Towards Better Spherical Sliced-Wasserstein Distance Learning with Data-Adaptive Discriminative Projection Direction

Hongliang Zhang[1], Shuo Chen[2], Lei Luo[1]*, Jian Yang[1]*

[1]PCA Lab, Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, School of Computer Science and Engineering, Nanjing University of Science and Technology, China
[2]School of Intelligence Science and Technology, Nanjing University, China
{zhang1hongliang, cslluo, csjyang}@njust.edu.cn, shuo.chen@nju.edu.cn

Abstract

Spherical Sliced-Wasserstein (SSW) has recently been proposed to measure the discrepancy between spherical data distributions in various fields, such as geology, medical domains, computer vision, and deep representation learning. However, in the original SSW, all projection directions are treated equally, which is too idealistic and cannot accurately reflect the importance of different projection directions for various data distributions. To address this issue, we propose a novel data-adaptive Discriminative Spherical Sliced-Wasserstein (DSSW) distance, which utilizes a projected energy function to determine the discriminative projection direction for SSW. In our new DSSW, we introduce two types of projected energy functions to generate the weights for projection directions with complete theoretical guarantees. The first type employs a non-parametric deterministic function that transforms the projected Wasserstein distance into its corresponding weight in each projection direction. This improves the performance of the original SSW distance with negligible additional computational overhead. The second type utilizes a neural network-induced function that learns the projection direction weight through a parameterized neural network based on data projections. This further enhances the performance of the original SSW distance with less extra computational overhead. Finally, we evaluate the performance of our proposed DSSW by comparing it with several state-of-the-art methods across a variety of machine learning tasks, including gradient flows, density estimation on real earth data, and self-supervised learning.

Introduction

In real-world scenarios, more and more tasks involve defining data distributions on the hypersphere, highlighting the importance and universality of spherical geometry. These tasks include characterizing the density distribution of geophysical (Di Marzio, Panzera, and Taylor 2014; An et al. 2024; Hu et al. 2023, 2024) or meteorology data (Besombes et al. 2021), magnetic imaging of the brain in the medical field (Vrba and Robinson 2001), texture mapping in computer graphics (Dominitz and Tannenbaum 2010), etc, where the latent representation is mapped to a bounded space commonly known as a sphere (Wang and Isola 2020; Chen et al. 2020).

The distribution analysis on the hypersphere is often focused on the statistical study of directions, orientations, and rotations. It is known as circle or sphere statistical analysis (Jammalamadaka and Sengupta 2001). Recently, there has been a growing interest in comparing probability measures on the hypersphere using Optimal Transport (OT) (Cui et al. 2019). This is driven by its appealing statistical, geometrical, and topological properties (Peyré, Cuturi et al. 2019a).

Two critical challenges in applying OT theory are the high computational complexity and the curse of dimensionality (Peyré, Cuturi et al. 2019b). These issues have led to a growing focus on developing faster solving tools (Cuturi 2013) and computationally efficient alternative distance metrics (Kolouri et al. 2019a). One such metric is the Sliced-Wasserstein (SW) distance, which has lower computational complexity and does not suffer from the curse of dimensionality (Nietert et al. 2022). This distance metric inherits similar topological properties from the Wasserstein distance (Nadjahi et al. 2020) and is widely used as an alternative solution for comparing probability measures. The SW distance is defined as the expectation of the one-dimensional Wasserstein distance between two projected measures over a uniform distribution on the unit sphere. However, due to the intractability of this expectation, Monte Carlo estimation is often used to approximate the SW distance. Some SW variants aim to learn a discriminative or optimal projections distribution from training data, PAC-SW (Ohana et al. 2023) learns a discriminative projection direction distribution from training data based on the PAC-Bayesian theory. DSW (Nguyen et al. 2021) learns optimal projections distribution from training data by the neural network.

Recently, SW distance has been employed to compare probability measures on the hypersphere due to its computational efficiency and simplicity of implementation. This has led to the development of spherical sliced OT approaches (Bonet et al. 2023; Quellmalz, Beinert, and Steidl 2023; Tran et al. 2024). The main challenge in developing these methods is extending the classical Radon transform to its spherical counterparts. Quellmalz et al. (Quellmalz, Beinert, and Steidl 2023) introduced two spherical extensions for the Radon transform to define sliced OT on the sphere: the vertical slice transform (Rubin 2018), and the normalized semicircle transform (Groemer 1998). Bonet et al. (Bonet

*corresponding authors

et al. 2023) proposed a new spherical Radon transform and leveraged the closed-form solution of the Wasserstein distance on the circle to define SSW for empirical probability measures. Both works (Bonet et al. 2023; Quellmalz, Beinert, and Steidl 2023) map a distribution defined on a hypersphere to its marginal distributions on a unit circle, and solve circular OT to compare these marginals. However, calculating the OT between two one-dimensional measures defined on a circle is more expensive. To address this issue, Tran et al. (Tran et al. 2024) first adopted the stereographic projection to transform the hypersphere into a hyperplane, and then utilized the classic Radon transform to define Stereographic Spherical Sliced-Wasserstein (S3W).

These existing works on SSW usually assume that all projection directions contribute equally when calculating the expectation of one-dimensional Wasserstein distance. However, this assumption is too idealistic and inconsistent with real-world situations, as this practice ignores the discriminative information from different projection directions in SSW. This paper aims to learn the better SSW distance to handle practical applications by considering the discrimination between projection directions. Specifically, we emphasize that different projection directions in SSW have varying degrees of importance and propose using projected weights adaptively learned from data to characterize them, where the weight is directly proportional to the Wasserstein distance of the corresponding projection direction. As weights can reflect the distribution of the data, this approach effectively captures valuable discriminative information hidden in various directions. This is beneficial in improving the accuracy of SSW distance. Towards this end, we propose two types of the projected energy function (*i.e.*, non-parametric and parametric forms) to learn the weights of projection directions under the projection of supports, taking into account the efficiency and performance. In the non-parametric form, the weight is calculated from the projected Wasserstein distance using a non-parametric function, such as softmax, identity, or polynomial function. In the parametric form, a parametric neural network such as linear, nonlinear, or attention mechanisms is used to generate the weight based on the input of the projected supports. Our new method effectively characterizes the importance of each projection direction, allowing for a more precise computation of the discrepancy between real-world distributions. Our contributions are as follows:

- We first propose learning discriminative projection direction for SSW distance which is implemented by the non-parametric function and the parametric neural network to consider specific data distributions.
- We provide the corresponding theoretical analysis and mathematical derivation to guarantee the topological and statistical properties of the novel DSSW distance.
- We apply our DSSW distance to several classical machine learning tasks, and the experimental results show that our DSSW is superior to the existing SSW, S3W, SW and Wasserstein distance.

