Supplement to "Distributional Sliced-Wasserstein and Applications to Generative Modeling"

In this supplementary material, we collect several proofs and remaining materials that were deferred from the main paper. In Appendix A, we provide the proofs of the main results in the paper. In Appendix B, additional properties of distributional sliced-Wasserstein (DSW) distance are provided. In Appendix C, we discuss distributional generalized sliced-Wasserstein distance (DGSW) and its dual form and properties. We describe in detail the applications of DSW and DGSW to generative modelings in Appendix D. Furthermore, we provide additional experiments and experiment settings in Appendices E and G.

A PROOFS

In this appendix, we collect the proofs for all the results in the main text.

A.1 PROOF OF THEOREM 1

We first show that the distributional sliced-Wasserstein distance satisfies the triangle inequality property for any three probability measures μ_1, μ_2, and μ_3. In fact, from the definition of distributional sliced-Wasserstein distance for admissible $C > 0$, for any $\epsilon > 0$ we find that

$$
\begin{aligned}
\text{DSW}_p(\mu_1, \mu_2; C) &\overset{(i)}{\leq} \left\{\mathbb{E}_{\theta\sim\sigma_\epsilon^*} W_p^p\big(\mathcal{R}I_{\mu_1}(\cdot,\theta), \mathcal{R}I_{\mu_2}(\cdot,\theta)\big)\right\}^{\frac{1}{p}} + \epsilon \\
&\overset{(ii)}{\leq} \left\{\mathbb{E}_{\theta\sim\sigma_\epsilon^*}\big[W_p(\mathcal{R}I_{\mu_1}(\cdot,\theta), \mathcal{R}I_{\mu_3}(\cdot,\theta)) + W_p^p(\mathcal{R}I_{\mu_3}(\cdot,\theta), \mathcal{R}I_{\mu_2}(\cdot,\theta))\big]\right\}^{\frac{1}{p}} + \epsilon \\
&\overset{(iii)}{\leq} \left\{\mathbb{E}_{\theta\sim\sigma_\epsilon^*} W_p^p(\mathcal{R}I_{\mu_1}(\cdot,\theta), \mathcal{R}I_{\mu_3}(\cdot,\theta))\right\}^{\frac{1}{p}} \\
&\quad + \left\{\mathbb{E}_{\theta\sim\sigma_\epsilon^*} W_p^p\big(\mathcal{R}I_{\mu_3}(\cdot,\theta), \mathcal{R}I_{\mu_2}(\cdot,\theta)\big)\right\}^{\frac{1}{p}} + \epsilon \\
&\leq \sup_{\sigma\in\mathbb{M}_C} \left\{\mathbb{E}_{\theta\sim\sigma}\big[W_p^p\big(\mathcal{R}I_{\mu_1}(\cdot,\theta), \mathcal{R}I_{\mu_3}(\cdot,\theta)\big)\big]\right\}^{\frac{1}{p}} \\
&\quad + \sup_{\sigma\in\mathbb{M}_C} \left\{\mathbb{E}_{\theta\sim\sigma}\big[W_p^p\big(\mathcal{R}I_{\mu_3}(\cdot,\theta), \mathcal{R}I_{\mu_2}(\cdot,\theta)\big)\big]\right\}^{\frac{1}{p}} + \epsilon \\
&= \text{DSW}_p(\mu_1, \mu_3; C) + \text{DSW}_p(\mu_2, \mu_3; C) + \epsilon,
\end{aligned}
$$

where the existence of σ_ϵ^* in (i) is from the definition of supremum; inequality in (ii) is due to the triangle inequality with Wasserstein distance of order p; inequality in (iii) follows from the application of the Minkowski inequality. By letting $\epsilon \to 0$ in the above inequality, we obtain the conclusion with the triangle inequality of distributional sliced-Wasserstein distance.

The non-negativity and symmetry of distributional sliced-Wasserstein distance follow directly from the non-negativity and symmetry of Wasserstein distance. For the identity property, it is straightforward that if $\mu_1 \equiv \mu_2$ then $\text{DSW}_p(\mu_1, \mu_2) = 0$. On the other hand, if $\text{DSW}_p(\mu_1, \mu_2) = 0$, an application of Fourier transform as that in (Bonnotte, 2013) leads to $\mu_1 \equiv \mu_2$.

As a consequence, for any $p \geq 1$ and admissible $C > 0$, $\text{DSW}_p(\cdot, \cdot; C)$ is a well-defined metric in the space of Borel probability measures with finite p-th moment.

A.2 PROOF OF THEOREM 2

(a) From the definition of distributional sliced-Wasserstein distance, for any $p \geq 1$ and admissible $C > 0$ we find that

$$
\text{DSW}_p(\mu, \nu; C) \leq \sup_{\sigma\in\mathbb{M}} \left(\mathbb{E}_{\theta\sim\sigma}\left[W_p^p(\mathcal{R}I_\mu(\cdot,\theta), \mathcal{R}I_\nu(\cdot,\theta))\right]\right)^{\frac{1}{p}} = \text{maxSW}_p(\mu, \nu),
$$

where $\mathbb{M}$ is the space of all probability measures. The inequality is due to the fact that $\mathbb{M}_C \subseteq \mathbb{M}$ for all admissible $C > 0$. The second equality is true because $W_p(\mathcal{R}I_\mu(\cdot,\theta), \mathcal{R}I_\nu(\cdot,\theta)) \leq$

$\text{maxSW}_p(\mu,\nu)$ for all $\theta \in \mathbb{S}^{d-1}$, which leads to $\mathbb{E}_{\theta\sim\sigma}\big[W_p^p(\mathcal{R}I_\mu(\cdot,\theta),\mathcal{R}I_\nu(\cdot,\theta))\big] \leq \text{maxSW}_p^p(\mu,\nu)$. The inequality becomes equality when σ is the Dirac measure at θ^* that maximizes the value of $W_p(\mathcal{R}I_\mu(\cdot,\theta),\mathcal{R}I_\nu(\cdot,\theta))$.

Furthermore, we have

$$\begin{aligned} W_p^p(\mathcal{R}I_\mu(\cdot,\theta),\mathcal{R}I_\nu(\cdot,\theta)) &= \inf_{\pi\in\Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} |x^\top\theta - y^\top\theta|^p d\pi(x,y) \\ &\leq \inf_{\pi\in\Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} \|x-y\|^p d\pi(x,y) = W_p^p(\mu,\nu), \end{aligned}$$

where the last inequality is due to the fact that the length of the side of the right triangle $|(x-y)^\top\theta|$ is less than length of its hypotenuse $\|x-y\|$ for all $\theta \in \mathbb{S}^{d-1}$. Therefore, $\text{maxSW}_p(\mu,\nu) \leq W_p(\mu,\nu)$ for any $p \geq 1$.

Putting the above results together, we obtain the conclusion of part (a) of the theorem.

(b) Denote $\bar{\sigma} = \sum_{i=1}^d \frac{1}{d}\delta_{\theta_i}$ where $\theta_1 = \theta^*$, which maximizes the value of $W_p(\mathcal{R}I_\mu(\cdot,\theta),\mathcal{R}I_\nu(\cdot,\theta))$ and $\theta_1,\ldots,\theta_d$ form an orthonormal basis in $\mathbb{R}^d$. Simple algebra shows that

$$\mathbb{E}_{\theta,\theta'\sim\bar{\sigma}}\left[|\theta^\top\theta'|\right] = \sum_{1\leq i,j\leq d} \big(\frac{1}{d}\big)^2 |\theta_i^\top\theta_j| = \frac{1}{d}.$$

Since $C \geq \frac{1}{d}$, the above equation indicates that $\bar{\sigma} \in \mathbb{M}_C$. Therefore, we find that

$$\begin{aligned} \text{DSW}_p(\mu,\nu;C) \geq& \bigg(\mathbb{E}_{\theta\sim\bar{\sigma}}\Big[W_p^p(\mathcal{R}I_\mu(\cdot,\theta),\mathcal{R}I_\nu(\cdot,\theta))\Big]\bigg)^{\frac{1}{p}} \\ =& \bigg(\sum_{i=1}^d \frac{1}{d} W_p^p(\mathcal{R}I_\mu(\cdot,\theta_i),\mathcal{R}I_\nu(\cdot,\theta_i))\bigg)^{\frac{1}{p}} \\ \geq& \big(\frac{1}{d}\big)^{\frac{1}{p}} W_p(\mathcal{R}I_\mu(\cdot,\theta_1),\mathcal{R}I_\nu(\cdot,\theta_1)) = \big(\frac{1}{d}\big)^{\frac{1}{p}} \text{maxSW}_p(\mu,\nu). \end{aligned}$$

Moreover, for any $p \geq 1$, $\text{SW}_p(\mu,\nu) \leq \text{maxSW}_p(\mu,\nu)$. Collecting the previous results, we reach the conclusion of part (b).

