Optimal Transport for Generative Modeling

Presenter: Son Nguyen
VinAI Resident

21/10/2020

Outline

1. A brief review of Optimal Transport

- Monge/Kantorovich formulation
- Wasserstein distance
- Sliced Wasserstein distance

2. Recap Deep Generative Models

- Variational Autoencoders (VAE)
- Generative Adversarial Networks (GAN)

3. Generative Modeling from Optimal Transport view

- (Sliced) Wasserstein Generative Adversarial Networks (WGAN, SWGAN)
- (Sliced) Wasserstein Autoencoders (WAE, SWAE)

4. References

Outline

1. A brief review of Optimal Transport
   - Monge/Kantorovich formulation
   - Wasserstein distance
   - Sliced Wasserstein distance

2. Recap Deep Generative Models
   - Variational Autoencoders (VAE)
   - Generative Adversarial Networks (GAN)

3. Generative Modeling from Optimal Transport view
   - (Sliced) Wasserstein Generative Adversarial Networks (WGAN, SWGAN)
   - (Sliced) Wasserstein Autoencoders (WAE, SWAE)

4. References

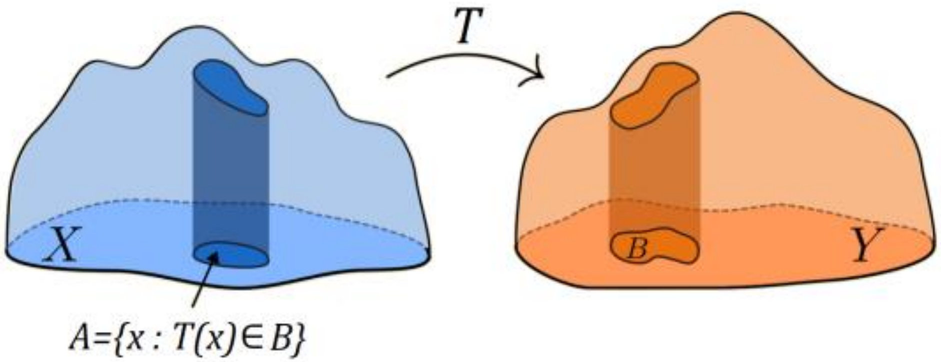
A brief review of Optimal Transport

❑ Monge formulation

Definition: We say that $T : X \to Y$ transports $\mu \in \mathcal{P}(X)$ to $v \in \mathcal{P}(Y)$ and we call it a **transport map** if:

$v(B) = \mu(T^{-1}(B))$ or $v(B) = \mu(A)$ for all v-measurable sets B

shorthand: $v = T_{\#}\mu$



A brief review of Optimal Transport

❑ Monge formulation

Monge's Optimal Transport Problem:

Given $\mu \in \mathcal{P}(X)$ and $v \in \mathcal{P}(Y)$:

$$min_T \mathbb{M}(T) = \int_X c(x, T(x)) d\mu(x)$$

over measurable maps $T : X \to Y$ subject to $v = T_{\#}\mu$

- Monge only considered the problem with $c(x, y) = |x - y|$. (super hard with L^2 cost)
- The key of hardness in Monge's problem is the **non-linear** constraint: $v(B) = \mu(T^{-1}(B))$
- In continuous case, the constraint require transport map is **bijective** and **differentiable**, it is equivalent to:

$$f(x) = g(T(x))|det(\nabla T(x))| \text{ ,where } d\mu(x) = f(x)dx, dv(y) = g(y)dy$$

A brief review of Optimal Transport

❑ Monge formulation

Monge Formulation's cons:

- mass is **mapped**, it means that mass is **not split → hard constraint**
- transport map may be not exist.

For example: $\mu = \delta_{x_1}, v = \frac{1}{2}\delta_{y_1} + \frac{1}{2}\delta_{y_2}$ then $v(y_1) = \frac{1}{2}$ but $\mu(T^{-1}(y_1)) \in \{0, 1\}$ depending on weather $x_1 \in T^{-1}(y_1)$. Hence no transport maps exist

There are two importance cases where transport maps exist:

1. The discrete case when $\mu = \frac{1}{n}\sum_{i=1}^{N} \delta_{x_i}$ and $v = \frac{1}{n}\sum_{j=1}^{N} \delta_{y_j}$
2. The absolutely continuous case when $d\mu(x) = f(x)dx$ and $dv(y) = g(y)dy$

A brief review of Optimal Transport

❑ Kantorovich Formulation

- Consider a measure $\pi \in \mathcal{P}(X,Y)$ and think of $d\pi(x,y)$ as the amount of mass transferred from x to y. This allows mass can be moved to **multiple locations**
- We have the constraints:

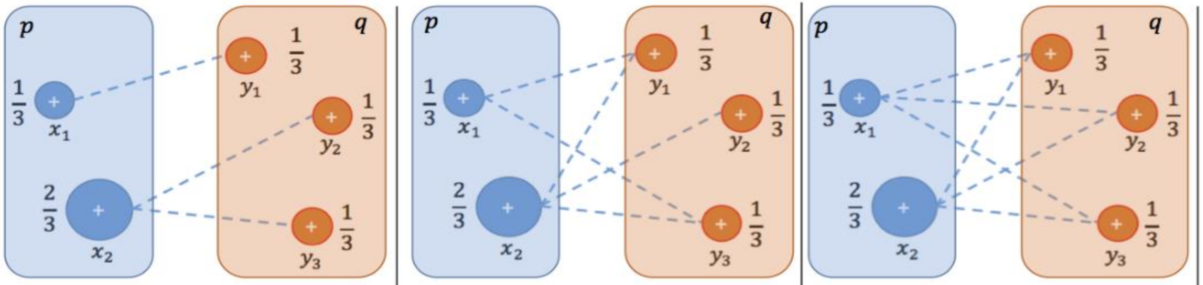
$\pi(A \times Y) = \mu(A)$ and $\pi(X \times B) = v(B)$ for all measurable sets $A \subseteq X, B \subseteq Y$

- π is a **joint distribution** which has first marginal $\mu \in \mathcal{P}(X)$ and second marginal $v \in \mathcal{P}(Y)$

- π is called **transport plan** and set of such transport plan $\Pi(\mu, v)$

A brief review of Optimal Transport

❑ Kantorovich Formulation



$$\gamma = \begin{array}{c} \\ \\ \end{array}\begin{array}{ccc|c} y_1 & y_2 & y_3 & \\ \frac{1}{3} & 0 & 0 & x_1 \\ 0 & \frac{1}{3} & \frac{1}{3} & x_2 \end{array}$$

$$\gamma = \begin{array}{ccc|c} y_1 & y_2 & y_3 & \\ \frac{1}{9} & 0 & \frac{2}{9} & x_1 \\ \frac{2}{9} & \frac{1}{3} & \frac{1}{9} & x_2 \end{array}$$

$$\gamma = \begin{array}{ccc|c} y_1 & y_2 & y_3 & \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} & x_1 \\ \frac{2}{9} & \frac{2}{9} & \frac{2}{9} & x_2 \end{array}$$

$$\sum_i \gamma_{i\cdot} = [\frac{1}{3} \quad \frac{1}{3} \quad \frac{1}{3}] \qquad \sum_j \gamma_{\cdot j} = \begin{bmatrix} \frac{1}{3} \\ \frac{2}{3} \end{bmatrix}$$

A brief review of Optimal Transport

❑ Kantorovich Formulation

Kantorovich's Optimal Transport Problem:

Given $\mu \in \mathcal{P}(X)$ and $v \in \mathcal{P}(Y)$

$$min_{\pi}\mathbb{K}(\pi) = \int_{X\times Y} c(x,y)d\pi(x,y)$$

Assume that there exists a optimal transport map $T^* : X \to Y$ subject to Monge formulation. Then we define $d\pi(x,y) = d\mu(x)\delta_{y=T^*(x)}$. It is easy to show that $\pi \in \Pi(x,y)$

$$\pi(A \times Y) = \int_A \delta_{T^*(x)\in Y} d\mu(x) = \mu(A)$$

$$\pi(X \times B) = \int_X \delta_{T^*(x)\in B} d\mu(x) = \mu((T^*)^{-1}(B)) = v(B)$$

$$\int_{X\times Y} c(x,y)d\pi(x,y) = \int_X \int_Y c(x,y)\delta_{y=T^*(x)} dy d\mu(x) = \int_X c(x,T^*(x))d\mu(x)$$

A brief review of Optimal Transport

❑ Kantorovich Formulation

Kantorovich's Optimal Transport Problem:

Kantorovich problem between two **discrete measures** $\mu = \sum_{i=1}^{m} \alpha_i \delta_{x_i}, v = \sum_{j=1}^{n} \beta_j \delta_{y_j}$ where $\sum_{i=1}^{m} \alpha_i = 1 = \sum_{j=1}^{n} \beta_j, \alpha_i \geq 0, \beta_j \geq 0$ then Kantorovich problem become a linear programme with linear constraint.

$$min_{\pi} \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} \pi_{ij}$$

A brief review of Optimal Transport

❑ Kantorovich Formulation

Kantorovich’s Optimal Transport Problem:

Primal problem: $KP(\mu, v) = min_\pi \int_{X\times Y} c(x,y) d\pi(x,y)$

$$\pi(A \times Y) = \mu(A) \quad \pi(X \times B) = v(B)$$

Dual problem: $DP(\mu, v) = \sup_{(\varphi,\psi)\in\Phi_c} \int_X \varphi d\mu + \int_Y \psi dv$

$$\Phi_c = \{(\varphi, \psi) \in L^1(\mu) \times L^1(v) : \varphi(x) + \psi(y) \leq c(x,y)\}$$

$$\int_X |f| d\mu < \infty$$

$$\boxed{DP(\mu, v) \leq KP(\mu, v)}$$

A brief review of Optimal Transport

❑ Wasserstein Distance

Definition: Let μ, v are two probability measures in the set of probability measure with finite $p'th$ moment defined on a given metric space (Ω, d), i.e. exist some x_0:

$$\int_\Omega d(x, x_0)^p d\mu(x) < +\infty$$

For $p \geq 1, c(x, y) = d^p(x, y) = |x - y|^p$ then:

$$W_p(\mu, v) = (\min_{\pi \in \Pi(\mu, v)} \int_{\Omega \times \Omega} d^p(x, y) d\pi(x, y))^{\frac{1}{p}}$$

When $p = 1$ Wasserstein Distance becomes Earth Mover Distance

A brief review of Optimal Transport

❑ **Wasserstein Distance**

Kantorovich dual form of 1-Wasserstein:

$$W_1(\mu,\nu) = \sup_{\substack{f,g \\ f(x)+g(y)\leq\|x-y\|}} \int f d\mu(x) + \int g d\nu(y)$$

$$= \sup_{f} \int f d\mu(x) - \int f d\nu(y) \quad \text{where } f:\mathbb{R}^d \to \mathbb{R},\ \mathrm{Lip}(f) \leq 1$$

A brief review of Optimal Transport

❑ Wasserstein Distance

<u>Special case</u>: Wasserstein distance **has closed-form** solution in **one dimension.**

- Discrete case: $\mu = \frac{1}{n}\sum_{i=1}^{N}\delta_{x_i}$ and $v = \frac{1}{n}\sum_{j=1}^{N}\delta_{y_j}$. Sort $x_1 \leq \ldots \leq x_n$ and $y_1 \leq \ldots \leq y_n$

$$\boxed{W_p^p(\mu, v) = \frac{1}{n}\sum_{i=1}^{n}|x_i - y_i|^p}$$

- Continuous case:
  - the cumulative distribution function: $F_\mu(x) = \mu((-\infty, x]) = \int_{-\infty}^{x} I_\mu(\tau)d\tau$
  - the pseudo-inverse: $F_\mu^{-1}(t) = \inf\{x \in \mathbb{R} : F_\mu(x) \geq t\}$
  - the unique optimal transport map: $f(x) = F_v^{-1}(F_\mu(x))$

$$\boxed{W_p(\mu, v) = \left(\int_X d^p(x, F_v^{-1}(F_\mu(x)))d\mu(x)\right)^{\frac{1}{p}} = \left(\int_0^1 d^p(F_\mu^{-1}(z), F_v^{-1}(z))dz\right)^{\frac{1}{p}}}$$

A brief review of Optimal Transport

❑ Sliced Wasserstein distance

Randon transform:

- **project higher-dimensional** probability densities into sets of **one-dimensional** marginal distributions and compare these marginal distributions via the Wasserstein distance.

  → take advantage of the **closed-form solution** of Wasserstein distance on 1-D.

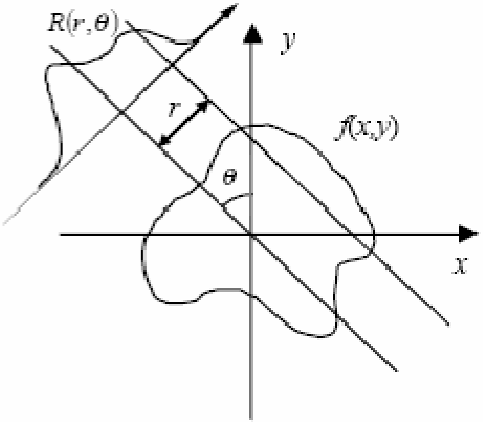
- These **one dimensional** marginal distributions obtained through **Radon Transform:**

$$\boxed{\mathcal{R}p_X(t;\theta) = \int_X p_X(x)\delta(t-\theta\cdot x)dx, \ \ \forall\theta\in\mathbb{S}^{d-1}, \ \forall t\in\mathbb{R}}$$

$p_X(x)$ is a d −dimensional probability density,

$\mathbb{S}^{d-1}$ is the d-dimensional unit sphere

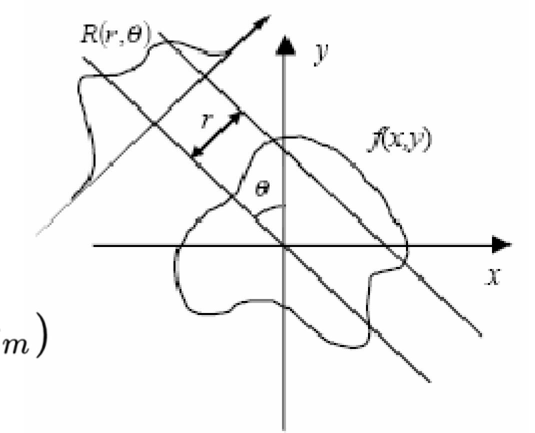
$\mathcal{R}_{p_X}(;\theta)$ is a one-dimensional slice of $p_X(x)$



A brief review of Optimal Transport

❑ Sliced Wasserstein distance

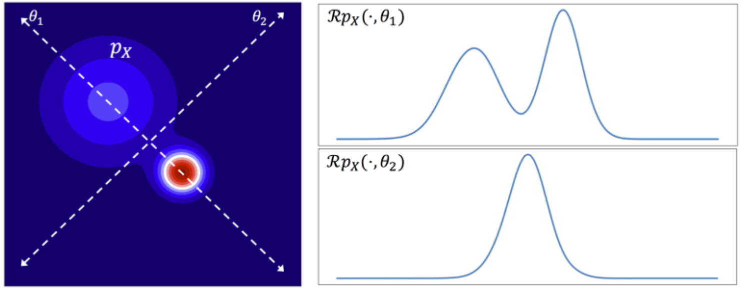
Randon transform:



$$\mathcal{R}p_X(t;\theta) = \int_X p_X(x)\delta(t-\theta\cdot x)dx,\ \ \forall\theta\in\mathbb{S}^{d-1},\ \forall t\in\mathbb{R}$$

Radon Transform of a empirical distribution $p_X(x) = \frac{1}{M}\sum_{m=1}^{M}\delta(x-x_m)$ respect to $\theta\in\mathbb{S}^{d-1}$:

$$\begin{aligned} Rp_X(t,\theta) &= \frac{1}{M}\sum_{m=1}^{M}\int_X \delta(x-x_m)\delta(t-\langle\theta,x\rangle)dx \\ &= \frac{1}{M}\sum_{m=1}^{M}\delta(t-\langle\theta,x_m\rangle) \end{aligned}$$



A brief review of Optimal Transport

❑ Sliced Wasserstein distance

Formulation:

Given two probability measures μ, v with the probability density I_μ, I_v respectively:

$$SW_p(\mu, v) = \left(\int_{\mathbb{S}^{d-1}} W_p^p(\mathcal{R}I_\mu(.\,,\theta), \mathcal{R}I_v(.\,,\theta))d\theta \right)^{\frac{1}{p}}$$
$$\approx \left(\frac{1}{L} \sum_{l=1}^{L} W_p^p(\mathcal{R}I_\mu(.\,,\theta_l), \mathcal{R}I_v(.\,,\theta_l)) \right)^{\frac{1}{p}}$$

(use Monte Carlo scheme to approximate SW_p distance by drawn samples θ_l uniformly on $\mathbb{S}^{d-1}$)

- $SW_p^p(\mu, v) \leq \alpha_{d,p} W_p^p(\mu, v)$, with $\alpha_{d,p} = \frac{1}{d} \int_{\mathbb{S}^{d-1}} \|\theta\|_p^p d\theta \leq 1$
- The **sensitivity** and **discriminativeness** of Sliced Wasserstein distance depend on the number and the importance of projections L.

A brief review of Optimal Transport

❑ Sliced Wasserstein distance

Slice-based improved distances:

- **Max-Sliced Wasserstein distance**: to find a **single** linear projection that **maximizes** the distance of the probability measures in the projected space.

$$\boxed{max - SW_p(I_\mu, I_v) = max_{\theta \in \mathbb{S}^{d-1}} W_p(\mathcal{R}I_\mu(.\,,\theta), \mathcal{R}I_v(.\,,\theta))}$$

E.g: $I_\mu = \mathcal{N}(0, I), I_v = \mathcal{N}(x_0, I)$ then $\mathcal{R}I_\mu(.\,,\theta) = \mathcal{N}(0,1), \mathcal{R}I_v(.\,,\theta) = \mathcal{N}(\langle x_0, \theta\rangle, I)$.