Background

The goal of this work is to define a DSSW distance on the hypersphere $S^{d-1} = \left\{x \mid x \in \mathbb{R}^d,\ \|x\|_2 = 1\right\}$. It is necessary to first review the definition of Wasserstein distance on manifolds: the SW distance on $\mathbb{R}^d$ and the SSW distance on the hypersphere S^{d-1}.

Wasserstein Distance. Let M be a Riemannian manifold equipped with the distance $d(\cdot,\cdot) : M \times M \to \mathbb{R}_+$. For $1 \leq p < +\infty$, let two probability measures μ and $\nu \in \mathcal{P}_p(M) = \{\mu \mid \mu \in \mathcal{P}(M),\ \int_M d^p(x, x_0)\,\mathrm{d}\mu(x) < +\infty \text{ for any } x_0 \in M\}$ be defined on manifold M with p finite moments, and $\mathcal{P}(M)$ means the set of all probability measures defined on M. The aim of OT is to transport the mass from μ to ν in a way that minimizing the expectation of transport distance. So the p-Wasserstein distance (Peyré, Cuturi et al. 2019b) can be defined as

$$W_p^p(\mu,\nu) = \inf_{\gamma\in\Pi(\mu,\nu)} \int_{M\times M} d(x,y)^p\,\mathrm{d}\gamma(x,y), \quad (1)$$

where $\Pi(\mu,\nu)$ is the set of couplings of μ and ν.

Unfortunately, for discrete probability measures with n samples, the Wasserstein distance can be calculated by linear programs with the computational complexity of $\mathcal{O}\left(n^3 \log n\right)$, so it is computationally expensive. Therefore, the alternative distance metrics with lower computational complexity are explored in Euclidean spaces. One of the widely adopted alternative distances is the SW distance.

Sliced-Wasserstein Distance. For one dimensional measures $\mu,\ \nu \in \mathcal{P}_p(\mathbb{R}^d)$, the Wasserstein distance between μ and ν has the closed form as

$$W_p^p(\mu,\nu) = \int_0^1 \left|F_\mu^{-1}(t) - F_\nu^{-1}(t)\right|^p \mathrm{d}t, \quad (2)$$

where F_μ^{-1} and F_ν^{-1} are the quantile functions of μ and ν. This property can be used to define the p-SW distance (Bonneel et al. 2015) as

$$SW_p^p(\mu,\nu) = \int_{S^{d-1}} W_p^p(P^\theta\mu, P^\theta\nu)\mathrm{d}\lambda(\theta), \quad (3)$$

where λ is the uniform distribution on the unit sphere $S^{d-1} = \left\{\theta \mid \theta \in \mathbb{R}^d,\ \|\theta\|_2 = 1\right\}$. For any $x \in \mathbb{R}^d$, we define $P^\theta(x) = \langle x, \theta\rangle$ termed as the projection of x. Since the expectation in the definition of the SW distance is intractable to calculate, the Monte-Carlo estimation is adopted to approximate the SW distance with the computational complexity of $\mathcal{O}\left(Ln\left(d + \log n\right)\right)$ as

$$\widehat{SW}_p^p(\mu,\nu) = \frac{1}{L}\sum_{\ell=1}^{L} W_p^p(P^{\theta_\ell}\mu, P^{\theta_\ell}\nu), \quad (4)$$

where $\{\theta_\ell\}_{\ell=1}^L \overset{i.i.d.}{\sim} \mathcal{U}(\mathbb{S}^{d-1})$ are termed as projection directions sampled from the spherical uniform distribution $\mathcal{U}(\mathbb{S}^{d-1})$, and L is the number of projections used for Monte-Carlo approximation.

Spherical Sliced-Wasserstein Distance. For $\mu,\ \nu \in \mathcal{P}_p(S^{d-1})$, we can define the SSW distance (Bonet et al. 2023) between μ and ν as

$$SSW_p^p(\mu,\nu) = \int_{\mathbb{V}_{d,2}} W_p^p(P_\#^U\mu, P_\#^U\nu)\mathrm{d}\sigma(U), \quad (5)$$

where

$$P_{\#}^{U}(x) = \frac{U^T x}{\|U^T x\|_2}, \tag{6}$$

and σ is the uniform distribution over the Stiefel manifold (Bendokat, Zimmermann, and Absil 2024) $\mathbb{V}_{d,2} = \{U \mid U \in \mathbb{R}^{d\times 2},\ U^T U = I_2\}$. $P_{\#}^{U}(x)$ denotes the geodesic projection on the circle determined by U. The SSW_1 can be computed by the binary search algorithm or the level median formulation, while SSW_2 can be calculated via Proposition 1 in (Bonet et al. 2023).

Method

It is well-known that computing the SSW distance involves averaging the projected Wasserstein distances across all projection directions. This means that each projection direction is given equal weight, resulting in a lack of discrimination. To improve the discriminative power of the projection directions, we propose assigning different weights to each direction. Our approach introduces formulating a novel SSW distance that incorporates these weights. Additionally, we present the projected energy function designed to generate these weights for the projection directions.

Definition 1 (**DSSW Distance**). *For $p \geq 1$, dimension $d \geq 1$, two probability measures $\mu \in \mathcal{P}_p(S^{d-1})$ and $\nu \in \mathcal{P}_p(S^{d-1})$, and the projected energy function $f : \mathbb{S}^n \times \mathbb{S}^n \to (0, 1)$, the DSSW distance between μ and ν is defined as follows:*

$$DSSW_p^p(\mu, \nu; f) = \int_{\mathbb{V}_{d,2}} f\left(P_{\#}^{U}\mu, P_{\#}^{U}\nu\right) \cdot W_p^p(P_{\#}^{U}\mu, P_{\#}^{U}\nu)\mathrm{d}\sigma(U), \tag{7}$$

where σ is the uniform distribution over the Stiefel manifold (Bendokat, Zimmermann, and Absil 2024) $\mathbb{V}_{d,2} = \{U \mid U \in \mathbb{R}^{d\times 2},\ U^T U = I_2\}$. $P_{\#}^{U}(x)$ denotes the geodesic projection on the circle determined by U.

