Stereographic Spherical Sliced Wasserstein Distances

Huy Tran* , Yikun Bai*, Abihith Kothapalli*, Ashkan Shahbazi, Xinran Liu, Rocio Diaz Martin[†], Soheil Kolouri

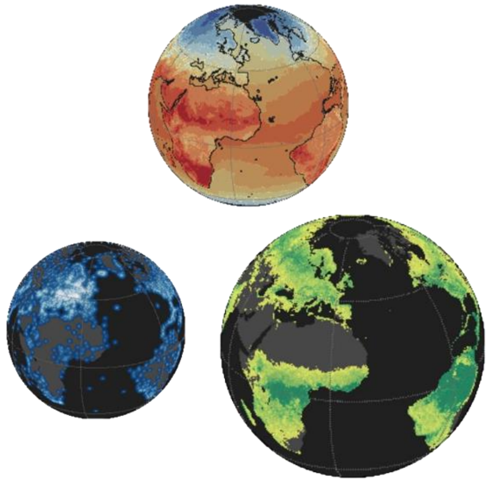
Department of Computer Science, Vanderbilt University
(†) Department of Mathematics, Vanderbilt University
(∗) Equal contribution

ICML International Conference On Machine Learning

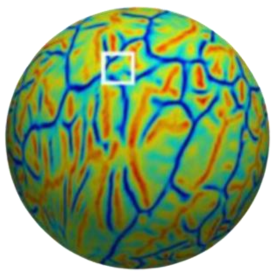
VANDERBILT UNIVERSITY

@huytransformer

Some applications of interests

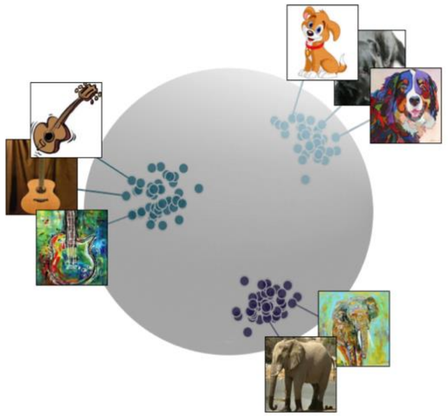


Geosciences & Astronomy
(e.g., Earth data)



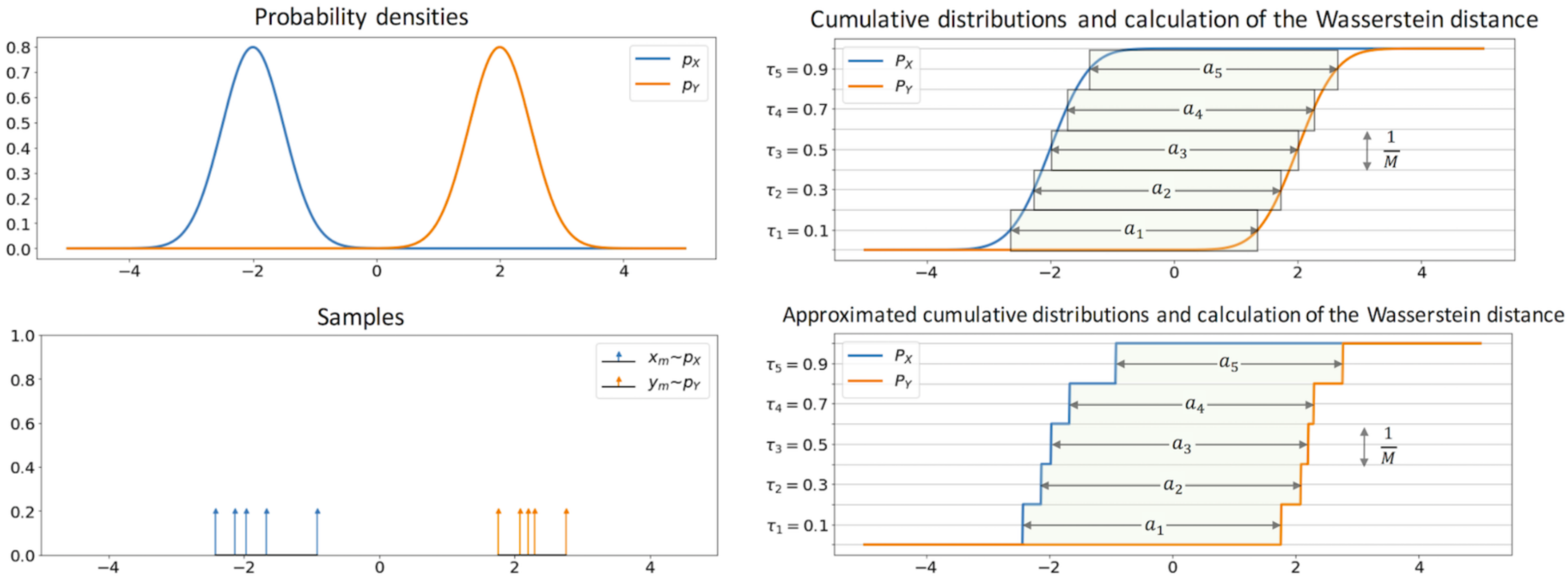
Neuroscience
(e.g., Cortical signals)

