CMPS 6720: Machine Learning

Deep Generative Models

Jihun Hamm

Deep Generative Models

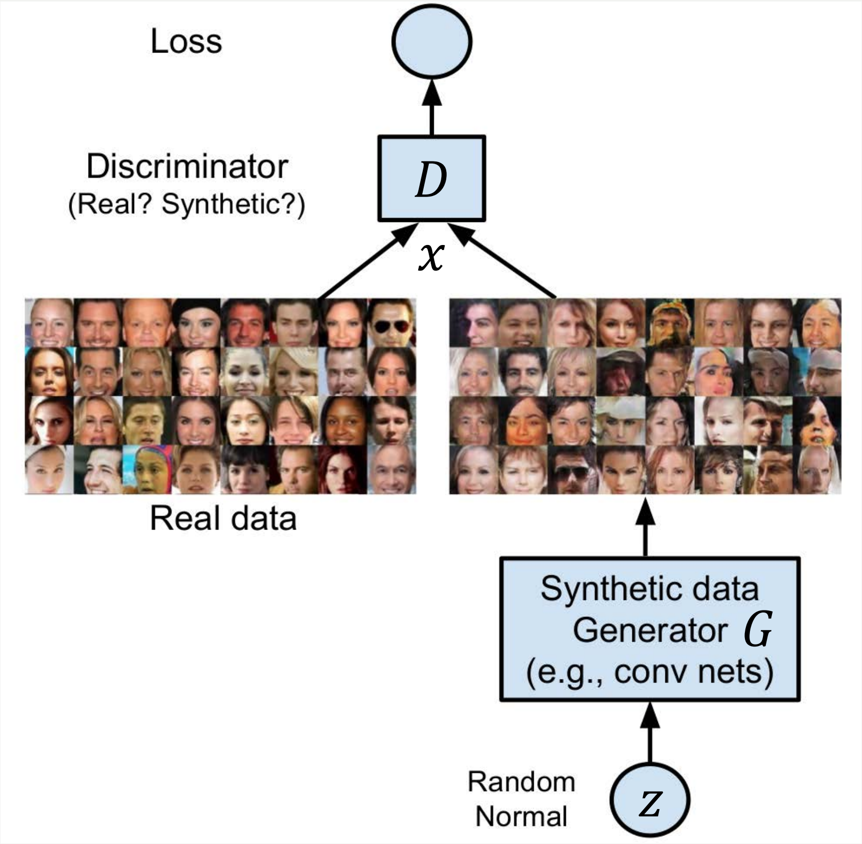
- Since ~2010, Deep Neural Nets have made a great progress in **supervised** learning problems
  - Conv Nets, VGG, Inception, ResNets, U-Nets, …
  - Classification of 1000-classes from ImageNet
  - State-of-the-art results

- Since ~2014, Deep Neural Nets is also making a big impact in **unsupervised** learning problems
  - “Deep generative models”
    - Generative Adversarial Nets [Goodfellow et al., 2014]
    - Variational Autoencoder [Kingma & Welling, 2014]
  - Modeling complex data distributions like faces, scenery, etc
  - State-of-the-art results

Cf. Generative model for classification

- Generative models can be used for both *unsupervised* and *supervised learning*
- Classification using discriminative models
  - Examples: Logistic regression, Conditional Random Fields
  - Model $P(y|x)$. Don't care about $P(x)$ ➡ Easy to learn !
  - Train: Learn params of $P(y|x)$ from data
  - Test: Infer $y = argmax_c P(y = c|x)$
- Classification using generative models
  - Examples: Bayes Nets, Markov Nets, Mixture models
  - Model $P(x|y)$ and $P(y)$ ➡ Can generate new data !
  - Train: Learn params of $P(x|y),\ P(y)$ from data
  - Test: Infer $y = argmax_c P(y = c|x) = P(x|y = c)P(y = c)/P(c)$

Generative Adversarial Nets



GAN-related work

- Variants
  - Deep convolutional GAN (DCGAN) [Radford et al.'15]
  - Conditional GAN [Mirza et al.'14]
  - Adversarially learned inference (ALI) [Dumoulin et al.'16]
  - Adversarial autoencoder (AAE) [Makhzani et al.'15]
  - Energy-based GAN (EBGAN) [Zhao et al.'16]
  - Wasserstein GAN (WGAN) [Arjovsky et al.'17]
  - Boundary equilibrium GAN (BEGAN) [Berthelot et al.'17]
  - Bayesian GAN [Saatchi et al.'17]
  - …
- Applications

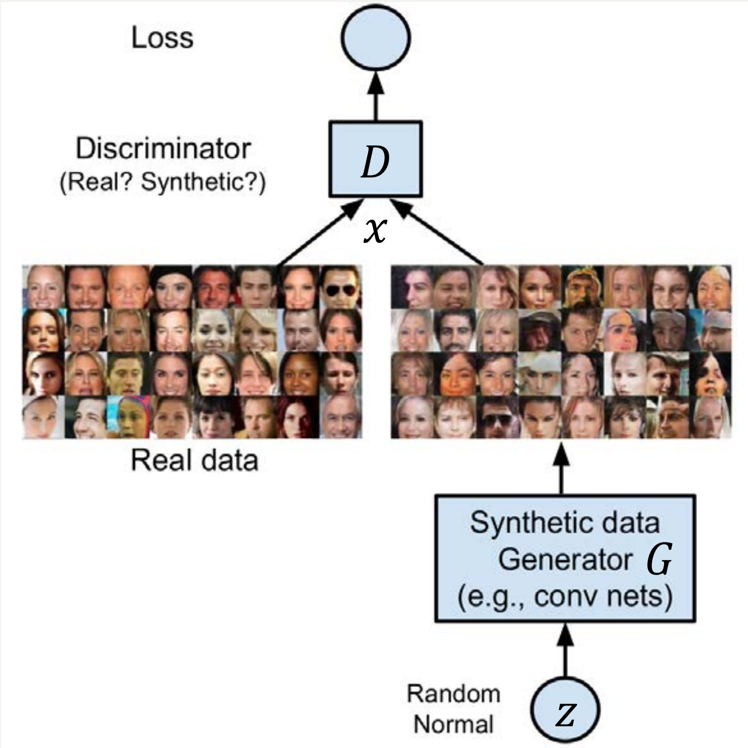
  …

More examples of adversarial ML

- GAN
- Domain adaptation
- Robust classification: narrow-sense adversarial learning
- Privacy-preservation / fair learning
- Attack on Deep NN
- Training data poisoning
- Hyperparameter learning
- Meta-learning
- …

GAN architecture

- $D(x): X \rightarrow [0,1]$ is the discriminator
  - Input: example x
  - Output: probability that x is from $P_{data}(x)$
  - Usually a conv net, can be any structure



- $G(z)$: $R^d \rightarrow X$ is the generator
  - Input: random noise $z \sim P_z(z)$
    (typically $P_z(z) = N(0, I)$)
  - Output: fake data
  - Usually a conv net, can be any structure
  - Complex distribution $P(x)$ can be modeled using a large-enough G

GAN objective

- Find G and D which solves

$$\min_G \max_D \; V(G,D), \quad \text{where}$$

$$\begin{aligned} V(G,D) &= E_{x\sim P_{data}}[\log D(x)] + E_{z\sim P_z}[\log(1-D(G(z)))] \\ &= E_{x\sim P_{data}}[\log D(x)] + E_{x\sim P_{fake}}[\log(1-D(x))] \end{aligned}$$

  - In practice, use random samples $x_1, \dots, x_N$ and $z_1, \dots, z_M$:

$$V(G,D) = \frac{1}{N}\sum_i \log D(x_i) + \frac{1}{M}\sum_j \log(1-D(G(z_i)))$$

- It is an instance of minimax problems

Minimax optimization

- General form: $\min_{u \in U} \max_{v \in V} f(u, v)$
  - Zero-sum leader-follower games
  - u: leader/min player, v: follower/max player
  - $f(u, v)$: payoff of follower/max player

Minimax optimization

inner maximization

- General form: $\min_{u \in U} \boxed{\max_{v \in V} f(u, v)}$

  - Zero-sum leader-follower games
  - u: leader/min player, v: follower/max player
  - $f(u, v)$: payoff of follower/max player

Minimax optimization

inner maximization

- General form: $\boxed{\min_{u \in U} \boxed{\max_{v \in V} f(u, v)}}$

outer minimization

  - Zero-sum leader-follower games
  - u: leader/min player, v: follower/max player
  - $f(u, v)$: payoff of follower/max player

GAN is JS-divergence minimization

- Proposition 1 [Goodfellow et al.,2014]

  Given G, the optimal discriminator D is $D^*(G) = \frac{P_{data}(x)}{P_{data}(x)+P_{fake}(x)}$

- Proof:

  $V(G,D) = \int \left[P_{data}(x) \log D(x) \ + \ P_{fake}(x) \log(1 - D(x))\right] dx.$

  For any $(a,b) \in R^2 - (0,0)$, the function $y \mapsto a \log(y) + b \log(1-y)$ achieves its maximum in $[0,1]$ at $\frac{a}{a+b}$. The discriminator does not need to be defined outside of $Supp(P_{data}) \cup Supp(P_{fake})$, concluding the proof.

Cont'd

- Def: $JSD(p||q) = \frac{1}{2}KL(p||\frac{p+q}{2}) + \frac{1}{2}KL(q||\frac{p+q}{2})$
- Theorem 1. $\max_D V(G,D) = -\log 4 + 2 \cdot JSD(P_{data}||P_{fake})$
- Corollary: GAN objective is $\min_G JSD(P_{data}||P_{fake})$, that is, find the generator G so that real and synthetic data are indistinguishable

- Proof:

$$\max_D V(G,D) = V(G,D^*)$$
$$= E_{x\sim P_{data}}[\log D^*(x)] + E_{x\sim P_{fake}}\left[\log\left(1 - D^*(x)\right)\right]$$
$$= E_{x\sim P_{data}}[\log \frac{P_{data}(x)}{P_{data}(x) + P_{fake}(x)}] + E_{x\sim P_{fake}}[\log \frac{P_{fake}(x)}{P_{data}(x) + P_{fake}(x)}]$$
$$= -\log 4 \ + KL(P_{data}||\frac{P_{data} + P_{fake}}{2}) + KL(P_{fake}||\frac{P_{data} + P_{fake}}{2})$$

Optimization by Gradient Descent

- Simplest way to solve min (or max) approximately
- Alternating

$$u \leftarrow u - \rho \nabla_u f, \quad v \leftarrow v + \eta \nabla_v f$$

- Simultaneous

$$\binom{u}{v} \leftarrow \binom{u}{v} + \begin{pmatrix} -\rho I & 0 \\ 0 & \eta I \end{pmatrix} \nabla f$$

- GAN optimization (original version)
  - Assume parameters $G(z; u)$ and $D(x; v)$
  - Alternate between $\boldsymbol{u} \leftarrow \boldsymbol{u} - \boldsymbol{\rho} \boldsymbol{\nabla}_{\boldsymbol{u}} \boldsymbol{V}$ and $\boldsymbol{v} \leftarrow \boldsymbol{v} + \boldsymbol{\eta} \boldsymbol{\nabla}_{\boldsymbol{v}} \boldsymbol{V}$
  - Note that it isn't exactly solving a minimax problem