The projected energy function f transforms the projection of two probability measures μ and ν into the weights of the projection directions. It can effectively learn the weights for the projection directions from the data distribution.

We now provide the detailed definition and formulation of the projected energy function f as follows:

Definition 2 (**Projected Energy Function**). *For $p \geq 1$, dimension $d \geq 1$, two probability measures $\mu \in \mathcal{P}_p(S^{d-1})$ and $\nu \in \mathcal{P}_p(S^{d-1})$, and two transformation functions g and h, the projected energy function f used to calculate the weights for the ℓ-th projection direction is defined as follows:*

$$f\left(P_{\#}^{U_\ell}\mu, P_{\#}^{U_\ell}\nu\right) := \frac{g\left(h(P_{\#}^{U_\ell}\mu, P_{\#}^{U_\ell}\nu)\right)}{\sum_{k=1}^{L} g\left(h(P_{\#}^{U_k}\mu, P_{\#}^{U_k}\nu)\right)}, \tag{8}$$

where $P_{\#}^{U_\ell}\mu$ is the projection of μ on the ℓ-th projection direction.

Taking into account the efficiency, we propose the non-parametric form of the projected energy function f. In this form, both the functions h and g are non-parametric. Specifically, h is defined as Eq. (1) to calculate the projected Wasserstein distance, while $g : [0, +\infty) \to (0, +\infty)$ transforms the projected Wasserstein distance into the weights of the projection directions. Following the implementation in (Nguyen and Ho 2024), the non-parametric g can be the exponential function (*i.e.*, $g(x) = e^x$), the identity function (*i.e.*, $g(x) = x$), or the polynomial function (*i.e.*, $g(x) = x^2$). Then, normalization is performed on the weights obtained from the projection directions. The non-parametric projected energy function shows that the weights are proportional to the Wasserstein distance for the respective projection direction. Given the above calculation steps, it can be seen that the only additional step in the original SSW calculation process is the calculation of the weights for the projection direction. Therefore, the added calculation time can be disregarded. This means that the proposed DSSW with the non-parametric projected energy function achieves better performance with minimal additional computing overhead compared to the original SSW. In the subsequent sections, DSSW with the exponential, identity, and polynomial functions will be referred to as DSSW (exp), DSSW (identity), and DSSW (poly), respectively.

Considering the accuracy, we present the parametric projected energy function f. In this case, h is represented by a parameterized neural network h_ψ, where ψ denotes the learnable weights of the neural network h_ψ. The network h_ψ can be a linear neural network, a nonlinear neural network, or an attention mechanism. The detailed training configuration for the parameterized neural network h_ψ is described in Algorithm 2 in Appendix Section B. Meanwhile, g is a non-parametric function and is specialized as the exponential function (*i.e.*, $g(x) = e^x$). The normalization operation combined with the non-parametric function g is equivalent to calculating the famous Softmax function. Using the Stochastic Gradient Descent (SGD) method to obtain more precise projection directions via the parameterized neural network h_ψ, our proposed DSSW with a parameterized projected energy function outperforms that with the non-parametric projected energy function and the original SSW. However, it does come with a higher computational cost compared to these two forms. In the subsequent sections, we will refer to DSSW with the linear neural network, nonlinear neural network, and attention mechanism as DSSW (linear), DSSW (nonlinear), and DSSW (attention), respectively.

Proposition 1. *For any $p \geq 1$ and the projected energy function f, the DSSW distance $DSSW_p$ is positive and symmetric.*

The definition of the DSSW distance implies that it does not satisfy identity due to the case that the different points on the hypersphere S^{d-1} may share the same projection on the circle determined by U. The proofs for the related propositions and theorems can be found in Appendix Section A.

Proposition 2. *For any $p \geq 1$ and the projected energy function f, let μ_k, $\mu \in \mathcal{P}_p(S^{d-1})$. If $\lim_{k\to+\infty} \mu_k = \mu$, then $\lim_{k\to+\infty} DSSW_p^p(\mu_k, \mu; f) = 0$.*

Proposition 2 indicates that $DSSW_p$ is asymptotically convergent. It implies that our DSSW distance also satisfies the property of weak convergence that is one of the most crucial requirements that a distance metric should satisfy (Nadjahi et al. 2019).

Proposition 3. *For any $p \geq 1$, suppose that for μ, $\nu \in P(S^1)$, with empirical measures $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n}\delta_{x_i}$, and $\hat{\nu} = \frac{1}{n}\sum_{i=1}^{n}\delta_{y_i}$, where $\{x_i\}_{i=1}^{n} \sim \mu$, $\{y_i\}_{i=1}^{n} \sim \nu$ are independent samples, we have*

$$\mathbb{E}[|W_p^p(\hat{\mu}_n, \hat{\nu}_n) - W_p^p(\mu, \nu)|] \leq \beta(p, n), \tag{9}$$

where $\beta(p, n)$ is independent of the dimensionality d and only depends on p and n. Then, for the projected energy function f and μ, $\nu \in P(S^{d-1})$ with empirical measures $\hat{\mu}$ and $\hat{\nu}$, there exists a universal constant C such that

$$\mathbb{E}[|DSSW_p^p(\hat{\mu}_n, \hat{\nu}_n; f) - DSSW_p^p(\mu, \nu; f)|] \leq C\beta(p, n). \tag{10}$$

Proposition 3 demonstrates that the sample complexity of DSSW is independent of the dimension. This insight also verifies that our DSSW distance, akin to the SW distance, can avoid the curse of dimensionality.

Theorem 1. *For any $p \geq 1$, two probability measures μ and $\nu \in P(S^1)$, and the projected energy function f, there exists a universal constant C such that the error made with the Monte Carlo estimate of $DSSW_p^p$ can be bounded as*

$$\begin{aligned}&\mathbb{E}_U\left[\left|\widehat{DSSW}_{p,L}^{p}(\mu, \nu; f) - DSSW_p^p(\mu, \nu; f)\right|\right]^2 \\ &\leq \frac{C^2}{L} Var_U\left(W_p^p\left(P_{\#}^U\mu, P_{\#}^U\nu\right)\right),\end{aligned} \tag{11}$$

where $\widehat{DSSW}_{p,L}^{p}(\mu, \nu; f) = \frac{1}{L}\sum_{\ell=1}^{L} f\left(P_{\#}^{U_\ell}\mu, P_{\#}^{U_\ell}\nu\right) \cdot W_p^p\left(P_{\#}^{U_\ell}\mu, P_{\#}^{U_\ell}\nu\right)$ with $\{U_\ell\}_{\ell=1}^{L} \sim \sigma$ independent samples. L is referred to as the number of projections.