Computer vision
(e.g., 360^{o} images)



Machine Learning
(e.g., self-supervised learning)

The Sliced-Wasserstein Distances



$$W_c(\mu,\nu) = \int_0^1 c(P_X^{-1}(\tau), P_Y^{-1}(\tau))d\gamma(s_1,s_2)$$

In the discrete case, it becomes a sorting problem

Time complexity: $\mathcal{O}(NlogN)$ where N is the number of samples

The Stereographic Projection

Bridging the spherical and Euclidean manifolds

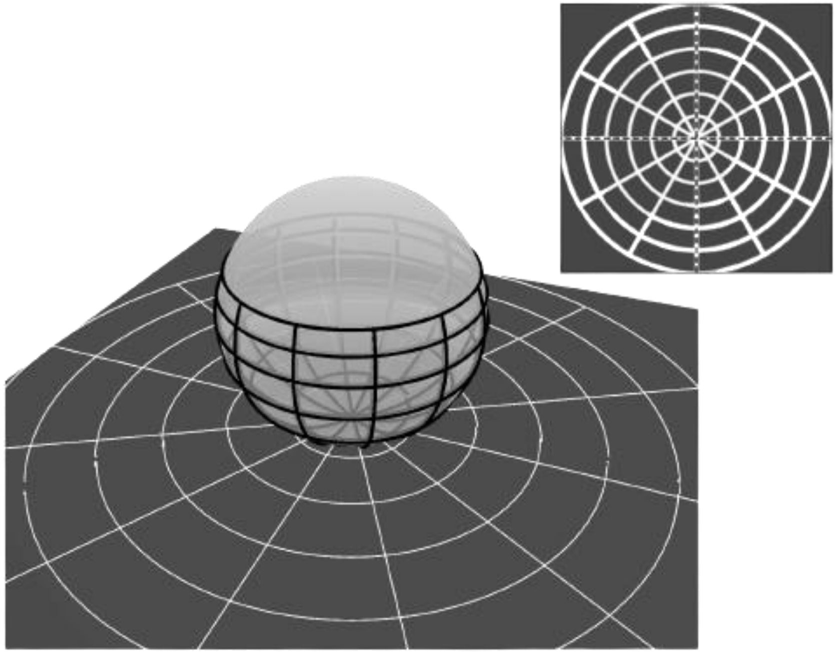
Let $s \in \mathbb{S}^d \backslash \{s_n\}$ then the projection $x = \phi(s)$ is given by

$$[x]_i = \frac{2[s]_i}{1-[s]_{i+1}}, \quad for\ i = 1,2, \dots d$$

Moreover, the inverse $s = \phi^{-1}(x)$ is given by

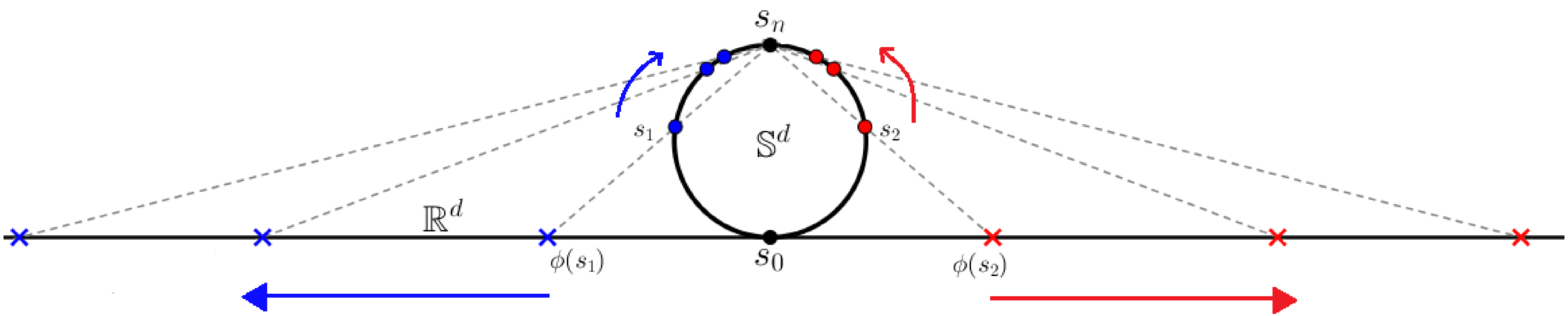
$$[s]_i = \frac{2[x]_i}{\sum_{j=1}^{d}[x]_j^2+1} = \frac{2[x]_i}{\|x\|^2 + 1}$$