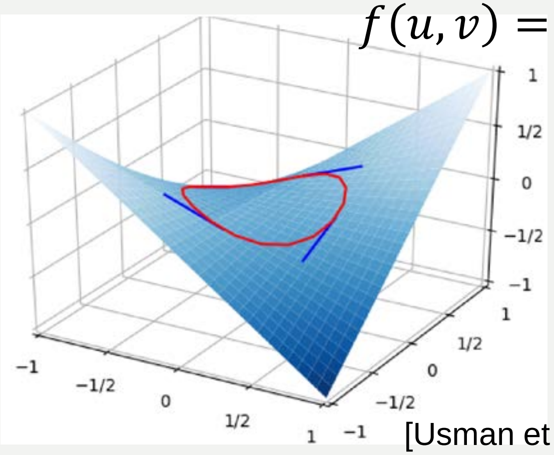
Gradient descent is unstable

- Gradient descent does not always converge

→ Modify gradient descent

[Mescheder et al.'17; Nagarajan et al.'17; Roth et al.'17]

$f(u, v) = uv$



[Usman et al.'18]

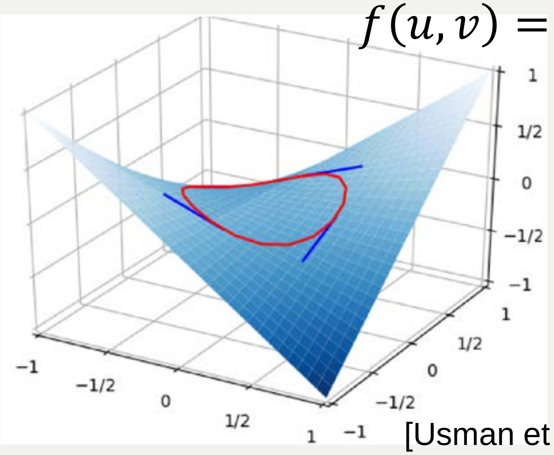
Gradient descent is unstable

- Gradient descent does not always converge

→ Modify gradient descent

[Mescheder et al.'17; Nagarajan et al.'17; Roth et al.'17]

$f(u, v) = uv$



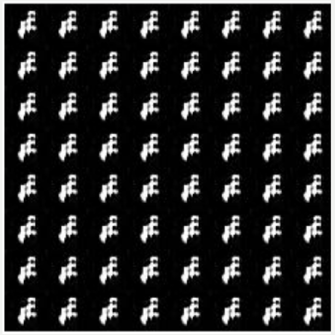
[Usman et al.'18]

- GAN training fails frequently

→ Change the GAN objective

[Uehara et al.'16; Nowozin et al.'16; Arjovsky et al.'17]

vs



[Mets et al.'17]

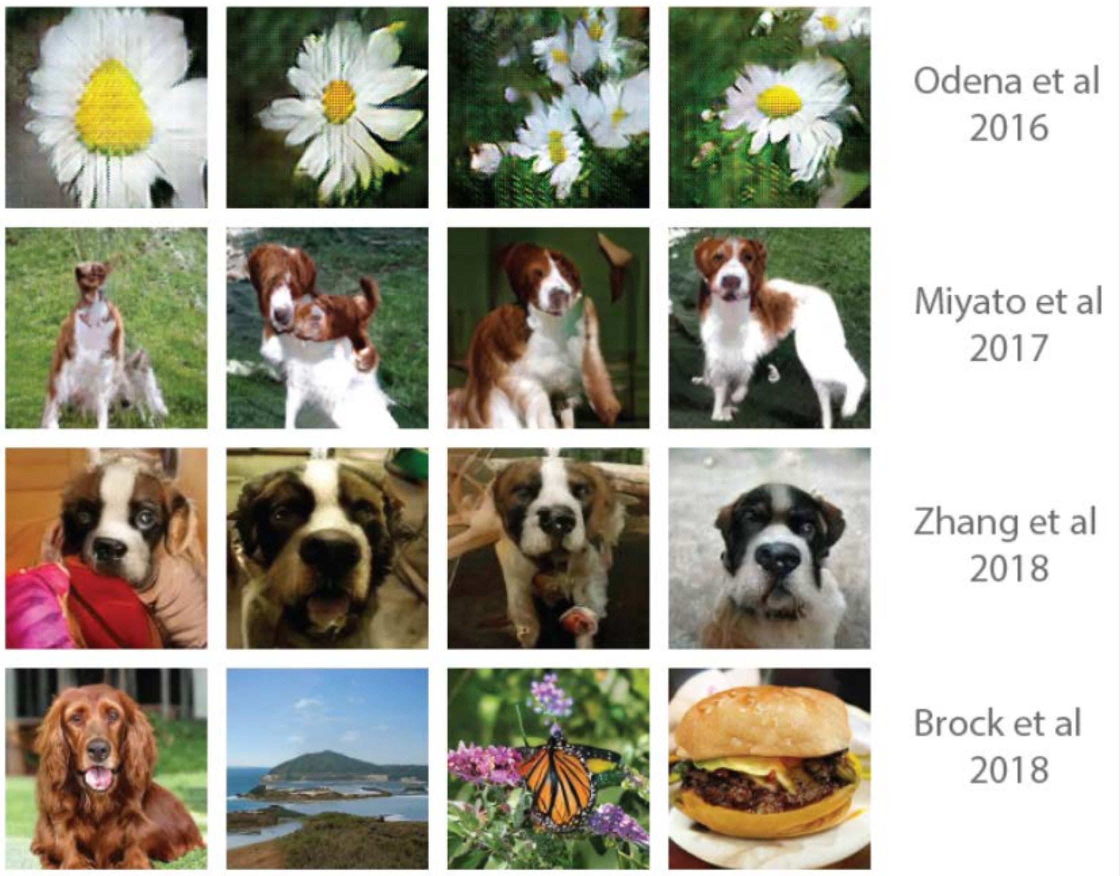
Alternative losses $V(G, D)$

- Minimax (original) GAN
  - $\min_G \max_D \; E_{P_{data}}[\log D(x)] + E_{P_{fake}}[\log(1 - D(x))]$
- Non-saturating GAN
  - $\max_D \; E_{P_{data}}[\log D(x)] + E_{P_{fake}}[\log(1 - D(x))]$
  - $\min_G \; -E_{P_{fake}}[\log D(x)]$
- Wasserstein GAN
  - $\min_G \max_D \; E_{P_{data}}[D(x)] - E_{P_{fake}}[D(x)],$ D is 1-Lipschitz, real-valued
- Least-squares GAN
  - $\max_D \; E_{P_{data}}[(D(x) - 1)^2] - E_{P_{fake}}[D(x)^2],$ D is real-valued
  - $\min_G \; -E_{P_{fake}}[(D(x) - 1)^2]$
- Additional regularizers: $\|\nabla D(x)\|^2$ or $(\|\nabla D(x)\| - 1)^2$

Progress on face generation

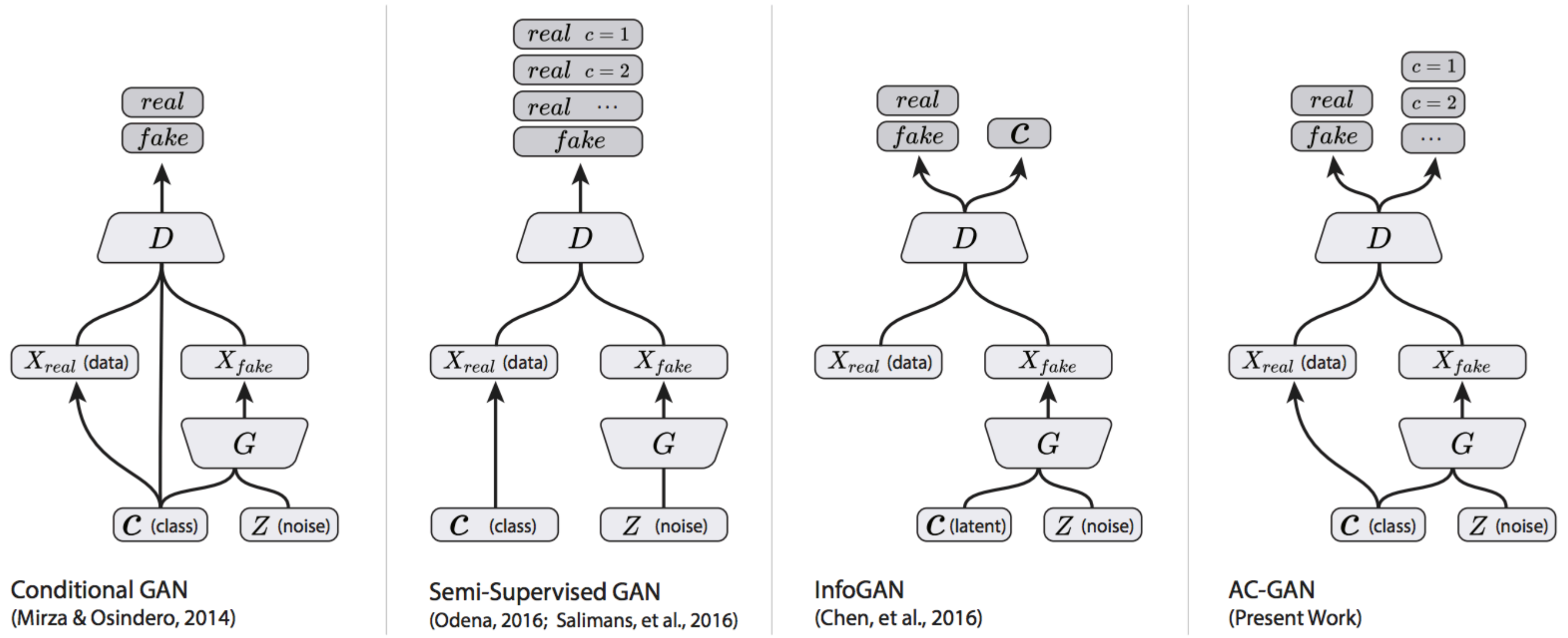
Goodfellow.,2019

Progress on ImageNet



(Odena 2018)

GAN variants that use labels

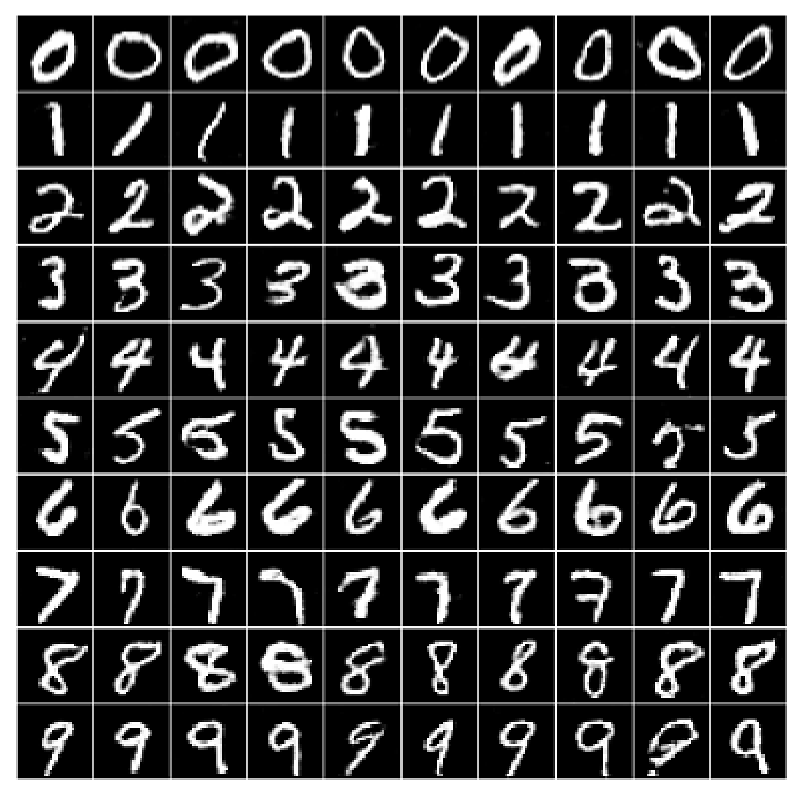


https://machinelearningmastery.com/how-to-develop-an-auxiliary-classifier-gan-ac-gan-from-scratch-with-keras/

