Efficient Variants of Wasserstein Distance in Hyperbolic Space via Space-Filling Curve Projection

Tao Li, Cheng Meng, Hongteng Xu, and Jun Zhu

***Abstract*—Hyperbolic spaces have been considered pervasively for embedding hierarchically structured data in the recent decade. However, there is a lack of studies focusing on efficient distance metrics for comparing probability distributions in hyperbolic spaces. To bridge the gap, we propose a novel metric called the hyperbolic space-filling curve projection Wasserstein (SFW) distance. The idea is to first project two probability distributions onto a space-filling curve to obtain a closed-form coupling between them and then calculate the transport distance between these two distributions in the hyperbolic space accordingly. Theoretically, we show the SFW distance is a proper metric and is well-defined for probability measures with bounded supports. Statistical convergence rates for the proposed estimator are provided as well. Moreover, we propose two variants of the SFW distance based on geodesic and horospherical projections, respectively, to combat the curse-of-dimensionality. Empirical results on synthetic and real-world data indicate that the SFW distance can effectively serve as a surrogate of the popular Wasserstein distance with low complexity.**

***Index Terms*—Hilbert curve, hyperbolic space, optimal transport, Wasserstein distance.**

I. Introduction

HYPERBOLIC spaces have gained significant attention in the recent decade as a pervasive tool for embedding hierarchically structured data [1], including graphs [2], [3], words [4], [5], and images [6], [7]. Hyperbolic embedding has been successfully applied in various tasks, such as text generation [8], image segmentation [9], drug embedding [10], molecular generation [11], recommendation system [12], and reinforcement learning [13]. Consequently, researchers have extended the conventional tools used for Euclidean space to also encompass hyperbolic spaces, including the generalization of Gaussian distributions [14], [15], hyperbolic neural networks [16], [17], and hyperbolic variational auto-encoders [18]. These extensions have enormous potential to enhance the capabilities of machine learning models.

Distribution comparison is a fundamental task in many machine learning tasks including clustering [19], [20], classification [21], [22], [23], generative modeling [24], [25], and domain adaptation [26], [27]. Wasserstein distance as a metric for comparing distributions has recently attracted considerable attention in the machine learning community, and it has shown great potential in many challenging problems [28], [29], [30]. However, the computation cost of Wasserstein distance is expensive, and super-cubical with respect to the number of samples of each distribution [31]. To alleviate the computational burden, many surrogates of the Wasserstein distance have been proposed, e.g., Sinkhorn divergence [32], the sliced-Wasserstein (SW) distance [33], the generalized SW (GSW) distance [34], Hilbert curve-based Wasserstein distance [35], [36], among others. However, when it comes to hyperbolic spaces, these surrogates are not inherently well-suited since they are originally defined using Euclidean distances and projections.

One exception is the recently proposed hyperbolic SW (HSW) distance [37], which utilized geodesic and horospherical projections to extend the SW distance to hyperbolic spaces. Despite the computational efficiency, HSW distance may not serve as a decent surrogate of the Wasserstein distance. Take two hyperbolic distributions in Fig. 1(a) as an example. We fix the source distribution (in blue) while shifting the central component of the target distribution (in red) along the geodesic. In particular, the source distribution μ is uniform on the ring $(\sqrt{1.16}, 0.4\cos(\theta), 0.4\sin(\theta))$, where $\theta \in [0, 2\pi)$, and the target distribution is $\nu = 0.8\mu + 0.2\chi_{\{x_t\}}$ where $x_t = \cosh(t)x^0 + \sinh(t)v$ for direction $v = (0, 0, 1)^{\mathrm{T}}$ and $x^0 = (1, 0, 0)^{\mathrm{T}}$. χ is the indicator function. Fig. 1(b) shows the discrepancy between these two distributions under different x_t with respect to different distance measures, including the Wasserstein distance, Sinkhorn method, horospherical hyperbolic SW (HHSW) distance, geodesic hyperbolic SW (GHSW) and our proposed approach. The result indicates both HHSW distance and GHSW distance may lead to coarse approximations of the Wasserstein distance, whose tendency with respect to x_t can even be opposite to the Wasserstein distance. This observation indicates that using HSW distances as surrogates for the Wasserstein distance in hyperbolic spaces may yield suboptimal outcomes in some learning tasks. Power Voronoi

Received 28 July 2024; revised 21 January 2025; accepted 10 March 2025. This work was supported by the National Natural Science Foundation of China under Grant 92270110. *(All authors contributed equally and the order of authors' names is alphabetical.) (Corresponding author: Cheng Meng.)*

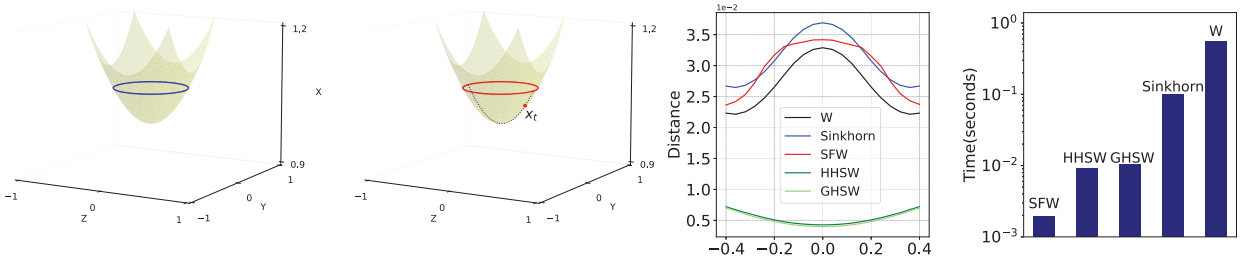
Tao Li and Jun Zhu are with the Institute of Statistics and Big Data, Renmin University of China, Beijing 100872, China (e-mail: 2019000153@ruc.edu.cn; dfsxzj@ruc.edu.cn).

Cheng Meng is with the Center for Applied Statistics, Institute of Statistics and Big Data, Renmin University of China, Beijing 100872, China (e-mail: chengmeng@ruc.edu.cn).

Hongteng Xu is with the Gaoling School of Artificial Intelligence, Renmin University of China, Beijing 100872, China, and also with Beijing Key Laboratory of Big Data Management and Analysis, Beijing 100872, China (e-mail: hongtengxu@ruc.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TNNLS.2025.3551275, provided by the authors.

Digital Object Identifier 10.1109/TNNLS.2025.3551275



(a) (b) (c)

Fig. 1. (a) Illustration of the source and the target distribution in a hyperbolic space. (b) Distance is calculated by different metrics with respect to different values of x_t. (c) Comparison between different metrics with respect to their runtime. The proposed SFW distance provides an effective and efficient surrogate of the Wasserstein distance.

diagram has been used to calculate the hyperbolic Wasserstein distance [38]. However, their work primarily addresses the distance between a continuous measure and a discrete measure, which is not the main focus of this study.

In this study, we propose a novel metric called hyperbolic space-filling curve projection Wasserstein (SFW) distance for distribution comparison in hyperbolic spaces. The idea is to first project two probability distributions onto a space-filling curve [39] to obtain a closed-form coupling between them and then calculate the transport distance between these two distributions in the hyperbolic space accordingly. Compared to geodesic and horospherical projections, the space-filling curve projection strategy offers advantages in preserving the inherent structure of the data distribution since such a projection enjoys the locality-preserving property, i.e., the locality between data points in the high-dimensional space being approximately preserved in the projected 1-D space [40], [41], [42]. Our SFW distance provides an effective and efficient surrogate of the Wasserstein, as illustrated in Fig. 1(b) and (c).

We present a comprehensive analysis of the SFW distance, demonstrating its effectiveness as a well-defined metric for probability measures with bounded supports in hyperbolic spaces. We show the computational complexity for calculating the empirical SFW distance is nearly linear in sample size. In addition, we introduce two variants of the SFW distance to address the curse-of-dimensionality. We evaluate the performance of the SFW distance and its variants on various machine learning tasks, including data classification and generative modeling, and compare them with state-of-the-art methods. Our empirical results demonstrate the superior performance of the proposed metrics in both synthetic and real-data settings.

II. Background

A. Wasserstein Distance and SW Distance

Let (M, d_M) be a metric space. Consider two probability measures $\mu, \nu \in \mathcal{P}_p(M) = \{\mu \in \mathcal{P}_p(M), \int_M d_M(x, x_0)^p d\mu(x) < \infty \text{ for any } x_0 \in M\}$. The p-Wasserstein distance [28] between μ and ν is defined as

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

where $\Pi(\mu, \nu)$ is the set of all couplings: $\Pi(\mu, \nu) = \{\gamma \in \mathcal{P}(M \times M) \quad \text{s.t. } \forall \text{ Borel set } A, B \subset M, \gamma(A \times M) = \mu(A), \gamma(M \times B) = \nu(B)\}$. If the geodesic distance in hyperbolic space is used as $d_M(x, y)$, the p-Wasserstein distance is naturally referred to as the hyperbolic p-Wasserstein distance [38].

The main bottleneck of the Wasserstein distance is its high computational complexity, making it inapplicable for large-scale data. Specifically, (1) can be solved using linear programs with a computational complexity of $O(n^3 \log(n))$ for two discrete probability measures with n observations. A fast approximate solution to the Wasserstein distance is provided by the Sinkhorn algorithm, which incorporates an entropic regularizer [43]. Recently, several efficient variants of the Sinkhorn algorithm have been proposed to enhance its performance [44], [45], [46].

