EECS 498/598: Deep Learning
Lecture 9. Generative Models 2

Honglak Lee

03/15/2019

Overview

GANs

- Improved techniques for training GANs
- Conditional Generation
- Case study
  - SOTA image synthesis
- Other applications

VAEs

- Improved techniques for training VAEs
- Conditional Generation
- Case study
  - Sequence data
  - Discrete latent variables
- Other applications

GANs

- Improved techniques for training GANs
  - Tips and Tricks
  - Learning interpretable latent representations
  - Principled methods with better training objectives
- Conditional Generation
  - Class conditioning
  - Conditioning on richer structures: Text, bounding boxes, images, etc.
- Case study
  - SOTA image synthesis
- Other applications

GANs

- **Improved techniques for training GANs**
  - **Tips and Tricks**
  - Learning interpretable latent representations
  - Principled methods with better training objectives
- Conditional Generation
  - Class conditioning
  - Conditioning on richer structures: Text, bounding boxes, images, etc.
- Case study
  - SOTA image synthesis
- Other applications

Review: GAN

- Real data distribution P_r

  Generator distribution P_g, implemented as $x = G(z), z \sim P(z)$

$$\min_G \max_D \quad V(D,G) \;=\; \mathbb{E}_{x\sim P_r}[\log D(x)] + \mathbb{E}_{x\sim P_g}[\log(1-D(x))]$$

- Discriminator objective

$$\min_D \; -\mathbb{E}_{x\sim P_r}[\log D(x)] - \mathbb{E}_{x\sim P_g}[\log(1-D(x))]$$

- Generator objective

$$\min_G \quad \mathbb{E}_{x\sim P_g}[\log(1-D(x))]$$

Review: GAN

- Real data distribution P_r

  Generator distribution P_g, implemented as $x = G(z), z \sim P(z)$

$$\min_G \max_D \; V(D,G) = \mathbb{E}_{x\sim P_r}[\log D(x)] + \mathbb{E}_{x\sim P_g}[\log(1 - D(x))]$$

- Discriminator objective

$$\min_D \; -\mathbb{E}_{x\sim P_r}[\log D(x)] - \mathbb{E}_{x\sim P_g}[\log(1 - D(x))]$$

- Generator objective

$$\min_G \; \mathbb{E}_{x\sim P_g}[\log(1 - D(x))]$$

Review: GAN

- Real data distribution P_r

  Generator distribution P_g, implemented as $x = G(z), z \sim P(z)$

$$\min_G \max_D \quad V(D,G) \;=\; \mathbb{E}_{x\sim P_r}[\log D(x)] + \mathbb{E}_{x\sim P_g}[\log(1 - D(x))]$$

- Discriminator objective

$$\min_D \; -\mathbb{E}_{x\sim P_r}[\log D(x)] - \mathbb{E}_{x\sim P_g}[\log(1 - D(x))]$$

- Generator objective

$$\min_G \quad \mathbb{E}_{x\sim P_g}[\log(1 - D(x))]$$

GAN training

- Training GANs is hard
- Training GANs requires finding a Nash equilibrium of a non-convex game
- Continuous, high dimensional parameters
- Gradient descent techniques are designed to optimize convex cost functions, can fail to converge when used to seek Nash equilibrium
- In practice, many tricks are used to train GANs

Improved techniques for training GANs

Feature Matching

- Generate data that matches the statistics of real data
- Train generator to match expected value of intermediate discriminator layer:

$$||\mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}} \mathbf{f}(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})} \mathbf{f}(G(\boldsymbol{z}))||_2^2$$

(Where f(x) is some activations of an intermediate layer)

- Still no guarantee of reaching G*
- Works well empirically

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. In *Advances in neural information processing systems* (pp. 2234-2242).

Improved techniques for training GANs

Minibatch Discrimination

- Discriminator looks at generated examples independently
- Can’t discern generator mode collapse
- Solution: Use other examples as side information and do collective discrimination
- If the generator has low entropy, much lower than real data, it may be easier to detect this with a discriminator that sees multiple samples.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. In *Advances in neural information processing systems* (pp. 2234-2242).

Improved techniques for training GANs

- **Historical Averaging**

$$||\boldsymbol{\theta} - \frac{1}{t}\sum_{i=1}^{t}\boldsymbol{\theta}[i]||^2$$

- **Label Smoothing**
  - e.g., 0.1 or 0.9 instead of 0 or 1
  - Negative values set to zero
- **Virtual Batch Normalization**
  - Use reference minibatch of examples that are chosen once and fixed at the start of training, and on x itself. The reference batch is normalized using only its own statistics.
  - Expensive, used only in generator

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. In *Advances in neural information processing systems* (pp. 2234-2242).

GANs

- **Improved techniques for training GANs**
  - Tips and Tricks
  - **Learning interpretable latent representations**
  - Principled methods with better training objectives
- Conditional Generation
  - Class conditioning
  - Conditioning on richer structures: Text, bounding boxes, images, etc.
- Case study
  - SOTA image synthesis
- Other applications

InfoGAN

- In vanilla GAN we use a noise variable to control generation
- Can we discover interpretable latent variables and use them to control generation ?
- Idea: Apply mutual-Information regularizer between conditioning latent variable and generated output

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*(pp. 2172-2180).

InfoGAN

- Decompose the latent variable into two parts $z \rightarrow [z, c_1, \ldots, c_L]$
  - Incompressible noise z
  - Latent code $c = [c_1, \ldots, c_L]$
- Maximize mutual information $I(c; G(z, c)) = H(c) - H(c|G(z, c))$

$$\min_G \max_D V_I(D, G) = V(D, G) - \lambda I(c; G(z, c))$$

- Maximizing $I(c; G(z, c))$ directly is intractable as it required access to posterior $P(c|x)$
  - Maximize a lower bound based on auxiliary distribution $Q(c|x)$

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*(pp. 2172-2180).

InfoGAN

- Mutual Information lower bound

$$
\begin{aligned}
I(c; G(z,c)) &= H(c) - H(c|G(z,c)) \\
&= \mathbb{E}_{x\sim G(z,c)}[\mathbb{E}_{c'\sim P(c|x)}[\log P(c'|x)]] + H(c) \\
&= \mathbb{E}_{x\sim G(z,c)}[\underbrace{D_{\mathrm{KL}}(P(\cdot|x) \parallel Q(\cdot|x))}_{\geq 0} + \mathbb{E}_{c'\sim P(c|x)}[\log Q(c'|x)]] + H(c) \\
&\geq \mathbb{E}_{x\sim G(z,c)}[\mathbb{E}_{c'\sim P(c|x)}[\log Q(c'|x)]] + H(c) \\
&= E_{c\sim P(c),x\sim G(z,c)}[\log Q(c|x)] + H(c) \\
&= L_I(G,Q)
\end{aligned}
$$

- InfoGAN objective $\min_{G,Q} \max_{D} V_{\mathrm{InfoGAN}}(D,G,Q) = V(D,G) - \lambda L_I(G,Q)$

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*(pp. 2172-2180).

InfoGAN

- Lower bound has a simple practical interpretation

$$L_I(G,Q) = E_{c\sim P(c),x\sim G(z,c)}[\log Q(c|x)] + H(c)$$

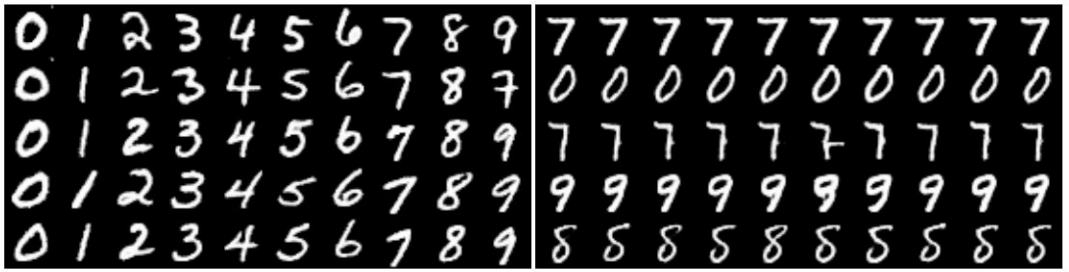
Train a network to recover the code from the output

Can be optimized for simple distributions
For simplicity, fix the code distribution and treat as constant

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*(pp. 2172-2180).

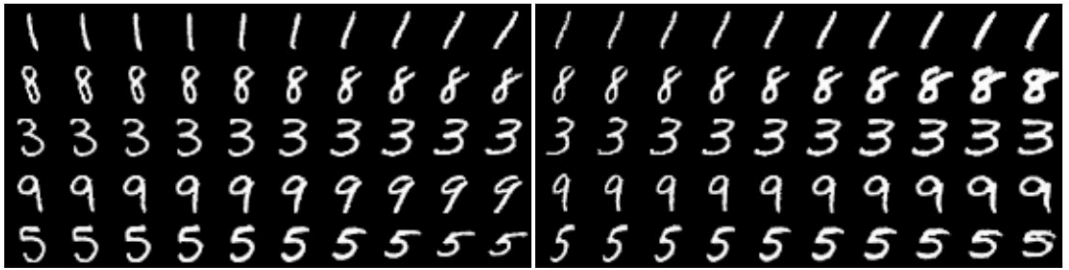
InfoGAN

- Learning interpretable latent code on MNIST $c = [c_1, c_2, c_3]$: $c_1 \sim Cat(10), c_2, c_3 \sim Unif(-1, 1)$



(a) Varying c_1 on InfoGAN (Digit type)

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



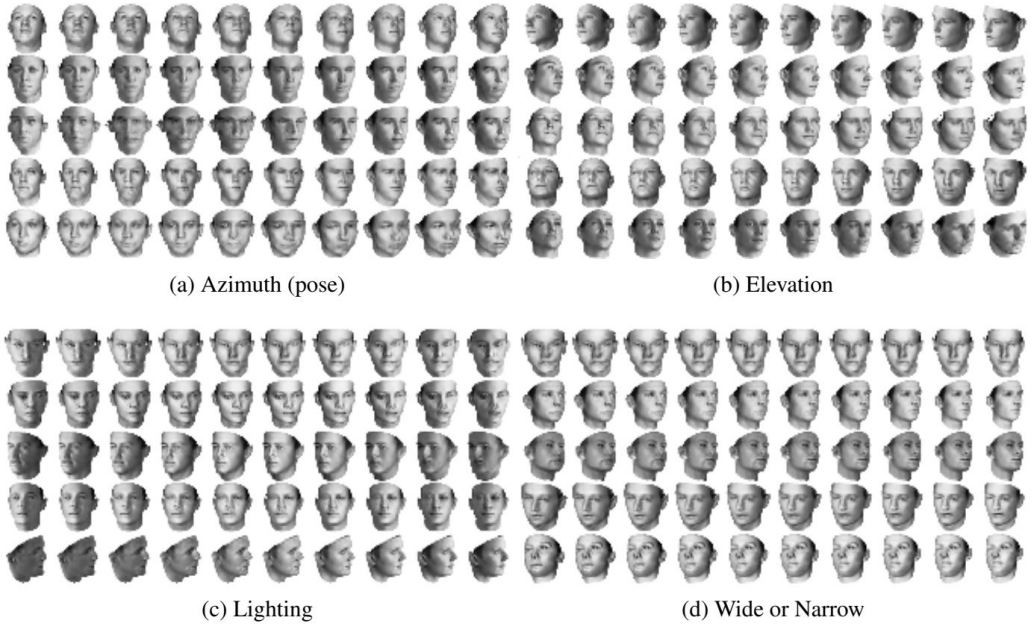
(c) Varying c_2 from -2 to 2 on InfoGAN (Rotation)

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

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*(pp. 2172-2180).

InfoGAN

- Manipulating latent code on 3D faces



Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*(pp. 2172-2180).

GANs

- **Improved techniques for training GANs**
  - Tips and Tricks
  - Learning interpretable latent representations
  - **Principled methods with better training objectives**
- Conditional Generation
  - Class conditioning
  - Conditioning on richer structures: Text, bounding boxes, images, etc.
- Case study
  - SOTA image synthesis
- Other applications

Recap: Vanilla GAN

- Real data distribution P_r

  Generator distribution P_g, implemented as $x = G(z), z \sim P(z)$

$$\min_G \max_D \quad V(D,G) \;=\; \mathbb{E}_{x\sim P_r}[\log D(x)] + \mathbb{E}_{x\sim P_g}[\log(1 - D(x))]$$

- Discriminator objective

$$-\mathbb{E}_{x\sim P_r}[\log D(x)] - \mathbb{E}_{x\sim P_g}[\log(1 - D(x))]$$

- Generator objective

$$\mathbb{E}_{x\sim P_g}[\log(1 - D(x))]$$

GAN Training

- Better discriminator leads to vanishing gradients for generator
  - For the optimal discriminator, it can be shown that the generator loss is equivalent to a Jenson-Shannon divergence

Optimal Discriminator

$$D^*(x) = \frac{P_r(x)}{P_r(x) + P_g(x)}$$

Generator loss becomes

$$\longrightarrow \quad 2JS(P_r||P_g) - 2\log 2$$

- If P_r and P_g have low support, we can show that the generator gradient vanishes

GAN Training

- Idea: Replace JS divergence with Wasserstein distance
  - When the distributions are supported by low dimensional manifolds
    - KL or JS are binary, no meaningful gradient
    - W is continuous and differentiable, hence always sensible

- Distance metrics
  - KL

$$KL(P||Q) = \mathbb{E}_P \log \frac{P}{Q}$$

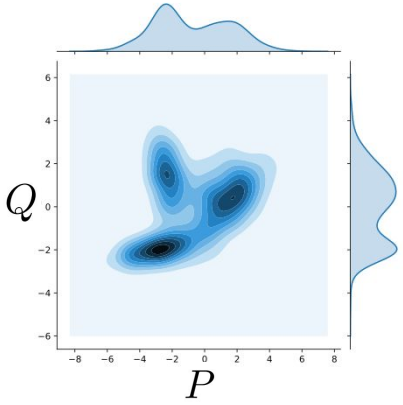
  - JS

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

  - Wasserstein

$$W(P||Q) = \inf_{\gamma \in \Pi(P,Q)} \mathbb{E}_{(x,y)\sim\gamma}[||x-y||]$$

Wasserstein distance



- $\gamma(x, y)$ indicates a plan to “transport “mass” from x to y, when deforming P into Q.
- Valid transformation should satisfy the following constraint:

$$\int \gamma(x, y)dx = Q(y) \qquad \int \gamma(x, y)dy = P(x)$$

- Given a cost function c(x,y) that's defined as a cost to move a probability mass from x to y, the cost of the transport plan can be defined as:

$$\mathbb{E}_{(x,y)\sim\gamma}\left[c(x, y)\right] = \int \gamma(x, y)c(x, y)dxdy$$

- $\Pi(P,Q)$ denotes the set of all joint distributions $\gamma(x, y)$ whose marginals are P and Q, respectively
- The Wasserstein (or Earth-Mover) distance is then the cost of the “optimal” transport plan; if c is L2 distance, it can be

$$\begin{aligned} W(P||Q) &= \inf_{\gamma\in\Pi(P,Q)} \mathbb{E}_{(x,y)\sim\gamma}\left[c(x, y)\right] \\ &= \inf_{\gamma\in\Pi(P,Q)} \mathbb{E}_{(x,y)\sim\gamma}\left[\|x - y\|\right] \end{aligned}$$

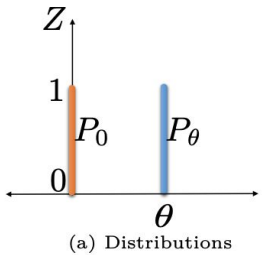
https://en.wikipedia.org/wiki/Wasserstein_metric

Distance metrics

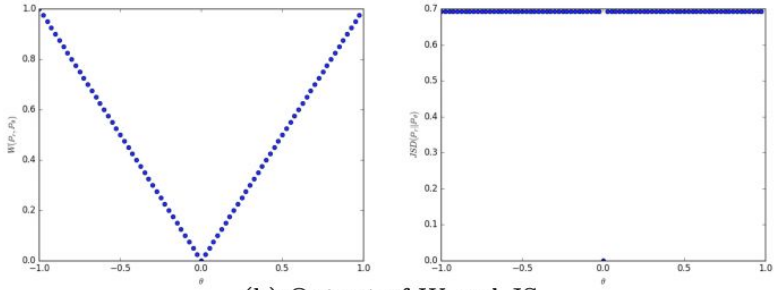
- When the distributions are supported by low dimensional manifolds
  - KL or JS are binary, no meaningful gradient
  - W is continuous and differentiable, hence always sensible

P_0: distribution of $(0, Z)$, where $Z \sim U[0, 1]$
P_θ: distribution of (θ, Z), where θ is a single real parameter

- $KL(P_0||P_\theta) = KL(P_\theta||P_0) = \begin{cases} +\infty & \text{if } \theta \neq 0 \\ 0 & \text{if } \theta = 0 \end{cases}$
- $JS(P_0||P_\theta) = \begin{cases} \log 2 & \text{if } \theta \neq 0 \\ 0 & \text{if } \theta = 0 \end{cases}$
- $W(P_0||P_\theta) = |\theta|$



(a) Distributions


(b) Output of W and JS

Slide credit: Chunyuan Li

Wasserstein distance

- The infimum is intractable
- Wasserstein distance has a dual form

$$W(P_r, P_g) = \sup_{||f||_L \leq 1} \mathbb{E}_{x \sim P_r}[f(x)] - \mathbb{E}_{x \sim P_g}[f(x)]$$
$$= \frac{1}{K} \sup_{||f||_L \leq K} \mathbb{E}_{x \sim P_r}[f(x)] - \mathbb{E}_{x \sim P_g}[f(x)]$$

  where sup is over all the K-Lipschitz functions

- Consider a w-parameterized family of functions $\{f_w\}_{w \in W}$ that are all K-Lipschitz

$$W(P_r, P_g) = \max_{w \in W} \mathbb{E}_{x \sim P_r}[f_w(x)] - \mathbb{E}_{x \sim P_g}[f_w(x)]$$

  For example, $W = [-c, c]^l$

Slide credit: Chunyuan Li

WGAN

$$W(P_r, P_g) = \max_{w \in W} \mathbb{E}_{x \sim P_r}[f_w(x)] - \mathbb{E}_{x \sim P_g}[f_w(x)]$$

- The loss for discriminator/critic

$$\mathbb{E}_{x \sim P_r}[f_w(x)] - \mathbb{E}_{x \sim P_g}[f_w(x)]$$

- The loss for generator

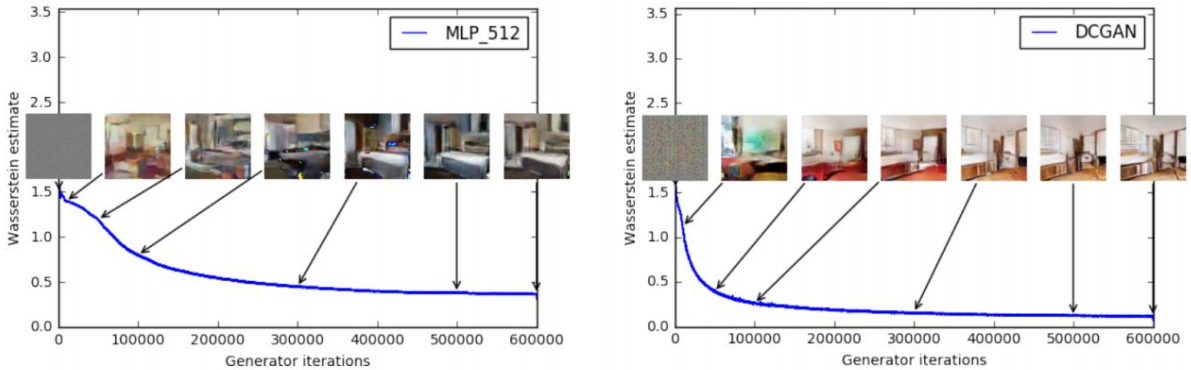
$$-\mathbb{E}_{x \sim P_g}[f_w(x)] = -\mathbb{E}_{z \sim p(z)}[f_w(g_\theta(z))]$$

- Main difference from vanilla GAN
  - Remove sigmoid in last layer of D
  - Remove the log in loss of D and G
  - Clip the parameter updates of D in an interval centered at 0

Slide credit: Chunyuan Li

WGAN

- The WGAN critic provides a meaningful loss metric that correlates with the generator's convergence and sample quality.
- WGAN attempts to train the critic relatively well before each generator update, the loss function at this point is an estimate of the EM distance.



