressive Models

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

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

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

Autoregressive Models

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

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

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

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

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

Autoregressive Models

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

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

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

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

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

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

Autoregressive Models

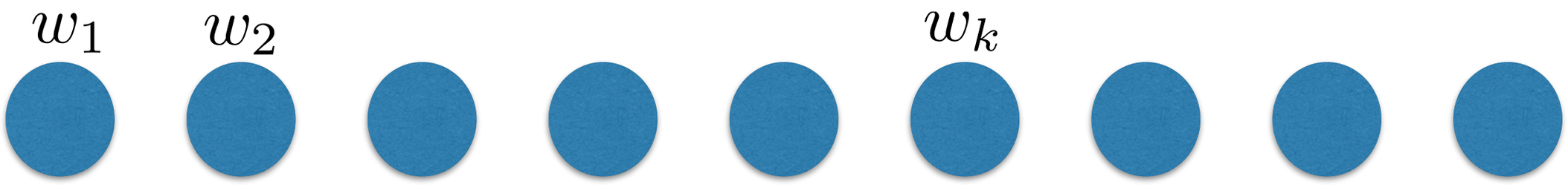
- Time Series
  - Speech, Music
  - Video
  - Language
  - Other time series (Weather, Finance, …)
- Spatially ordered data, Multi-Resolution data
  - Images

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

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

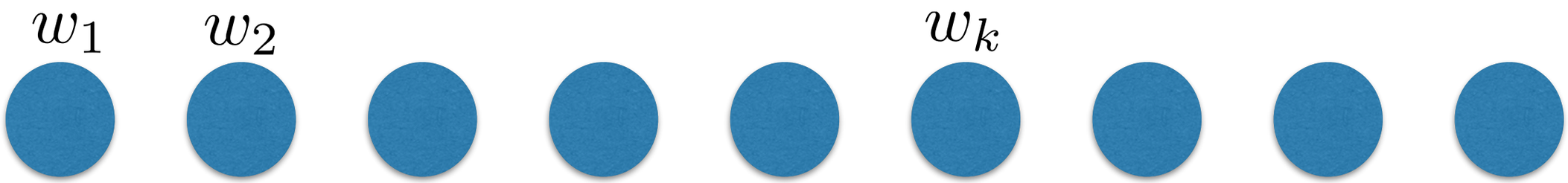
Word2vec [Mikolov et al.'13].

- Unsupervised learning "success story".



Word2vec [Mikolov et al.'13].

- Unsupervised learning "success story".

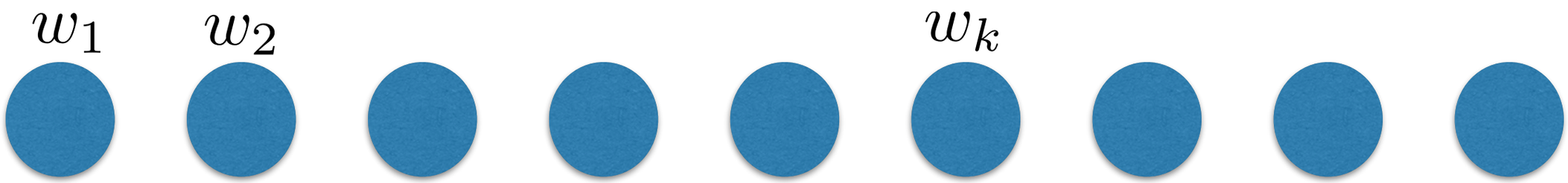


- Language creates a notion of similarity between words:

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

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

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

$$p(D = 1|c, w) = \sigma(\langle v_w, v_c \rangle) \ , \ v_w, v_c \in \mathbb{R}^d$$

.

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

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

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

- Training with Maximum Likelihood: $\sigma(x) = \dfrac{1}{1 + e^{-x}}$

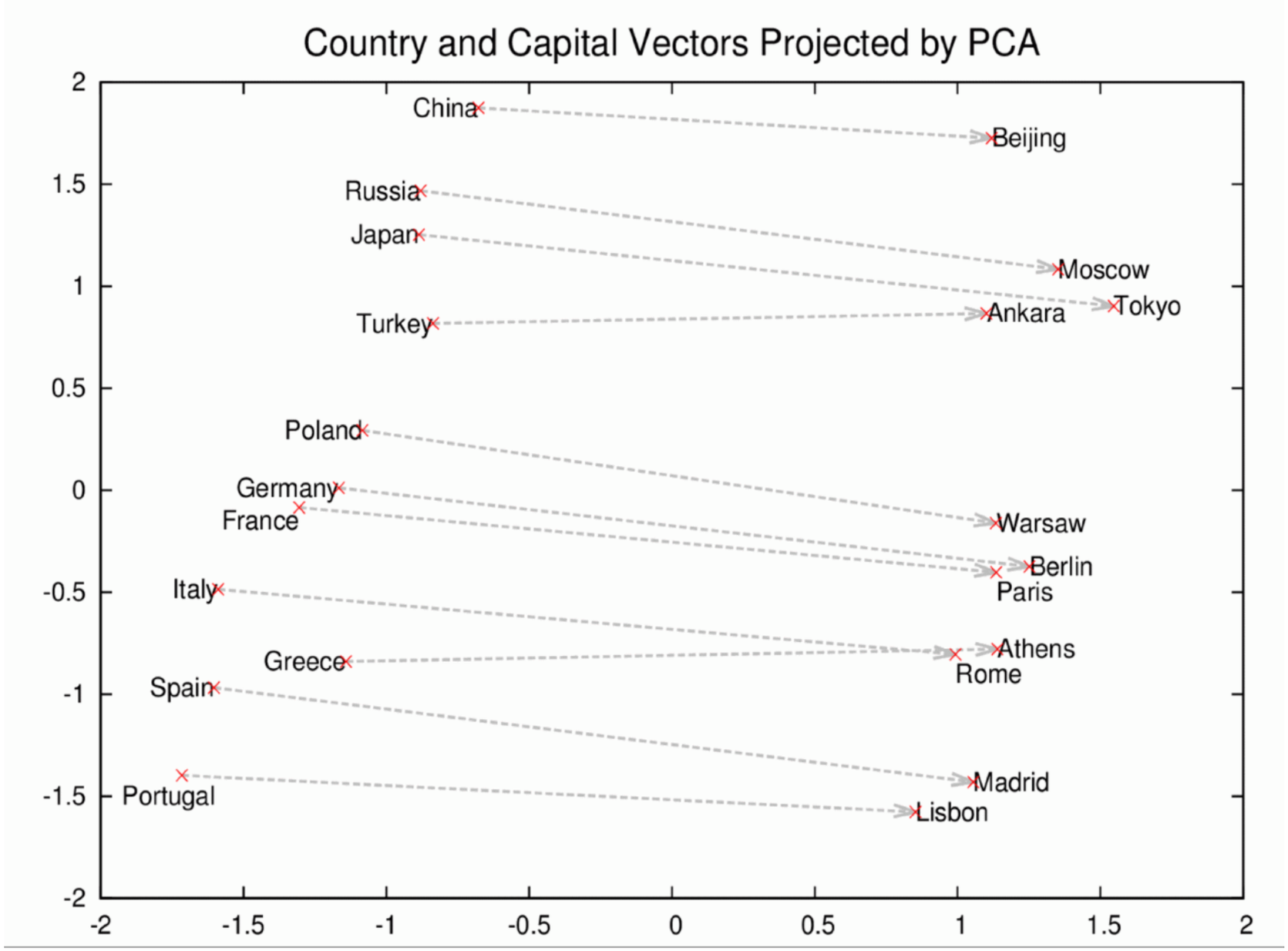
$$\arg\max_\theta \prod_{(w,c)\sim\mathcal{D}} p(D = 1|c, w, \theta) \prod_{(w,c)\sim\mathcal{D}'} p(D = 0|c, w, \theta)$$