Equivalence of $\text{DSW}_p(\cdot,\cdot;C)$ to other distances: Based on the result of Theorem 2.1 in (Bayraktar & Guo, 2019), maxSW_p, SW_p, and W_p are equivalent distances for any $p \geq 1$. In particular, for any sequence $(\mu_n)_{n\geq 1} \in \mathcal{P}_p(\mathbb{R}^d)$ and $\mu \in \mathcal{P}_p(\mathbb{R}^d)$, the following holds

$$\lim_{n\to\infty} \text{maxSW}_p(\mu_n,\mu) = 0 \iff \lim_{n\to\infty} \text{SW}_p(\mu_n,\mu) = 0 \iff \lim_{n\to\infty} W_p(\mu_n,\mu) = 0. \tag{3}$$

Now, if we have $\lim_{n\to\infty} \text{maxSW}_p(\mu_n,\mu) = 0$ for $p \geq 1$, the result of part (a) shows that $\lim_{n\to\infty} \text{DSW}_p(\mu_n,\mu;C) = 0$. On the other hand, when $\lim_{n\to\infty} \text{DSW}_p(\mu_n,\mu;C) = 0$, as long as $C \geq \frac{1}{d}$ and $p \geq 1$, the result of part (b) leads to $\lim_{n\to\infty} \text{maxSW}_p(\mu_n,\mu) = 0$. As a consequence, when $C \geq \frac{1}{d}$ and $p \geq 1$ we have

$$\lim_{n\to\infty} \text{DSW}_p(\mu_n,\mu;C) = 0 \iff \lim_{n\to\infty} \text{maxSW}_p(\mu_n,\mu) = 0. \tag{4}$$

Combining the results in equations (3) and (4), we reach the conclusion that when $C \geq \frac{1}{d}$ and $p \geq 1$, $\text{DSW}_p(\cdot,\cdot;C)$, maxSW_p, SW_p, and W_p are equivalent distances.

B Additional Studies with Distributional Sliced-Wasserstein Distance

In this appendix, we provide further studies with distributional sliced-Wasserstein distance.

B.1 DISCUSSION OF THE CONSTRAINT IN DSW

We first compute $\mathbb{E}_{\theta,\theta'\sim\sigma^{d-1}}\left[|\theta^\top\theta'|\right]$ where σ^{d-1} is the uniform distribution on the unit sphere $\mathbb{S}^{d-1}$.

Theorem 3. *For uniform measure σ^{d-1} on the unit sphere $\mathbb{S}^{d-1}$, we have*

$$\int_{\theta,\theta'\sim\sigma^{d-1}} |\theta^\top\theta'| d\sigma^{d-1}(\theta) d\sigma^{d-1}(\theta') = \frac{\Gamma(\frac{d}{2})}{\pi^{\frac{1}{2}}\Gamma(\frac{d+1}{2})},$$

where $\Gamma(.)$ is the Gamma function.

Remark. *The result of Theorem 3 indicates that as long as $C \geq \frac{\Gamma(\frac{d}{2})}{\pi^{\frac{1}{2}}\Gamma(\frac{d+1}{2})}$, we have $\sigma^{d-1} \in \mathbb{M}_C$. Furthermore, by Gautschi's inequality for the Gamma function, we find that*

$$\frac{1}{\pi^{\frac{1}{2}}(\frac{d+1}{2})^{\frac{1}{2}}} < \frac{\Gamma(\frac{d}{2})}{\pi^{\frac{1}{2}}\Gamma(\frac{d+1}{2})} < \frac{1}{\pi^{\frac{1}{2}}(\frac{d-1}{2})^{\frac{1}{2}}}$$

For $d \geq 3$, we have $2d^2/(d+1) > \pi$. Hence, we obtain that

$$\frac{1}{\pi^{\frac{1}{2}}(\frac{d+1}{2})^{\frac{1}{2}}} > \frac{1}{d}.$$

Given the above bound, when $C \geq \frac{\Gamma(\frac{d}{2})}{\pi^{\frac{1}{2}}\Gamma(\frac{d+1}{2})}$, the set $\mathbb{M}_C$ contains both σ^{d-1} and $\bar{\sigma} = \sum_{i=1}^d \frac{1}{d}\delta_{\theta_i}$ where $\theta_1, \ldots, \theta_d$ form any orthonormal basis in $\mathbb{R}^d$. Furthermore, for $d = 2$, we have

$$\frac{\Gamma(1)}{\pi^{\frac{1}{2}}\Gamma(\frac{2+1}{2})} = \frac{2}{\pi} > \frac{1}{2}.$$

Therefore, when $d = 2$ and $C \geq \frac{\Gamma(\frac{d}{2})}{\pi^{\frac{1}{2}}\Gamma(\frac{d+1}{2})}$, the set $\mathbb{M}_C$ also contains $\bar{\sigma} = \sum_{i=1}^d \frac{1}{d}\delta_{\theta_i}$.

Proof. Since σ^{d-1} is the uniform measure on the unit sphere $\mathbb{S}^{d-1}$, the integral

$$\int_{\theta\sim\sigma^{d-1}} |\theta^\top\theta'| d\sigma^{d-1}(\theta)$$

is the same for all fixed θ'. Hence for any fixed $\theta^* \in \mathbb{S}^{d-1}$, we obtain

$$I = \int_{\theta,\theta'\sim\sigma^{d-1}} |\theta^\top\theta'| d\sigma^{d-1}(\theta) d\sigma^{d-1}(\theta') = \int_{\theta\sim\sigma^{d-1}} |\theta^\top\theta^*| d\sigma^{d-1}(\theta).$$

Without loss of generality, we choose $\theta^* = (1, 0, \ldots, 0)$, I is equal to

$$\int_{\theta\sim\sigma^{d-1}} |\theta^{(1)}| d\sigma^{d-1}(\theta),$$

where $\theta = (\theta^{(1)}, \ldots, \theta^{(d)})$. For any measurable subset S of $\mathbb{S}^{d-1}$, let $A(S)$ be the area of S on the surface of $\mathbb{S}^{d-1}$ and $A(\mathbb{S}^{d-1})$ be the area of the surface of $\mathbb{S}^{d-1}$ which is equal to

$$A(\mathbb{S}^{d-1}) = \frac{d\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)}.$$

Now, we have

$$\int_{\theta\sim\sigma^{d-1}} |\theta^{(1)}| d\sigma^{d-1}(\theta) = \frac{1}{A(\mathbb{S}^{d-1})} \int_{\theta\in\mathbb{S}^{d-1}} |\theta^{(1)}| dA(\mathbb{S}^{d-1}(\theta)).$$

Let H_1 be the hyperplane formed by $\theta^{(2)}, \ldots, \theta^{(d)}$ and H_θ be the hyperplane tangent to the sphere $\mathbb{S}^{d-1}$ at θ. Then θ is the normal vector to H_θ and $\theta^* = (1, 0, \ldots, 0)$ is orthogonal to H_1. Let α be the angle between θ and θ^*. Then

$$dA(\mathbb{S}^{d-1}(\theta)) \cos(H_1, H_\theta) = d\theta^{(2)} \ldots d\theta^{(d)}$$

$$dA(\mathbb{S}^{d-1}(\theta)) = \frac{1}{\cos(\theta, \theta^*)} d\theta^{(2)} \ldots d\theta^{(d)} = \frac{1}{|\theta^{(1)}|} d\theta^{(2)} \ldots d\theta^{(d)}.$$

Return to the integral, we find that

$$\begin{aligned} I = \int_{\theta\sim\sigma^{d-1}} |\theta^{(1)}| d\sigma^{d-1}(\theta) &= \frac{1}{A(\sigma^{d-1})} \int_{\sum_{i=1}^d (\theta^i)^2=1} |\theta^{(1)}| \frac{1}{|\theta^{(1)}|} d\theta^{(2)} \dots d\theta^{(d)} \\ &= \frac{2}{A(\sigma^{d-1})} \int_{\theta^{(1)}>0, \sum_{i=2}^d (\theta^{(i)})^2 \leq 1} d\theta^{(2)} \dots d\theta^{(d)} \\ &= \frac{2}{A(\mathbb{S}^{d-1})} V(\mathbb{B}^{d-1}) \\ &= \frac{2\Gamma(\frac{d}{2})}{2\pi^{\frac{d}{2}}} \times \frac{\pi^{\frac{d-1}{2}}}{\Gamma(\frac{d-1}{2}+1)} \\ &= \frac{\Gamma(\frac{d}{2})}{\pi^{\frac{1}{2}}\Gamma(\frac{d+1}{2})}, \end{aligned}$$

where $\mathbb{B}^{d-1}$ is the unit ball in the $d-1$ dimensional space and $V(\mathbb{B}^{d-1})$ is its corresponding volume. As a consequence, we obtain the conclusion of the theorem. □

B.2 Approximation of dual value of DSW

Now, we give a detailed form of the objective function $\text{DS}(f_\phi)$ in the dual form of DSW in equation (2). In particular, simple calculation shows that

$$\begin{aligned} \nabla_\phi \text{DS}(f_\phi) = \frac{1}{p} \Big\{ \mathbb{E}_{\theta\sim\sigma^{d-1}} \big[W_p^p(\mathcal{R}I_\mu(., f_\phi(\theta)), \mathcal{R}I_\nu(., f_\phi(\theta))\big] \Big\}^{\frac{1}{p}-1} \\ \times \mathbb{E}_{\theta\sim\sigma^{d-1}} \big[\nabla_\phi W_p^p(\mathcal{R}I_\mu(., f_\phi(\theta)), \mathcal{R}I_\nu(., f_\phi(\theta))\big] - \lambda_C \mathbb{E}_{\theta,\theta'\sim\sigma^{d-1}} \big[\nabla_\phi |f_\phi(\theta)^\top f_\phi(\theta')| \big]. \end{aligned} \tag{5}$$

Since the outer expectations in equation (5) are intractable to compute, we employ the standard Monte Carlo scheme to approximate these expectations. Therefore, we obtain the following approximation:

$$\begin{aligned} \nabla_\phi \text{DS}(f_\phi) \approx \frac{1}{p} \Big\{ \frac{1}{n} \sum_{i=1}^n \big[W_p^p(\mathcal{R}I_\mu(., f_\phi(\theta_i)), \mathcal{R}I_\nu(., f_\phi(\theta_i))\big] \Big\}^{\frac{1}{p}-1} \\ \times \left\{ \frac{1}{n} \sum_{i=1}^n \Big[\nabla_\phi W_p^p(\mathcal{R}I_\mu(., f_\phi(\theta_i)), \mathcal{R}I_\nu(., f_\phi(\theta_i))\Big] \right\} - \frac{\lambda_C}{n(n-1)} \sum_{1\leq i\neq j\leq n} \nabla_\phi \big| (f_\phi(\theta_i))^\top f_\phi(\theta_j) \big|, \end{aligned}$$

where $\theta_1, \dots, \theta_n$ are i.i.d. samples from the unit sphere $\mathbb{S}^{d-1}$.