$$[s]_{d+1} = \frac{\sum_{j=1}^{d}[x]_j^2-1}{\sum_{j=1}^{d}[x]_j^2+1} = \frac{\|x\|^2 - 1}{\|x\|^2 + 1}$$



$\phi: \mathbb{S}^d \backslash \{s_n\} \rightarrow \mathbb{R}^d$

Distance distortion



The Stereographic Projection can incur significant distortion.

At the extreme, consider points $s_1 = \left[-\epsilon, 0, \dots, 0, \sqrt{1-\epsilon^2}\right]$ and $s_2 = \left[\epsilon, 0, \dots, 0, \sqrt{1-\epsilon^2}\right]$. We have that

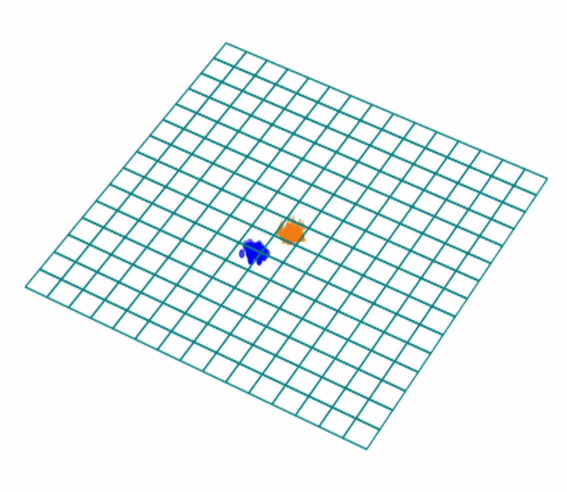
$$\lim_{\epsilon \to 0} \cos^{-1}(s_1, s_2) = 0$$

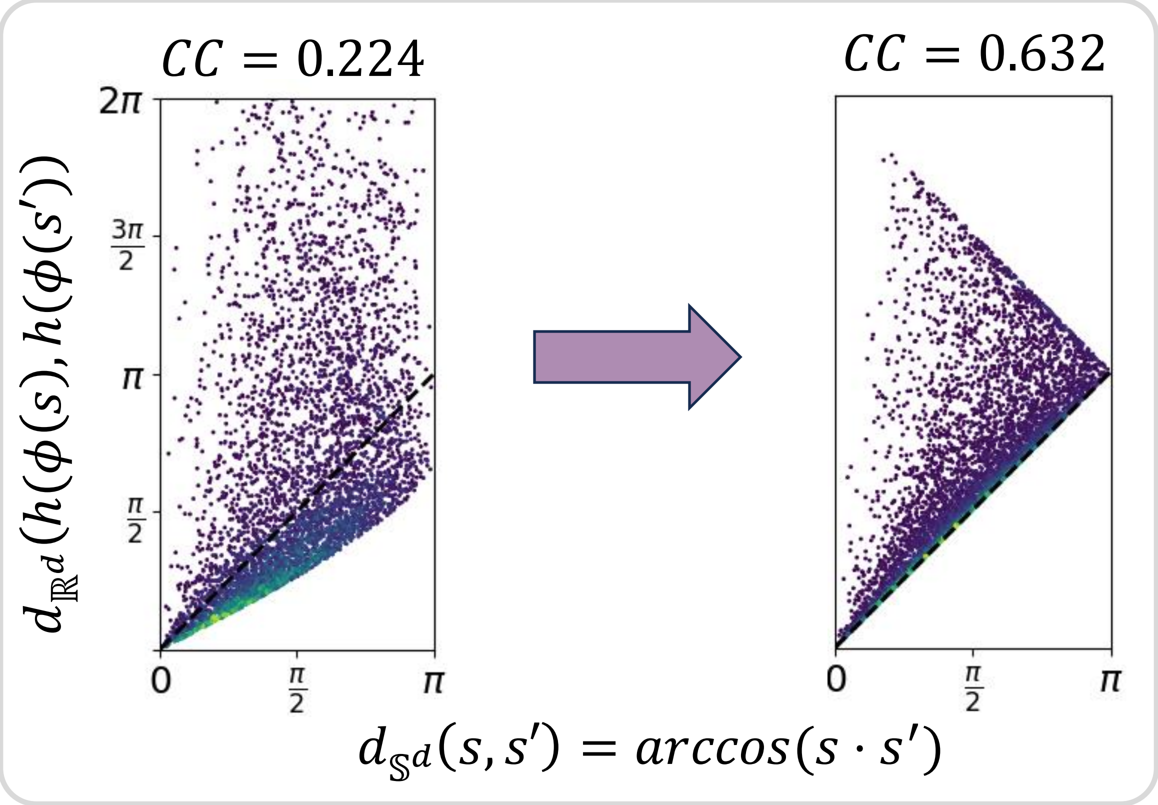
$$\lim_{\epsilon \to 0} \|\phi(s_1) - \phi(s_2)\| = \infty$$