In high dimension space, sampled uniform θ would be nearly orthogonal to a fixed vector x_0

→ the sliced distance will be 0 → the best direction is $\theta = x_0$

A brief review of Optimal Transport

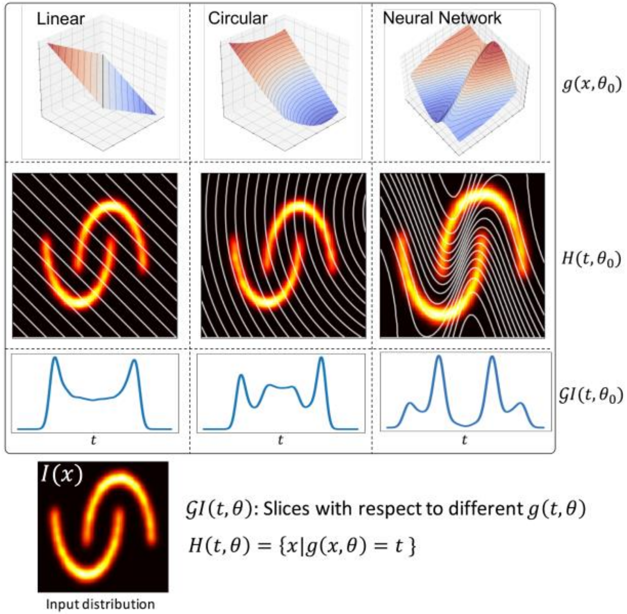
❑ Sliced Wasserstein distance

Slice-based improved distances:

- **Generalized Sliced-Wasserstein distance**: using **Generalized Radon Transform** which projects original distribution on **hypersurface:**

$$\mathcal{G}I(t,\theta) = \int_{\mathbb{R}^d} I(x)\delta(t - g(x,\theta))dx$$

$$GSW_p(I_\mu, I_v) = \left(\int_{\Omega_\theta} W_p^p(\mathcal{G}I_\mu(.\,,\theta), \mathcal{G}I_v(.\,,\theta))d\theta \right)^{\frac{1}{p}}$$



$\mathcal{G}I(t,\theta)$: Slices with respect to different $g(t,\theta)$

$H(t,\theta) = \{x|g(x,\theta) = t\}$

- **Generalized max Sliced-Wasserstein distance:**

$$max - GSW_p(I_\mu, I_v) = max_{\theta\in\Omega_\theta} W_p(\mathcal{G}I_\mu(.\,,\theta), \mathcal{G}I_v(.\,,\theta))$$

Outline

1. A brief review of Optimal Transport
    - Monge/Kantorovich formulation
    - Wasserstein distance
    - Sliced Wasserstein distance

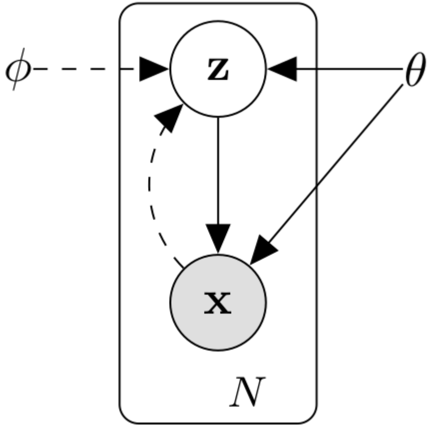
2. Recap Deep Generative Models
    - Variational Autoencoders (VAE)
    - Generative Adversarial Networks (GAN)

3. Generative Modeling from Optimal Transport view
    - (Sliced) Wasserstein Generative Adversarial Networks (WGAN, SWGAN)
    - (Sliced) Wasserstein Autoencoders (WAE, SWAE)

4. References

Recap Deep Generative Models

❑ Variational Autoencoders (VAE)



- A directed probabilistic model with **latent variable** z, global parameter θ:

$$p_\theta(x, z) = p_\theta(z) p_\theta(x|z)$$

- **Goal**: maximize the marginal log-likelihood of the dataset:

$$\log p_\theta(X) = \Sigma_{i=1}^{n} \log p_\theta(x_i)$$

- **Challenge**: marginal log-likelihood of any data point is **intractable** in general

- **Key idea:** Use variational (E-M) method → maximize a **variational lower bound** instead:

$$\boxed{\begin{aligned}\log p_\theta(x) &= \mathcal{L}(\theta, \phi; x) + \mathcal{KL}(q_\phi(z|x) \| p_\theta(z|x)) \\ &\geq \mathcal{L}(\theta, \phi; x) = \mathbb{E}_{q_\phi(z|x)} \log p_\theta(x|z) - \mathcal{KL}(q_\phi(z|x) \| p_\theta(z))\end{aligned}}$$

Recap Deep Generative Models

❑ Variational Autoencoders (VAE)

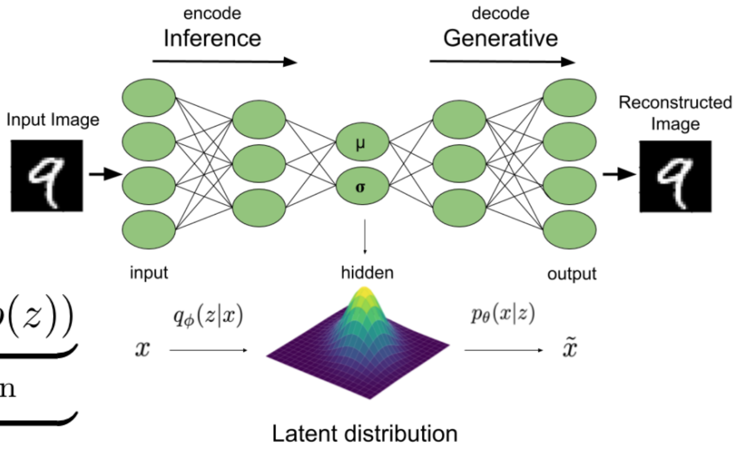
$$\mathcal{L}(\theta, \phi; x) = \mathbb{E}_{q_\phi(z|x)} \log p_\theta(x|z) - \mathcal{KL}(q_\phi(z|x) \| p_\theta(z))$$

- **Algorithm**: maximize the variational lower bound
  - use **amortized inference:** variational parameter ϕ is output of a mapping parametrized by a neural net with input x. (this neural net is **global**)
  - optimize ϕ, θ with stochastic gradient method
    - use Monte Carlo sampling + **reparametrization trick** to estimate gradient.

Recap Deep Generative Models

❑ Variational Autoencoders (VAE)

- **The Autoencoder perspective:**



$$\log p_\theta(x) \geq \underbrace{\underbrace{(E_{z\sim q_x(z)} \log p_\theta(x|z))}_{\text{Reconstruction loss}} \quad - \quad \underbrace{KL(q_\phi(z|x)||p(z))}_{\text{Regularization}}}_{L(\theta,\phi)\text{ - VAE objective}}$$

- $q_\phi(z|x)$: probabilistic **encoder** or **inference** network
- $p_\theta(x|z)$: probabilistic **decoder** or **generative** network (θ is a neural net)

Recap Deep Generative Models

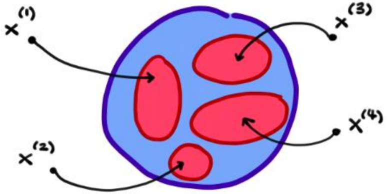
❑ Variational Autoencoders (VAE)

- **The Autoencoder perspective:** $\log p_\theta(x) \geq \underbrace{\underbrace{\left(E_{z\sim q_x(z)} \log p_\theta(x|z)\right)}_{\text{Reconstruction loss}} \quad - \quad \underbrace{KL(q_\phi(z|x)||p(z))}_{\text{Regularization}}}_{L(\theta,\phi)\text{ - VAE objective}}$

- Variational objective of VAE has **two goals with a trade-off**: <u>reconstruct and generate</u> or equivalently <u>inference and learning</u>

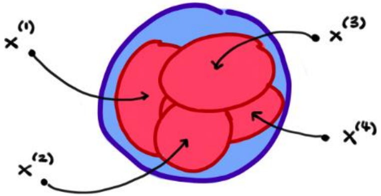
  $\hat{z} \sim q_\phi(z|x), \hat{x} \sim p_\theta(x|\hat{z})$ → reconstruction

  $\hat{x} \sim p_\theta(x) \leftrightarrow \hat{z} \sim p_\theta(z), \hat{x} \sim p_\theta(x|\hat{z})$ → generate sample



Better REC
Worse KL

- Need a **principle** (unlike maximum likelihood), or other **objective formulations** for AE to balance the above 2 goals.