Denote ϕ^* as the fixed point of the stochastic gradient ascent algorithm. Then, we can use f_{ϕ^*} as the local maxima of the optimization problem (2). By using Monte Carlo method to approximate the expectation in equation (2), we obtain the following approximation:

$$\begin{aligned} \text{DSW}_p^*(\mu,\nu;C) \approx \Big\{ \frac{1}{n} \sum_{i=1}^n \big[W_p^p\big(\mathcal{R}I_\mu(\cdot, f_{\phi^*}(\theta_i)), \mathcal{R}I_\nu(\cdot, f_{\phi^*}(\theta_i))\big)\big] \Big\}^{1/p} \\ - \frac{\lambda_C}{n(n-1)} \sum_{1\leq i\neq j\leq n} \big| (f_{\phi^*}(\theta_i))^\top f_{\phi^*}(\theta_j) \big| + \lambda_C C. \end{aligned}$$

B.3 Statistical guarantee of DSW

In this appendix, we provide the statistical guarantee of DSW.

Theorem 4. *Given probability measure P supported on a compact subset $\Theta \subset \mathbb{R}^d$. Assume that $X_1, \dots, X_n$ are i.i.d. data from P. Denote $P_n = \frac{1}{n}\sum_{i=1}^n \delta_{X_i}$ the empirical measure of the data points $X_1, \dots, X_n$. Then, for any admissible regularizing constant $C > 0$ and for any $p \geq 1$, we obtain that*

$$\mathbb{E}\Big[\textit{DSW}_p(P_n, P; C) \Big] \leq c\sqrt{\frac{d\log n}{n}},$$

where $c > 0$ is some universal constant.

Remark. *The result of Theorem 4 demonstrates that DSW has similar statistical guarantees as other sliced distances and does not suffer from the curse of dimensionality. Therefore, it is an appealing distance for applications in generative modeling.*

Proof. The proof of Theorem 4 is a direct application of Theorem 2 and statistical guarantee of max-sliced Wassertein distance. Here, we provide the proof for the completeness. In particular, based on the result of Theorem 2, we obtain that

$$\mathbb{E}\Big[\text{DSW}_p(P_n, P; C)\Big] \leq \mathbb{E}\Big[\text{maxSW}_p(P_n, P)\Big].$$

Therefore, it is sufficient to demonstrate that $\mathbb{E}\Big[\text{maxSW}_p(P_n, P)\Big] \leq c\sqrt{\frac{d\log n}{n}}$ for some universal constant $c > 0$. In order to simplify the presentation, we denote a few notation. First, we define $\mathcal{H}$ the set of half-spaces $H_{\theta,x} = \{y \in \mathbb{R}^d : \langle y, \theta\rangle \leq x\}$ for any $\theta \in \mathbb{S}^{d-1}$ and $x \in \mathbb{R}$. Then, it has been shown that $\mathcal{H}$ has at most $d+1$ Vapnik–Chervonenkis (VC) dimension (Wainwright, 2019). The VC inequality implies that

$$\sup_{H\in\mathcal{H}} |P_n(H) - P(H)| \leq \sqrt{\frac{32}{n}\big[(d+1)\log(n+1) + \log(8/\delta)\big]} =: c_{n,\delta}$$

with probability at least $1-\delta$, for any $\delta \in (0,1)$. On the other hand, we have

$$\sup_{H\in\mathcal{H}} |P_n(H) - P(H)| = \sup_{x\in\mathbb{R},\theta\in\mathbb{S}^{d-1}} |F_{n,\theta}(x) - F_\theta(x)|,$$

where $F_{n,\theta}$ and F_θ are respectively the cumulative distribution functions (CDF) of $\mathcal{R}I_{P_n}(\cdot,\theta)$ and $\mathcal{R}I_P(\cdot,\theta)$. Given the above equation and the closed-form of Wasserstein distance in one dimension, we find that

$$\begin{aligned}
\text{maxSW}_p^p(P_n, P) &= \max_{\theta\in\mathbb{S}^{d-1}} \int_0^1 |F_{n,\theta}^{-1}(u) - F_\theta^{-1}(u)|^p du \\
&= \max_{\theta\in\mathbb{S}^{d-1}} \int_{\mathbb{R}} |F_{n,\theta}(x) - F_\theta(x)|^p dx \\
&\leq \text{diam}(\Theta) \sup_{x\in\mathbb{R},\theta\in\mathbb{S}^{d-1}} |F_{n,\theta}(x) - F_\theta(x)|^p \leq \text{diam}(\Theta) c_{n,\delta}^p.
\end{aligned}$$

By using the above inequality, we obtain that $\mathbb{E}\Big[\text{maxSW}_p(P_n, P)\Big] \leq c\sqrt{\frac{d\log n}{n}}$ for some universal constant $c > 0$. As a consequence, we reach the conclusion of Theorem 4. □

C AN EXTENSION TO DISTRIBUTIONAL GENERALIZED SLICED-WASSERSTEIN DISTANCE

We now consider an extension of DSW to non-linear projections via generalized Radon transform. The constant $C > 0$ is *generalized admissible* if the set $\overline{\mathbb{M}}_C$ of probability measures σ on the compact set of feasible parameters Ω_θ satisfying $\mathbb{E}_{\theta,\theta'\sim\sigma}\left[|\cos(\theta,\theta')|\right] \leq C$ is not empty.

Definition 3. *Given two probability measures μ and ν on $\mathbb{R}^d$ with finite p-th moments where $p \geq 1$ and a generalized admissible regularizing constant $C > 0$. The* distributional generalized sliced-Wasserstein distance *(DGSW) of order p between μ and ν is defined as follows:*

$$\textit{DGSW}_p(\mu,\nu;C) := \sup_{\sigma\in\overline{\mathbb{M}}_C} \Big\{\mathbb{E}_{\theta\sim\sigma} W_p^p(\mathcal{G}I_\mu(\cdot,\theta), \mathcal{G}I_\nu(\cdot,\theta))\Big\}^{1/p},$$

where $\mathcal{G}$ is generalized Radon transform defined in Section 2.2.

The DGSW distance uses the advantage of non-linear projections to capture more complex structures of the target probability measures. We show that as long as the generalized Radon transform is injective, DGSW is a proper metric in the probability space.

Theorem 5. *For any $p \geq 1$ and generalized admissible $C > 0$, as long as the generalized Radon transform is injective, the distributional generalized sliced-Wasserstein is a well-defined metric in the space of Borel probability measures with finite p-th moment.*

The proof of Theorem 5 simply follows the proof argument of Theorem 1 under the injectivity of GRT; thus, it is omitted. In order to compute DGSW, we also utilize the dual form of DGSW as that of DSW distance.

Dual form of distributional generalized sliced-Wasserstein distance: Similar to the distributional sliced-Wasserstein distance, we use the dual form of distributional generalized sliced-Wasserstein distance to approximate the value of distributional generalized sliced-Wasserstein distance. Recall that, for any $\theta, \theta' \in \mathbb{R}^d$, $\cos(\theta, \theta') = \frac{\theta^\top \theta'}{\|\theta\| \|\theta'\|}$.

Definition 4. *For any $p \geq 1$ and generalized admissible $C > 0$, there exists a non-negative constant λ_C depending on C such that the dual form of DGSW distance takes the following form*

$$
\begin{aligned}
DGSW_p^*(\mu, \nu; C) := -\sup_{\lambda \geq 0} \inf_{\sigma \in \mathbb{M}} \Bigg\{ &-\left(\mathbb{E}_{\theta \sim \sigma} \left[W_p^p(\mathcal{G}I_\mu(\cdot, \theta), \mathcal{G}I_\nu(\cdot, \theta)) \right] \right)^{1/p} \\
&+ \lambda \left(\mathbb{E}_{\theta, \theta' \sim \sigma} \left[\frac{|\theta^\top \theta'|}{\|\theta\| \, \|\theta'\|} \right] - C \right) \Bigg\} \\
= \sup_{\sigma \in \bar{\mathbb{M}}} \Bigg\{ &\left(\mathbb{E}_{\theta \sim \sigma} \left[W_p^p(\mathcal{G}I_\mu(\cdot, \theta), \mathcal{G}I_\nu(\cdot, \theta)) \right] \right)^{1/p} - \lambda_C \mathbb{E}_{\theta, \theta' \sim \sigma} \left[\frac{|\theta^\top \theta'|}{\|\theta\| \, \|\theta'\|} \right] \Bigg\} \\
&+ \lambda_C C,
\end{aligned}
$$

where $\bar{\mathbb{M}}$ denotes the space of all probability measures on the compact set of feasible parameter Ω_θ.

