Amortized Projection Optimization for Sliced Wasserstein Generative Models

Khai Nguyen, Nhat Ho

Presenter: Yue Xiang

2022.6.30

中國人民大學
RENMIN UNIVERSITY OF CHINA

Outlines

Wasserstein Distances and Its Variants

Apply (Sliced)-Wasserstein Distances to Generative Models

Amortized Sliced Wasserstein and Amortized Models

Experiments

Wasserstein Distances and Its Variants

Apply (Sliced)-Wasserstein Distances to Generative Models

Amortized Sliced Wasserstein and Amortized Models

Experiments

Wasserstein$-p$ Distance

The Wasserstein- p distance between two probability measures $\mu \in \mathcal{P}_p\left(\mathbb{R}^d\right)$ and $\nu \in \mathcal{P}_p\left(\mathbb{R}^d\right)$:

$$\mathrm{W}_p(\mu, \nu) := \left(\inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|_p^p d\pi(x, y)\right)^{\frac{1}{p}}.$$

Wasserstein$-p$ Distance

The Wasserstein- p distance between two probability measures $\mu \in \mathcal{P}_p\left(\mathbb{R}^d\right)$ and $\nu \in \mathcal{P}_p\left(\mathbb{R}^d\right)$:

$$\mathrm{W}_p(\mu, \nu) := \left(\inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|_p^p d\pi(x, y) \right)^{\frac{1}{p}}.$$

- Computational complexity: $\mathcal{O}\left(m^3 \log m\right)$
- Curse of dimensionality: sample complexity: $\mathcal{O}\left(m^{-1/d}\right)$

m is the number of supports of two mini-batch measures.

Wasserstein$-p$ Distance ($d = 1$)

When $d = 1$, the Wasserstein distance has a closed form:

$$W_p(\mu, \nu) = \left(\int_0^1 \mid F_\mu^{-1}(z) - F_\nu^{-1}(z) |^p dz \right)^{1/p} \tag{1}$$

F_μ, F_ν: cumulative distribution function (CDF) of μ and ν.

- Computational complexity: $\mathcal{O}(m \log m)$
- No curse of dimensionality: $\mathcal{O}\left(m^{-1/2}\right)$

Sliced-Wasserstein Distance

$$
\begin{aligned}
\mathrm{SW}_p(\mu,\nu) &:= \left(\int_{\mathbb{S}^{d-1}} \mathrm{W}_p^p(\theta\sharp\mu,\theta\sharp\nu)d\theta\right)^{\frac{1}{p}} \\
&\approx \left(\frac{1}{L}\sum_{i=1}^{L} \mathrm{W}_p^p\left(\theta_i\sharp\mu,\theta_i\sharp\nu\right)\right)^{\frac{1}{p}}.
\end{aligned}
$$

- For each $\theta \in \mathbb{S}^{d-1}$, $\mathrm{W}_p^p(\theta\sharp\mu,\theta\sharp\nu)$ can be computed in linear time $\mathcal{O}(n\log n)$ (n is the number of supports of μ and ν).

Sliced-Wasserstein Distance

$$
\begin{aligned}
\mathrm{SW}_p(\mu, \nu) &:= \left(\int_{\mathbb{S}^{d-1}} \mathrm{W}_p^p(\theta \sharp \mu, \theta \sharp \nu) d\theta \right)^{\frac{1}{p}} \\
&\approx \left(\frac{1}{L} \sum_{i=1}^{L} \mathrm{W}_p^p\left(\theta_i \sharp \mu, \theta_i \sharp \nu\right) \right)^{\frac{1}{p}}.
\end{aligned}
$$

- For each $\theta \in \mathbb{S}^{d-1}$, $\mathrm{W}_p^p(\theta \sharp \mu, \theta \sharp \nu)$ can be computed in linear time $\mathcal{O}(n \log n)$ (n is the number of supports of μ and ν).
- $\theta_1, \ldots, \theta_L \sim \mathcal{U}\left(\mathbb{S}^{d-1}\right)$, L should be sufficiently large compared to the dimension d.