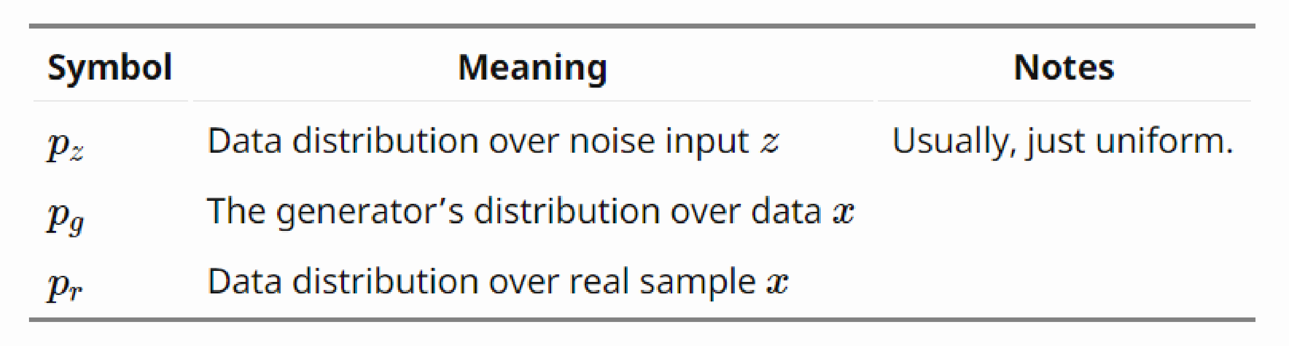


Worse REC
Better KL

Recap Deep Generative Models

❑ Generative Adversarial Networks (GAN)

Formulation:



Symbol	Meaning	Notes
p_z	Data distribution over noise input z	Usually, just uniform.
p_g	The generator's distribution over data x	
p_r	Data distribution over real sample x	

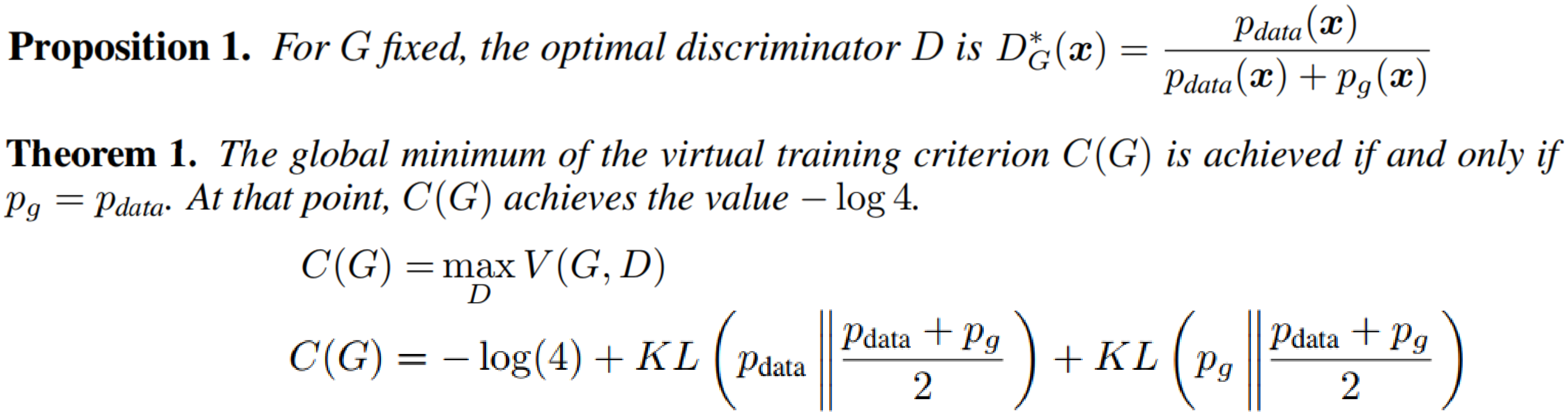
GANs is formulated as a minimax game b/w Generator G and Discriminator D:

$$\begin{aligned}\min_G \max_D L(D,G) &= \mathbb{E}_{x\sim p_r(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \\ &= \mathbb{E}_{x\sim p_r(x)}[\log D(x)] + \mathbb{E}_{x\sim p_g(x)}[\log(1-D(x)]\end{aligned}$$

Recap Deep Generative Models

❑ Generative Adversarial Networks (GAN)

<u>Optimality in GANs</u>:



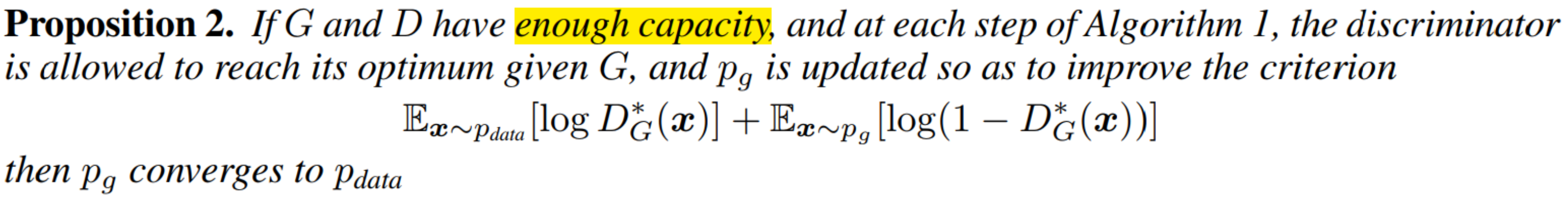
Proposition 1. *For G fixed, the optimal discriminator D is* $D^*_G(\boldsymbol{x}) = \dfrac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})}$

Theorem 1. *The global minimum of the virtual training criterion $C(G)$ is achieved if and only if $p_g = p_{data}$. At that point, $C(G)$ achieves the value $-\log 4$.*

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

$$C(G) = -\log(4) + KL\left(p_{\text{data}} \left\| \frac{p_{\text{data}} + p_g}{2}\right.\right) + KL\left(p_g \left\| \frac{p_{\text{data}} + p_g}{2}\right.\right)$$

Training GANs is equivalent to minimizing the <u>Jensen-Shannon divergence</u> b/w the data and generative distributions.



Proposition 2. *If G and D have enough capacity, and at each step of Algorithm 1, the discriminator is allowed to reach its optimum given G, and p_g is updated so as to improve the criterion*

$$\mathbb{E}_{\boldsymbol{x}\sim p_{data}}[\log D^*_G(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{x}\sim p_g}[\log(1 - D^*_G(\boldsymbol{x}))]$$

then p_g converges to p_{data}

Recap Deep Generative Models

❑ Generative Adversarial Networks (GAN)

Problem with training GANs:

- **non convergence**: unstable training, vanishing gradient
- mode colapsing

Why **non convergence**? The issue from f −divergence family (KL, Jensen-Shanon…)

When $\theta \neq 0$:

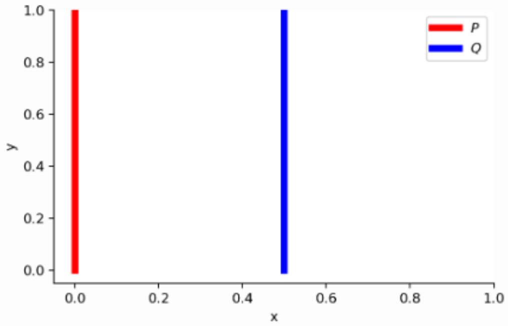
$$D_{KL}(P\|Q) = \sum_{x=0, y\sim U(0,1)} 1 \cdot \log \frac{1}{0} = +\infty$$

$$D_{KL}(Q\|P) = \sum_{x=\theta, y\sim U(0,1)} 1 \cdot \log \frac{1}{0} = +\infty$$

$$D_{JS}(P,Q) = \frac{1}{2}\left(\sum_{x=0, y\sim U(0,1)} 1 \cdot \log \frac{1}{1/2} + \sum_{x=0, y\sim U(0,1)} 1 \cdot \log \frac{1}{1/2}\right) = \log 2$$

$$\forall (x,y) \in P, x = 0 \text{ and } y \sim U(0,1)$$

$$\forall (x,y) \in Q, x = \theta, 0 \leq \theta \leq 1 \text{ and } y \sim U(0,1)$$

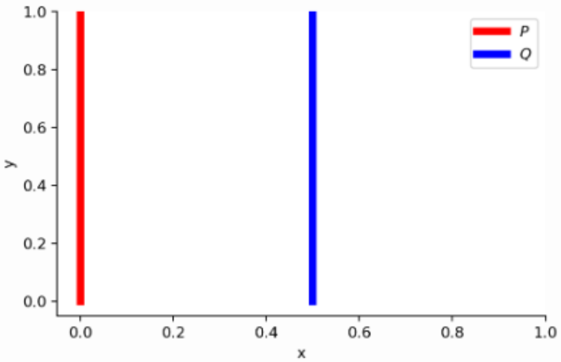


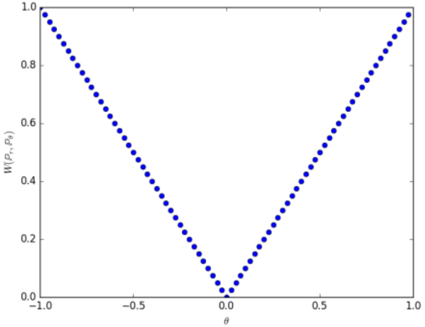
Recap Deep Generative Models

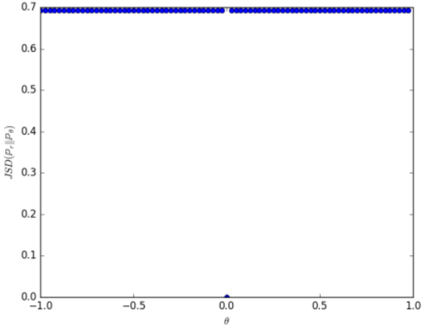
❑ Generative Adversarial Networks (GAN)

<u>Solutions from Optimal Transport</u>:

$$\forall (x,y) \in P, x = 0 \text{ and } y \sim U(0,1)$$
$$\forall (x,y) \in Q, x = \theta, 0 \leq \theta \leq 1 \text{ and } y \sim U(0,1)$$







$$D_{JS}(P,Q) = \frac{1}{2}\left(\sum_{x=0, y\sim U(0,1)} 1 \cdot \log \frac{1}{1/2} + \sum_{x=0, y\sim U(0,1)} 1 \cdot \log \frac{1}{1/2}\right) = \log 2$$

$$W(P,Q) = |\theta|$$

- All member of f −divergence has cons: can not be computed when two distributions are <u>disjoint support or continuous-discrete</u>, <u>not a distance</u>, <u>not very meaningful</u>

→ Optimal transport distances overcome these problems !