From the duality theory, we obtain that $\text{DGSW}_p(\mu, \nu; C) \geq \text{DGSW}_p^*(\mu, \nu; C)$ for any $p \geq 1$ and admissible $C > 0$. Similar to DSW distance, the dual form of DGSW provides an efficient way to approximate the DGSW distance. We show that when the compact set of feasible parameter $\Omega_\theta = \mathbb{S}^{d-1}$, similar reparametrization trick like that of the dual form of DSW distance can be applied to the dual form of DGSW distance. In particular, when $\Omega_\theta = \mathbb{S}^{d-1}$, we obtain the equivalent dual form of DGSW as follows:

$$
\begin{aligned}
\text{DGSW}_p^*(\mu, \nu; C) = \sup_{f \in \mathcal{F}} \Bigg\{ &\left(\mathbb{E}_{\theta \sim \sigma^{d-1}} \left[W_p^p\big(\mathcal{G}I_\mu(\cdot, f(\theta)), \mathcal{G}I_\nu(\cdot, f(\theta))\big) \right] \right)^{1/p} \\
&- \lambda_C \mathbb{E}_{\theta, \theta' \sim \sigma^{d-1}} \left[\left| f(\theta)^\top f(\theta') \right| \right] \Bigg\} + \lambda_C C,
\end{aligned} \tag{6}
$$

where $\mathcal{F}$ is a class of Borel measurable functions from $\mathbb{S}^{d-1}$ to $\mathbb{S}^{d-1}$ and $\lambda_C > 0$ is some positive constant given in Definition 4. Then, in order to find an optimal f, we can parameterize f as f_ϕ, which we can think as (deep) neural network. From here, with similar argument as that of equation (5), we can approximate the gradient of the objective function in equation (6) with respect to ϕ and then use stochastic gradient ascent algorithm to update ϕ. Finally, we can use the fixed point of the algorithm to approximate the dual value of DGSW in equation (6).

D Applications to Generative Modeling

The DSW and DGSW distances can potentially be applied in settings where there is a benefit of employing an optimal-transport type of distance in a computationally efficient manner. In this section, we discuss two general settings where the DSW and DGSW distances can be immediately applied. The first setting is a standard generative modeling task using the minimum expected distance estimator framework (Bernton et al., 2019) where a generative model is fitted to a data distribution by minimizing an appropriate divergence. The second setting is a joint contrastive inference task where both a generative model and inference model are learned jointly, again by minimizing some divergence in the joint space of observed variable and latent variable. In each setting, we apply the DSW to these tasks as well as its generalized version, the DGSW.

D.1 MINIMUM EXPECTED DISTRIBUTIONAL SLICED-WASSERSTEIN ESTIMATOR

Minimum expected distance estimators (Bernton et al., 2019) are widely used recently due to its efficiency in learning implicit generative models. Popular estimators include those based on OT distances (Arjovsky et al., 2017; Genevay et al., 2018; Tolstikhin et al., 2018) due to their smooth and differentiable objectives especially when the supports of the data and the generative distributions are not the same. In sliced-Wasserstein cases, SW and Max-SW have been employed with rigorous theoretical analyses in various works (Bayraktar & Guo, 2019; Deshpande et al., 2019; 2018; Nadjahi et al., 2019). They enjoy the benefits of the Wasserstein distance in one dimension and obtain fast speed in training the model. In this paper, we introduce a new novel estimator by replacing SW and Max-SW by our new DSW distance, which we refer to as *minimum expected distributional sliced-Wasserstein estimator*. The new estimator is defined as follows:

$$\hat{\theta}_n = \arg\min_{\theta\in\Theta} \mathbb{E}[\text{DSW}_p(\hat{\mu}_n, \hat{\mu}_{\theta,m})|X_{1:n}], \tag{7}$$

where Φ is the parameter space, $\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^n \delta_{X_i}$ is the empirical measure, and $\hat{\mu}_{\theta,m} = \frac{1}{m}\sum_{i=1}^m \delta_{Y_i}$ denotes the empirical distribution that is obtained by sampling i.i.d samples from μ_θ. In practice, μ_θ is created by pushing a simple distribution ϵ (such as the standard Gaussian) through a neural net, parameterized by θ, i.e., $\mu_\theta = T_\theta \sharp \epsilon$.

D.2 DISTRIBUTIONAL SLICED-WASSERSTEIN JOINT CONTRASTIVE INFERENCE

Learning both a generator and an inference model, i.e., an encoder, is a central task in latent-variable modeling. A general framework for performing this task is called joint contrastive inference (Dumoulin et al., 2016). Let $p_\theta(z,x) = p(z)p_\theta(x|z)$ be a generative model, $q_\phi(z|x)$ be an amortized inference model and define the data-induced aggregated joint inference model as $\hat{q}_\phi(z,x) = p_{data}(x)q_\phi(z|x)$. The joint contrastive inference framework then minimizes some divergence between the two structured joint distributions $p_\theta(z,x)$ and $\hat{q}_\phi(z,x)$. This can be seen as a generalized version of amortized inference. There are some well-known examples of this kind of inference such as the Variational Autoencoder (Kingma & Welling, 2013), Adversarially Learned Inference (Dumoulin et al., 2016), and Wasserstein Variational Inference (Ambrogioni et al., 2018). By using the DSW distance, we obtain a new joint contrastive inference method which inherits the benefits of optimal transport family of distances, yet remains scalable and computationally efficient. In particular, we learn both a generator and an inference model by solving:

$$(\theta_m, \phi_m) = \arg\min_{\theta\in\Theta,\phi\in\Phi} \mathbb{E}_{\hat{q}_\phi(z,x),p_\theta(z,x)}[\text{DSW}_p(\hat{q}_{\phi,m}(x,z), \hat{p}_{\theta,m}(z,x))], \tag{8}$$

where Θ, Φ are the parameter spaces, $\hat{q}_{\phi,m}(z,x)$ and $\hat{p}_{\theta,m}(z,x)$ are empirical distributions that sampled i.i.d data from $\hat{q}_\phi(z,x)$ and $p_\theta(z,x)$ respectively.

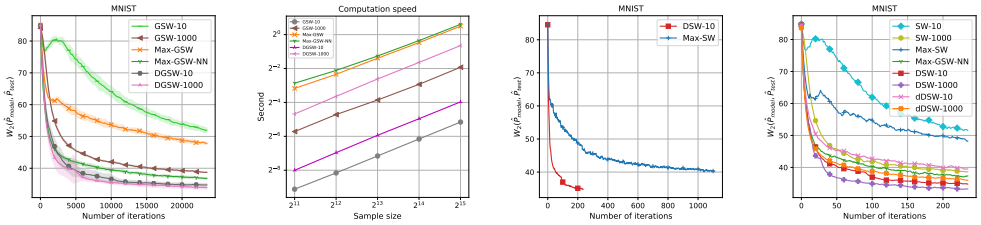
E ADDITIONAL EXPERIMENTS

In this appendix, we provide additional experimental results to yield more understandings about the minimum expected distance framework, which uses the new proposed distances. The appendix is divided into three parts, namely Appendices E.1, E.2 and E.3. Appendix E.1 is devoted to showing the performances of DGSW (see Appendix C for its definition) versus the generalized versions of other sliced distances on various factors which could affect the effectiveness of those methods. We also compare DSW to the recent augumented sliced Wasserstein method (ASW) (Chen et al., 2020). Then we show the generated images from slice-based distances method for MNIST, CelebA and LSUN, when the number of projections varies. In Appendix E.2, we compare DSW to the projected robust subspace Wasserstein (PRW) in (Paty & Cuturi, 2019; Lin et al., 2020) on MNIST dataset. The comparison is to show Wasserstein-2 distance between the learned distribution and the data distribution versus the execution time. Finally, Appendix E.3 includes a comparison between DSW, DGSW, SW, Max-SW, Max-GSW, and Max-GSW-NN for the joint contrastive inference task on MNIST dataset.

E.1 GENERATIVE MODELS

DGSW results on MNIST: Figure 4(a) shows the convergence of estimators of the learned distribution to the data distribution based on “generalized" sliced distances in the sense of Wasserstein-2

distance. Here, we use the circular function as the defining function for both GSW, Max-GSW, and DGSW (the polynomial function is very expensive in high-dimension). With 10 projections, DGSW produces better performance than GSW with 1000 projections, Max-GSW and Max-GSW-NN. There is a little improvement in the Wasserstein-2 score with DGSW when we increase the number of projections from 10 to 1000. For the computational speed shown in Figure 4(b), DGSW-10 is faster than other reported methods, except the GSW-10 which has the worst Wasserstein-2 score.



(a) (b) (c) (d)

Figure 4: (a) Comparison between DGSW, GSW, Max-GSW and Max-GSW-NN using W_2 distance as metric. Here, GSW, Max-GSW and DGSW use circular function. (b) The computional speed over size of samples. (c) Comparison of Max-SW and DSW with an increasing number of iterations. (d) Comparison between Discrete DSW (dDSW) and other distances including the general version of DSW.