Max-Sliced Wasserstein Distance

$$\text{Max}-\text{SW}(\mu, \nu) := \max_{\theta \in \mathbb{S}^{d-1}} \text{W}_p(\theta \sharp \mu, \theta \sharp \nu).$$

- overcome the curse of dimensionality of the Wasserstein distance,
- overcome the issues of Monte Carlo samplings in sliced-Wasserstein distance.

Max-SW Distance: Projected Gradient Descent:

Algorithm 1 Max-SW

Input: Probability measures: μ, ν, learning rate η, max number of iterations T.
Initialize θ
while θ not converge or reach T **do**
  $\theta = \theta - \nabla_\theta \mathrm{W}_p(\theta\sharp\mu, \theta\sharp\nu)$
  $\theta = \frac{\theta}{||\theta||_2}$
end while
Return: θ

Wasserstein Distances and Its Variants

Apply (Sliced)-Wasserstein Distances to Generative Models

Amortized Sliced Wasserstein and Amortized Models

Experiments

Generative Models

$$\min_{\phi \in \Phi} \mathcal{D}(\mu, \nu),$$

where $\mathcal{D}(\cdot, \cdot)$ can be Wasserstein distance or SW distance or Max-SW distance.

Generative Models

$$\min_{\phi \in \Phi} \mathcal{D}(\mu, \nu),$$

where $\mathcal{D}(\cdot, \cdot)$ can be Wasserstein distance or SW distance or Max-SW distance.

- The number of training samples is often huge, e.g., one million.
- The dimension of data is also large,e.g., ten thousand.

Mini-batch Loss Based on Wasserstein Distances

$$\tilde{\mathcal{D}}(\mu, \nu) := \mathbb{E}_{X,Y \sim \mu^{\otimes m} \otimes \nu^{\otimes m}} \mathcal{D}\left(P_X, P_Y\right)$$

where $m \geq 1$ is the mini-batch size and $\mathcal{D}$ is a Wasserstein metric.

Mini-batch Max-sliced Wasserstein Distance

$$\text{m-MAX-SW}(\mu, \nu) = \mathbb{E}_{X,Y \sim \mu^{\otimes m} \otimes \nu^{\otimes m}} \left[\max_{\theta \in \mathbb{S}^{d-1}} \text{W}_p \left(\theta \sharp P_X, \theta \sharp P_Y \right) \right]$$

Mini-batch Max-sliced Wasserstein Distance

$$\text{m-MAX-SW}(\mu, \nu) = \mathbb{E}_{X,Y \sim \mu^{\otimes m} \otimes \nu^{\otimes m}} \left[\max_{\theta \in \mathbb{S}^{d-1}} \mathrm{W}_p \left(\theta \sharp P_X, \theta \sharp P_Y \right) \right]$$

Each pair of mini-batch contains its own optimization problem of finding the ”max” slice.

Train Generative Models with Mini-Batch Max-SW

Algorithm 2 Training generative models with mini-batch max-sliced Wasserstein

Input: Data probability measure μ, model learning rate η_1, slice learning rate η_2, model maximum number of iterations T_1, slice maximum number of iterations T_2, number of mini-batches k (is often set to 1).

Initialize ϕ, the model probability measure ν_ϕ

while ϕ not converge or reach T_1 **do**

  $\nabla_\phi = 0$

  Sample $(X_1, Y_{\phi,1}), \ldots, (X_k, Y_{\phi,k}) \sim \mu^{\otimes m} \otimes \nu_\phi^{\otimes m}$

  for $i = 1$ to k **do**

    while θ not converge or reach T_2 **do**

      $\theta = \theta - \nabla_\theta \mathrm{W}_p(\theta \sharp P_{X_i}, \theta \sharp P_{Y_{\phi,i}})$

      $\theta = \frac{\theta}{||\theta||_2}$

    end while

    $\nabla_\phi = \nabla_\phi + \frac{1}{k} \nabla_\phi \mathrm{W}_p(\theta \sharp P_{X_i}, \theta \sharp P_{Y_{\phi,i}})$

  end for

  $\phi = \phi - \nabla_\phi$

end while

Return: ϕ, ν_ϕ

Avoid Nested-Loop in Mini-Batch Max-SW?