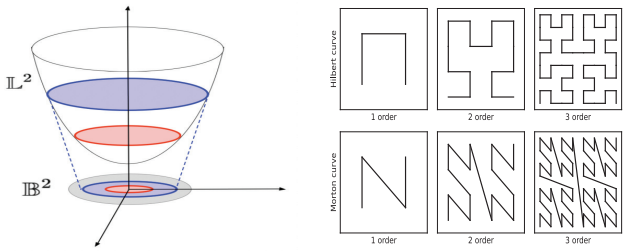
In the recent decade, SW distance has been proposed to alleviate the computational burden of Wasserstein distance in Euclidean space [33]. Note that the Wasserstein distance enjoys a closed-form solution for 1-D probability measures μ and ν in Euclidean space

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

where $F_\mu(t) = \mu((-\infty, t])$ and $F_\nu(t) = \nu((-\infty, t])$ are the cumulative distribution functions (cdfs) for μ and ν, respectively. This motivates the development of the sliced Wasserstein distance, which projects d-dimensional probability measures to 1-D space and computes the 1-D Wasserstein distance in this reduced dimension. Let $\mathbb{V}^{d,q} = \{\mathbf{U} \in \mathbb{R}^{d \times q} : \mathbf{U}^\top \mathbf{U} = I_q\}$ $(q < d)$ be the set of orthogonal matrices and $P_\mathbf{U}(x) = \mathbf{U}^\top x$ be the linear transformation for $x \in \mathbb{R}^d$. Denote $P_{\mathbf{U}\#}\mu$ as the pushforward of μ by $P_\mathbf{U}$, which corresponds to the distribution of the projected samples. For all $\mu, \nu \in \mathcal{P}_p(\mathbb{R}^d)$, the p-sliced Wasserstein distance is given by

$$\mathrm{SW}_p(\mu, \nu) = \left(\int_{\mathbf{U} \in \mathbb{V}^{d,1}} W_p^p \left(P_{\mathbf{U}\#}\mu, P_{\mathbf{U}\#}\nu \right) \mathrm{d}\sigma(\mathbf{U}) \right)^{1/p} \quad (3)$$

where σ is the uniform distribution on $\mathbb{V}^{d,1}$. Using a Monte-Carlo scheme, SW distance can be approximated in $O(Ln \log(n))$ time, where L is the number of random projections and n is the number of samples. Some recent studies develop several extensions for SW distances to other spaces, e.g., sphere and hyperbolic spaces [37], [47].



(a) (b)

Fig. 2. (a) Two hyperbolic models: Lorentz model $\mathbb{L}^2$ and Poincaré ball $\mathbb{B}^2$. (b) Discrete approximations of the Hilbert and Morton space-filling curve.

B. Hyperbolic Spaces

Hyperbolic spaces have been considered pervasively for embedding hierarchically structured data in the recent decade [1]. There are two widely used parameterizations of a hyperbolic manifold $\mathbb{H}^d$, i.e., the Poincaré ball $\mathbb{B}^d$ and the Lorentz model $\mathbb{L}^d$, as illustrated in Fig. 2(a). While these two parameterizations are known to be equivalent (isometric), each enjoys its own advantages in different applications [37]. In particular, these parameterizations of hyperbolic models yield different ambient spaces, resulting in slightly different SFW distances, see Section III-A of Supplementary Material for more discussions.

1) Lorentz Model: Lorentz model $\mathbb{L}^d \subset \mathbb{R}^{d+1}$ can be defined as $\mathbb{L}^d = \{(x_1, x_2, \ldots, x_d, x_{d+1}) \in \mathbb{R}^{d+1}, \langle x, x\rangle_{\mathbb{L}} = -1, x_1 > 0\}$, where $\forall x, y \in \mathbb{R}^{d+1}, \langle x, y\rangle_{\mathbb{L}} = -x_1 y_1 + \sum_{i=2}^{d+1} x_i y_i$. The geodesic distance, which denotes the length of the shortest path between two points in this manifold, is defined as

$$\forall x, y \in \mathbb{L}^d, d_{\mathbb{L}}(x, y) = \operatorname{arccosh}(-\langle x, y\rangle_{\mathbb{L}}). \tag{4}$$

2) Poincaré Ball: Poincaré ball $\mathbb{B}^d \subset \mathbb{R}^d$ can be defined as $\mathbb{B}^d = \{x \in \mathbb{R}^d, \|x\|_2 < 1\}$, with geodesic distance

$$\forall x, y \in \mathbb{B}^d, d_{\mathbb{B}}(x, y) = \operatorname{arccosh}\left(1 + 2\frac{\|x-y\|_2^2}{\left(1-\|x\|_2^2\right)\left(1-\|y\|_2^2\right)}\right). \tag{5}$$

The following mappings show how to switch between the Lorentz model and Poincaré ball:

$$\forall x \in \mathbb{L}^d, P_{\mathbb{L}^d \to \mathbb{B}^d}(x) = \frac{1}{1+x_1}\left(x_2, \ldots, x_d, x_{d+1}\right) \tag{6}$$

$$\forall x \in \mathbb{B}^d, P_{\mathbb{B}^d \to \mathbb{L}^d}(x) = \frac{1}{1-\|x\|_2^2}\left(1+\|x\|_2^2, 2x_1, \ldots, 2x_d\right). \tag{7}$$

C. Operations on Hyperbolic Spaces

Consider the Lorentz model $\mathbb{L}^d$. Denote $x^0 = (1, 0, \ldots, 0) \in \mathbb{L}^d$. Tangent spaces are described formally as

$$T_x\mathbb{L}^d = \left\{v \in \mathbb{R}^{d+1}, \langle v, x\rangle_{\mathbb{L}} = 0\right\}.$$

The formula of parallel transport from x to y is

$$\forall v \in T_x\mathbb{L}^d, PT_{x\to y}(v) = v + \frac{\langle y, v\rangle_{\mathbb{L}}}{1-\langle x, y\rangle_{\mathbb{L}}}(x+y). \tag{8}$$

Furthermore, the exponential map $\exp\mathrm{Map}_x\colon T_x\mathbb{L}^d \to \mathbb{L}^d$ is

$$\forall v \in T_x\mathbb{L}^d, \exp\mathrm{Map}_x(v) = \cosh\left(\|v\|_{\mathbb{L}}\right) x + \sinh\left(\|v\|_{\mathbb{L}}\right)\frac{v}{\|v\|_{\mathbb{L}}} \tag{9}$$

where $\|v\|_{\mathbb{L}} = \sqrt{-\langle v, v\rangle_{\mathbb{L}}}$.

The logarithmic maps $\log\mathrm{Map}_x\colon \mathbb{L}^d \to T_x\mathbb{L}^d$ is the inverse map of $\exp\mathrm{Map}_x$.

The exponential map $\exp\mathrm{Map}_x\colon T_x\mathbb{B}^d \to \mathbb{B}^d$ for the Poincaré model is

$$\begin{aligned}\exp\mathrm{Map}_x(v) &= \frac{A}{B}\\ A &= \lambda_x\left(\cosh\left(\lambda_x\|v\|_2\right) + \left\langle x, \frac{v}{\|v\|_2}\right\rangle \sinh\left(\lambda_x\|v\|_2\right)\right) x\\ &\quad + \frac{1}{\|v\|_2}\sinh\left(\lambda_x\|v\|_2\right) v\\ B &= 1 + \left(\lambda_x - 1\right)\cosh\left(\lambda_x\|v\|_2\right)\\ &\quad + \lambda_x\left\langle x\frac{v}{\|v\|_2}\right\rangle \sinh\left(\lambda_x\|v\|_2\right)\end{aligned} \tag{10}$$

where $\lambda_x = (2/(1-\|x\|_2^2))$. Specifically, if $x = \mathbf{0}_d$, we have

$$\exp\mathrm{Map}_{\mathbf{0}_d}(v) = \tanh\left(\|v\|_2\right)\frac{v}{\|v\|_2} \tag{11}$$

$$\log\mathrm{Map}_{\mathbf{0}_d}(y) = \tanh^{-1}\left(\|y\|_2\right)\frac{y}{\|y\|_2}. \tag{12}$$

D. Distribution on Hyperbolic Spaces

A common distribution in hyperbolic spaces is the wrapped normal distribution [14]. This distribution $x \sim \mathcal{G}(\mu, \Sigma)$ can be sampled by the following steps: first, drawing $v \sim N(0, \Sigma)$ and then transforming it into $v \in T_x\mathbb{L}^d$ by concatenating a 0 in the first coordinate; second, using parallel transport to transport v from $T_{x^0}\mathbb{L}^d$ to $T_\mu\mathbb{L}^d$; third, projecting the samples onto the manifold by the exponential map $\exp\mathrm{Map}_\mu$.

E. Optimization on Hyperbolic Spaces

Gradient descent-based methods are developed on hyperbolic space [1], [48], [49], [50], [51]. Among them, the Riemannian gradient descent is

$$\forall k > 0, x_{k+1} = \exp\mathrm{Map}_{x_k}\left(-\gamma \mathrm{grad} f\left(x_k\right)\right) \tag{13}$$

where $f\colon M \to \mathbb{R}$.

1) Lorentz Model: For $M = \mathbb{L}^d$, the Riemannian gradient is

$$\mathrm{gradf(x)} = \mathrm{Proj}_{\mathrm{x}}\left(\mathrm{J}\nabla \mathrm{f}\left(\mathrm{x}\right)\right) \tag{14}$$

where $J = \mathrm{diag}(-1, 1, \ldots, 1)$ and $\mathrm{Proj}_{\mathrm{x}}(z) = z + \langle \mathrm{x}, z\rangle_{\mathbb{L}}\mathrm{x}$.

2) Poincaré Ball: For $M = \mathbb{B}^d$, the Riemannian gradient is

$$\mathrm{grad} f(x) = \frac{\left(1-\|\theta\|_2^2\right)^2}{4}\nabla f(x). \tag{15}$$

After computing the exponential map, we could perform the Riemannian gradient descent [16].