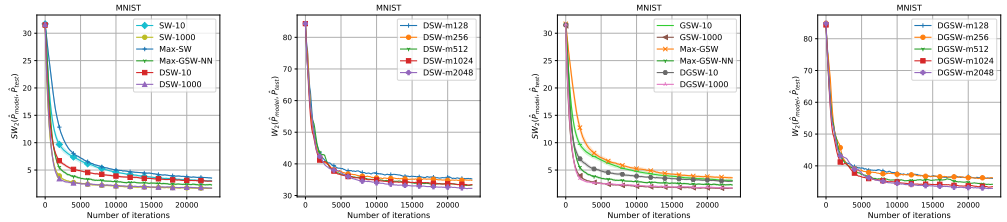
Increasing number of iterations of Max-SW: We increase the number of epochs of Max-SW to 800 in the generative model task on the MNIST dataset. According to the result in Figure 4(c), we observe that the model distribution is closer to the data distribution when the number of iterations increases in Max-SW; however, Max-SW's result is still worse than DSW-10's result. The experiment result suggests that Max-SW requires several more iterations than DSW to get a comparable result in the generative modeling task.

Discrete DSW: We test the performance of a variant of DSW, which is called Discrete DSW (dDSW). The idea of dDSW is that instead of searching in space of all probabilities measures which satisfy the concentration constraint, dDSW searches for the optimal distribution over directions that contains n supports (here, n is also called the number of projections). This distribution also needs to satisfies the concentration constraint to avoid the collapsing of its supports. We then run experiments with dDSW and other sliced-based Wasserstein distances on the MNIST dataset for the generative modeling task. The results are given in Figure 4(d). We observe that dDSW performs quite well comparing to SW and max-SW, namely, dDSW is better than SW with the same number of projections and both dDSW-10 and dDSW-1000 are better than Max-SW. However, both dDSW-10 and dDSW-1000 are worse than DSW-10 and DSW-1000. It suggests that DSW has a better performance than dDSW when the number of projections is similar.

Finally, we would like to remark that when the number of projections n in dDSW is sufficiently large, there is no difference between dDSW and DSW except the optimization problem. It is because the set of discrete measures on the unit sphere is dense over the set of continuous measures on the unit sphere, namely, any continuous distributions on the unit sphere can be approximated sufficiently well by discrete measures. However, as the experiments in Figure 4(d) indicate, we need to choose large number of projections in dDSW such that it has comparable performance with DSW with smaller number of projections; for example, dDSW with 1000 projections is still not as good as DSW with 10 projections in terms of Wasserstein-2 score.

Effects of the number of samples: We conduct experiments to show how sample size (m in Appendix D.1) affects the results of DSW and DGSW in the MEDE framework. According to Figure 5(b), increasing the sample size leads to better performance of DSW. Similarly, increasing the sample size in the MEDE framework that uses DGSW (with circular defining function) helps improve the results, see in Figure 5(d).

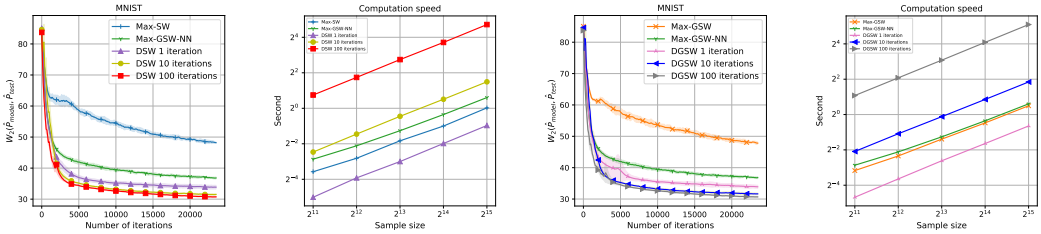
Effects of the number of gradient-updates: In both DSW and DGSW cases, we use a pushforward measure for the distribution over the sphere, and we use neural nets to find it. To learn these neural nets, we use gradient ascent to update their weights. In this experiment, we aim to find out how the number of iterations to update these neural net, affects the performance including the convergence



(a) (b) (c) (d)

Figure 5: (a) Comparison between performances of DSW to SW, Max-SW and Max-GSW-NN using SW_2 distance as metric. (b) The effect of number of samples in minibatch to the convergence of DSW. (c) Comparing DGSW to GSW and Max-GSW-NN using SW_2 distance as metric. Here, GSW and DGSW use circular function. (d) The effect of number of samples in minibatch to the convergence of DGSW.

behavior and computation speed. By increasing the number of updates from 1 to 10, both in DSW and DGSW, model distributions are closer to data distribution; from 10 to 100 updates the results are improved but not too much, see the results in Figures 6(a) and 6(c). However, increasing update steps also lead to a computation problem as the computational time increases considerably. When using 10 or 100 update steps, DSW and DGSW are slower than Max-SW, Max-GSW (50 gradient updates to find the max direction), and Max-GSW-NN (50 gradient updates for the defining neural net function).



(a) (b) (c) (d)

Figure 6: (a) and (c): Increasing the number of times to update push forward measure can improve the performance of both DSW and DGSW; (b) and (d): However, increasing the number of times to update push forward measure leads to slower computation speed.

Table 1: FID score of generator models trained on CIFAR10 (100 epochs), CelebA (50 epochs), and LSUN (20 epochs) datasets in 64x64 resolution. Results are averaged from 5 different runs.

Model	n	CIFAR-10	CelebA	LSUN
SW	10^2	109.7 $\pm$ 5.64	90.11 $\pm$ 10.11	101.57 $\pm$ 3.24
GSW	10^2	103.11 $\pm$ 6.92	87.18 $\pm$ 8.97	92.58 $\pm$ 4.78
ASW	10^2	138.26 $\pm$ 8.31	122.11 $\pm$9.09	
DSW	10^2	**62.83** $\pm$ **6.24**	75.94$\pm$ 5.54	**46.02** $\pm$ **2.15**
DGSW	10^2	68.01 $\pm$ 7.74	**71.08**$\pm$ **4.24**	46.91 $\pm$ 3.98
Max-SW		136.04 $\pm$ 8.35	100.09 $\pm$ 8.34	123.74 $\pm$ 5.51
Max-GSW-NN		86.04$\pm$ 8.68	81.57$\pm$7.72	56.83 $\pm$ 4.04
SW	10^4	98.61 $\pm$ 3.62	82.02 $\pm$ 6.33	62.75 $\pm$ 4.77
GSW	10^4	93.51 $\pm$ 6.12	84.22 $\pm$ 7.93	68.04 $\pm$ 2.17
ASW	10^4	121.38 $\pm$ 6.83	101 $\pm$ 7.36	
DSW	10^4	**56.42** $\pm$ **3.78**	66.85$\pm$ 7.22	**39.68** $\pm$ **2.33**
DGSW	10^4	60.01 $\pm$ 5.58	**65.8**$\pm$ **4.42**	42.04 $\pm$ 4.21

Table 2: Computational speed per minibatch on CelebA and CIFAR10 dataset

Model	n	Second/Minibatch
SW	10^2	0.178
GSW	10^2	0.181
ASW	10^2	0.298
DSW	10^2	0.21
DGSW	10^2	0.212
Max-SW		1.821
Max-GSW-NN		1.895
SW	10^4	0.615
GSW	10^4	0.632
ASW	10^4	1.561
DSW	10^4	1.312
DGSW	10^4	1.384

Quantitative results: We provide full FID scores of all distances mentioned in the papers and also the recent augmented sliced Wasserstein (ASW) (Chen et al., 2020) in Table 1. Based on the results in that table, DSW and DGSW (circular) achieve the best performance among all sliced distances. We also report the computational speed per minibatch in Table 2. The results show that DSW-100 is faster than DSW-10000 while its FID is lower. Regarding ASW, in our experiment, we find that the injective neural network, which is used to transform two target measures, is quite unstable to train and our obtained results with that distance are not good. Moreover, ASW is slower than DSW because ASW needs to double the dimension and still utilizes the uniform measure to slice on the new space. Note that, we use the implementation of ASW in `https://github.com/ShwanMario/ASWD`.

Qualitative results: We show random generated images from trained generators on MNIST, CelebA, CIFAR10 and LSUN datasets in Figures 7-11. Overall, we can see that the distributional approaches, i.e., DSW and DGSW distances, help to improve the quality of synthetic images in both linear and non-linear projection cases.

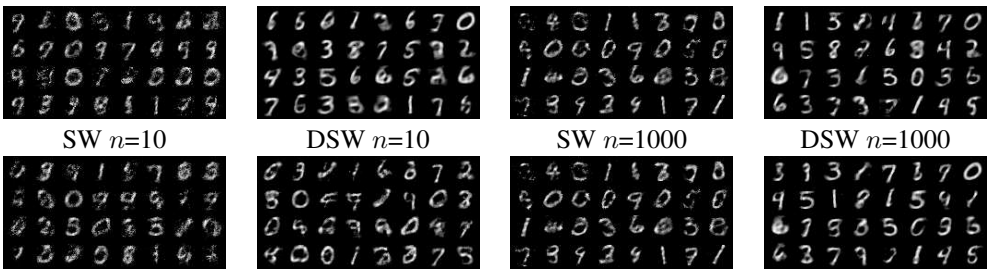


Figure 7: MNIST generated images from different generators, n is the number of projections.

