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 $\nu \in \mathcal{P}(Y)$ and we call it a transport map if:

$$v(B) = \mu(T^{-1}(B)) \text{ or } v(B) = \mu(A)$$

for all $\nu$-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 $\nu \in \mathcal{P}(Y)$:

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

over measurable maps $T : X \to Y$ subject to $\nu = T_\# \mu$

- 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: $\nu(B) = \mu(T^{-1}(B))$
- In continuous case, the constraint require transport map is bijective and differentiable, it is equivalent to:

$$f(x) = g(T(x))|\det(\nabla T(x))| \quad \text{where } d\mu(x) = f(x)dx, \quad dv(y) = g(y)dy$$
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: \( \mu = \delta_{x_1} \), \( \nu = \frac{1}{2} \delta_{y_1} + \frac{1}{2} \delta_{y_2} \) then \( \nu(y_1) = \frac{1}{2} \) 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 \( \nu = \frac{1}{n} \sum_{j=1}^{N} \delta_{y_j} \)
2. The absolutely continuous case when \( d\mu(x) = f(x)dx \) and \( d\nu(y) = g(y)dy \)
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:

\[ \pi(A \times Y) = \mu(A) \text{ and } \pi(X \times B) = v(B) \text{ for all measurable sets } A \subseteq X, B \subseteq Y \]

• \( \pi \) is a joint distribution which has first marginal \( \mu \in \mathcal{P}(X) \) and second marginal \( v \in \mathcal{P}(Y) \)

• \( \pi \) is called transport plan and set of such transport plan \( \Pi(\mu, v) \)
A brief review of Optimal Transport

- Kantorovich Formulation

\[
\gamma = \begin{bmatrix}
\frac{1}{3} & 0 & 0 \\
0 & \frac{1}{3} & \frac{1}{3}
\end{bmatrix} x_1
\]

\[
\gamma = \begin{bmatrix}
\frac{1}{9} & 0 & \frac{2}{9} \\
\frac{2}{9} & 1 & \frac{2}{9}
\end{bmatrix} x_2
\]

\[
\gamma = \begin{bmatrix}
\frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\
\frac{2}{9} & \frac{2}{9} & \frac{2}{9}
\end{bmatrix} x_2
\]

\[
\sum_i \gamma_{i} = \begin{bmatrix}
\frac{1}{3} & \frac{1}{3} & \frac{1}{3}
\end{bmatrix}
\]

\[
\sum_j \gamma_{j} = \begin{bmatrix}
\frac{1}{3} \\
\frac{2}{3}
\end{bmatrix}
\]
A brief review of Optimal Transport

Kantorovich Formulation

Kantorovich’s Optimal Transport Problem:

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

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

Assume that there exists an 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 $\pi \in \Pi(x, y)$

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

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

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

Kantorovich Formulation

Kantorovich’s Optimal Transport Problem:

Kantorovich problem between two discrete measures \( \mu = \sum_{i=1}^{m} \alpha_i \delta_{x_i} \), \( \nu = \sum_{j=1}^{n} \beta_j \delta_{y_j} \) 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: \[ DP(\mu, v) = \sup_{(\varphi, \psi) \in \Phi_c} \int_X \varphi d\mu + \int_Y \psi dv \]

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

\[ \int_X |f| d\mu < \infty \]

\[ DP(\mu, v) \leq KP(\mu, v) \]
A brief review of Optimal Transport

- **Wasserstein Distance**

  **Definition:** Let $\mu$, $\nu$ are two probability measures in the set of probability measure with finite $p'$th moment defined on a given metric space $(\Omega, d)$, 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:

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

  When $p = 1$ Wasserstein Distance becomes Earth Mover Distance
A brief review of Optimal Transport