Furthermore, the exponential map in (13) can be replaced more generally by a retraction. We could use a retraction $R_x(v) = x + v$ instead of the exponential map, and add a

projection, to constrain the value to remain within the Poincaré ball [1]

$$\text{proj}(x) = \begin{cases} \dfrac{x}{\|x\|_2} - \epsilon, & \text{if } \|x\| \geq 1 \\ x, & \text{otherwise} \end{cases} \tag{16}$$

where $\epsilon = 10^{-5}$ is a small constant. The algorithm becomes

$$x_{k+1} = \text{proj}\left(x_k - \gamma \frac{\left(1 - \|x_k\|_2^2\right)^2}{4} \nabla f(x_k)\right). \tag{17}$$

F. Space-Filling Curves

A space-filling curve is a continuous mapping C from a 1-D interval, typically $[0, 1]$, to a higher-dimensional cube $[0, 1]^d$.[1] It fills the space without gaps or overlaps, that is, $[0, 1]^d \subset C([0, 1])$ and the set $\{t \in [0, 1]^d$: set $C^{-1}(t)$has at least two elements$\}$ has Lebesgue measure zero. In other words, it is a curve that passes through every point in the space in a way that preserves the locality and continuity of the points. Due to these nice properties, space-filling curves have been widely applied in many fields, including computer graphics, image processing, scientific computing, and geographic information systems [39], [42]. Some well-known space-filling curves include the Morton curve, the Hilbert curve, the Peano curve, and the Sierpiński curve. Fig. 2(b) illustrates the discrete version of the Hilbert and Morton space-filling curves, respectively.

To be more detailed, we provide the definition of the Hilbert curve as an example [52]. For integer $m \geq 0$, we define 2^{dm} intervals

$$I_d^m(k) = \left[\frac{k}{2^{dm}}, \frac{k+1}{2^{dm}}\right], \quad k = 0, \ldots, 2^{dm} - 1 \tag{18}$$

and let $\mathcal{I}_d^m = \{I_d^m(k) \mid k < 2^{dm}\}$.

For $\kappa = (k_1, \ldots, k_d)$ with $k_j \in \{0, 1, \ldots, 2^m - 1\}$, we define 2^{dm} subcubes of $[0, 1]^d$

$$E_d^m(\kappa) = \prod_{j=1}^{d} \left[\frac{k_j}{2^m}, \frac{k_j + 1}{2^m}\right] \tag{19}$$

and let $\mathcal{E}_d^m = \{E_d^m(\kappa) \mid \kappa \in \mathcal{K}_d^m\}$ where $\mathcal{K}_d^m = \{0, 1, \ldots, 2^m - 1\}^d$ is the set of indices κ.

Then, there is a sequence of mappings $H_m : \mathcal{I}_d^m \to \mathcal{E}_d^m$ satisfying the following three properties.

1) For $k \neq k'$, $H_m(I_d^m(k)) \neq H_m(I_d^m(k'))$.
2) The two subcubes $H_m(I_d^m(k))$ and $H_m(I_d^m(k+1))$ have one $(d-1)$-dimensional face in common.
3) If we split $I_d^m(k)$ into the 2^d successive subintervals $I_d^{m+1}(k_\ell), k_\ell = 2^d k + \ell, \ell = 0, \ldots, 2^d - 1$, then the $H_{m+1}(I_d^{m+1}(k_\ell))$ are subcubes whose union is $H_m(I_d^m(k))$.

The Hilbert curve is defined by $H(x) = \lim_{m \to \infty} H_m(x)$.

Space-filling curves enjoy the so-called locality-preserving property [41], [42], [52]. In particular, as stated in [41], "One of the most desired properties from such linear mappings is clustering, which means the locality between objects in the multidimensional space being preserved in the linear space." Mathematically, such locality-preserving property can be stated as, for any $x, y \in [0, 1]$, one has

$$\|C(x) - C(y)\|_2 \leq 2\sqrt{d+3}|x - y|^{1/(\kappa d)} \tag{20}$$

where κ is the locality-preserving coefficient of the space-filling curve. Existing literature shows that the Hilbert, Peano, and Sierpiński space-filling curves satisfy the inequality (20) with $\kappa = 1$, while the Morton space-filling curve performs slightly worse such that its coefficient $\kappa = \log_2 3$ [42], [52].

III. Proposed Method

A. Quantile Function for Probability Measures in Hyperbolic Space via Space-Filling Curves

In this study, we focus on Borel probability measures in a hyperbolic space with bounded supports and denote the set of such measures as $\mathcal{P}_\infty(\mathbb{H}^d)$. The quantile function, also known as the inverse cdf, is crucial for deriving the closed-form solution for the Wasserstein distance, as shown in (2). However, extending the quantile function to hyperbolic spaces is a challenging task. We propose to utilize space-filling curves to achieve the goal.

Definition 1: Denote the support of the probability measure $\mu \in \mathcal{P}_\infty(\mathbb{H}^d)$ as Ω_μ. Let space-filling curve $C_\mu : [0, 1] \to \widetilde{\Omega}_\mu$, where $\widetilde{\Omega}_\mu$ is the smallest hyper-rectangle that covers Ω_μ. Denote $\mathcal{K}$ as a dense set in $[0, 1]$, satisfying $C_\mu([0, s])$ is Borel measurable set for any $s \in \mathcal{K}$. Let $g_\mu(t) = \inf_{s \in \mathcal{K},\ s \geq t} \mu(C_\mu([0, s]) \cap \mathbb{H}^d)$, and $g_\mu^{-1}(t) = \inf_{s \in [0,1],\ g_\mu(s) > t} s$. The quantile function for μ in hyperbolic space $\mathbb{H}^d$ via space-filling curves C is defined as

$$Q_{\mu,C}(t) = C_\mu\left(g_\mu^{-1}(t)\right).$$

Remark 1: Existence of $\mathcal{K}$ is easy to prove. One could take $\mathcal{K} = \{m_1/2^{m_2} : m_1, m_2 \in \mathbb{N}, m_1 \leq 2^{m_2}\}$ for the Hilbert curve and $\mathcal{K} = \{m_1/3^{m_2} : m_1, m_2 \in \mathbb{N}, m_1 \leq 3^{m_2}\}$ for the Peano curve [39], [42]. Different choices of $\mathcal{K}$ won't affect the definition since $g_\mu^{-1}(t)$ is increasing and $\mathcal{K}$ is dense.

Remark 2: Similar to inequality (20), it is easy to show that such a locally preserving property still holds in hyperbolic spaces. We refer to supplementary material for more details.

B. Hyperbolic SFW Distance

Inspired by the closed form of 1-D Wasserstein distance (2), we develop the hyperbolic SFW distance defined as follows.

Definition 2: (Hyperbolic SFW Distance): Consider two probability measures $\mu, \nu \in \mathcal{P}_\infty(\mathbb{H}^d)$. Denote the quantile function for μ, ν in hyperbolic space $\mathbb{H}^d$ via space-filling curves C as $Q_{\mu,C}, Q_{\nu,C}$, respectively. For $p \in \mathbb{Z}_+$, the p-order hyperbolic SFW distance is defined as

$$\mathcal{SFW}_{p,C}(\mu, \nu) = \left(\int_0^1 d_{\mathbb{H}}\left(Q_{\mu,C}(t), Q_{\nu,C}(t)\right)^p \mathrm{d}t\right)^{1/p}.$$

From the above definition, the fundamental idea of SFW distance is first projecting two probability distributions along the space-filling curve to obtain an efficient and effective coupling between them and then calculating the corresponding transport distance between two distributions in the hyperbolic

[1]The range $[0, 1]^d$ could be extended to the hyper-rectangle $\prod_{i=1}^d [a_i, b_i]$ simply by linear transformation.

space according to the coupling. The following theoretical results show that SFW distance is a proper metric and serves as an upper bound of the p-Wasserstein distance. All the proofs are relegated to supplementary material.

Theorem 1: $\mathcal{SFW}_{p,C}$ is a well-defined metric in $\mathcal{P}_\infty(\mathbb{H}^d)$, and $W_p(\mu,\nu) \le \mathcal{SFW}_{p,C}(\mu,\nu)$ for any $\mu,\nu \in \mathcal{P}_\infty(\mathbb{H}^d)$.

1) Topological Properties of the SFW Distance: Theorem 1 tells us that SFW distance induces a stronger topology compared to Wasserstein distance. More precisely, the sequence of probability measures $\{\mu_n\}$ always converges in Wasserstein distance as $n \to \infty$ if it converges in SFW distance, i.e., $\mathcal{SFW}_{p,C}(\mu_n,\mu) \to 0 \Rightarrow W(\mu_n,\mu) \to 0$. Furthermore, we conduct a comparison between SFW distance and the total variation (TV) distance regarding their induced topology.

Theorem 2: Let $\widetilde{\Omega}_\mu, \widetilde{\Omega}_{\mu_n}$ be the smallest hyper-rectangle that covers the supports of the probability measure $\mu, \mu_n \in \mathcal{P}_\infty(\mathbb{H}^d)$, respectively. When $\widetilde{\Omega}_{\mu_n} = \widetilde{\Omega}_\mu$ for all n's, we have $TV(\mu_n,\mu) \to 0 \Rightarrow \mathcal{SFW}_{p,C}(\mu_n,\mu) \to 0$.