Slide credit: Chunyuan Li

Other widely used techniques for GAN training

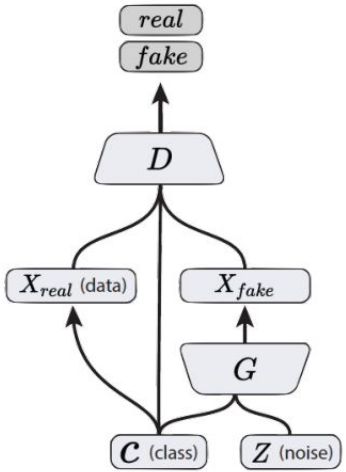
- BEGAN - BEGAN: Boundary Equilibrium Generative Adversarial Networks
- LSGAN - Least Squares Generative Adversarial Networks
- EBGAN - Energy-based Generative Adversarial Network
- MDGAN - Mode Regularized Generative Adversarial Networks
- Unrolled GAN

GANs

- Improved techniques for training GANs
  - Tips and Tricks
  - Learning interpretable latent representations
  - Principled methods with better training objectives
- **Conditional Generation**
  - Class conditioning
  - Conditioning on richer structures: Text, bounding boxes, images, etc.
- Case study
  - SOTA image synthesis
- Other applications

Conditional GANs

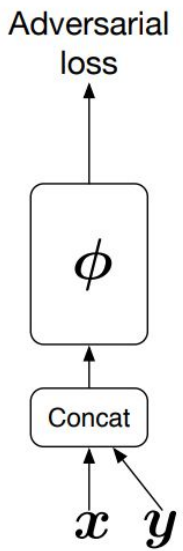
- Simple modification to the original GAN framework that conditions the model on additional information



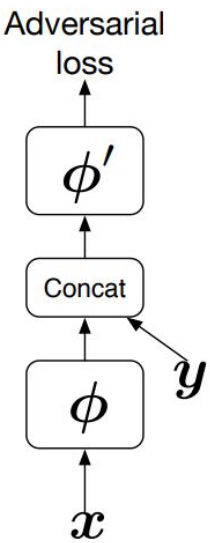
(Mirza & Osindero, 2014)

Incorporating conditioning information

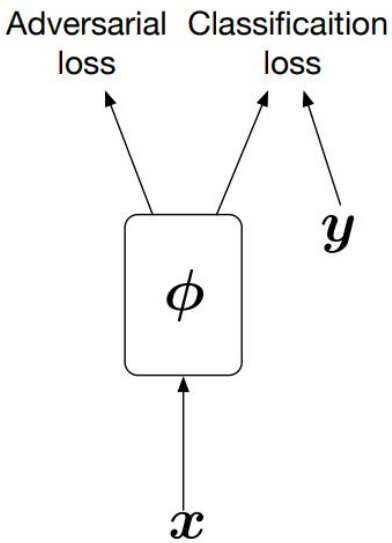
cGAN, input concat
(Mirza & Osindero, 2014)



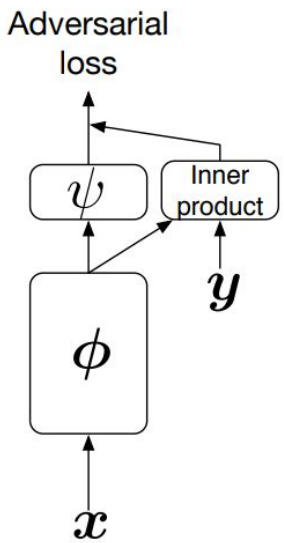
cGAN, hidden concat
(Reed et al., 2016)



AC-GAN
(Odena et al., 2017)



cGAN, projection discriminator
(Miyato et al., 2018)



Miyato, T., & Koyama, M. (2018). cGANs with projection discriminator. *arXiv preprint arXiv:1802.05637*.

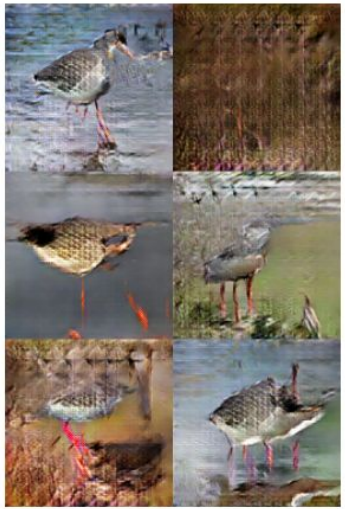
Class conditional generation

- Samples from AC-GAN

monarch butterfly

goldfinch

daisy



redshank

grey whale

Odena, A., Olah, C., & Shlens, J. (2017, August). Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (pp. 2642-2651). JMLR. org.

Class conditional generation

- Samples from cGAN with projection discriminator

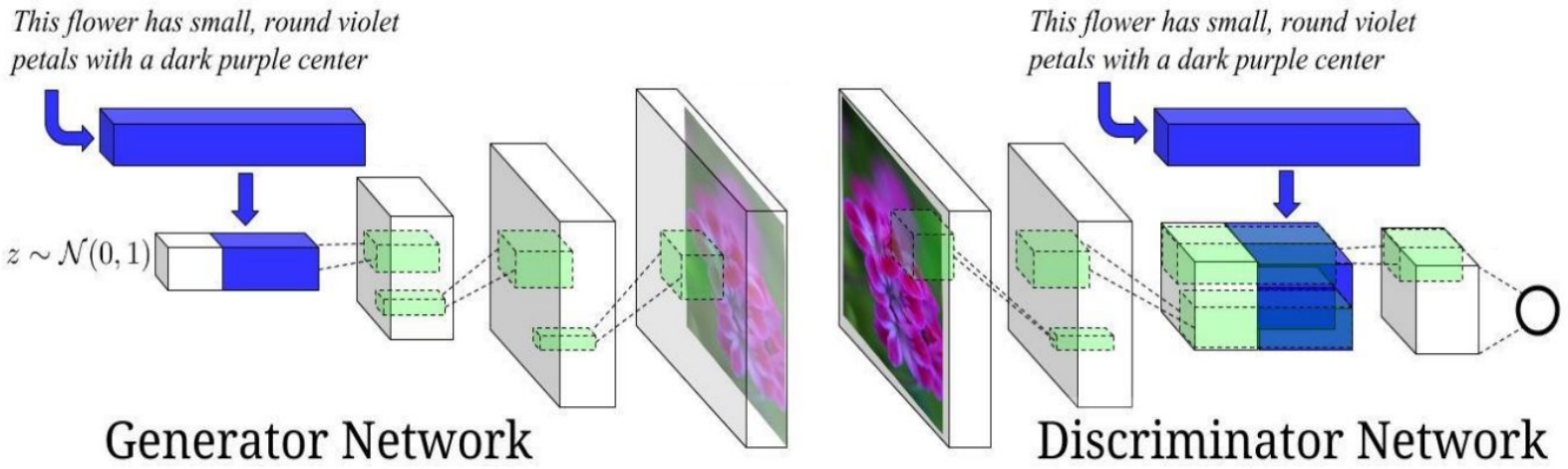
Tibetan terrier

Mushroom

Miyato, T., & Koyama, M. (2018). cGANs with projection discriminator. *arXiv preprint arXiv:1802.05637*.

Conditional generation

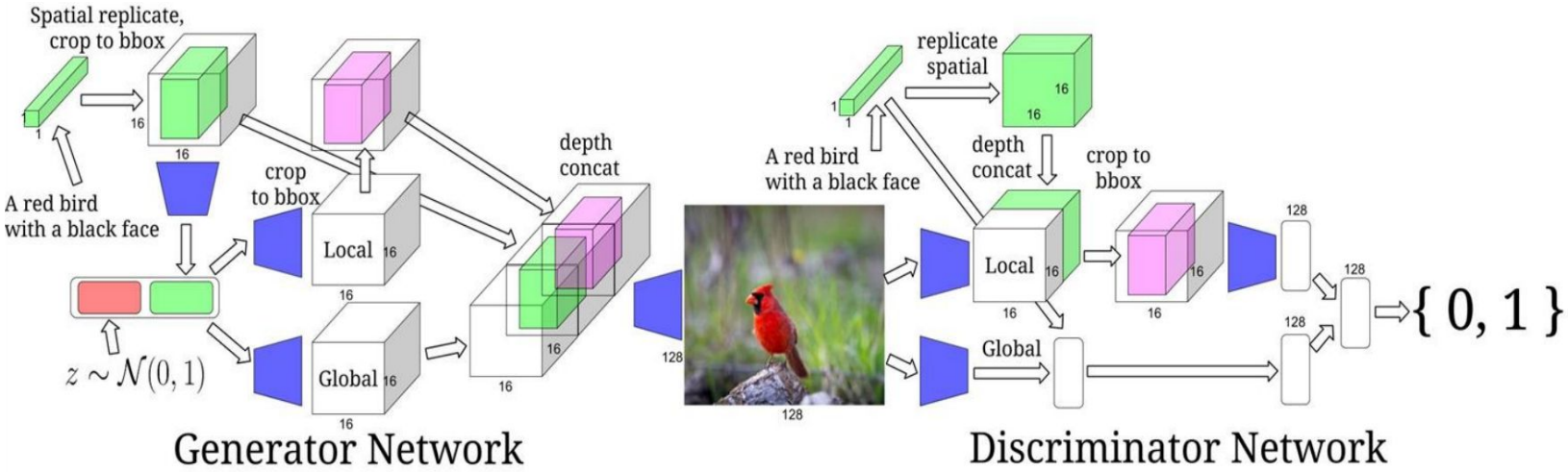
- Conditioning on text



Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., & Lee, H. (2016). Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*.

Conditional generation

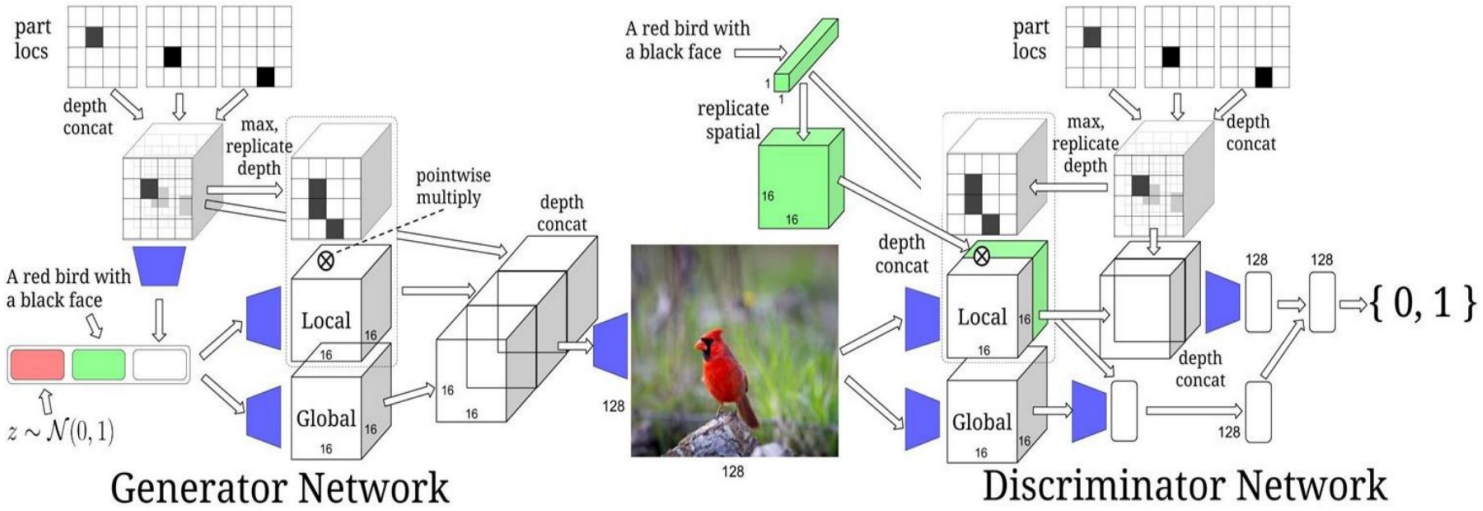
- Conditioning on bounding box



Reed, S. E., Akata, Z., Mohan, S., Tenka, S., Schiele, B., & Lee, H. (2016). Learning what and where to draw. In *NIPS* (pp. 217-225).

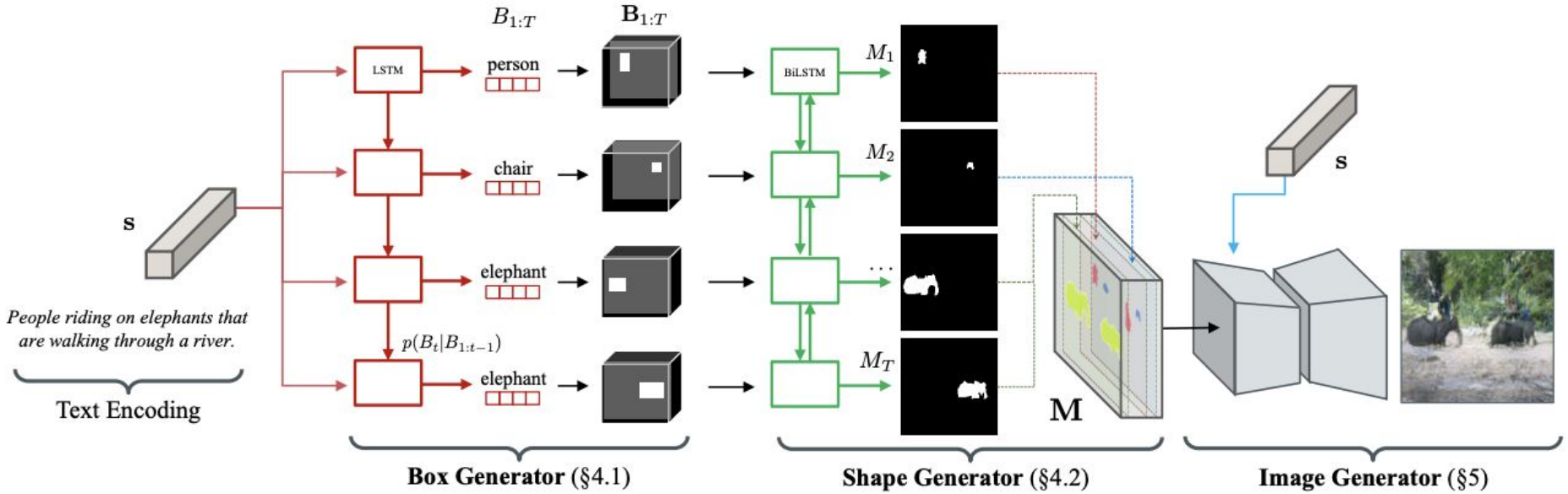
Conditional generation

- Conditioning on keypoints



Reed, S. E., Akata, Z., Mohan, S., Tenka, S., Schiele, B., & Lee, H. (2016). Learning what and where to draw. In *NIPS* (pp. 217-225).

Hierarchical semantic layout



Hong, S., Yang, D., Choi, J., & Lee, H. (2018). Inferring semantic layout for hierarchical text-to-image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 7986-7994).

Conditional generation

- Conditioning on text

this small bird has a pink breast and crown, and black primaries and secondaries.

this magnificent fellow is almost all black with a red crest, and white cheek patch.

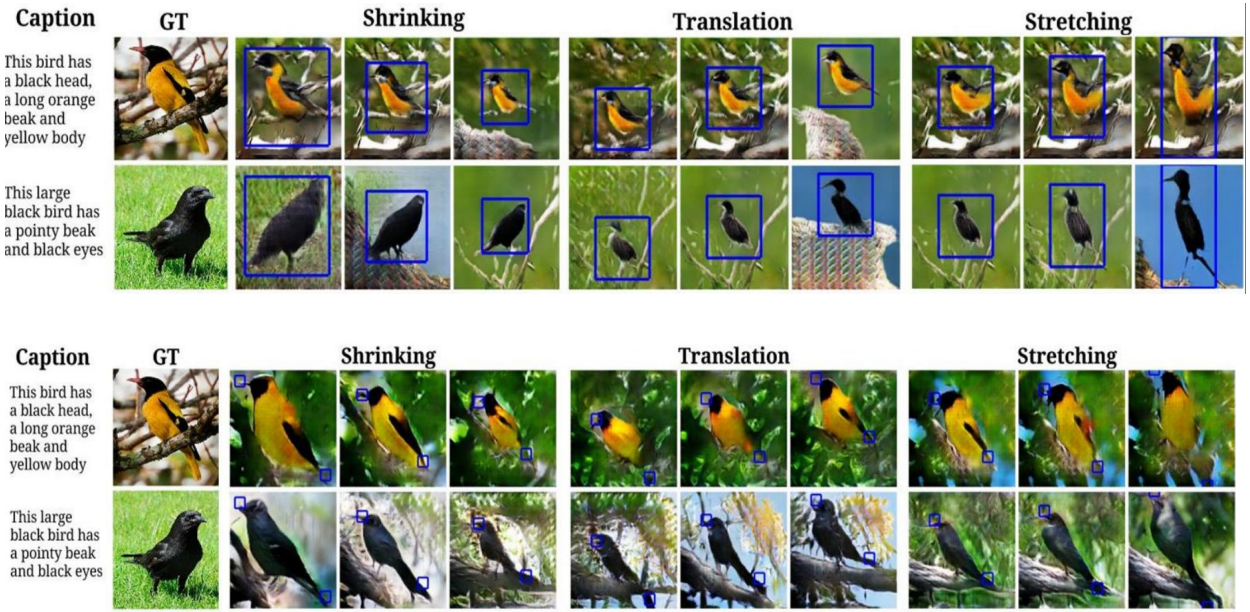
the flower has petals that are bright pinkish purple with white stigma

this white and yellow flower have thin white petals and a round yellow stamen

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., & Lee, H. (2016). Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*.
Reed, S. E., Akata, Z., Mohan, S., Tenka, S., Schiele, B., & Lee, H. (2016). Learning what and where to draw. In *NIPS* (pp. 217-225).