Wasserstein Distance

Kantorovich dual form of 1-Wasserstein:

\[
W_1(\mu, \nu) = \sup_{f,g} \int f \, d\mu(x) + \int g \, d\nu(y) \quad \text{subject to } f(x) + g(y) \leq ||x-y||
\]

\[
= \sup_{f} \int f \, d\mu(x) - \int f \, d\nu(y) \quad \text{where } f : \mathbb{R}^d \rightarrow \mathbb{R}, \text{ Lip}(f) \leq 1
\]
A brief review of Optimal Transport

- Wasserstein Distance

**Special case:** Wasserstein distance has **closed-form** solution in **one dimension**.

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

\[
W_p^p(\mu, \nu) = \frac{1}{n} \sum_{i=1}^{n} |x_i - y_i|^p
\]

- **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: \( f(x) = F_\nu^{-1}(F_\mu(x)) \)

\[
W_p(\mu, \nu) = \left( \int_X d^p(x, F_\nu^{-1}(F_\mu(x)))d\mu(x) \right)^{\frac{1}{p}} = \left( \int_0^1 d^p(F_\mu^{-1}(z), F_\nu^{-1}(z))dz \right)^{\frac{1}{p}}
\]
A brief review of Optimal Transport

- Sliced Wasserstein distance

Radon 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**:

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

- \( p_X(x) \) 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

Radon transform:

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

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

$$R_{p_X}(t, \theta) = \frac{1}{M} \sum_{m=1}^{M} \int_X \delta(x - x_m) \delta(t - \langle \theta, x \rangle) dx$$

$$= \frac{1}{M} \sum_{m=1}^{M} \delta(t - \langle \theta, x_m \rangle)$$
A brief review of Optimal Transport

❑ Sliced Wasserstein distance

Formulation:

Given two probability measures $\mu$, $\nu$ with the probability density $I_\mu$, $I_\nu$ respectively:

$$SW_p(\mu, \nu) = \left( \int_{S^{d-1}} W_p^p (RI_\mu(\cdot, \theta), RI_\nu(\cdot, \theta)) d\theta \right)^{1/p}$$

$$\approx \left( \frac{1}{L} \sum_{l=1}^{L} W_p^p (RI_\mu(\cdot, \theta_l), RI_\nu(\cdot, \theta_l)) \right)^{1/p}$$

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

- $SW_p^p(\mu, \nu) \leq \alpha_{d,p} W_p^p(\mu, \nu)$, with $\alpha_{d,p} = \frac{1}{d} \int_{S^{d-1}} ||\theta||_p d\theta \leq 1$

- The sensitivity and discriminativeness of Sliced Wasserstein distance depend on the number and the importance of projections $L$.  

17
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.

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

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 \(\theta\) 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**:

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

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

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

\[
\max - GSW_p(I_\mu, I_v) = \max_{\theta \in \Omega_\theta} W_p(\mathcal{G}I_\mu(\cdot, \theta), \mathcal{G}I_v(\cdot, \theta))
\]
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 \( \theta \):
  \[
p_{\theta}(x, z) = p_{\theta}(z)p_{\theta}(x|z)
\]

- **Goal**: maximize the marginal log-likelihood of the dataset:
  \[
  \log p_{\theta}(X) = \sum_{i=1}^{n} \log p_{\theta}(x_i)
  \]

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

- **Key idea**: Use variational (E-M) method \( \rightarrow \) maximize a **variational lower bound** instead:
  \[
  \log p_{\theta}(x) = \mathcal{L}(\theta, \phi; x) + \mathcal{KL}(q_{\phi}(z|x)\|p_{\theta}(z|x))
  \geq \mathcal{L}(\theta, \phi; x) = \mathbb{E}_{q_{\phi}(z|x)} \log p_{\theta}(x|z) - \mathcal{KL}(q_{\phi}(z|x)\|p_{\theta}(z))
  \]
Recap Deep Generative Models

- Variational Autoencoders (VAE)

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

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

Variational Autoencoders (VAE)

The Autoencoder perspective:

\[
\log p_\theta(x) \geq \left( E_{z \sim q_\phi(z)} \log p_\theta(x|z) \right) - KL(q_\phi(z|x) || p(z))
\]