Comparison with the special case of Max-GSW-NN: In Max-GSW-NN (Kolouri et al., 2019), one possible choice of neural network defining function is $g(x, \theta) = \langle x, f(\theta) \rangle$ where $f : \mathbb{S}^{d-1} \to \mathbb{S}^{d-1}$. That function f induces a probability measure on $\mathbb{S}^{d-1}$. Hence, optimizing f is equivalent to optimize over the set of probability measures without any constraints, which gives us an effect that is similar to max-SW. In contrast, the function f in our DSW is to find a push-forward probability measure that distributes high probability to informative directions, and this probability measure is regularized to avoid collapsing to a Dirac measure. To support our previous claim, we also do extra experiments on MNIST in Table 3 to clarify the role of the function f of DSW which makes DSW different from the given special case of Max-GSW-NN. The result shows that this version of Max-GSW-NN is similar to DSW when $\lambda_C = 0$ and both of them have the same performance as Max-SW.

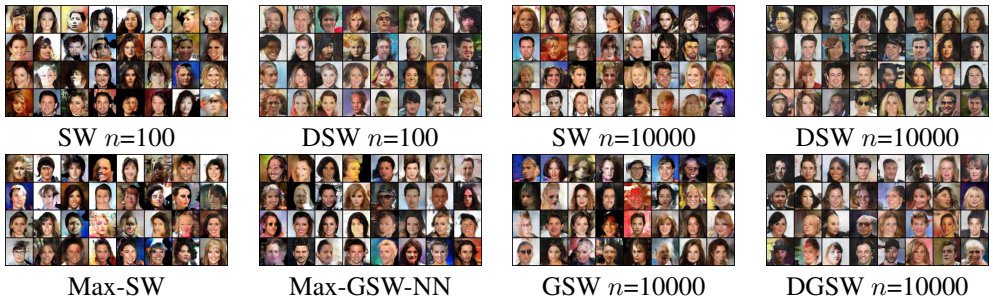


Figure 8: CelebA generated images from different generators, n is the number of projections.

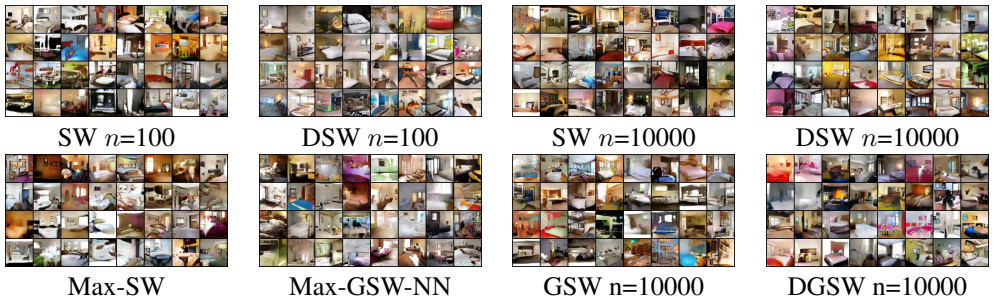


Figure 9: LSUN generated images from different generators where n is the number of projections.

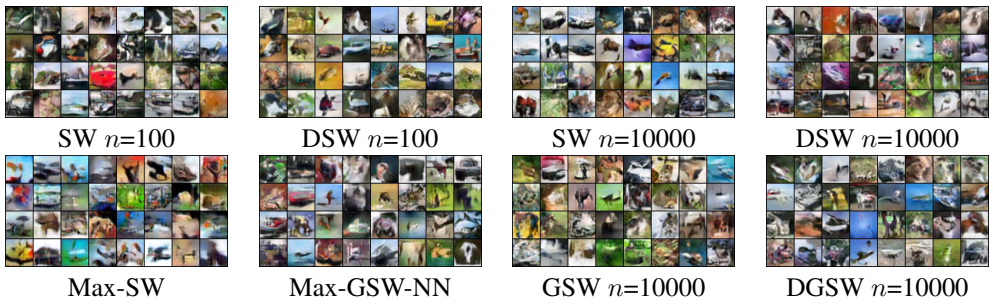


Figure 10: CIFAR10 generated images from different generators, n is the number of projections.

Figure 11: ASW generated images on CelebA and CIFAR10.

E.2 Comparison with Projected Robust Subspace Wasserstein and Wasserstein distance

As shown in (Paty & Cuturi, 2019; Lin et al., 2020), the main idea of projected robust subspace Wasserstein (PRW) is to find the optimal subspace (dimension ≥ 2) such that the Wasserstein-2 distance between two projected measures is maximal.

We first recall the definition of PRW in (Paty & Cuturi, 2019).

Table 3: Comparison with the special case of Max-GSW-NN, denoting Max-GSW-NN(*) in the table, that uses the defining function $g(x, \theta) = \langle x, f(\theta)\rangle$ where $f : \mathbb{S}^{d-1} \to \mathbb{S}^{d-1}$.

Model	λ_C	Wasserstein-2
Max-SW	-	48.64
Max-GSW-NN (*)	-	49.21
DSW-10	0	49.81
DSW-10	1	38.41
DSW-10	10	**33.40**
DSW-10	100	40.08
DSW-10	1000	46.07

Definition 5. *Let $\mathbb{V}_k(\mathbb{R}^d) = \{U \in \mathbb{R}^{d\times k} : U^\top U = I_k\}$ and $\mu, \nu \in \mathcal{P}(\mathbb{R}^d)$. Then, the projection robust 2-Wasserstein (PRW) distance between μ, ν is given by:*

$$PRW_k(\mu, \nu) = \max_{U\in\mathbb{V}_k(\mathbb{R}^d)} W_2(U^\top\sharp\mu, U^\top\sharp\nu). \tag{9}$$

Since the projected dimension is bigger than 1, PRW does not have closed-form solution on the projected space.

Experiments on generative model: We continue to use the MEDE framework on the same settings as previous experiments to compare DSW with PRW and Wasserstein distance (WD). The Wasserstein distance is computed via linear programming algorithm. To solve the optimization on Stiefel manifold in PRW, we use the "geoopt" library (Kochurov et al., 2020). We use one gradient step to solve the optimization problem of both DSW and PRW per one generator update. The experiments are carried out with both DSW and PRW on MNIST dataset. The number of projections of DSW takes value 10 and 1000 and the dimension of the subspace of PRW belongs to the set $\{2, 5, 10, 50\}$. We report the Wasserstein-2 results and the computational time in Table 4 and the generated images in Figure 12.

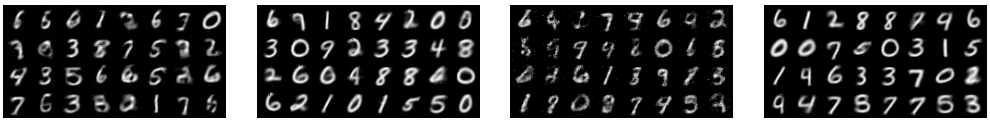
Table 4: Empirical Wasserstein-2 score and computation speed per minibatch on MNIST dataset.

Model	k-dimension	Wasserstein-2	Second/Minibatch
DSW-10	-	34.4	**0.003**
DSW-1000	-	33.11	0.018
PRW	2	65.39	0.086
PRW	5	35.99	0.092
PRW	10	26.57	0.11
PRW	50	**24.38**	0.12
WD	-	24.40	0.11

According to Table 4, DSW with 10 projections obtains a better Wasserstein-2 score than the PRW with 5-dimensional subspace, while its corresponding computational time is 30 times faster that of PRW. When PRW searches for the 50-dimensional subspace, the Wasserstein-2 score only improves 32.25% meanwhile the computational time increases by 10 times.

Next, we show some generated images from both DSW and PRW. We observe that these images are consistent with Wasserstein-2 score in the previous experiments.

Investigation on minibatch's size: We adjust the minibatch's size in the set {32, 64, 128, 256, 512, 1024} and then train the generative model with SW, DSW and Wasserstein distance (WD). The Wasserstein-2 scores of trained models on MNIST dataset are given in Table 5. Consistently, all distances performs better when the size of minibatch increases. When the minibatch's size is 1024, the DSW-10 takes around 0.006 second per minibatch. On the other hand, the Wasserstein distance takes around 0.4 second per minibatch. Therefore, the DSW-10 is around 65 times faster than the Wasserstein distance. Note that, when the minibatch size is 512, the DSW-10 is around 40 times faster than the Wasserstein distance. It suggests that the larger the minibatch size, the faster the DSW compared to the Wasserstein distance.

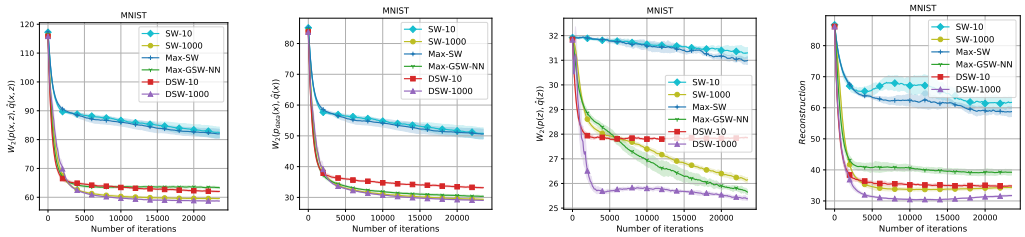


DSW n=10 WD PRW k=5 PRW k=50

Figure 12: MNIST generated images from generators of DSW and PRW. Here, n is the number of projections of DSW and k is the projected dimension of PRW.