Q: How can we avoid the nested-loop in mini-batch Max-SW due to several local optimization problems?

Avoid Nested-Loop in Mini-Batch Max-SW?

Q: How can we avoid the nested-loop in mini-batch Max-SW due to several local optimization problems?
A: Amortized optimization.

Avoid Nested-Loop in Mini-Batch Max-SW?

Q: How can we avoid the nested-loop in mini-batch Max-SW due to several local optimization problems?
A: Amortized optimization.

- Solve all optimization problems independently ✖

Avoid Nested-Loop in Mini-Batch Max-SW?

Q: How can we avoid the nested-loop in mini-batch Max-SW due to several local optimization problems?
A: Amortized optimization.

- Solve all optimization problems independently ✖
- Train an amortized model to predict informative slicing directions for all mini-batch measures✔

Amortized Model

For each context variable x in the context space $\mathcal{X}$, $\theta^\star(x)$ is the solution of the optimization problem $\theta^\star(x) = \arg\min_{\theta \in \Theta} \mathcal{L}(\theta, x)$, where Θ is the solution space.
A parametric function $f_\psi : \mathcal{X} \to \Theta$, where $\psi \in \Psi$, is called an amortized model if

$$f_\psi(x) \approx \theta^\star(x), \quad \forall x \in \mathcal{X}.$$

Train the Amortized Model

The amortized model is trained by the amortized optimization objective:

$$\min_{\psi \in \Psi} \mathbb{E}_{x \sim p(x)} \mathcal{L}\left(f_\psi(x), x\right),$$

where $p(x)$ is a probability measure on $\mathcal{X}$ which measures the "importance" of optimization problems.

Amortized Sliced Wasserstein Distance

Let $p \geq 1, m \geq 1$, and μ, ν are two probability measure in $\mathcal{P}\left(\mathbb{R}^d\right)$. Given an amortized model $f_\psi : \mathbb{R}^{dm} \times \mathbb{R}^{dm} \to \mathbb{S}^{d-1}$ where $\psi \in \Psi$, the amortized sliced Wasserstein between μ and ν is defined as:

$$\mathcal{A}-SW(\mu,\nu) := \max_{\psi \in \Psi} \mathbb{E}_{(X,Y)\sim\mu^{\otimes m}\otimes\nu^{\otimes m}}\left[W_p\left(f_\psi(X,Y)\sharp P_X, f_\psi(X,Y)\sharp P_Y\right)\right].$$

Amortized Sliced Wasserstein Distance

Let $p \geq 1, m \geq 1$, and μ, ν are two probability measure in $\mathcal{P}\left(\mathbb{R}^d\right)$. Given an amortized model $f_\psi : \mathbb{R}^{dm} \times \mathbb{R}^{dm} \to \mathbb{S}^{d-1}$ where $\psi \in \Psi$, the amortized sliced Wasserstein between μ and ν is defined as:

$$\mathcal{A} - SW(\mu, \nu) := \max_{\psi \in \Psi} \mathbb{E}_{(X,Y) \sim \mu^{\otimes m} \otimes \nu^{\otimes m}} \left[W_p \left(f_\psi(X, Y) \sharp P_X, f_\psi(X, Y) \sharp P_Y \right) \right].$$

$$\text{m-Max} - \text{SW}(\mu, \nu) = \mathbb{E}_{X,Y \sim \mu^{\otimes m} \otimes \nu^{\otimes m}} \left[\max_{\theta \in \mathbb{S}^{d-1}} \text{W}_p \left(\theta \sharp P_X, \theta \sharp P_Y \right) \right].$$

Amortized Sliced Wasserstein Distance

Proposition 1. The amortized sliced Wasserstein losses are positive and symmetric. However, they are not metrics since they do not satisfy the identity property, namely, $\mathcal{A}$-SW$(\mu, \nu) = 0 \Leftrightarrow \mu = \nu$.

Amortized Sliced Wasserstein Distance

Proposition 2. The amortized sliced Wasserstein losses are lower-bounds of the mini-batch maxsliced Wasserstein loss, namely, $\mathcal{A}$-SW$(\mu, \nu) \leq$ m-Max-SW(μ, ν) for all probability measures μ and ν on $\mathbb{R}^d$.