GAN vs C-GAN

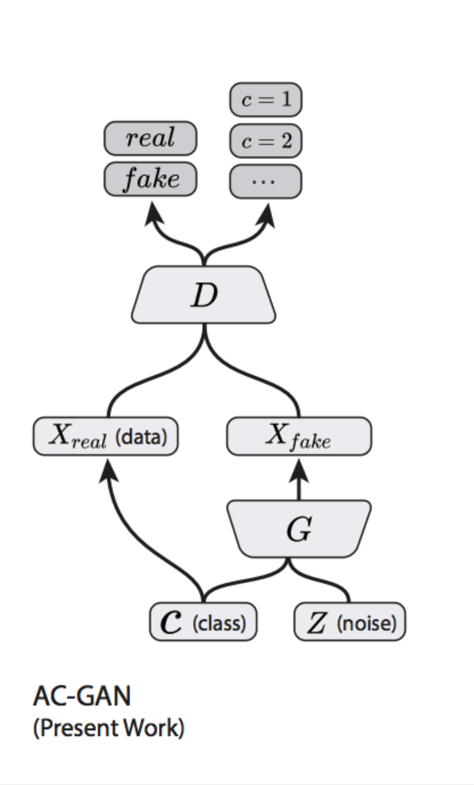
GAN

C-GAN



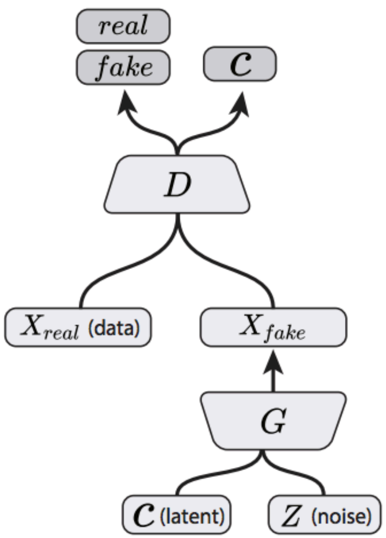
AC-GAN

- Different from C-GAN
  - No label input to D
  - D output both real/fake and class
  - Label info has to be encoded in x_{fake}
- C-GAN loss
  - $\max_D E_{P_{data}}[\log D(x|c)] + E_{P_{fake}}[\log(1 - D(x|c))]$
  - $\min_G -E_{P_{fake}}[\log D(x|c)]$
- AC-GAN loss
  - $\max_D \quad \cdots \; + \; E_{P_{data}}[\log P(c|x)] + E_{P_{fake}}[\log P\,(c|x)]$
  - $\min_G \quad \cdots \; - \; E_{P_{fake}}[\log P(c|x)]$

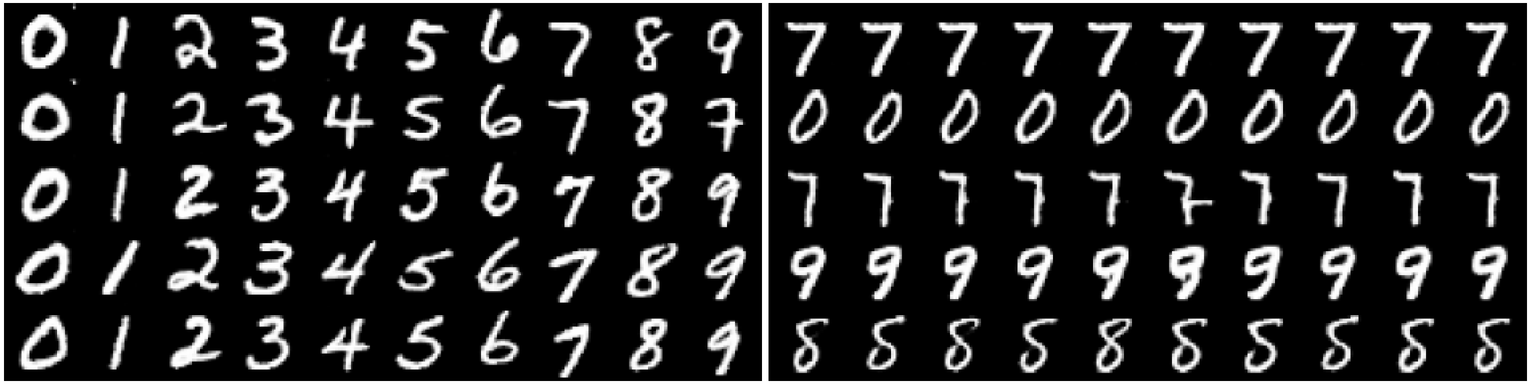


InfoGAN

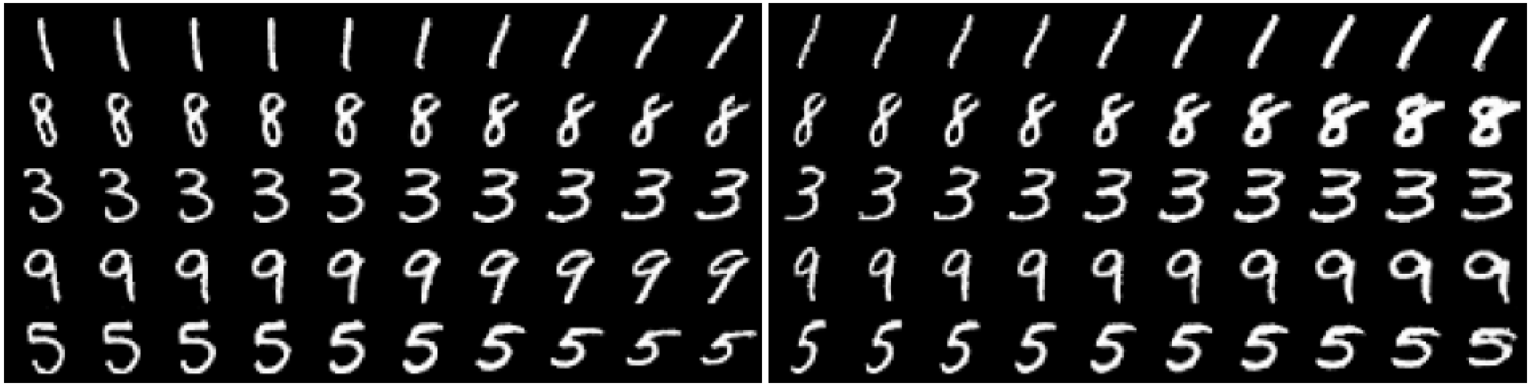
- Different from C-GAN
  - Label *c* of examples is not given, but is random
  - MNIST example, c=[digits (0-9), rotation, thickness]^T
  - Label info has to be encoded in x_{fake},
    measured by mutual information $I(c,\ x_{fake} = G(z,c))$
- InfoGAN loss
  - $$\max_D E_{x\sim P_{data}}[\log D(x)] + E_{(z,c)\sim P_{z,c}}[\log(1 - D(G(z,c)))] + \lambda I(c, G(z,c))$$
  - $$\min_{G,Q} -E_{(z,c)\sim P_{z,c}}[\log D(G(z,c))] - \lambda I(c, G(z,c))$$
  - Lemma: $I\big(c; G(z,c)\big) \geq E_{c\sim P(c),\, x\sim G(z,c)}[\log Q(c|x)]$
    for any $Q(c|x)$.
  - That is, we don't need to estimate mutual information directly !!!



InfoGAN
(Chen, et al., 2016)



(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)

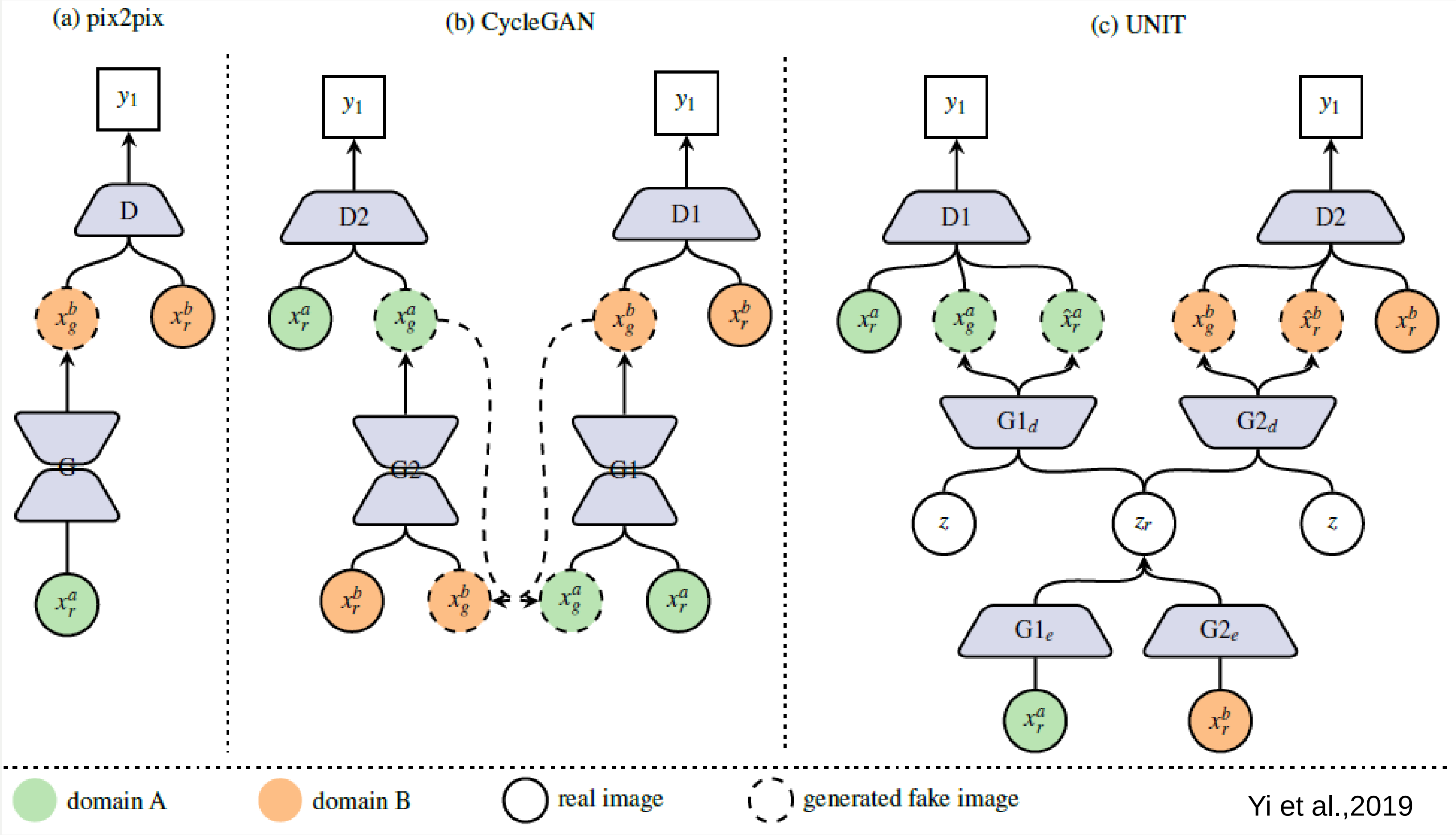
(a) Azimuth (pose)