2) Comparison With Existing Methods: The proposed SFW distance enjoys several critical advantages over the Wasserstein distance and hyperbolic SW distance. First, SFW distance has less computational complexity. Second, SFW distance is a well-defined metric and provides a decent transport plan between the input probability measures as a byproduct, while hyperbolic SW distance may not satisfy these. Last but not least, SFW distance serves as a decent surrogate of the Wasserstein distance. SFW distance computes distance in original hyperbolic spaces rather than the projected 1-D space. Fig. 1 intuitively shows the difference between these two strategies. The reason for the opposite trend observed in hyperbolic SW distance, in contrast to the Wasserstein distance and SFW distance, is that hyperbolic SW distance computes distances using transformed 1-D data points. This transformation may break the distance structure of the original distributions.

C. Statistical Property

Let $\{x_i\}_{i=1}^n \sim \mu$, whose empirical measure is defined by $\mu_n = (1/n)\sum_{i=1}^n \delta_{x_i}$. The following theorem provides an upper bound for the statistical convergence rate of the empirical SFW distance.

Theorem 3: Assume that probability measures $\mu,\nu \in \mathcal{P}_\infty(\mathbb{H}^d)$. Let $\{x_i\}_{i=1}^n$ and $\{y_i\}_{i=1}^n$ be two i.i.d. samples, which are generated from probability measures μ and ν, respectively. Let $\{x_{(i)^*}\}_{i=1}^n$ and $\{y_{(i)^*}\}_{i=1}^n$ be the sorted samples along the space-filling curve C_μ and C_ν, respectively. Then, we have almost surely

$$\left(\frac{1}{n}\sum_{i=1}^n d_{\mathbb{H}}(x_{(i)^*}, y_{(i)^*})^p\right)^{\frac{1}{p}} \to \mathcal{SFW}_{p,C}(\mu,\nu).$$

Furthermore, we have

$$\left|\mathbb{E}\left(\frac{1}{n}\sum_{i=1}^n d_{\mathbb{H}}(x_{(i)^*}, y_{(i)^*})^p\right)^{\frac{1}{p}} - \mathcal{SFW}_{p,C}(\mu,\nu)\right| \lesssim O\left(n^{-\frac{1}{2\max\{p,\kappa d'\}}}\right)$$

where κ is the locality-preserving coefficient of the space-filling curve, $d' = d$ if $\mathbb{H}^d = \mathbb{B}^d$ and $d' = d+1$ if $\mathbb{H}^d = \mathbb{L}^d$.

Algorithm 1 Computation of SFW Distance

1: **Input:** $(\{x_i\}_{i=1}^m, \boldsymbol{a})$, $(\{y_j\}_{j=1}^n, \boldsymbol{b})$, k-order discrete space-filling curve C_k
2: Map $\{x_i\}_{i=1}^m$ to $\{x'_i\}_{i=1}^m$, $\{y_j\}_{j=1}^n$ to $\{y'_j\}_{j=1}^n$, through space-filling curve $C_k O((m+n)dk)$
3: Calculate the optimal transport plan $\mathbf{P}$ between $(\{x'_i\}_{i=1}^m, \boldsymbol{a})$ and $(\{y'_j\}_{j=1}^n, \boldsymbol{b})$ using sorting and the North-West corner rule. Let $\mathcal{S} := \{(i,j)|P_{ij} \neq 0\} O(m\log(m) + n\log(n))$
4: **Output:** $\mathbf{P}$, $\mathcal{SFW}_{p,C_k} = \left(\sum_{(i,j)\in\mathcal{S}} d_{\mathbb{H}}(x_i,y_j)^p P_{ij}\right)^{1/p}$

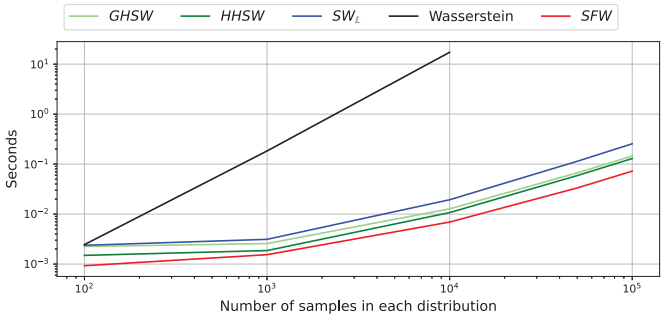


Fig. 3. GPU time versus different n when $d = 3$.

Theorem 3 tells us the empirical SFW distance is to compute the distance between two space-filling curve sorted samples. In addition, we know that the convergence rate of the empirical SFW distance is no more than $O(n^{-1/(2p)} + n^{-1/(2\kappa d')})$, which is slightly larger than the convergence rate of the Wasserstein distance, i.e., $O(n^{-1/(2p)} + n^{-1/d'})$ provided by [53].

D. Numerical Implementation

In practice, the empirical SFW distance is to compute the distance between two samples sorted by the space-filling curve. We utilize recursively space-filling curve sorting algorithm [39], [54], [55], [56]. The complexity of sorting based on the k-order discrete space-filling curve for n points in d-dimensional space is $O(ndk)$ [54], [55], [57]. Algorithm 1 demonstrates the process of computing SFW distance. Solving the optimal transport problem in Step 3 requires $O(n\log n + m\log m)$ time. As suggested by [39], we select k that of the order $O(\log(n))$ in practice. Hence, the overall computational complexity of SWF distance is at the order of $O(n\log(n)d)$ when $m = O(n)$. We compare the runtime in Fig. 3.

We analyze the effect of k in the proposed SFW distance using a synthetic example. We generate two samples of size N from the bounded distribution in $\mathbb{H}^d$ and calculate the SFW distance between these two samples. All the experiments are replicated 100 times. The left panel of Fig. 4(b) shows the average SFW distances versus different N, and the right panel shows the average CPU time for generating the k-order discrete version of the space-filling curve when $d = 2$. The results for $d = 10$ are shown in Fig. 4(c). From these two figures, we observe that the SFW distance is not sensitive to the choice of k as long as k is not too small. In addition, we observe that the computational cost for generating the k-order discrete

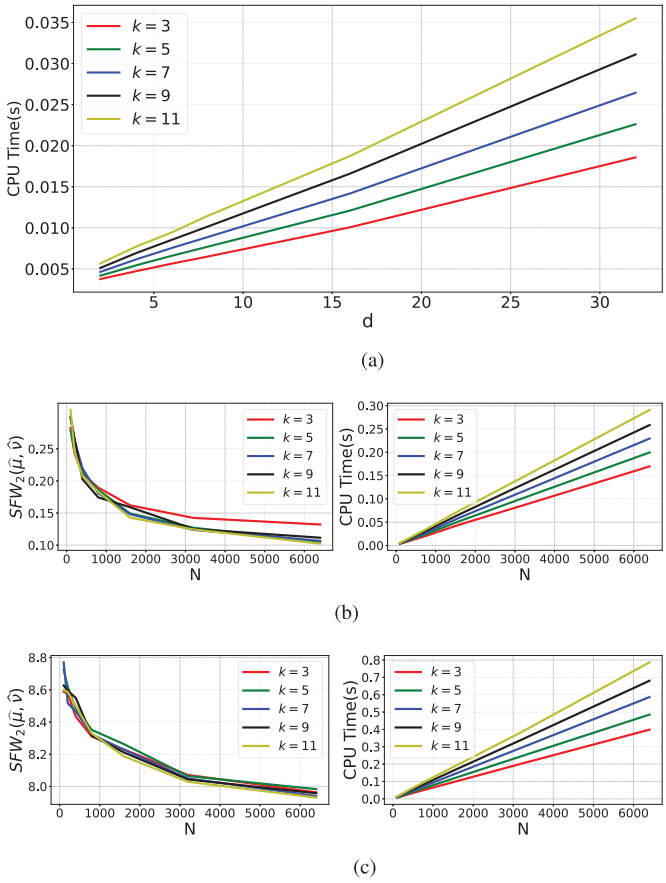


Fig. 4. (a) CPU time for generating the k-order discrete space-filling curve versus d when $N = 100$. (b) Left: SFW distance versus N when $d = 2$. Right: CPU time for generating the k-order discrete space-filling curve versus N when $d = 2$. (c) Left: SFW distance versus N when $d = 10$. Right: CPU time for generating the k-order discrete space-filling curve versus N when $d = 10$.

version of the space-filling curve is linear to n. Furthermore, the results in Fig. 4(a) indicate that the computational cost for generating the k-order discrete version of the space-filling curve is linear to d.

IV. Variants of the Hyperbolic SFW Distance

The theoretical results presented in Section III suggest that, similar to the Wasserstein distance, the proposed SFW distance might also encounter the curse-of-dimensionality. Motivated by the projection-robust Wasserstein distance [58], [59], we propose two natural variants of the SFW distance to mitigate this limitation. There are two main projection methods, geodesic projection, and horospherical projection, that map points in hyperbolic space to subspaces while preserving information [37], [60]. Fig. 5 gives a toy example to show how these two projections operate. We develop the closed-form solution of the geodesic projection and provide an algorithm in Algorithm 2. Results about the horospherical projection are provided in Supplementary Material.

Since the Lorentz model and Poincaré ball could be converted interchangeably (6) and (7), we only consider the Lorentz model $\mathbb{L}^d$ for simplicity. Subspace of $\mathbb{L}^d$ corresponding to $\mathbf{U} \in \mathbb{V}_{d,q}$ is $\mathbb{L}^d \cap \mathcal{U}$, where $\mathcal{U}$ is the subspace spanned

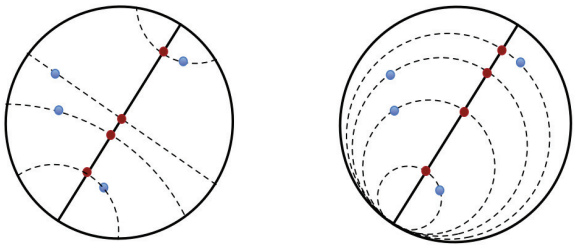


Fig. 5. (a) Geodesic projection and (b) horospherical projection of (blue) points on a geodesic (black line) in $\mathbb{B}^2$. Projected points on the geodesic are in red.