Minimizing distortion

We propose an analytic function with bounded range $h(\cdot)$ such that $\mathrm{d}_{\mathbb{R}^d}(h(\phi(\cdot)), h(\phi(\cdot))) \approx \mathrm{d}_{\mathbb{S}^d}(\cdot,\cdot)$

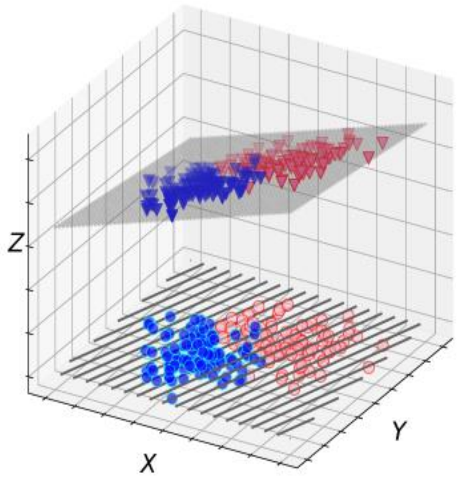
$$h_a(x) = \arccos\left(\frac{\|x\|^2 - 1}{\|x\|^2 + 1}\right)\frac{x}{\|x\|}$$

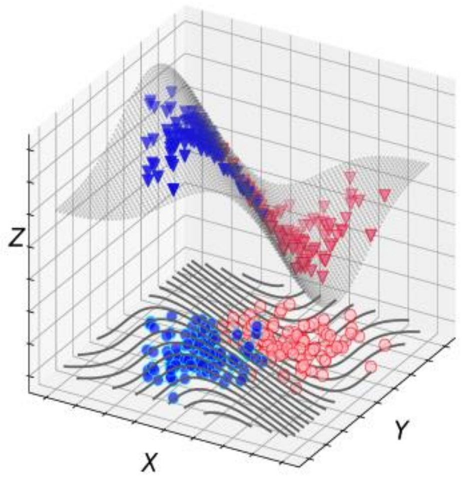




Alternatively, we could use a neural network to learn a nearly isometric embedding.

Spatial Slices





$$\mathcal{R}f(t,\theta) = \int_X f(x)\delta(t - \langle x, \theta \rangle)dx$$

$$\mathcal{H}f(t,\theta) = \int_{R^d} f(x)\delta(t - \langle h(x), \theta \rangle)dx$$

