Generative Models

Shihua Zhang

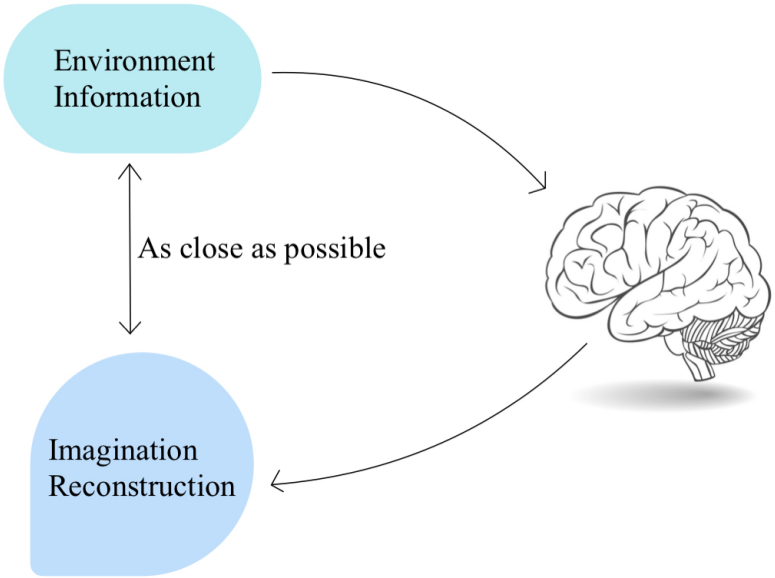
December 19, 2024

Outline

Generative Models: Intuition

Human's learning process can be roughly divided into two parts:

- Recognition Process:
  Observing objects and taking in new information.
- Imagination Process:
  Recalling information and reconstructing them in mind.



These processes can be simulated by a structure, called generative model, with compatible precision and generality.

Generative Models

- Aim: Generate new contents (pictures, texts, codes, etc.).
- A two-step solution:
  - Learn a distribution from the observed data
  - Sample from the learned distribution

Statistical Models: Early Generative Models

Maximum likelihood: Given a family of hypothesis distributions $\{p_\theta(x)\}$ (such as exponential family of distributions) and a set of data $\{x_i\}$, one can optimize the log-likelihood to learn the distribution:

$$\max_\theta \frac{1}{n} \sum_i \log p_\theta(x_i) \sim \max \int p_{real}(x) \log p_\theta(x)\, dx \sim \min KL(p_{real} | p_\theta)$$

Statistical Models: Early Generative Models

Maximum likelihood: Given a family of hypothesis distributions $\{p_\theta(x)\}$ (such as exponential family of distributions) and a set of data $\{x_i\}$, one can optimize the log-likelihood to learn the distribution:

$$\max_\theta \frac{1}{n}\sum_i \log p_\theta(x_i) \sim \max \int p_{real}(x) \log p_\theta(x)\, dx \sim \min KL(p_{real}|p_\theta)$$

Sampling: To sample from the learned distribution, one can use Markov chain Monte Carlo (MCMC) method or other methods.

Neural Networks Are Satisfying Candidates

Observation:

- Statistical models approximate the unknown distribution p_{real} by a family of hypothesis distributions $\{p_\theta\}$.
- One need a hypothesis distribution that sufficiently large to finely approximate p_{real}.

Neural Networks Are Satisfying Candidates

Observation:

- Statistical models approximate the unknown distribution p_{real} by a family of hypothesis distributions $\{p_\theta\}$.
- One need a hypothesis distribution that sufficiently large to finely approximate p_{real}.

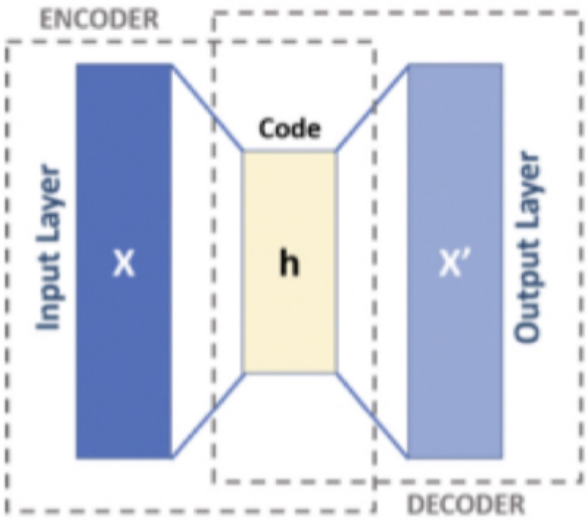
Question: Where find this "large" family of distributions?
Answer: Neural network methods are satisfying candidates:

- Neural networks are universal approximators.
- There are many powerful optimizers available for neural networks.

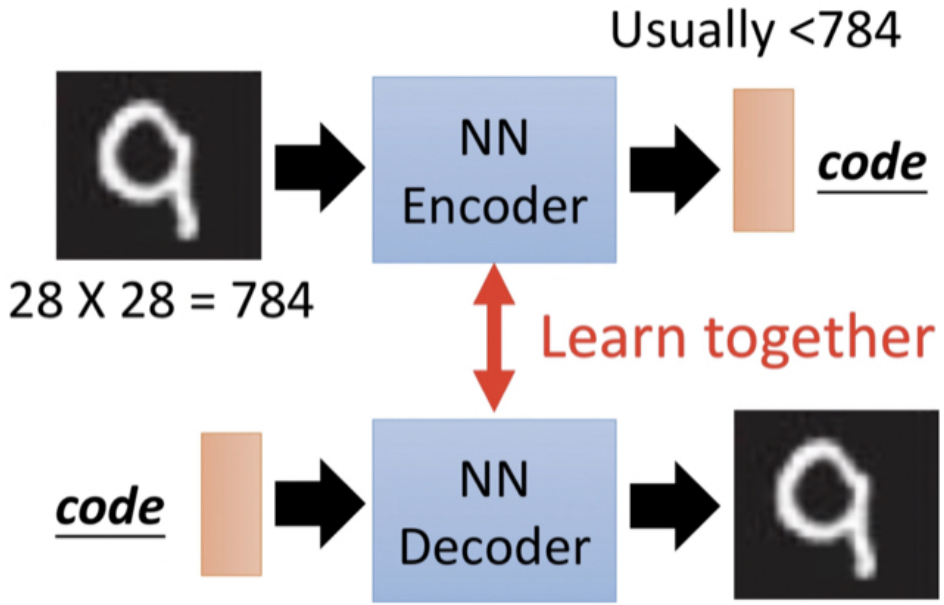
Generative Models: Early Attempts

Autoencoder (AE) can reconstruct input data.

- The simplest AE is a neural network with one hidden layer.



Schema of a basic Autoencoder



Generative Model: Early Attempts

- AE reconstructs its inputs by minimizing the difference between inputs and outputs.
  - Encoder & Decoder:

$$\phi : X \to Z$$
$$\psi : Z \to X$$
$$\phi, \psi = \text{argmin}|x - (\psi \circ \phi)x|$$

Generative Model: Early Attempts

- AE reconstructs its inputs by minimizing the difference between inputs and outputs.
  - Encoder & Decoder:

$$\phi : X \to Z$$
$$\psi : Z \to X$$
$$\phi, \psi = \text{argmin}|x - (\psi \circ \phi)x|$$

Specify it as a NN:

$$h = \sigma(Wx + b)$$
$$x' = \sigma'(W'h + b')$$

Generative Model: Early Attempts

- AE reconstructs its inputs by minimizing the difference between inputs and outputs.
  - Encoder & Decoder:

$$\phi : X \to Z$$
$$\psi : Z \to X$$
$$\phi, \psi = \text{argmin}|x - (\psi \circ \phi)x|$$

Specify it as a NN:

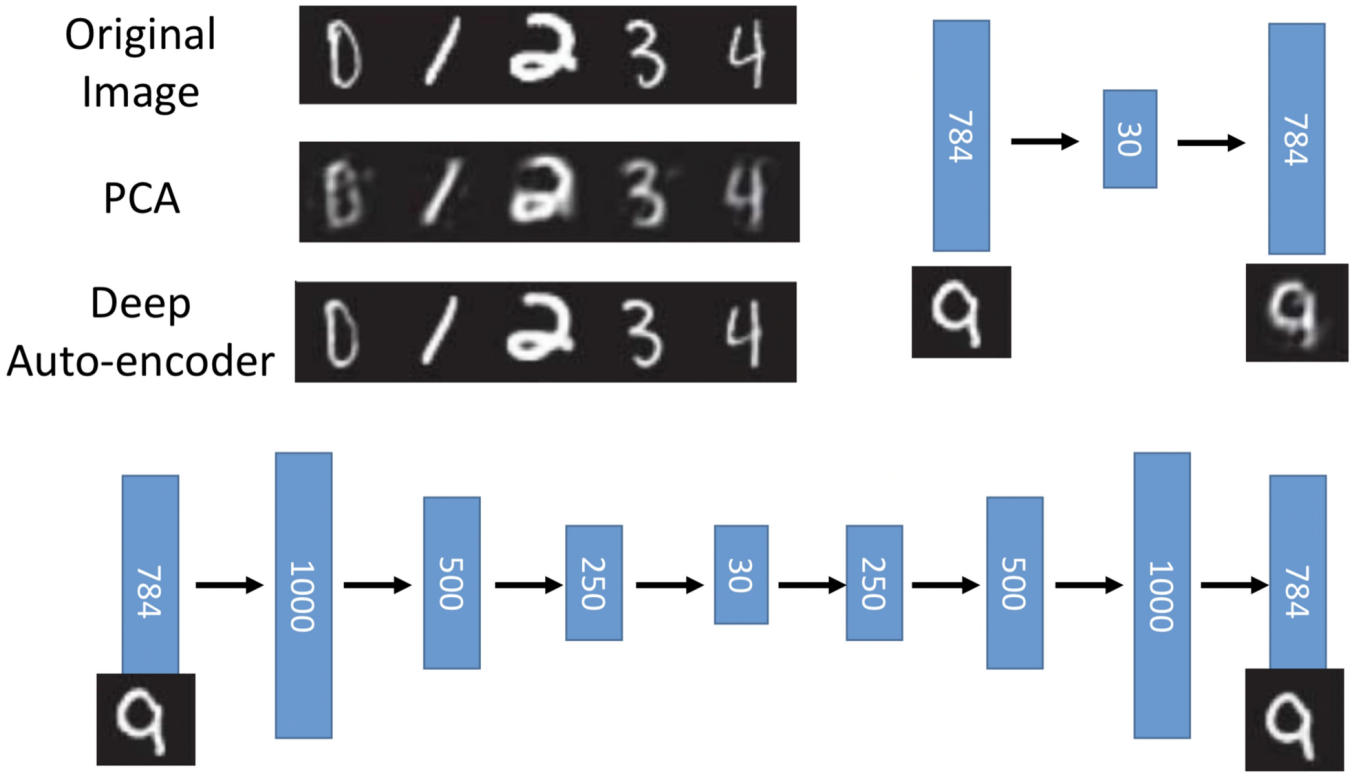
$$h = \sigma(Wx + b)$$
$$x' = \sigma'(W'h + b')$$

  - Reconstruction Error:

$$L(x, x') = |x - x'|^2 = |x - \sigma'(W'\sigma(Wx + b) + b')|^2$$

Generative Model: Early Attempts

- Experimental Result



- Remark: AE is only capable of reconstructing data, and its generative ability is relatively weak.

Timeline: Deep Generative Models

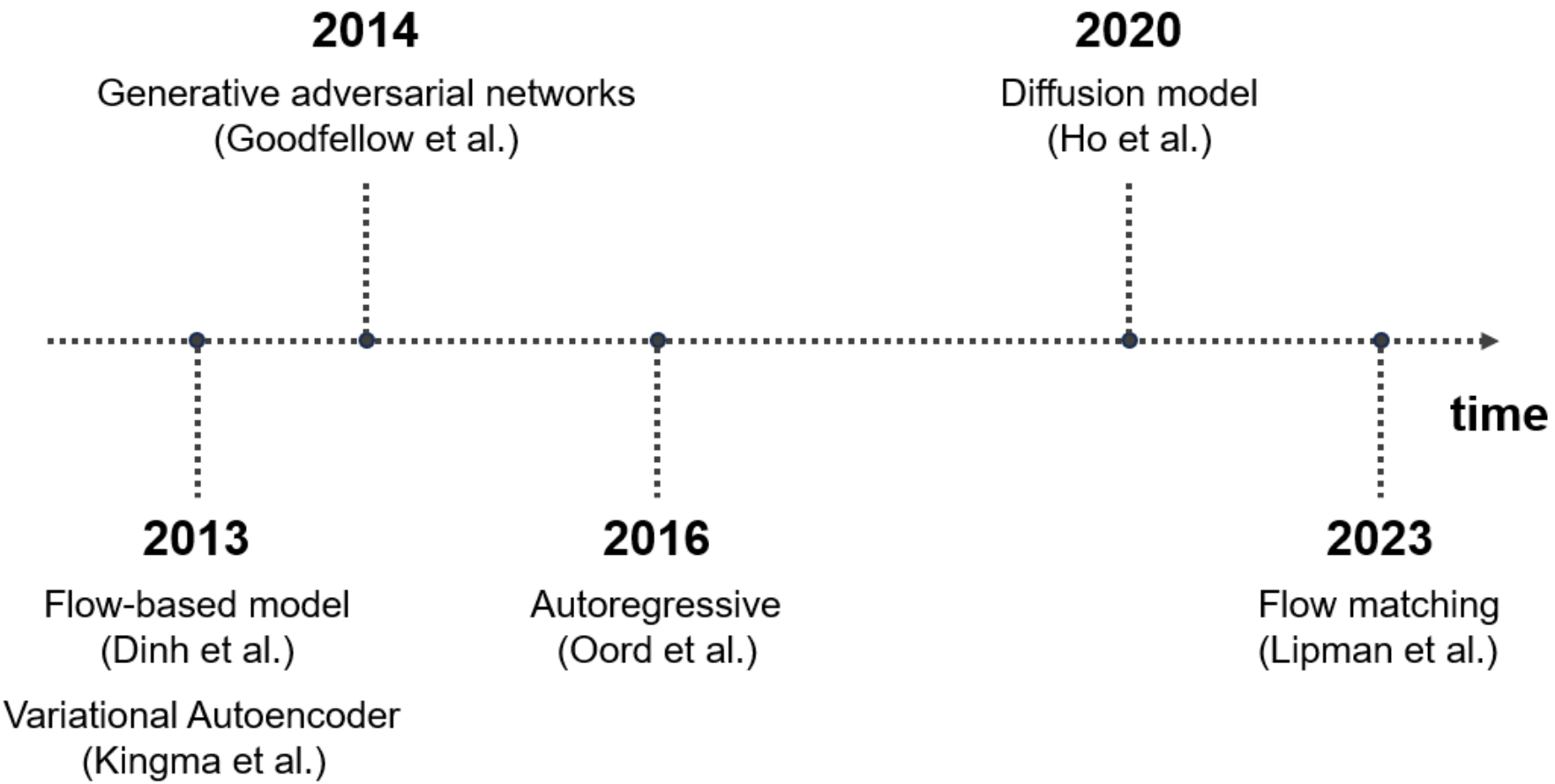


Figure: Timeline of deep generative models

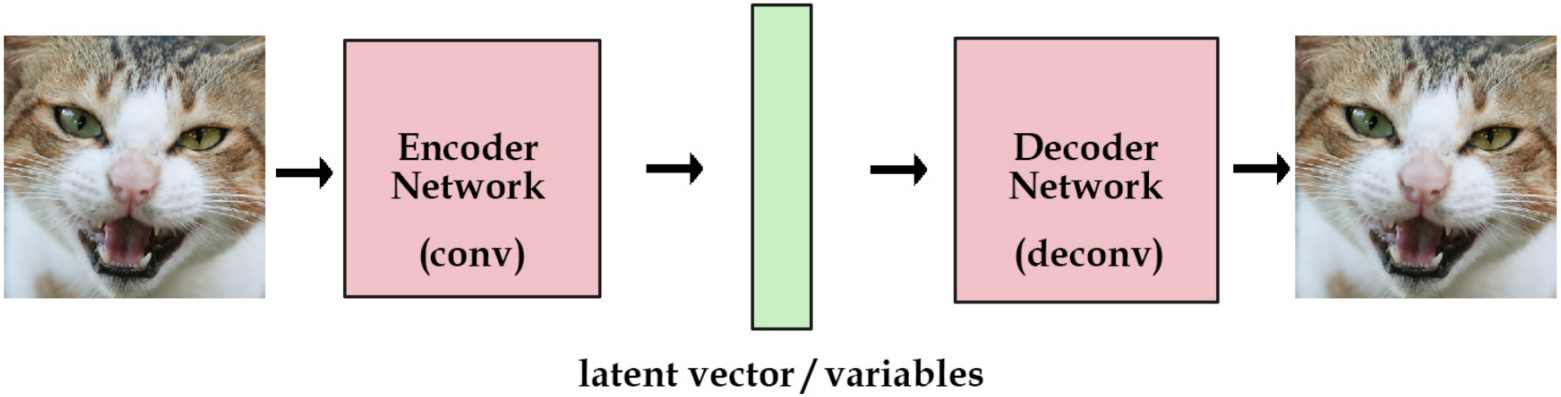
Outline

[1] Kingma D.P., Welling M. Auto-Encoding Variational Bayes. ICLR, 2014.

[2] Goodfellow I.J., et al. Generative Adversarial Networks. NIPS, 2014.

Variational Autoencoder (VAE)

- Variational Autoencoder (VAE) originates from AE.
- VAE has stronger generative ability.



左：第1世代，中：第9世代，右：原始图像

Construct Hypothesis Distributions by Neural Networks

- A generator network induces a distribution:

$$p_\theta(x) = \int p(z) p_\theta(x|z)\, dz$$

where $p(z)$ is a prior distribution in latent space.

Construct Hypothesis Distributions by Neural Networks

- A generator network induces a distribution:

$$p_\theta(x) = \int p(z) p_\theta(x|z)\, dz$$

  where $p(z)$ is a prior distribution in latent space.

- Problem: if z is low-dimensional and the decoder is deterministic, then $p_\theta(x) = 0$ almost everywhere!
  - The model only generates samples over a low-dimensional sub-manifold of $\mathcal{X}$.

Construct Hypothesis Distributions by Neural Networks

- A generator network induces a distribution:

$$p_\theta(x) = \int p(z) p_\theta(x|z)\, dz$$

  where $p(z)$ is a prior distribution in latent space.

- Problem: if z is low-dimensional and the decoder is deterministic, then $p_\theta(x) = 0$ almost everywhere!
  - The model only generates samples over a low-dimensional sub-manifold of $\mathcal{X}$.
- Solution: define a noisy observation model, for example:

$$p_\theta(x|z) = \mathcal{N}(x; G_\theta(z), \eta I)$$

  where G_θ is the function computed by the decoder with parameters θ.

Variational Inference

Goal:

$$\max_{\theta} \log p_\theta(x), \text{ i.e. } \max_{\theta} \log \int p(z) p_\theta(x|z)\, dz$$

However, computing $p_\theta(x)$ directly is intractable.

Variational Inference

Goal:

$$\max_{\theta} \log p_\theta(x), \text{ i.e. } \max_{\theta} \log \int p(z) p_\theta(x|z)\, dz$$

However, computing $p_\theta(x)$ directly is intractable.

Idea: Introduce a variational distribution $q_\psi(z|x)$ to approximate the posterior $p_\theta(z|x)$, and use Jensen's Inequality to derive a lower bound on $\log p_\theta(x)$.