- **Reconstruction loss**
- **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 (\( \theta \) is a neural net)
Recap Deep Generative Models

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

  - Variational objective of VAE has **two goals with a trade-off**: reconstruct and generate or equivalently inference and learning
    \[
    \hat{z} \sim q_\phi(z|x), \hat{x} \sim p_\theta(x|\hat{z}) \rightarrow \text{reconstruction}
    \]
    \[
    \hat{x} \sim p_\theta(x) \leftrightarrow \hat{z} \sim p_\theta(z), \hat{x} \sim p_\theta(x|\hat{z}) \rightarrow \text{generate sample}
    \]

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

Generative Adversarial Networks (GAN)

Formulation:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_z$</td>
<td>Data distribution over noise input $z$</td>
<td>Usually, just uniform.</td>
</tr>
<tr>
<td>$p_g$</td>
<td>The generator’s distribution over data $x$</td>
<td></td>
</tr>
<tr>
<td>$p_r$</td>
<td>Data distribution over real sample $x$</td>
<td></td>
</tr>
</tbody>
</table>

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

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

$$= \mathbb{E}_{x \sim p_r(x)}[\log D(x)] + \mathbb{E}_{x \sim p_g(x)}[\log(1 - D(x))]$$
Recap Deep Generative Models

Generative Adversarial Networks (GAN)

Optimality in GANs:

Proposition 1. For G fixed, the optimal discriminator \( D^*_G(x) \) is
\[
D^*_G(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)}
\]

Theorem 1. The global minimum of the virtual training criterion \( C(G) \) is achieved if and only if \( p_g = p_{\text{data}} \). 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 Jensen-Shannon divergence 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
\[
E_{x \sim p_{\text{data}}} [\log D^*_G(x)] + E_{x \sim p_g} [\log(1 - D^*_G(x))]
\]
then \( p_g \) converges to \( p_{\text{data}} \).
Recap Deep Generative Models

- Generative Adversarial Networks (GAN)

**Problem with training GANs:**

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

Why **non convergence**? The issue from $f$ — divergence family (KL, Jensen-Shannon...)

When $\theta \neq 0$:

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

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

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

Generative Adversarial Networks (GAN)

Solutions from Optimal Transport:

- All member of $f$ —divergence has cons: can not be computed when two distributions are disjoint support or continuous-discrete, not a distance, not very meaningful

$\rightarrow$ 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 $P_r, P_\theta (P_g)$ be the data and model (generative) distribution respectively. WGAN minimizes the $W_1$ distance between $P_r, P_\theta$ via Kantorovich duality:

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

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: $\lambda \mathbb{E}_{\tilde{x} \sim P_{\tilde{x}}}[(\|\nabla_{\tilde{x}} D(\tilde{x})\|_2 - 1)^2]$, where $\tilde{x}$ sampled from $\tilde{x}$ (fake) and $x$ (real) with $\epsilon$ uniformly sampled in $[0,1]$: $\hat{x} \leftarrow \epsilon x + (1 - \epsilon)\tilde{x}$
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 ...

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

\textbf{Require:} The gradient penalty coefficient \( \lambda \), the number of critic iterations per generator iteration \( n_{\text{critic}} \), the batch size \( m \), Adam hyperparameters \( \alpha, \beta_1, \beta_2 \).

\textbf{Require:} initial critic parameters \( w_0 \), initial generator parameters \( \theta_0 \).