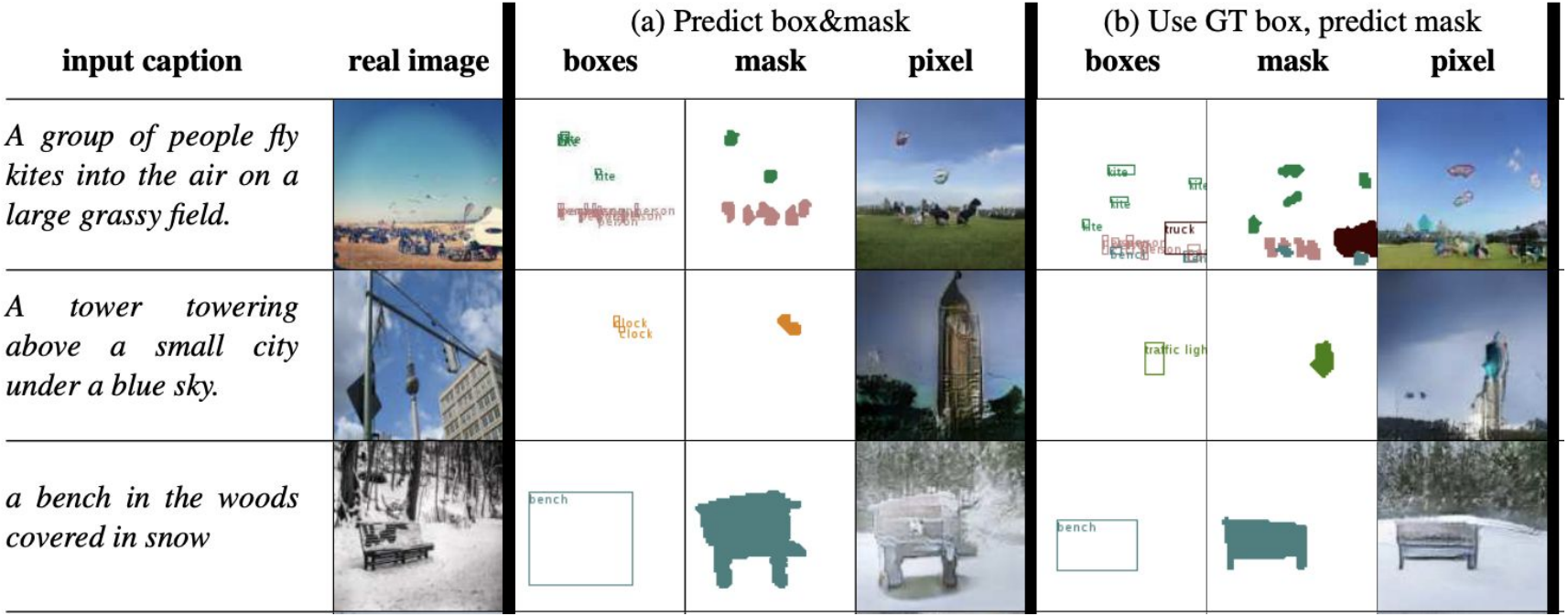
Conditional generation

- Conditioning on bounding boxes and keypoints



Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., & Lee, H. (2016). Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*.
Reed, S. E., Akata, Z., Mohan, S., Tenka, S., Schiele, B., & Lee, H. (2016). Learning what and where to draw. In *NIPS* (pp. 217-225).

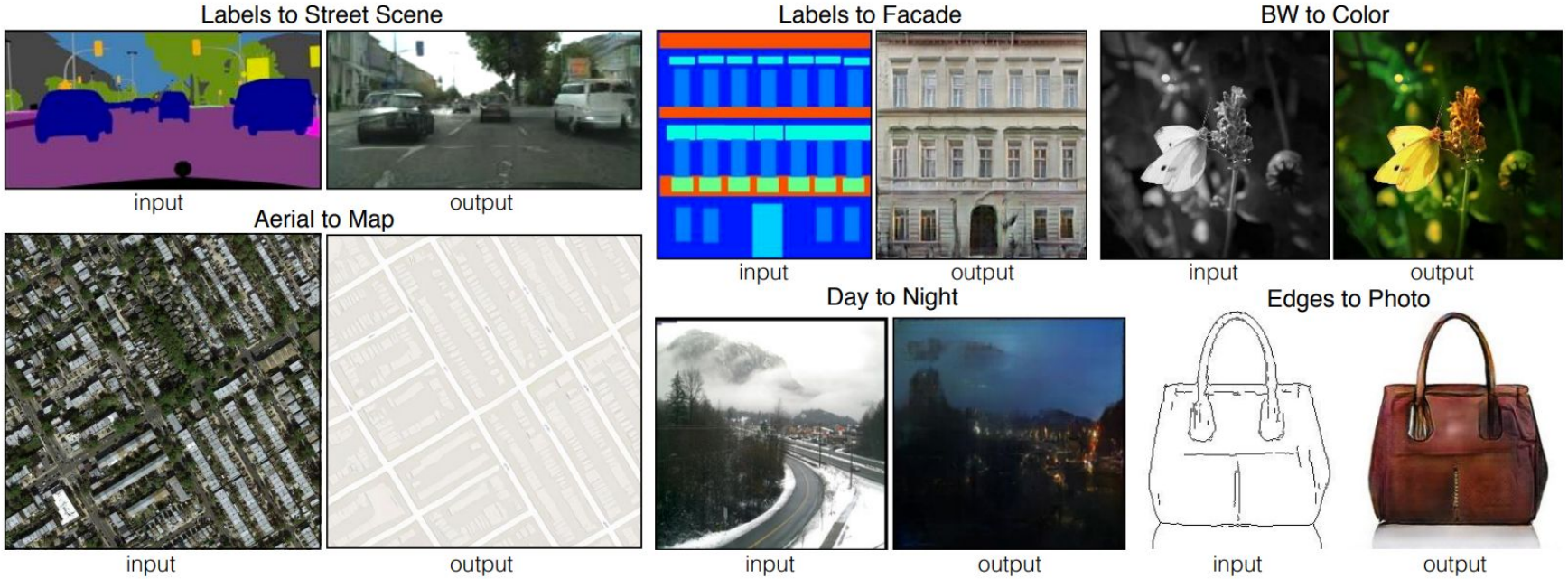
Conditional generation



Hong, S., Yang, D., Choi, J., & Lee, H. (2018). Inferring semantic layout for hierarchical text-to-image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 7986-7994).

Image-to-Image Translation

- Many problems can be posed as image-to-image translation problems

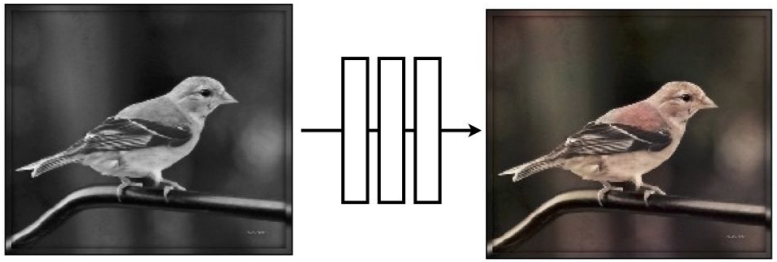


Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1125-1134).

Image-to-Image Translation

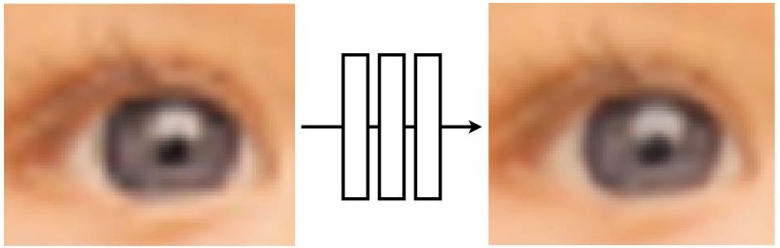
- What is the right loss function for image to image translation ?
- L2 regression leads to mode averaging

Image colorization



[Zhang, Isola, Efros, ECCV 2016]

Super-resolution



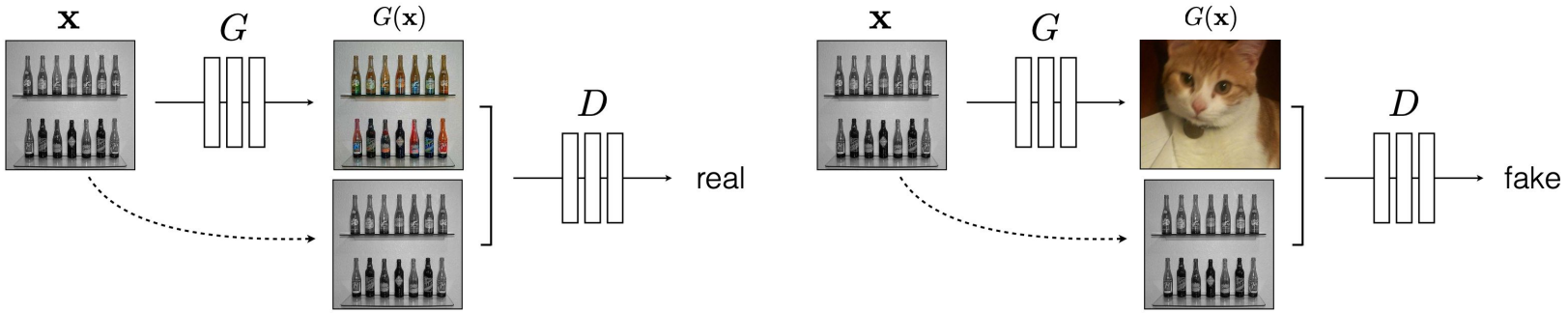
[Johnson, Alahi, Li, ECCV 2016]

- Use a conditional GAN to learn the loss

Slide credit: Phillip Isola

Image-to-Image Translation

- Conditional GAN loss
- Note that there is no noise input
  - Generator chooses to ignore it



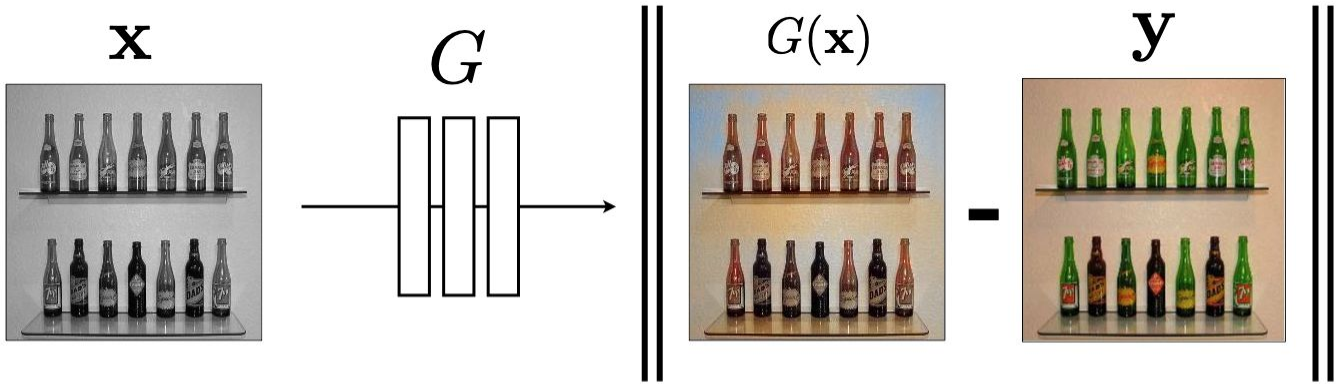
$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x}, \mathbf{y}))\;]$$

Slide credit: Phillip Isola

Image-to-Image Translation

- Conditional GAN loss

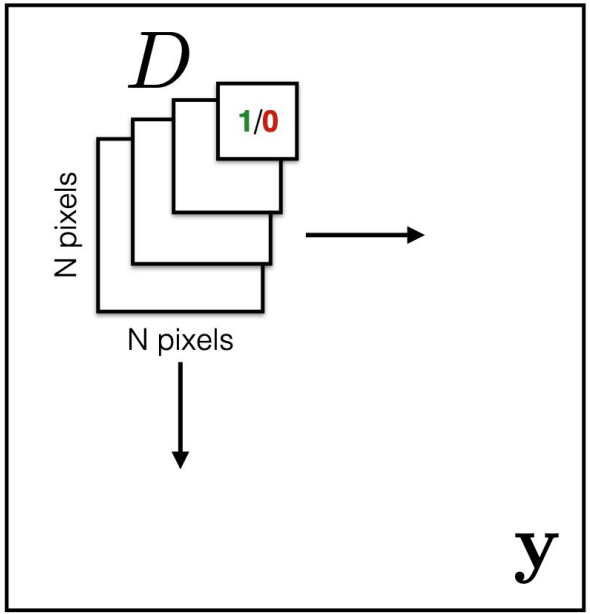
$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G)$$



Slide credit: Phillip Isola

Image-to-Image Translation

- Patch discriminator



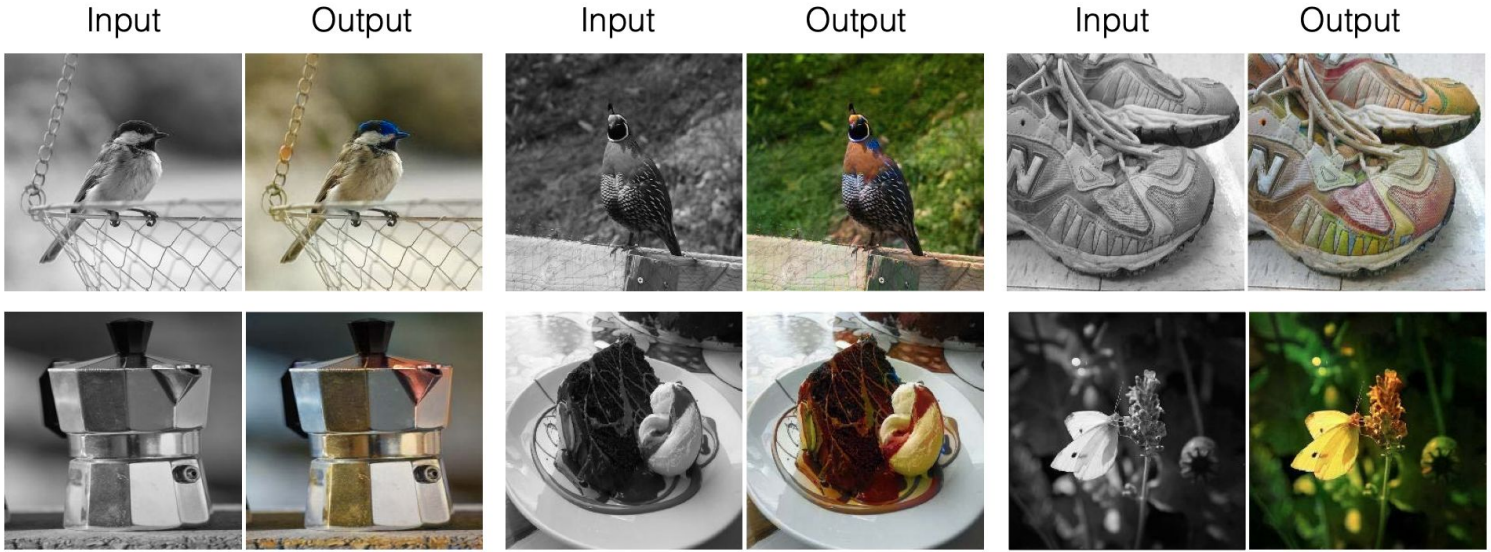
Rather than penalizing if output *image* looks fake, penalize if each overlapping *patch* in output looks fake

- Faster, fewer parameters
- More supervised observations
- Applies to arbitrarily large images

Slide credit: Phillip Isola

Image-to-Image Translation

- BW to color



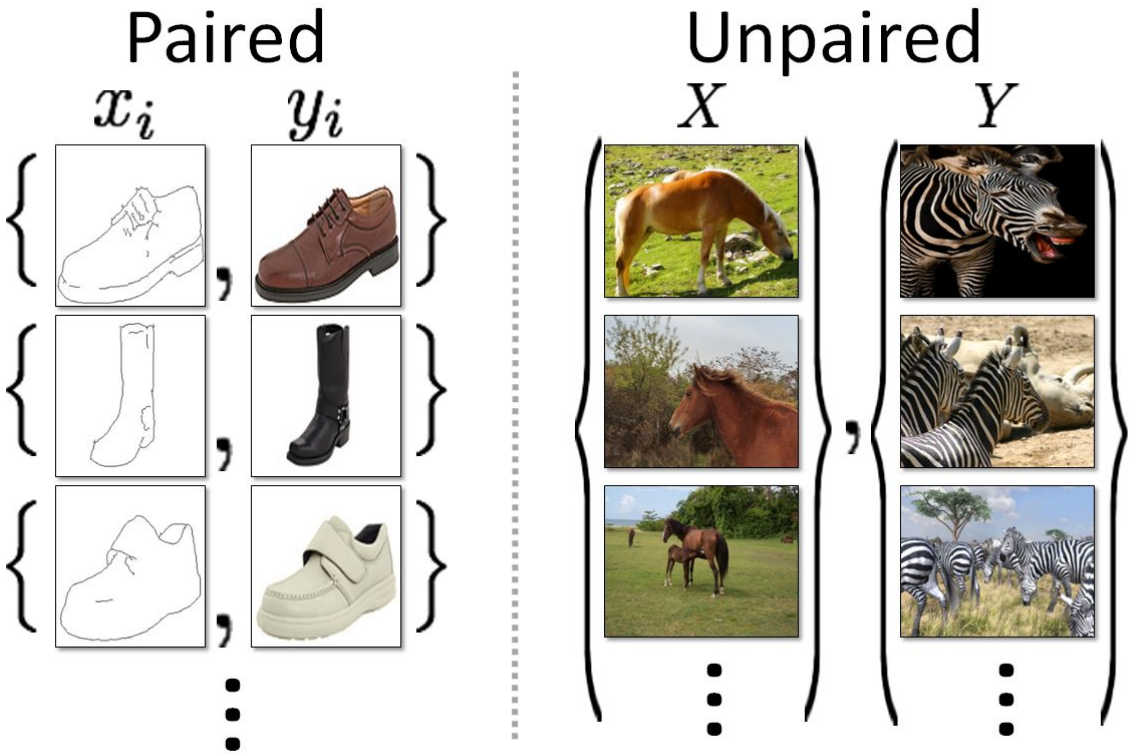
Slide credit: Phillip Isola

Edges to Images

Sketches to Images

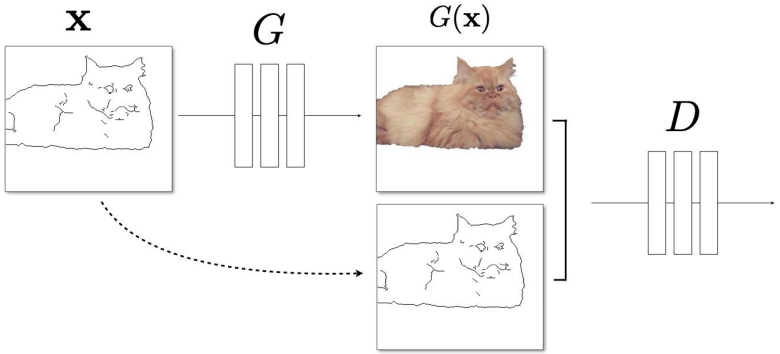
Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1125-1134).