(b) Presence or absence of glasses

(c) Hair style

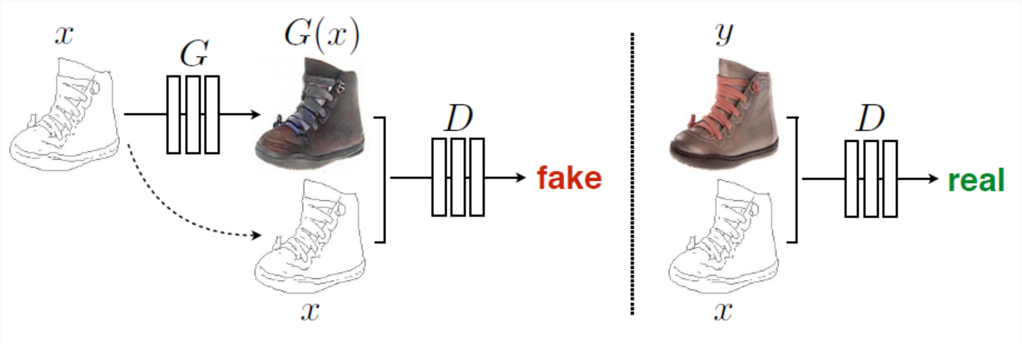
(d) Emotion

Image-to-image translation

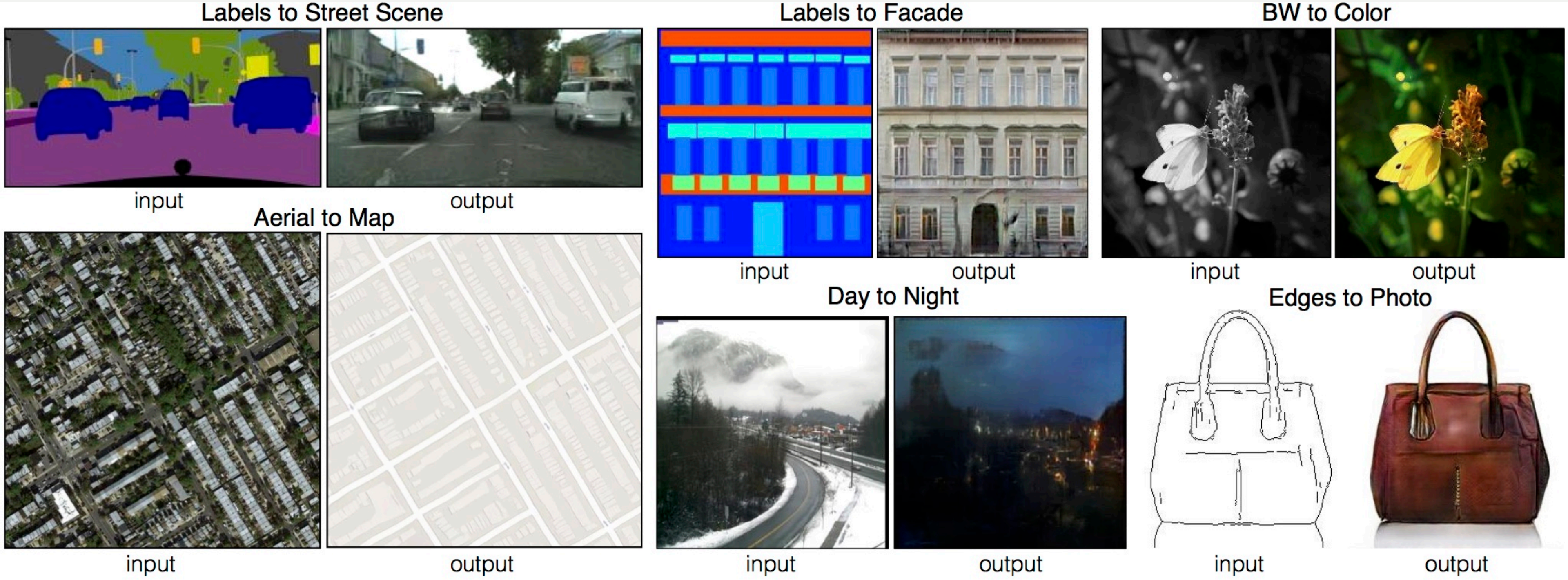


Yi et al.,2019

Pix2pix [Isola et al.,17]

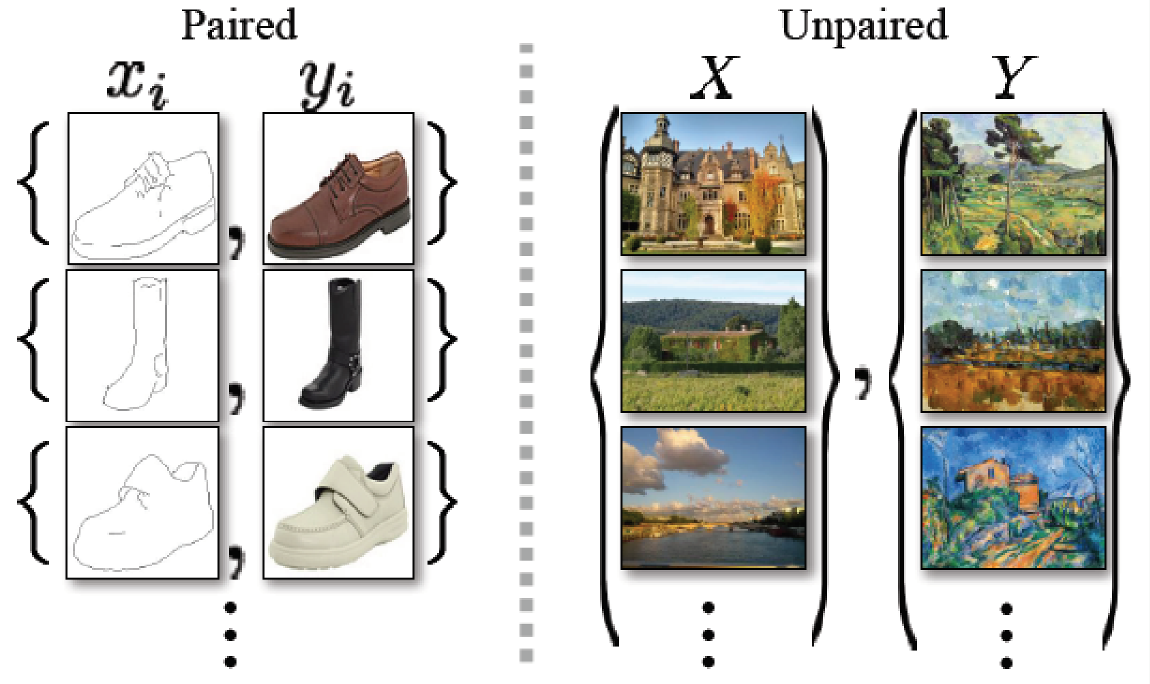


- $G(x)$: takes an image and outputs another image
- $D(x)$: use both x and y (similar to C-GAN)
- Loss
  - $\min_G \max_D E_{x,y\sim P_{data}}[\log D(x,y)] + E_{(x,z)\sim P_{x,z}}[\log(1 - D(x, G(x,z)))]$
    $+ \lambda E_{(x,y,z)} \|y - G(z,x)\|_1$


Labels to Street Scene
input
output
Aerial to Map
input
output
Labels to Facade
input
output
BW to Color
input
output
Day to Night
input
output
Edges to Photo
input
output

CycleGAN [Zhu et al.,'17]