Linear Amortized Model

- Assumption: the optimal projecting direction lies on the subspace that is spanned by the basis $\{x_1, \ldots, x_m, y_1, \ldots, y_m, w_0\}$ where $X = (x_1, \ldots, x_m)$ and $Y = (y_1, \ldots, y_m)$.

Linear Amortized Model

- Assumption: the optimal projecting direction lies on the subspace that is spanned by the basis $\{x_1, \ldots, x_m, y_1, \ldots, y_m, w_0\}$ where $X = (x_1, \ldots, x_m)$ and $Y = (y_1, \ldots, y_m)$.
- Given $X, Y \in \mathbb{R}^{dm}$, and the one-one "reshape" mapping $T : \mathbb{R}^{dm} \to \mathbb{R}^{d \times m}$, the linear amortized model is defined as:

$$f_\psi(X, Y) := \frac{w_0 + T(X)w_1 + T(Y)w_2}{\|w_0 + T(X)w_1 + T(Y)w_2\|_2^2},$$

where $w_1, w_2 \in \mathbb{R}^m, w_0 \in \mathbb{R}^d$ and $\psi = (w_0, w_1, w_2)$.

Generalized Linear Amortized Model

- Assumption: the optimal projecting direction lies on the subspace that is spanned by the basis $\{x'_1, \ldots, x'_m, y'_1, \ldots, y'_m\}$ where $g_{\psi_1}(X) = (x'_1, \ldots, x'_m)$ and $g_{\psi_1}(Y) = (y'_1, \ldots, y'_m)$, *e.g.*, $g_{v_1}(X) = (W_2\sigma\left(W_1x_1\right) + b_0, \ldots, W_2\sigma\left(W_1x_m\right) + b_0)$, where $\sigma(\cdot)$ is the Sigmoid function.

Generalized Linear Amortized Model

- Assumption: the optimal projecting direction lies on the subspace that is spanned by the basis $\{x'_1, \ldots, x'_m, y'_1, \ldots, y'_m\}$ where $g_{\psi_1}(X) = (x'_1, \ldots, x'_m)$ and $g_{\psi_1}(Y) = (y'_1, \ldots, y'_m)$, *e.g.*, $g_{v_1}(X) = (W_2\sigma(W_1 x_1) + b_0, \ldots, W_2\sigma(W_1 x_m) + b_0)$, where $\sigma(\cdot)$ is the Sigmoid function.
- Given $X, Y \in \mathbb{R}^{dm}$, and the one-one "reshape" mapping $T : \mathbb{R}^{dm} \to \mathbb{R}^{d\times m}$, the generalized linear amortized model is defined as:

$$f_\psi(X, Y) := \frac{w_0 + T(g_{\psi_1}(X))\, w_1 + T(g_{\psi_1}(Y))\, w_2}{\|w_0 + T(g_{\psi_1}(X))\, w_1 + T(g_{\psi_1}(Y))\, w_2\|_2^2},$$

where $w_1, w_2 \in \mathbb{R}^m, w_0 \in \mathbb{R}^d, \psi_1 \in \Psi_1, g_{\psi_1} : (\mathbb{R}^d)^{\otimes m} \to (\mathbb{R}^d)^{\otimes m}$ and $\psi = (w_0, w_1, w_2, \psi_1)$.

Non-Linear Amortized Model

- Assumption: the optimal projecting direction lies on the **image** of the function $h_{\psi_2}(\cdot)$ that maps from the subspace spanned by $\{x_1, \ldots, x_m, y_1, \ldots, y_m\}$ where $X = (x_1, \ldots, x_m)$ and $Y = (y_1, \ldots, y_m)$.