Paired vs Unpaired data



Slide credit: Jun-Yan Zhu

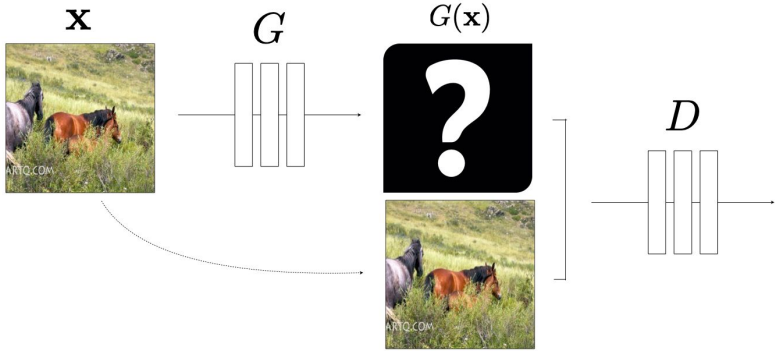
Paired vs Unpaired data



When paired data is available

Conditional GAN

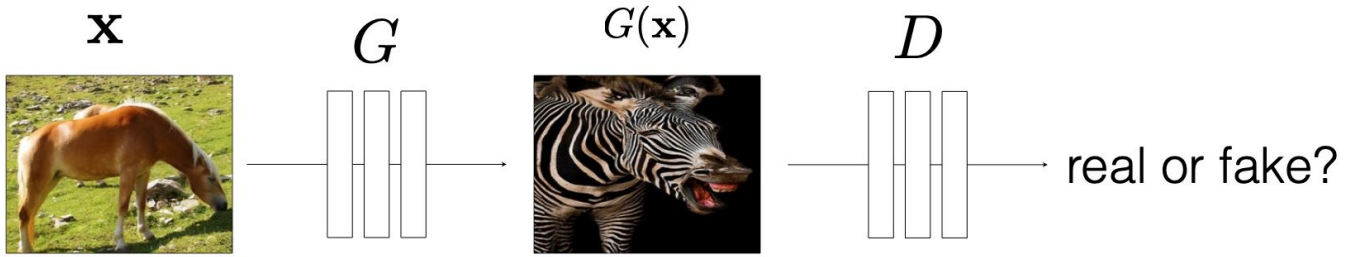
$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x}, \mathbf{y}))\;]$$



No Input/Output pairs!

Slide credit: Jun-Yan Zhu

Learning from unpaired data



$$\arg\min_G \max_D \; \mathbb{E}_{\mathbf{x},\mathbf{y}}[\; \log D(G(\mathbf{x})) \quad + \quad \log(1 - D(\mathbf{y}))\;]$$

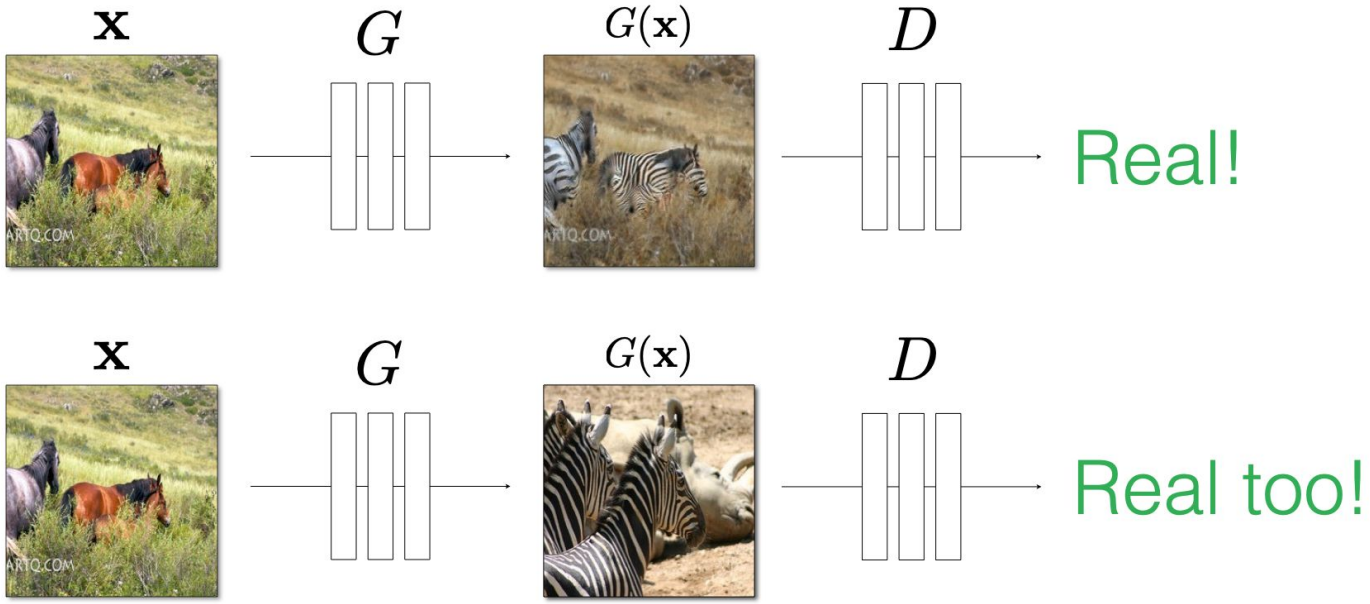
Usually loss functions check if output matches a target *instance*

GAN loss checks if output is part of an admissible *set*

Slide credit: Jun-Yan Zhu

Learning from unpaired data

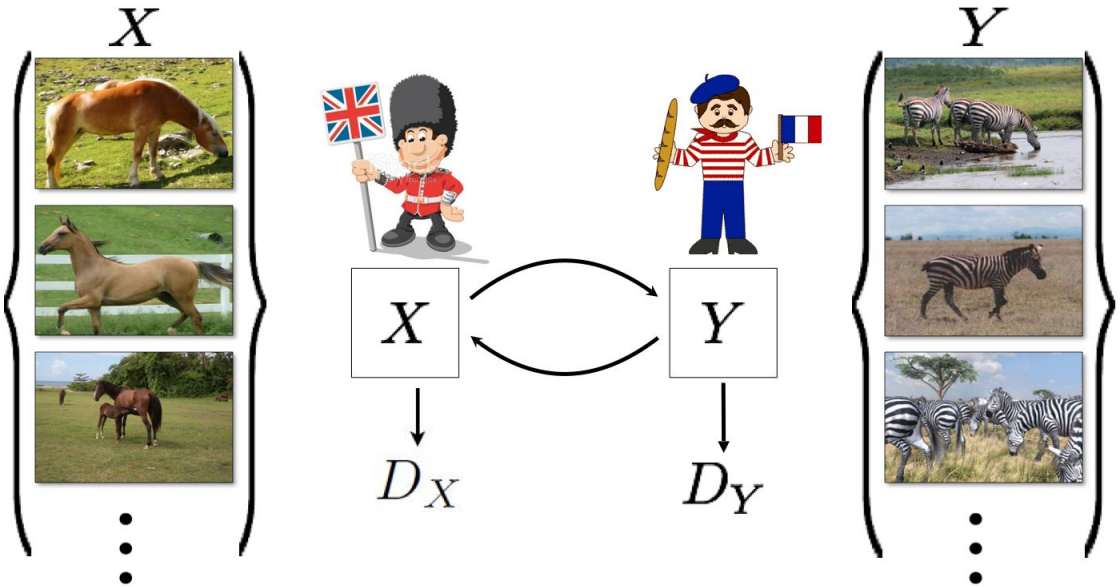
- There is nothing to force output to be compatible with input



Slide credit: Jun-Yan Zhu

Cycle-consistent Adversarial Networks

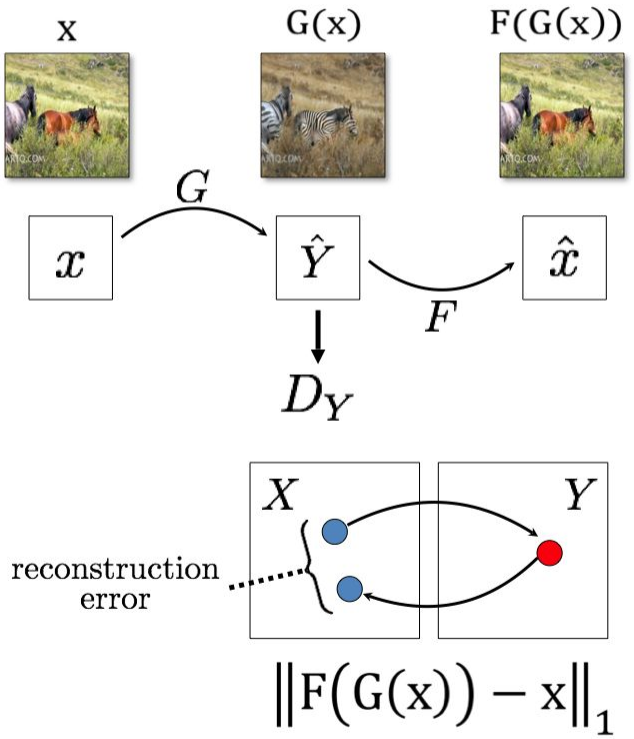
- Mapping between domains X and Y using an adversarial discriminator