Outline

1. A brief review of Optimal Transport
    - Monge/Kantorovich formulation
    - Wasserstein distance
    - Sliced Wasserstein distance

2. Recap Deep Generative Models
    - Variational Autoencoders (VAE)
    - Generative Adversarial Networks (GAN)

3. Generative Modeling from Optimal Transport view
    - (Sliced) Wasserstein Generative Adversarial Networks (WGAN, SWGAN)
    - (Sliced) Wasserstein Autoencoders (WAE, SWAE)

4. References

Generative Modeling from Optimal Transport view

❑ Wasserstein GAN (WGAN)

- Let P_r, P_θ (P_g) be the data and model (generative) distribution respectively. WGAN minimizes the W_1 distance between P_r, P_θ via Kantorovich duality:

$$W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x \sim \mathbb{P}_r}[f(x)] - \mathbb{E}_{x \sim \mathbb{P}_\theta}[f(x)]$$

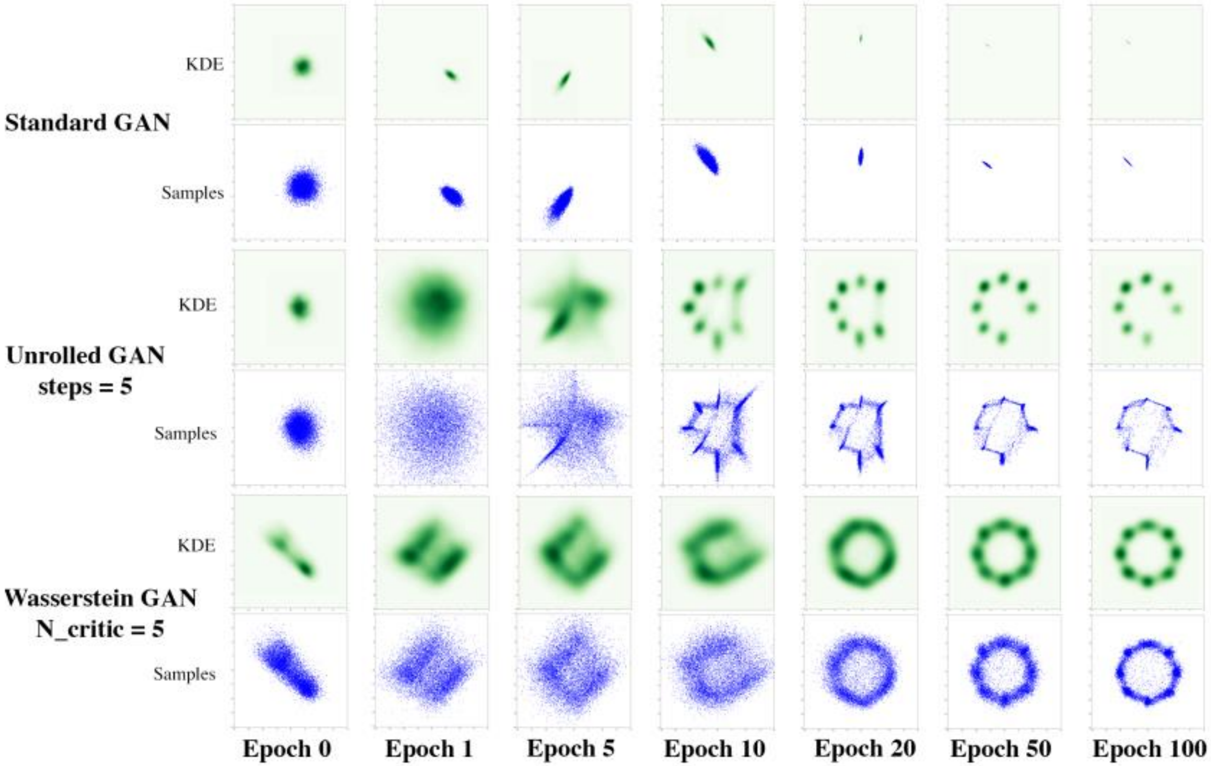
or K −Lipschitz equivalently:

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

- Relax Lipschitz constraint by parametrizing f with a neural net D and use:
  - Weight clipping: $w \leftarrow \text{clip}(w, -c, c)$
  - Gradient penalty: $\lambda \underset{\hat{\boldsymbol{x}} \sim \mathbb{P}_{\hat{\boldsymbol{x}}}}{\mathbb{E}} \left[(\|\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})\|_2 - 1)^2\right]$, where $\hat{x}$ sampled from $\tilde{x}$ (fake) and x (real) with ϵ uniformly sampled in [0,1]: $\hat{\boldsymbol{x}} \leftarrow \epsilon \boldsymbol{x} + (1 - \epsilon)\tilde{\boldsymbol{x}}$

Generative Modeling from Optimal Transport view

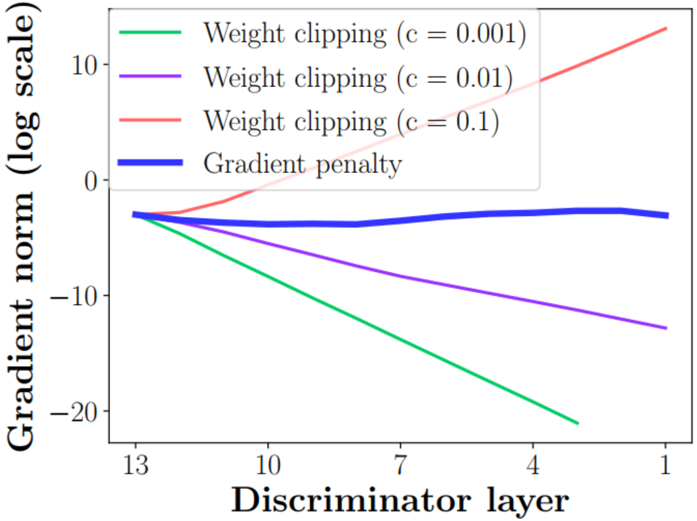
❑ **Wasserstein GAN (WGAN)**

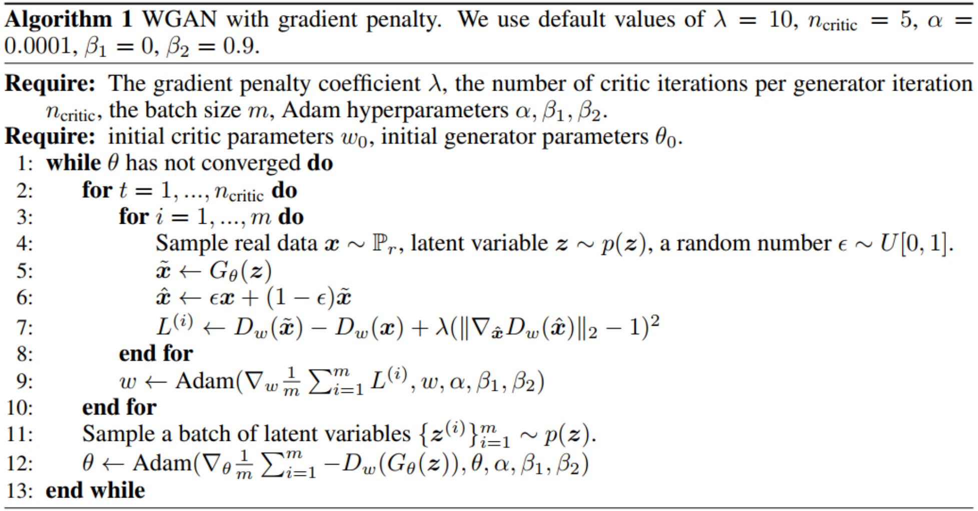


Generative Modeling from Optimal Transport view

❑ Wasserstein GAN (WGAN)

- **Weight clipping**: simple, effective in some cases, but slow convergence, unstable gradient (vanishing or exploding), similar to difference constraint: L2 clipping, weight norm, L2-L1 …





Algorithm 1 WGAN with gradient penalty. We use default values of $\lambda = 10$, $n_{\text{critic}} = 5$, $\alpha = 0.0001$, $\beta_1 = 0$, $\beta_2 = 0.9$.

Require: The gradient penalty coefficient λ, the number of critic iterations per generator iteration n_{critic}, the batch size m, Adam hyperparameters α, β_1, β_2.