Variational Inference

Goal:

$$\max_{\theta} \log p_\theta(x), \text{i.e. } \max_{\theta} \log \int p(z) p_\theta(x|z)\, dz$$

However, computing $p_\theta(x)$ directly is intractable.

Idea: Introduce a variational distribution $q_\psi(z|x)$ to approximate the posterior $p_\theta(z|x)$, and use Jensen's Inequality to derive a lower bound on $\log p_\theta(x)$.

$$\log p_\theta(x) = \log \int p(z) p_\theta(x|z)\, dz = \log \int q_\psi(z|x) \frac{p(z)}{q_\psi(z|x)} p_\theta(x|z)\, dz$$

$$\geqslant \int q_\psi(z|x) \log \left[\frac{p(z) p_\theta(x|z)}{q_\psi(z|x)} \right] dz \quad \text{(Jensen's Inequality)}$$

$$= \mathbb{E}_{q_\psi(z|x)}[\log p_\theta(x|z)] + \mathbb{E}_{q_\psi(z|x)}[\frac{\log p(z)}{\log q_\psi(z|x)}] \, (\text{ELBO})$$

Let's look at these two terms in turn.

Understanding the Role of Two Terms

- Let's look at the first term $\mathbb{E}_{q_\psi(z|x)}[\log p_\theta(x|z)]$.
- Since we assumed a Gaussian observation model:

$$\log p_\theta(x|z) = \log \mathcal{N}(x; G_\theta(z), \eta I)$$

$$= \log \left[\frac{1}{(2\pi\eta)^{D/2}} \exp\left(-\frac{1}{2\eta} \|x - G_\theta(z)\|^2 \right) \right]$$

$$= -\frac{1}{2\eta} \|x - G_\theta(z)\|^2 + \text{const.}$$

- This term is the expected squared error in reconstructing x from z.
- Let's call it the **reconstruction term**.

Understanding the Role of Two Terms

- The second term is $\mathbb{E}_{q_\psi(z|x)}\left[\log \frac{p(z)}{q_\psi(z|x)}\right]$.
- This is $-D_{\mathrm{KL}}(q_\psi(z|x)\|p(z))$, where D_{KL} is the Kullback-Leibler (KL) divergence.

$$D_{\mathrm{KL}}(q_\psi(z|x)\|p(z)) \triangleq \mathbb{E}_{q_\psi(z|x)}\left[\log \frac{q_\psi(z|x)}{p(z)}\right]$$

- KL divergence is a widely used measure of distance between probability distributions, though it doesn't satisfy the axioms to be a distance metric.
- Typically, $p(z) = \mathcal{N}(0, I)$. Hence, the KL term encourages q to be close to $\mathcal{N}(0, I)$.

Understanding the Role of Two Terms

- Let's think about the role of each of the two terms.
- The reconstruction term:

$$\mathbb{E}_q[\log p(x|z)] = -\frac{1}{2\sigma^2}\mathbb{E}_q[\|x - G_\theta(z)\|^2] + \text{const}$$

  is minimized when q is a point mass on

$$z_* = \arg\min_z \|x - G_\theta(z)\|^2.$$

- But a point mass would have infinite KL divergence (Exercise: check this). So the KL term forces q to be more spread out.

Variational Inference

- Try to maximize the variational lower bound, or variational free energy:

$$\log p_\theta(x) \geqslant \mathcal{F}(\theta, \psi) = \mathbb{E}_{q_\psi(z|x)}[\log p_\theta(x|z)] - D_{\mathrm{KL}}(q_\psi(z|x)\|p(z)).$$

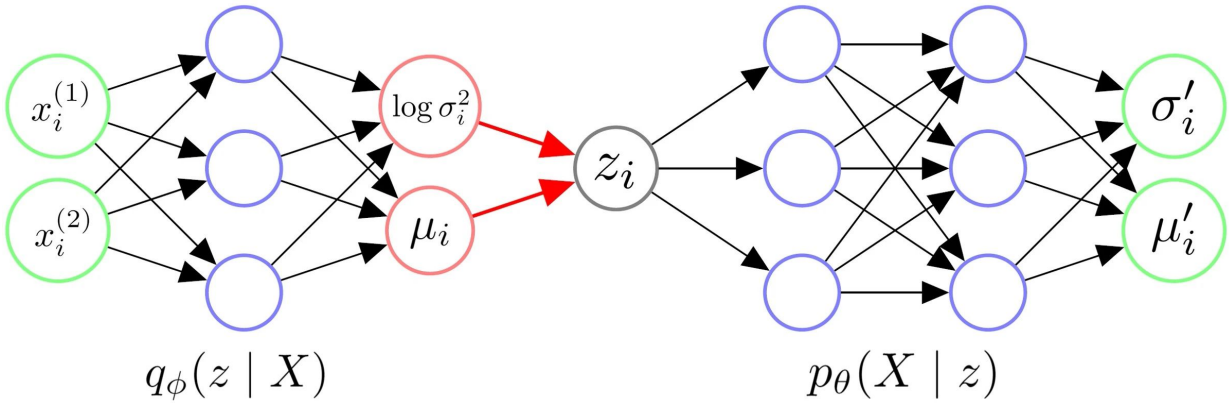
- The term "variational" is a historical accident: "variational inference" used to be done using variational calculus, but this isn't how we train VAEs.
- One would choose q_ψ to make the bound as tight as possible.
- It is possible to show that the gap is given by:

$$\log p(x) - \mathcal{F}(\theta, \psi) = D_{\mathrm{KL}}(q_\psi(z|x)\|p_\theta(z|x)).$$

- Therefore, q_ψ should be as close as possible to the posterior distribution $p_\theta(z|x)$.

VAE: Realization by Neural Networks

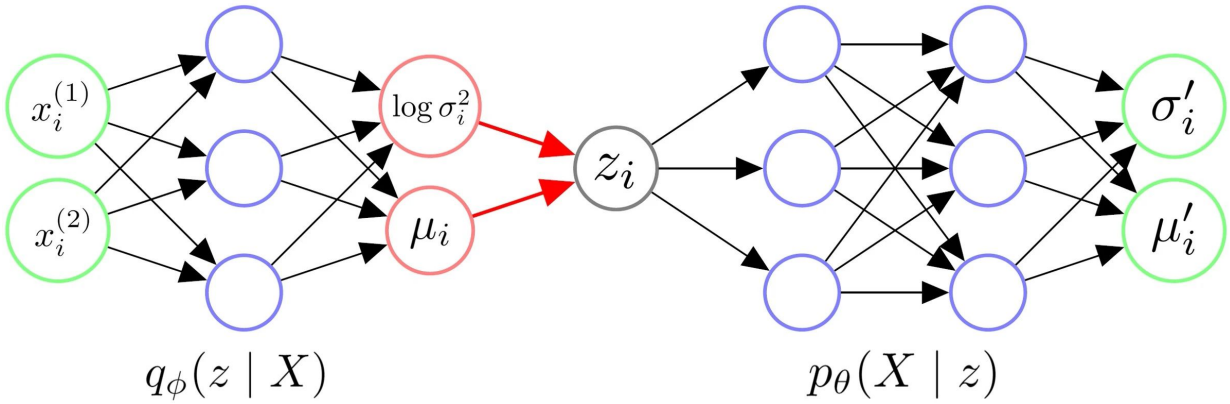
- Basic Encoder-Decoder structure:



When training: regularization term is integrable and error term can be replaced be the Euclidean distance.
When generating: draw samples from $P(Z)$.

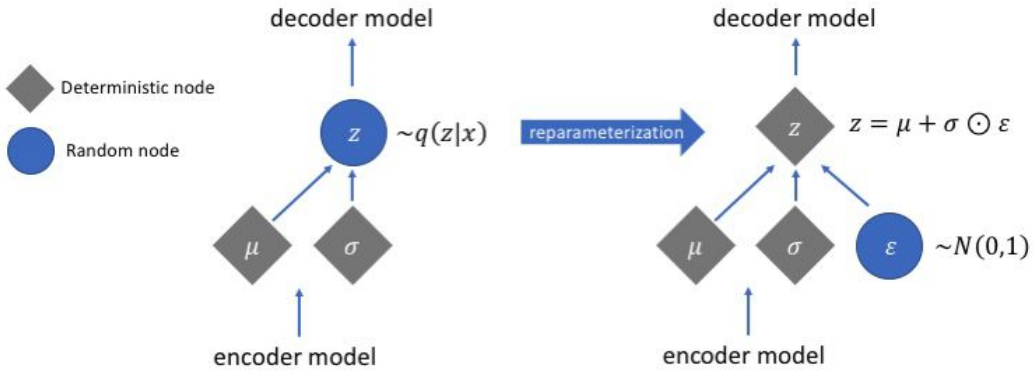
VAE: Realization by Neural Networks

- Basic Encoder-Decoder structure:



When training: regularization term is integrable and error term can be replaced be the Euclidean distance.
When generating: draw samples from $P(Z)$.

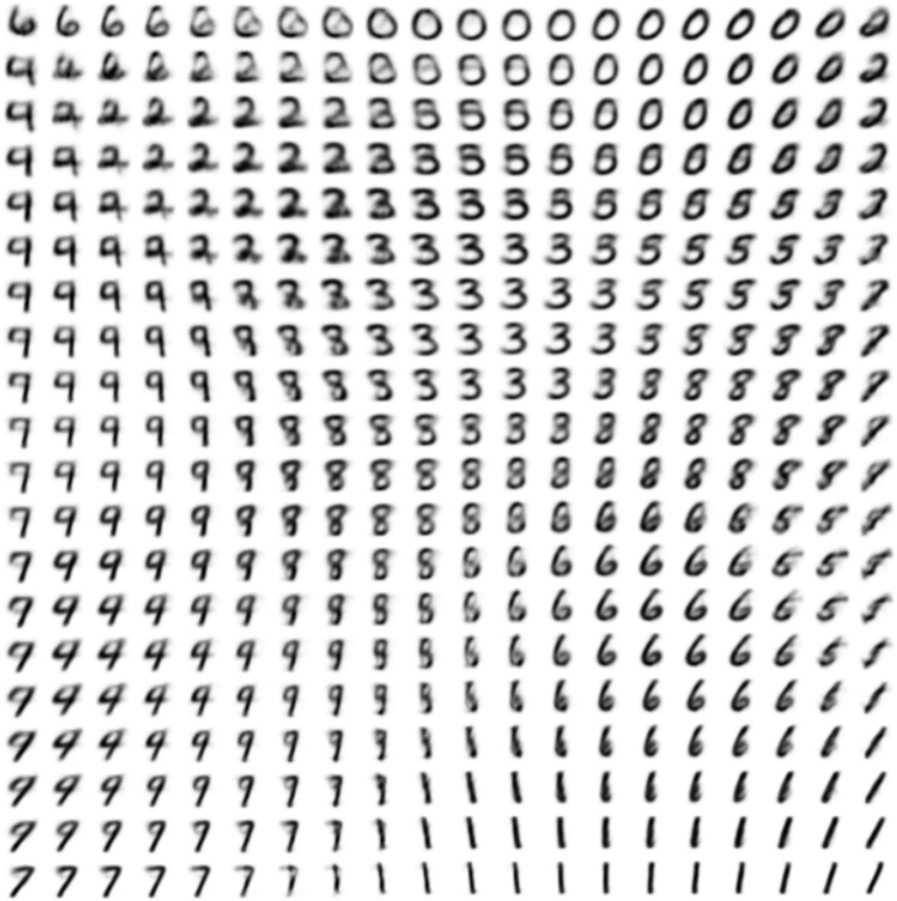
- Reparametrision Trick:



VAE: Experimental Result

- Genertes new data

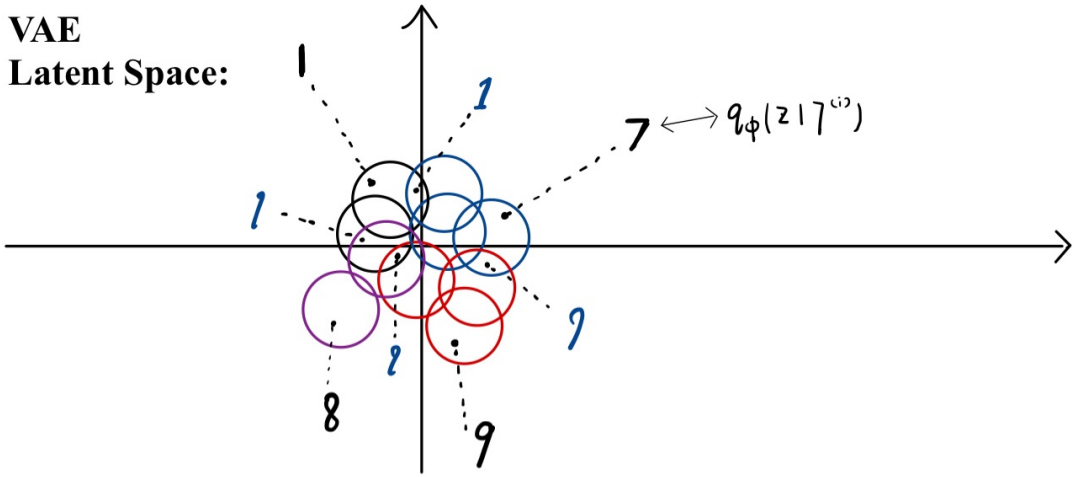
(a) Learned Frey Face manifold

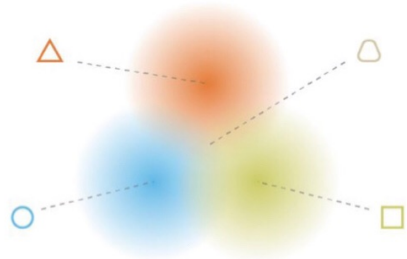


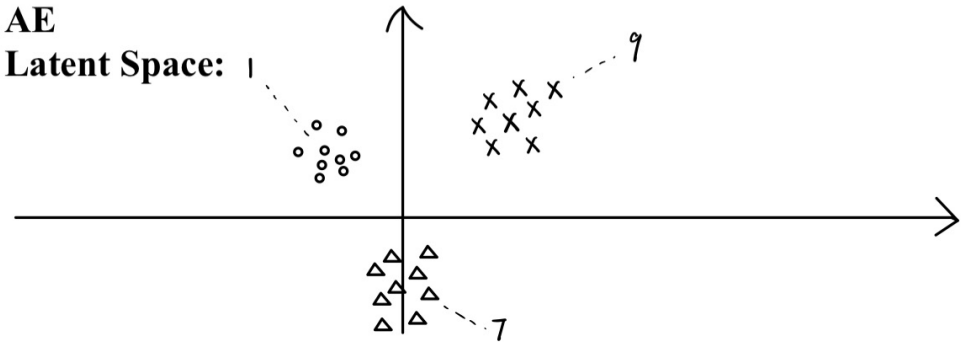
(b) Learned MNIST manifold

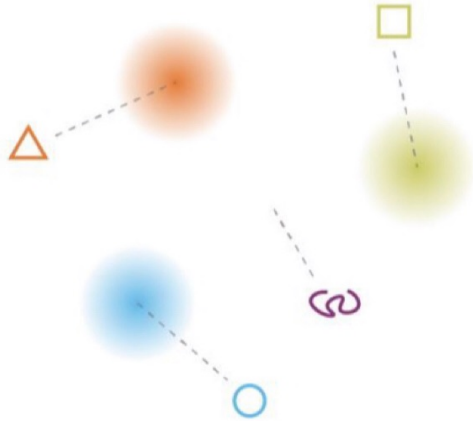
VAE: Experimental Result

- Explanation of Experimental Result









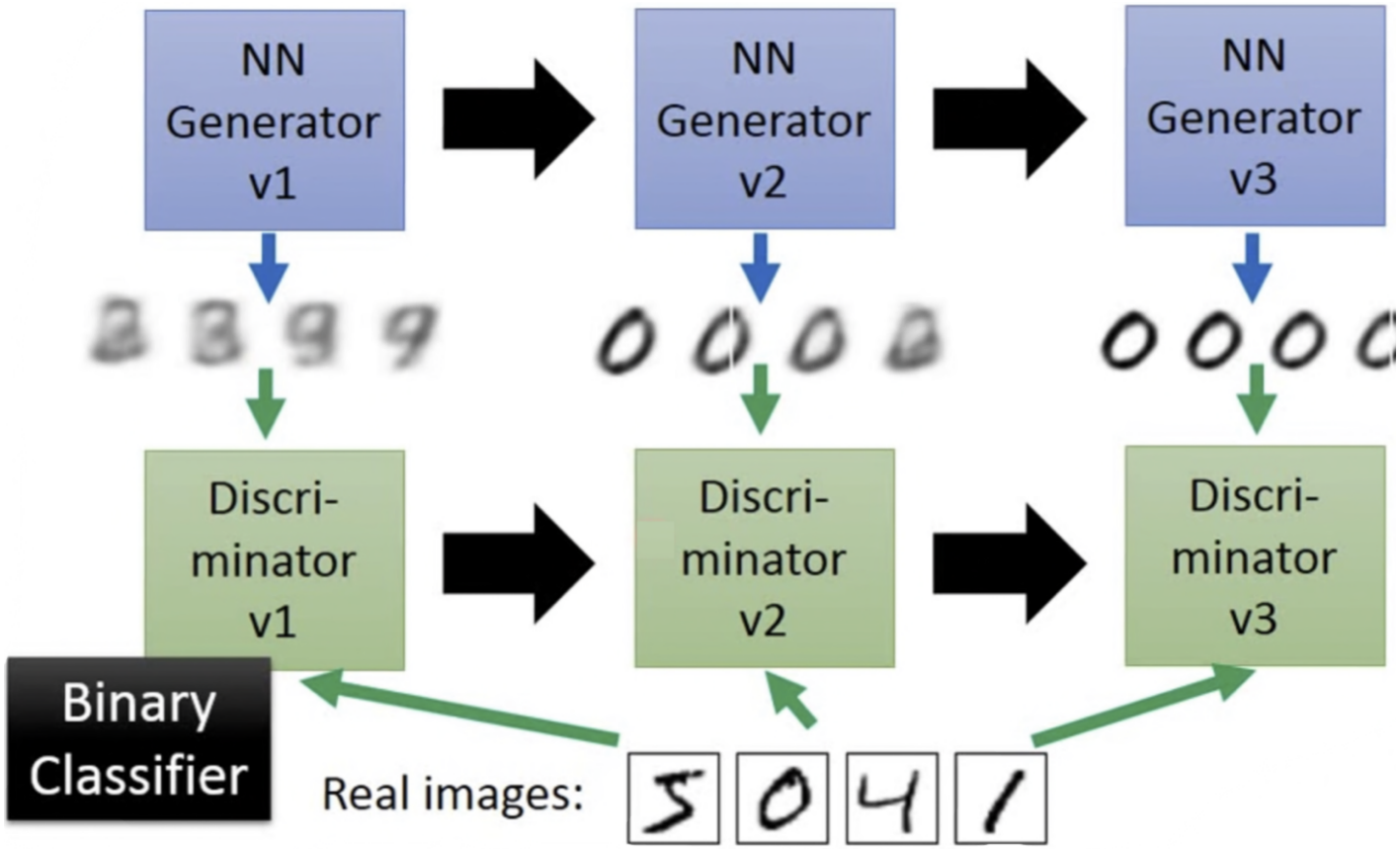
VAE: Problem

- Contrary to AE, VAE perform well on generality rather than precision. For instance, the images generated by VAE are often vague.

- GAN provides an new way to grant both precision and generality.

GAN: Intuition

- Given a set of real images.
- Generator: sample (fake) pictures (from the learned distribution).
- Discriminator: distinguishes these fake images from the true ones.