Non-Linear Amortized Model

- Assumption: the optimal projecting direction lies on the **image** of the function $h_{\psi_2}(\cdot)$ that maps from the subspace spanned by $\{x_1, \ldots, x_m, y_1, \ldots, y_m\}$ where $X = (x_1, \ldots, x_m)$ and $Y = (y_1, \ldots, y_m)$.
- Given $X, Y \in \mathbb{R}^{dm}$, and the one-one mapping $T : \mathbb{R}^{dm} \to \mathbb{R}^{d \times m}$, the non-linear amortized model is defined as:

$$f_\psi(X, Y) := \frac{h_{\psi_2}\left(w_0 + T(X)w_1 + T(Y)w_2\right)}{\|h_{\psi_2}\left(w_0 + T(X)w_1 + T(Y)w_2\right)\|_2^2},$$

where $w_1, w_2 \in \mathbb{R}^m, w_0 \in \mathbb{R}^d, \psi_2 \in \Psi_2, h_{\psi_2} : \mathbb{R}^d \to \mathbb{R}^d$ and $\psi = (w_0, w_1, w_2, \psi_2)$.

Amortized Sliced Wasserstein Generative Models

Train a generative model ν_ϕ parametrized by $\phi \in \Phi$:

$$\min_{\phi\in\Phi} \max_{\psi\in\Psi} \mathbb{E}_{(X,Y_\phi)\sim\mu^{\otimes m}\otimes\nu_\phi^{\otimes m}} \mathrm{W}_p \left(f_\psi\left(X, Y_\phi\right)\sharp P_X, f_\psi\left(X, Y_\phi\right)\sharp P_{Y_\phi}\right)$$

$$:= \min_{\phi\in\Phi} \max_{\psi\in\Psi} \mathcal{L}\left(\mu, \nu_\phi, \psi\right).$$

Amortized Sliced Wasserstein Generative Models

Train a generative model ν_ϕ parametrized by $\phi \in \Phi$:

$$\min_{\phi\in\Phi} \max_{\psi\in\Psi} \mathbb{E}_{(X,Y_\phi)\sim\mu^{\otimes m}\otimes\nu_\phi^{\otimes m}} \mathrm{W}_p\left(f_\psi\left(X, Y_\phi\right)\sharp P_X, f_\psi\left(X, Y_\phi\right)\sharp P_{Y_\phi}\right)$$

$$:= \min_{\phi\in\Phi} \max_{\psi\in\Psi} \mathcal{L}\left(\mu, \nu_\phi, \psi\right).$$

- Minimax problem $\Rightarrow$ alternating stochastic gradient descent-ascent

Amortized Sliced Wasserstein Generative Models

Train a generative model ν_ϕ parametrized by $\phi \in \Phi$:

$$\min_{\phi\in\Phi} \max_{\psi\in\Psi} \mathbb{E}_{(X,Y_\phi)\sim\mu^{\otimes m}\otimes\nu_\phi^{\otimes m}} \mathrm{W}_p \left(f_\psi \left(X, Y_\phi\right) \sharp P_X, f_\psi \left(X, Y_\phi\right) \sharp P_{Y_\phi} \right)$$

$$:= \min_{\phi\in\Phi} \max_{\psi\in\Psi} \mathcal{L}\left(\mu, \nu_\phi, \psi\right).$$

- Minimax problem $\Rightarrow$ alternating stochastic gradient descent-ascent
- The stochastic gradients of ϕ and ψ can be estimated from mini-batches $(X_1, Y_{\phi,1}) \ldots (X_k, Y_{\phi,k}) \sim \mu^{\otimes m} \otimes \nu_\phi^{\otimes m}$:

$$\nabla_\phi \mathcal{L}\left(\mu, \nu_\phi, \psi\right) = \frac{1}{k} \nabla_\phi \mathrm{W}_p \left(f_\psi \left(X_i, Y_{\phi,i}\right) \sharp P_{X_i}, f_\psi \left(X_i, Y_{\phi,i}\right) \sharp P_{Y_{\phi,i}} \right),$$

$$\nabla_\psi \mathcal{L}\left(\mu, \nu_\phi, \psi\right) = \frac{1}{k} \nabla_\psi \mathrm{W}_p \left(f_\psi \left(X_i, Y_{\phi,i}\right) \sharp P_{X_i}, f_\psi \left(X_i, Y_{\phi,i}\right) \sharp P_{Y_{\phi,i}} \right).$$

Amortized Sliced Wasserstein Generative Models

Algorithm 3 Training generative models with amortized sliced Wasserstein

Input: Data probability measure μ, model learning rate η_1, amortized learning rate η_2, maximum number of iterations T, number of mini-batches k (is often set to 1).