Theorem 1 highlights that the projection complexity of DSSW depends on the convergence rate of the Monte Carlo approximation towards the true integral that has been derived for sliced-based distances in (Nadjahi et al. 2020). This indicates that the estimation error in the Monte Carlo approximation is determined by the number of projections L and the variance of the evaluations of the Wasserstein distance (Nadjahi et al. 2020).

Implementation Details

Similar to the SSW distance, we adopt Monte-Carlo estimation to approximate the integral on $\mathbb{V}_{d,2}$ as in Eq. (5).

We begin by randomly sampling L projections $\{U_\ell\}_{\ell=1}^{L}$ from the uniform distribution σ on the Stiefel manifold $\mathbb{V}_{d,2}$. Each projection is obtained by first constructing a matrix $Z \in \mathbb{R}^{d\times 2}$ with each element drawn from the standard normal distribution $\mathcal{N}(0, 1)$, followed by QR decomposition of each projection.

We then project the points on the circle S^1 according to Eq. (6) and calculate the coordinate of each point in each projection direction on this circle S^1 using the formula $\tilde{x}_i^\ell = (\pi + \text{atan2}(-x_{i,1}^\ell, -x_{i,2}^\ell))/(2\pi)$.

Then, we can compute the Wasserstein distance on the circle S^1 and determine the weights of the projection directions using Eq. (8). The detailed computation procedure of computing the Wasserstein distance on the circle S^1 for $p = 1$ and $p = 2$ can be referred to (Bonet et al. 2023). Finally, we can calculate the DSSW distance using Eq. (7). The pseudocode for computing the DSSW distance is provided in Algorithms 1 and 2 in Appendix Section B.

Computation Complexity. Given n samples from μ and m samples from ν, along with L projections. Just as the work (Bonet et al. 2023), we can finish the QR factorization of L matrices of size $d \times 2$ in $\mathcal{O}(dL)$. Projecting the points on the circle S^1 can be finished in $\mathcal{O}((m+n)dL)$. The complexity of computing the general SSW_p can be written as $\mathcal{O}\left(L(m+n)\left(d+\log\left(\frac{1}{\epsilon}\right)\right)+Ln\log n+Lm\log m\right)$, where ϵ denotes the desired accuracy. The complexity of calculating the weights of the projection directions is $\mathcal{O}(L)$ when using the non-parametric projected energy function f. When using the parametric projected energy function f, the complexity of calculating the weights of the projection directions is $\mathcal{O}(TL)$, where T is the maximum iterations for training the parameterized neural network. Therefore, the total complexity of computing the DSSW distance utilizing the non-parametric projected energy function f is $\mathcal{O}\left(L(m+n)\left(d+\log\left(\frac{1}{\epsilon}\right)\right)+Ln\log n+Lm\log m+L\right)$. In contrast, for the parametric projected energy function f, the total complexity of our proposed method is $\mathcal{O}\left(L(m+n)\left(d+\log\left(\frac{1}{\epsilon}\right)\right)+Ln\log n+Lm\log m+TL\right)$.

Runtime Comparison. We conducted runtime comparisons between various distances between the uniform distribution and the von Mises-Fisher distribution on $\mathbb{S}^{100}$. The results, shown in Figure 1, are averaged over 50 iterations for varying sample sizes of each distribution. For all sliced approaches, we used $L = 200$ projections. The results in Figure 1 include our DSSW with the non-parametric projected energy function variant (exp). It can be observed that the runtime curve of DSSW (exp) closely aligns with that of SSW, indicating that the additional computing overhead introduced by our DSSW (exp) is negligible. Due to space limitations, runtime comparisons for DSSW with other non-parametric projected energy function variants (identity and poly) and DSSW with the parametric projected energy function variants (linear, nonlinear, and attention) are provided in Appendix Section C.1. Furthermore, we explore the evolution of our DSSW across varying dimensions, number of projections, and rotation numbers in Appendix Section C.2, along with the runtime analysis of the proposed DSSW in Appendix Section C.3.

Experiments

In line with previous works (Bonet et al. 2023; Tran et al. 2024), we conducted five different numerical experiments to validate the effectiveness of our method in comparison to SW, SSW (Bonet et al. 2023) and S3W distance (Tran et al. 2024), where our DSSW distance serves as a loss to measure the distribution discrepancy on the sphere. The results of these experimental are detailed in this section and Appendix Section C. All our experiments are implemented by

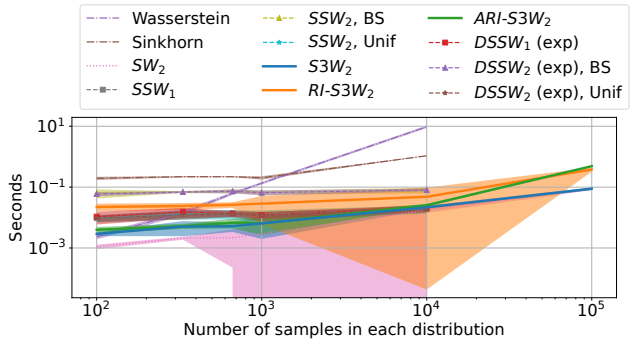


Figure 1: Runtime comparison for Wasserstein distance, Sinkhorn distance with geodesic distance as cost function, SW_2 (SW distance), SSW_1 distance with the level median, SSW_2 distance with binary search (BS), SSW_2 distance against a uniform distribution (Unif), $S3W_2$ distance, $RI\text{-}S3W_2$ (rotationally invariant extension of $S3W_2$) distance, $ARI\text{-}S3W_2$ (amortized rotationally invariant extension of $S3W_2$) distance, $DSSW_1$ (exp) (ours), $DSSW_2$ (exp), BS (ours), $DSSW_2$ (exp), Unif (ours).

PyTorch (Paszke et al. 2019) on Ubuntu 20.04 and a single NVIDIA RTX 4090 GPU.