GAN: Theorical Analysis

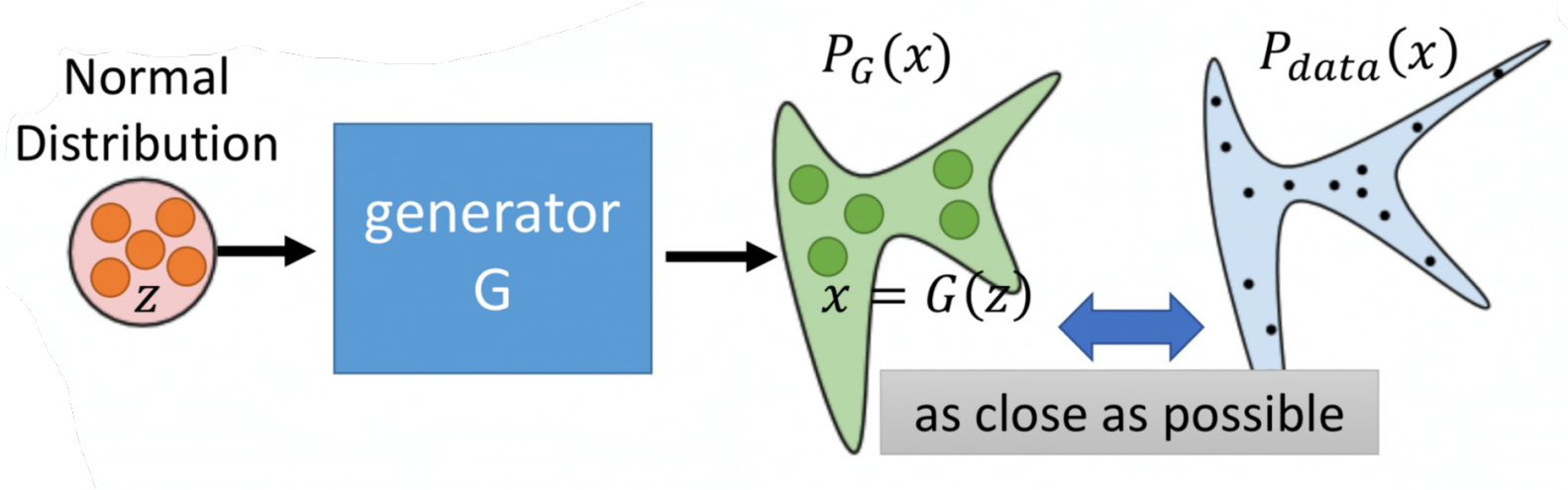
- Generator: A map G from random noise space Z to data space X.

$$G: \ Z \longrightarrow X, \ \text{random noise sample } z \mapsto \text{generated data } G(z)$$

- Discriminator: A valuation funtion D from X to a score ranged in [0, 1].

$$D: \ X \longrightarrow [0, \ 1], \quad x_{true} \xrightarrow{\text{close}} 1, \ x_{fake} \xrightarrow{\text{close}} 0$$

- Cheat-Distinguish Process



GAN: Theorical Analysis

- Optimization Objective

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

Intuitively: D good $\uparrow$, G good $\downarrow$

- Key Insight
  - For a fixed generator G, the optimal discriminator D is:

$$D_G^*(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)}$$

This balances the likelihood of real data $p_{\text{data}}(x)$ and generated data $p_g(x)$.

GAN: Theoretical Analysis

- Intuition for the Minimax Process
  - Discriminator's Perspective (Maximization):

$$V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{x \sim p_g}[\log(1 - D(x))]$$

For a fixed G, D achieves maximum when:

$$D_G^*(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)}$$

Intuitively: D estimates the probability that x is real.

  - Generator's Perspective (Minimization):

$$C(G) = \max_D V(D, G)$$

The global minimum is achieved when $p_g = p_{\text{data}}$, as this makes D unable to distinguish real and generated data.

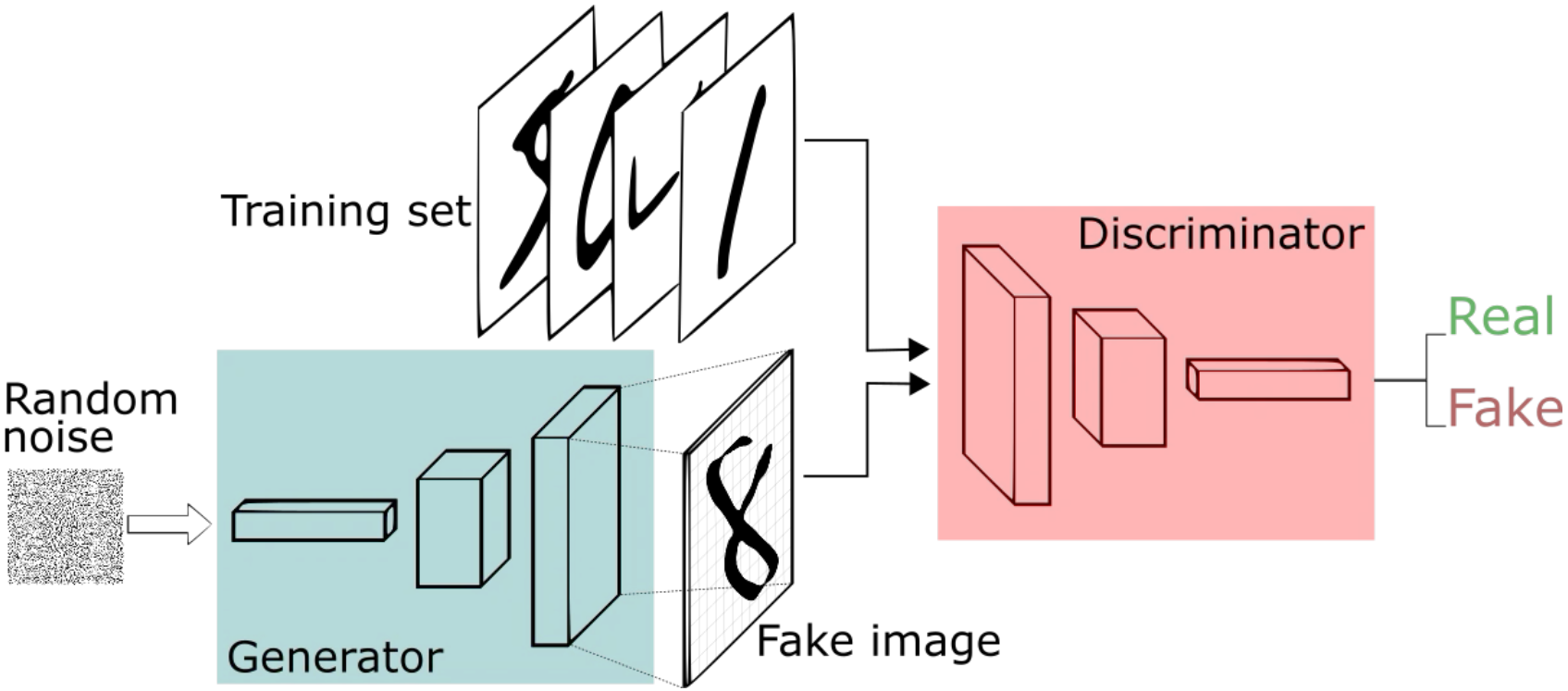
GAN: Theoretical Analysis

- Divergence Perspective
  - The loss function for the generator $C(G)$ can be rewritten as:

$$C(G) = -\log 4 + 2 \cdot \mathrm{JS}(p_{\text{data}} \| p_g)$$

    JS: Jensen-Shannon divergence, which measures the similarity between two distributions.
    Intuition: Minimizing $C(G)$ reduces the divergence between p_g and p_{data}.
  - At the global minimum, $p_g = p_{\text{data}}$, and $\mathrm{JS}(p_{\text{data}} \| p_g) = 0$.

GAN: Learning Process



GAN: Algorithm

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

  for k steps **do**

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
  - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  end for

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Outline

Similarity: VAE & GAN

- Measure the discrepancy between distributions
  - VAE
    There is a relation between MLE and KL divergence:

$$\begin{aligned} \mathrm{KL}(\mathbb{P}_x,\ \mathbb{P}_g) &= \int_x p_x \log \frac{p_x}{p_g} = \frac{1}{N} \sum_{x^{(i)} \in D} \log \frac{p_x(x^{(i)})}{p_g(x^{(i)})} \\ &= -\frac{1}{N} \sum_{x^{(i)}} \log p_g(x^{(i)}) + \frac{1}{N} \sum_{x^{(i)}} \log p_x(x^{(i)}) \end{aligned}$$

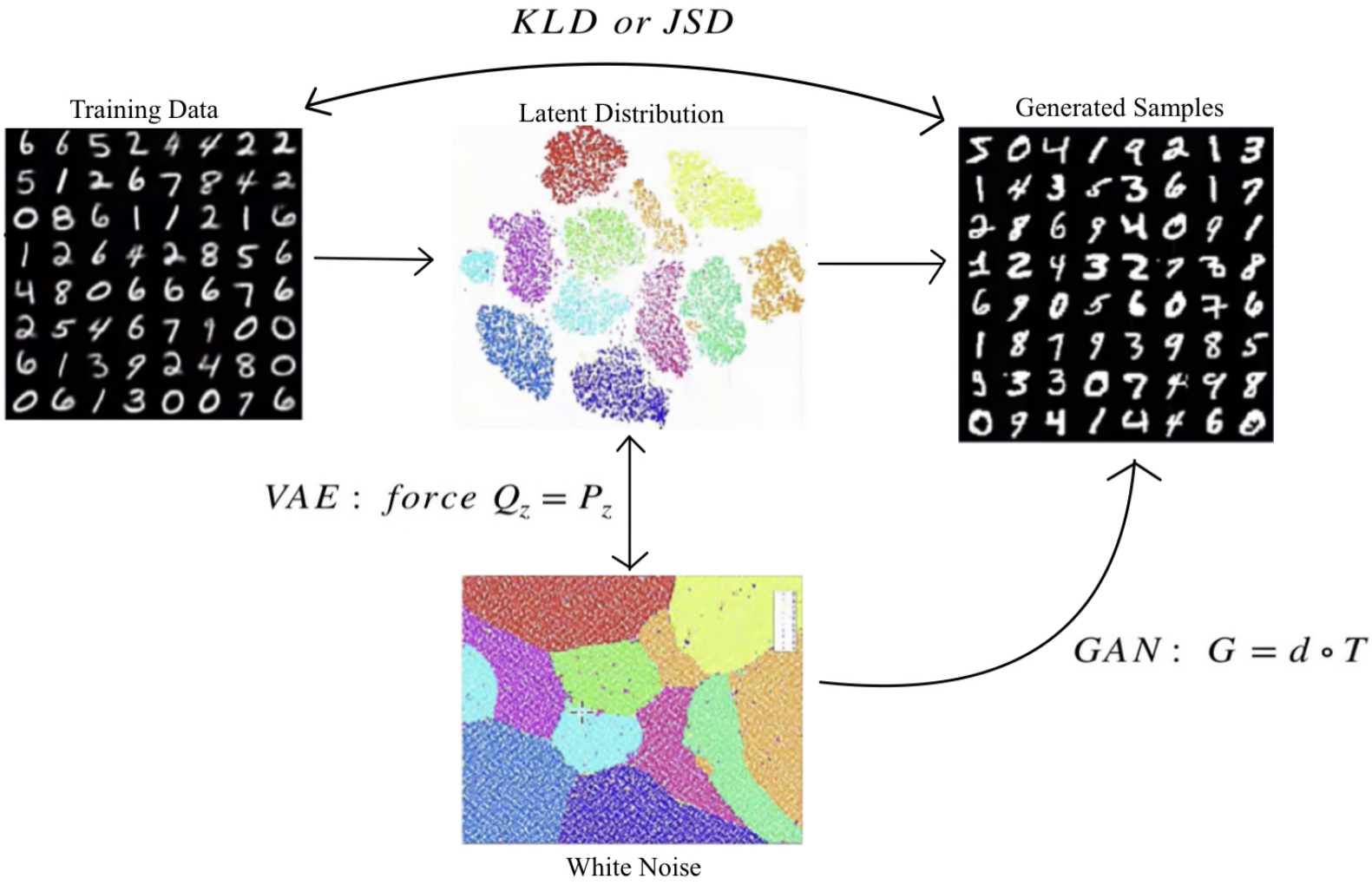
  - GAN

$$\log D^*(x) + \log(1 - D^*(G(z))) = \mathrm{JS}(\mathbb{P}_x,\ \mathbb{P}_g) + \log 2$$

Similarity: Both learn distributions by minimizing certain “distances”.

Similarity: VAE & GAN

- Encoder-Decoder structure

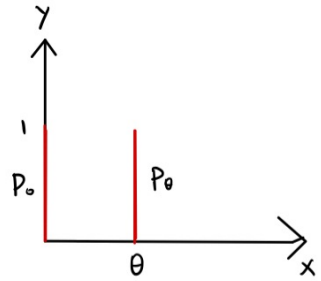


Drawback of GAN: Gradient Vanishing

- An easy example[3]

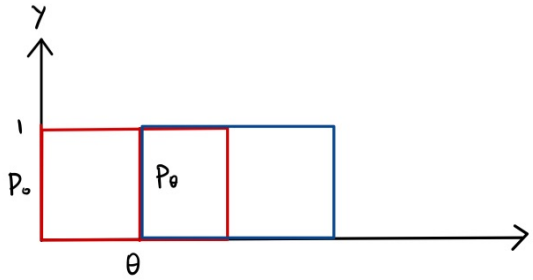
An easy example

Example 1 (Learning parallel lines). Let $Z \sim U[0,1]$ the uniform distribution on the unit interval. Let $\mathbb{P}_0$ be the distribution of $(0, Z) \in \mathbb{R}^2$ (a 0 on the x-axis and the random variable Z on the y-axis), uniform on a straight vertical line passing through the origin. Now let $g_\theta(z) = (\theta, z)$ with θ a single real parameter. It is easy to see that in this case,



- $W(\mathbb{P}_0, \mathbb{P}_\theta) = |\theta|$,
- $JS(\mathbb{P}_0, \mathbb{P}_\theta) = \begin{cases} \log 2 & \text{if } \theta \neq 0\ , \\ 0 & \text{if } \theta = 0\ , \end{cases}$
- $KL(\mathbb{P}_\theta \| \mathbb{P}_0) = KL(\mathbb{P}_0 \| \mathbb{P}_\theta) = \begin{cases} +\infty & \text{if } \theta \neq 0\ , \\ 0 & \text{if } \theta = 0\ , \end{cases}$

When D^*, $\nabla_\theta G = 0$ *(no intersect case)*

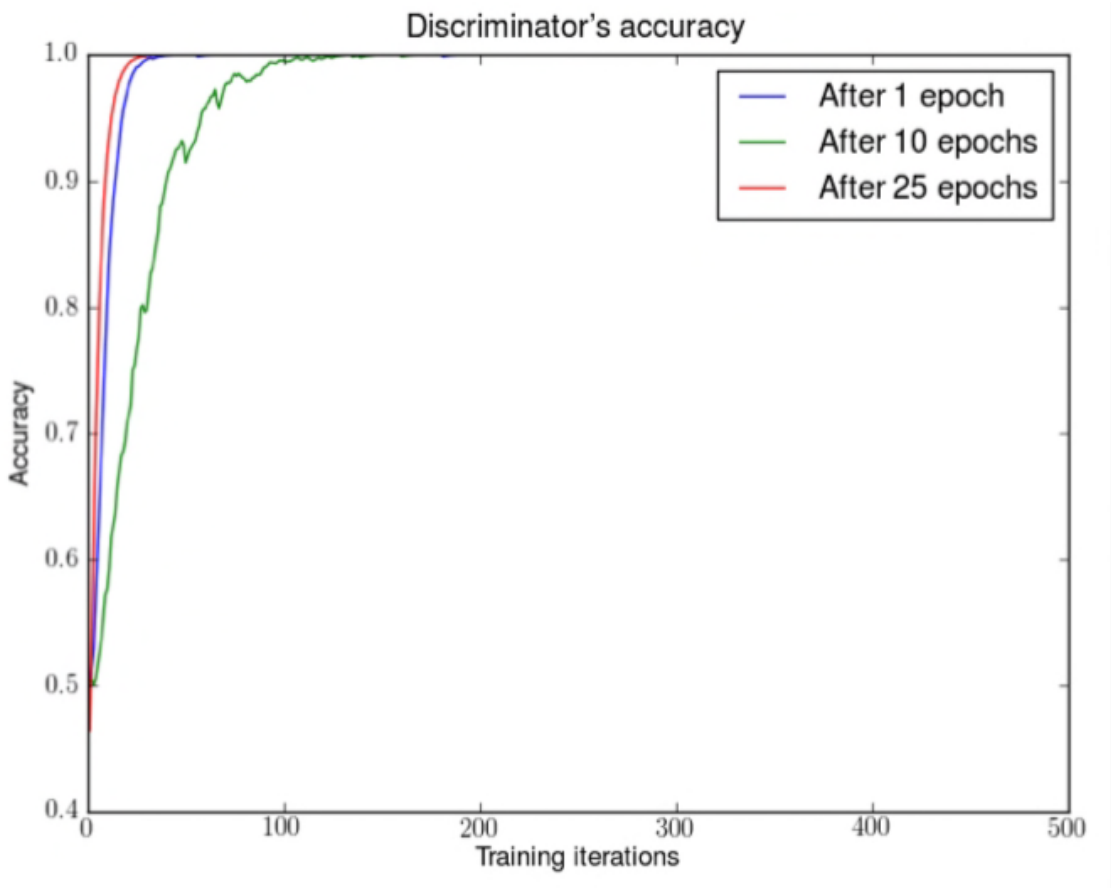


When D^*, $\nabla_\theta G \neq 0$ *(intersect case)*

[3] Arjovsky M., Bottou L. Towards Principled Methods for Training Generative Adversarial Networks. Stat, 2017, 1050.

Drawback of GAN: Gradient Vanishing

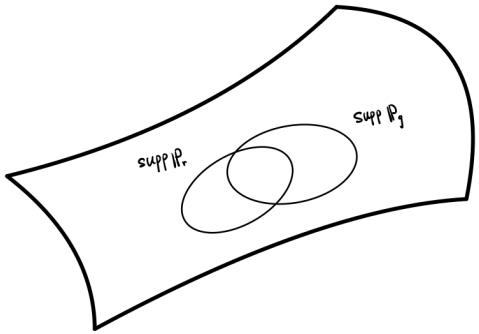
- Experiment: With different G fixed, train D and test D's accuracy:



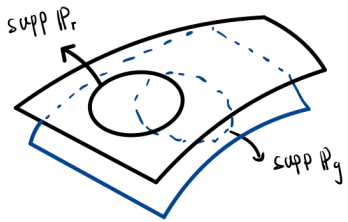
- Observation: Perfect discriminator D_G^* can always be trained.

Drawback of GAN: Gradient Vanishing

- Perfect Discriminator Theorem
  In fact, D_G^* can always be trained theoritically.
  - Key Lemmas:
    1. $\mathbb{P}_g$ and $\mathbb{P}_{data}$ are supported on low dimensional manifolds.
    2. In almost all cases, $\text{supp}\mathbb{P}_g \cap \text{supp}\mathbb{P}_{data} = \varnothing$ *or N*, *N null set*.
  - Examples:



$$\mathbb{P}(happen) = 0$$



$$\mathbb{P}(happen) = 1$$

Drawback of GAN: Gradient Vanishing

- Perfect Discriminator Theorem

Theorem 1

If two distributions $\mathbb{P}_{data}$ and $\mathbb{P}_g$ have support contained on two disjoint compact subsets M and P respectively, then there is a smooth optimal discrimator D^* that has accuracy 1 and $\nabla_x D^*(x) = 0$ for all $x \in M \cup P$.