Initialize ϕ, the model probability measure ν_ϕ.

Initialize ψ, the amortized model f_ψ.

while ϕ, ψ not converge or reach T **do**

$\nabla_\phi = 0; \nabla_\psi = 0$

Sample $(X_1, Y_{\phi,1}), \ldots, (X_k, Y_{\phi,k}) \sim \mu^{\otimes m} \otimes \nu_\phi^{\otimes m}$

for $i = 1$ to k **do**

$\nabla_\phi = \nabla_\phi + \frac{1}{k}\nabla_\phi \mathrm{W}_p(f_\psi(X_i, Y_{\phi,i})\sharp P_{X_i}, f_\psi(X_i, Y_{\phi,i})\sharp P_{Y_{\phi,i}})$

$\nabla_\psi = \nabla_\psi + \frac{1}{k}\nabla_\psi \mathrm{W}_p(f_\psi(X_i, Y_{\phi,i})\sharp P_{X_i}, f_\psi(X_i, Y_{\phi,i})\sharp P_{Y_{\phi,i}})$

end for

$\phi = \phi - \nabla_\phi$

$\psi = \psi + \nabla_\psi$

end while

Return: ϕ, ν_ϕ

Comparison

while ϕ not converge or reach T_1 **do**
  $\nabla_\phi = 0$
  Sample $(X_1, Y_{\phi,1}), \ldots, (X_k, Y_{\phi,k}) \sim \mu^{\otimes m} \otimes \nu_\phi^{\otimes m}$
  for $i = 1$ to k **do**
    while θ not converge or reach T_2 **do**
      $\theta = \theta - \nabla_\theta \mathrm{W}_p(\theta \sharp P_{X_i}, \theta \sharp P_{Y_{\phi,i}})$
      $\theta = \frac{\theta}{||\theta||_2}$
    end while
    $\nabla_\phi = \nabla_\phi + \frac{1}{k} \nabla_\phi \mathrm{W}_p(\theta \sharp P_{X_i}, \theta \sharp P_{Y_{\phi,i}})$
  end for
  $\phi = \phi - \nabla_\phi$
end while

(a) Mini-batch max-SW

while ϕ, ψ not converge or reach T **do**
  $\nabla_\phi = 0; \nabla_\psi = 0$
  Sample $(X_1, Y_{\phi,1}), \ldots, (X_k, Y_{\phi,k}) \sim \mu^{\otimes m} \otimes \nu_\phi^{\otimes m}$
  for $i = 1$ to k **do**
    $\nabla_\phi = \nabla_\phi + \frac{1}{k} \nabla_\phi \mathrm{W}_p(f_\psi(X_i, Y_{\phi,i}) \sharp P_{X_i}, f_\psi(X_i, Y_{\phi,i}) \sharp P_{Y_{\phi,i}})$
    $\nabla_\psi = \nabla_\psi + \frac{1}{k} \nabla_\psi \mathrm{W}_p(f_\psi(X_i, Y_{\phi,i}) \sharp P_{X_i}, f_\psi(X_i, Y_{\phi,i}) \sharp P_{Y_{\phi,i}})$
  end for
  $\phi = \phi - \nabla_\phi$
  $\psi = \psi + \nabla_\psi$
end while