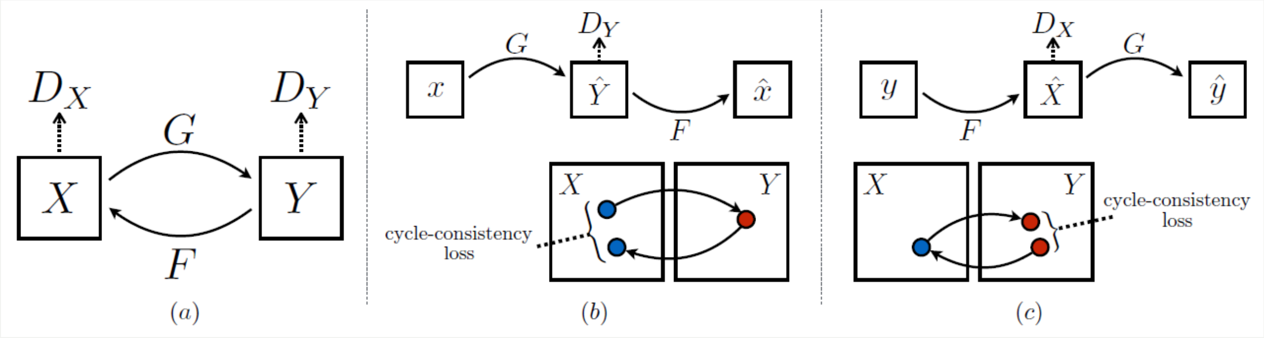
- Does not require paired data

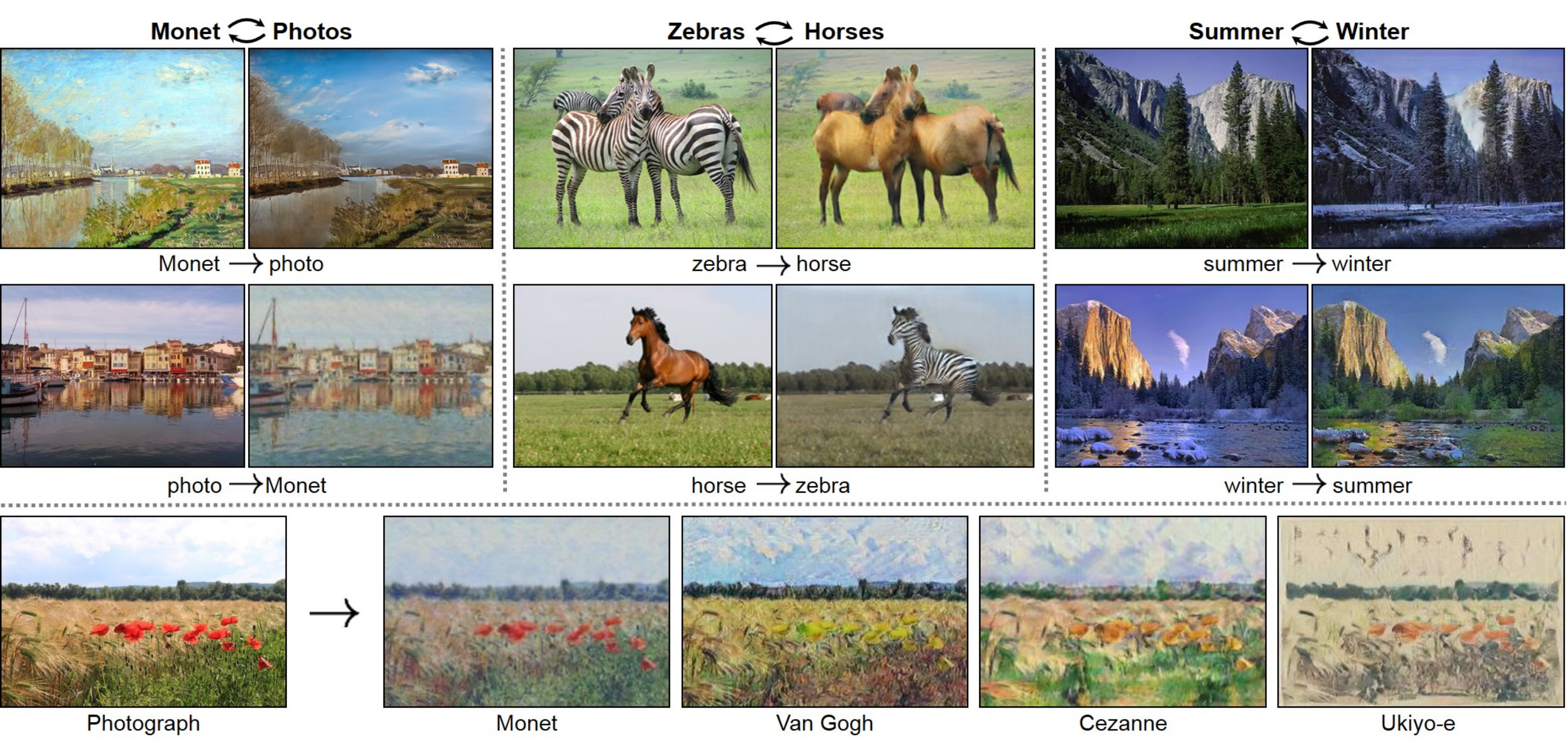


- Two domains: X and Y
- Two generators: F and G
- Two discriminators: D_X and $D_Y T$
- Loss: $L_{GAN}(G, F, D_Y, X, Y) + L_{GAN}(F, D_X, Y, X) + \lambda L_{CYC}(G, F)$ where

$$L_{CYC}(G,F) = E_{x \sim P_{data}(x)} \left\| F\big(G(x)\big) - x \right\|_1 + E_{y \sim P_{data}(y)} \left\| G\big(F(y)\big) - y \right\|_1$$

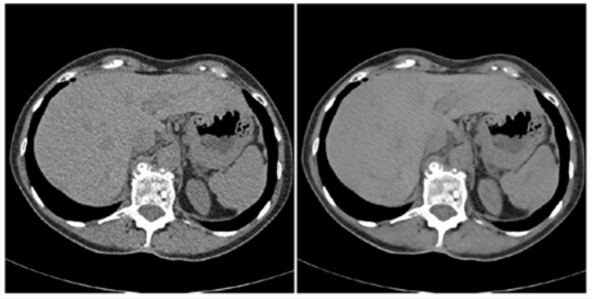
  - Cyclic consistency



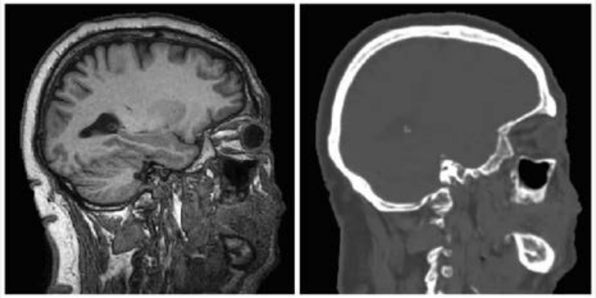

Monet ⇄ Photos
Zebras ⇄ Horses
Summer ⇄ Winter
Monet → photo
zebra → horse
summer → winter
photo → Monet
horse → zebra
winter → summer
Photograph
→
Monet
Van Gogh
Cezanne
Ukiyo-e

Medical Imaging [Yi et al.'19]

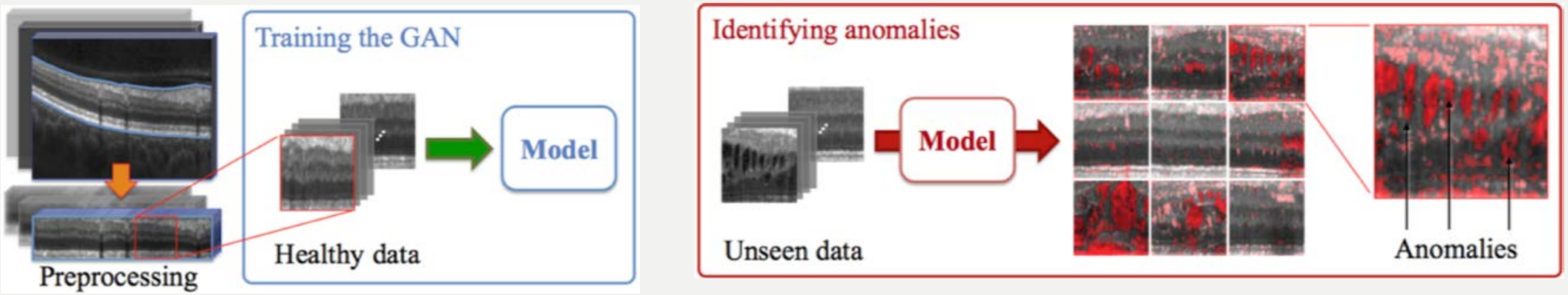
Denoising [Yi et al.'18]



Modality transfer [Wolterink et al.'17]



Anomaly detection [Schlegl et al.'17]



Summary

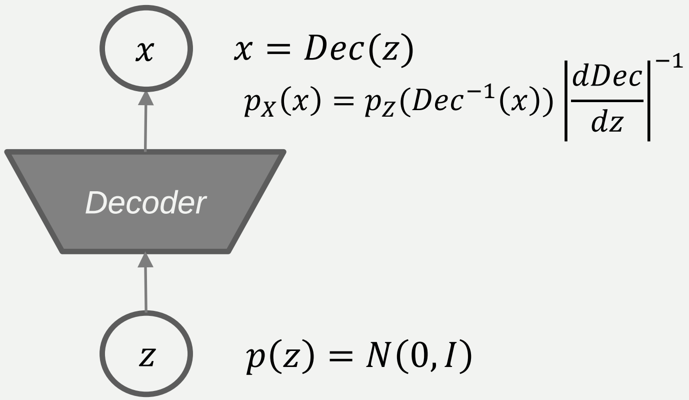
- GAN can be extremely useful for data generation
- Many interesting properties
- Underlying mechanism not completely understood
- Optimization still difficult
- Lacking universal metric of data generation performance
- Topic of intense ongoing research

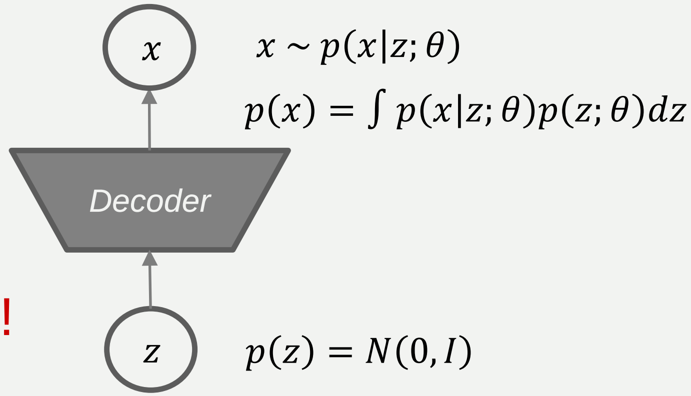
References

- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y., 2014. Generative adversarial nets, NIPS
- Radford, A., Metz, L. and Chintala, S., 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.
- Nowozin, S., Cseke, B. and Tomioka, R., 2016. f-gan: Training generative neural samplers using variational divergence minimization, NIPS
- Odena, A., 2016. Semi-supervised learning with generative adversarial networks. arXiv preprint arXiv:1606.01583.
- Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A. and Chen, X., 2016. Improved techniques for training gans, NIPS
- Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I. and Abbeel, P., 2016. Infogan: Interpretable representation learning by information maximizing generative adversarial nets, NIPS
- Arjovsky, M., Chintala, S., and Bottou, L., 2017. Wasserstein gan. arXiv preprint arXiv:1701.07875.
- Nagarajan, V. and Kolter, J. Z., 2017. Gradient descent gan optimization is locally stable, NIPS
- Mescheder, L., Nowozin, S., and Geiger, A. 2017. The numerics of gans, NIPS
- Zhu, J.Y., Park, T., Isola, P. and Efros, A.A., 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks, CVPR
- Isola, P., Zhu, J.Y., Zhou, T. and Efros, A.A., 2017. Image-to-image translation with conditional adversarial networks, CVPR
- Odena, A., Olah, C. and Shlens, J., 2017, August. Conditional image synthesis with auxiliary classifier gans, ICML
- Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. and Courville, A.C., 2017. Improved training of wasserstein gans, NIPS
- Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z. and Smolley, P., 2017. Least squares generative adversarial networks, CVPR
- Huang, H., Yu, P.S. and Wang, C., 2018. An introduction to image synthesis with generative adversarial nets. arXiv preprint arXiv:1803.04469.
- Yi, X., Walia, E. and Babyn, P., 2019. Generative adversarial network in medical imaging: A review. Medical image analysis

Back to generative model

- GAN as generative model
  - x is determined by latent variable z by decoder $x = Dec(z)$
  - $p(x) = p_Z(Dec^{-1}(x))\det\left|\frac{dDec}{dz}\right|^{-1}$
    - Difficult to compute
- Parametric generative model
  - x is sampled from $p(x|z;\theta)$ by decoder $x \sim Dec(z) = p(x|z;\theta)$
  - $p(x) = \int p(x|z;\theta)p(z;\theta)dz$
    - May be easy or difficult to compute



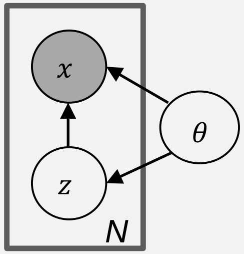


Caution: These are NOT graphical models !