- No constraint on input-output compatibility

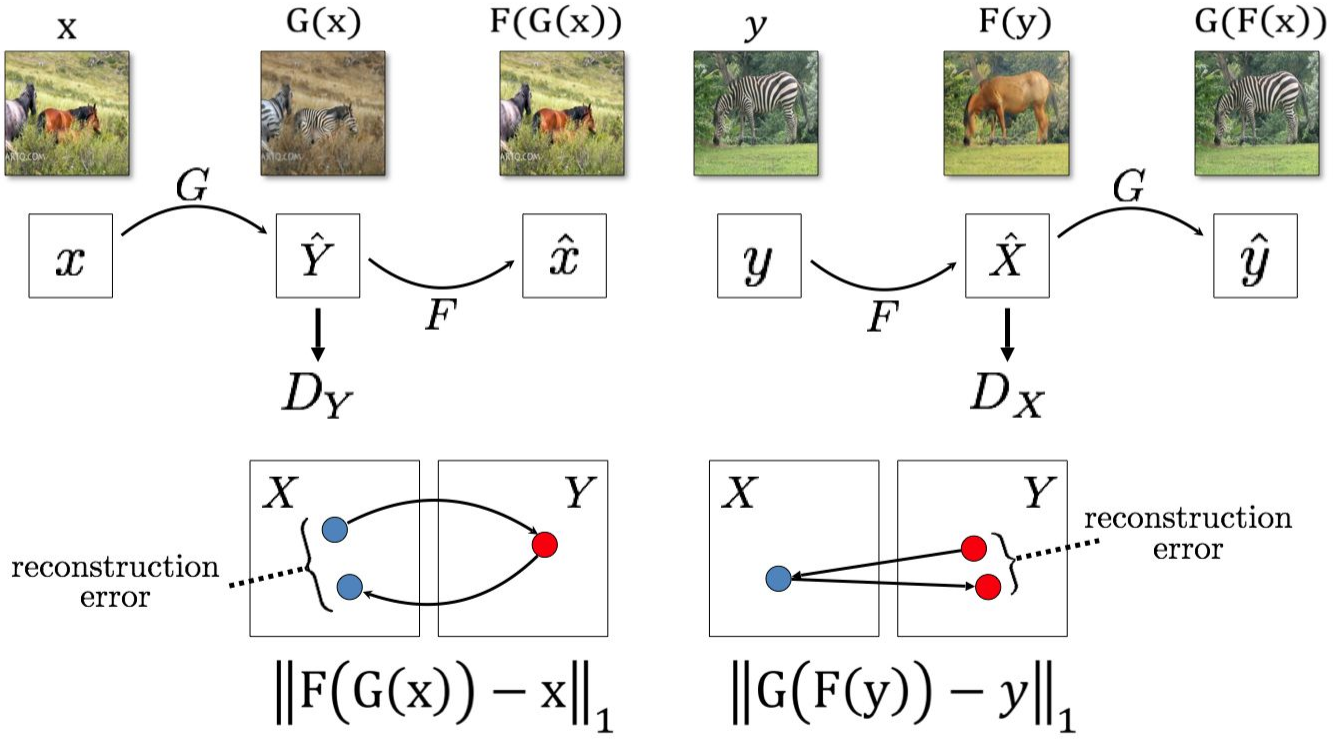
Slide credit: Jun-Yan Zhu

Cycle-consistency loss



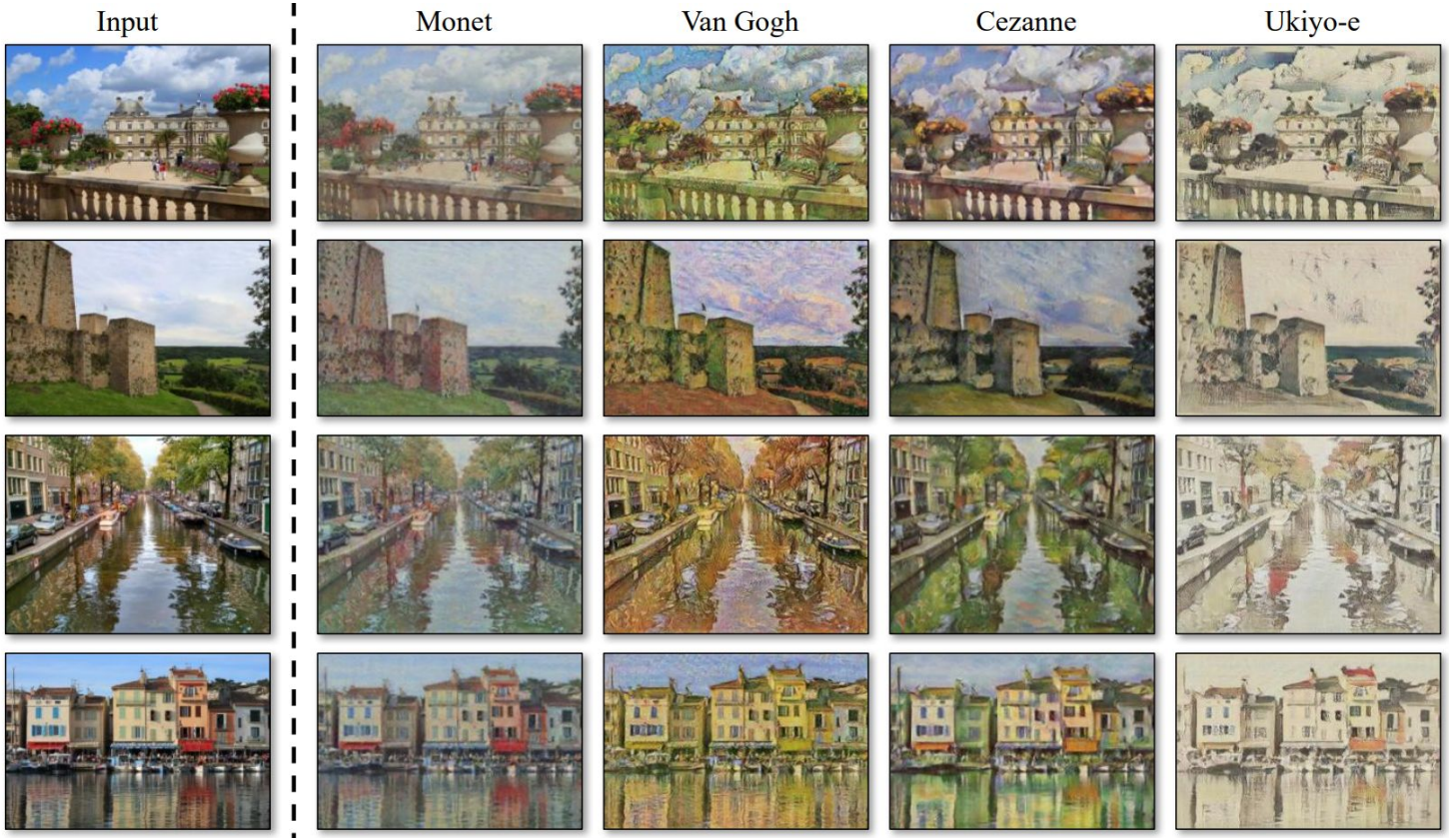
Slide credit: Jun-Yan Zhu

Cycle-consistency loss



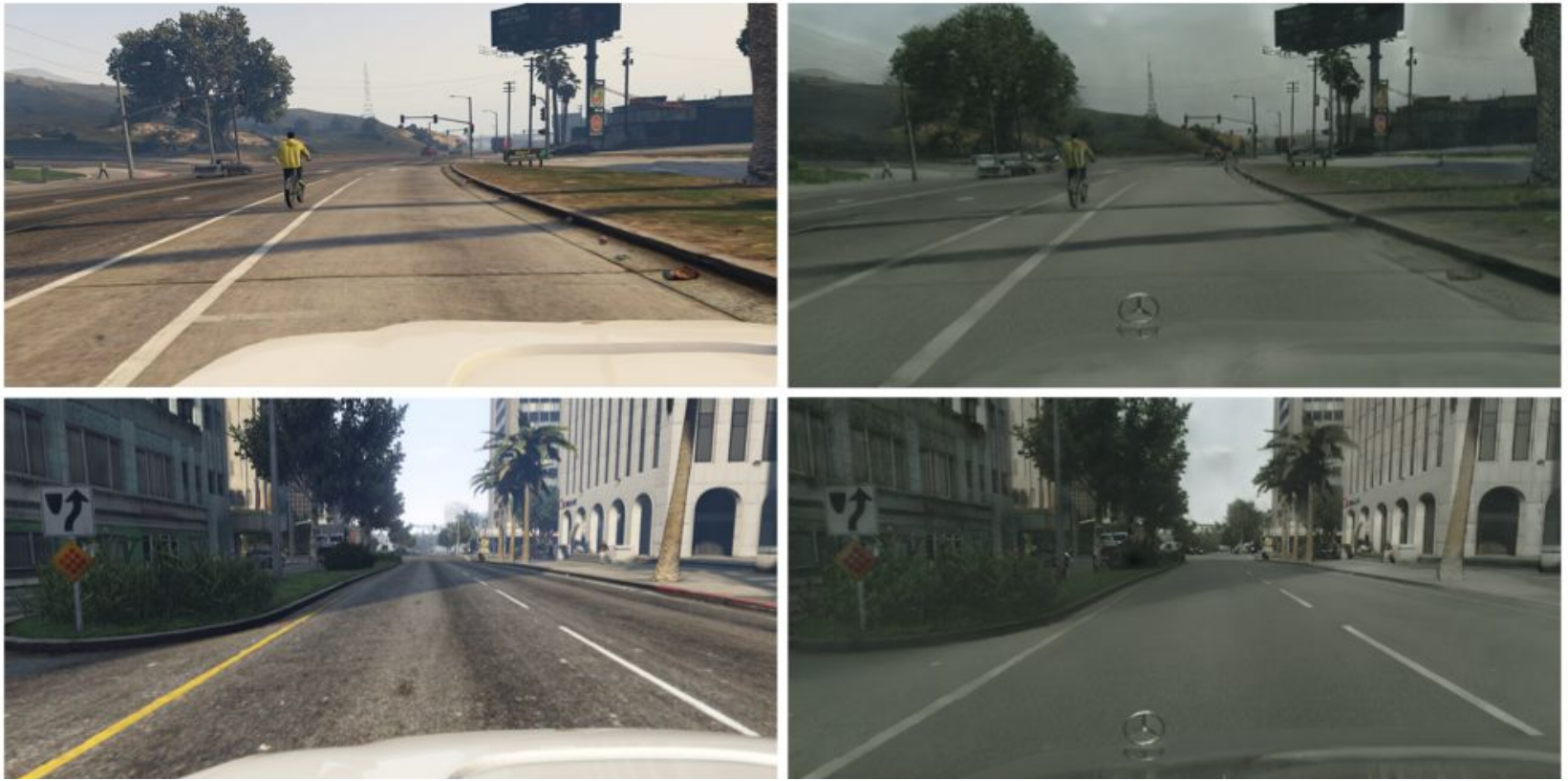
Slide credit: Jun-Yan Zhu

Style Transfer



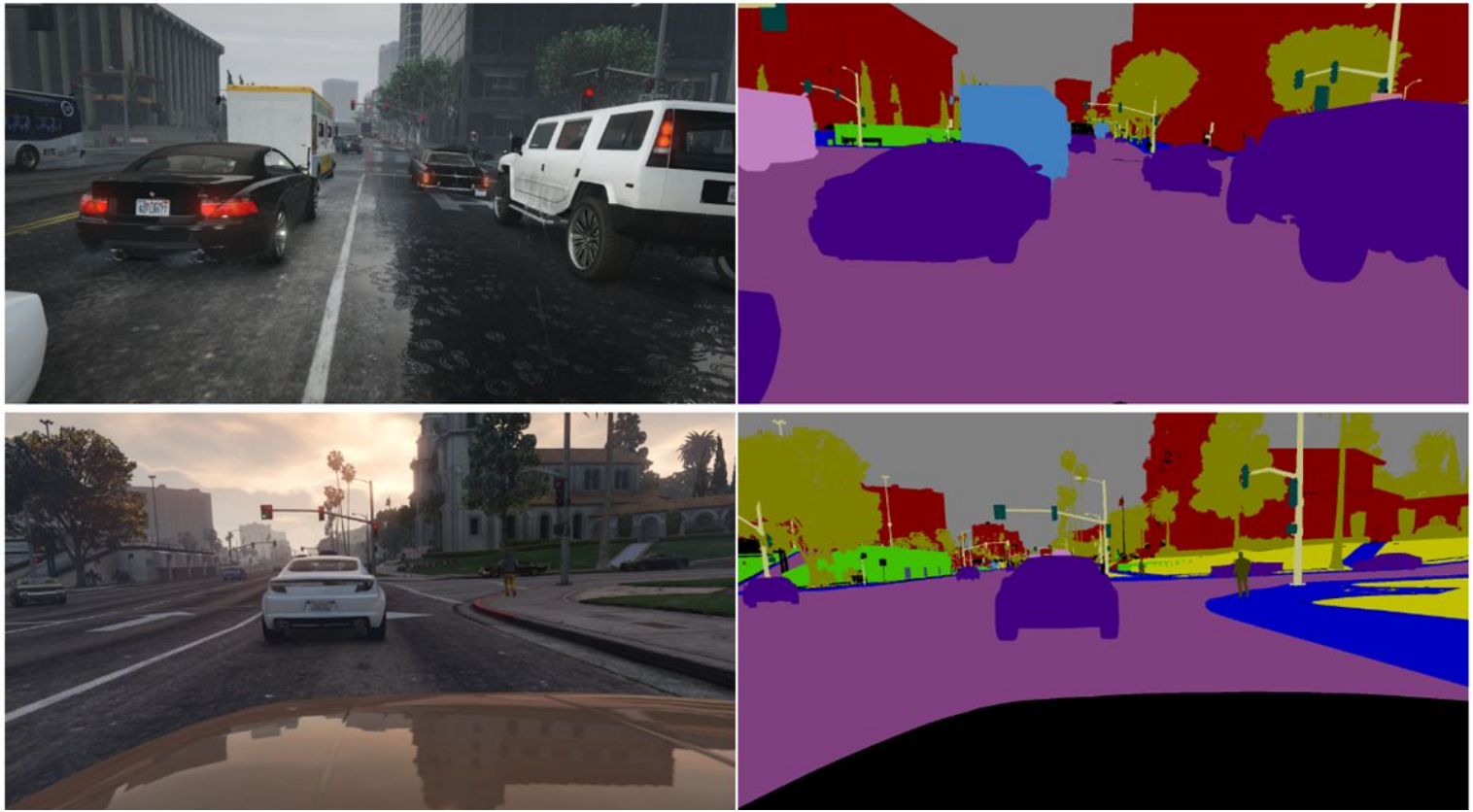
Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2223-2232).

CG2Real: GTA to Real streetview



Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2223-2232).

Segmentation



Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2223-2232).

Other applications of CycleGAN

Attribute Editing [Lu et al.]

Low-res Bald Bangs

Object Editing [Liang et al.]

Mask Input Output

Photo Enhancement [Ignatov et al.]

Input Enhanced image

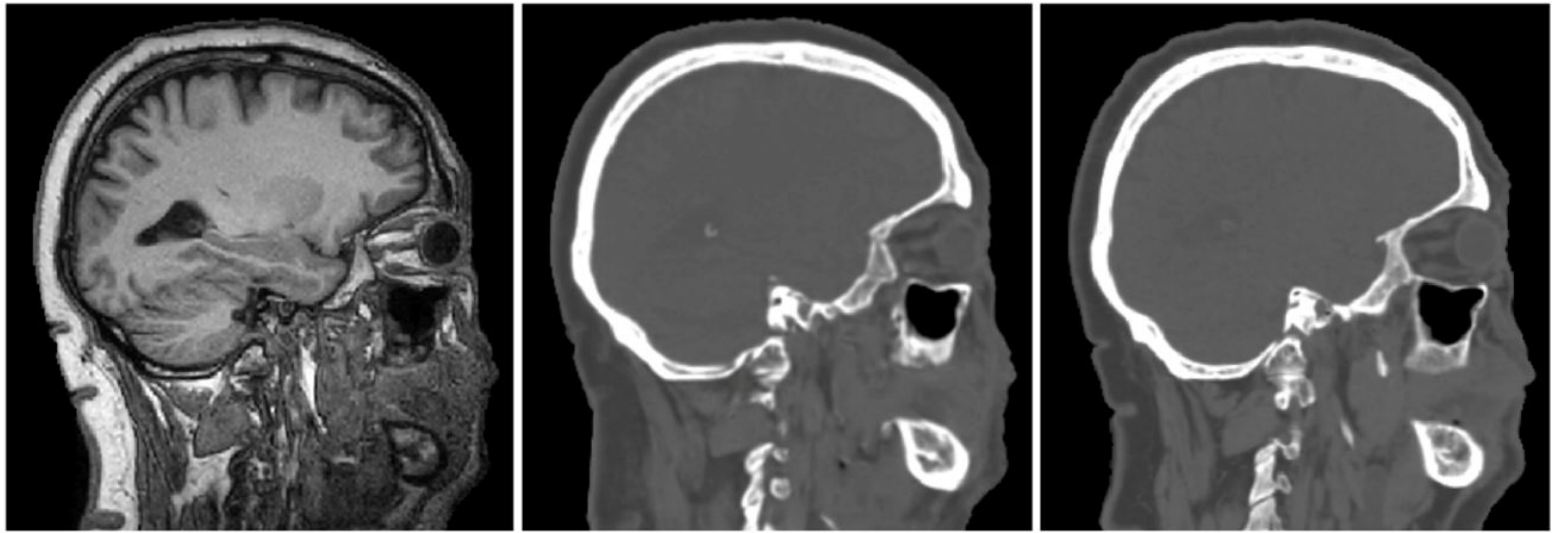
Data generation [Wang et al.]

samples by CycleWGAN

Slide credit: Jun-Yan Zhu

Medical Imaging

- MR to CT [Wolterink et al.]



Input MR | Generated CT | Ground Truth CT

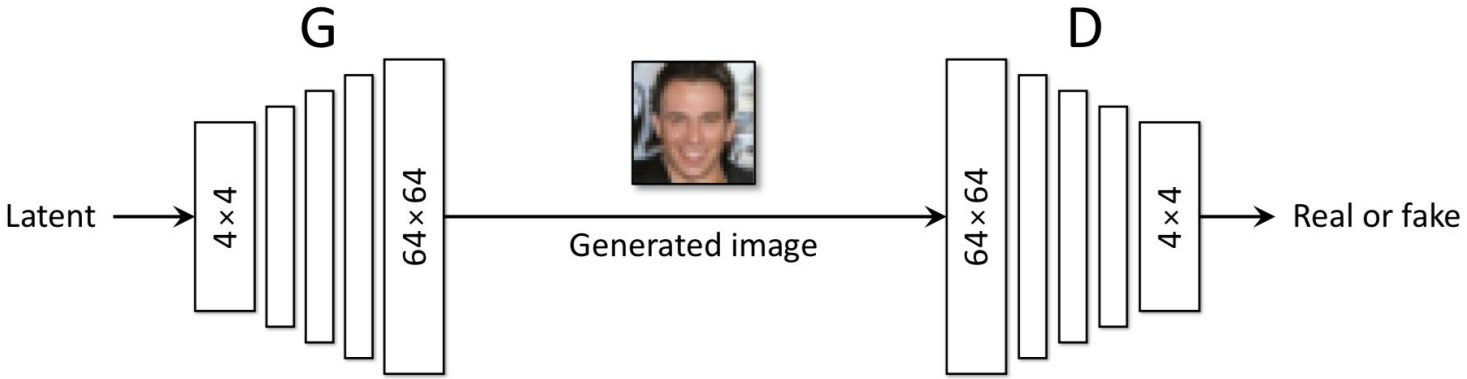
Slide credit: Jun-Yan Zhu

GANs

- Improved techniques for training GANs
  - Tips and Tricks
  - Learning interpretable latent representations
  - Principled methods with better training objectives
- Conditional Generation
  - Class conditioning
  - Conditioning on richer structures: Text, bounding boxes, images, etc.
- **Case study**
  - **SOTA image synthesis**
- Other applications

Generating images

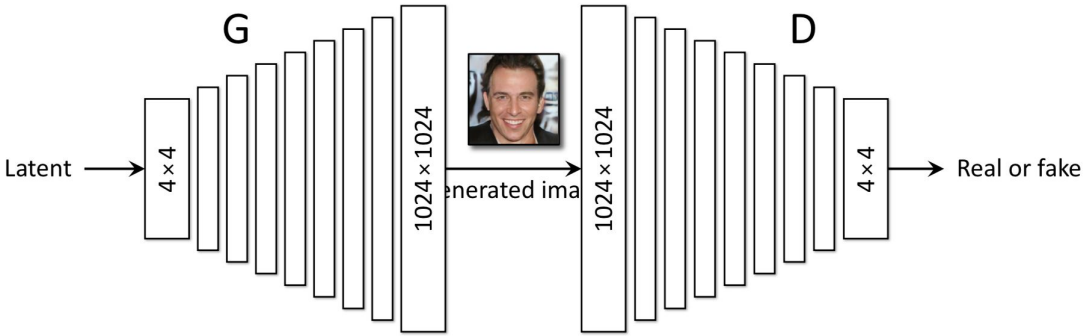
- GANs for generating low-resolution images



Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

Generating high-resolution images

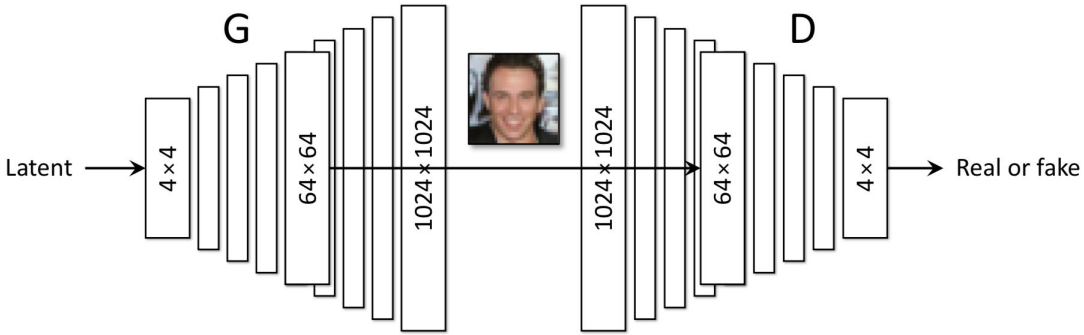
- Moving towards higher resolutions
- Optimizing for the high resolution image directly can be difficult



Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

Generating high-resolution images

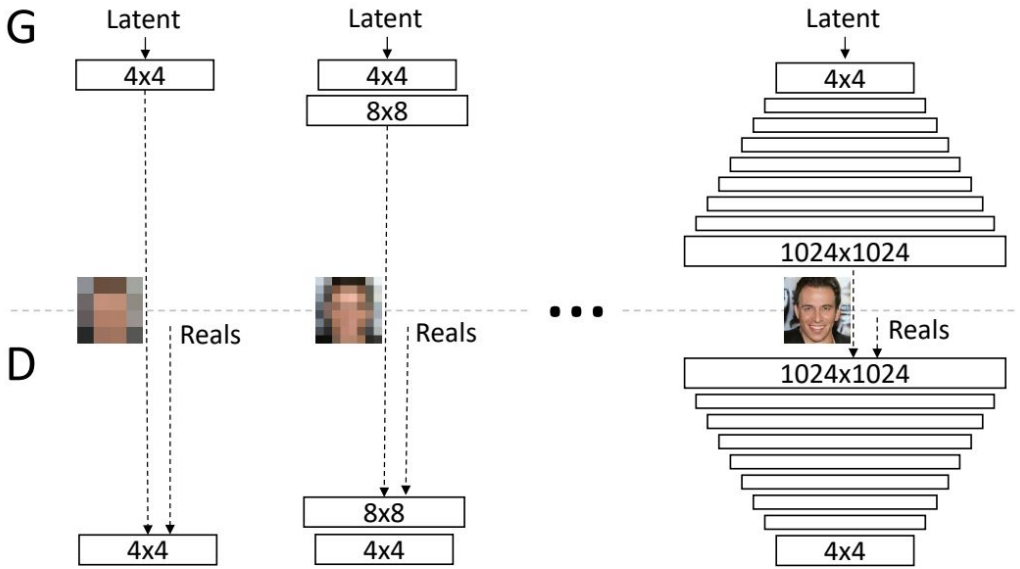
- Moving towards higher resolutions
- Optimizing for the high resolution image directly can be difficult
- Multi-scale approach!



Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

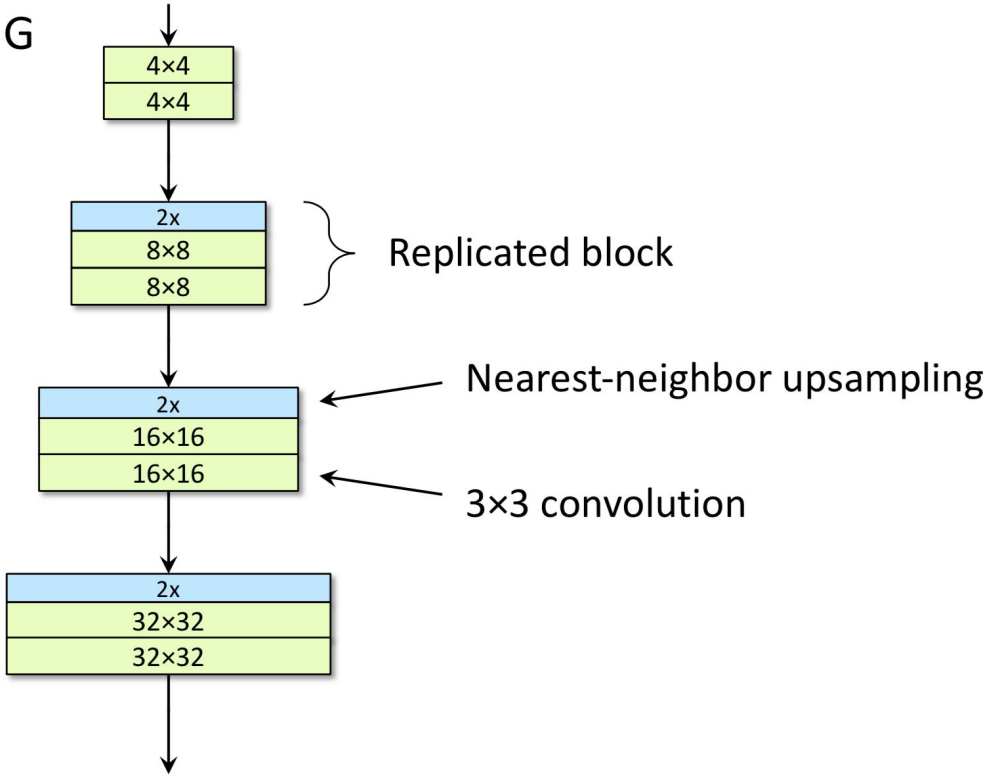
Generating high-resolution images

- Moving towards higher resolutions
- Optimizing for the high resolution image directly can be difficult
- Multi-scale approach!



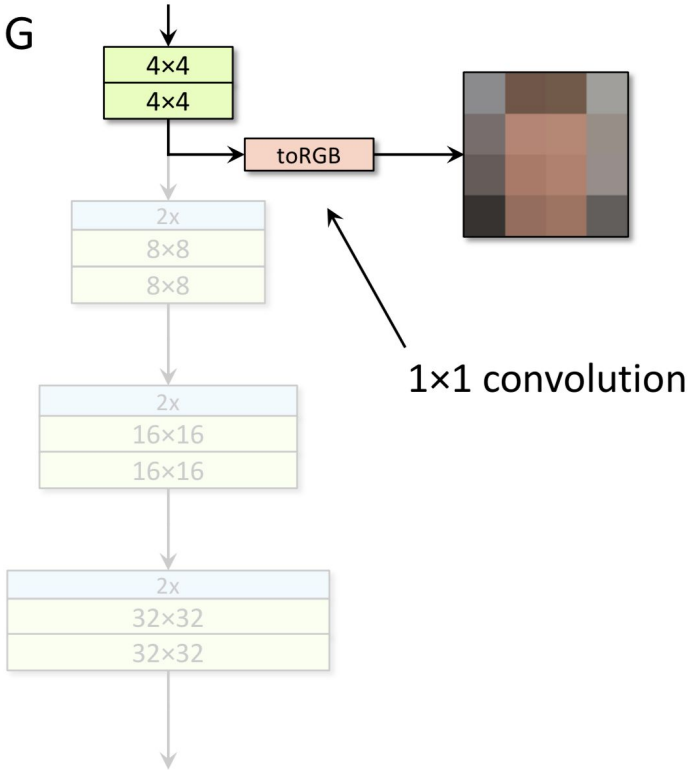
Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