Algorithm 2 Computation of SFW$^{\text{IPR}}$ Distance for Lorentz Model (Geodesic Projection)

1: **Input:** $(\{x_i\}_{i=1}^m, \boldsymbol{a})$, $(\{y_j\}_{j=1}^n, \boldsymbol{b})$, k-order discrete space-filling curve C_k, number of projections L, projected dimension q

2: **For** $l = 1$ **to** L **do**

a) Draw a random orthogonal matrix $\mathbf{U} \in \mathbb{R}^{d\times q}$, and let $\mathbf{U}_* = \begin{pmatrix} 1 & \mathbf{0}_q^{\mathrm{T}} \\ \mathbf{0}_{d-1} & \mathbf{U} \end{pmatrix}$

b) Let $x_i' = \mathbf{U}_*^\top x_i / \sqrt{-\langle \mathbf{U}_*^\top x_i, \mathbf{U}_*^\top x_i \rangle_{\mathbb{L}}}$ and $y_i' = \mathbf{U}_*^\top y_i / \sqrt{-\langle \mathbf{U}_*^\top y_i, \mathbf{U}_*^\top y_i \rangle_{\mathbb{L}}}$

c) $D_l =$ Algorithm 1$[(\{x_i'\}_{i=1}^m, \boldsymbol{a}), (\{y_j'\}_{j=1}^n, \boldsymbol{b}), C_k]$

3: **Output:** $\mathcal{SFW}_{p,q,C_k}^{\mathcal{IPR}} = \left(\frac{1}{L}\sum_{l=1}^L D_l^p\right)^{1/p}$.

by matrix $\mathbf{U}_* = \begin{pmatrix} 1 & \mathbf{0}_q^{\mathrm{T}} \\ \mathbf{0}_{d-1} & \mathbf{U} \end{pmatrix} \in \mathbb{R}^{(d+1)\times(q+1)}$. For $x \in \mathbb{L}^d$, geodesic projection of x corresponding to $\mathbf{U}$ is $\text{Proj}_{\mathbf{U}}(x) = \mathbf{U}_*^{\mathrm{T}} \arg\min_{y\in\mathbb{L}^d\cap\mathcal{U}} d_{\mathbb{L}}(x, y)$.

Lemma 1: Geodesic projection of $x \in \mathbb{L}^d$ corresponding to $\boldsymbol{U}$ is $Proj_{\mathbf{U}}(x) = (\mathbf{U}_*^T x/(-\langle \mathbf{U}_*^T x, \mathbf{U}_*^T x\rangle_{\mathbb{L}})^{1/2})$.

Definition 3: (Integral Projection Robust Hyperbolic SFW Distance): Consider two probability measures $\mu, \nu \in \mathcal{P}_\infty(\mathbb{H}^d)$. For $p \in \mathbb{Z}_+$, the p-order q-dimensional($q \le d$) integral projection robust hyperbolic space-filling curve projection Wasserstein (SFW^{IPR}) distance is defined as

$$\mathcal{SFW}_{p,q,C}^{\mathcal{IPR}}(\mu,\nu) = \left(\int_{\mathbf{U}\in\mathbb{V}_{d,q}} \mathcal{SFW}_{p,C}\left(Proj_{\mathbf{U}\#}\mu, Proj_{\mathbf{U}\#}\nu\right)^p \mathrm{d}\sigma(\mathbf{U})\right)^{1/p}$$

where $Proj_{\mathbf{U}}: \mathbb{H}^d \to \mathbb{H}^q$ is either geodesic projection or horospherical projection.

Theorem 4: $\mathcal{SFW}_{p,q,C}^{\mathcal{IPR}}$ is a pseudo-distance in $\mathcal{P}_\infty(\mathbb{H}^d)$. Specifically, when $q = d$, $\mathcal{SFW}_{p,q,C}^{\mathcal{IPR}}$ is a well-defined metric in $\mathcal{P}_\infty(\mathbb{H}^d)$.

Based on Theorem 3, we could prove that an upper bound for the convergence rate of SFW$^{\text{IPR}}$ is $O(n^{-(1/2\max\{p,\kappa q'\})})$, where $q' = q$ for the Poincaré ball and $q' = q + 1$ for the Lorentz model. The detailed theoretical results are relegated to Supplementary Material.

We consider a synthetic example to demonstrate the empirical convergence of the proposed distances. We generate two

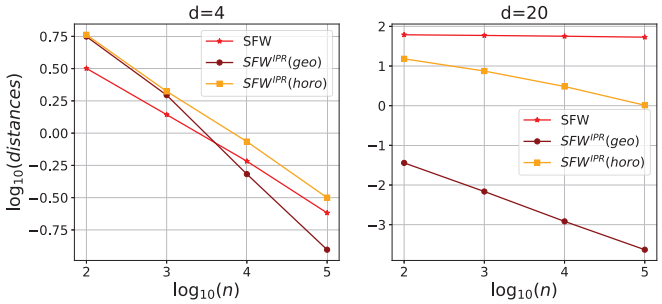


Fig. 6. Comparison for convergence rates of different dimensions. Left: $d = 4$. Right: $d = 20$. Each curve represents the average distance with respect to 100 replications.

samples of size n from the standard d-dimensional wrapped normal distributions in the Lorentz model, and we calculate the distances between these two samples with respect to different distance metrics. Fig. 6 shows the average distances with respect to 100 replications versus n for $d = 4$ (left) and $d = 20$ (right), respectively. We observe that when $d = 20$, the SFW distance converges slowly as expected, while SFW^{IPR}(geo) and SFW^{IPR}(horo) converge much faster, which shows that SFW^{IPR} combats curse-of-dimensionality.

V. Applications in Machine Learning Tasks

To validate the feasibility and efficiency of our SFW distance and its variants, we compare Wasserstein distance (with the geodesic distance as cost), SW distance [33] (denote $\text{SW}_{\mathbb{B}}$, $\text{SW}_{\mathbb{L}}$ for SW in Poincaré ball and Lorentz model, respectively), and hyperbolic SW [37] (denote GHSW, HHSW for geodesic, HHSW distance, respectively). To demonstrate the advantages of hyperbolic spaces over Euclidean space, we also consider the distribution comparison methods in Euclidean space, including maximum mean discrepancy (MMD), SW distance, Hilbert curve projection (HCP) distance [36]. For SFW distance and its variants, we set $p = 2$ and $q = 2$, take the Morton curve, and use the Poincaré ball. All experiments are implemented by a single RTX 3090 GPU.

A. Comparison of Different Discrepancies

1) Evolution: Following the experiment in [37], we compare the evolution of each method between wrapped normal distributions $\mathcal{G}(\mu, \Sigma)$ [14], where one is centered and the other moves along a geodesic. We plot the evolution of the distances between $\mathcal{G}(x_0, I_3)$ and $\mathcal{G}(x_t, I_3)$ where $x_0 = (1, 0, 0, 0)^{\mathrm{T}}$, $x_t = \cosh(t)x_0 + \sinh(t)v$ for $t \in [-10, 10]$ and direction $v = (0, 1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3})^{\mathrm{T}}$. As shown in Fig. 7, SFW has almost the same trend as Wasserstein distance. $\text{SW}_{\mathbb{L}}$ explodes when the two distributions are getting far from each other, which might bring numerical instabilities. GHSW and $\text{SW}_{\mathbb{B}}$ have small derivatives when the two distributions are far from each other, while HHSW has small derivatives when the two distributions are close to each other, which might lead to slower convergence for gradient-based learning tasks; see Fig. 8.

TABLE I

Document Classification Accuracy and GPU Time Tested on BBCSPORT (Each Document in BBCSPORT Has Nearly 100 Words)

Methods	BBCSPORT	TWITTER	AMAZON	CLASSIC	*GPU Time(s)*
HCP(Euclidean)	$85.1_{\pm 3.5}$	$70.1_{\pm 2.1}$	$83.1_{\pm 0.8}$	$80.0_{\pm 1.6}$	$\mathbf{57.5}_{\pm 0.8}$
HHSW(Poincaré)	$75.9_{\pm 3.4}$	$69.4_{\pm 1.6}$	$80.0_{\pm 1.3}$	$87.8_{\pm 0.9}$	$925.8_{\pm 4.7}$
GHSW(Poincaré)	$87.4_{\pm 2.8}$	$\mathbf{70.7}_{\pm 1.5}$	$86.9_{\pm 1.0}$	$87.3_{\pm 0.5}$	$1259.8_{\pm 9.7}$
SFW(Poincaré)	$\mathbf{88.5}_{\pm 2.7}$	$70.0_{\pm 1.8}$	$\mathbf{87.4}_{\pm 0.6}$	$\mathbf{88.2}_{\pm 0.8}$	$\mathbf{119.3}_{\pm 3.8}$
W(Poincaré)	$\mathbf{96.5}_{\pm 1.0}$	$\mathbf{71.2}_{\pm 1.3}$	$\mathbf{92.8}_{\pm 0.9}$	$\mathbf{96.9}_{\pm 0.4}$	$906.1_{\pm 2.2}$

2) Gradient Flow: We consider the problem $\min_\mu W_2(\mu, \nu)$, where ν is a fixed target distribution, and μ is the source distribution initialized as a wrapped normal distribution μ_0 and updated iteratively via $\partial_t \mu_t = -\nabla W_2(\mu_t, \nu)$ by a Riemannian gradient descent [48]. Following the experiment in [37], we consider four different distributions for the target ν, and approximate the Wasserstein distance W_2 by $\text{SW}_{\mathbb{B}}$, $\text{SW}_{\mathbb{L}}$, GHSW, HHSW, and SFW. The experiments are replicated ten times for each method with the same learning rate, and we record the averaged two-Wasserstein distance between μ_t and ν at each iteration in Fig. 8. We can find that applying SFW always accelerates the learning process and yields superior results. When the target is close to the border, $\text{SW}_{\mathbb{L}}$ suffers from numerical instabilities, and $\text{SW}_{\mathbb{B}}$ and GHSW are slower to converge. And, when the target possesses more weights near the origin, HHSW has a slower convergence.