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

$\mathcal{D}$: positive contexts $\quad$ $\mathcal{D}'$: 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.



[google]

Video Prediction

- Rather than modeling the density of natural images

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

we may be also interested in modeling the conditional distributions

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

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

Video Prediction

- Rather than modeling the density of natural images

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

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

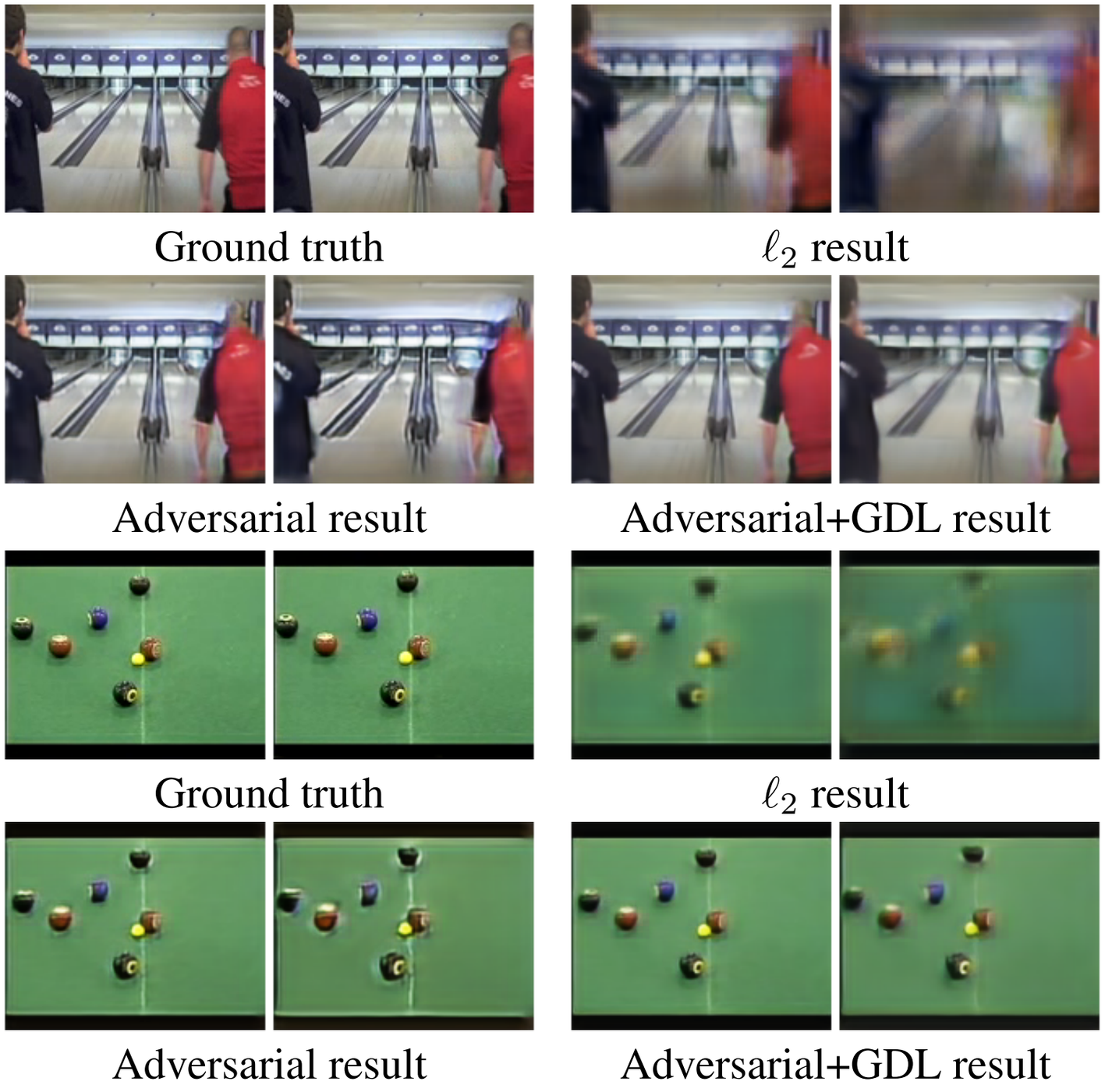
- Similarly, can we find a signal representation $\Phi(x_t)$ that is consistent with the "video language" metric? i.e.