Theorem 2

Let $\mathbb{P}_{data}$ and $\mathbb{P}_g$ be two distributions that have support contained in two closed manifolds M and P that don't perfectly align and don't have full dimension. We further assume that $\mathbb{P}_{data}$ and $\mathbb{P}_g$ are continuous in their respective manifolds, meaning that if there is a set A with measure 0 in M, then $\mathbb{P}_{data}(A) = 0$ (and analogously for $\mathbb{P}_g$). Then, there exists an optimal discriminator D^* that has accuracy 1 and for almost any x in M or P, D^* is smooth in a neighbourhood of x and $\nabla_x D^*(x) = 0$.

Drawback of GAN: Gradient Vanishing

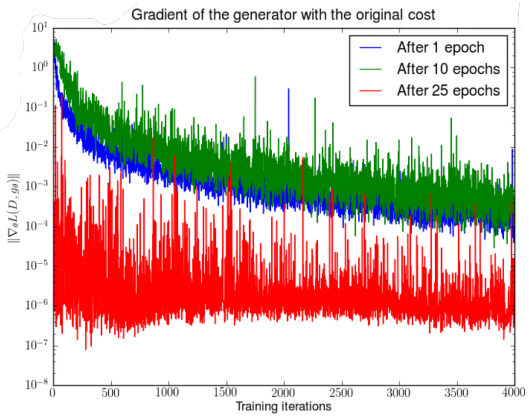
- Gradient Vanishing Theorem

Theorem 3

Let $g_\theta : Z \to X$ be a differentiable function that induces a distribution $\mathbb{P}_g$. Let $\mathbb{P}_{data}$ be the real data distribution. Let D be a differentiable discriminator. If the conditions of Theorems 3 or 4 are satisfied, $|D - D^*| < \epsilon$, and $\mathbb{E}_{z\sim p(z)}|J_\theta g_\theta(z)|_2^2 \leqslant M^2$, then

$$\nabla_\theta |\mathbb{E}_{z\sim p(z)}[log(1 - D(g_\theta(z)))]|_2 \leqslant M\frac{\epsilon}{1-\epsilon}$$

- Experimental support



Drawback of GAN: Instability

- log(1-s) case:

$$V = E_{x \sim P_{data}}[logD(x)]$$

$$+E_{x \sim P_G}[log(1 - D(x))]$$

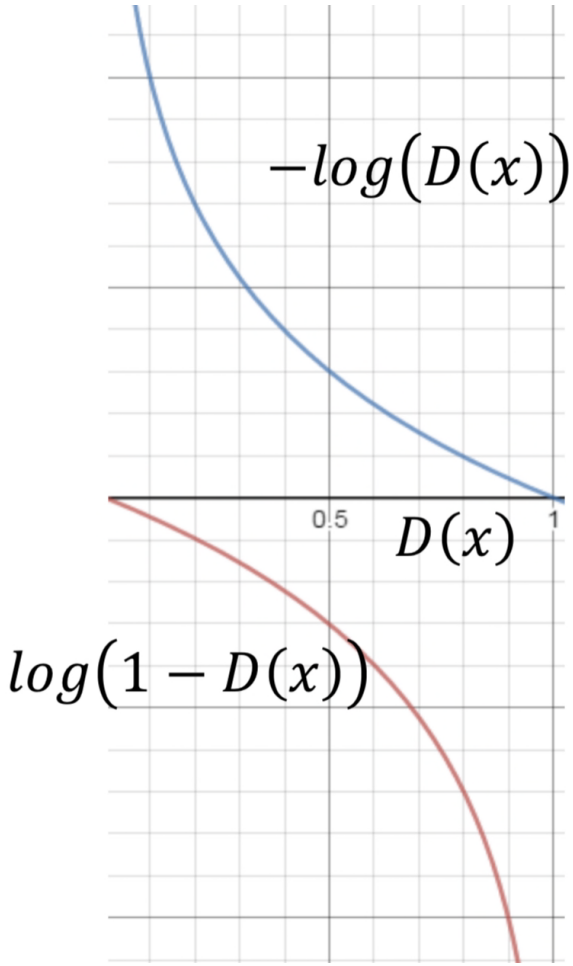
Slow at the beginning

Minimax GAN (MMGAN)

$$V = E_{x \sim P_G}[-log(D(x))]$$

Real implementation:
label x from P_G as positive

Non-saturating GAN (NSGAN)



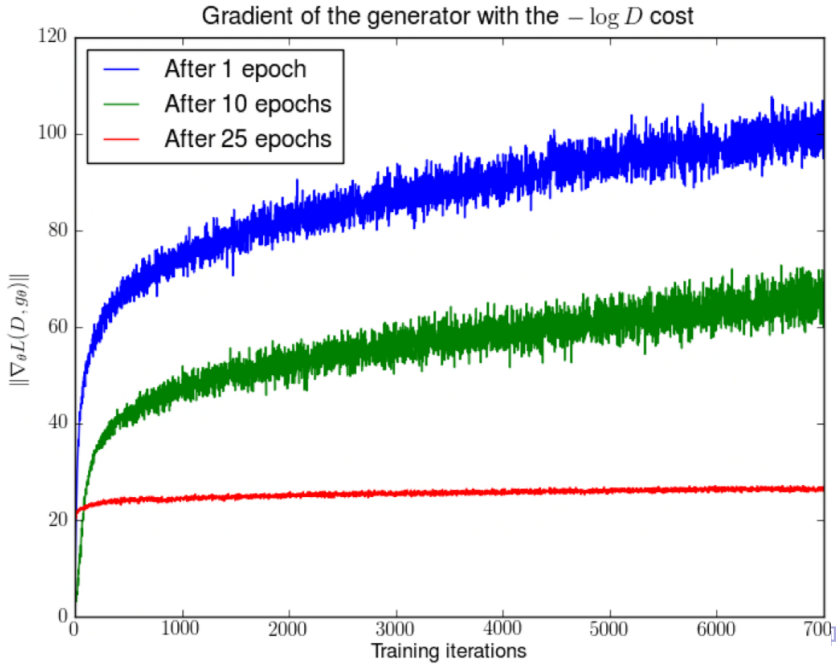
Drawback of GAN: Instability

- -log(s) case:

Theorem 4

Let $\mathbb{P}_{data}$ and $\mathbb{P}_{g_\theta}$ be two continuous distributions, with densities $\mathbb{P}_{data}$ and $\mathbb{P}_{g_\theta}$ respectively. Let D^* be the optimal discriminator, fixed for a value θ_0. Therefore,

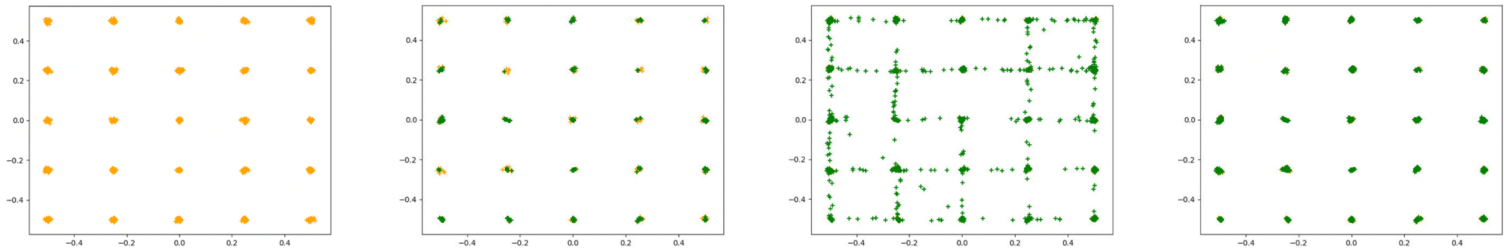
$$E_{z\sim p(z)}[-\nabla_\theta log D^*(g_\theta(z))|_{\theta=\theta_0}] = \nabla_\theta[KL(P_{g_\theta}|P_{data}) - 2JSD(P_{g_\theta}|P_{data})]|_{\theta=\theta_0}$$



Drawback of GAN: Mode Collapse & Mode Mixture

- Mode Collapse

- Mode Mixture



Drawback of VAE: Weak Regularizer

We sample from $Q(Z)$ when training and $P(Z)$ in testing in latent space, hence $Q(Z)$ must match $P(Z)$.

- Experimental Test:[4]

	ELBO	Avg. KL	Mutual info. ②	Marg. KL ③
2D latents	-129.63	7.41	7.20	0.21
10D latents	-88.95	19.17	10.82	8.35
20D latents	-87.45	20.2	10.67	9.53

$Q(Z)$ generally does not match $P(Z)$ in VAE.

- Reason: Weak Regularizer

$$\text{ELBO}^{(i)} = \mathcal{L}(\theta, \phi, x^{(i)}) = -\text{KL}(q(z|x^{(i)}, p_\theta(z)) + \mathbb{E}_{q_\phi(z|x^{(i)})}(p(x^{(i)})|z)$$

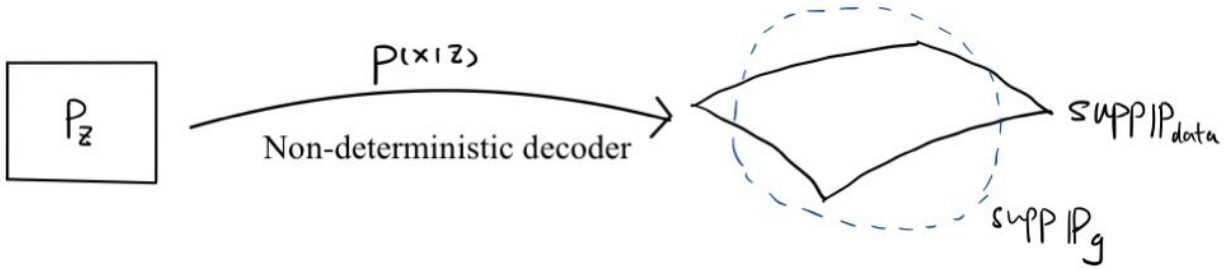
where $\text{KL}(q|p)$ as a regularizer is too weak for $Q(Z) = P(Z)$.

[4] Locatello F., et al. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations. 2018.

Drawback of VAE: Blurry

Generated data are not concertrated on a low dimensional manifold!
This leads to blurry.

- Explanation



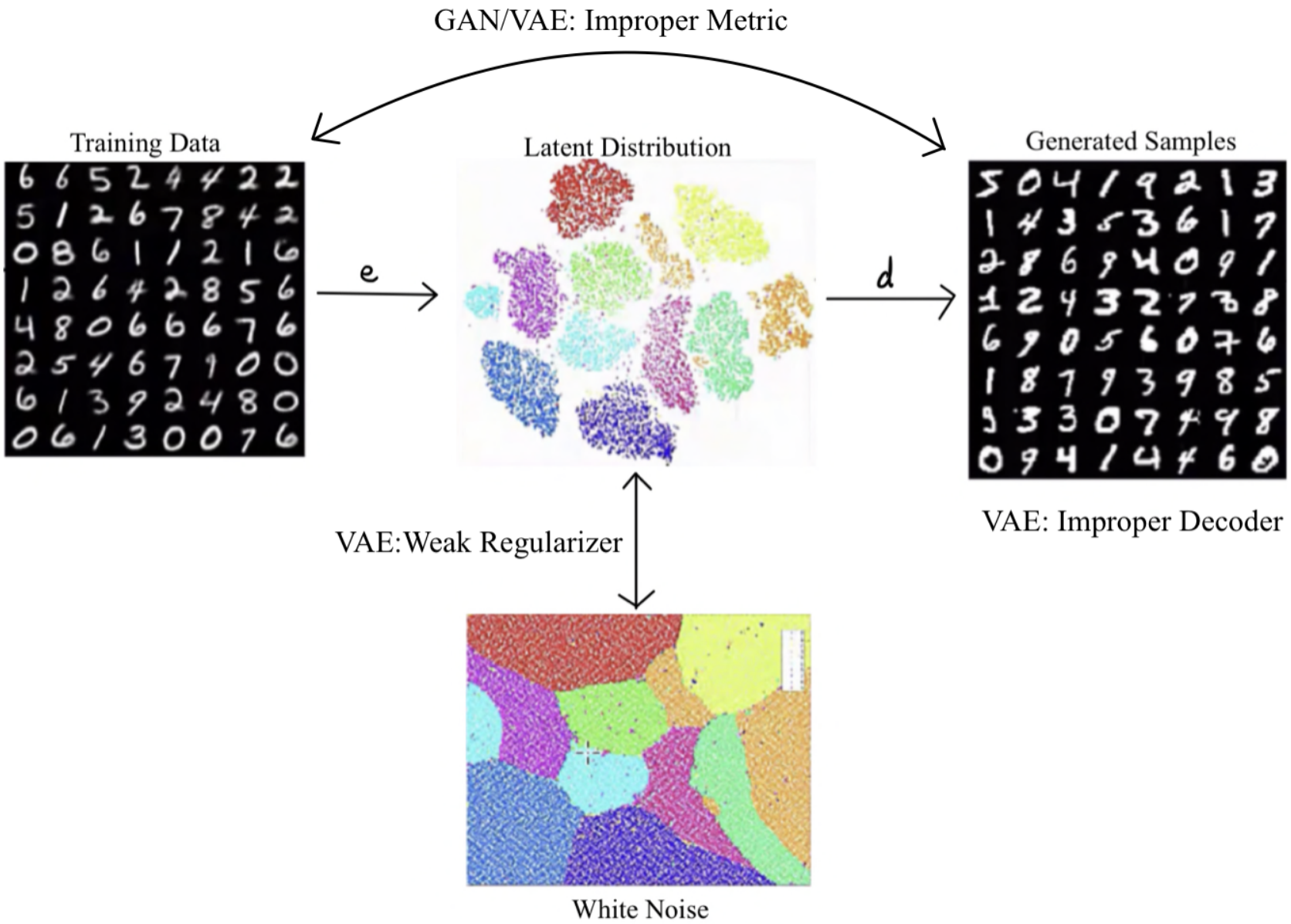
- Experiment[5]

Table 2: Quantitative comparison with FID

	Adversarial				Non-Adversarial		Reference	
Dataset	NS GAN	LSGAN	WGAN	BEGAN	VAE	GLANN	AE	Ours
MNIST	6.8±0.5	7.8±0.6	6.7±0.4	13.1±1.0	23.8±0.6	8.6±0.1	5.5	**6.2±0.2**
Fansion	26.5±1.6	30.7±2.2	21.5±1.6	22.9±0.9	58.7±1.2	13.0±0.1	4.7	**10.1±0.3**
CIFAR-10	58.5±1.9	87.1±47.5	55.2±2.3	71.4±1.6	65.4±0.2	46.5±0.2	28.2	**38.3±0.5**
CelebA	55.0±3.3	53.9±2.8	41.3±2.0	**38.9±0.9**	85.7±3.8	46.3±0.1	67.5	68.4± 0.5

[5] An D., et al. AE-OT-GAN: Training GANs from data specific latent distribution. 2020.

Summary of Drawbacks



Outline

[6] Adler, Jonas, and Sebastian Lunz. "Banach wasserstein gan." NIPS, 2018.

[7] Tolstikhin, Ilya, et al. "Wasserstein auto-encoders." arXiv:1711.01558, 2017.

Wasserstein Distance: An Introduction

- Kantorovich Relaxation

$$W_c(X, Y) = \inf_{\Gamma \in P(P_X, P_G)} \mathbb{E}_{\Gamma(X,Y)}[c(X, Y)]$$

Γ: coupling, a joint distribution satisfying marginal restriction
Notice: optimal coupling always exists

Wasserstein Distance: An Introduction

- Kantorovich Relaxation

$$W_c(X, Y) = \inf_{\Gamma \in P(P_X, P_G)} \mathbb{E}_{\Gamma(X,Y)}[c(X, Y)]$$

  Γ: coupling, a joint distribution satisfying marginal restriction
  Notice: optimal coupling always exists

- Kantorovich duality

$$W_c(X, Y) = \sup_{\psi(y)+\phi(x)\leqslant c(x,y)} [\mathbb{E}_{P_X}\psi(y) + \mathbb{E}_{P_G}\phi(x)]$$

- Define c-transform $\phi^c(y) = \inf_x[c(x, y) - \phi(x)]$, then

$$W_c(X, Y) = \sup_{\phi}[\mathbb{E}_{P_X}\phi^c(y) + \mathbb{E}_{P_G}\phi(x)]$$

- If c is 1-norm and ϕ is 1-Lipchitz, $\phi^c = -\phi$ and then

$$W_c(X, Y) = \sup_{|\phi|_L\leqslant 1} [\mathbb{E}_{P_G}\phi(x) - \mathbb{E}_{P_X}\phi(y)]$$

Wasserstein Distance: A Better Metric

- Wasserstein Metric as a Weaker Metric

  Example 1 (Learning parallel lines). Let $Z \sim U[0,1]$ the uniform distribution on the unit interval. Let $\mathbb{P}_0$ be the distribution of $(0, Z) \in \mathbb{R}^2$ (a 0 on the x-axis and the random variable Z on the y-axis), uniform on a straight vertical line passing through the origin. Now let $g_\theta(z) = (\theta, z)$ with θ a single real parameter. It is easy to see that in this case,

  - $W(\mathbb{P}_0, \mathbb{P}_\theta) = |\theta|$,
  - $JS(\mathbb{P}_0, \mathbb{P}_\theta) = \begin{cases} \log 2 & \text{if } \theta \neq 0\ , \\ 0 & \text{if } \theta = 0\ , \end{cases}$
  - $KL(\mathbb{P}_\theta \| \mathbb{P}_0) = KL(\mathbb{P}_0 \| \mathbb{P}_\theta) = \begin{cases} +\infty & \text{if } \theta \neq 0\ , \\ 0 & \text{if } \theta = 0\ , \end{cases}$

- Observation
  - Only W converges at 0.
  - Moreover, only W is continuous (even differentiable) at 0.

Wasserstein Distance: A Better Metric

- Wasserstein Metric as a Weaker Metric

Theorem 5

Let $\mathbb{P}$ be a distribution on a compact space $\mathcal{X}$ and $(\mathbb{P}_n)_{n\in\mathbb{N}}$ be a sequence of distributions on $\mathcal{X}$. Then, considering all limits as $n \to \infty$.

1. *The following statements are equivalent*
   - $\delta(\mathbb{P}_n, \mathbb{P}) \to 0$ *with* δ *the total variation distance.*
   - $JS(\mathbb{P}_n, \mathbb{P}) \to 0$ *with JS the Jensen-Shannon divergence.*
2. *The following statements are equivalent*
   - $W(\mathbb{P}_n, \mathbb{P}) \to 0$
   - $\mathbb{P}_n \xrightarrow{\mathcal{D}} \mathbb{P}$ *where* $\xrightarrow{\mathcal{D}}$ *represents convergence in distribution for random variables.*
3. $KL(\mathbb{P}_n|\mathbb{P}) \to 0$ *or* $KL(\mathbb{P}|\mathbb{P}_n) \to 0$ *imply the statements in 1.*
4. *The statements in 1 imply the statements in 2.*

- Conclusion: Difficulty to convergence: $KL > JS > W$.