3) Document Classification: Document classification can be achieved by comparing the Wasserstein distance between two documents' word embedding sets, as the Word Mover distance [21] does.

Recent studies have shown that the hyperbolic space could be preferred over the Euclidean space for embedding words [1], [4], [5]. SFW distance could provide an efficient surrogate of the Wasserstein distance in this problem, as demonstrated by the following experiment. We consider four datasets, e.g., BBCSPORT, TWITTER, AMAZON, and CLASSIC dataset, in which each document is represented as a set of 100-D word embeddings derived by the pretrained Euclidean and hyperbolic word embedding model [5]. We randomly split the dataset into 80% for training and 20% for testing, used the K-NN algorithm ($k = 20$) based on different metrics, and reported the averaged results in 100 trials. Table I shows that SFW performs better than HCP, which indicates that using hyperbolic word embeddings improves classification accuracy. What's more, for hyperbolic distribution metrics, SFW outperforms HHSW. While GHSW performs similar to SFW, it takes over ten times longer. Although Wasserstein distance with geodesic groundcost has shown significant improvement, its computation time is about eight times that of SFW distance and this ratio will improve rapidly for longer documents.

B. Hyperbolic WAE

We design new members of Wasserstein autoencoder (WAE) [29] in hyperbolic space. Assume f is encoder and g is decoder, p_Z a prior distribution

$$\mathcal{L}(f, g) = \int c(x, g(f(x)))\mathrm{d}\mu(x) + \lambda D\left(f_{\#}\mu, p_Z\right)$$

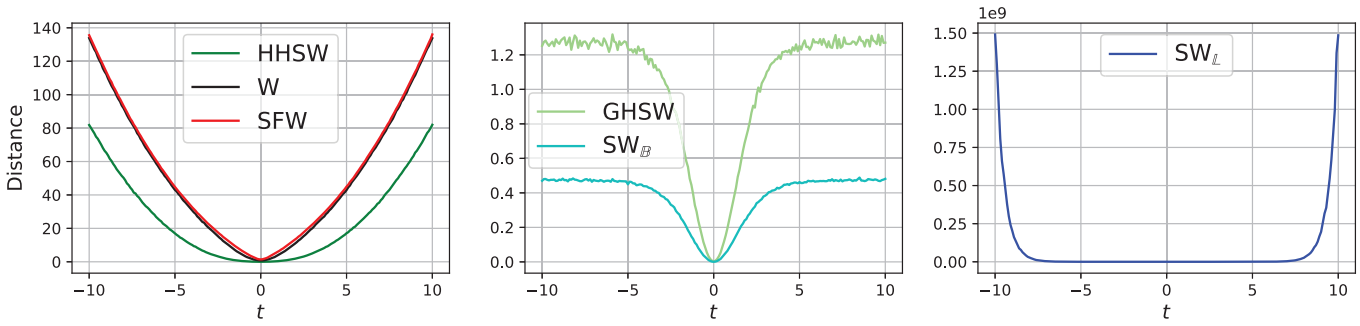


Fig. 7. Comparison of distances between two wrapped normal distributions with dimension $d = 3$.

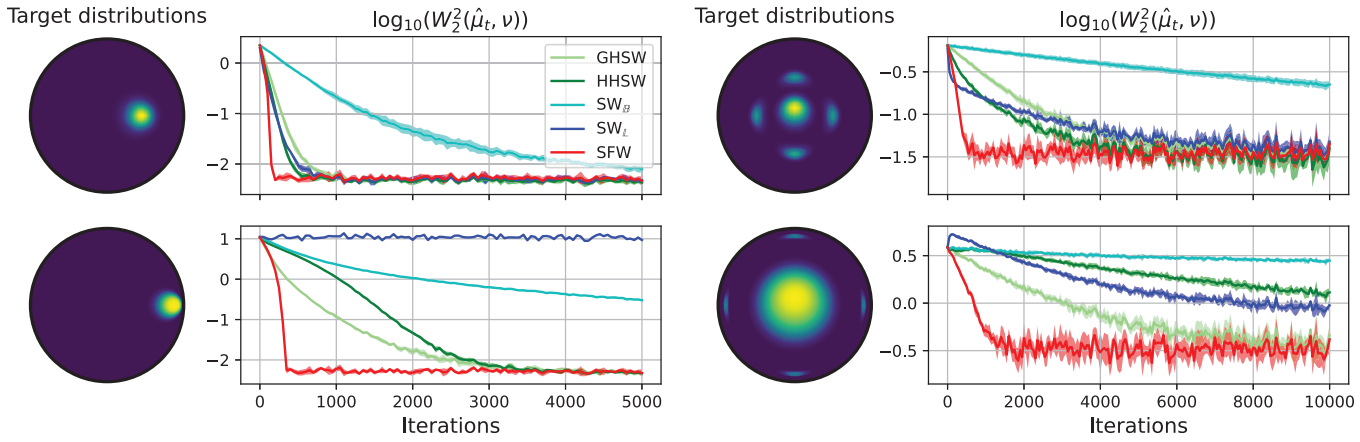


Fig. 8. Log-10 Wasserstein distance between a target and the gradient flow of SFW, GHSW, HHSW, $SW_{\mathbb{B}}$, and $SW_{\mathbb{L}}$.

where μ is the distribution of the data, λ is a hyperparameter, and D is distribution distance. In particular, when training autoencoders, we leverage SFW, SFW^{IPR}(geo), and SFW^{IPR}(horo) to penalize the distance between the latent prior distribution and the expected posterior distribution, which leads to three different generative models, denoted as SFWAE, SFW^{IPR}(geo)AE, and SFW^{IPR}(horo)AE. We test these three models in hierarchical data embedding and image generation tasks, comparing them with the original Euclidean space-based WAE using MMD [29] and the well-known sliced WAE (SWAE) [61]. We use the wrapped normal distribution as latent prior distribution for hyperbolic WAE and normal distribution for Euclidean WAE.

1) WAE for BDP: First, we test the capability of SFWAE to embed tree-structured data with low distortions. Theoretically, any tree-structured data can be embedded in the hyperbolic space with arbitrarily low distortion [62]. We train SFWAE to encode data generated from a branching diffusion process (BDP) that explicitly incorporates hierarchical structure to $\mathbb{B}^2$ (for the sake of visualization). We follow the generation of the synthetic BDP in [18].

We train models on noisy vector representations and, hence, have no access to the true hierarchical representation. We report the correlation between the Euclidean distance and the embedding distance in Table II, and the embedding results achieved by our methods are shown in Fig. 9. All models

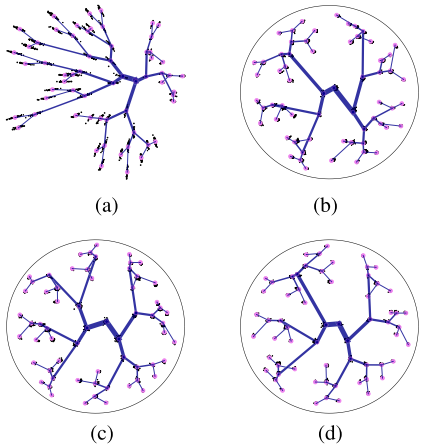


Fig. 9. Embeddings learned from different WAE models. Embeddings of noisy observations are represented by small black points, and violet points represent embeddings of true nodes. Blue lines represent the true hierarchy. (a) WAE. (b) SFWAE. (c) SFW^{IPR}(horo)AE. (d) SFW^{IPR}(geo)AE.

somewhat learn the hierarchical structure, yet WAE's latent representation is the most distorted.

2) WAE for MNIST and CelebA: Second, we test the feasibility of SFW^{IPR}(geo)AE and SFW^{IPR}(horo)AE in cases with high-dimensional latent space. Hyperbolic image embeddings provide a better alternative [7]. Here, we consider datasets MNIST and CelebA. The dimension of the latent distribution

TABLE II
NUMERICAL RESULTS OF DIFFERENT WAE MODELS ON THREE DATASETS

	Dataset	WAE(Euclidean)	SWAE(Euclidean)	HHSWAE	GHSWAE	SFWAE	SFW$^{\text{IPR}}$(horo)AE	SFW$^{\text{IPR}}$(geo)AE
Correlation(%) ↑	Synthetic BDP	$72.40_{\pm 4.75}$	$75.11_{\pm 1.18}$	$75.20_{\pm 2.60}$	$75.55_{\pm 2.13}$	$\mathbf{78.12}_{\pm 1.78}$	$77.02_{\pm 1.74}$	$75.85_{\pm 3.07}$
	Dataset	WAE(Euclidean)	SWAE(Euclidean)	HHSWAE	GHSWAE	SFWAE	SFW$^{\text{IPR}}$(horo)AE	SFW$^{\text{IPR}}$(geo)AE
FID ↓	MNIST	$19.02_{\pm 0.12}$	$15.25_{\pm 0.27}$	$17.73_{\pm 0.09}$	$17.48_{\pm 0.14}$	/	$14.85_{\pm 0.28}$	$\mathbf{13.69}_{\pm 0.08}$
	CelebA	$63.71_{\pm 0.94}$	$88.93_{\pm 1.39}$	$71.10_{\pm 0.72}$	$67.40_{\pm 0.98}$	/	$57.11_{\pm 1.14}$	$\mathbf{53.61}_{\pm 0.92}$

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

Fig. 10. Performance of our generators on MNIST generation. (a) WAE. (b) SFW$^{\text{IPR}}$(horo)AE. (c) SFW$^{\text{IPR}}$(geo)AE. (d) FID.