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

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

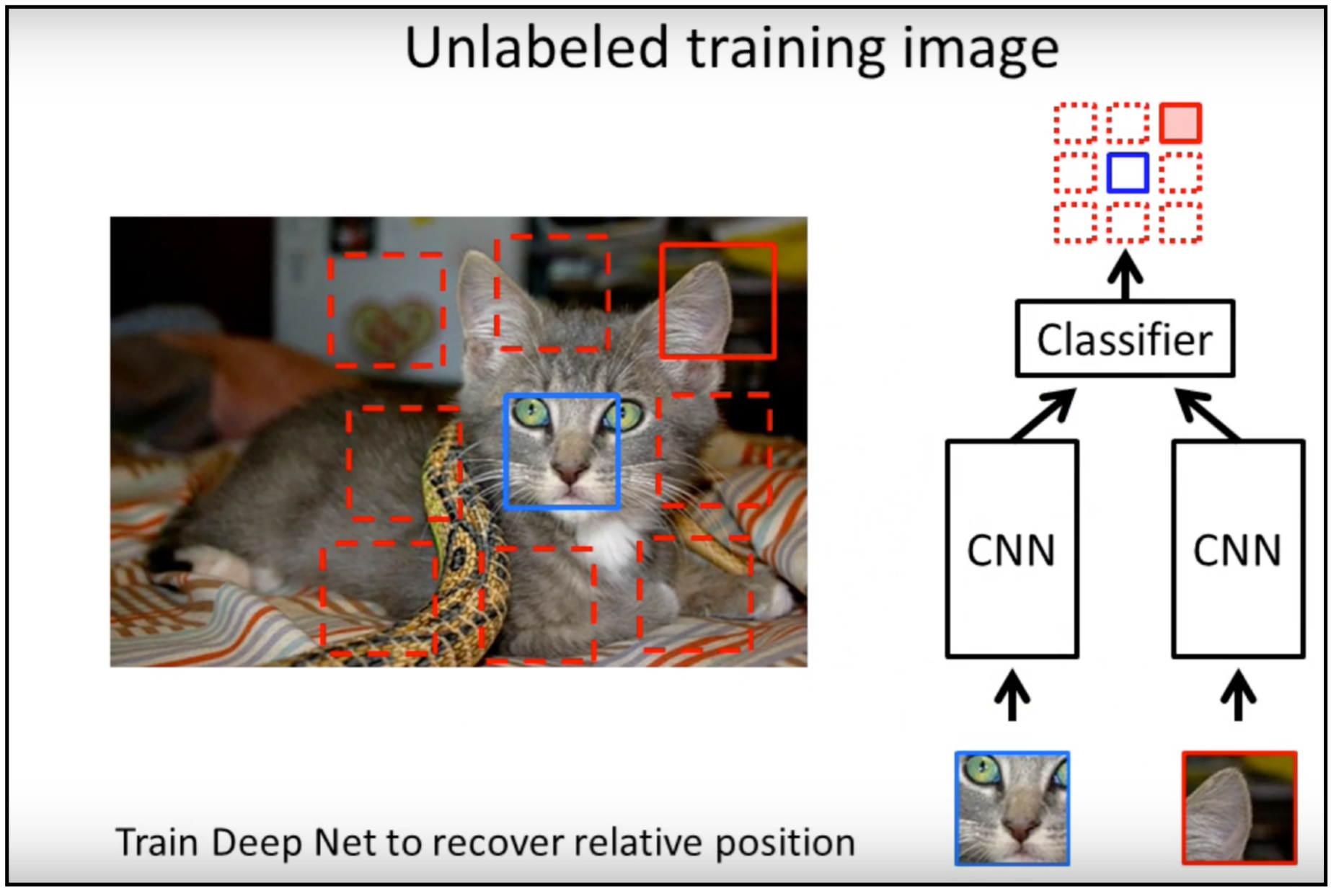
Video Prediction

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



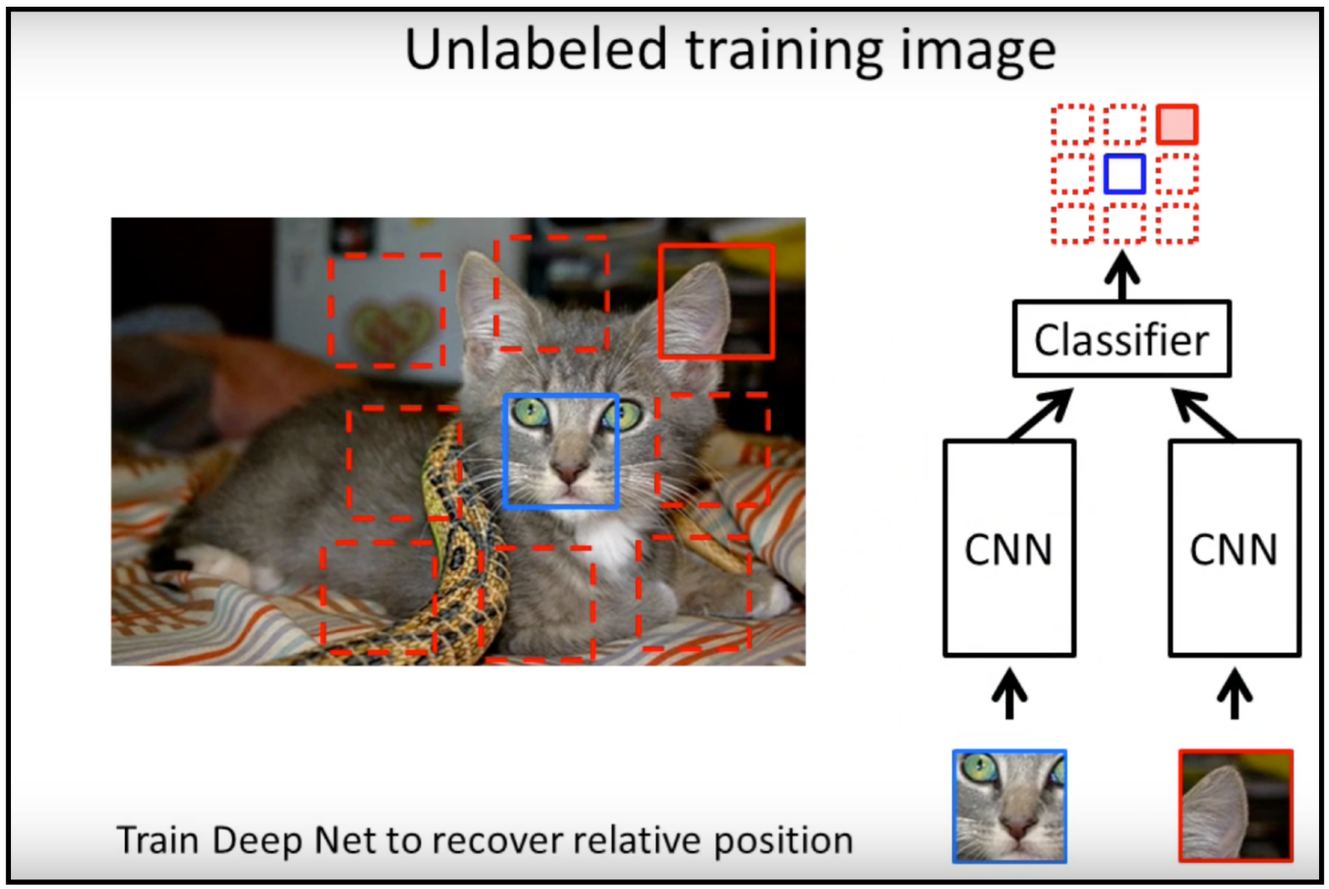
Patch Relative Configuration [Doerch et al.'15]