Progressive GAN



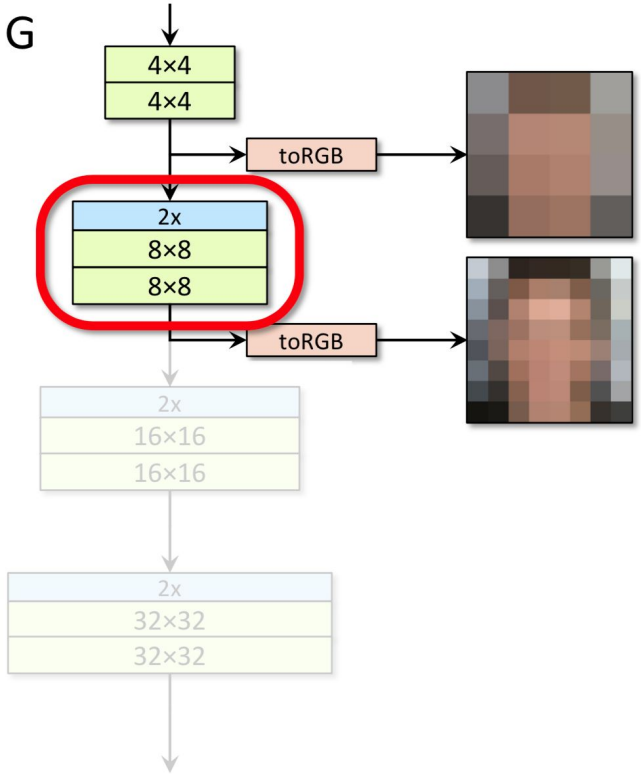
Slide credit: Tero Karras

Progressive GAN



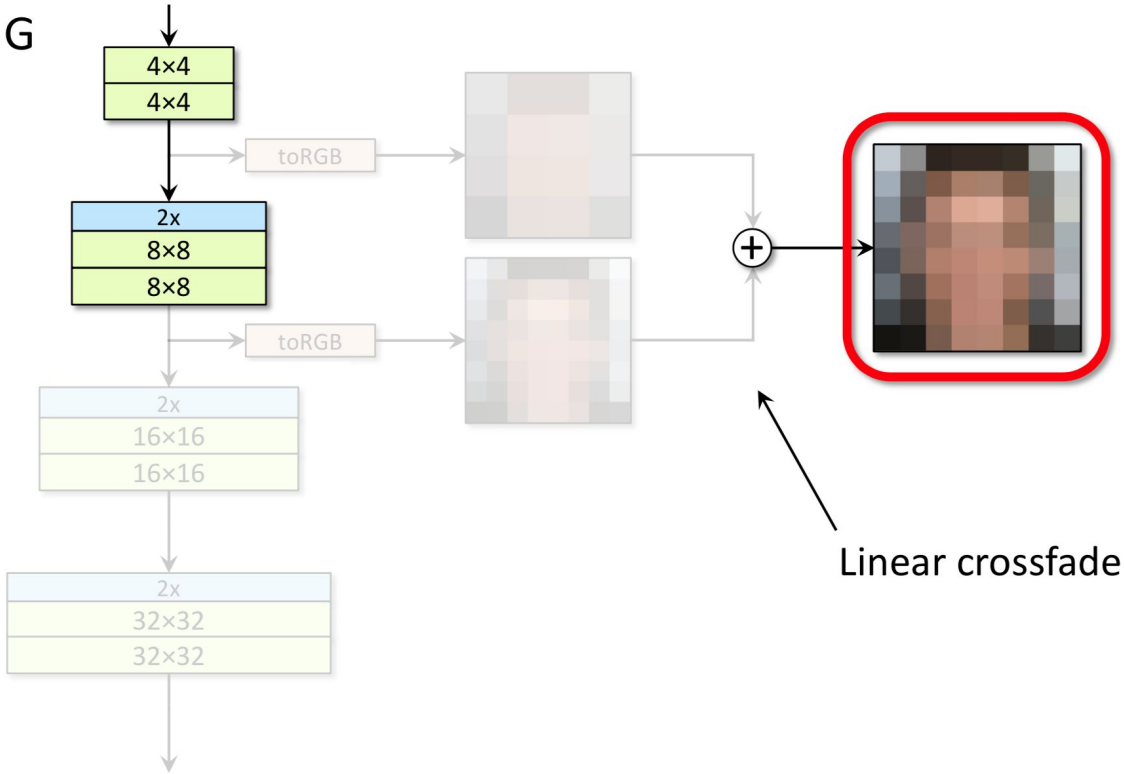
Slide credit: Tero Karras

Progressive GAN



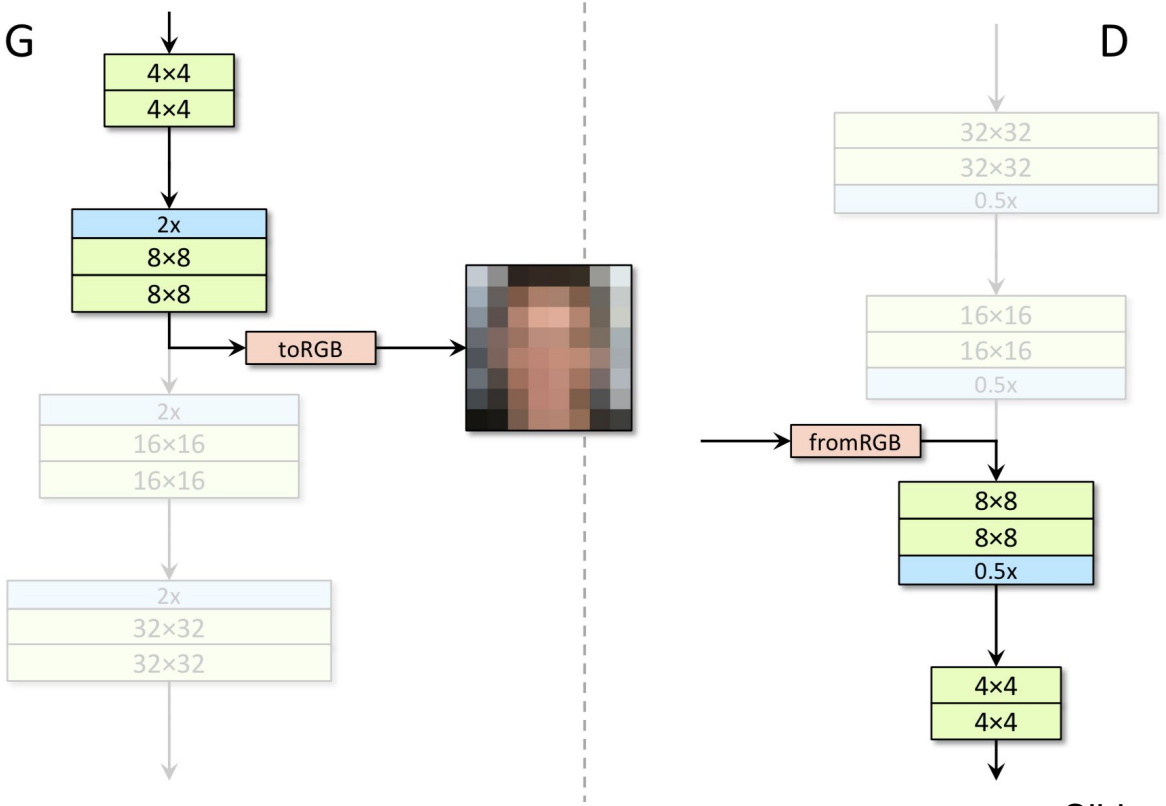
Slide credit: Tero Karras

Progressive GAN



Slide credit: Tero Karras

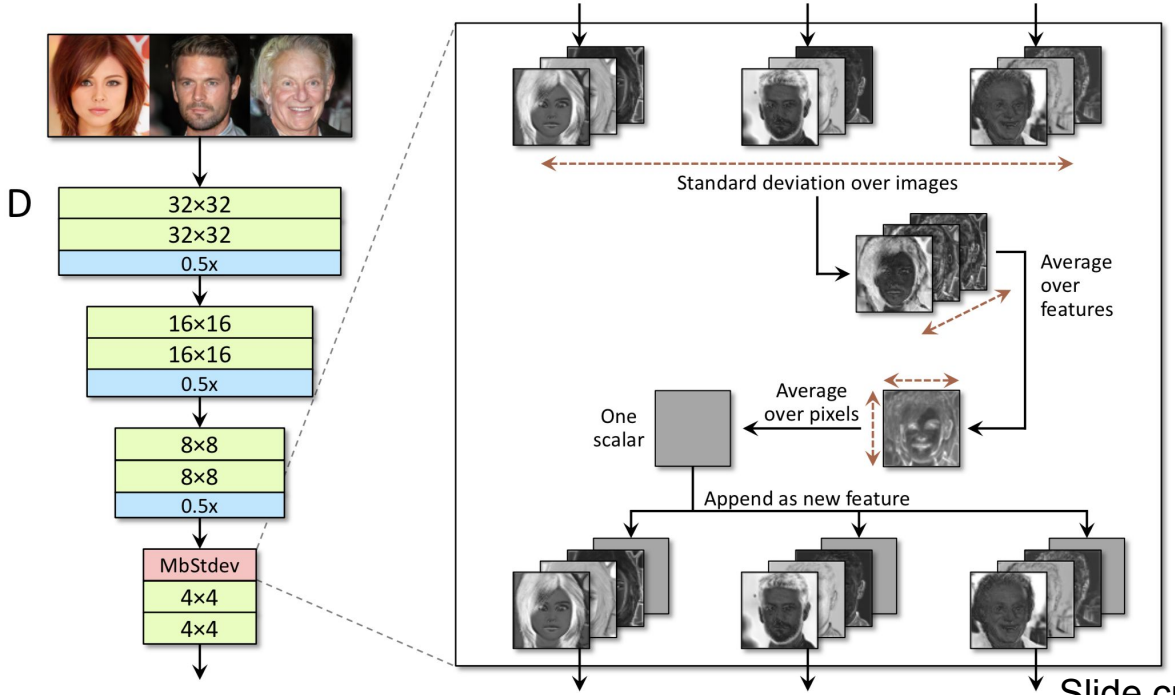
Progressive GAN



Slide credit: Tero Karras

Progressive GAN

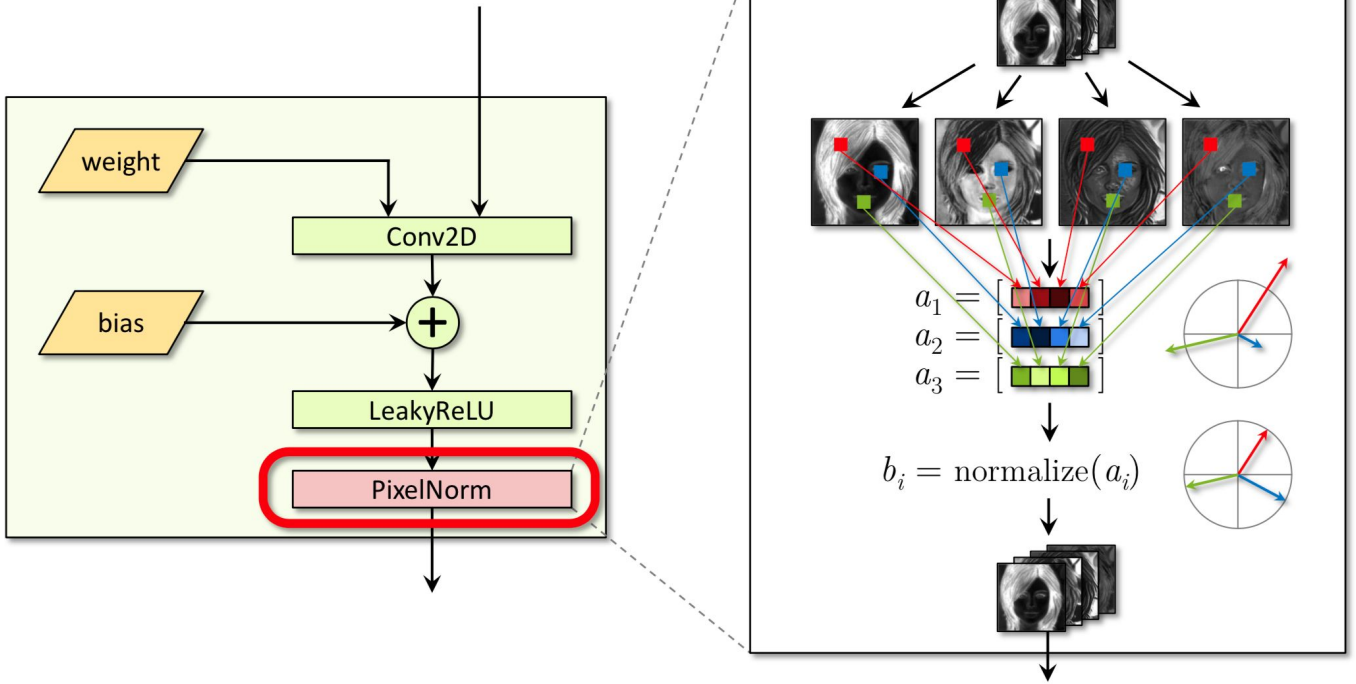
- Minibatch statistics are known to be important (Salimans et al.)
- Add minibatch variance as an additional feature



Slide credit: Tero Karras

Progressive GAN

- Pixel-wise normalization

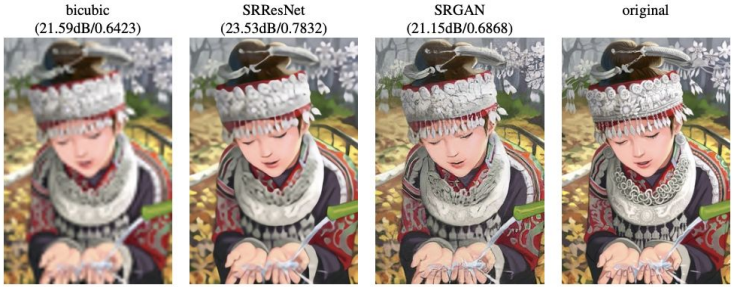


Slide credit: Tero Karras

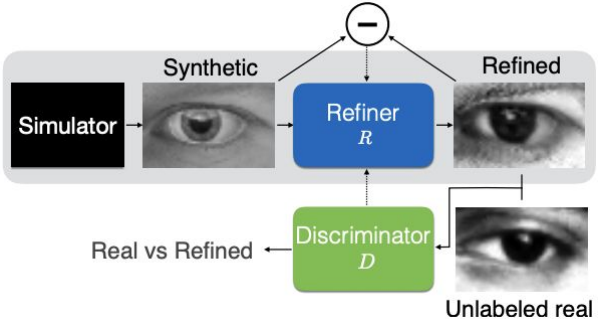
Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

Other GAN applications: Vision

Image super-resolution [Ledig et al., 2017]

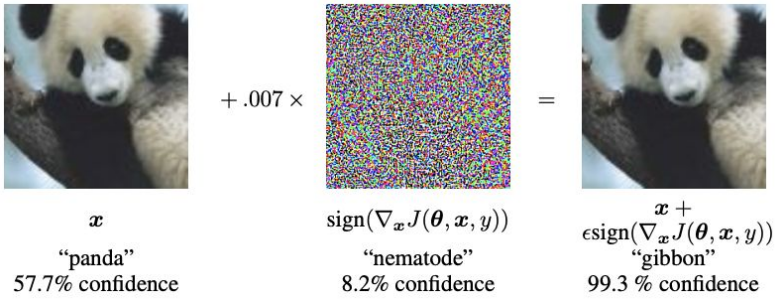


Domain Adaptation [Shrivastava et al., 2017]



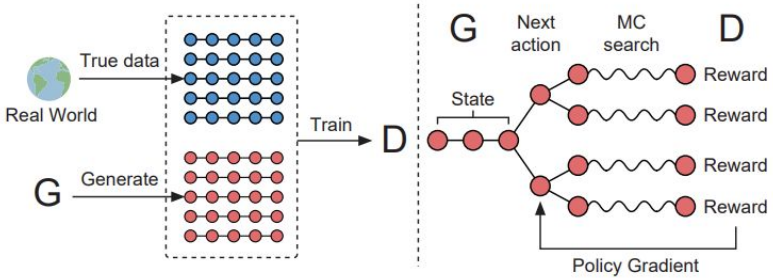
Video prediction [Matheiu et al., 2016]

Adversarial examples [Goodfellow et al., 2015]

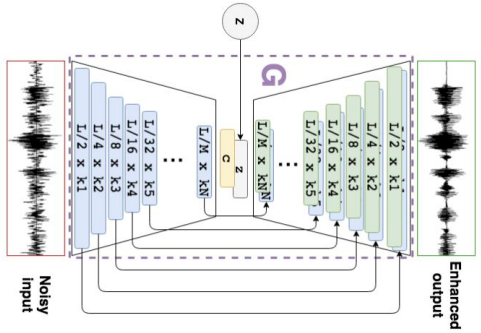


Other GAN applications: Sequence data

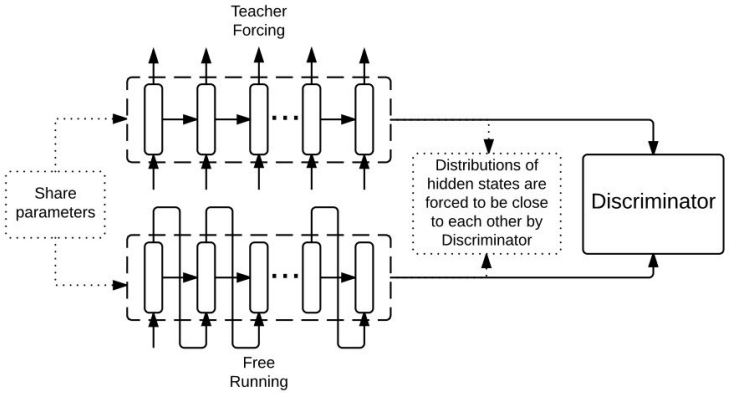
SeqGAN [Yu et al., 2017]



SEGAN [Pascal et al., 2017]



Professor-forcing [Goyal et al., 2016]



Text style transfer [Shen et al., 2017]

Sentiment transfer from negative to positive
I would recommend find another place. I would recommend this place again!
Do not like it at all! All in all, it's great!
Sentiment transfer from positive to negative
Really good food that is fast and healthy. Really bland and bad, and terrible.
You will notice that I have given this restaurant five stars. You should give this place zero stars.