Wasserstein Distance: A Better Metric

- Wasserstein Metric is indeed continuous and differentiable[8]

Theorem 6

Let $\mathbb{P}_r$ be a fixed distribution over $\mathcal{X}$. Let $\mathcal{Z}$ be a random variable (e.g. Gaussian) over another space Z. Let $g : \mathcal{Z} \times \mathbb{R}^d \to \mathcal{X}$ be a function, which will be denoted as $g_\theta(z)$ with z the first coordinate and θ the second. Let $\mathbb{P}_\theta$ denote the distribution of $g_\theta(z)$. Then,
1. If g is continuous in θ, so is $W(\mathbb{P}_r, \mathbb{P}_\theta)$.
2. If g is locally Lipschitz and satisfies a regularity assumption, then $W(\mathbb{P}_r, \mathbb{P}_\theta)$ is continuous everywhere, and differentiable almost everywhere.
3. Statements 1-2 are false for the Jensen-Shannon divergence $JS(\mathbb{P}_r, \mathbb{P}_\theta)$ and all the KLs.

[8] Arjovsky M , Chintala S , Bottou L . Wasserstein GAN[J]. 2017.

Wasserstein Distance: A Better Metric

- Wasserstein Metric is indeed continuous and differentiable

Theorem 7

Let g_θ be any feedforward neural network parameterized by θ, and p(z) a prior over z such that $\mathbb{E}_{z\sim p(z)}[|z|] \leqslant \infty$ (e.g. Gaussian, uniform, etc.). Then the regularity assumption mentioned above is satisfied and therefore $W(\mathbb{P}_r, \mathbb{P}_\theta)$ is continuous everywhere and differentiable almost everywhere.

Wasserstein metric is more suitable than JS and KL when measuring the distance between the generated and real data distributions.

Wasserstein GAN: Theory

- Different with the original GAN metric: $JS(\mathbb{P}_g, \mathbb{P}_r) \longrightarrow W(\mathbb{P}_g, \mathbb{P}_r)$
- Optimization objective

$$W(\mathbb{P}_X, \mathbb{P}_G) = \inf_{\gamma \in P(P_X, P_G)} \mathbb{E}_{\Gamma(X,Y)}[|x - y|]$$

Wasserstein GAN: Theory

- Different with the original GAN metric: $JS(\mathbb{P}_g, \mathbb{P}_r) \longrightarrow W(\mathbb{P}_g, \mathbb{P}_r)$
- Optimization objective

$$W(\mathbb{P}_X, \mathbb{P}_G) = \inf_{\gamma \in P(P_X, P_G)} \mathbb{E}_{\Gamma(X,Y)}[|x - y|]$$

Since the formula is intractable, we choose the dual fomulation:

$$W(\mathbb{P}_X, \mathbb{P}_G) = \sup_{|\phi|_L \leqslant 1} [\mathbb{E}_{P_G}\phi(x) - \mathbb{E}_{P_X}\phi(y)]$$

Wasserstein GAN: Theory

- Different with the original GAN metric: $JS(\mathbb{P}_g, \mathbb{P}_r) \longrightarrow W(\mathbb{P}_g, \mathbb{P}_r)$
- Optimization objective

$$W(\mathbb{P}_X, \mathbb{P}_G) = \inf_{\gamma \in P(P_X, P_G)} \mathbb{E}_{\Gamma(X,Y)}[|x - y|]$$

Since the formula is intractable, we choose the dual fomulation:

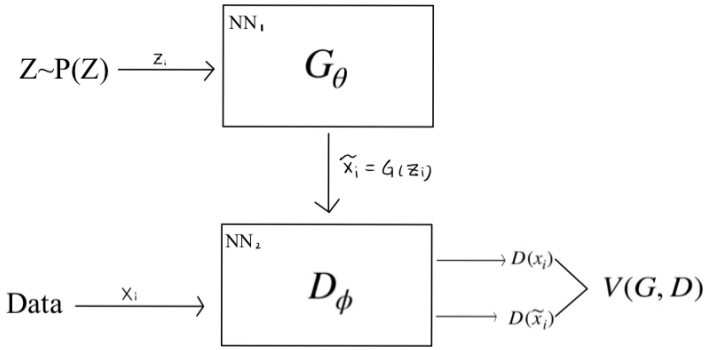
$$W(\mathbb{P}_X, \mathbb{P}_G) = \sup_{|\phi|_L \leqslant 1} [\mathbb{E}_{P_G}\phi(x) - \mathbb{E}_{P_X}\phi(y)]$$

Lipchitz can be obtained by setting weight space $\mathcal{W}$ as a compact space:

$$K \cdot W(\mathbb{P}_X, \mathbb{P}_G) = \max_{w \in \mathcal{W}} [\mathbb{E}_{P_G}\phi_w(x) - \mathbb{E}_{P_X}\phi_w(y)]$$

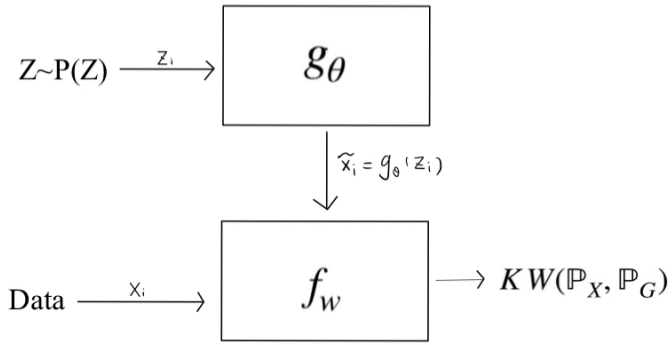
Wasserstein GAN: Neural Network Structure

GAN:



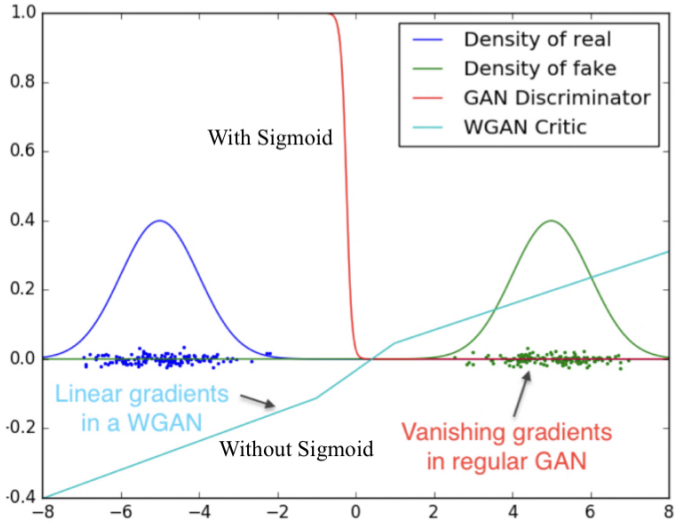
$$\min_\theta \max_w [\mathbb{E}_{P_{data}} \log D_w(x) - \mathbb{E}_{P_z} \log D_w(G_\theta(z)))]$$

WGAN:



$$\min_\theta \max_w [\mathbb{E}_{P_{data}} f_w(x) - \mathbb{E}_{P_z} f_w(G_\theta(z))]$$

Difference:
1. No sigmoid in output layer.
2. No log term in loss function.
3. Weight Clipping



Wasserstein GAN: Algorithm

Algorithm 1 WGAN, our proposed algorithm. All experiments in the paper used the default values $\alpha = 0.00005$, $c = 0.01$, $m = 64$, $n_{\text{critic}} = 5$.

Require: : α, the learning rate. c, the clipping parameter. m, the batch size. n_{critic}, the number of iterations of the critic per generator iteration.

Require: : w_0, initial critic parameters. θ_0, initial generator's parameters.

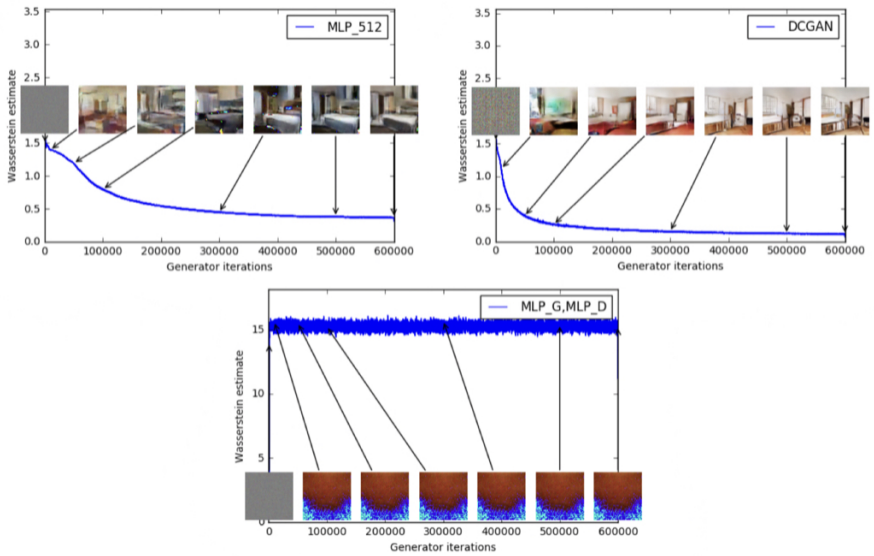
while θ has not converged **do**
  for $t = 0, ..., n_{\text{critic}}$ **do**
    Sample $\{x^{(i)}\}_{i=1}^{m} \sim \mathbb{P}_r$ a batch from the real data.
    Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of prior samples.
    $g_w \leftarrow \nabla_w \left[\frac{1}{m}\sum_{i=1}^{m} f_w(x^{(i)}) - \frac{1}{m}\sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))\right]$
    $w \leftarrow w + \alpha \cdot \text{RMSProp}(w, g_w)$
    $w \leftarrow \text{clip}(w, -c, c)$
  end for
  Sample $\{z^{(i)}\}_{i=1}^{m} \sim p(z)$ a batch of prior samples.
  $g_\theta \leftarrow -\nabla_\theta \frac{1}{m}\sum_{i=1}^{m} f_w(g_\theta(z^{(i)}))$
  $\theta \leftarrow \theta - \alpha \cdot \text{RMSProp}(\theta, g_\theta)$
end while

Wasserstein GAN: Experimental Result

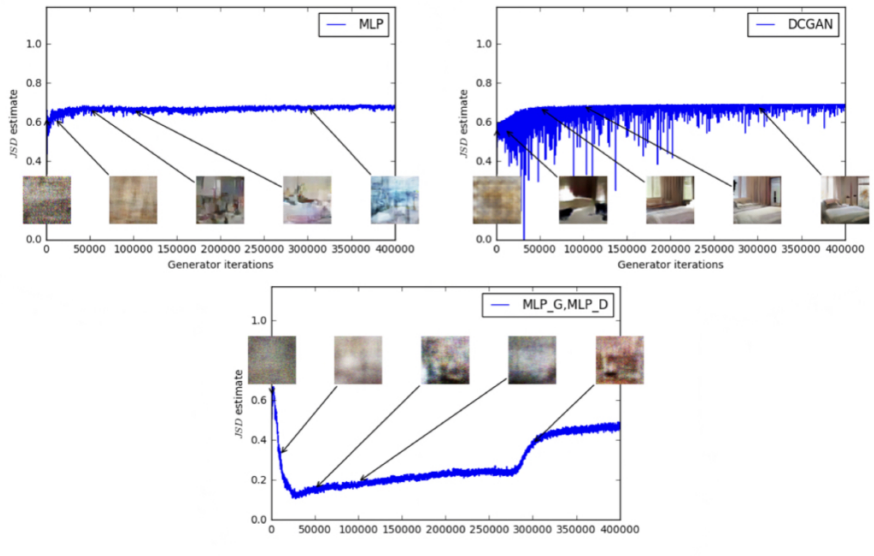
- Meaningful Loss Metric
  Notice: we can (and should) train the critic till optimality, which actually equals to $K \cdot W$

We have a explicit loss metric!

W:



JS:

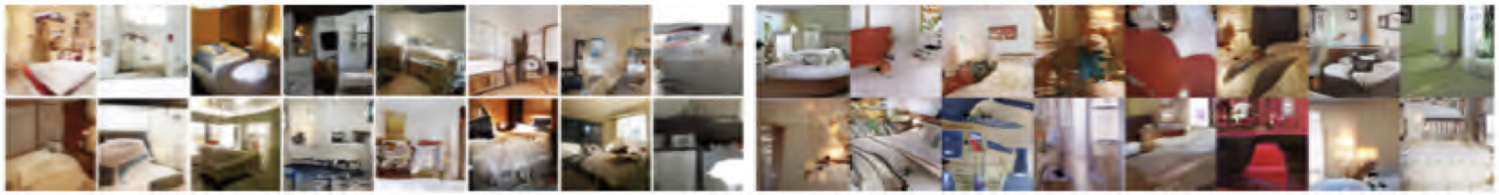


Wasserstein GAN: Experimental Result

- Robustness

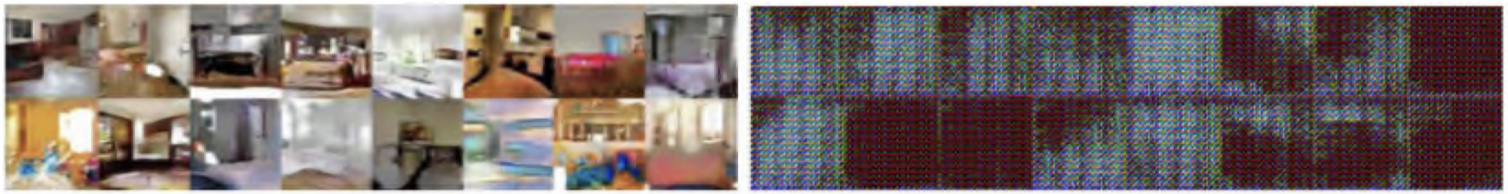
DCGAN Generator:

W: JS:



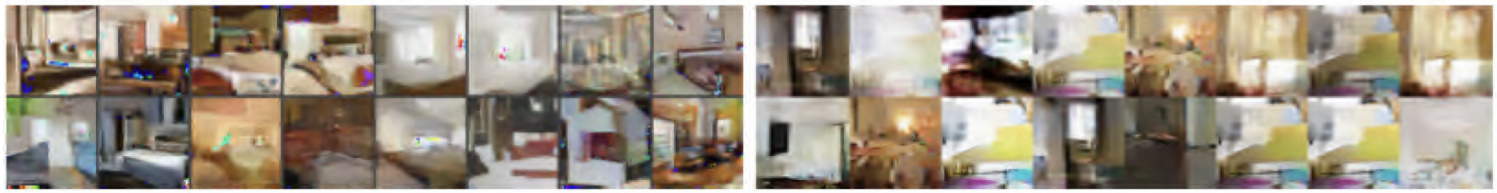
DCGAN Generator without Batch Normalization:

W: JS:



MLP Generator:

W: JS: (Mode Collapse)



Wasserstein Distance: Latent Variable Version

- What we need
  1. Deterministic decoder & Proper metric
  2. Stronger regularizer
- Recall Kantorovich's formula:
  $W_c(X, Y) = \sup_{\psi(y)+\phi(x)\leqslant c(x,y)} [\mathbb{E}_{P_X}\psi(y) + \mathbb{E}_{P_G}\phi(x)]$
- Formula with latent variable

Theorem 8

For $\mathbb{P}_G$ as defined above with deterministic $P_G(X|Z)$ and any function $G : \mathcal{Z} \to \mathcal{X}$

$$\inf_{\gamma \in P(P_X, P_G)} \mathbb{E}_{\Gamma(X,Y)}[c(x, y)] = \inf_{Q: Q_z = P_Z} \mathbb{E}_{P_X}\mathbb{E}_{Q(Z|X)}[c(X, G(Z))]$$

Why can we expect such a formula?

Wasserstein AE: Relaxation & AE Structure

- Relaxation fomulation
  From restriction term

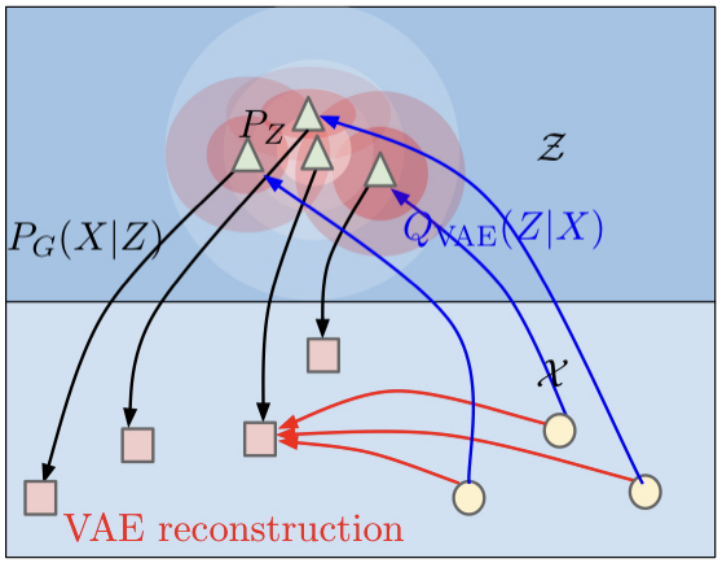
$$\inf_{Q:Q_Z=P_Z} \mathbb{E}_{P_X}\mathbb{E}_{Q(Z|X)}[c(X, G(Z))]$$

  to penalty term

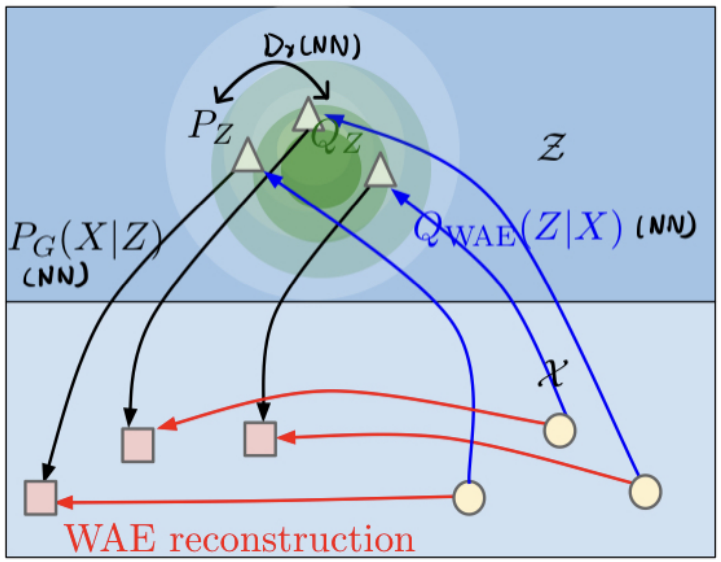
$$D_{WAE}(P_X, P_G) = \inf_{Q(Z|X)\in Q} \mathbb{E}_{P_X}\mathbb{E}_{Q(Z|X)}[c(X, G(Z))] + \lambda D_Z(Q_Z, P_Z)$$

- AE Structure

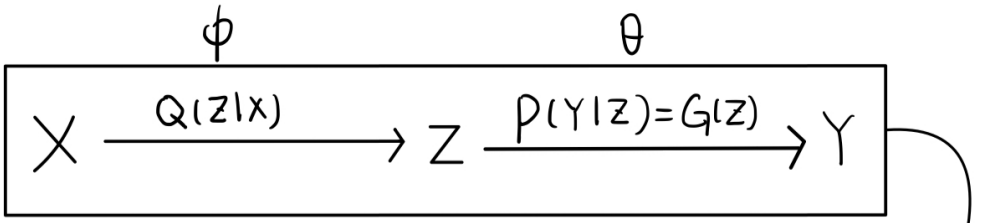
(a) VAE

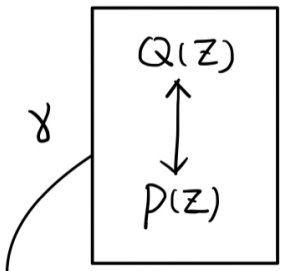


(b) WAE



Wasserstein AE: Penalties & Optimization





$$D_{\mathrm{WAE}}(P_X, P_G) := \inf_{Q(Z|X)\in\mathcal{Q}} \mathbb{E}_{P_X}\mathbb{E}_{Q(Z|X)}\big[c\big(X, G(Z)\big)\big] + \lambda \cdot \mathcal{D}_Z(Q_Z, P_Z)$$

$$GAN-based:\ D_Z(P_Z, Q_Z) = \mathbb{E}_{z\sim P_Z} log D(z) + \mathbb{E}_{\tilde{z}\sim Q_Z} log(1 - D(\tilde{z}))$$

$$MMD-based:\ D_Z(P_Z, Q_Z) = \|\mathbb{E}_{P_Z} k(z, \cdot) - \mathbb{E}_{Q_Z} k(z, \cdot)\|_{\mathcal{H}_k}$$

Wasserstein AE: Algorithm

Algorithm 1 Wasserstein Auto-Encoder with GAN-based penalty (WAE-GAN).