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



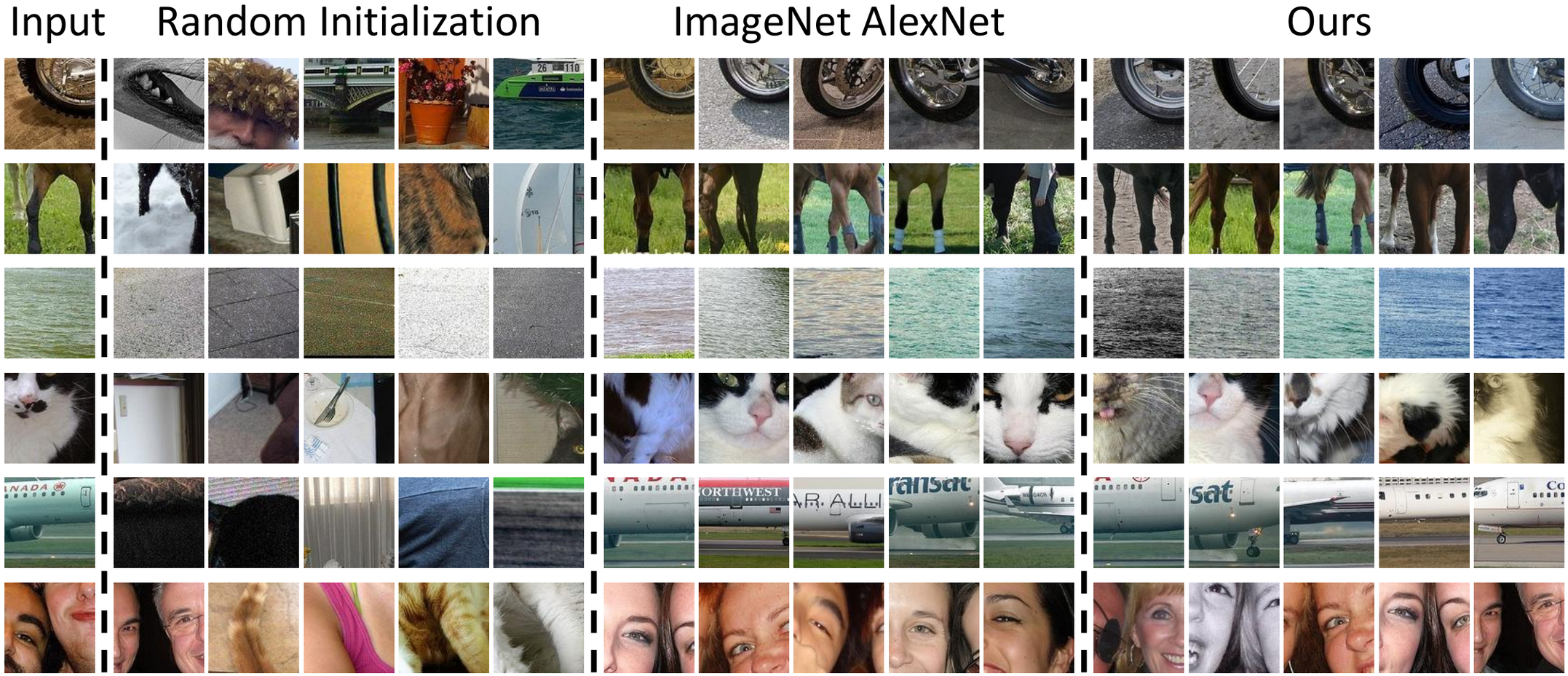
Patch Relative Configuration [Doerch et al.'15]

- 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 $\hat{x_{t+1}} = F(x_1, \ldots, x_t)$ require a loss or an associated likelihood

$$e.g.\ \|\hat{x}_{t+1} - x_{t+1}\|^2 \ \Leftrightarrow\ p(x_{t+1}|x_1, \ldots, x_t) = \mathcal{N}(F(x_1, \ldots, x_t), I)$$

- 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 $\hat{x}_{t+1} = F(x_1, \ldots, x_t)$ require a loss or an associated likelihood

$$e.g.\ \|\hat{x}_{t+1} - x_{t+1}\|^2 \ \Leftrightarrow\ p(x_{t+1}|x_1, \ldots, x_t) = \mathcal{N}(F(x_1, \ldots, x_t), I)$$

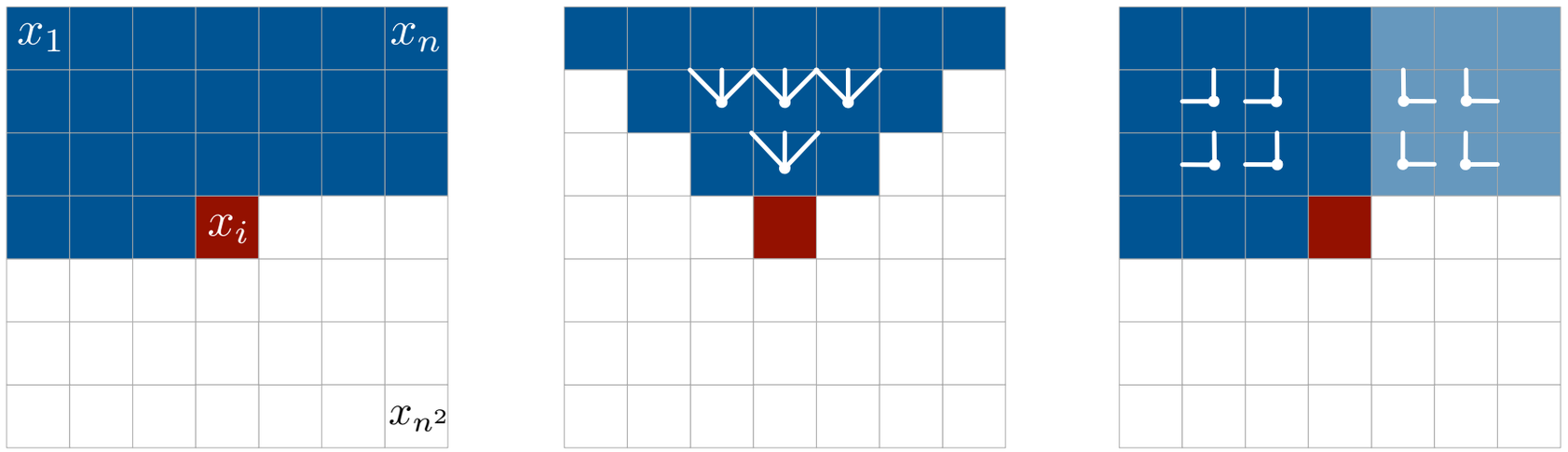
- In discrete domains we simply use a multinomial loss, in continuous domains there is no principled choice.