Overview

GANs

- Improved techniques for training GANs
- Conditional Generation
- Recent progress/SOTA image synthesis
- Other applications

VAEs

- **Improved techniques for training VAEs**
- **Conditional Generation**
- **Sequence data**
- **Discrete latent variables**
- **Other applications**

VAEs

- **Improved techniques for training VAEs**
  - **Importance-weighted Autoencoders (IWAE)**
  - Autoregressive Flow
  - VAE-GAN hybrids
- Conditional Generation
  - Class conditioning
  - Attribute conditioning
  - Semi-supervised learning
- Case study
  - Sequence data
  - Discrete latent variables
- Other applications

Limitations of vanilla VAE

- We want to learn flexible approximate posteriors to have a rich probability model
- Vanilla VAE (Auto-Encoding Variational Bayes) uses a single sample of the latent variable for each datapoint when estimating the gradient
- Importance-weighted Autoencoders (IWAE, ”eye-way”) learns a more flexible posterior by averaging multiple samples of the posterior scaled according to importance weights

Slide credit: David Duvenaud

Importance-weighted Autoencoders

- Improves the VAE by considering a tighter lower bound on $p(\mathbf{x}_i)$
- Consider an importance sampling estimate of $p(\mathbf{x}_i)$

$$\log p(\mathbf{x}_i) = \log \int p(\mathbf{x}_i|\mathbf{z};\theta)p(\mathbf{z})d\mathbf{z} = \log \int p(\mathbf{x}_i|\mathbf{z};\theta)p(\mathbf{z})\frac{q(\mathbf{z}|\mathbf{x}_i;\phi)}{q(\mathbf{z}|\mathbf{x}_i;\phi)}d\mathbf{z}$$

$$= \log \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i;\phi)}\left[\frac{p(\mathbf{x}_i|\mathbf{z};\theta)p(\mathbf{z})}{q(\mathbf{z}|\mathbf{x}_i;\phi)}\right] \approx \log \frac{1}{k}\sum_{m=1}^{k}\frac{p(\mathbf{x}_i|\mathbf{z}^{(m)};\theta)p(\mathbf{z}^{(m)})}{q(\mathbf{z}^{(m)}|\mathbf{x}_i;\phi)}$$

Slide credit: Daniel Hernandez–Lobato

Importance-weighted Autoencoders

- Improves the VAE by considering a tighter lower bound on $p(\mathbf{x}_i)$
- Consider an importance sampling estimate of $p(\mathbf{x}_i)$

$$\log p(\mathbf{x}_i) = \log \int p(\mathbf{x}_i|\mathbf{z};\theta)p(\mathbf{z})d\mathbf{z} = \log \int p(\mathbf{x}_i|\mathbf{z};\theta)p(\mathbf{z})\frac{q(\mathbf{z}|\mathbf{x}_i;\phi)}{q(\mathbf{z}|\mathbf{x}_i;\phi)}d\mathbf{z}$$

$$= \log \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i;\phi)}\left[\frac{p(\mathbf{x}_i|\mathbf{z};\theta)p(\mathbf{z})}{q(\mathbf{z}|\mathbf{x}_i;\phi)}\right] \approx \log \frac{1}{k}\sum_{m=1}^{k}\frac{p(\mathbf{x}_i|\mathbf{z}^{(m)};\theta)p(\mathbf{z}^{(m)})}{q(\mathbf{z}^{(m)}|\mathbf{x}_i;\phi)}$$

- On expectation, this estimate is a lower bound

$$\mathcal{L}_k(\mathbf{x}_i;\theta,\phi) = \mathbb{E}\left[\log \frac{1}{k}\sum_{m=1}^{k} w_m\right] \leq \log \mathbb{E}\left[\frac{1}{k}\sum_{m=1}^{k} w_m\right] = \log p(\mathbf{x}_i)$$

Slide credit: Daniel Hernandez–Lobato

Importance-weighted Autoencoders

- If k = 1, we obtain the VAE
- k > 1 can only improve the bound
- Optimization is done as in the VAE

$$\mathcal{L}_k(\mathbf{x}_i; \theta, \phi) = \mathbb{E}\left[\log \frac{1}{k}\sum_{m=1}^{k} w_m\right] \leq \log \mathbb{E}\left[\frac{1}{k}\sum_{m=1}^{k} w_m\right] = \log p(\mathbf{x}_i)$$

Slide credit: Daniel Hernandez–Lobato

VAE vs IWAE

- Performance of IWAE improves with larger k

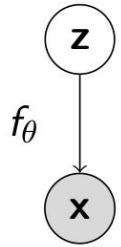
		MNIST				OMNIGLOT			
		VAE		IWAE		VAE		IWAE	
# stoch. layers	k	NLL	active units	NLL	active units	NLL	active units	NLL	active units
1	1	86.76	19	86.76	19	108.11	28	108.11	28
	5	86.47	20	85.54	22	107.62	28	106.12	34
	50	86.35	20	84.78	25	107.80	28	104.67	41
2	1	85.33	16+5	85.33	16+5	107.58	28+4	107.56	30+5
	5	85.01	17+5	83.89	21+5	106.31	30+5	104.79	38+6
	50	84.78	17+5	82.90	26+7	106.30	30+5	103.38	44+7

Slide credit: Daniel Hernandez–Lobato

VAEs

- **Improved techniques for training VAEs**
  - Importance-weighted Autoencoders (IWAE)
  - **Autoregressive Flow**
  - VAE-GAN hybrids
- Conditional Generation
  - Class conditioning
  - Attribute conditioning
  - Semi-supervised learning
- Case study
  - Sequence data
  - Discrete latent variables
- Other applications

Autoregressive Flow



- Flow model, the marginal likelihood p(x) is given by

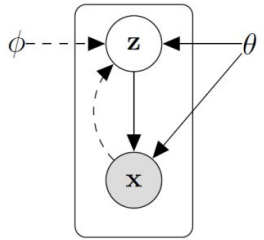
$$p_X(\mathbf{x};\theta) = p_Z\left(\mathbf{f}_\theta^{-1}(\mathbf{x})\right)\left|\det\left(\frac{\partial \mathbf{f}_\theta^{-1}(\mathbf{x})}{\partial \mathbf{x}}\right)\right|$$

  where $p_Z(z)$ is typically simple (e.g., a Gaussian).

- More complex prior ?
- Prior $p_Z(z)$ can be autoregressive $p_Z(\mathbf{z}) = \prod_i p(z_i \mid z_1, \cdots, z_{i-1})$.
- Autoregressive models are flows.

Slide credit: Stefan Ermano

VAE + Flow model



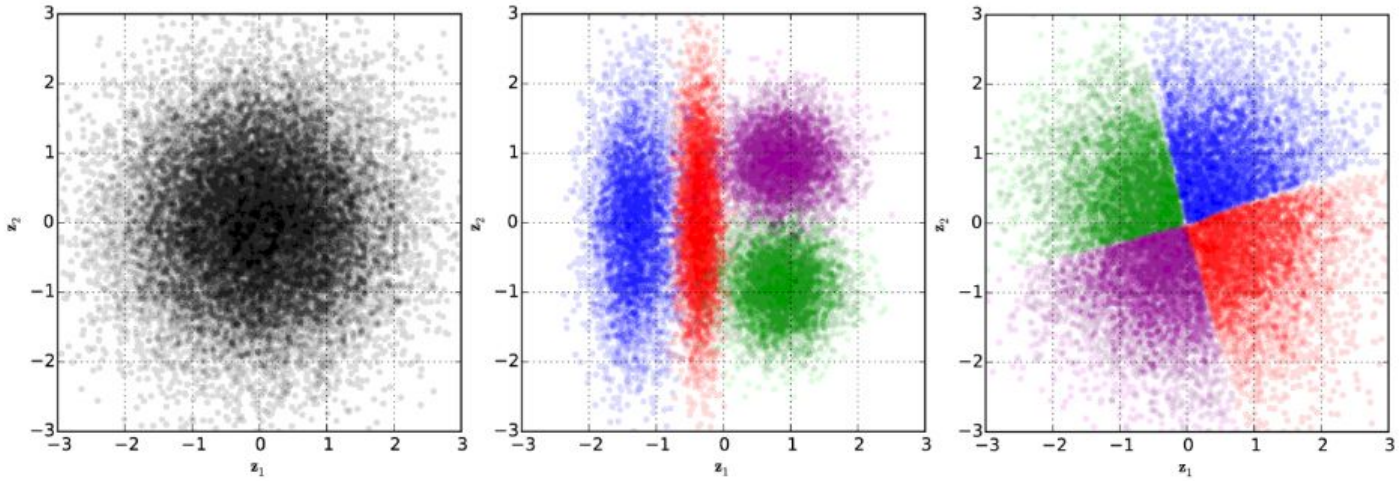
$$\log p(\mathbf{x};\theta) \geq \sum_{\mathbf{z}} q(\mathbf{z}|\mathbf{x};\phi)\log p(\mathbf{z},\mathbf{x};\theta) + H(q(\mathbf{z}|\mathbf{x};\phi)) = \underbrace{\mathcal{L}(\mathbf{x};\theta,\phi)}_{\text{ELBO}}$$

$$= \mathcal{L}(\mathbf{x};\theta,\phi) + \underbrace{D_{KL}(q(\mathbf{z}\mid\mathbf{x};\phi)\|p(\mathbf{z}|\mathbf{x};\theta))}_{\text{Gap between true log-likelihood and ELBO}}$$

- $q(\mathbf{z}|\mathbf{x};\phi)$ is often too simple (Gaussian) compared to the true posterior $p(\mathbf{z}|\mathbf{x};\theta)$, hence ELBO bound is loose
- Idea: Make posterior more flexible: $\mathbf{z}' \sim q(\mathbf{z}'|\mathbf{x};\phi)$, $\mathbf{z} = f_{\phi'}(\mathbf{z})$ for an invertible $f_{\phi'}$
- Still easy to sample from, and can evaluate density

Slide credit: Stefan Ermano

VAE + Flow model



(a) Prior distribution (b) Posteriors in standard VAE (c) Posteriors in VAE with IAF

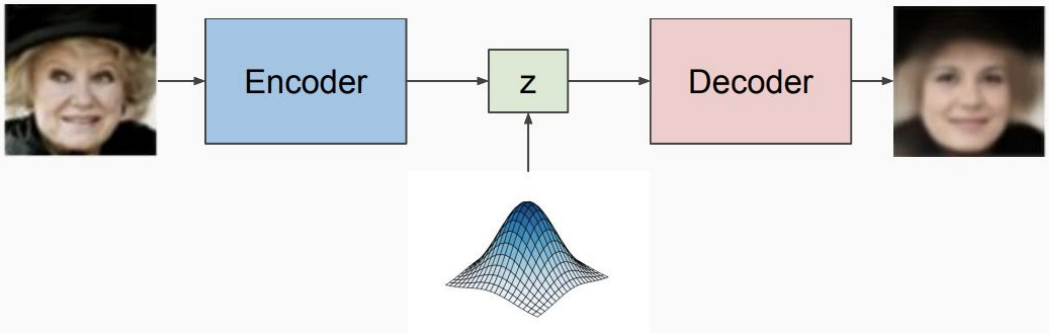
Posterior approximation is more flexible, hence we can get tighter ELBO (closer to true log-likelihood).

Slide credit: Stefan Ermano

VAEs

- **Improved techniques for training VAEs**
  - Importance-weighted Autoencoders (IWAE)
  - Autoregressive Flow
  - **VAE-GAN hybrids**
- Conditional Generation
  - Class conditioning
  - Attribute conditioning
  - Semi-supervised learning
- Case study
  - Sequence data
  - Discrete latent variables
- Other applications

VAEs and GANs

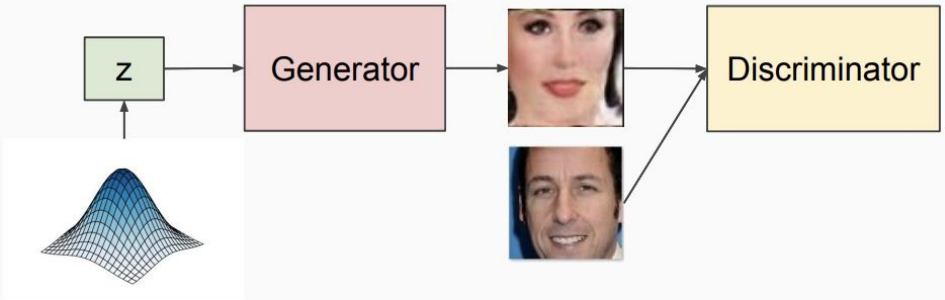


VAE

✓: Given an X **easy** to find z.
✓: Interpretable probability P(X)

X: Usually outputs blurry Images

GAN



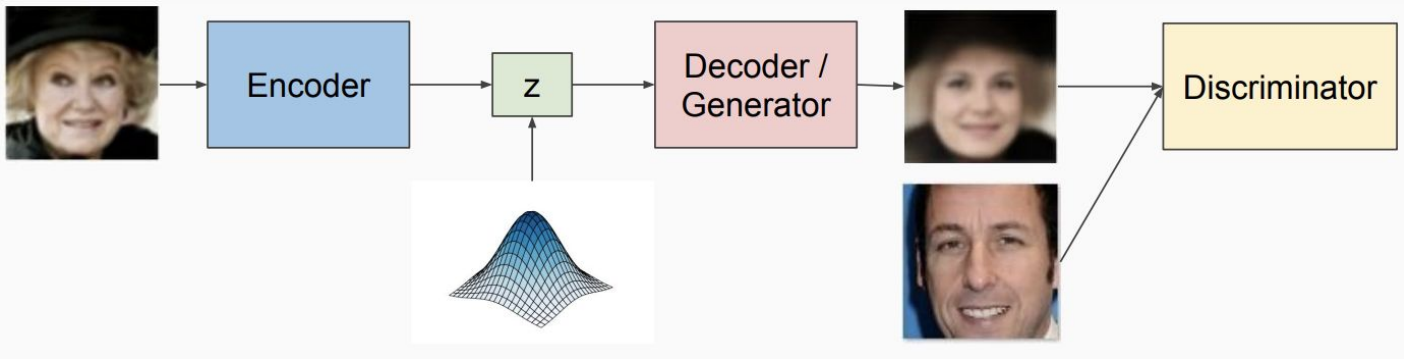
✓: Very sharp images

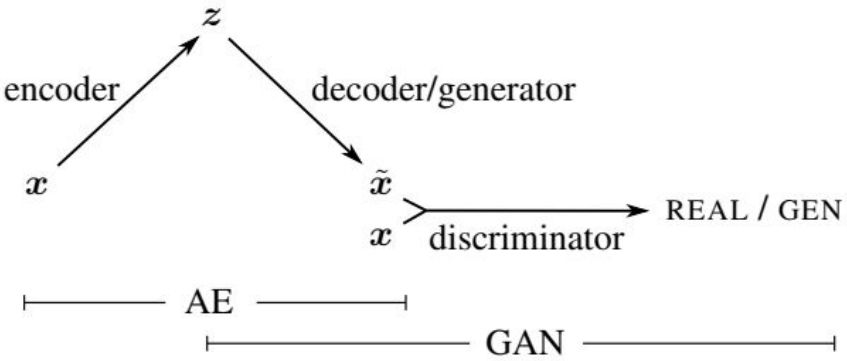
X: Given an X **difficult** to find z. (Need to backprop.)

✓/X: No explicit P(X).

Slide credit: Raymond Yeh, Junting Lou, Teck-Yian Lim

VAE + GAN (Best of both models)





KL divergence L2 difference

$$\mathcal{L} = \mathcal{L}_{\text{prior}} + \mathcal{L}_{\text{llike}}^{\text{Dis}_l} + \mathcal{L}_{\text{GAN}}$$

Slide credit: Raymond Yeh, Junting Lou, Teck-Yian Lim

VAE + GAN model

Larsen, A. B. L., Sønderby, S. K., Larochelle, H., & Winther, O. (2015). Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.093*

VAEs

- Improved techniques for training VAEs
  - Importance-weighted Autoencoders (IWAE)
  - Autoregressive Flow
  - VAE-GAN hybrids
- **Conditional Generation**
  - Class conditioning
  - Attribute conditioning
  - Semi-supervised learning
- Case study
  - Sequence data
  - Discrete latent variables
- Other applications

Conditional VAE

- What if we have labels? (e.g. digit labels or attributes) Or other inputs we wish to condition on (Y).
- None of the derivation changes.
- Replace P(X|z) with P(X|z,Y).
- Replace Q(z|X) with Q(z|X,Y).
- Go through the same KL divergence procedure, to get the same lower bound.

Slide credit: Raymond Yeh, Junting Lou, Teck-Yian Lim

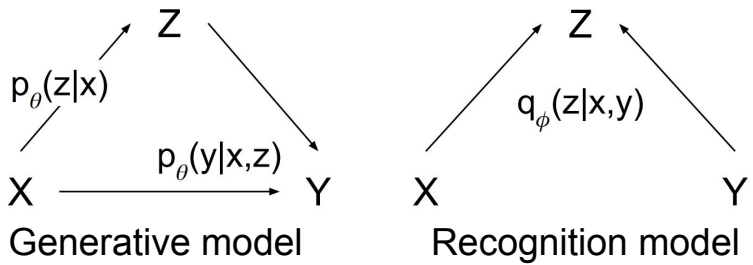
Conditional VAE

- VAE lower bound

$$\log p_\theta(\mathbf{x}) \geq -KL\left(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z})\right) + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right]$$

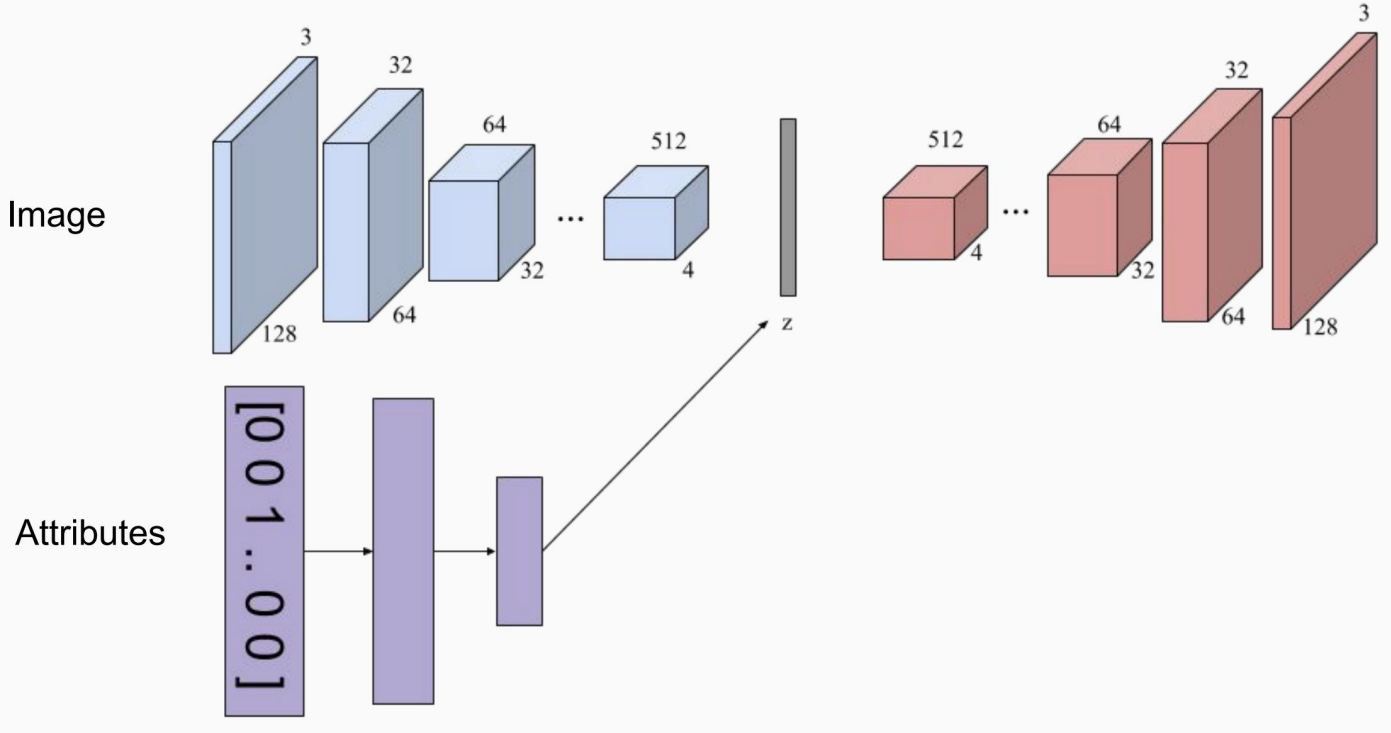
- CVAE lower bound

$$\log p_\theta(\mathbf{y}|\mathbf{x}) \geq -KL\left(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})\|p_\theta(\mathbf{z}|\mathbf{x})\right) + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log p_\theta(\mathbf{y}|\mathbf{x},\mathbf{z})\right]$$



Yan, X., Yang, J., Sohn, K., & Lee, H. (2016, October). Attribute2image: Conditional image generation from visual attributes. In *European Conference on Computer Vision* (pp. 776-791). Springer, Cham.

Common architecture for CVAE



Slide credit: Raymond Yeh, Junting Lou, Teck-Yian Lim

Semi-supervised VAE

- Learn conditional VAE from partially labeled data
- When label is observed, we can use the conditional VAE objective
- When label is missing, treat it as a latent variable over which we perform posterior inference
- Training objective is a sum of the above objectives

Semi-supervised VAE

- When label is observed (CVAE objective)

$$\log p_\theta(\mathbf{x}, y) \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},y)} \left[\log p_\theta(\mathbf{x}|y,\mathbf{z}) + \log p_\theta(y) + \log p(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x},y)\right] = -\mathcal{L}(\mathbf{x},y)$$

- When label is unobserved

$$\begin{aligned}\log p_\theta(\mathbf{x}) &\geq \mathbb{E}_{q_\phi(y,\mathbf{z}|\mathbf{x})} \left[\log p_\theta(\mathbf{x}|y,\mathbf{z}) + \log p_\theta(y) + \log p(\mathbf{z}) - \log q_\phi(y,\mathbf{z}|\mathbf{x})\right] \\ &= \sum\nolimits_y q_\phi(y|\mathbf{x})(-\mathcal{L}(\mathbf{x},y)) + \mathcal{H}(q_\phi(y|\mathbf{x})) = -\mathcal{U}(\mathbf{x}).\end{aligned}$$

- Discriminative term for labeled data: encoder used as classifier, otherwise encoder is only trained from unlabeled data

$$\mathcal{J} = \sum_{(x)\sim\tilde{p}_u} \mathcal{U}(x) + \sum_{(x,y)\sim\tilde{p}_l} \mathcal{L}(x,y) + \alpha \sum_{(x,y)\sim\tilde{p}_l} \ln q_\phi(y|x)$$

Kingma, D. P., Mohamed, S., Rezende, D. J., & Welling, M. (2014). Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*(pp. 3581-3589).