TABLE III
TEST CLASSIFICATION ACCURACY ON IMAGE CLASSIFICATION

	Dimension	[63]	SW$_{\mathbb{L}}$	SW$_{\mathbb{B}}$	W	HHSW	GHSW	SFW	SFW$^{\text{IPR}}$(horo)	SFW$^{\text{IPR}}$(geo)
CIFAR10	2	$90.64_{\pm 0.06}$	$91.13_{\pm 0.14}$	$91.84_{\pm 0.31}$	$91.67_{\pm 0.18}$	$91.28_{\pm 0.26}$	$91.39_{\pm 0.23}$	$91.84_{\pm 0.07}$	$\mathbf{92.01}_{\pm 0.23}$	$91.70_{\pm 0.13}$
	4	$90.59_{\pm 0.11}$	$91.74_{\pm 0.12}$	$91.68_{\pm 0.10}$	$91.98_{\pm 0.05}$	$91.98_{\pm 0.05}$	$91.66_{\pm 0.27}$	$\mathbf{92.34}_{\pm 0.17}$	$92.20_{\pm 0.10}$	$92.18_{\pm 0.14}$
	Dimension	[63]	SW$_{\mathbb{L}}$	SW$_{\mathbb{B}}$	W	HHSW	GHSW	SFW	SFW$^{\text{IPR}}$(horo)	SFW$^{\text{IPR}}$(geo)
CIFAR100	10	$59.19_{\pm 0.39}$	$62.30_{\pm 0.23}$	$60.36_{\pm 1.26}$	$58.82_{\pm 1.66}$	$62.80_{\pm 0.09}$	$61.45_{\pm 0.41}$	/	$63.78_{\pm 0.12}$	$\mathbf{64.29}_{\pm 0.15}$

is 10 for MNIST and 64 for CelebA. The autoencoding architecture of MNIST is similar to that in [61], and the autoencoding architecture of CelebA is similar to that in [29]. We add an exponential map after the last layer of encoders and a logarithmic map before the first layer of decoders to let the encoded samples lie on the hyperbolic space.

We compare the proposed methods with the Fréchet inception distance (FID) [64] between all testing samples. Table II lists the main differences between SFW$^{\text{IPR}}$(geo)AE, SFW$^{\text{IPR}}$(horo)AE and other baselines. We omitted the results of SFW for high-dimensional cases (MNIST and CelebA) due to the curse-of-dimensionality and instead focused on the results of SFW$^{\text{IPR}}$. Among these autoencoders, our methods have lower FID scores and converge faster. Image generation results achieved by our methods are shown in Fig. 10.

3) Image Classification on CIFAR10 and CIFAR100: Following the experiments in [37], we consider image classification on datasets CIFAR10 and CIFAR100 [65]. Let $\{(x_i, y_i)_{i=1}^n\}$ be a training set where $x_i \in \mathbb{R}^d$ and $y_i \in \{1, \ldots, C\}$ is the corresponding label. Researchers perform classification on the Poincaré ball [63]. They assign a prototype $p_c \in S^{d_z-1}$ to each class $c \in \{1, \ldots, C\}$ and learn an embedding on the hyperbolic space using a neural network f_θ followed by the exponential map. To be more precise, they consider minimizing the following loss:

$$\ell(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(B_{p_{y_i}}(z_i) - \lambda \log\left(1 - \|z_i\|_2^2\right)\right) \tag{21}$$

where $z_i = \exp\text{Map}_{\mathbf{0}_{d_z}}(f_\theta(x_i))$. The first term is the Busemann function, which will pull the representations of x_i closer to the prototype assigned to the class y_i. The second term, which could be decisive in improving accuracy, penalizes representation if it is far from the origin. The classification of the input is $\arg\max_c \langle z_i/\|z_i\|_2, p_c\rangle$.

Researchers propose to replace the second term with a global prior on the distribution of the representations [37]. More precisely, they add a discrepancy between the distribution z_i and a mixture of C wrapped normal distributions where the centers are $\{\alpha p_j\}_{j=1}^C$, $\alpha \in (0, 1)$. We choose HHSW, GHSW, SFW, SFW$^{\text{IPR}}$(geo) and SFW$^{\text{IPR}}$(horo) as the discrepancy. We also consider the original penalty [63] and a similar hyperspherical prototype method [66]. Similarly, we have omitted the results of SFW for CIFAR100 due to the curse-of-dimensionality.

Following the setting in [37], we use a Resnet-32 backbone with the exponential map at the last layer, optimize it with Adam, a learning rate of $5e^{-4}$, weight decay of $5e^{-5}$, batch size of 100, and train it for 1110 epochs with learning rate decay of 10 after 1000 and 1100 epochs. All parameters are the same as [37], and the only difference is that we take $\lambda = 0.1$

for SFW. In Table III, we report the classification accuracy on the test set. Our proposed methods outperform other methods for all the different dimensions.

VI. Conclusion

We propose a novel metric, called the hyperbolic SFW distance, which can measure the distance between two probability distributions in hyperbolic spaces with low complexity. Theoretically, we show the SFW distance is a proper metric and is well-defined for probability measures with bounded supports. Statistical and topological properties are provided as well. Moreover, we propose two variants of the SFW distance based on geodesic and horospherical projections, respectively, to combat the curse-of-dimensionality.

However, the SFW distance still suffers from some limitations. Similar to HSW distance, the SFW distance sacrifices the invariance to isometric transformations to reduce the computation cost. Besides, SFW distance could not quantify the discrepancy between two measures with different masses. We left these directions for our future work. Furthermore, we plan to extend space-filling curve projection-based distance to more general manifolds.

References

[1] M. Nickel and D. Kiela, "Poincaré embeddings for learning hierarchical representations," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–10.

[2] I. Balazevic, C. Allen, and T. Hospedales, "Multi-relational poincaré graph embeddings," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.

[3] B. P. Chamberlain, J. R. Clough, and M. P. Deisenroth, "Neural embeddings of graphs in hyperbolic space," in *Proc. CoRR*, May 2017, pp. 1–7.

[4] B. Dhingra, C. J. Shallue, M. Norouzi, A. M. Dai, and G. E. Dahl, "Embedding text in hyperbolic spaces," 2018, *arXiv:1806.04313*.

[5] A. Tifrea, G. Becigneul, and O.-E. Ganea, "Poincare glove: Hyperbolic word embeddings," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–24.

[6] S. Liu, J. Chen, L. Pan, C.-W. Ngo, T.-S. Chua, and Y.-G. Jiang, "Hyperbolic visual embedding learning for zero-shot recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9273–9281.

[7] V. Khrulkov, L. Mirvakhabova, E. Ustinova, I. Oseledets, and V. Lempitsky, "Hyperbolic image embeddings," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6418–6428.

[8] S. Dai, Z. Gan, Y. Cheng, C. Tao, L. Carin, and J. Liu, "APo-VAE: Text generation in hyperbolic space," 2020, *arXiv:2005.00054*.

[9] M. G. Atigh, J. Schoep, E. Acar, N. Van Noord, and P. Mettes, "Hyperbolic image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4443–4452.

[10] K. Yu, S. Visweswaran, and K. Batmanghelich, "Semi-supervised hierarchical drug embedding in hyperbolic space," *J. Chem. Inf. Model.*, vol. 60, no. 12, pp. 5647–5657, Dec. 2020.

[11] E. Qu and D. Zou, "Hyperbolic neural networks for molecular generation," 2022, *arXiv:2201.12825*.

[12] B. Paul Chamberlain, S. R. Hardwick, D. R. Wardrope, F. Dzogang, F. Daolio, and S. Vargas, "Scalable hyperbolic recommender systems," 2019, *arXiv:1902.08648*.

[13] E. Cetin, B. Chamberlain, M. M. Bronstein, and J. J. Hunt, "Hyperbolic deep reinforcement learning," in *Proc. Deep Reinforcement Learn. Workshop NeurIPS*, Jan. 2022, pp. 1–15.

[14] Y. Nagano, S. Yamaguchi, Y. Fujita, and M. Koyama, "A wrapped normal distribution on hyperbolic space for gradient-based learning," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2019, pp. 4693–4702.

[15] S. Cho, J. Lee, J. Park, and D. Kim, "A rotated hyperbolic wrapped normal distribution for hierarchical representation learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, Jan. 2022, pp. 17831–17843.

[16] O.-E. Ganea, G. Bécigneul, and Tnn. Hofmann, "Hyperbolic neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, Jan. 2018, pp. 5345–5355.

[17] R. Shimizu, Y. Mukuta, and T. Harada, "Hyperbolic neural Networks++," 2020, *arXiv:2006.08210*.

[18] E. Mathieu, C. Le Lan, C. J. Maddison, R. Tomioka, and Y. W. Teh, "Continuous hierarchical representations with poincaré variational auto-encoders," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 12565–12576.

[19] J. Ye, P. Wu, J. Z. Wang, and J. Li, "Fast discrete distribution clustering using Wasserstein barycenter with sparse support," *IEEE Trans. Signal Process.*, vol. 65, no. 9, pp. 2317–2332, May 2017.