- How about images?
  - We can treat them as discrete two-dimensional grids $x(u) \in \{0, 255\}$
  - Model each pixel from its "past" context:

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

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

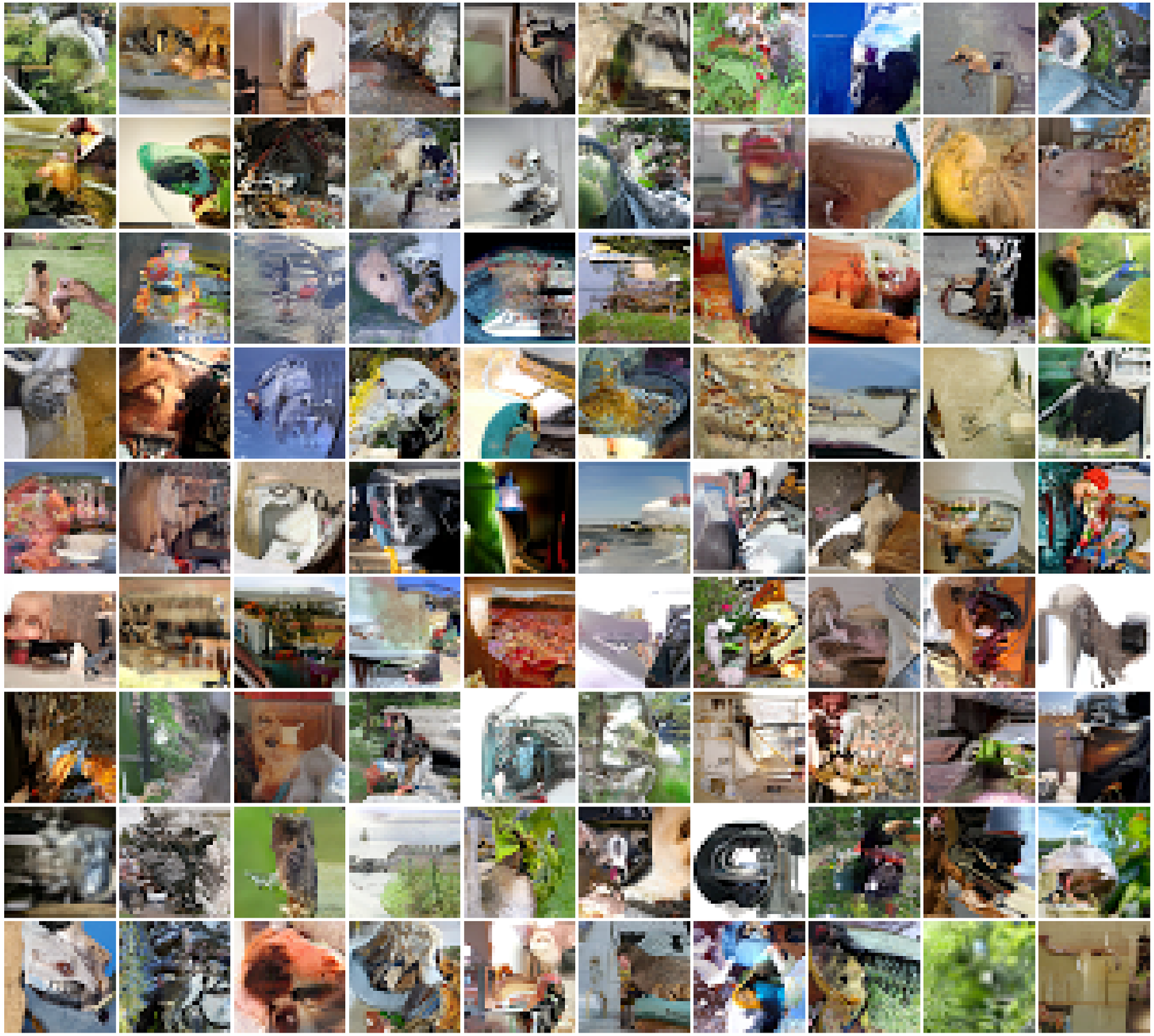
- Contexts are modeled using "diagonal BiLSTMS".



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

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

- 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]:	$\approx$ 84.62
DBN 2hl [2]:	$\approx$ 84.55
NADE [3]:	88.33
EoNADE 2hl (128 orderings) [3]:	85.10
EoNADE-5 2hl (128 orderings) [4]:	84.68
DLGM [5]:	$\approx$ 86.60
DLGM 8 leapfrog steps [6]:	$\approx$ 85.51
DARN 1hl [7]:	$\approx$ 84.13
MADE 2hl (32 masks) [8]:	86.64
DRAW [9]:	$\leq$ 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).

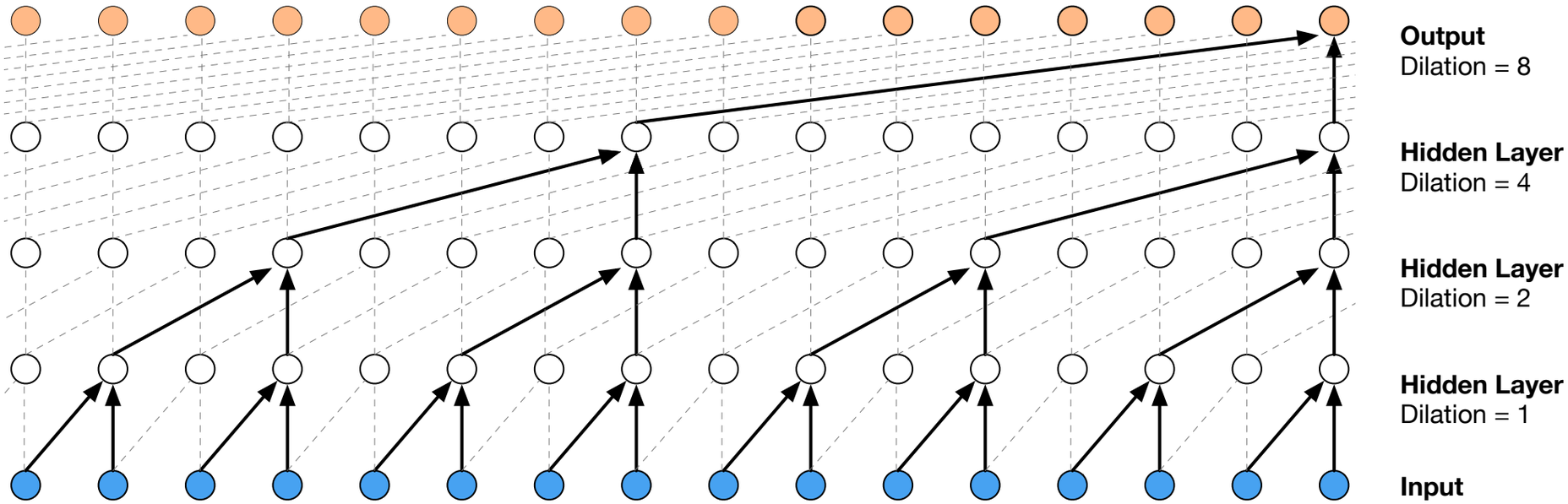
Model	NLL Test (Train)
Uniform Distribution:	8.00
Multivariate Gaussian:	4.70
NICE [1]:	4.48
Deep Diffusion [2]:	4.20
Deep GMMs [3]:	4.00
RIDE [4]:	3.47
PixelCNN:	3.14 (3.08)
Row LSTM:	3.07 (3.00)
Diagonal BiLSTM:	**3.00** (2.93)

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).