Require: Regularization coefficient $\lambda > 0$.

Initialize the parameters of the encoder Q_ϕ, decoder G_θ, and latent discriminator D_γ.

while (ϕ, θ) not converged **do**

Sample $\{x_1, \dots, x_n\}$ from the training set

Sample $\{z_1, \dots, z_n\}$ from the prior P_Z

Sample $\tilde{z}_i$ from $Q_\phi(Z|x_i)$ for $i = 1, \dots, n$

Update D_γ by ascending:

$$\frac{\lambda}{n} \sum_{i=1}^{n} \log D_\gamma(z_i) + \log\big(1 - D_\gamma(\tilde{z}_i)\big)$$

Update Q_ϕ and G_θ by descending:

$$\frac{1}{n} \sum_{i=1}^{n} c\big(x_i, G_\theta(\tilde{z}_i)\big) - \lambda \cdot \log D_\gamma(\tilde{z}_i)$$

end while

Algorithm 2 Wasserstein Auto-Encoder with MMD-based penalty (WAE-MMD).

Require: Regularization coefficient $\lambda > 0$, characteristic positive-definite kernel k.

Initialize the parameters of the encoder Q_ϕ, decoder G_θ, and latent discriminator D_γ.

while (ϕ, θ) not converged **do**

Sample $\{x_1, \dots, x_n\}$ from the training set

Sample $\{z_1, \dots, z_n\}$ from the prior P_Z

Sample $\tilde{z}_i$ from $Q_\phi(Z|x_i)$ for $i = 1, \dots, n$

Update Q_ϕ and G_θ by descending:

$$\frac{1}{n} \sum_{i=1}^{n} c\big(x_i, G_\theta(\tilde{z}_i)\big) + \frac{\lambda}{n(n-1)} \sum_{\ell \neq j} k(z_\ell, z_j)$$
$$+ \frac{\lambda}{n(n-1)} \sum_{\ell \neq j} k(\tilde{z}_\ell, \tilde{z}_j) - \frac{2\lambda}{n^2} \sum_{\ell, j} k(z_\ell, \tilde{z}_j)$$

end while

Wasserstein AE: Experimental Result

Encoder architecture:

$$\begin{aligned} x \in \mathcal{R}^{64\times64\times3} &\to \text{Conv}_{128} \to \text{BN} \to \text{ReLU} \\ &\to \text{Conv}_{256} \to \text{BN} \to \text{ReLU} \\ &\to \text{Conv}_{512} \to \text{BN} \to \text{ReLU} \\ &\to \text{Conv}_{1024} \to \text{BN} \to \text{ReLU} \to \text{FC}_{64} \end{aligned}$$

Decoder architecture:

$$\begin{aligned} z \in \mathcal{R}^{64} &\to \text{FC}_{8\times8\times1024} \\ &\to \text{FSConv}_{512} \to \text{BN} \to \text{ReLU} \\ &\to \text{FSConv}_{256} \to \text{BN} \to \text{ReLU} \\ &\to \text{FSConv}_{128} \to \text{BN} \to \text{ReLU} \to \text{FSConv}_{1} \end{aligned}$$

Adversary architecture for WAE-GAN:

$$\begin{aligned} z \in \mathcal{R}^{64} &\to \text{FC}_{512} \to \text{ReLU} \\ &\to \text{FC}_{512} \to \text{ReLU} \\ &\to \text{FC}_{512} \to \text{ReLU} \\ &\to \text{FC}_{512} \to \text{ReLU} \to \text{FC}_{1} \end{aligned}$$

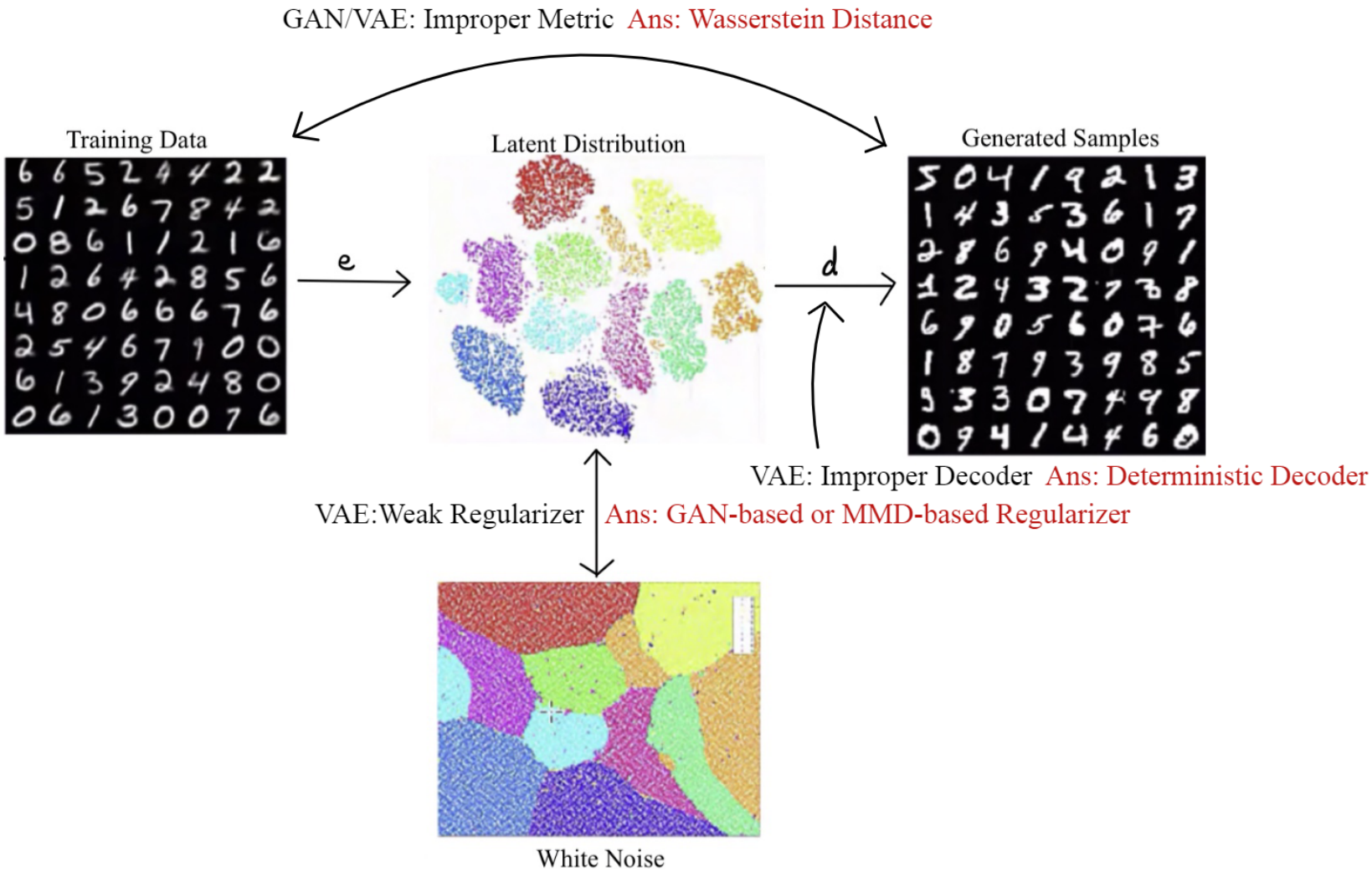
Wasserstein AE: Experimental Result

More close to data manifold

Algorithm	FID	Sharpness
VAE	63	3×10^{-3}
WAE-MMD	55	6×10^{-3}
WAE-GAN	42	6×10^{-3}
bigVAE	45	—
bigWAE-MMD	37	—
bigWAE-GAN	35	—
True data	2	2×10^{-2}

Table 1: FID (smaller is better) and sharpness (larger is better) scores for samples of various models for CelebA.

Review of the Solutions



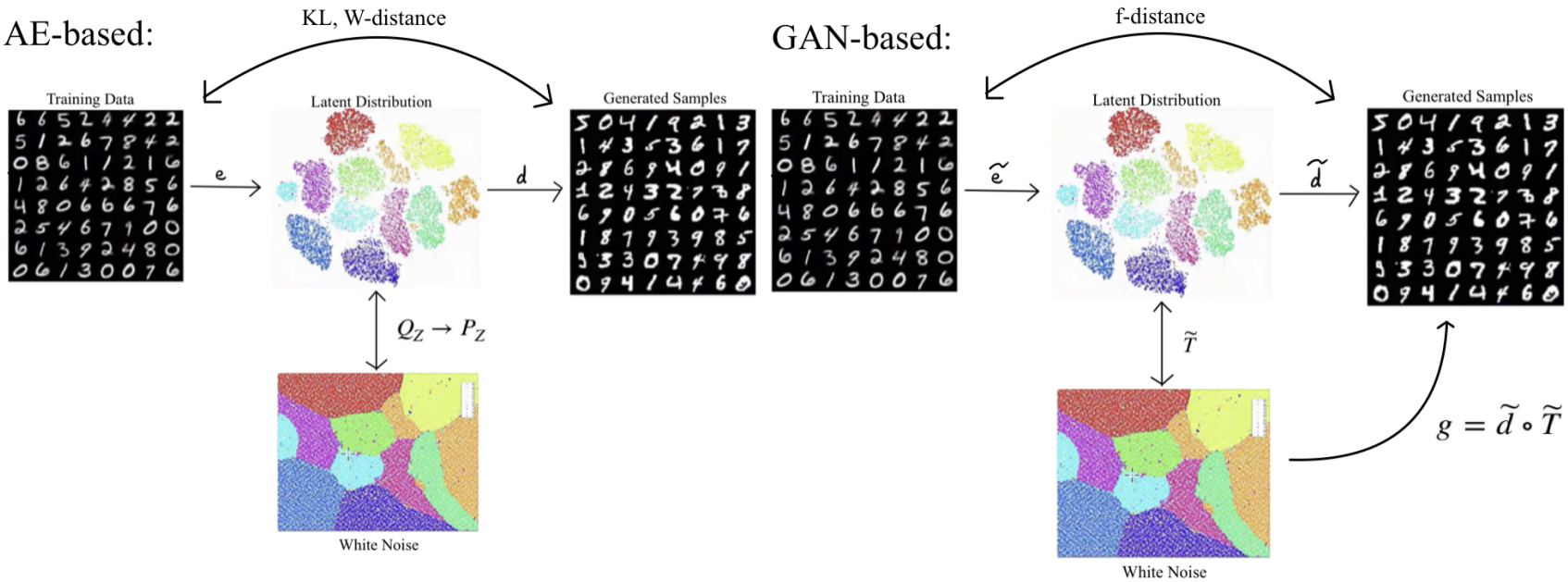
Outline

Unified Structure

Generative models mainly accomplish two tasks:

1. Manifold Learning: Dimension reduction and retaining the information of data distribution
2. Distribution Transportation (Measure Learning): Learn data distribution in low dimension

Key: Different models accomplish tasks in different ways:



Observation: both of them mix the two processes.

Problem Remained

1. What are the reasons for model collapse & mixture in GAN based models?
2. What are the reasons for relatively low quality of generated data in AE based models?

	CIFAR10		CelebA	
	Standard	ResNet	Standard	ResNet
WGAN-GP [14]	40.2	19.6	21.2	18.4
PGGAN [38]	-	18.8	-	16.3
SNGAN [32]	25.5	21.7	-	-
WGAN-div [20]	-	18.1	17.5	15.2
WGAN-QC [29]	-	-	-	12.9
AE-OT [2]	34.2	28.5	24.3	28.6
AE-OT-GAN	25.2	17.1	11.2	7.8

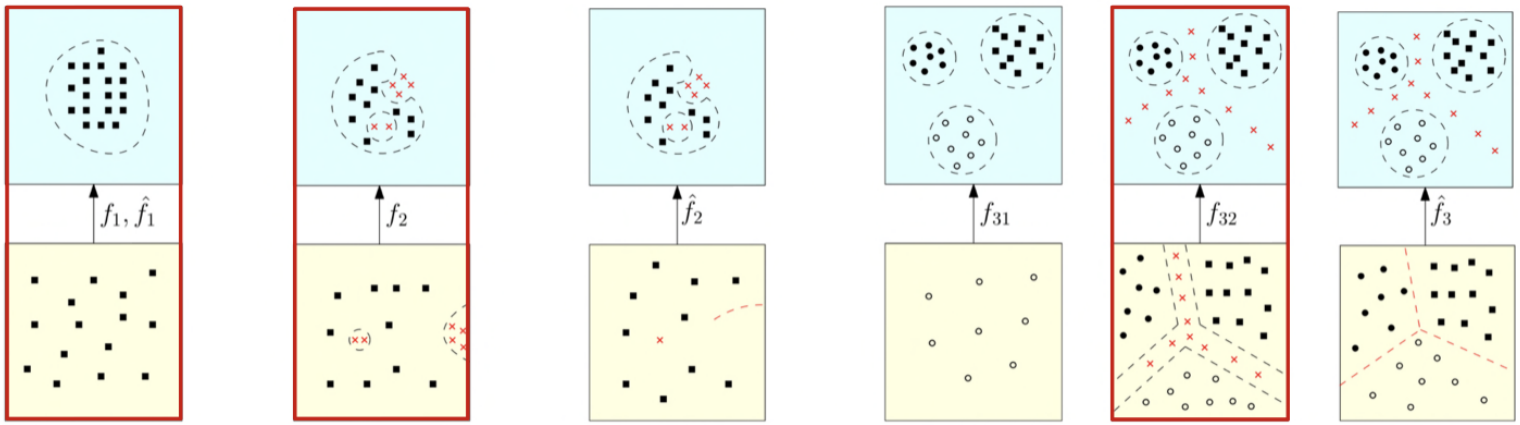
The comparison of FID between the proposed method and the state of the arts on Cifar10 and CelebA.

Algorithm	**FID**	**Sharpness**
VAE	63	3×10^{-3}
WAE-MMD	55	6×10^{-3}
WAE-GAN	42	6×10^{-3}
bigVAE	45	—
bigWAE-MMD	37	—
bigWAE-GAN	35	—
True data	2	2×10^{-2}

Table 1: FID (smaller is better) and sharpness (larger is better) scores for samples of various models for CelebA.

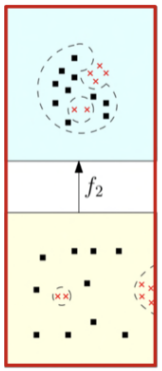
Reasons for Failure in GAN Based Models

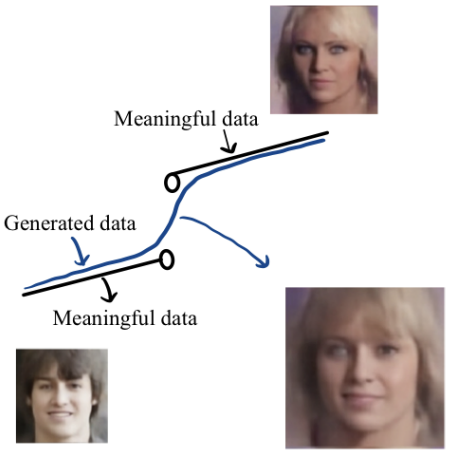
- Singular Point[9]

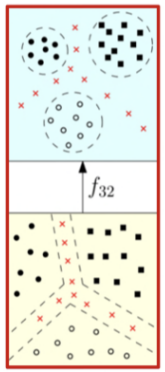


(a) convex support (b) concave support: single mode (c) concave support: multi modes

Singular point leads to discontinuity, while NN learns continuous map.



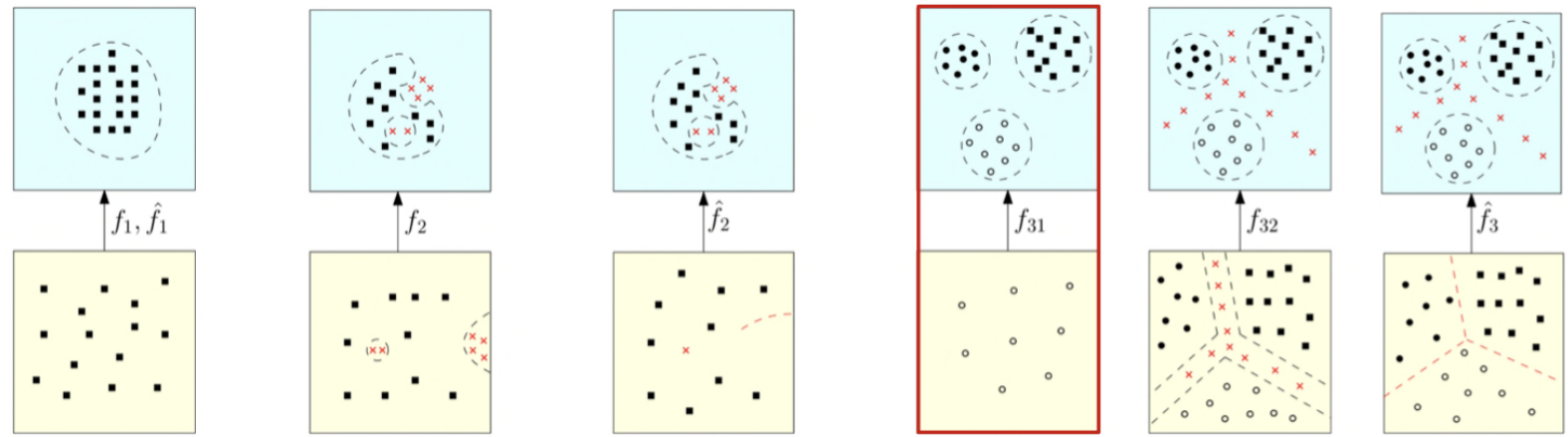




[9] Lei N., et al. Geometric Understanding of Deep Learning. 2018.