Marginal likelihood

- Graphic model for generative model
  - Short-hand notations:

    $p_\theta(z) \coloneqq p(z;\theta), \quad p_\theta(x|z) := p(x|z;\theta)$

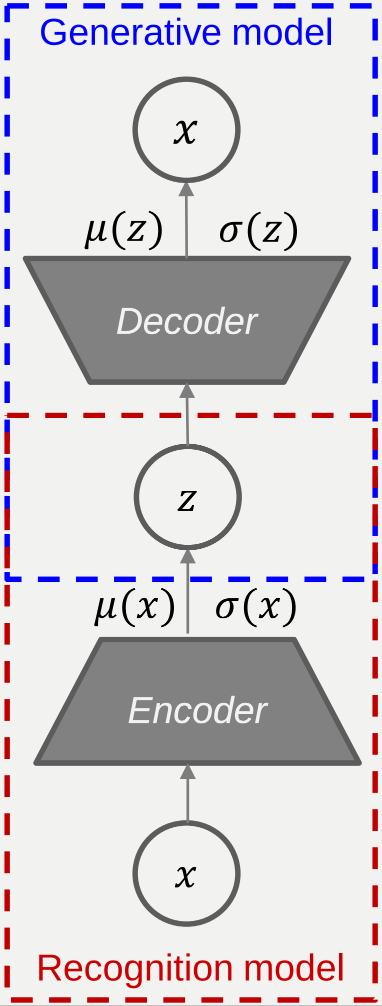


- Marginal likelihood evaluation
  - $p_\theta(x) = \int p_\theta(x|z)p_\theta(z)dz = E_z[p_\theta(x|z)]$
- Problem
  - Only in simple cases, (e.g., Gaussian) $p(x)$ is exactly computable
  - All other complex (and therefore interesting) cases, $p(x)$ is computable only approximately (e.g., by Monte-Carlo integration)
  - $p_\theta(x) = E_z[p_\theta(x|z)] \cong \frac{1}{N}\sum_i p_\theta(x|z_i), \quad z_i \sim p_\theta(z), \ iid$
- Better: use high-probability samples
  - $p_\theta(x) \cong E_{z\sim q_\phi}[p_\theta(x|z)] \ \cong \frac{1}{N}\sum_i p_\theta(x|z_i), \quad z_i \sim q_\phi(z|x), \ iid,$

    where $q_\phi(z|x) \cong p_\theta(z)$

Variational Autoencoder (VAE) [Kingma & Welling'14]

- Generative model
  - Assume $p(z) = N(0, I)$
  - Decoder (with parameter θ) takes z as input and outputs mean $u(z)$ and covariance $\sigma^2(z)$
  - x is sampled from $N(u(z), \sigma(z)I)$
  - $p(x, z)$ is not jointly Gaussian. cf. linear-Gaussian



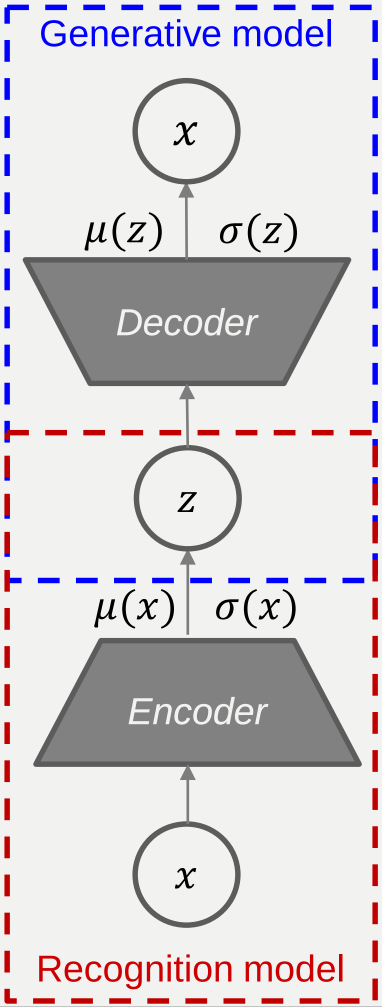
$p_\theta(x|z) = N(\mu(z), \sigma^2(z)I)$

$p(z) = N(0, I)$

$q(z|x;\phi) = N(\mu(x), \sigma^2(x)I)$

Variational Autoencoder (VAE) [Kingma & Welling'14]

- Recognition/inference model
  - Encoder (with parameter ϕ) takes x as input and outputs mean $u(x)$ and covariance $\sigma^2(x)$
  - z is sampled from $N(u(x), \sigma^2(x)I)$
  - Note $q_\phi(z|x)$ is not $p_\theta(z|x)$
  - Will be explained shortly
- Caution
  - VAE looks similar to AE, but it's a very different model
  - AE: deterministic $x \to z \to x$
  - VAE: probabilistic $x \to z \to x$



$p_\theta(x|z) = N(\mu(z), \sigma^2(z)I)$

$p(z) = N(0, I)$

$q_\phi(z|x) = N(\mu(x), \sigma^2(x)I)$

(Previously) Proof: EM increases log-likelihood

- Start with the log likelihood:

$$\log P(X;\theta) = \log P(X,Z;\theta) - \log P(Z|X;\theta)$$

- Take expectation over *Z* w.r.t. *q(Z)* on both sides:
- $LHS = \sum_z q(Z) \log P(X;\theta) = \log P(X;\theta)$
- $RHS = \sum_z q(Z) \log P(X,Z;\theta) \; - \sum_z q(Z) \log P(Z|X;\theta)$

$$= \;\cdots\; - q(Z)\log q(Z) + q(Z) \log q(Z)$$

$$= \sum_z q(Z) \log \frac{P(X,Z;\theta)}{q(Z)} - \sum_z q(Z) \log \frac{p(Z|X;\theta)}{q(Z)}$$

$$= l(q,\theta) \; + \; KL(q||p)$$

- Therefore $LHS = \log P(X;\theta) = RHS = l(q,\theta) + KL(q||p) \geq l(q,\theta)$
  - $l(q,\theta)$ is called **ELBO**: a lower bound of the evidence $\log p(X;\theta)$
  - Maximizing $l(q,\theta)$ also maximizes the evidence
  - This holds for any q
  - Variational method

ELBO for VAE

- Start with the log likelihood:
$$\log p_\theta(x) = l(q,\theta) + KL(q||p) \geq l(q,\theta)$$
- Let's choose q to be $q_\phi(z|x_i)$
- Then,
- $$l(\phi,\theta) = E_{q_\phi(z|x)} \log \frac{p_\theta(x,z)}{q_\phi(z|x)} = E_{q_\phi(z|x)} \log \frac{p_\theta(x|z)p_\theta(z)}{q_\phi(z|x)}$$
$$= E_{q_\phi(z|x)} \log p_\theta(x|z) \; + \; E_{q_\phi(z|x)} \log \frac{p_\theta(z)}{q_\phi(z|x)}$$
$$= E_{q_\phi(z|x)} \log p_\theta(x|z) \; - \; KL(q_\phi(z|x)||p_\theta(z))$$

likelihood / reconstruction error **prior**

- We maximize this lower bound $l(\phi,\theta)$ instead of the intractable marginal likelihood $\log p_\theta(z)$

- Likelihood/reconstruction error term
  - $E_{q_\phi(z|x)} \log p_\theta(x|z) \cong \frac{1}{L}\sum_l \log p_\theta(x|z_l) = \frac{1}{L}\sum_l \frac{-\|x-\mu(z_l)\|^2}{\sigma^2(z_l)}$,
  - L2 distance between two images x and μ

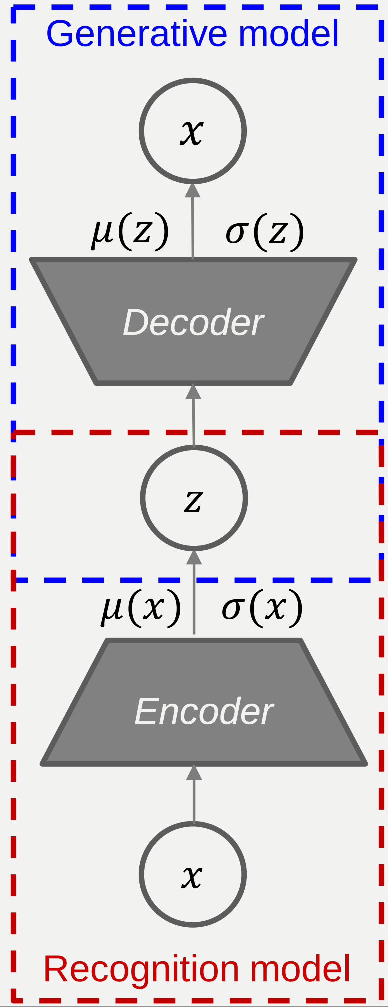
- Prior term
  - $KL(q_\phi(z|x)||p_\theta(z))$ computable in closed form (Was a HW problem)
  - $q = N(\mu(x), \sigma^2(x)I)$ and $p = N(0, I)$
  - And therefore $KL \cong \frac{1}{2}\sum_j\left(1 + \log\left(\sigma_j^2\right) - \mu_j^2 - \sigma_j^2\right)$

- Note that goodness of the bound is determined by
  - $KL\left(q_\phi(z|x)||p(z|x;\theta)\right)$

VAE loss revisited

Likelihood
/reconstruction error:
$E_{q_\phi(z|x)} \log p_\theta(x|z)$

Prior:
$-KL(q_\phi(z|x)||p_\theta(z))$



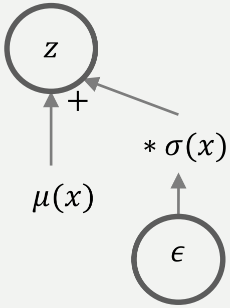
$p_\theta(x|z) = N(\mu(z), \sigma^2(z)I)$

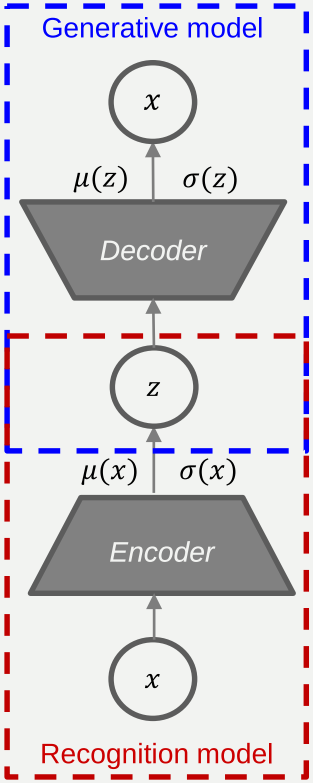
$p(z) = N(0, I)$

$q_\phi(z|x) = N(\mu(x), \sigma^2(x)I)$

Training

- Reparameterization trick
  - Sampling z from $q_\theta(z|x)$ is originally non-differentiable procedure
  - Trick: $z = \mu(x) + \sigma(x) * \epsilon, \quad \epsilon \sim N(0, I)$





  - Use same trick for sampling x from $p_\theta(x|z)$
- Can now train the model end-to-end with gradient descent using backpropagation