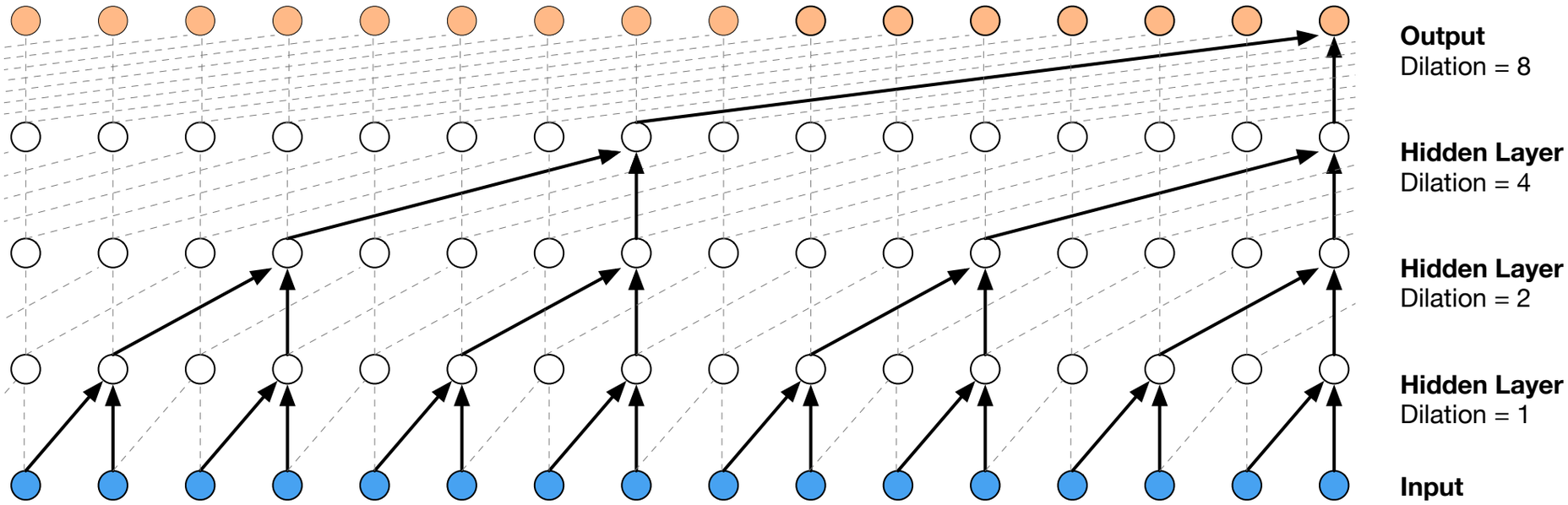
WaveNet [v.d Oord et al.]

- Auto-regressive model for speech.
- Speech is sampled at 16kHz and quantized with 256 bins.
- The model uses cross-entropy loss over those classes.



WaveNet [v.d Oord et al.]

- Auto-regressive model for speech.
- Speech is sampled at 16kHz and quantized with 256 bins.
- The model uses cross-entropy loss over those classes.



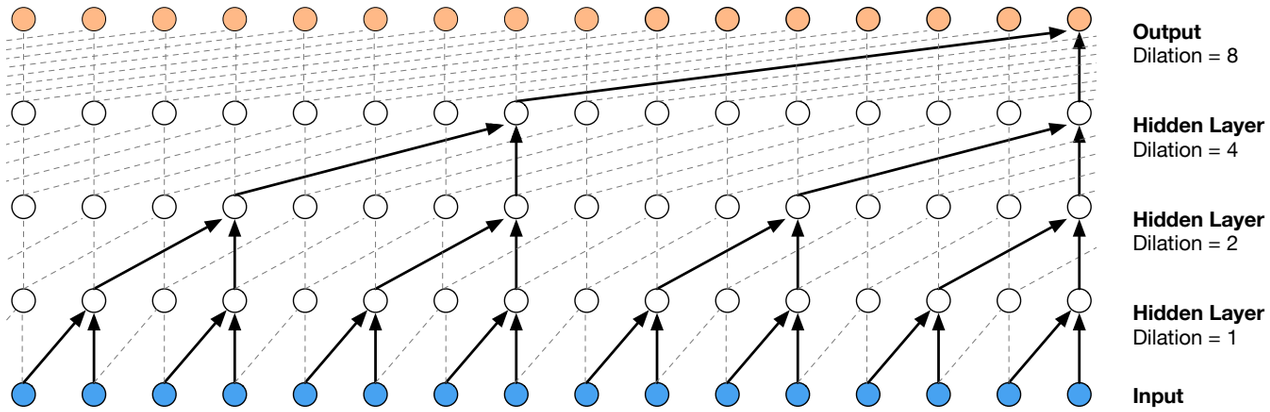
vs parametric model

text-to-speech

trained on music

WaveNet [v.d Oord et al.]

- Auto-regressive model for speech.
- Speech is sampled at 16kHz and quantized with 256 bins.
- The model uses cross-entropy loss over those classes.



- Pros: high-quality density model, very powerful, exploits inductive bias of local interactions, exact inference.
- Cons: expensive generation. Samples are sequentially generated!