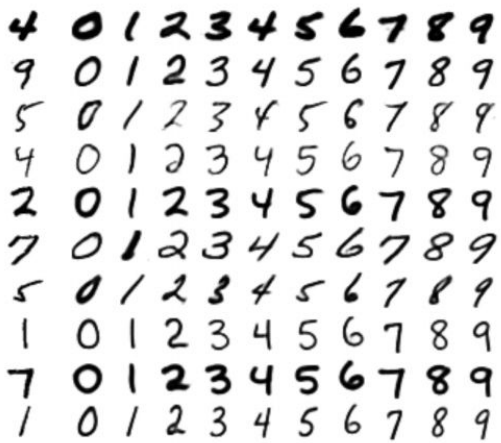
Semi-supervised conditional generation

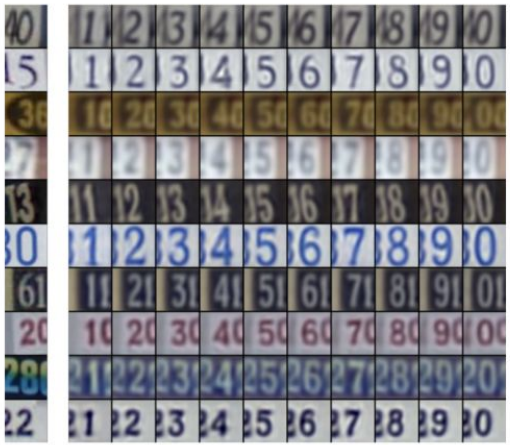
- Handwriting styles by fixing class label y, and varying 2 dimensional latent variable z

Kingma, D. P., Mohamed, S., Rezende, D. J., & Welling, M. (2014). Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*(pp. 3581-3589).

Semi-supervised conditional generation

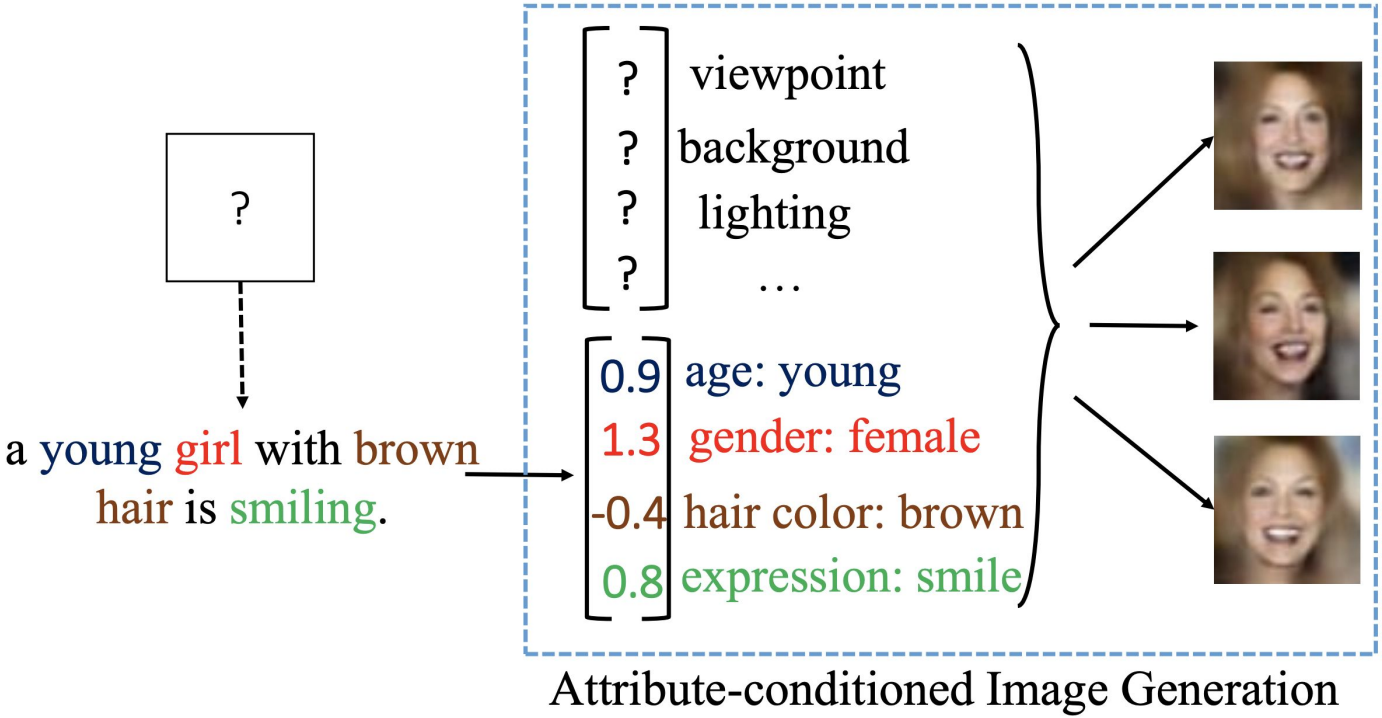
- The leftmost columns show images from the test set.
- The other columns show generated images x, where the latent variable z of each row is set to the value inferred from the test-set image on the left. Each column corresponds to a class label y.





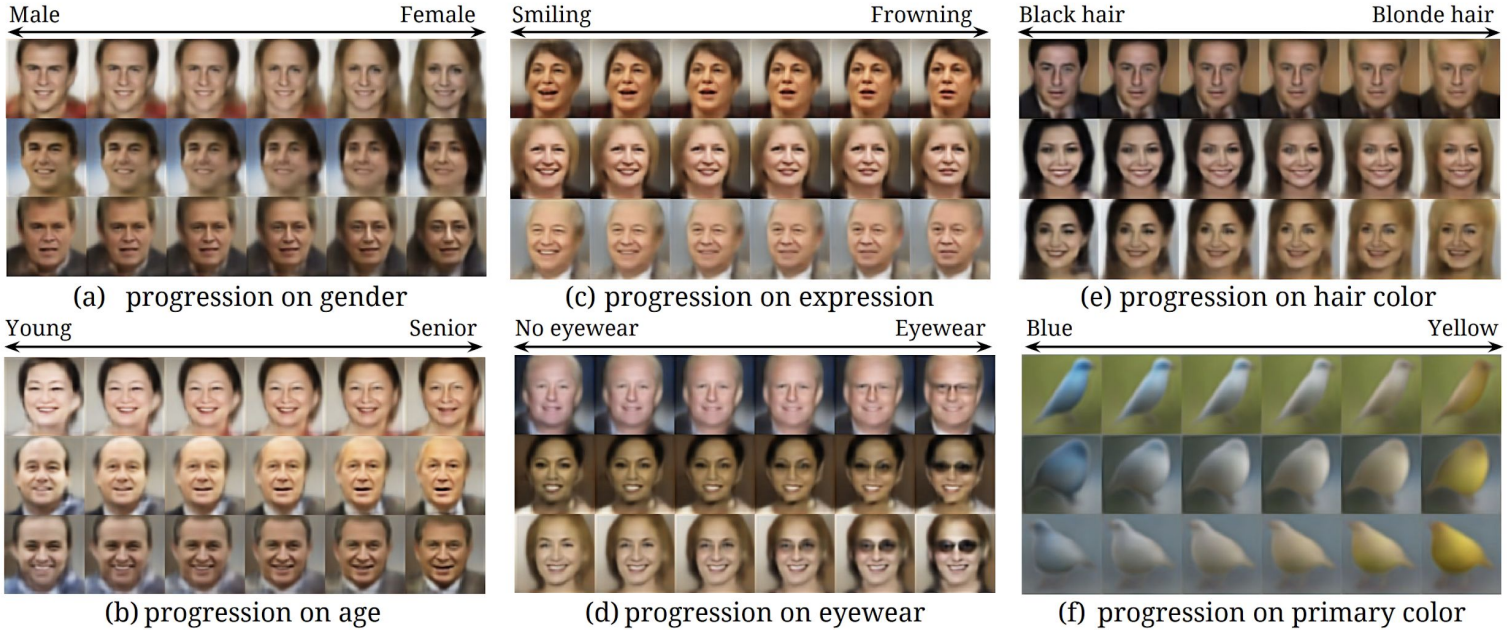
Kingma, D. P., Mohamed, S., Rezende, D. J., & Welling, M. (2014). Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*(pp. 3581-3589).

Attribute-conditioned Image generation



Yan, X., Yang, J., Sohn, K., & Lee, H. (2016, October). Attribute2image: Conditional image generation from visual attributes. In *European Conference on Computer Vision* (pp. 776-791). Springer, Cham.

Attribute-conditioned Image generation



$$p_\theta(x|y,z) \text{ with } z \sim \mathcal{N}(0, I) \text{ and } y = [y_\alpha, y_{rest}], \text{ where } y_\alpha = (1-\alpha)\cdot y_{min} + \alpha \cdot y_{max}$$

Yan, X., Yang, J., Sohn, K., & Lee, H. (2016, October). Attribute2image: Conditional image generation from visual attributes. In *European Conference on Computer Vision* (pp. 776-791). Springer, Cham.

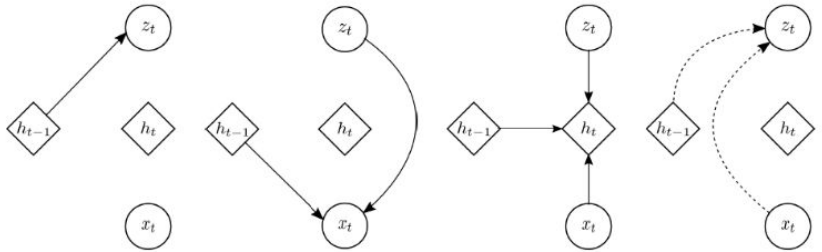
VAEs

- Improved techniques for training VAEs
  - Importance-weighted Autoencoders (IWAE)
  - Autoregressive Flow
  - VAE-GAN hybrids
- Conditional Generation
  - Class conditioning
  - Attribute conditioning
  - Semi-supervised learning
- **Case study**
  - **Sequence data**
  - **Discrete latent variables**
- Other applications

Variational RNN

- Goal: Learn a joint distribution over a sequence $p(x_1, \cdots, x_T)$
- VAE for sequential data, using latent variables $z_1, \cdots, z_T$

$$p(x_{\leq T}, z_{\leq T}) = \prod_{t=1}^{T} p(x_t \mid z_{\leq t}, x_{<t}) p(z_t \mid z_{<t}, x_{<t})$$



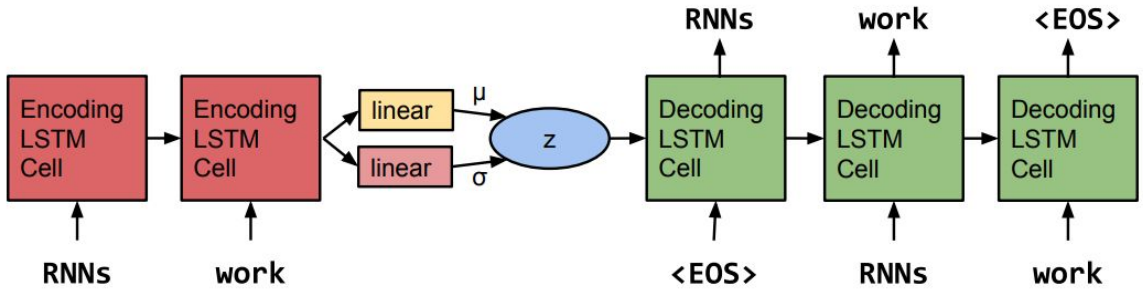
Chung et al, 2016

- Use RNNs to model the conditionals (similar to PixelRNN)
- Use RNNs for inference $p(z_{\leq T} | x_{\leq T}) = \prod_{t=1}^{T} q(z_t \mid z_{<t}, x_{\leq t})$
- Train like VAE to maximize ELBO. Conceptually similar to PixelVAE.

Slide credit: Stefan Ermano

Generating sentences from a continuous space

- VAE-based approach for modeling sentences
- Text auto-encoder with KL constraints on the latent variable



Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., & Bengio, S. (2015). Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*.

Generating sentences from a continuous space

- Allows for consistent latent space of sentences

Interpolation in latent space

Standard encoder-decoder

i went to the store to buy some groceries .
i store to buy some groceries .
i were to buy any groceries .
horses are to buy any groceries .
horses are to buy any animal .
horses the favorite any animal .
horses the favorite favorite animal .
horses are my favorite animal .

VAE

“ i want to talk to you . ”
“i want to be with you . ”
“i do n’t want to be with you . ”
i do n’t want to be with you .
she did n’t want to be with him .
he was silent for a long moment .
he was silent for a moment .
it was quiet for a moment .
it was dark and cold .
there was a pause .
it was my turn .

Generating sentences from a continuous space

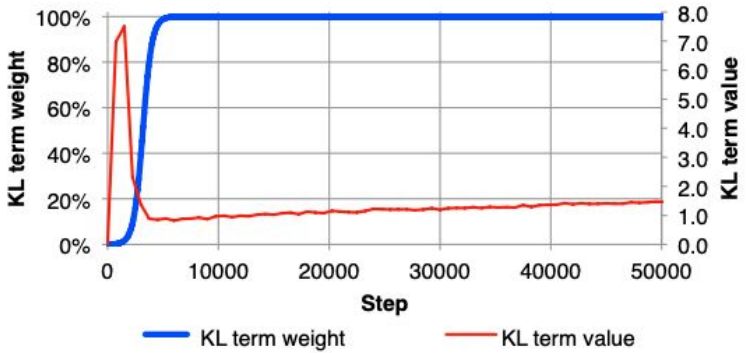
- Training difficulty
  - Of the two components in the VAE objective, the KL divergence term is much easier to learn!
- Also called Posterior Collapse
  - Results in the model learning to rely solely on decoder and ignore latent variable

$$\underbrace{\mathbb{E}_{\boldsymbol{z}\sim Q(\boldsymbol{z}|\boldsymbol{x})}[\log P(\boldsymbol{x} \mid \boldsymbol{z})]}_{\text{Requires good generative model}} - \underbrace{\mathcal{KL}[Q(\boldsymbol{z} \mid \boldsymbol{x})||P(\boldsymbol{z})]}_{\text{Just need to set the mean/variance of Q to be same as P}}$$

Slide credit: Graham Neubig

Generating sentences from a continuous space

- Some solutions to address posterior collapse
- Solution 1: KL annealing
  - Multiply KL term by a constant λ starting at zero, then gradually increase to 1
  - Result: model can learn to use z before getting penalized



- Solution 2
  - Weaken decoder P(x|z) so using z is essential
  - Use word dropout to occasionally skip inputting previous word in x

Slide credit: Graham Neubig

Handling discrete latent variables

- Many variables are better treated as discrete
  - Eg: Part-of-speech of a word, Class of a question, Speaker traits (gender, etc.)
- Reparameterization also possible for discrete variables!
- Original Categorical Sampling Method

$$\hat{\boldsymbol{z}} = \text{cat-sample}(P(\boldsymbol{z} \mid \boldsymbol{x}))$$

- Reparameterized Method

$$\hat{\boldsymbol{z}} = \text{argmax}(\log P(\boldsymbol{z} \mid \boldsymbol{x}) + \text{Gumbel}(0,1))$$

  - where the Gumbel distribution is

$$\text{Gumbel}(0,1) = -\log(-\log(\text{Uniform}(0,1)))$$

- Backprop is still not possible, due to argmax

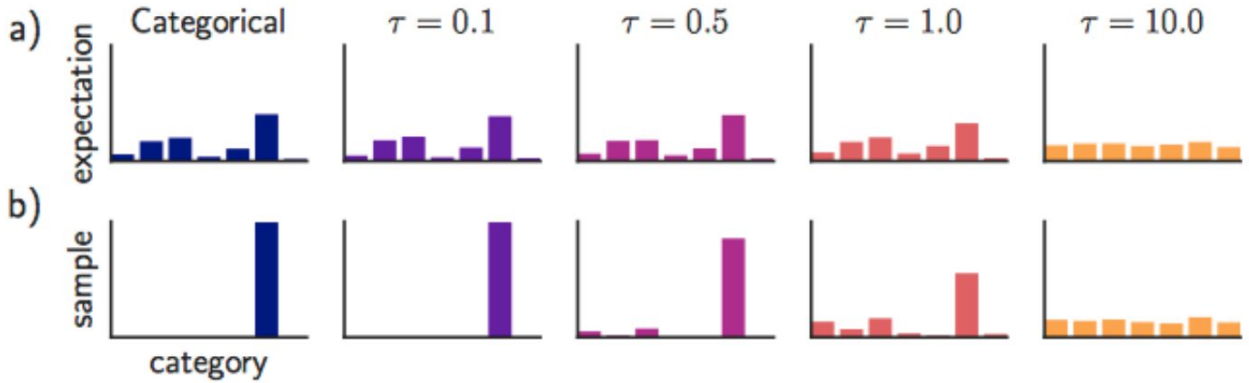
Slide credit: Graham Neubig

Gumbel-Softmax

- A way to soften the decision and allow for continuous gradients
- Instead of argmax, take softmax with temperature τ

$$\hat{\boldsymbol{z}} = \text{softmax}((\log P(\boldsymbol{z} \mid \boldsymbol{x}) + \text{Gumbel}(0,1))^{1/\tau})$$

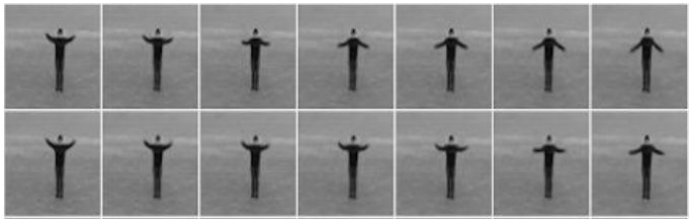
- As τ approaches 0, will approach max



Slide credit: Graham Neubig

Other VAE applications

Video prediction [Denton et al., 2018]



Sketch generation [Ha et al., 2017]

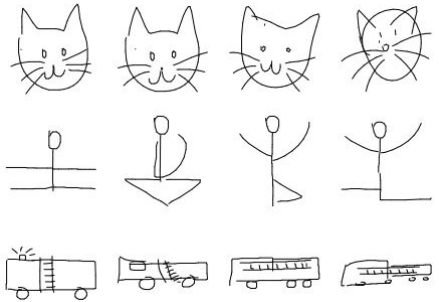
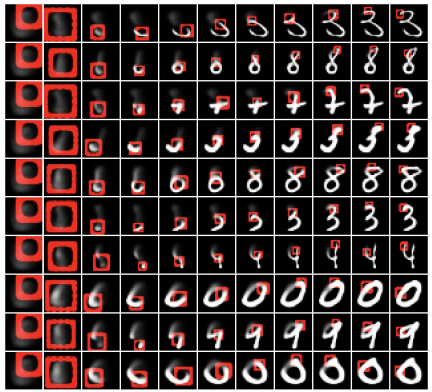


Image generation [Gregor et al., 2015]



Discrete models [Oord et al., 2017]