Reasons for Failure in GAN Based Models

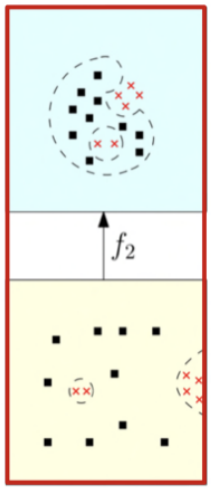
- Local Measure



convex support (b) concave support: single mode (c) concave support: multi modes

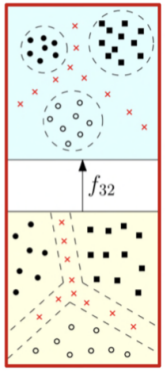
Reasons for Failure in GAN Based Models

- Single Mode Case

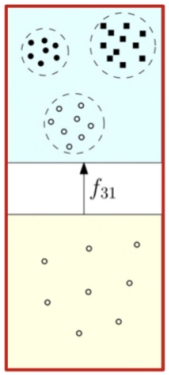


GAN/WGAN: Approximate a discontinuous map by NN

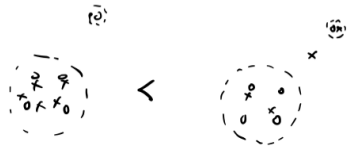
- Multi Mode Case



GAN/WGAN: Same as above

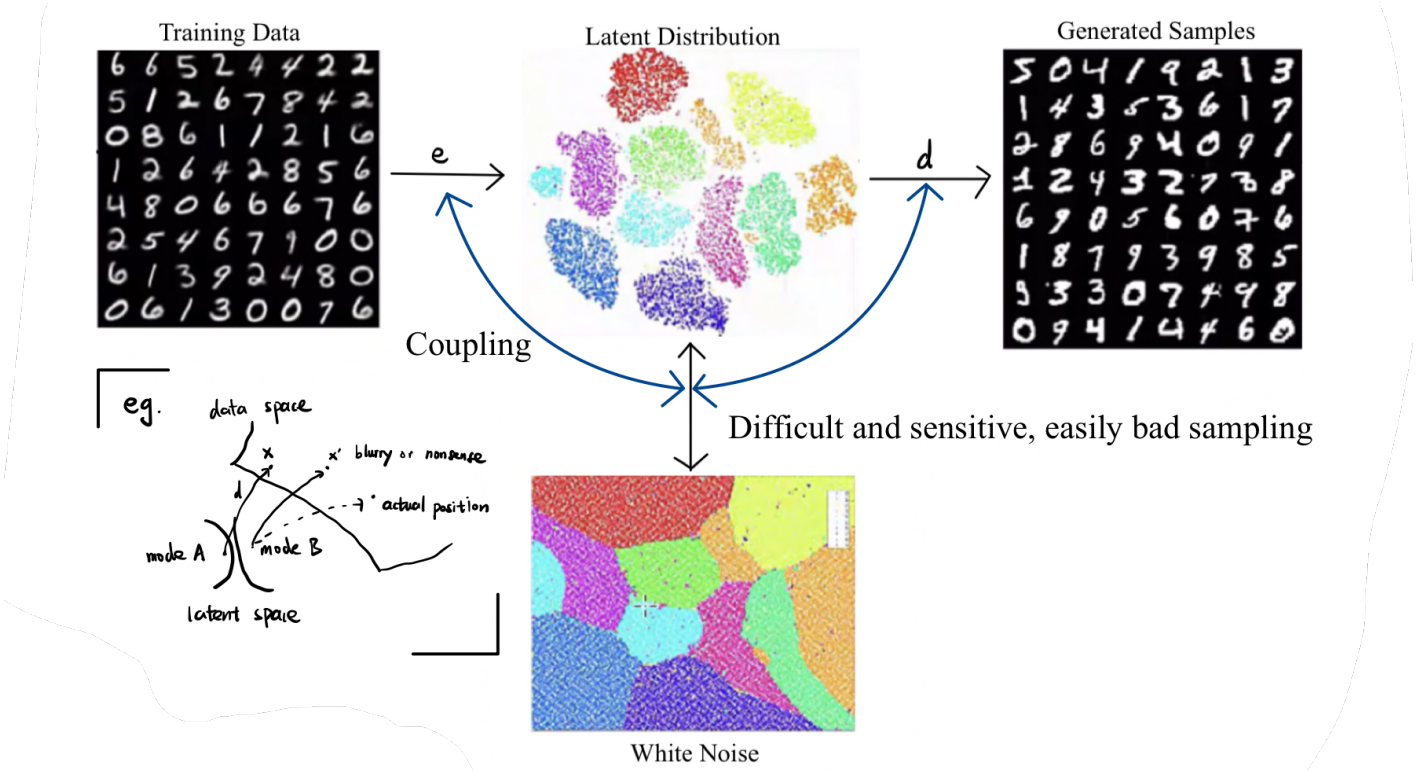


GAN: Local measure
WGAN: Average measure(far, small mode)



Reasons for Failure in AE Based Models

- Trade-off between nice latent manifold structure and quality of coding

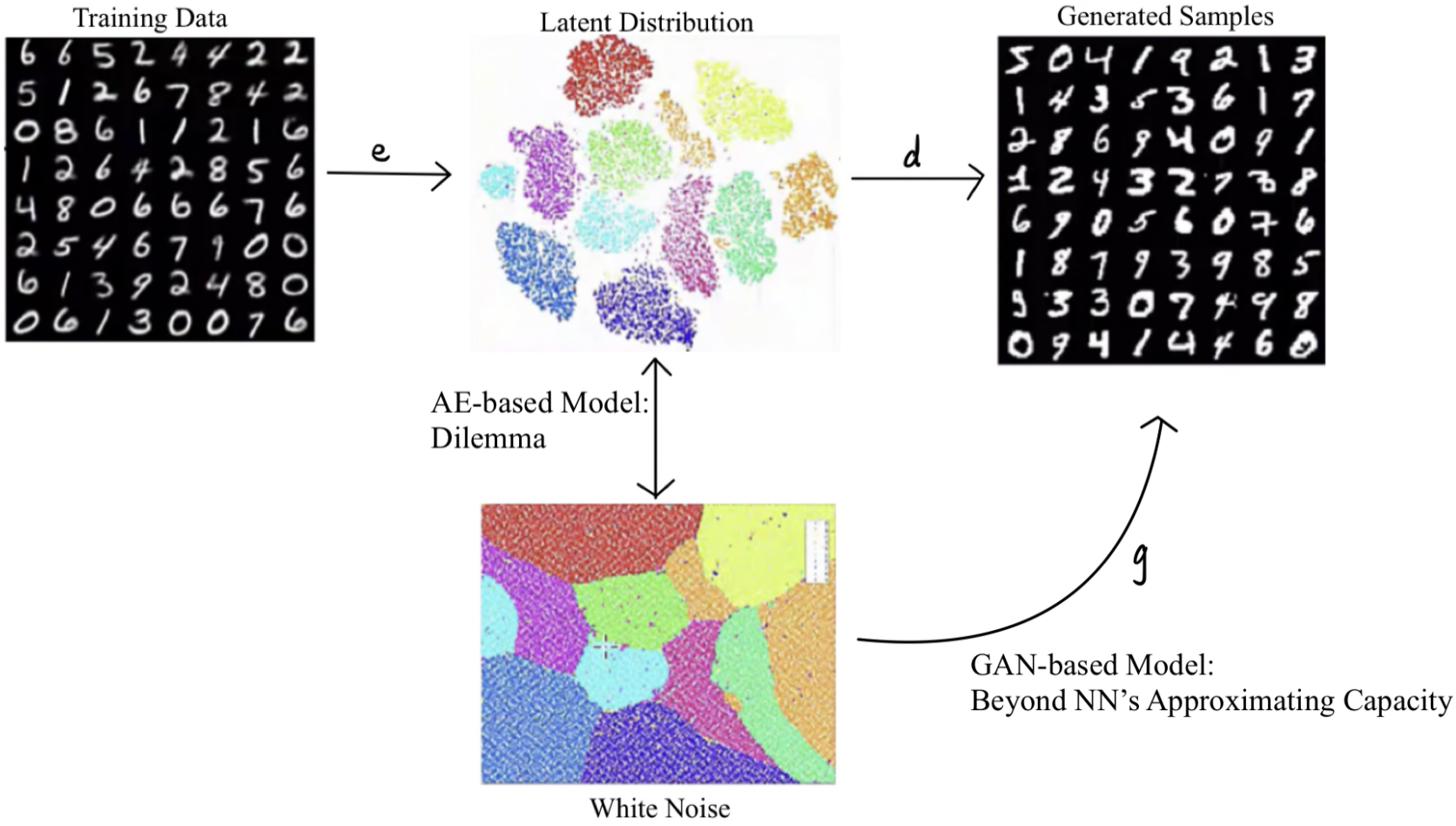


Problems of matching Q to prior P:

1. Difficult and sensitive, generating bad samples (too weak case)
2. Hinders the quality of coding (too strong case)

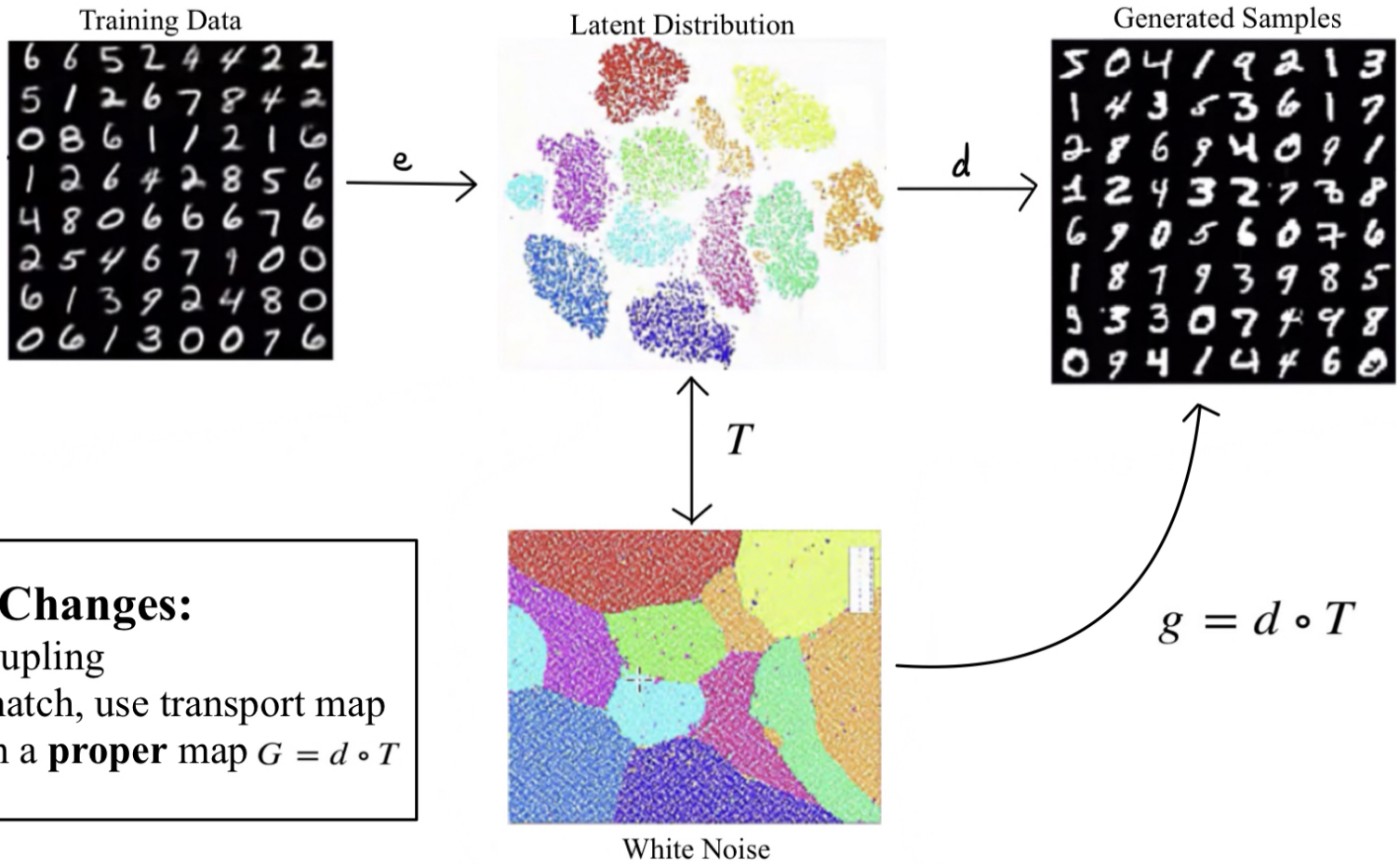
Dilemma: Match or Not Match?

Review of Problems Remained



Inspiration: At least we should not mix the two processes, i.e. manifold & measure learning.

Solution: A New Structure



Main Changes:
1. Decoupling
2. No match, use transport map
3. Learn a **proper** map $G = d \circ T$

Key: learn the proper manifold map d and measure transport map T respectively.

Measure Learning by OT

- Brenier Potential and OT[10]

Theorem 9

Suppose X and Y are the Euclidean space $\mathbb{R}^n$, and the transportation cost is the quadratic Euclidean distance $c(x, y) = |x - y|^2$. If μ is absolutely continuous and μ and ν have finite second order moment, then there exists a convex function $u : X \to \mathbb{R}$, its gradient map ∇u gives the Solution to the Monge's problem, where u is called Brenier's potential. Furthermore, the optimal mass transportation map is unique:

$$T = \nabla u$$

While T is discontinue, we can learn u instead, which is continue (convex) This is why we need T to be a OPTIMAL transport map, rather than ordinary transport map.

[10] Gu X., et al. Variational Principles for Minkowski Type Problems, Discrete Optimal Transport, and Discrete Monge-Ampere Equations. Mathematical Methods in Solid State & Superfluid Theory, 2013.

Measure Learning by OT

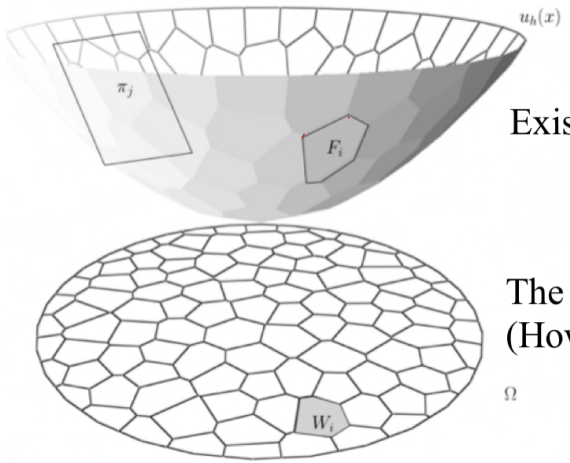
- Question: How to learn u?
  Analysis by geometic method provides some necessary conditions:
  - Locally: $\nabla u_h^i(x) = T(x) = y_i \Rightarrow u_h^i(x) =< x, y_i > +h_i, x \in \mathbb{R}^n$
  - Globally: u convex $\Rightarrow u(x) = \max_i\{u_h^i(x)\} = max_i< x, y_i > +h_i, x \in \mathbb{R}^n$
  - Measure Preserving
    $\Rightarrow w_i = v_i, W_i = \{x \in \mathbb{R}^n | u_h^i(x) = y_i\}, w_i = \mu_X(W_i \cap \Omega), \nu_i = \mu_Y(y_i)$

Measure Learning by OT

- Learning Brenier Potential by Convex Geometry Theory

Theorem 10 (the shape of *u*)

Suppose Ω is a compact convex polytope with non-empty interior in $\mathbb{R}^n$, $n_1, \cdots, n_k \subset \mathbb{R}^{n+1}$ are distinct k unit vectors, the $(n+1)$-th coordinates are negative, and $\nu_1, \cdots, \nu_k \geqslant 0$ so that $\sum_{i=1}^{k} \nu_i = vol(\Omega)$. Then there exists convex polytope $P \subset \mathbb{R}^{n+1}$ with exact k codimension-1 faces $F_1, \cdots, F_k$ so that n_i is the normal vector to F_i and the intersection between Ω and the projection of F_i is with volume v_i. Furthermore, such P is unique up to vertical translation.



Exist and unique, u is exactly the Brenier potential

The OT problem is transformed into a convex geometry problem (How to learn h?)

Measure Learning by OT

- Learning Brenier Potential by Convex Geometry Theory

Theorem 11 (compute u)

Let Ω be a compact convex domain in $\mathbb{R}^n$, $\{y_1, \cdots, y_k\}$ be a set of distinct points in $\mathbb{R}^n$ and μ a probability measure on Ω. Then for any $v_1, \cdots, v_k \geqslant 0$ with $\sum_{i=1}^k v_i = vol(\Omega)$, there exists $h = (h_1, \cdots, h_k) \in \mathbb{R}^k$, unique up to adding a constant $(c, \cdots, c)$, so that $w_i(h) = v_i$, for all i. The vectors h are exactly maximum points of the concave function

$$E(h) = \sum_{i=1}^{k} h_i v_i - \int_0^h \sum_{i=1}^{k} w_i(\eta) d\eta_i$$

on the open convex set $H = \{h \in \mathbb{R}^k | w_i(h) \geqslant 0\}$. Furthermore, ∇u_h minimizes the quadratic cost $\int_\Omega |x - T(x)|^2 d\mu(x)$ among all transport maps $T_\# \mu = \nu$, where the Dirac measure $\nu = \sum_{i=1}^k v_i \delta_{y_i}$.

Measure Learning by OT

- Learning Brenier Potential by Convex Geometry Theory
  The optimization is a convex problem.
  Jacobian:

$$\nabla E(h) = (\nu_1 - w_1(h), \nu_2 - w_2(h), \cdots, \nu_k - w_k(h))^T$$

  Hessian:

$$\frac{\partial^2 E(h)}{\partial h_i^2} = \frac{\partial w_i(h)}{\partial h_i} = \sum_{j \neq i} \frac{\partial w_i(h)}{\partial h_j} = -\frac{\mu(D_{ij})}{|e_{ij}|}$$

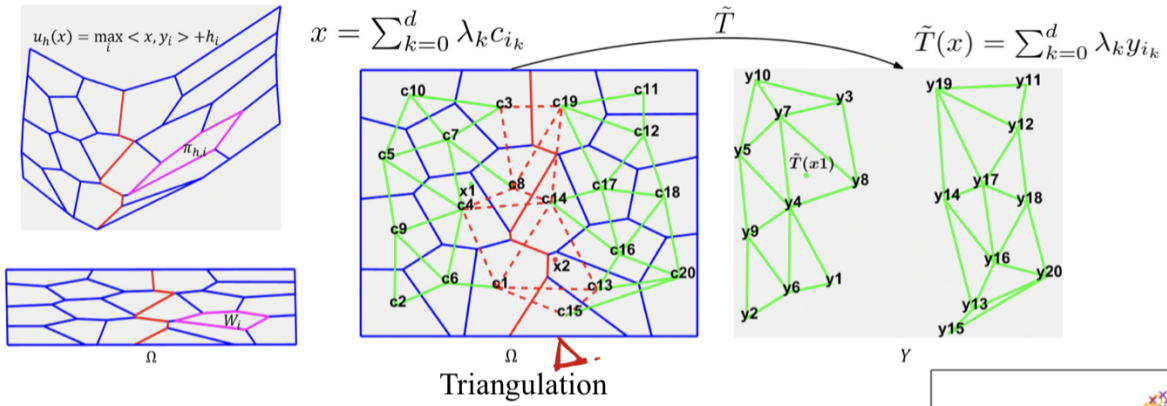
$$\frac{\partial^2 E(h)}{\partial h_i^2} = \frac{\partial w_i(h)}{\partial h_i} = \sum_{j \neq i} \frac{\partial w_i(h)}{\partial h_j}$$

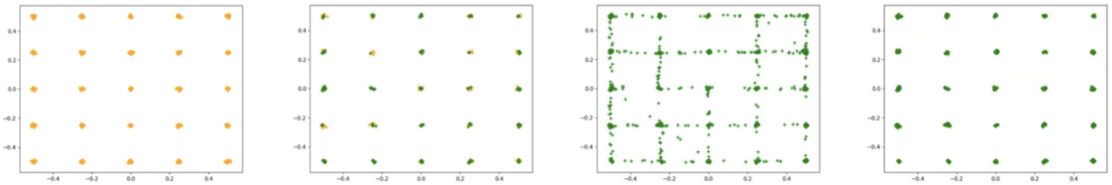
  Optimized by the Newton Method:

$$x_{n+1} = x_n - [Hf(x_n)]^{-1} \nabla f(x_n), n \geqslant 0$$

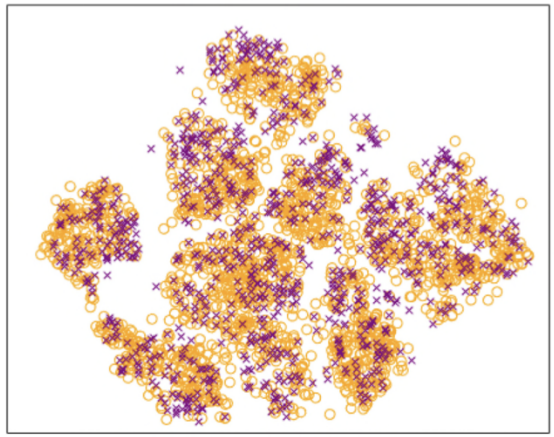
Measure Learning by OT

- Singularity Test and Extension of SDOT[11]