Gradient Flows on The Sphere

Suppose the explicit form of the target distribution is unknown and only samples $\{y_j \in \widehat{\nu_n}\}_{j=1}^N$ are available, our goal is to iteratively minimize the objective function $\arg\min_{\mu} d\left(\widehat{\mu_i}, \widehat{\nu_{n_i}}\right)$, where d is a distance metric such as SW, SSW, S3W, or DSSW. To achieve this goal, we employ the Projected Gradient Descent (PGD) algorithm (Madry et al. 2017) to estimate the target distribution with the update rule as follows:

$$\begin{cases} x_{i,k+1}^{'} = x_{i,k} - \gamma \cdot \nabla_{x_{i,k}} \text{DSSW}\left(\hat{\mu}_k, \hat{\nu}_{n_i}\right) \\ x_{i,k+1} = \dfrac{x_{i,k+1}^{'}}{\left\| x_{i,k+1}^{'} \right\|_2}, \end{cases} \tag{12}$$

where γ is the learning rate for the update rule, i denotes the index of the mini-batches, and k is the gradient step.

We present both qualitative and quantitative results using the Negative Log-likelihood (NLL) and the logarithm of the 2-Wasserstein distance ($\log W_2$) as evaluation metrics, including mean and standard deviation for each. The mini-batch results for all distances are shown in Table 1, and the Molleweide projections of the mini-batch are illustrated in Figure 2. Table 1 shows that our DSSW performs on par or better than other baselines. Additionally, it is evident that our DSSW with a parametric projected energy function surpasses the non-parametric from. Overall, DSSW demonstrates superior performance in accurately learning the target distribution compared to other distances.

Full-batch results are reported in Appendix Section C.4. These results indicate that all distance measures perform well in learning the target distribution. In cases where the

	Distance	NLL ↓	$\log W_2$ ↓
Mini-batch	SW	-282.48 ± 17.42	-2.77 ± 0.10
	SSW	-287.11 ± 6.22	-2.78 ± 0.08
	S3W	-181.38 ± 8.72	-2.61 ± 0.07
	RI-S3W (1)	-213.63 ± 19.24	-2.68 ± 0.09
	RI-S3W (5)	-256.23 ± 10.72	-2.77 ± 0.13
	RI-S3W (10)	-285.20 ± 13.22	-2.77 ± 0.11
	ARI-S3W (30)	-291.38 ± 16.48	-2.82 ± 0.12
	DSSW (exp)	-316.38 ± 6.90 ‡	-2.92 ± 0.10 ‡
	DSSW (identity)	-310.45 ± 5.00 ‡	-2.94 ± 0.12 ‡
	DSSW (poly)	-307.22 ± 7.50 ‡	-2.94 ± 0.11 ‡
	DSSW (linear)	-319.41 ± 6.72 ‡	-2.94 ± 0.10 ‡
	DSSW (nonlinear)	-319.96 ± 6.65 ‡	**-2.97 ± 0.15** ‡
	DSSW (attention)	**-320.16 ± 5.58** ‡	-2.93 ± 0.10 ‡

Table 1: Mini-batch comparison between different distances as loss for gradient flows averaged over 10 training runs. Notation "‡" indicates that DSSW variants are significantly better than the best baseline method using t-test when the significance level is 0.05.

density of the target distribution is known up to a constant, we utilize the Sliced-Wasserstein Variational Inference (SWVI) framework (Yi and Liu 2023), optimized through MCMC (Doucet, de Freitas, and Gordon 2001) methods. This approach does not require optimization or a tractable approximate posterior family, as detailed in Appendix C.8. The SWVI results further validate the accuracy of our DSSW method in approximating the target distribution compared to other competitors.

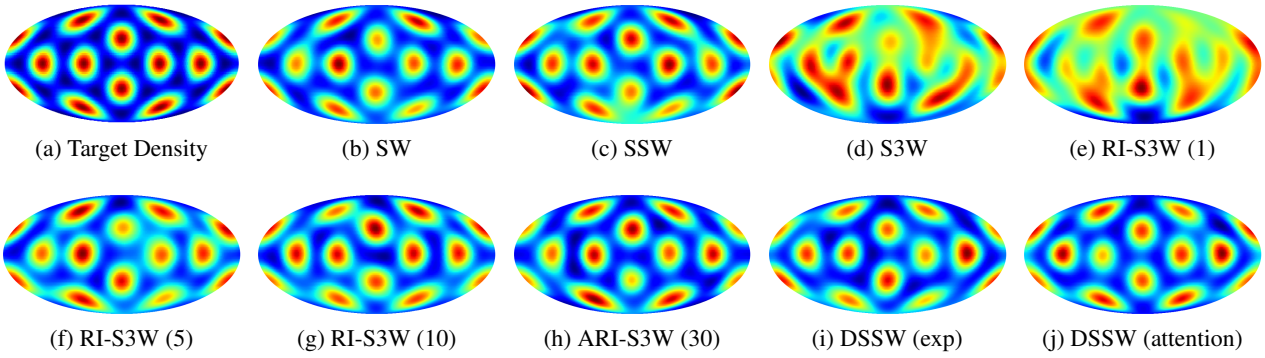
Earth Density Estimation

We evaluate the performance of our proposed DSSW on the density estimation task using normalizing flows on $\mathbb{S}^2$. In alignment with (Bonet et al. 2023; Tran et al. 2024), we adopt three datasets (Mathieu and Nickel 2020): Earthquake (NOAA 2022), Flood (Brakenridge 2017) and Fire (EOSDIS 2020). The earth's surface is modeled as a spherical manifold. Following the implementation of (Bonet et al. 2023; Tran et al. 2024), we utilize an exponential map normalizing flow model (Rezende et al. 2020), which is optimized by $\min_T DSSW\left(T_{\#}\mu, z\right)$. In this formulation, T is the transformation introduced by the model, μ is the data distribution known by sampling samples $\{x_i\}_{i=1}^N$, and z is a prior distribution on $\mathbb{S}^2$. The learned density f_μ can be obtained by

$$f_\mu(x) = z\left(T(x)\right)\left|\det J_T(x)\right|, \ \forall x \in \mathbb{S}^2, \tag{13}$$

where $J_T(x)$ means the the Jacobian of T at x.