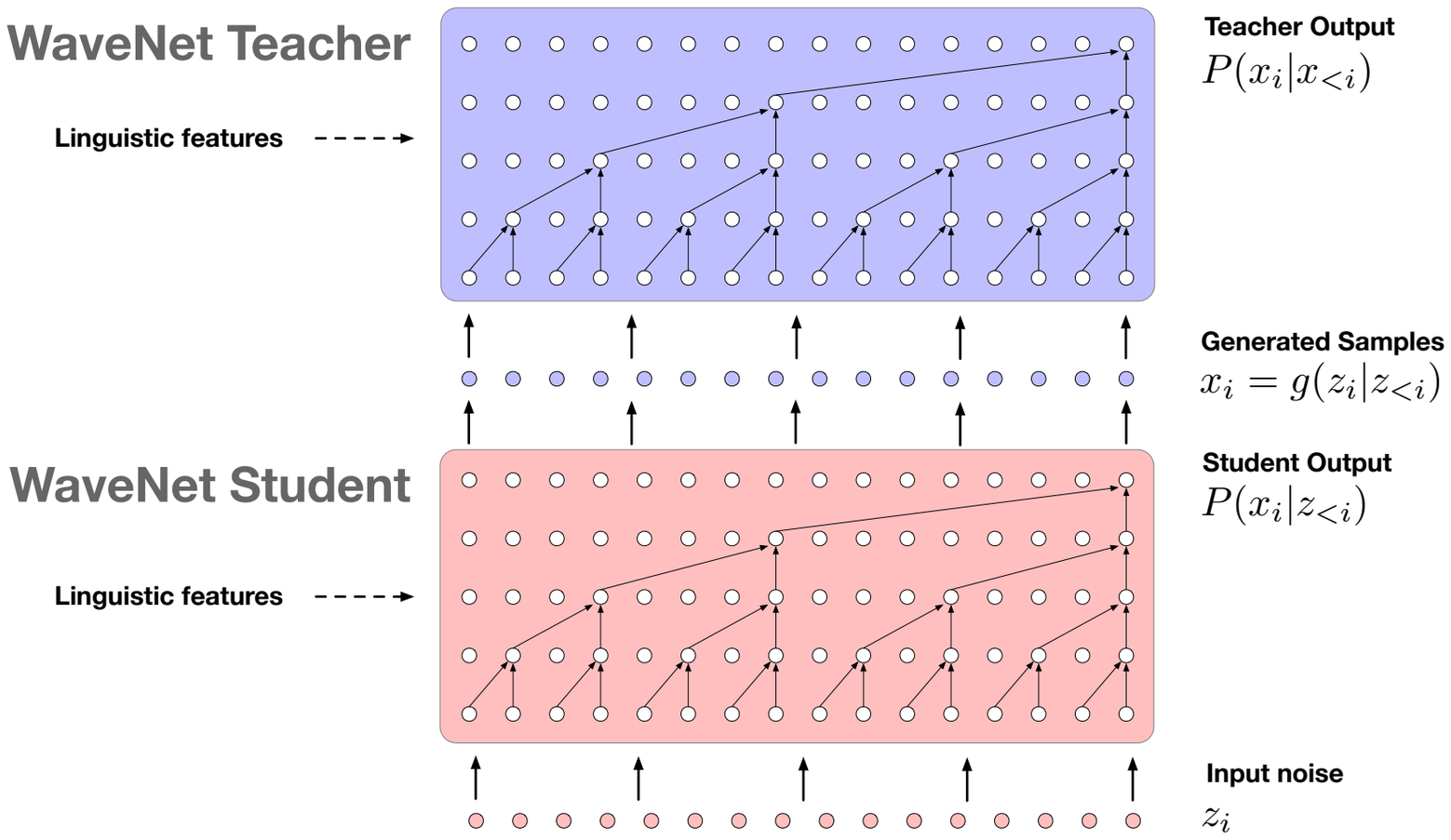
Parallel Wavenet [v.d. Oord et al.'17]

- This limitation is addressed with *parallel wavelet:*
  - Pretrained Wavelet teacher using previous model.
  - Student network is modeled as an Inverse Autoregressive Flow:

$$\mathbf{x} = \mathrm{IAF}(\mathbf{z}) \ , \ p(x_t \mid \mathbf{z}) = p(x_t \mid \mathbf{z}_{\leq t}) \ .$$

  - "Density Distillation": train student to minimise

$$D_{KL}(p_S \mid\mid p_T) = CE(p_S, p_T) - H(p_S) \ .$$



Conclusions and Open Directions

- Exact Inference in high-dimensional graphical models is generally intractable.
  - **Computational curse of dimensionality**
  - Two major algorithmic frameworks:
    - ❖ variational inference
    - ❖ MCMC.
  - Optimization is the bottleneck.

Conclusions and Open Directions

- Exact Inference in high-dimensional graphical models is generally intractable.
  - **Computational curse of dimensionality**
  - Two major algorithmic frameworks:
    - ❖ variational inference
    - ❖ MCMC.
  - Optimization is the bottleneck.

Conclusions and Open Directions

- Learning densities in high-dimensions is an ill-posed estimation problem
  - **Statistical curse of dimensionality**
  - Deep Generative Models:
    - ❖ Variational AutoEncoders: mixture models.
    - ❖ GANs: measure transportation models.
    - ❖ Auto-Regressive Models: exploit stationarity/locality of natural images/sounds.
  - Different tradeoffs between bias and variance
  - Geometrical priors are key to overcome this curse.
    - ❖ Most CV tasks are stable to geometrical transformations.
    - ❖ These are exploited with CNN architectures.

Optimization in Machine Learning

Optimization Set-up

- Our general problem is of the form

$$\min_{\Phi} \ \mathbb{E}_{z\sim\pi} f(z;\Phi) := F(\Phi) \ .$$

Supervised Learning:
$z = (x, y)$

$f(x, y; \Phi) = \log p(y|x; \Phi) = \ell(y, \Phi(x))$

π : joint data/labels distribution

Unsupervised Learning:
$z = x$

$f(x; \Phi) = \log \Phi(x)$

π : data distribution

Optimization Set-up

$$\min_{\Phi} \ \mathbb{E}_{z\sim\pi} f(z;\Phi) := F(\Phi) \ .$$

- Challenges:
  - *Statistical*: The function F to be optimized is unknown: only access to an estimator

$$\hat{F}_n(\Phi) = \frac{1}{n}\sum_{i<n} f(z_i,\Phi) \ , \ \{z_i\}_{i\leq n} : \text{training set.}$$

Optimization Set-up

$$\min_{\Phi} \ \mathbb{E}_{z\sim\pi} f(z;\Phi) := F(\Phi) \ .$$

- Challenges:
  - *Statistical*: The function F to be optimized is unknown: only access to an estimator

$$\hat{F}_n(\Phi) = \frac{1}{n}\sum_{i<n} f(z_i,\Phi) \ , \ \{z_i\}_{i\leq n} : \text{training set.}$$