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

```
while $\theta$ has not converged do
    for $t = 1, ..., n_{\text{critic}}$ do
        for $i = 1, ..., m$ do
            Sample real data $\boldsymbol{x} \sim \mathbb{P}_r$, latent variable $\boldsymbol{z} \sim p(\boldsymbol{z})$, a random number $\epsilon \sim U[0, 1]$.
            $\tilde{\boldsymbol{x}} \leftarrow G_\theta(\boldsymbol{z})$
            $\hat{\boldsymbol{x}} \leftarrow \epsilon \boldsymbol{x} + (1 - \epsilon)\tilde{\boldsymbol{x}}$
            $L^{(i)} \leftarrow D_w(\tilde{\boldsymbol{x}}) - D_w(\boldsymbol{x}) + \lambda(\|\nabla_{\hat{\boldsymbol{x}}} D_w(\hat{\boldsymbol{x}})\|_2 - 1)^2$
        end for
        $w \leftarrow \text{Adam}(\nabla_w \frac{1}{m} \sum_{i=1}^{m} L^{(i)}, w, \alpha, \beta_1, \beta_2)$
    end for
    Sample a batch of latent variables $\{\boldsymbol{z}^{(i)}\}_{i=1}^{m} \sim p(\boldsymbol{z})$.
    $\theta \leftarrow \text{Adam}(\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} -D_w(G_\theta(\boldsymbol{z})), \theta, \alpha, \beta_1, \beta_2)$
end while
```

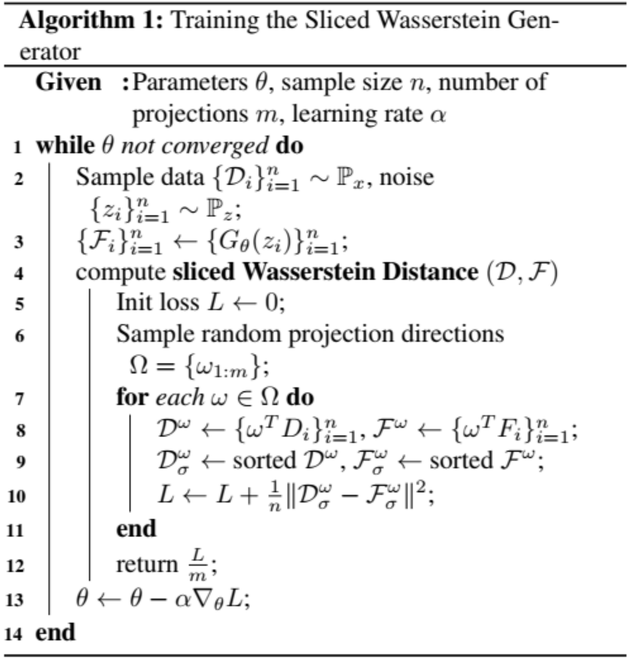
Generative Modeling from Optimal Transport view

❑ Sliced Wasserstein GAN (SWGAN)

- The correctness of the estimate in WGAN depends fundamentally on <u>how well the discriminator has been trained</u> → it seem to be difficult like the adversarial training in vanilla GAN.

- **SWGAN**:
  - only needs the generator, not need the critic / discriminator.
  - takes advantage of the **closed-form solution** of Wasserstein distance on 1-D.

  but:
  - equires large number of projections due to high dimensional space, $\approx \mathcal{O}(10^4)$



```
Algorithm 1: Training the Sliced Wasserstein Generator
Given :Parameters $\theta$, sample size $n$, number of
       projections $m$, learning rate $\alpha$
while $\theta$ not converged do
    Sample data $\{\mathcal{D}_i\}_{i=1}^n \sim \mathbb{P}_x$, noise
     $\{z_i\}_{i=1}^n \sim \mathbb{P}_z$;
    $\{\mathcal{F}_i\}_{i=1}^n \leftarrow \{G_\theta(z_i)\}_{i=1}^n$;
    compute sliced Wasserstein Distance $(\mathcal{D}, \mathcal{F})$
        Init loss $L \leftarrow 0$;
        Sample random projection directions
         $\Omega = \{\omega_{1:m}\}$;
        for each $\omega \in \Omega$ do
            $\mathcal{D}^\omega \leftarrow \{\omega^T D_i\}_{i=1}^n$, $\mathcal{F}^\omega \leftarrow \{\omega^T F_i\}_{i=1}^n$;
            $\mathcal{D}_\sigma^\omega \leftarrow$ sorted $\mathcal{D}^\omega$, $\mathcal{F}_\sigma^\omega \leftarrow$ sorted $\mathcal{F}^\omega$;
            $L \leftarrow L + \frac{1}{n}\|\mathcal{D}_\sigma^\omega - \mathcal{F}_\sigma^\omega\|^2$;
        end
        return $\frac{L}{m}$;
    $\theta \leftarrow \theta - \alpha \nabla_\theta L$;
end
```

Generative Modeling from Optimal Transport view

❑ Sliced Wasserstein GAN (SWGAN)

- **SWGAN**: **solutions for scaling to high dimensional**
  - a neural net based discriminator tries to **map the real and fake samples into a space** where it is easy to tell them apart
  - the two objectives, which are optimized independently (**not adversarial training**) of each other are:

$$\min_{\theta} \frac{1}{|\hat{\Omega}|} \sum_{\omega \in \hat{\Omega}} W_2^2(f_{\theta'}(\mathcal{D})^{\omega}, f_{\theta'}(\mathcal{F})^{\omega}(\theta)),$$

$$\min_{\theta'} \mathbb{E}[-\log(f'_{\theta'}(\mathcal{D}))] + \mathbb{E}[-\log(1 - f'_{\theta'}(\mathcal{F}))]$$

where θ is the generator weight, $f'_{\theta'}$ is the neural net (CNN) mapping data into subspace, $f_{\theta'}$ is the intermediate layer.

- Or using **max-Sliced Wasserstein** for GAN.

Generative Modeling from Optimal Transport view

- ❑ **Sliced Wasserstein GAN (SWGAN)**

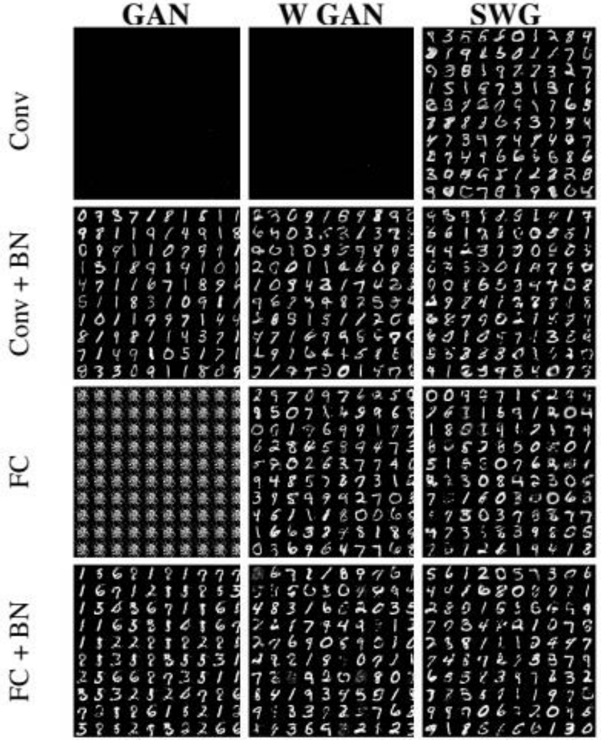


Figure 5. MNIST samples after 40k training iterations for different generator configurations. Batch size = 250, Learning rate = 0.0005, Adam optimizer

Generative Modeling from Optimal Transport view

❑ Wasserstein Autoencoder

- Focus on **latent variable models** P_G: $p_G(x) := \int_{\mathcal{Z}} p_G(x|z)p_z(z)dz, \quad \forall x \in \mathcal{X}$
  - use **non-random decoders** for simplicity (similar results for random decoders)
  - the optimal transport cost to estimate the distance between P_X and P_G is considered in **the primal form:**

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

- <u>**Reparametrization of the couplings**</u>**:**

Theorem 1. *For P_G as defined above with deterministic $P_G(X|Z)$ and any function $G\colon \mathcal{Z} \to \mathcal{X}$*

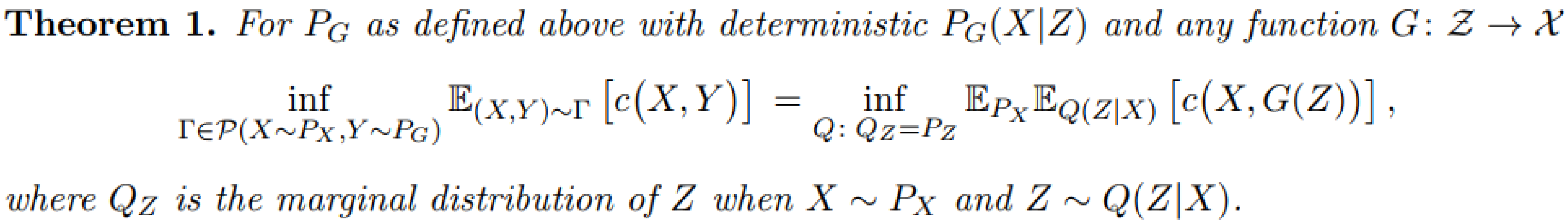
$$\inf_{\Gamma \in \mathcal{P}(X \sim P_X, Y \sim P_G)} \mathbb{E}_{(X,Y)\sim\Gamma}\big[c\big(X,Y\big)\big] = \inf_{Q\colon Q_Z = P_Z} \mathbb{E}_{P_X}\mathbb{E}_{Q(Z|X)}\big[c\big(X, G(Z)\big)\big],$$

where Q_Z is the marginal distribution of Z when $X \sim P_X$ and $Z \sim Q(Z|X)$.