The NLL values of density estimation on three earth datasets are demonstrated in Table 2. Stereo (Gemici, Rezende, and Mohamed 2016) first projects samples from $\mathbb{S}^2$ to $\mathbb{R}^2$ and then applies the Real NVP (Dinh, Sohl-Dickstein, and Bengio 2017) model in the projected space. The results show that our proposed DSSW outperforms all other baselines. Specifically, on the Earthquake dataset, DSSW (linear) achieves the best performance, while DSSW (nonlinear) performs best on the Flood and Fire datasets. These results



(a) Target Density (b) SW (c) SSW (d) S3W (e) RI-S3W (1)

(f) RI-S3W (5) (g) RI-S3W (10) (h) ARI-S3W (30) (i) DSSW (exp) (j) DSSW (attention)

Figure 2: The Mollweide projections for mini-batch projected gradient descent. We use 1, 5, and 30 rotations for RI-S3W (1), RI-S3W (5), and RI-S3W (10), respectively. We also use 30 rotations with a pool size of 1000 for ARI-S3W (30).

Method	Dataset		
	Earthquake ↓	Flood ↓	Fire ↓
Stereo	2.04 ± 0.19	1.85 ± 0.03	1.34 ± 0.11
SW	1.12 ± 0.07	1.58 ± 0.02	0.55 ± 0.18
SSW	0.84 ± 0.05	1.26 ± 0.03	0.24 ± 0.18
S3W	0.88 ± 0.09	1.33 ± 0.05	0.36 ± 0.04
RI-S3W (1)	0.79 ± 0.07	1.25 ± 0.02	0.15 ± 0.06
ARI-S3W (50)	0.78 ± 0.06	1.24 ± 0.04	0.10 ± 0.04
DSSW (exp)	0.70 ± 0.09 ‡	1.22 ± 0.04 ‡	0.05 ± 0.08 ‡
DSSW (identity)	0.76 ± 0.08 ‡	1.23 ± 0.06 ‡	0.10 ± 0.13
DSSW (poly)	0.74 ± 0.05 ‡	1.23 ± 0.08 ‡	0.22 ± 0.21
DSSW (linear)	**0.69 ± 0.04** ‡	1.21 ± 0.03 ‡	0.09 ± 0.04 ‡
DSSW (nonlinear)	0.71 ± 0.06 ‡	**1.20 ± 0.03** ‡	**0.05 ± 0.05** ‡
DSSW (attention)	0.70 ± 0.08 ‡	1.21 ± 0.03 ‡	0.08 ± 0.07 ‡

Table 2: Earth datasets results. We evaluate the NLL on test data averaged over 5 training runs. We use 1 rotation for RI-S3W (1). We also use 50 rotations with a pool size of 1000 for ARI-S3W (50). The results of baselines are cited from (Tran et al. 2024). The notation "‡" indicates that DSSW variants are significantly better than the best baseline method using t-test when the significance level is 0.05.

indicate that DSSW is more suitable for fitting data on the sphere than other methods.

In addition, the density visualization using various distances on test data is shown in Appendix Section C.5. These visualizations further support that DSSW estimates a more accurate density distribution than other distances.

Sliced-Wasserstein Autoencoder

In this section, we employ the classical SWAE (Kolouri et al. 2019b) framework to evaluate the performance of various distances in generative modeling. Let α denote an encoder, and β be a decoder in this framework. The goal of SWAE is to force the encoded embedding to follow a prior distribution in the latent space. For this experiment, we utilize a mixture of vMF distributions with 10 components on $\mathbb{S}^2$ as the prior distribution denoted as z. The training objective of the revised SWAE is

$$\min_{\alpha,\beta} \mathbb{E}_{x\sim\mu}\left[c\left(x,\alpha\left(\beta\left(x\right)\right)\right)\right]+\eta\cdot DSSW_2^2\left(\alpha_{\#}\mu,z\right), \quad (14)$$

where μ is the unknown data distribution that we only have access to samples, c refers to the reconstruction loss that is implemented as the standard Binary Cross Entropy (BCE) loss, and η denotes the regularization coefficient.

The results on the CIFAR10 (Krizhevsky and Hinton 2009) benchmark for SWAE with vMF prior are shown in Table 3. Our DSSW outperforms other methods in terms of $\log W_2$ and NLL. In terms of the reconstruction loss BCE, SW is better than other competitors. In a word, all the regularization priors are superior to the original supervised autoencoder (AE). This phenomenon indicates that a prior on the hypersphere can enhance the training performance of SWAE (Davidson et al. 2018; Xu and Durrett 2018).

Moreover, we also demonstrate the details about the network architectures, training configurations, additional results and the latent space visualization on the MNIST (Lecun et al. 1998) benchmark in Appendix Section C.6.

Self-Supervised Representation Learning

It has been proven that the contrastive objective can be decomposed into an alignment loss, which forces positive representations coming from the same image to be similar, and a uniformity loss, which preserves the maximum information of the feature distribution and thus avoids collapsing representations (Wang and Isola 2020; Zhang et al. 2024; Zheng et al. 2023). Similar to SSW (Bonet et al. 2023) and S3W (Tran et al. 2024), we replace the Gaussian kernel uniformity loss with our DSSW. Therefore, the overall pretrained objective of the self-supervised learning network can be defined as:

$$\begin{aligned}\mathcal{L}_{\text{DSSW}-\text{SSL}} &= \frac{1}{n}\sum_{i=1}^{n}\|z_i^A - z_i^B\|_2^2 \\ &+\frac{\eta}{2}\left(DSSW_2^2(z^A,\nu)+DSSW_2^2(z^B,\nu)\right),\end{aligned} \quad (15)$$

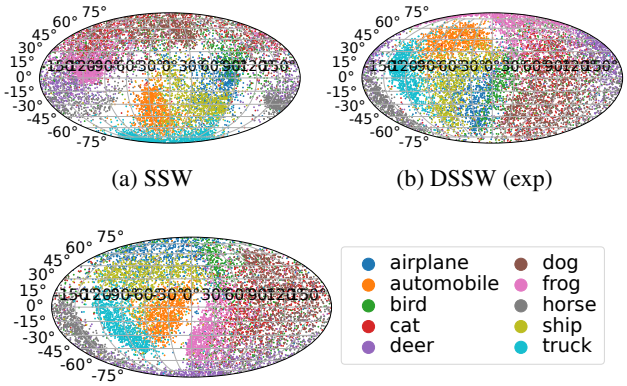
where z^A and z^B denote the hyperspherical projections of the representations from the network for two augmented versions of the same images, $\nu = \text{Unif}\left(\mathbb{S}^{d-1}\right)$ is the uniform