  - *Analytical*: In practice, we search within a parametric functional class

$$\mathcal{F} = \{\Phi = \Phi(\,\cdot\,;\Theta)\,;\ \Theta \in \mathcal{X}\}$$

Optimization Set-up

$$\min_{\Phi} \ \mathbb{E}_{z\sim\pi} f(z;\Phi) := F(\Phi) \ .$$

- Challenges:
  - *Statistical*: The function F to be optimized is unknown: only access to an estimator

$$\hat{F}_n(\Phi) = \frac{1}{n}\sum f(z_i,\Phi) \ , \ \{z_i\}_{i\leq n} : \text{training set.}$$

  - *Analytical*: In practice, we search within a parametric functional class

$$\mathcal{F} = \{\Phi = \Phi(\,\cdot\,;\Theta)\,;\ \Theta\in\mathcal{X}\}$$

  - *Numerical*: Algorithms to optimize $\hat{F}_n$.
    - What can we say when $\hat{F}_n$ is non-convex?
    - What is the convergence rate of iterative solutions to stationary points? $\|\Theta^{(k)} - \Theta^*\| = O(h(k)) \ .$

Decomposition of Error

[Bottou, Bousquet '08]

- Define

$$\Phi^* = \arg\min_{\Phi} F(\Phi) \ , \ \text{optimal model} \ ,$$

$$\Phi^*_{\mathcal{F}} = \arg\min_{\Phi \in \mathcal{F}} F(\Phi) \ , \ \text{optimal achievable model in } \mathcal{F} \ ,$$

$$\Phi_{\mathcal{F},n} = \arg\min_{\Phi \in \mathcal{F}} \hat{F}_n(\Phi) \ , \ \text{optimal empirical model in } \mathcal{F} \ ,$$

$$\widetilde{\Phi}_{\mathcal{F},n} = \ \text{solution of our optimization of } \min_{\Phi \in \mathcal{F}} \hat{F}_n(\Phi) \ ,$$

Decomposition of Error

[Bottou, Bousquet '08]

- Define

$$\Phi^* = \arg\min_{\Phi} F(\Phi) \ , \ \text{optimal model} \ ,$$

$$\Phi^*_{\mathcal{F}} = \arg\min_{\Phi \in \mathcal{F}} F(\Phi) \ , \ \text{optimal achievable model in } \mathcal{F} \ ,$$

$$\Phi_{\mathcal{F},n} = \arg\min_{\Phi \in \mathcal{F}} \hat{F}_n(\Phi) \ , \ \text{optimal empirical model in } \mathcal{F} \ ,$$

$$\widetilde{\Phi}_{\mathcal{F},n} = \ \text{solution of our optimization of } \min_{\Phi \in \mathcal{F}} \hat{F}_n(\Phi) \ ,$$

- *Remark*: we can also modify empirical risk minimization with a regularizer (structured risk minimization):

$$\hat{F}_n(\Theta) = \frac{1}{n} \sum_{i \leq n} f(z_i, \Phi(\Theta)) + \lambda \mathcal{R}(\Theta) \ ,$$

Decomposition of Error

[Bottou, Bousquet '08]

- Define

$$\Phi^* = \arg\min_{\Phi} F(\Phi) \ , \ \text{optimal model} \ ,$$

$$\Phi^*_{\mathcal{F}} = \arg\min_{\Phi\in\mathcal{F}} F(\Phi) \ , \ \text{optimal achievable model in } \mathcal{F} \ ,$$

$$\Phi_{\mathcal{F},n} = \arg\min_{\Phi\in\mathcal{F}} \hat{F}_n(\Phi) \ , \ \text{optimal empirical model in } \mathcal{F} \ ,$$

$$\widetilde{\Phi}_{\mathcal{F},n} = \ \text{solution of our optimization of } \min_{\Phi\in\mathcal{F}} \hat{F}_n(\Phi) \ ,$$

- Regret is decomposed as

$$\begin{aligned} F(\widetilde{\Phi}_{\mathcal{F},n}) - F(\Phi^*) = {} & F(\Phi^*_{\mathcal{F}}) - F(\Phi^*) && \text{(approximation error)} \\ & + F(\Phi_{\mathcal{F},n}) - F(\Phi^*_{\mathcal{F}}) && \text{(estimation error)} \\ & + F(\widetilde{\Phi}_{\mathcal{F},n}) - F(\Phi_{\mathcal{F},n}) \ . && \text{(optimization error)} \end{aligned}$$

Constrained Approximation, Estimation and Optimization

[Bottou, Bousquet '08]

- Our goal is thus to minimize regret with respect to

$$\text{model } \mathcal{F}, \text{ optimization tolerance } \rho, \text{ number of examples } n$$

$$\text{subject to } n \leq n_{max}, \text{ compute time } T \leq T_{max}$$

Constrained Approximation, Estimation and Optimization

[Bottou, Bousquet '08]

- Our goal is thus to minimize regret with respect to

$$\text{model } \mathcal{F}, \text{ optimization tolerance } \rho, \text{ number of examples } n$$

$$\text{subject to } n \leq n_{max}, \text{ compute time } T \leq T_{max}$$

- Constrained optimization trade-offs:

Approximation error decreases as $\mathcal{F}$ gets larger

Estimation error decreases as n gets larger.
Estimation error increases as $\mathcal{F}$ gets larger.

Optimization error increases as ρ gets larger.

Conclusions and Open Problems

- Stochastic Optimization and large scale learning.
  - In most learning tasks, we are faced with an objective function of the form

$$E(\theta) = \frac{1}{L}\sum_{l\leq L} \ell(\Phi(x_l;\theta), y_l) + \mathcal{R}(\theta) = L^{-1}\sum_{l\leq L} E_l(\theta) \ .$$

  - Stochastic Gradient Descent [Robbins & Monro,'50s]:

$$\theta^{(n+1)} = \theta^{(n)} - \gamma\nabla_\theta E_{l_n}(\theta^{(n)}) \ . \ \ l_n \sim \mathrm{Cat}[1, L] \ .$$

  - Excellent computational complexity.

  - Generalizes better than plain gradient descent.

  - Tradeoffs between estimation

SGD Questions

- Stochastic Optimization and Implicit Regularization.
  - In some setups (e.g. matrix factorization, least squares), authors have proved that SGD reaches optimal statistical rates.
  - Also true in deep models? Why?
  - How is SGD related to Markov Chains?
    - Replacing the sampling noise with Gaussian noise leads to Langevin Dynamics, which converge to Gibbs distributions of the form $p(\theta) \propto e^{-\beta E(\theta)}$.

Mathematics of Deep Learning

- Upcoming graduate-level course next spring.

- Two Main Topics:
  - Optimization: explore the links between stochastic optimization, non-convexity and generalization.
  - Geometry: study how geometric properties of the data and the tasks (e.g. invariance, group symmetries, stability) can be leveraged to beat the curse of dimensionality.