Generative Modeling from Optimal Transport view

❑ Wasserstein Autoencoder

- **<u>Reparametrization of the couplings</u>:**

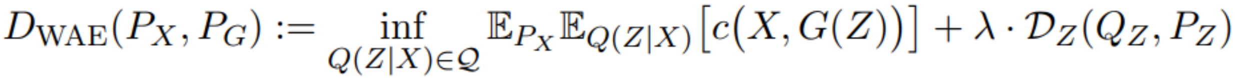


Theorem 1. *For P_G as defined above with deterministic $P_G(X|Z)$ and any function $G\colon \mathcal{Z} \to \mathcal{X}$*

$$\inf_{\Gamma \in \mathcal{P}(X \sim P_X, Y \sim P_G)} \mathbb{E}_{(X,Y)\sim\Gamma}\big[c(X,Y)\big] = \inf_{Q\colon Q_Z = P_Z} \mathbb{E}_{P_X}\mathbb{E}_{Q(Z|X)}\big[c(X,G(Z))\big],$$

where Q_Z is the marginal distribution of Z when $X \sim P_X$ and $Z \sim Q(Z|X)$.

- **Proof:** condition $Q_Z = P_Z$ associated to the constraints on the marginals of transport plan Γ.
- Relax the constraints on Q_Z by adding a **penalty** to the objective:



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

where $\mathcal{Q}$ is any nonparametric set of probabilistic encoders, $\mathcal{D}_Z$ is an arbitrary divergence between Q_Z and P_Z.

- use **deep neural networks** to parametrize both encoders Q and decoders G.

Generative Modeling from Optimal Transport view

❑ Wasserstein Autoencoder

- **Formulation:** use D_Z is GAN or MMD regularizers:

  - **WAE-GAN:**

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

    - P_Z, Q_Z are the true and fake distribution respectively.
    - low dimension, P_Z is simple, nice shape, easy to matching

  - **WAE-MMD:**

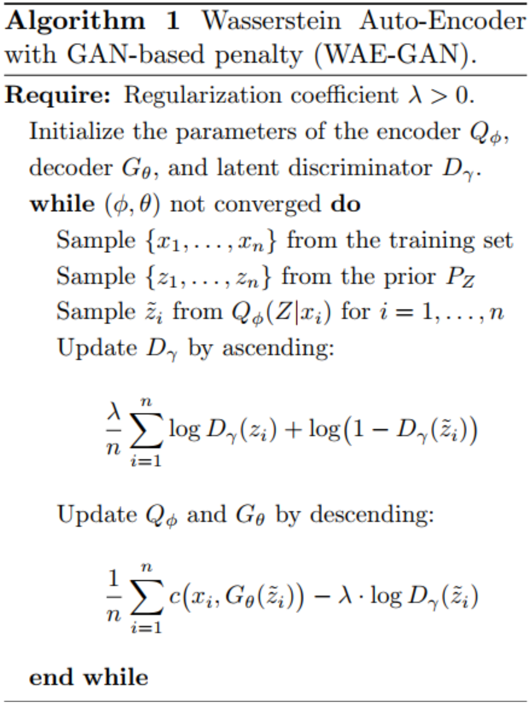
$$D_{WAE-GAN}(P_X, P_G) = \inf_{Q(Z|X)\in\mathcal{Q}} \mathbb{E}_{P_X}\mathbb{E}_{Q(Z|X)}[c(X, G(Z))] + \lambda D_{MMD}(Q_Z, P_Z)$$

    - performs well when matching high-dimensional standard normal distributions
    - not need to tune as training GAN

Generative Modeling from Optimal Transport view

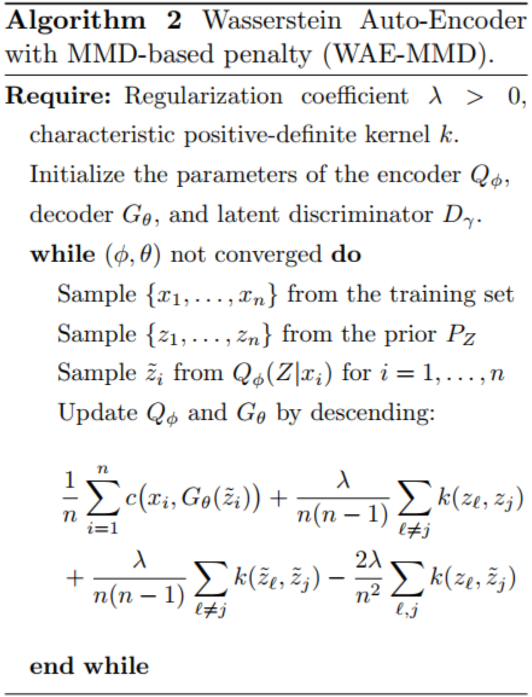
❑ Wasserstein Autoencoder

- **<u>Formulation</u>:** use D_Z is GAN or MMD regularizers:



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

```
Require: Regularization coefficient $\lambda > 0$.
  Initialize the parameters of the encoder $Q_\phi$,
  decoder $G_\theta$, and latent discriminator $D_\gamma$.
  while $(\phi, \theta)$ 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 $D_\gamma$ by ascending:
        $\frac{\lambda}{n} \sum_{i=1}^{n} \log D_\gamma(z_i) + \log\bigl(1 - D_\gamma(\tilde{z}_i)\bigr)$
    Update $Q_\phi$ and $G_\theta$ by descending:
        $\frac{1}{n} \sum_{i=1}^{n} c\bigl(x_i, G_\theta(\tilde{z}_i)\bigr) - \lambda \cdot \log D_\gamma(\tilde{z}_i)$
  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 $Q_\phi$,
  decoder $G_\theta$, and latent discriminator $D_\gamma$.
  while $(\phi, \theta)$ 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 $Q_\phi$ and $G_\theta$ by descending:
        $\frac{1}{n} \sum_{i=1}^{n} c\bigl(x_i, G_\theta(\tilde{z}_i)\bigr) + \frac{\lambda}{n(n-1)} \sum_{\ell \neq j} k(z_\ell, z_j)$
        $+ \frac{\lambda}{n(n-1)} \sum_{\ell \neq j} k(\tilde{z}_\ell, \tilde{z}_j) - \frac{2\lambda}{n^2} \sum_{\ell, j} k(z_\ell, \tilde{z}_j)$
  end while
```

Generative Modeling from Optimal Transport view

❑ Wasserstein Autoencoder

- **Properties**:
  - An explanation for why VAEs tend to generate **blurry** images

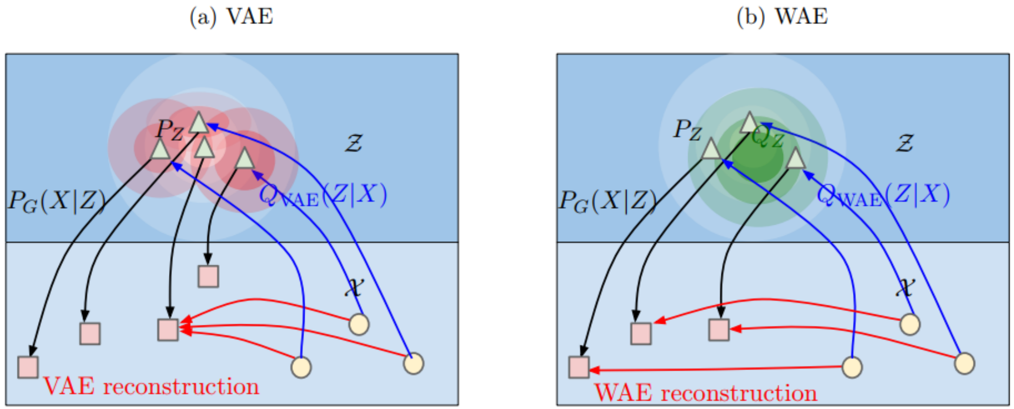
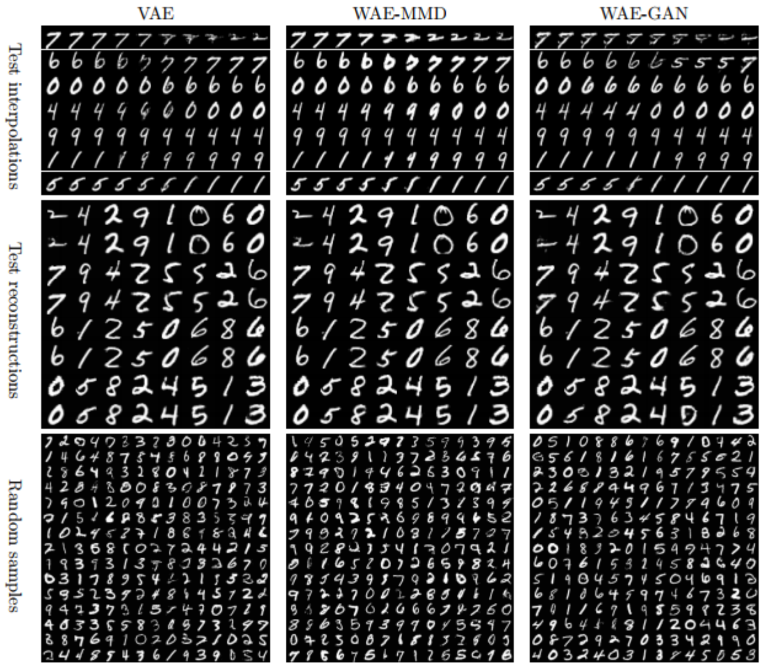


Figure 1: Both VAE and WAE minimize two terms: the reconstruction cost and the regularizer penalizing discrepancy between P_Z and distribution induced by the encoder Q. VAE forces $Q(Z|X = x)$ to match P_Z for all the different input examples x drawn from P_X. This is illustrated on picture (a), where every single red ball is forced to match P_Z depicted as the white shape. Red balls start intersecting, which leads to problems with reconstruction. In contrast, WAE forces the continuous mixture $Q_Z := \int Q(Z|X)dP_X$ to match P_Z, as depicted with the green ball in picture (b). As a result latent codes of different examples get a chance to stay far away from each other, promoting a better reconstruction.