Method	η	$\log W_2 \downarrow$	NLL $\downarrow$	BCE $\downarrow$
Supervised AE	1	-0.1313 ± 0.8101	0.0031 ± 0.0126	0.6329 ± 0.0021
SSW	10	-3.2368 ± 0.1836	0.0008 ± 0.0019	0.6323 ± 0.0017
SW	0.001	-3.2537 ± 0.1116	-0.0004 ± 0.0030	**0.6307 ± 0.0005**
S3W	0.001	-3.0541 ± 0.2244	-0.0000 ± 0.0027	0.6310 ± 0.0012
RI-S3W (5)	0.001	-2.8317 ± 0.8168	0.0004 ± 0.0034	0.6330 ± 0.0049
ARI-S3W (5)	0.001	-3.1639 ± 0.1744	0.0002 ± 0.0028	0.6315 ± 0.0016
DSSW (exp)	10	-3.3607 ± 0.1349 ‡	-0.0012 ± 0.0051 ‡	0.6321 ± 0.0006
DSSW (identity)	10	-3.4203 ± 0.0402 ‡	-0.0011 ± 0.0025 ‡	0.6318 ± 0.0006
DSSW (poly)	10	-3.3454 ± 0.1117 ‡	**-0.0018 ± 0.0021** ‡	0.6330 ± 0.0024
DSSW (linear)	10	-3.4027 ± 0.0480 ‡	-0.0002 ± 0.0038 ‡	0.6314 ± 0.0004
DSSW (nonlinear)	10	-3.4078 ± 0.0753 ‡	-0.0014 ± 0.0045 ‡	0.6323 ± 0.0013
DSSW (attention)	10	**-3.4242 ± 0.0337** ‡	-0.0002 ± 0.0044 ‡	0.6324 ± 0.0016

Table 3: CIFAR10 results for SWAE with vMF prior. We evaluate the latent regularization loss ($\log W_2$ and NLL), along with the BCE loss on the test data for $d = 3$. We use 5 rotations for RI-S3W (5). We also use 5 rotations with the pool size of 1000 for ARI-S3W (5). The notation "‡" indicates that DSSW variants are significantly better than the best baseline method using a t-test when the significance level is 0.05.

d	Method	E $\uparrow$	$\mathbb{S}^9$ $\uparrow$
10	Supervised	92.38	91.77
	hypersphere	79.76	74.57
	SimCLR	79.69	72.78
	SW-SSL (η=1, L=200)	74.45	68.35
	SSW-SSL (η=20, L=200)	70.46	64.52
	S3W-SSL (η=0.5, L=200)	78.54	73.84
	RI-S3W(5)-SSL (η=0.5, L=200)	79.97	74.27
	ARI-S3W(5)-SSL (η=0.5, L=200)	79.92	75.07
	DSSW-SSL (exp, η=100, L=200)	80.10	76.30
	DSSW-SSL (identity, η=105, L=200)	79.65	75.14
	DSSW-SSL (poly, η=105, L=200)	78.46	73.69
	DSSW-SSL (linear, η=100, L=200)	**80.15**	**76.87**
	DSSW-SSL (nonlinear, η=100, L=200)	79.73	76.61
	DSSW-SSL (attention, η=100, L=200)	79.66	75.98

Table 4: Linear evaluation on CIFAR10 for $d = 10$. E denotes the encoder output. We use 5 rotations for RI-S3W (5). We also use 5 rotations with the pool size of 1000 for ARI-S3W (5). The results are compared with methods cited from (Tran et al. 2024).



(a) SSW (b) DSSW (exp)

(c) DSSW (linear)

Figure 3: Projected features on $\mathbb{S}^2$ for CIFAR10

distribution on the hypersphere and $\eta > 0$ acts as a regularization coefficient to balance the alignment loss and the uniformity loss in Eq. (15).

We conduct experiments on CIFAR10 (Krizhevsky and Hinton 2009) by adopting ResNet18 (He et al. 2016) as the encoder. The results of the standard linear classifier evaluation for $d = 10$ are reported in Table 4. The results indicate that our DSSW (exp) and DSSW (linear) are superior to other self-supervised methods in terms of the accuracy for the encoder output and the projected features on $\mathbb{S}^9$. As expected, the supervised method achieves the highest precision due to the additional supervised signals.

Implementation details and the additional results of the standard linear classifier evaluation on $\mathbb{S}^2$ are reported in Appendix Section C.7. Furthermore, we visualize the projected features on $\mathbb{S}^2$ in Figure 3, the visualization plot demonstrates that the cluster result of the projected features on $\mathbb{S}^2$ obtained by our DSSW is better than other methods.

Conclusion

In this work, we propose a novel approach termed DSSW distance that emphasizes the importance of the projection direction. Our proposed DSSW employ a non-parametric projected energy function to learn a discriminative projection direction, considering both efficiency and accuracy. Our proposed DSSW has been proven to be effective and competitive in various applications. However, the issue of reducing the additional computing overhead caused by training the parametric neural network remains to be addressed in future research. Additionally, the idea of learning discriminative projection direction from the specific data distribution can also be extended to other non-Euclidean Sliced-Wasserstein methods.

Acknowledgments

This work was supported by the National Science Fund of China under Grant Nos. U24A20330, 62361166670, 62276135 and 62176124.

References

An, X.; Zhao, L.; Gong, C.; Wang, N.; Wang, D.; and Yang, J. 2024. SHaRPose: Sparse High-Resolution Representation for Human Pose Estimation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(2): 691–699.

Bendokat, T.; Zimmermann, R.; and Absil, P.-A. 2024. A Grassmann manifold handbook: Basic geometry and computational aspects. *Advances in Computational Mathematics*, 50(1): 6.

Besombes, C.; Pannekoucke, O.; Lapeyre, C.; Sanderson, B.; and Thual, O. 2021. Producing realistic climate data with generative adversarial networks. *Nonlinear Processes in Geophysics*, 28(3): 347–370.

Bonet, C.; Berg, P.; Courty, N.; Septier, F.; Drumetz, L.; and Pham, M. T. 2023. Spherical Sliced-Wasserstein. In *The Eleventh International Conference on Learning Representations*.

Bonneel, N.; Rabin, J.; Peyré, G.; and Pfister, H. 2015. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51: 22–45.