Training

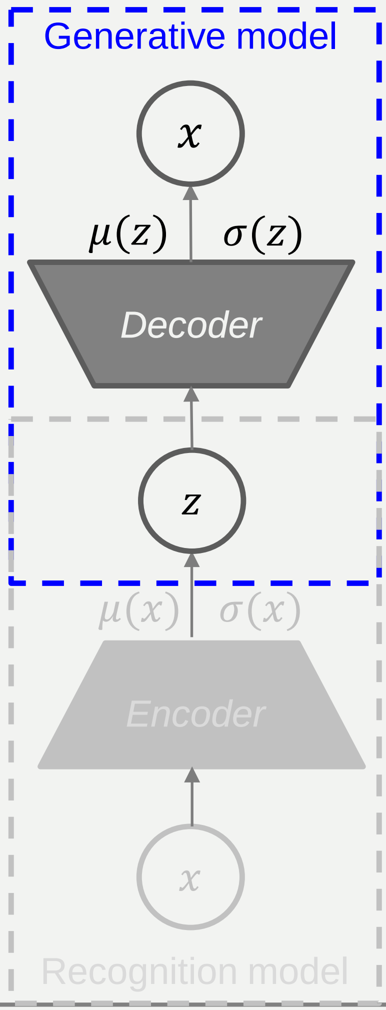
- Loss

$$l(\phi,\theta) = E_{q_\phi(z|x)} \log p_\theta(x|z) \; - \; KL(q_\phi(z|x)||p_\theta(z))$$

likelihood / reconstruction error **prior**

- Repeat
  - Sample *M* points $x_1, \cdots, x_M$
  - Sample *M* noise $\epsilon_1, \dots, \epsilon_M$ from $N(0, I)$
  - Perform forward- and backward-propagation
  - $\theta \leftarrow \theta - \nabla_\theta l(\phi,\theta)$
  - $\phi \leftarrow \phi - \nabla_\phi l(\phi,\theta)$

Generating data after training is finished

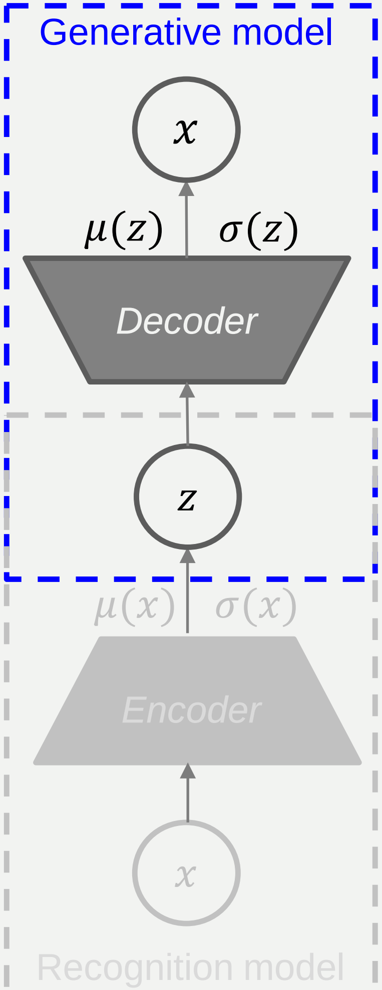


$p_\theta(x|z) = N(\mu(z), \sigma^2(z)I)$

$p(z) = N(0, I)$

$q(z|x;\phi) = N(\mu(x), \sigma^2(x)I)$

Generating data after training is finished

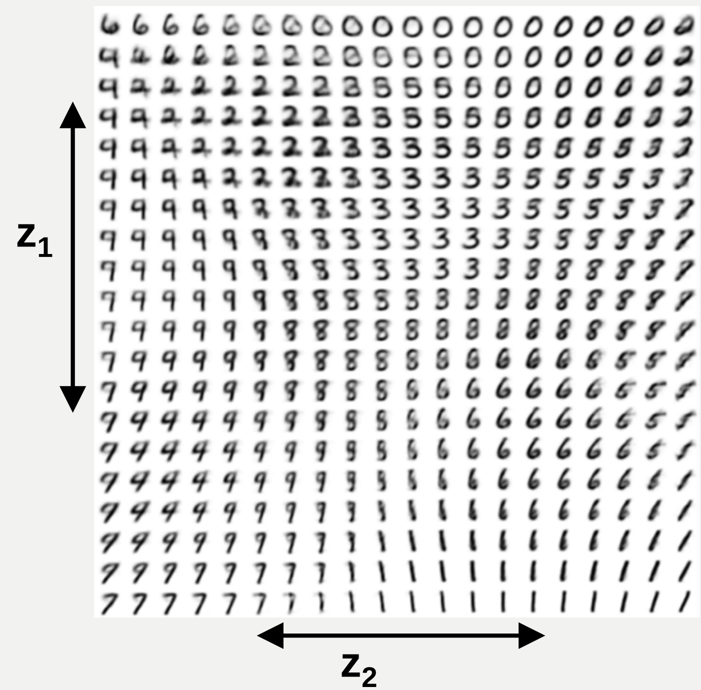


$p_\theta(x|z) = N(\mu(z), \sigma^2(z)I)$

$p(z) = N(0, I)$

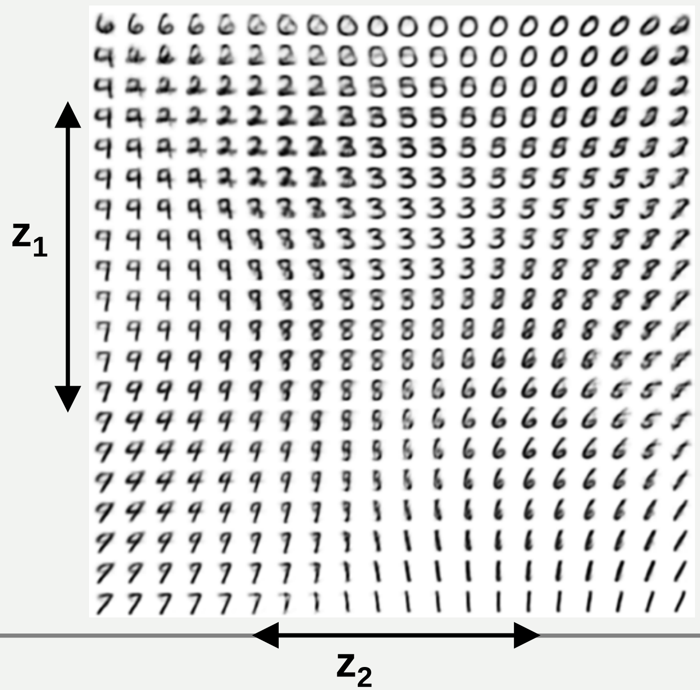
$q(z|x;\phi) = N(\mu(x), \sigma^2(x)I)$

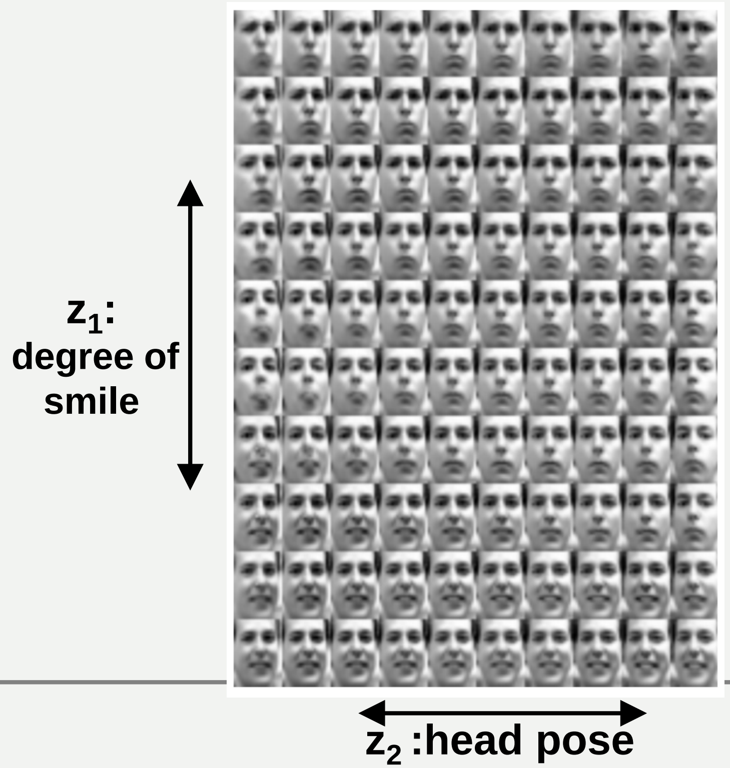
Data manifold for 2-d **z**



Kingma & Welling, 2014

- Diagonal prior on z
  - $z = (z_1, \dots, z_d) \sim N(0, I)$
  - z_i's are independent
  - Each z_i encode interpretable factors of variation





More examples

32x32 CIFAR-10

Labeled Faces in the Wild

VAE variants

- Using label information
  - Semi-supervised VAE [Kingma et al.,'14][Siddarth et al.,'17]
  - Structured-output VAE [Sohn et al.,'15]
  - Mixture of Gaussian VAE [Dilokthanakul et al.,'16][Gosh et al.,'19]
- Disentangling with β-VAE [Higgins et al.,'17]
  - $E_{q_\phi(z|x)} \log p_\theta(x|z) - \beta * KL(q_\phi(z|x)||p_\theta(z)), \quad \beta > 1$
- VQ-VAE [van den Oord et al.,'17], TD-VAE [Gregor et al.,'19]

Conditional VAE

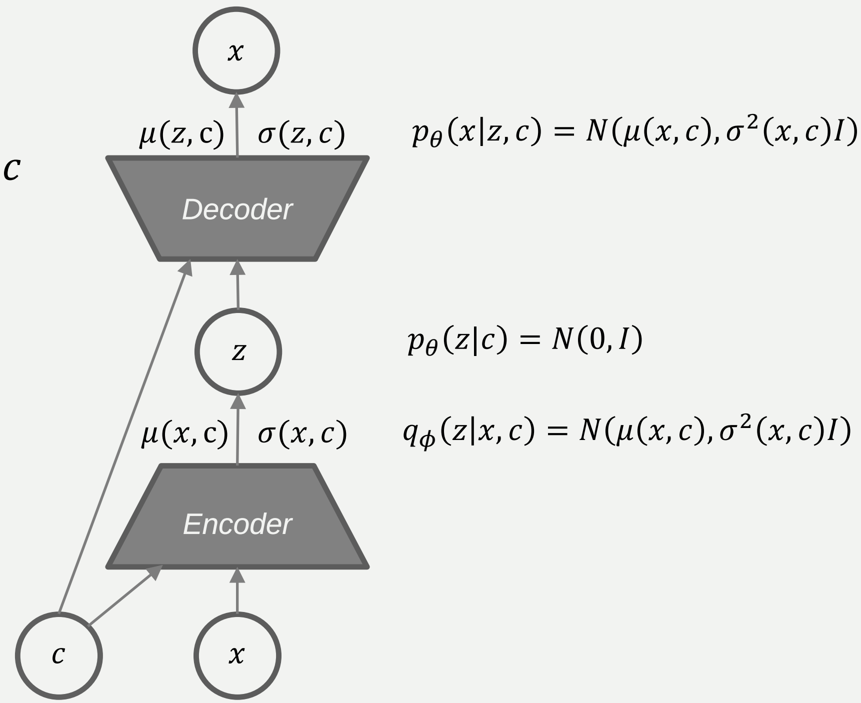
- Conditional VAE (supervised)
  - Label c observed during training/testing
  - Both encoder and decoder use the label c
  - $p_\theta(z|c) = N(0, I)$: same as before
  - Loss:

    $$\log p_\theta(x|c) \geq E_{q_\phi(z|x,c)} \log p_\theta(x|z,c) - KL(q_\phi(z|x,c)||p_\theta(z|c))$$