Generative Modeling from Optimal Transport view

❑ Wasserstein Autoencoder

- **Properties**:
  - An explanation for why VAEs tend to generate **blurry** images



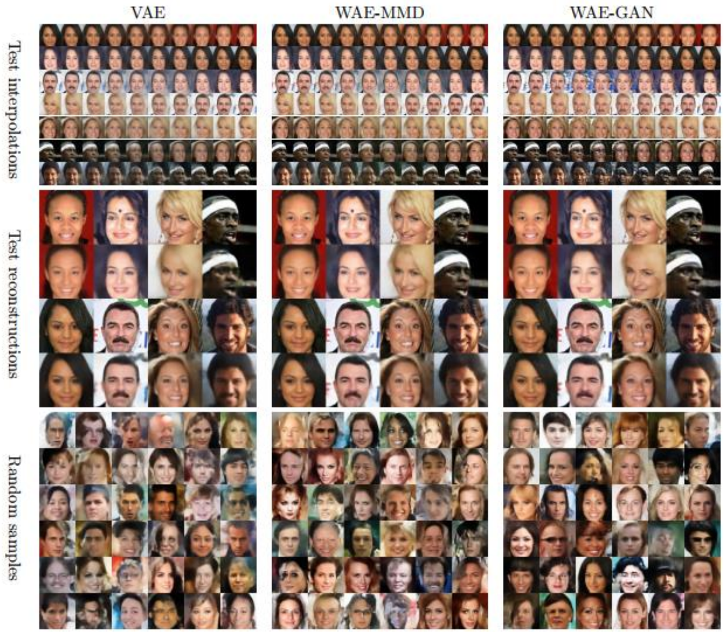


Figure 3: VAE (left column), WAE-MMD (middle column), and WAE-GAN (right column) trained on CelebA dataset. In "test reconstructions" odd rows correspond to the real test points.

Generative Modeling from Optimal Transport view

❑ Wasserstein Autoencoder

- **Properties**:
  - reconstruction term of WAE not come from Gaussian (majority) which needs to tune the variance.
  - when $c(x, y) = \|x - y\|_2^2$, WAE-GAN is equivalent to adversarial auto-encoders (AAE), but generalizes AAE in two ways: any cost $c(x, y)$ and discrepancy measure D_Z.
  - allows both probabilistic and deterministic encoder-decoder pairs of any kind.

Generative Modeling from Optimal Transport view

❑ Sliced Wasserstein Autoencoder

- avoids the need to perform **adversarial training** in the encoding space and is not restricted to closed-form distributions.

- takes advantage of the **closed-form solution** of Wasserstein distance on 1-D.
- fast, simple, effective with small number of projections (z is low dimension), $\approx \mathcal{O}(10)$

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

- can use **max/generalized version** of sliced distance as the regularization instead of SW.

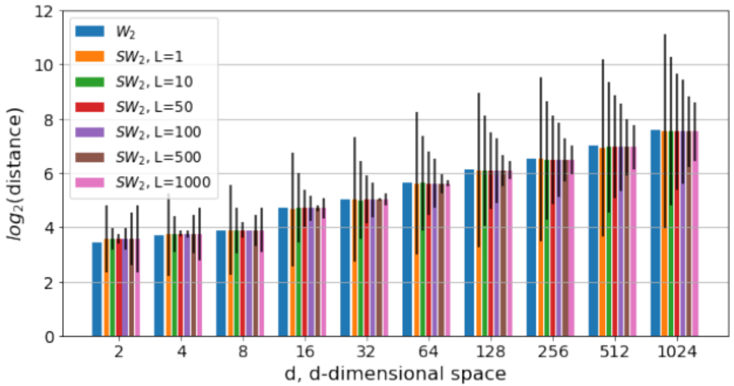
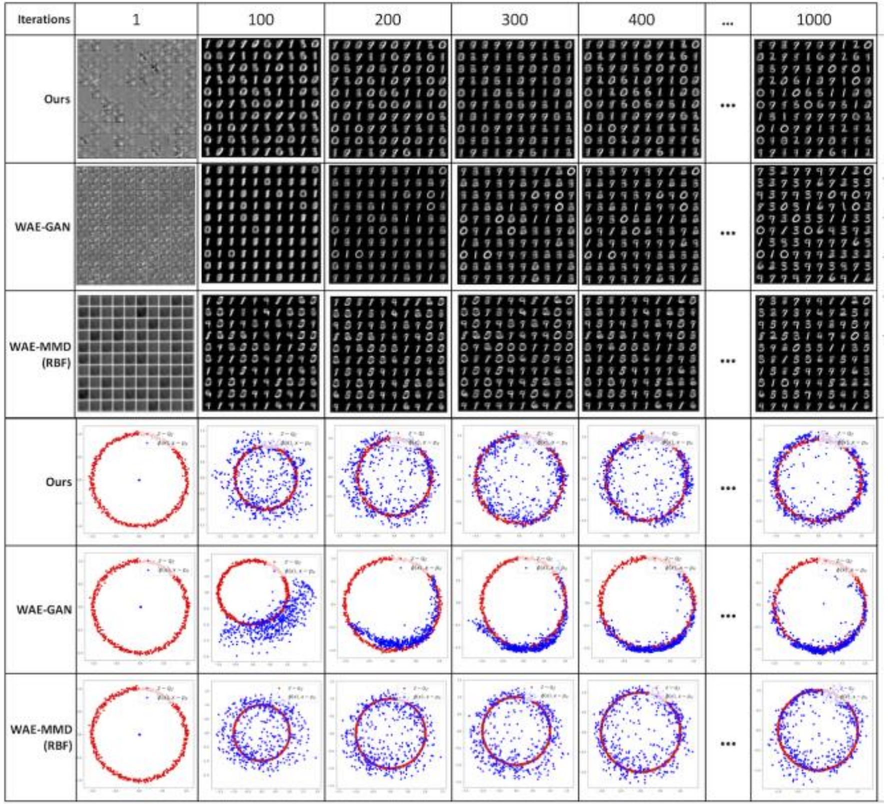
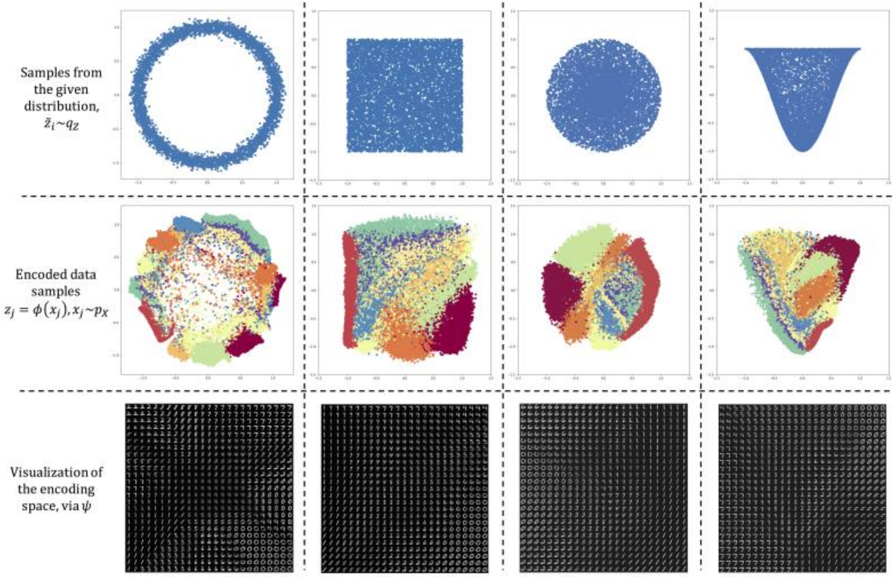


Figure 2: SW approximations (scaled by $1.22\sqrt{d}$) of the W-2 distance in different dimensions, $d \in \{2^n\}_{n=1}^{10}$, and different number of random slices, L.

Generative Modeling from Optimal Transport view

❑ Sliced Wasserstein Autoencoder





Generative Modeling from Optimal Transport view

❑ Further reading

- Recent advances of Optimal Transport facilitate applications in generative modeling: (sliced) Gromov-Wasserstein, Sinkhorn, Randkhorn …

References