[20] N. Ho, X. Nguyen, M. Yurochkin, H. H. Bui, V. Huynh, and D. Phung, "Multilevel clustering via Wasserstein means," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1501–1509.

[21] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, "From word embeddings to document distances," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 957–966.

[22] G. Huang, C. Guo, M. J. Kusner, Y. Sun, F. Sha, and K. Q. Weinberger, "Supervised word mover's distance," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 4862–4870.

[23] A. Rakotomamonjy, A. Traoré, M. Berar, R. Flamary, and N. Courty, "Distance measure machines," 2018, *arXiv:1803.00250*.

[24] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[25] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2014, pp. 1–14.

[26] J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Wasserstein distance guided representation learning for domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, Apr. 2018, pp. 4058–4065.

[27] C.-Y. Lee, T. Batra, M. H. Baig, and D. Ulbricht, "Sliced Wasserstein discrepancy for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10285–10295.

[28] C. Villani, *Optimal Transport: Old and New*, vol. 338. Cham, Switzerland: Springer, 2009.

[29] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schölkopf, "Wasserstein auto-encoders," in *Proc. 6th Int. Conf. Learn. Represent.*, Feb. 2018, pp. 1–16.

[30] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 214–223.

[31] O. Pele and M. Werman, "Fast and robust Earth mover's distances," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 460–467.

[32] J. Feydy, T. Séjourné, F.-X. Vialard, S.-I. Amari, A. Trouve, and G. Peyre, "Interpolating between optimal transport and MMD using sinkhorn divergences," in *Proc. 22nd Int. Conf. Artif. Intell. Statist.*, vol. 89, 2019, pp. 2681–2690.

[33] N. Bonneel, J. Rabin, G. Peyré, and H. Pfister, "Sliced and radon Wasserstein barycenters of measures," *J. Math. Imag. Vis.*, vol. 51, no. 1, pp. 22–45, Jan. 2015.

[34] S. Kolouri, K. Nadjahi, U. Şimşekli, R. Badeau, and G. K. Rohde, "Generalized sliced Wasserstein distances," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2019, pp. 261–272.

[35] E. Bernton, P. E. Jacob, M. Gerber, and C. P. Robert, "Approximate Bayesian computation with the Wasserstein distance," *J. Roy. Stat. Soc. B, Stat. Methodol.*, vol. 81, no. 2, pp. 235–269, Apr. 2019.

[36] T. Li, C. Meng, H. Xu, and J. Yu, "Hilbert curve projection distance for distribution comparison," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 7, pp. 4993–5007, Jul. 2024.

[37] C. Bonet, L. Chapel, L. Drumetz, and N. Courty, "Hyperbolic sliced-Wasserstein via geodesic and horospherical projections," in *Proc. Topological, Algebr. Geometric Learn. Workshops*, Jan. 2022, pp. 334–370.

[38] J. Shi, W. Zhang, and Y. Wang, "Shape analysis with hyperbolic Wasserstein distance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5051–5061.

[39] M. Bader, *Space-filling Curves: An Introduction With Applications in Scientific Computing*. Cham, Switzerland: Springer, 2012.

[40] D. J. Abel and D. M. Mark, "A comparative analysis of some two-dimensional orderings," *Int. J. Geographical Inf. Syst.*, vol. 4, no. 1, pp. 21–31, Jan. 1990.

[41] B. Moon, H. V. Jagadish, C. Faloutsos, and J. H. Saltz, "Analysis of the clustering properties of the Hilbert space-filling curve," *IEEE Trans. Knowl. Data Eng.*, vol. 13, no. 1, pp. 124–141, Jan./Feb. 2001.

[42] G. Zumbusch, *Parallel Multilevel Methods: Adaptive Mesh Refinement and Loadbalancing*. Cham, Switzerland: Springer, 2012.

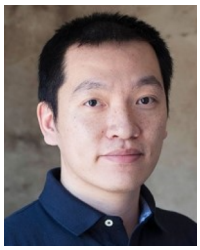
[43] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2013, pp. 2292–2300.
[44] J. M. Altschuler, J. Weed, and P. Rigollet, "Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Jan. 2017, pp. 1964–1974.
[45] T. Lin, N. Ho, and M. I. Jordan, "On the efficiency of entropic regularized algorithms for optimal transport," *J. Mach. Learn. Res.*, vol. 23, no. 137, pp. 1–42, 2022.
[46] M. Z. Alaya, M. Bérar, G. Gasso, and A. Rakotomamonjy, "Screening Sinkhorn algorithm for regularized optimal transport," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, Jan. 2019, pp. 12169–12179.
[47] C. Bonet, P. Berg, N. Courty, F. Septier, L. Drumetz, and M.-T. Pham, "Spherical sliced-Wasserstein," 2022, *arXiv:2206.08780*.
[48] N. Boumal, *An Introduction to Optimization on Smooth Manifolds*. Cambridge, U.K.: Cambridge Univ. Press, 2023.
[49] B. Wilson and M. Leimeister, "Gradient descent in hyperbolic space," 2018, *arXiv:1805.08207*.
[50] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ, USA: Princeton Univ. Press, 2008.
[51] S. Bonnabel, "Stochastic gradient descent on Riemannian manifolds," *IEEE Trans. Autom. Control*, vol. 58, no. 9, pp. 2217–2229, Sep. 2013.
[52] Z. He and A. B. Owen, "Extensible grids: Uniform sampling on a space filling curve," *J. Roy. Stat. Soc. B, Stat. Methodol.*, vol. 78, no. 4, pp. 917–931, Sep. 2016.
[53] V. M. Panaretos and Y. Zemel, "Statistical aspects of Wasserstein distances," *Annu. Rev. Statist. Appl.*, vol. 6, pp. 405–431, Mar. 2019.
[54] A. Tanaka, "Study on a fast ordering of high dimensional data to spatial index," Ph.D. dissertation, Kyushu Inst. Technol., Kitakyushu, Japan, 2001.
[55] Y. Imamura, T. Shinohara, K. Hirata, and T. Kuboyama, "Fast Hilbert sort algorithm without using Hilbert indices," in *Proc. Int. Conf. Similarity Search Appl.*Cham, Switzerland: Springer, Jan. 2016, pp. 259–267.
[56] A. Fabri and S. Pion, "CGAL: The computational geometry algorithms library," in *Proc. 17th ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, 2009, pp. 538–539.
[57] C. H. Hamilton and A. Rau-Chaplin, "Compact Hilbert indices: Space-filling curves for domains with unequal side lengths," *Inf. Process. Lett.*, vol. 105, no. 5, pp. 155–163, Feb. 2008.
[58] T. Lin, C. Fan, N. Ho, M. Cuturi, and M. I. Jordan, "Projection robust Wasserstein distance and Riemannian optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, Jan. 2020, pp. 9383–9397.
[59] T. Lin, Z. Zheng, E. Chen, M. Cuturi, and M. I. Jordan, "On projection robust optimal transport: Sample complexity and model misspecification," in *Proc. Int. Conf. Artif. Intell. Statist.*, Jan. 2020, pp. 262–270.
[60] I. Chami, A. Gu, D. T. Nguyen, and C. Ré, "HoroPCA: Hyperbolic dimensionality reduction via horospherical projections," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2021, pp. 1419–1429.
[61] S. Kolouri, P. E. Pope, C. E. Martin, and G. K. Rohde, "Sliced Wasserstein auto-encoders," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–19.
[62] R. Sarkar, "Low distortion Delaunay embedding of trees in hyperbolic plane," in *Proc. 19th Int. Symp. Graph Drawing (GD)*, Eindhoven, The Netherlands. New York, NY, USA: Springer, Sep. 2011, pp. 355–366.
[63] M. G. Atigh, M. Keller-Ressel, and P. Mettes, "Hyperbolic Busemann learning with ideal prototypes," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, Jan. 2021, pp. 103–115.
[64] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Jan. 2017, pp. 6626–6637.
[65] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf
[66] P. Mettes, E. Van der Pol, and C. G. M. Snoek, "Hyperspherical prototype networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, Jan. 2019, pp. 1487–1497.

Tao Li received the B.S. degree in mathematics from Nanjing University, Nanjing, China, in 2019. He is currently pursuing the Ph.D. degree with the Institute of Statistics and Big Data, Renmin University of China, Beijing, China.

His research interests include optimal transport problems, generative models, sufficient dimension reduction, and variable selection.

Cheng Meng received the Ph.D. degree from the Department of Statistics, University of Georgia, Athens, GA, USA, in 2020.

He is currently an Assistant Professor (Tenure-Track) at the Institute of Statistics and Big Data, Renmin University of China, Beijing, China. His research interests include numerical linear algebra, optimal transport problems, sufficient dimension reduction, nonparametric statistics, and machine learning.



Hongteng Xu received the Ph.D. degree from the School of Electrical and Computer Engineering, Georgia Institute of Technology (Georgia Tech), Atlanta, GA, USA, in 2017.

From 2018 to 2020, he was a Senior Research Scientist at Infinia ML, Inc., Durham, NC, USA. In the same time period, he is a Visiting Faculty Member at the Department of Electrical and Computer Engineering, Duke University, Durham. He is currently an Associate Professor (Tenure-Track) at the Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China. His research interests include machine learning and its applications, especially optimal transport theory, sequential data modeling and analysis, deep learning techniques, and their applications in computer vision and data mining.

Jun Zhu received the B.S. degree in statistics from Southeast University, Nanjing, China, in 2022. He is currently pursuing the Ph.D. degree with the Institute of Statistics and Big Data, Renmin University of China, Beijing, China.

His research interests include optimal transport problems, sufficient dimension reduction, and time series analysis.