Brakenridge, G. 2017. Global active archive of large flood events.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In III, H. D.; and Singh, A., eds., *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 1597–1607. PMLR.

Cui, L.; Qi, X.; Wen, C.; Lei, N.; Li, X.; Zhang, M.; and Gu, X. 2019. Spherical optimal transportation. *Computer-Aided Design*, 115: 181–193.

Cuturi, M. 2013. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In Burges, C.; Bottou, L.; Welling, M.; Ghahramani, Z.; and Weinberger, K., eds., *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Davidson, T. R.; Falorsi, L.; De Cao, N.; Kipf, T.; and Tomczak, J. M. 2018. Hyperspherical variational auto-encoders. In *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, 856–865. Association For Uncertainty in Artificial Intelligence (AUAI).

Di Marzio, M.; Panzera, A.; and Taylor, C. C. 2014. Nonparametric regression for spherical data. *Journal of the American Statistical Association*, 109(506): 748–763.

Dinh, L.; Sohl-Dickstein, J.; and Bengio, S. 2017. Density estimation using Real NVP. In *International Conference on Learning Representations*.

Dominitz, A.; and Tannenbaum, A. 2010. Texture mapping via optimal mass transport. *IEEE transactions on visualization and computer graphics*, 16(3): 419–433.

Doucet, A.; de Freitas, N.; and Gordon, N. J., eds. 2001. *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. Springer. ISBN 978-1-4419-2887-0.

EOSDIS. 2020. Land, atmosphere near real-time capability for eos (lance) system operated by nasa's earth science data and information system (esdis).

Gemici, M. C.; Rezende, D.; and Mohamed, S. 2016. Normalizing flows on riemannian manifolds. *arXiv preprint arXiv:1611.02304*.

Groemer, H. 1998. On a spherical integral transformation and sections of star bodies. *Monatshefte für Mathematik*, 126(2): 117–124.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hu, X.; Zhong, B.; Liang, Q.; Zhang, S.; Li, N.; and Li, X. 2024. Towards Modalities Correlation for RGB-T Tracking. *IEEE Transactions on Circuits and Systems for Video Technology*.

Hu, X.; Zhong, B.; Liang, Q.; Zhang, S.; Li, N.; Li, X.; and Ji, R. 2023. Transformer tracking via frequency fusion. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(2): 1020–1031.

Jammalamadaka, S. R.; and Sengupta, A. 2001. *Topics in circular statistics*, volume 5. world scientific.

Kolouri, S.; Nadjahi, K.; Simsekli, U.; Badeau, R.; and Rohde, G. 2019a. Generalized Sliced Wasserstein Distances. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Kolouri, S.; Pope, P. E.; Martin, C. E.; and Rohde, G. K. 2019b. Sliced Wasserstein Auto-Encoders. In *International Conference on Learning Representations*.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario.

Lecun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. *stat*, 1050(9).

Mathieu, E.; and Nickel, M. 2020. Riemannian continuous normalizing flows. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 2503–2515. Curran Associates, Inc.

Nadjahi, K.; Durmus, A.; Chizat, L.; Kolouri, S.; Shahrampour, S.; and Simsekli, U. 2020. Statistical and Topological Properties of Sliced Probability Divergences. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 20802–20812. Curran Associates, Inc.

Nadjahi, K.; Durmus, A.; Simsekli, U.; and Badeau, R. 2019. Asymptotic Guarantees for Learning Generative Models with the Sliced-Wasserstein Distance. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Nguyen, K.; and Ho, N. 2024. Energy-based sliced wasserstein distance. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems*, volume 36, 18046–18075. Curran Associates, Inc.

Nguyen, K.; Ho, N.; Pham, T.; and Bui, H. 2021. Distributional Sliced-Wasserstein and Applications to Generative Modeling. In *International Conference on Learning Representations*.

Nietert, S.; Goldfeld, Z.; Sadhu, R.; and Kato, K. 2022. Statistical, Robustness, and Computational Guarantees for Sliced Wasserstein Distances. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 28179–28193. Curran Associates, Inc.

NOAA. 2022. Ncei/wds global significant earthquake database.

Ohana, R.; Nadjahi, K.; Rakotomamonjy, A.; and Ralaivola, L. 2023. Shedding a PAC-Bayesian Light on Adaptive Sliced-Wasserstein Distances. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, 26451–26473.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Peyré, G.; Cuturi, M.; et al. 2019a. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6): 355–607.

Peyré, G.; Cuturi, M.; et al. 2019b. Computational Optimal Transport: With Applications to Data Science. *Foundations and Trends® in Machine Learning*, 11(5-6): 355–607.

Quellmalz, M.; Beinert, R.; and Steidl, G. 2023. Sliced optimal transport on the sphere. *Inverse Problems*, 39(10): 105005.

Rezende, D. J.; Papamakarios, G.; Racaniere, S.; Albergo, M.; Kanwar, G.; Shanahan, P.; and Cranmer, K. 2020. Normalizing Flows on Tori and Spheres. In III, H. D.; and Singh, A., eds., *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 8083–8092. PMLR.

Rubin, B. 2018. The vertical slice transform in spherical tomography. *arXiv preprint arXiv:1807.07689*.

Tran, H.; Bai, Y.; Kothapalli, A.; Shahbazi, A.; Liu, X.; Martin, R. D.; and Kolouri, S. 2024. Stereographic Spherical Sliced Wasserstein Distances. In *International Conference on Machine Learning*.

Vrba, J.; and Robinson, S. E. 2001. Signal processing in magnetoencephalography. *Methods*, 25(2): 249–271.

Wang, T.; and Isola, P. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International conference on machine learning*, 9929–9939. PMLR.

Xu, J.; and Durrett, G. 2018. Spherical Latent Spaces for Stable Variational Autoencoders. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Yi, M.; and Liu, S. 2023. Sliced Wasserstein variational inference. In Khan, E.; and Gonen, M., eds., *Proceedings of The 14th Asian Conference on Machine Learning*, volume 189 of *Proceedings of Machine Learning Research*, 1213–1228. PMLR.

Zhang, H.; Chen, S.; Luo, L.; and Yang, J. 2024. Few-shot learning with long-tailed labels. *Pattern Recognition*, 156: 110806.

Zheng, Y.; Zhan, J.; He, S.; Dong, J.; and Du, Y. 2023. Curricular contrastive regularization for physics-aware single image dehazing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5785–5794.