1: while \( \theta \) has not converged do
2: \hspace{1em} for \( t = 1, \ldots, n_{\text{critic}} \) do
3: \hspace{2em} for \( i = 1, \ldots, m \) do
4: \hspace{3em} Sample real data \( x_i \sim \mathbb{P}_r \), latent variable \( z_i \sim p(z) \), a random number \( \epsilon \sim U[0, 1] \).
5: \hspace{3em} \mu_i \leftarrow G_\theta(z_i)
6: \hspace{3em} \hat{x}_i \leftarrow \epsilon x_i + (1 - \epsilon) \mu_i
7: \hspace{3em} L^{(i)} \leftarrow D_w(x_i) - D_w(\hat{x}_i) + \lambda(\|\nabla_\mu D_w(\hat{x}_i)\|_2 - 1)^2
8: \hspace{3em} end for
9: \hspace{1em} w \leftarrow \text{Adam}(\nabla_w \frac{1}{m} \sum_{i=1}^{m} L^{(i)}, w, \alpha, \beta_1, \beta_2)
10: end for
11: Sample a batch of latent variables \( \{z^{(i)}\}_{i=1}^{m} \sim p(z) \).
12: \theta \leftarrow \text{Adam}(\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} -D_w(G_\theta(z_i)), \theta, \alpha, \beta_1, \beta_2)
13: end while
```
Generative Modeling from Optimal Transport view

- Sliced Wasserstein GAN (SWGAN)
  - The correctness of the estimate in WGAN depends fundamentally on how well the discriminator has been trained → it seems 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:
      - requires large number of projections due to high dimensional space, $\approx \mathcal{O}(10^4)$

Algorithm 1: Training the Sliced Wasserstein Generator

```
while $\theta$ not converged do
    Sample data $\{D_i\}_{i=1}^n \sim \mathbb{P}_x$, noise $\{z_i\}_{i=1}^n \sim \mathbb{P}_z$;
    $\{F_i\}_{i=1}^n \leftarrow \{G_{\theta}(z_i)\}_{i=1}^n$;
    compute sliced Wasserstein Distance $(D, F)$
    Init loss $L \leftarrow 0$;
    Sample random projection directions $\Omega = \{\omega_{1:m}\}$;
    for each $\omega \in \Omega$ do
        $D_\omega \leftarrow \{\omega^T D_i\}_{i=1}^n$, $F_\omega \leftarrow \{\omega^T F_i\}_{i=1}^n$;
        $D_\sigma$ and $F_\sigma$ be sorted $D_\omega$ and $F_\omega$;
        $L \leftarrow L + \frac{1}{m} \|D_\sigma - F_\sigma\|^2$;
    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:

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

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

where \(\theta\) 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:

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

- Reparametrization of the couplings:

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

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

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

Reparametrization of the couplings:

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

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

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 $\Gamma$.
- Relax the constraints on $Q_Z$ by adding a **penalty** to the objective:

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

where $Q$ is any nonparametric set of probabilistic encoders, $D_Z$ 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**:
    
    $$D_{WAE-GAN}(P_X, P_G) = \inf_{Q(Z|X) \in Q} E_{P_X} E_{Q(Z|X)} [c(X, G(Z))] + \lambda D_{GAN}(Q_Z, P_Z)$$

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

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

    - performs well when matching high-dimensional standard normal distributions
    - not need to tune as training GAN
Generative Modeling from Optimal Transport view