Table 5: Effect of minibatch'size on the empirical Wasserstein-2 score.

Model	32	64	128	256	512	1024
SW-10	61.98	60.16	57.99	54.01	50.84	49.71
DSW-10	37.53	37.34	35.42	35.38	34.40	33.74
SW-1000	41.02	40.53	39.80	38.63	37.77	37.45
DSW-1000	37.36	36.77	35.45	34.28	33.11	32.93
WD	34.34	31.20	28.71	26.62	24.40	23.51



(a) (b) (c) (d)

Figure 13: Joint inference model comparisons among DSW, SW, Max-SW, and Max-GSW-NN.

E.3 Joint contrastive inference

We test the performance of our distances in training encoder-generator models on MNIST using joint contrastive inference (JCI). In JCI, the joint generative latent-observed distribution $p_\theta(z, x) = p(z)p_\theta(x|z)$ is matched with the empirical joint latent-observed distribution $\hat{q}_\phi(z, x) = p_{\text{data}}(x)q_\phi(z|x)$ by minimizing a chosen distance (see Appendix D for a description of these models). We evaluate how close the two joint latent-observed distributions $p_\theta(z, x)$ and $\hat{q}_\phi(z, x)$ are, how close their corresponding marginals are (in Wasserstein-2 distance) and the ability of the encoder-generator in reconstructing images. These metrics are shown in Figures 13(a)-(d). The results show that DSW achieves better performance than SW using the same number of projections, with DSW-1000 achieves the best performance among all the other baselines in all metrics. We give experiments to compare DGSW with other non-linear sliced-Wasserstein distances in the joint contrastive inference task in Figure 14. We observed the same behavior as the linear case, the distributional version of GSW using circular function achieves better performance than the other non-linear sliced-based distances.

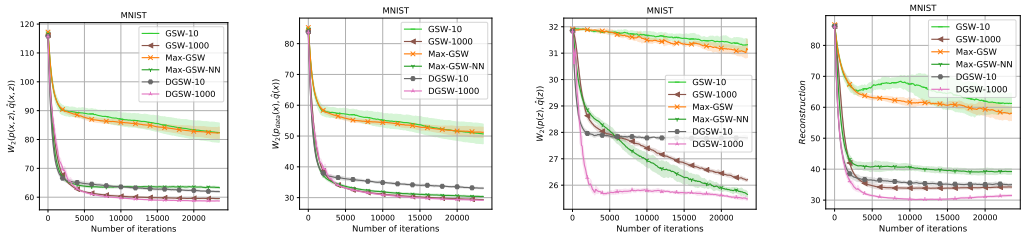


Figure 14: Joint inference model comparisons between non-linear sliced distances.

In order to illustrate the ability to reconstruct images of joint inference models, we show reconstructed images from the MNIST dataset. With 10 projections, SW and GSW were not able to recreate the digits; however, DSW and DGSW can recreate the digits quite correctly. Furthermore, Max-GSW-NN performs well in this task and is better than Max-SW and Max-GSW. When having enough number of projections (for example 1000), it is very hard to compare SW, GSW, DSW, and DGSW by eyes. Nevertheless, according to reconstruction error plots in Figures 13 and 14, DSW, and DGSW distances are still better than the other sliced-based distances.

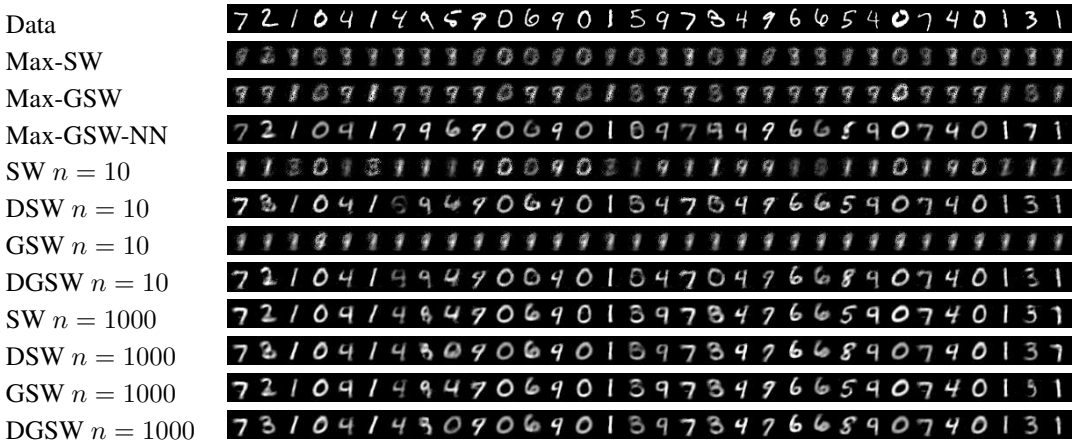


Figure 15: MNIST dataset reconstruction images (n is the number of projections).

F Color Transfer

Color transfer is a famous application of optimal transport (Rabin et al., 2014; Bonneel et al., 2015; Perrot et al., 2016). The goal of color transfer is to map the color plate of the given source RGB image to the given target RGB image. In this appendix, we follow the approach from Muzellec & Cuturi (2019), namely, we use K-means algorithm (with 3000 clusters) to get the quantizations of two input images. With the obtained quantizations, we then perform the distribution transfer algorithm based on sliced-based Wasserstein distances (Rabin et al., 2010; Bonneel et al., 2015). More specifically, we project two quantizations into 1D distributions using the corresponding directions which are drawn from distributions of the sliced-based distances (e.g., SW - uniform distribution, MaxSW - best Dirac distribution, DSW - the optimal distribution over slices). Then, we find 1D alignments between 1D distributions (Monge maps between corresponding projected distributions). After that, with each alignment, we move pixels of the source image with the corresponding pixels of the target images. Finally, we take the average over all directions to get the final transferred images. In our experiment with color transfer, we set the regularized parameter $\lambda_C = 10$ in DSW (cf. equation (2)), and the learning rate equals 0.005 in Max-SW and DSW.

We present the qualitative images in Figure 16. According to the experiment results, we find that SW and DSW create smoother images compared to Max-SW, which creates some noise parts in its transferred images. Furthermore, DSW produces more lively and realistic images than SW, especially when the number of projections is small (e.g., 10 projections).

Comparing with projection pursuit methods: Furthermore, as suggested in (Meng et al., 2019), we also test two versions of SW which use the projection pursuit methods to find the most "informative" projecting direction (correspondingly, directional regression (Li et al., 2007) or sliced average variance estimator (Li, 1991)). We then denote these two variants of SW as drSW and saveSW respectively. We use the code from `https://github.com/ChengzijunAixiaoli/PPMM/`.

The main difference in this framework of color transfer is that it allows to do transfer color with many iterations while in the previous setting we can only do once. Each iteration requires finding the transportation map to move color from the current image (start from the source image) to the target image. In SW and DSW, the transportation map is the average of n 1D transportation maps (n is the