(b) Amortized SW

Benchmarks and Evaluations

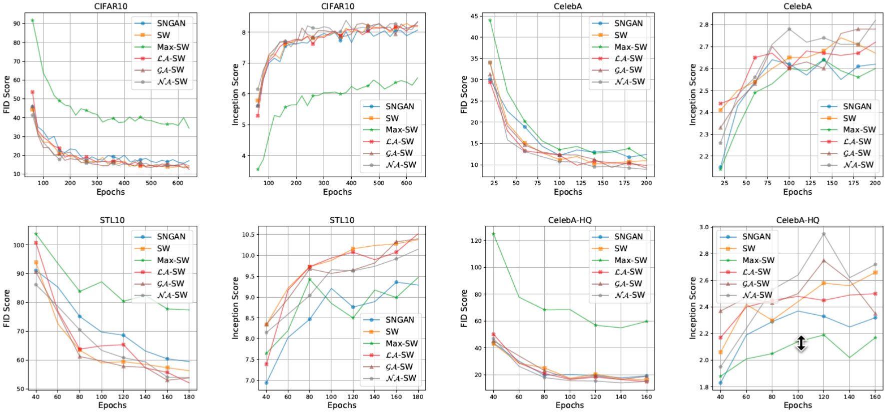
- Benchmarks: CIFAR10 $(32 \times 32$, STL10 (96×96), CelebA (64×64), and CelebAHQ (128×128)
- Evaluations:
  - quantitative: FID score, Inception score (IS)
  - qualitative: randomly generated images

FID and IS Scores

Table 1: Summary of FID and IS scores of methods on CIFAR10 (32x32), CelebA (64x64), STL10 (96x96), and CelebA-HQ (128x128). We observe that $\mathcal{A}$-SW losses provide the best results among all the training losses.

Method	CIFAR10 (32x32)		CelebA (64x64)		STL10 (96x96)		CelebA-HQ (128x128)	
	FID (↓)	IS (↑)	FID (↓)	IS (↑)	FID (↓)	IS (↑)	FID (↓)	IS (↑)
SNGAN (baseline)	17.09	8.07	12.41	2.61	59.48	9.29	19.25	2.32
SW	14.11	8.19	10.45	2.70	56.32	10.37	16.17	2.65
Max-SW	34.41	6.52	11.28	2.60	77.40	9.46	29.50	2.36
$\mathcal{LA}$-SW (ours)	**12.51**	8.22	9.82	2.72	**52.08**	**10.52**	**14.94**	2.50
$\mathcal{GA}$-SW (ours)	13.54	8.33	9.21	2.78	53.80	10.40	18.97	2.34
$\mathcal{NA}$-SW (ours)	14.44	**8.35**	**8.91**	**2.82**	53.90	10.14	15.17	**2.72**

Convergence: FID and IS Over Training Epochs



- FID lines of $\mathcal{A}$-SW are usually under the lines of other losses.
- IS lines of $\mathcal{A}$-SW are usually above the lines of other's.
- $\mathcal{A}$-SW usually help the generative models converge faster.

Computational Time and Memory

Table 2: Computational time and memory of methods (reported in the number of iterations per a second and megabytes (MB).

Method	CIFAR10 (32x32)		CelebA (64x64)		STL10 (96x96)		CelebA-HQ	
	Iters/s (↑)	Mem (↓)	Iters/s (↑)	Mem (↓)	Iters/s (↑)	Mem (↓)	Iters/s (↑)	Mem (↓)
SNGAN (baseline)	19.97	1740	6.31	6713	9.33	3866	10.41	3459
SW (L=1)	18.73	2078	6.17	8011	9.31	4597	10.25	4111
SW (L=100)	18.42	2093	6.15	8015	9.11	4609	10.17	4120
SW (L=1000)	14.96	2112	6.13	8047	9.03	4616	9.63	4143
SW (L=10000)	5.84	2421	4.21	8353	6.50	4780	5.17	4428
Max-SW (T_2=1)	18.61	2078	6.17	8011	9.23	4597	10.22	4111
Max-SW (T_2=10)	18.16	2078	6.15	8011	9.17	4597	10.16	4111
Max-SW (T_2=100)	13.47	2078	5.78	8011	8.32	4597	8.13	4111
$\mathcal{LA}$-SW (ours)	18.58	2086	6.17	8021	9.23	4600	10.19	4115
$\mathcal{GA}$-SW (ours)	17.27	4151	6.07	10083	9.08	5251	10.11	6163
$\mathcal{NA}$-SW (ours)	17.67	4134	6.13	10068	9.11	5249	10.15	6152

- Using sliced Wasserstein models gives better generative quality but it also costs more computational time and memory.
- $\mathcal{LA} - $ SW is the best option of sliced Wasserstein models.