- **Wasserstein Autoencoder**
  - **Formulation:** use $D_Z$ is GAN or MMD regularizers:

```latex
\begin{algorithm}
  \textbf{Algorithm 1} \textit{Wasserstein Auto-Encoder with GAN-based penalty (WAE-GAN).}
  \begin{algorithmic}
    \Require Regularization coefficient $\lambda > 0$.
    \State Initialize the parameters of the encoder $Q_{\phi}$, decoder $G_{\theta}$, and latent discriminator $D_\gamma$.
    \While {$\phi, \theta$ not converged}
      \State Sample $\{x_1, \ldots, x_n\}$ from the training set
      \State Sample $\{z_1, \ldots, z_n\}$ from the prior $P_Z$
      \State Sample $\tilde{z}_i$ from $Q_{\phi}(Z|x_i)$ for $i = 1, \ldots, n$
      \State Update $D_\gamma$ by ascending:
      \State \quad $\frac{\lambda}{n} \sum_{i=1}^{n} \log D_\gamma(z_i) + \log(1 - D_\gamma(\tilde{z}_i))$
      \State Update $Q_{\phi}$ and $G_{\theta}$ by descending:
      \State \quad $\frac{1}{n} \sum_{i=1}^{n} c(x_i, G_{\theta}(\tilde{z}_i)) - \lambda \cdot \log D_\gamma(\tilde{z}_i)$
    \EndWhile
  \end{algorithmic}
\end{algorithm}

\begin{algorithm}
  \textbf{Algorithm 2} \textit{Wasserstein Auto-Encoder with MMD-based penalty (WAE-MMD).}
  \begin{algorithmic}
    \Require Regularization coefficient $\lambda > 0$, characteristic positive-definite kernel $k$.
    \State Initialize the parameters of the encoder $Q_{\phi}$, decoder $G_{\theta}$, and latent discriminator $D_\gamma$.
    \While {$\phi, \theta$ not converged}
      \State Sample $\{x_1, \ldots, x_n\}$ from the training set
      \State Sample $\{z_1, \ldots, z_n\}$ from the prior $P_Z$
      \State Sample $\tilde{z}_i$ from $Q_{\phi}(Z|x_i)$ for $i = 1, \ldots, n$
      \State Update $Q_{\phi}$ and $G_{\theta}$ by descending:
      \State \quad $\frac{1}{n} \sum_{i=1}^{n} c(x_i, G_{\theta}(\tilde{z}_i)) + \frac{\lambda}{n(n-1)} \sum_{i \neq j} k(z_i, z_j)$
      \State \quad $+ \frac{\lambda}{n(n-1)} \sum_{i \neq j} k(\tilde{z}_i, \tilde{z}_j) - \frac{2\lambda}{n^2} \sum_{i,j} k(\tilde{z}_i, \tilde{z}_j)$
    \EndWhile
  \end{algorithmic}
\end{algorithm}
```
Generative Modeling from Optimal Transport view

- **Wasserstein Autoencoder**

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

![Diagram](image)

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 \( c(x, y) = \|x - y\|^2 \), 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), $\approx O(10)$

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

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

![Image showing SW approximations](image.png)

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, $L$. 

43
Generative Modeling from Optimal Transport view

- Sliced Wasserstein Autoencoder

<table>
<thead>
<tr>
<th>Generations</th>
<th>1</th>
<th>100</th>
<th>200</th>
<th>300</th>
<th>400</th>
<th>...</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td><img src="image1" alt="Image" /></td>
<td><img src="image2" alt="Image" /></td>
<td><img src="image3" alt="Image" /></td>
<td><img src="image4" alt="Image" /></td>
<td><img src="image5" alt="Image" /></td>
<td><img src="image6" alt="Image" /></td>
<td><img src="image7" alt="Image" /></td>
</tr>
<tr>
<td>WAE-GAN</td>
<td><img src="image8" alt="Image" /></td>
<td><img src="image9" alt="Image" /></td>
<td><img src="image10" alt="Image" /></td>
<td><img src="image11" alt="Image" /></td>
<td><img src="image12" alt="Image" /></td>
<td><img src="image13" alt="Image" /></td>
<td><img src="image14" alt="Image" /></td>
</tr>
<tr>
<td>WAE-MMD (BBF)</td>
<td><img src="image15" alt="Image" /></td>
<td><img src="image16" alt="Image" /></td>
<td><img src="image17" alt="Image" /></td>
<td><img src="image18" alt="Image" /></td>
<td><img src="image19" alt="Image" /></td>
<td><img src="image20" alt="Image" /></td>
<td><img src="image21" alt="Image" /></td>
</tr>
</tbody>
</table>

- Samples from the given distribution, $x_i \sim q_2$
- Encoded data samples $z_j = \phi(x_j), x_j \sim p_x$
- Visualization of the encoding space, via $\phi$
Generative Modeling from Optimal Transport view

Further reading

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