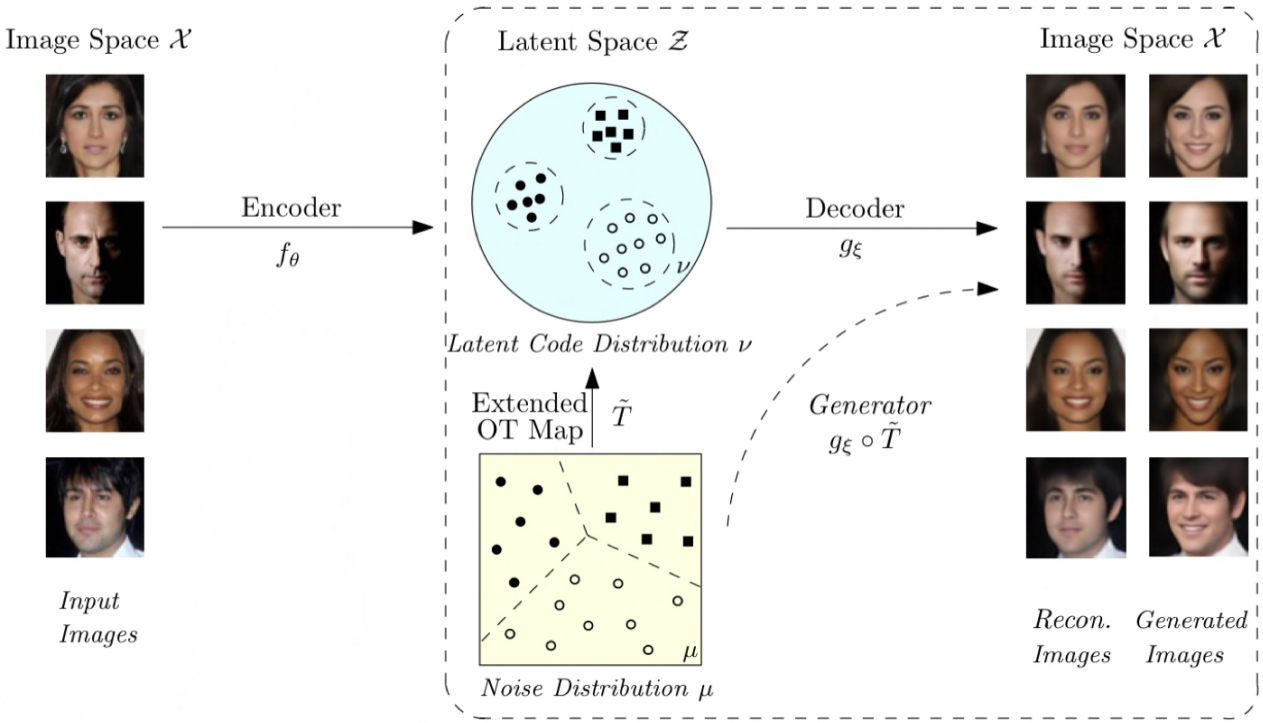


(a) Target Distribution (b) $\hat{\theta}$ too small (c) $\hat{\theta}$ too large (d) Proper $\hat{\theta}$



[11] An D., et al. AE-OT-GAN: Training GANs from data specific latent distribution. 2020.

AE-OT: Theory



Algorithm 1 Semi-discrete OT Map

Input: Latent codes $Y = \{y_i\}_{i \in \mathcal{I}}$, empirical latent code distribution $\nu = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \delta_{y_i}$, number of Monte Carlo samples N, positive integer s.
Output: Optimal transport map $T(\cdot)$.
Initialize $h = (h_1, h_2, \ldots, h_{|\mathcal{I}|}) \leftarrow (0, 0, \ldots, 0)$.
repeat
  Generate N uniformly distributed samples $\{x_j\}_{j=1}^N$.
  Calculate $\nabla h = (\hat{w}_i(h) - \nu_i)^T$.
  $\nabla h = \nabla h - mean(\nabla h)$.
  Update h by Adam algorithm with $\beta_1 = 0.9, \beta_2 = 0.5$.
  if $E(h)$ has not decreased for s steps **then**
    $N \leftarrow N \times 2$.
  end if
until Converge
OT map $T(\cdot) \leftarrow \nabla(\max_i \langle \cdot, y_i \rangle + h_i)$.

Algorithm 2 Generate latent code

Input: Optimal transport map $T(\cdot)$, number of samples to generate n, angle threshold $\hat{\theta}$.
Output: Generated latent code P.
Compute $\hat{c}_i$ by Monte Carlo method.
repeat
  Sample $x \sim \mu$, Find the smallest $d+1$ vertex around x as $\{d(x, \hat{c}_{i_0}), d(x, \hat{c}_{i_1}), \ldots, d(x, \hat{c}_{i_d})\}$.
  Compute dihedral angles θ_{i_k} between π_{i_0} and π_{i_k}.
  Select θ_{i_k} with $\theta_{i_k} \leq \hat{\theta}$, result in $\hat{i}_k = 0, 1, \ldots, d_1$.
  if $\forall k, \theta_{i_k} > \hat{\theta}$ **then** Abandon x
  else Generate latent code $\tilde{T}(x) = \sum_{k=0}^{d_1} \lambda_k T(\hat{c}_{\hat{i}_k})$ with $\lambda_k = d^{-1}(x, \hat{c}_{\hat{i}_k}) / \sum_{j=0}^{d_1} d^{-1}(x, \hat{c}_{\hat{i}_j})$.
  end if
until Generate n new latent code

AE-OT: Result

- MM & MC are solved

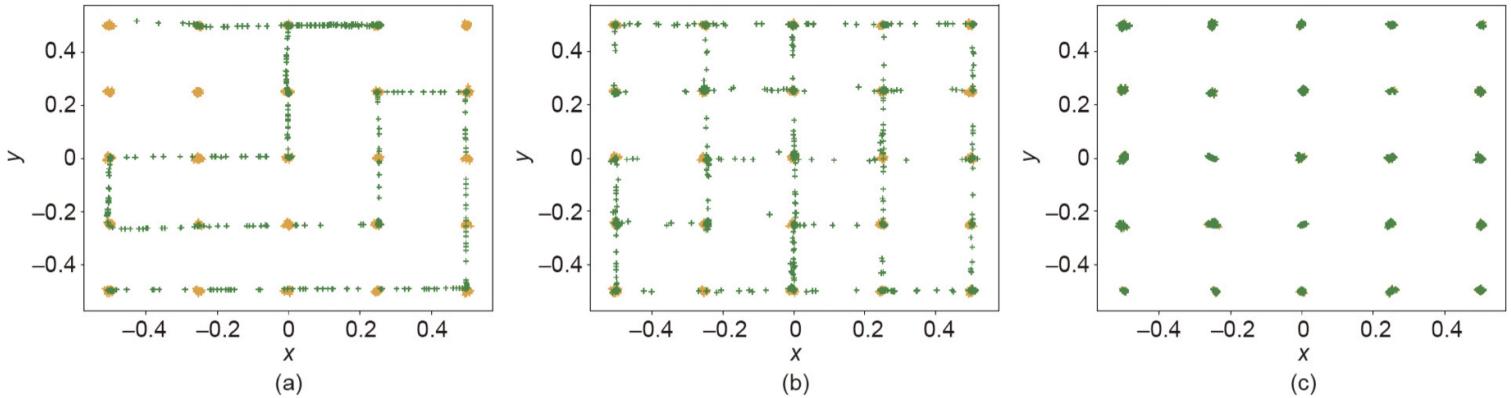


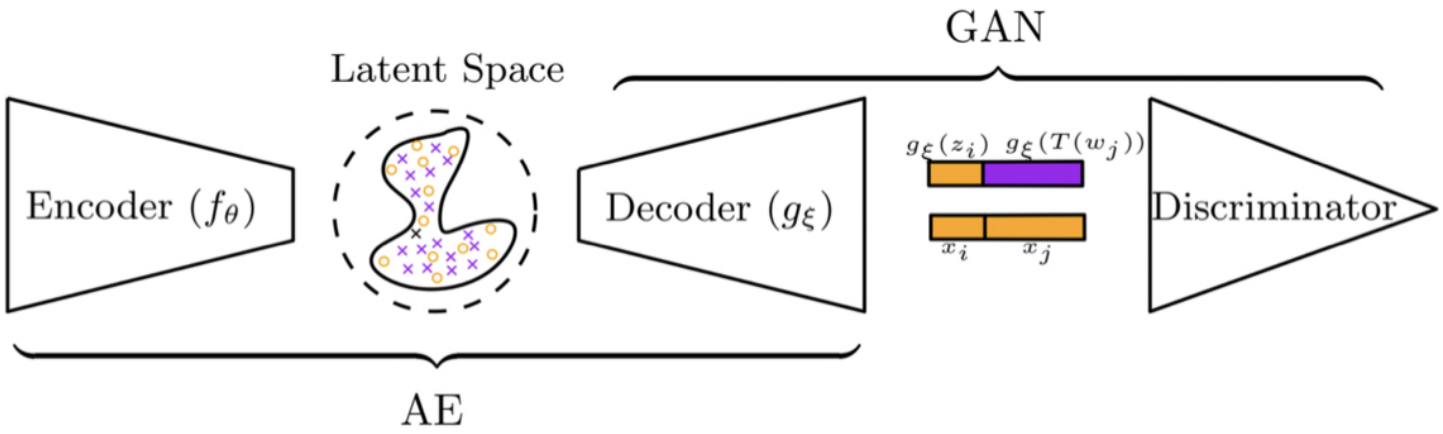
Fig. 13. Mode collapse comparison on a 2D grid dataset. (a) GAN; (b) PacGAN4; (c) AE–OT. Orange marks are real samples and green marks are generated ones.

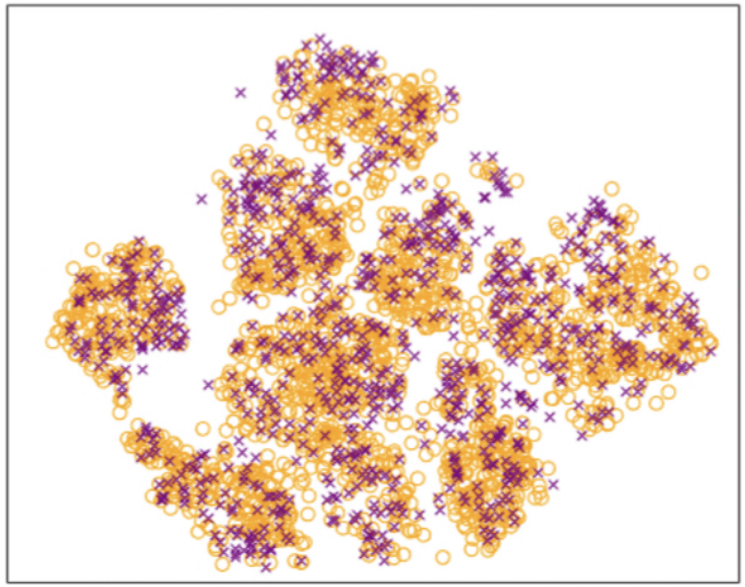
- Generated data quality

Table 2: Quantitative comparison with FID

	Adversarial				Non-Adversarial		Reference	
Dataset	NS GAN	LSGAN	WGAN	BEGAN	VAE	GLANN	AE	Ours
MNIST	6.8±0.5	7.8±0.6	6.7±0.4	13.1±1.0	23.8±0.6	8.6±0.1	5.5	**6.2±0.2**
Fansion	26.5±1.6	30.7±2.2	21.5±1.6	22.9±0.9	58.7±1.2	13.0±0.1	4.7	**10.1±0.3**
CIFAR-10	58.5±1.9	87.1±47.5	55.2±2.3	71.4±1.6	65.4±0.2	46.5±0.2	28.2	**38.3±0.5**
CelebA	55.0±3.3	53.9±2.8	41.3±2.0	**38.9±0.9**	85.7±3.8	46.3±0.1	67.5	68.4± 0.5

AE-OT-GAN: Theory





AE-OT-GAN: Experimental Result

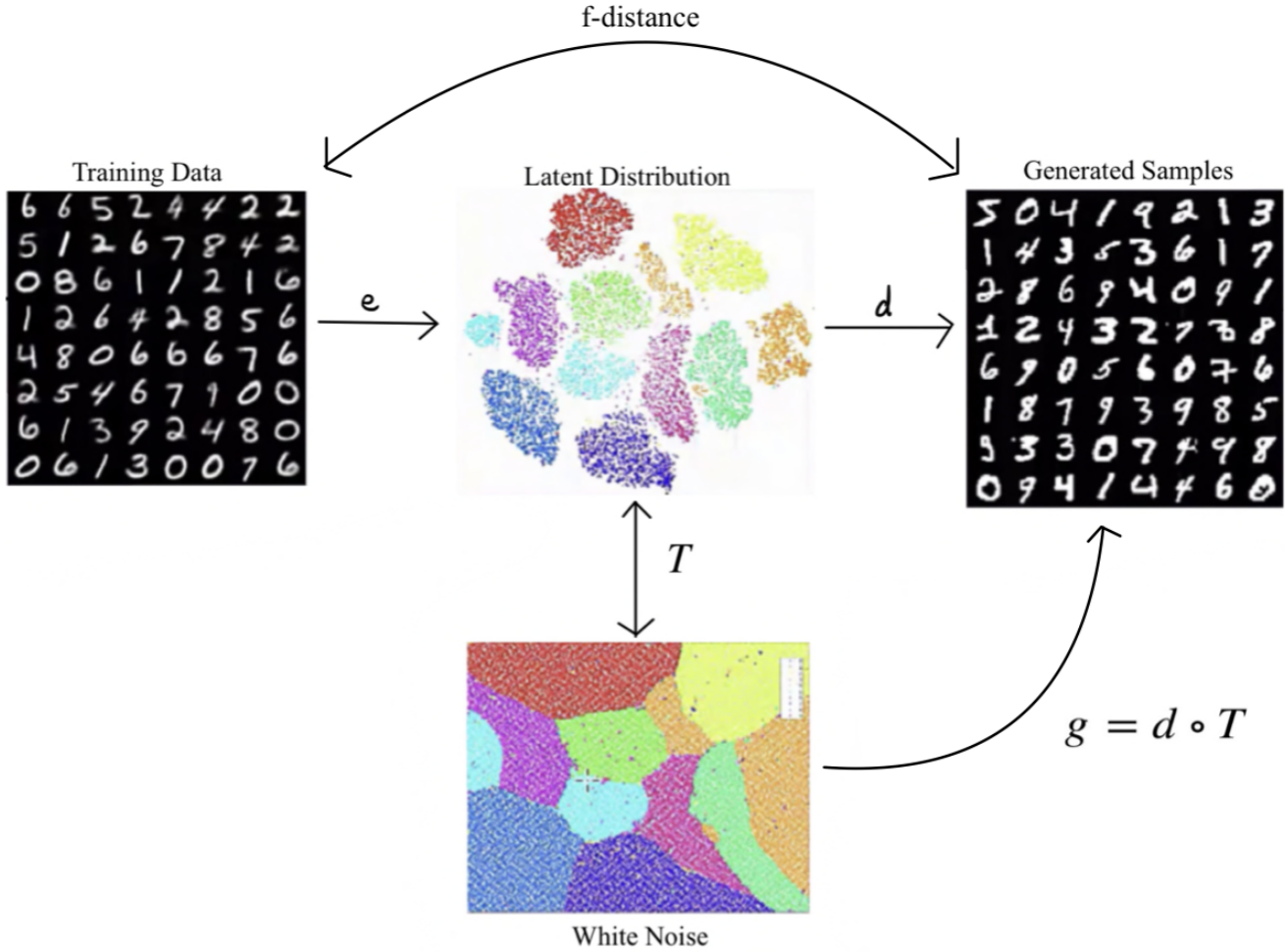
	CIFAR10		CelebA	
	Standard	ResNet	Standard	ResNet
WGAN-GP [14]	40.2	19.6	21.2	18.4
PGGAN [38]	-	18.8	-	16.3
SNGAN [32]	25.5	21.7	-	-
WGAN-div [20]	-	18.1	17.5	15.2
WGAN-QC [29]	-	-	-	12.9
AE-OT [2]	34.2	28.5	24.3	28.6
AE-OT-GAN	25.2	17.1	11.2	7.8

Table 1. The comparison of FID between the proposed method and the state of the arts on Cifar10 and CelebA.

Outline

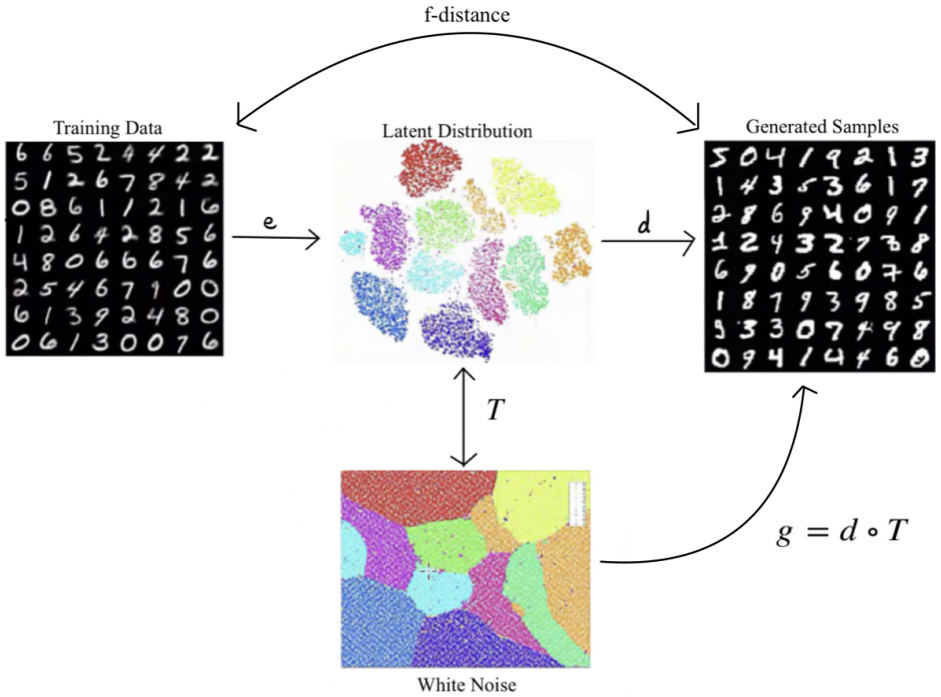
Conclusion and Further Problems

- Decoupling Method
  1. Manifold Learning process remains uncertain.
  2. The computation complexity of OT map.



Conclusion and Further Problems

- Coupling Method
  1. Mode Collapse & Mode Mixture
  2. Is W-distance really useful?[12]



[12] Stanczuk J., et al. Wasserstein GANs Work Because They Fail (to Approximate the Wasserstein Distance). 2021.