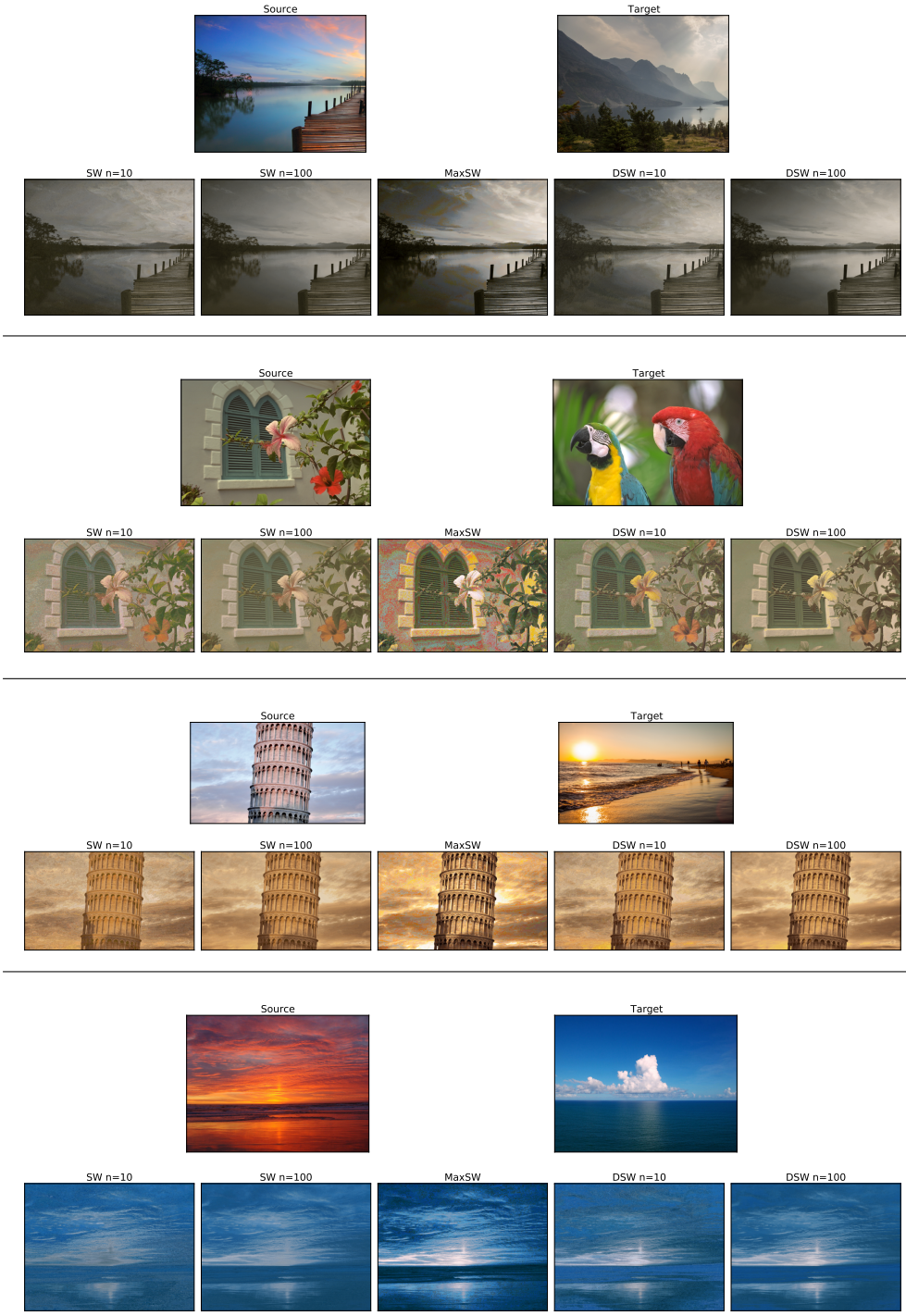
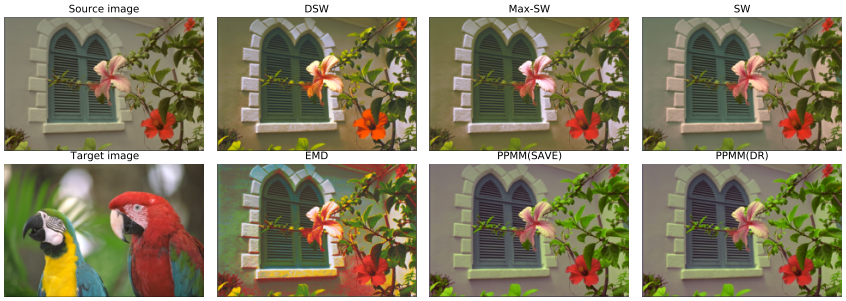


Figure 16: Color Transfer using SW, Max-SW, and DSW (n is the number of projections). The first rows show the source and the target images, and the second rows show the transferred images using the corresponding distances. We use the code from `https://github.com/BorisMuzellec/SubspaceOT/` for the implementation of color transfer. Images are taken from (Reinhard et al., 2001; Bonneel & Coeurjolly, 2019; Flamary & Courty, 2017), `https://github.com/chia56028/Color-Transfer-between-Images`.

number of projections), while in Max-SW, drSW (DR), saveSW (SAVE) the transportation map is provided by the "optimal" projection that is found by the corresponding method.

We show the transfered images in Figure 17 and Figure 18 ($n = 10$ for DSW and SW, $\lambda_C = 10$ in DSW, and the learning rate equals 0.01 in Max-SW and DSW). From Figure 17, when the number of iterations is 1, DSW looks more similar to the original OT transfer (EMD) than Max-SW and SW. Increasing the number of iterations to 10, DSW is slightly better than SW and Max-SW. On the other hand, images from DSW are not as smooth as images from drSW and saveDR. However, drSW and saveDR create some colors that are different from the target image (e.g., the pink color). In Figure 18, DSW is still better than Max-SW and SW because its images are closer to the EMD's images. Projection pursuit methods (drSW and saveSW) also perform well in this case; however it is hard to know whether which method is the best.



The number of iterations is set to 1

The number of iterations is set to 10

Figure 17: Color Transfer using drSW and saveSW using the code from `https://github.com/ChengzijunAixiaoli/PPMM/`.

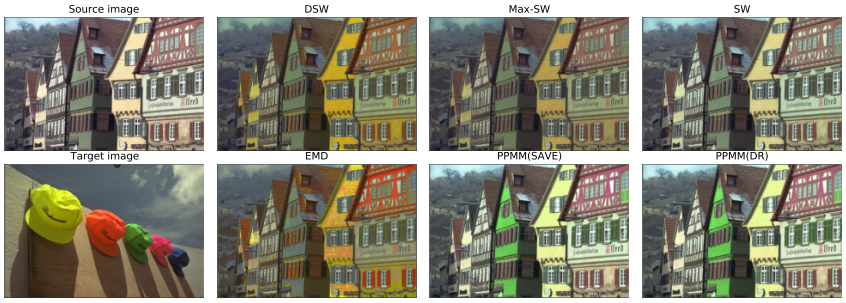
G Experiment Settings

We use a multi layer perceptron (MLP) network with one layer (d^2 parameters where d is the dimension of comparing distributions) and normalized output as the f function in the dual empirical forms of DSW and DGSW. In all experiments, we use norm 2 as the ground metric for the Wasserstein distance. For GSW and DGSW, we use $r = 1000$ for circular function. We use the code at `https://github.com/kimiandj/gsw` for Max-SW and Max-GSW-NN (use 3 MLP layers with Leaky ReLU activation as defining function). In this implementation, Max-SW and Max-GSW-NN uses 50 gradient updates per minibatch to find the optimal direction.

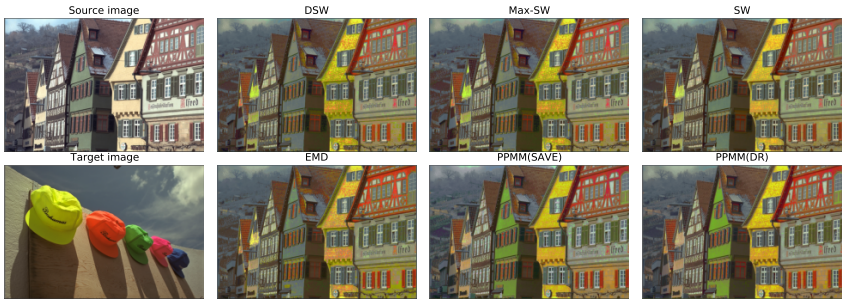
We train models on MNIST, CelebA, CIFAR10 with batch size = 512. On LSUN we use batch size = 4096. We use Adam optimizer for all models with learning rate=0.0005 and betas=(0.5, 0.999), p =2. The range for hidden layer size of the MLP defining function of Max-GSW-NN is (32,100,784,1000). We tune λ_C of DSW and DGSW by grid searching in $(1, 10, 100, 1000)$ in every experiment. The number of epochs for MNIST is 200, CelebA is 50, CIFAR10 is 100, and LSUN is 20.

In evaluation, we use empirical distribution with 10000 samples from two target distribution to compute discrete Wasserstein distance via linear programming..

Generator architecture was used for MNIST dataset:
$z \in \mathbb{R}^{32} \to FC_{100} \to ReLU \to FC_{200} \to ReLU \to FC_{400} \to ReLU \to FC_{784} \to ReLU$



The number of iterations is set to 1



The number of iterations is set to 10

Figure 18: Color Transfer using drSW and saveSW using the code from `https://github.com/ChengzijunAixiaoli/PPMM/`

Generator architecture was used for CelebA, CIFAR10 and LSUN dataset $z \in \mathbb{R}^{100} \rightarrow TransposeConv_{512} \rightarrow BatchNorm \rightarrow ReLU \rightarrow TransposeConv_{256} \rightarrow BatchNorm \rightarrow ReLU \rightarrow TransposeConv_{128} \rightarrow BatchNorm \rightarrow ReLU \rightarrow TransposeConv_{64} \rightarrow BatchNorm \rightarrow ReLU \rightarrow TransposeConv_{1} \rightarrow Tanh$
Discriminator architecture was used for CelebA, CIFAR10 and LSUN dataset:

First part: $x \in \mathbb{R}^{64\times64\times3} \rightarrow Conv_{64} \rightarrow LeakyReLU_{0.2} \rightarrow Conv_{128} \rightarrow BatchNorm \rightarrow LearkyReLU_{0.2} \rightarrow Conv_{256} \rightarrow BatchNorm \rightarrow LearkyReLU_{0.2} \rightarrow Conv_{512} \rightarrow BatchNorm \rightarrow Tanh$
Second part: $Conv_1 \rightarrow Sigmoid$

Joint Contrastive inference encoder architecture on MNIST:
$x \in \mathbb{R}^{28\times28} \rightarrow FC_{400} \rightarrow LeakyReLU_{0.2} \rightarrow FC_{200} \rightarrow LeakyReLU_{0.2} \rightarrow FC_{100} \rightarrow LeakyReLU_{0.2} \rightarrow FC_{32}$

Joint Contrastive inference deocder architecture on MNIST:
$z \in \mathbb{R}^{32} \rightarrow FC_{100} \rightarrow ReLU \rightarrow FC_{200} \rightarrow ReLU \rightarrow FC_{400} \rightarrow ReLU \rightarrow FC_{784} \rightarrow ReLU$