  - Example:

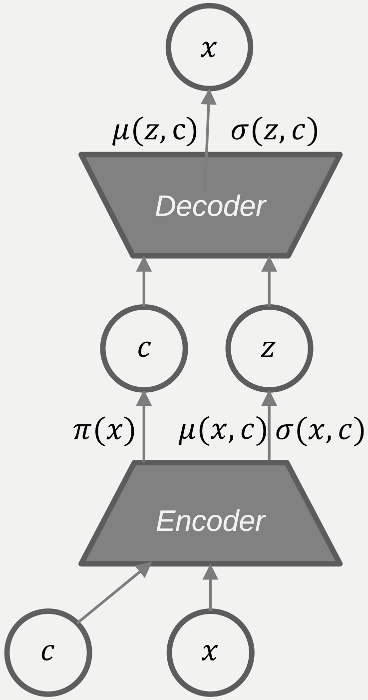
    (Cf. GAN vs C-GAN)



- "Conditional VAE" [Sohn et al.,'15]
  - Learn a map from image x to image y

Semi-supervised VAE

- Semi-supervised VAE (M2)
  - C-VAE cannot be used with mixed labeled and unlabeled examples
  - Decoder uses true/predicted label
    - $p_\theta(x|z,c\ ) = N(\mu(x,c), \sigma^2(x,c)I)$
  - Encoder predicts c and z
    - $q_\phi(z,c|x) = q_\phi(z|x)q_\phi(c|x)$
      $q_\phi\ (z|x,c) = N(\mu(x,c), \sigma^2(x,c)I)$
      $q_\phi(c|x) = Cat(\pi(x))$
  - Loss for labeled example
    - $\log p_\theta(x,c) \geq E_{q_\phi(z|x,c)}\ [\log p_\theta(x|c,z) + \log p_\theta(c) + \log p(z) - \log q_\phi(z|x,c)]$
  - Loss for unlabeled example
    - $\log p_\theta(x) \geq E_{q_\phi(z,c|x)}\left[\log p_\theta(x|c,z) + \log p_\theta(y) + \log p(z) - \log q_\phi(z,c|x)\right]$
  - Several models possible
    - [Kingma et al.,'14][Siddharth et al.,'17], also see https://pyro.ai/examples/ss-vae.html



$p_\theta(x|z,c) = N(\mu(x,c), \sigma^2(x,c)I)$

$p_\theta(z) = N(0,I)$
$p_\theta(c) = Cat(\pi)$

$q_\phi(z,c|x) = q_\phi(z|x)q_\phi(c|x)$
$q_\phi(z|x,c) = N(\mu(x,c), \sigma^2(x,c)I)$
$q_\phi(c|x) = Cat(\pi(x))$

Mixture of Gaussian VAE for classification

- MoG VAE [Gosh et al.,'19]
  - Label c observed during training only
  - Decoder predicts x from z
    - $p_\theta(x, c|z) = p_\theta(x|z)p_\theta(c|z)$
    - $p_\theta(z) = \sum_c w_c\, N(\mu_c, \sigma_c^2 I)$: mixture of Gaussians
    - One Gaussian per class, centered at fixed μ_c
  - Loss

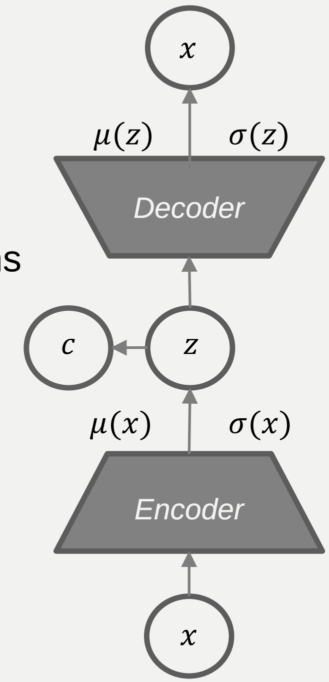
$$\log p_\theta(x, c) \geq E_{q(z|x;\phi)} \log p_\theta(x, c|z) - KL(q_\phi(z|x)||p_\theta(z))$$

  - Test time: approximate classification
    - $\operatorname{argmax}_c p_\theta(c|x) = \operatorname{argmax}_c p_\theta(x, c)$

$$= \operatorname{argmax}_c \int p_\theta(x, c|z)p_\theta(z)dz$$

$$= \cdots = \operatorname{argmax}_c \int p_\theta(z|x)p_\theta(c|z)dz$$

$$\cong \operatorname{argmax}_c E_{q_\phi(z|x)}\, p_\theta(c|z) \cong \operatorname{argmax}_c \frac{1}{M}\sum_i p_\theta(c|z_i)$$ (use Monte-Carlo sampling again)



$p_\theta(x|z) = N(\mu(z), \sigma^2(z)I)$

$p_\theta(z|c) = N(\mu_c, I)$ where μ_c is one-hot encoding

$p_\theta(c|z) = \frac{p_\theta(z|c)p_\theta(c)}{\sum_c p_\theta(z|c)p_\theta(c)}$

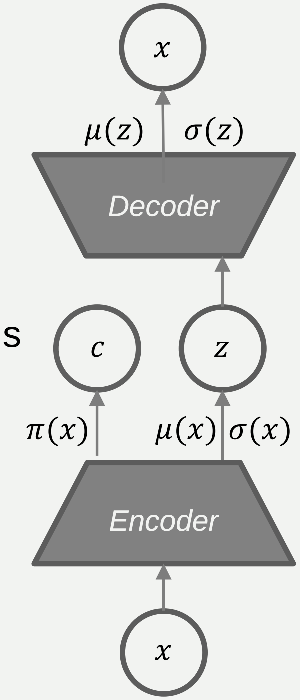
$q_\phi(z|x) = N(\mu(x), \sigma^2(x)I)$

Mixture of Gaussian VAE for clustering

- MoG VAE [Zhao et al.,'19]
  - Unsupervised learning
  - Label/cluster number c not observed
  - Decoder predicts x from z
    - $p_\theta(x,z|c) = p_\theta(x|z)p_\theta(z|c)$
    - $p_\theta(z) = \sum_c w_c\, N(\mu_c, \sigma_c^2 I)$: mixture of Gaussians
    - One Gaussian per class, centered at μ_c
  - Encoder predicts c and z from x
    - $q_\phi(z,c|x) = q_\phi(z|x)q_\phi(c|x)$
    - $q_\phi(z|x) = N(\mu(x), \sigma^2(x)I)$
    - $q_\phi(c|x) = Cat(\pi(x))$
  - Loss

$$\log p_\theta(x) \geq \sum_k q_\phi(c^k|x) \log \frac{\pi^k}{q_\phi(c^k|x)} + \sum_k q_\phi(c^k|x) l^k(x),$$

where $l^k(x) = E_{q_\phi(z|x)}\, p_\theta(x|z) - KL(q_\phi(z|x) || \, p_\theta(z|c^k))$



$p_\theta(x|z) = N(\mu(x), \sigma^2(x)I)$

$p_\theta(z|c) = N(\mu_c, I)$
$p_\theta(c) = Cat(\pi)$
$p_\theta(c|z) = \frac{p(Z|C)p(c)}{\sum_c p(Z|C)p(c)}$

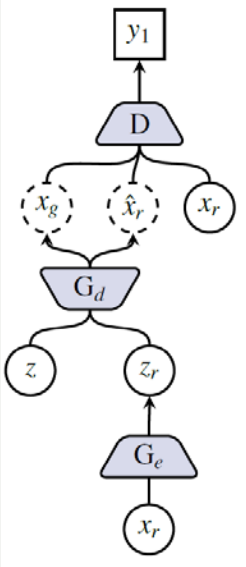
$q_\phi(z,y|x) = q_\phi(z|x)q_\phi(c|x)$
$q_\phi(z|x) = N(\mu(x), \sigma^2(x)I)$
$q_\phi(c|x) = Cat(\pi(x))$

VAE vs GAN

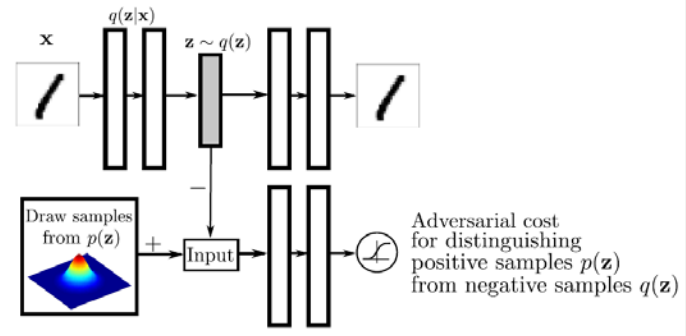
- Pros of VAE
  - Principled approach to generative data
  - Can use all previous knowledge about probabilistic models
  - Allows inference of $q_\phi(z|x)$, can be useful feature representation for other tasks
- Cons of VAE
  - Maximizes lower bound of likelihood. Cannot evaluate the likelihood directly
  - Samples blurrier and lower quality compared to state-of-the-art (GANs)

VAE+GAN

- VAEGAN [Larsen et al.,'16]
  - Pixelwise likelihood term $E_q \log p(x|z)$ causes blurriness
  - Use GAN to measure whole-image likelihood
  - VAE: ELBO = L_{prior} + $L_{pixel\text{-}likelihood}$
  - VAEGAN: ELBO = L_{prior} + $L_{disc\text{-}likelihood}$ + L_{GAN}



- Adversarial autoencoder
  - [Makehazani et al.,'14]
  - Unsupervised/supervised
  - Semi-supervised
  - Dimensionality-reduction



- Energy-based GAN [Zhao et al.,'17], …

References

- Kingma, D., Welling, M. (2014) Auto-Encoding Variational Bayes, ICLR
- Kingma, D., Rezende, D., Mohamed, S., Welling, M. (2014) Semi-Supervised Learning with Deep Generative Models, NIPS
- Sohn, K., Lee, H., & Yan, X. (2015) Learning structured output representation using deep conditional generative models, NIPS
- Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B. (2015) Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*
- Dilokthanakul, N., Mediano, P. A., Garnelo, M., Lee, M. C., Salimbeni, H., Arulkumaran, K., & Shanahan, M. (2016). Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*.
- Larsen, A., Sønderby, A., Larochelle, H., Winther, O. (2016) Autoencoding beyond pixels using a learned similarity metric, ICML
- Zhao, J., Mathieu, M., & LeCun, Y. (2016). Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*.
- Siddharth, N., Paige, B., Van de Meent, J. W., Desmaison, A., Goodman, N., Kohli, P., ... & Torr, P. (2017). Learning disentangled representations with semi-supervised deep generative models. NIPS
- Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., & Lerchner, A. (2018). Understanding disentangling in beta-VAE. *arXiv preprint arXiv:1804.03599*
- Zhao, Q., Honnorat, N., Adeli, E., Pfefferbaum, A., Sullivan, E. V., & Pohl, K. M. (2019, June). Variational Autoencoder with Truncated Mixture of Gaussians for Functional Connectivity Analysis, IPMI
- Ghosh, P., Losalka, A., & Black, M. J. (2019, July). Resisting adversarial attacks using gaussian mixture variational autoencoders. AAAI