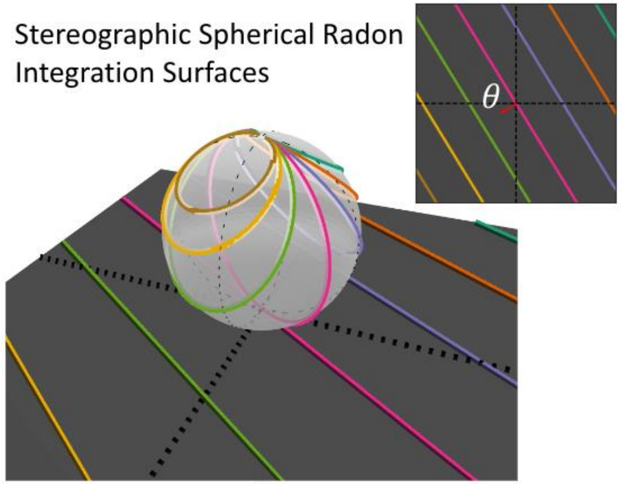
Figure credit: Chen et al. 2022

The Stereographic Spherical Radon Transform

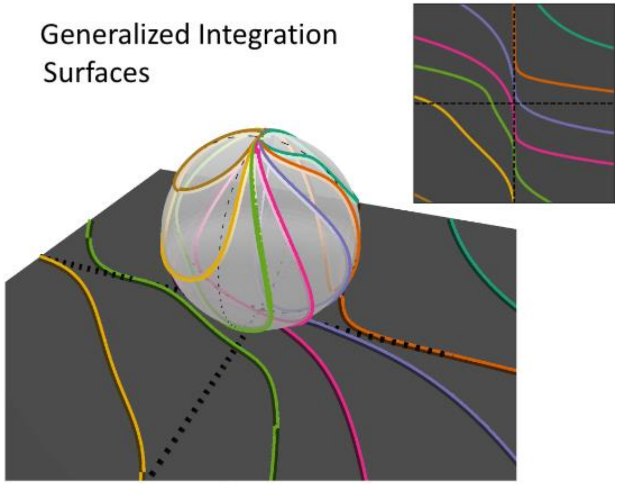
Let $\mu \in \mathcal{M}(\mathbb{S}^d)$ denote a Radon measure on $\mathbb{S}^d$ such that $\mu(\{s_n\}) = 0$. We denote the Stereographic Projection $\phi: \mathbb{S}^d\{s_n\} \to \mathbb{R}^d$. Then, the Stereographic Spherical Radon Transform, its generalized are respectively defined as

$$S_{\mathcal{R}}(\mu) = \mathcal{R}(\phi_{\#}\mu) \in \mathcal{M}(\mathbb{R} \times \mathbb{S}^{d-1})$$

$$S_{\mathcal{H}}(\mu) = \mathcal{H}(\phi_{\#}\mu) \in \mathcal{M}(\mathbb{R} \times \mathbb{S}^{d-1})$$



$H_{t,\theta} = \{x \in \mathbb{S}^d \,|\, \langle \phi(x), \theta \rangle = t\}$



$H_{t,\theta} = \{x \in \mathbb{S}^d \,|\, \langle h \circ \phi(x), \theta \rangle = t\}$

The S3W distances

Let $\mu, \nu \in \mathcal{P}_p(\mathbb{S}^d)$ denote two probability measures on the sphere in $\mathbb{R}^{d+1}$. The S3W distance can be defined as

$$S3W_{\mathcal{H},p}(\mu,\nu) = \left[\int_{\mathbb{S}^{d'-1}} W_p^p(\mathcal{S}_{\mathcal{H}}(\mu)_\theta, \mathcal{S}_{\mathcal{H}}(\nu)_\theta) d\sigma_{d'}(\theta)\right]^{\frac{1}{p}}$$

where $\sigma_{d'} \in \mathcal{P}(\mathbb{S}^{d'-1})$ represents a uniform measure on $\mathbb{S}^{d'-1}$

$$S3W_p(\hat{\mu},\hat{\nu}) \approx \left[\frac{1}{L}\sum_{l=1}^{L}\sum_{m=1}^{M}\left|\left\langle h\left(\phi_{\in}\left(x_{\pi_l[m]}\right)\right),\theta_l\right\rangle - \left\langle h\left(\phi_{\in}\left(y_{\pi_{l'}[m]}\right)\right),\theta_l\right\rangle\right|^p\right]^{\frac{1}{p}}$$

Rotation-Invariant Extension (RI-S3W)

Let $SO(d+1)$ denote the special orthogonal group in $\mathbb{R}^{d+1}$ and let R ∈ $O(d+1)$ denote a rotation. For $\mu \in \mathcal{P}_p(\mathbb{S}^d)$, we define the rotation-invariant extension of S3W as follows

$$RI\text{-}S3W_{\mathcal{H},p}(\mu,\nu) = \mathbb{E}_{R\sim\omega}[S3W_{\mathcal{H},p}(R_{\#}\mu, R_{\#}\nu)]$$

where ω denotes the normalized Haar measure on $SO(d+1)$

$$RI\text{-}S3W_p(\hat{\mu},\hat{\nu}) \approx \frac{1}{N_R}\sum_{n=1}^{N_R} S3W_p[(R_n)_{\#}\hat{\mu},(R_n)_{\#}\hat{\nu}]$$

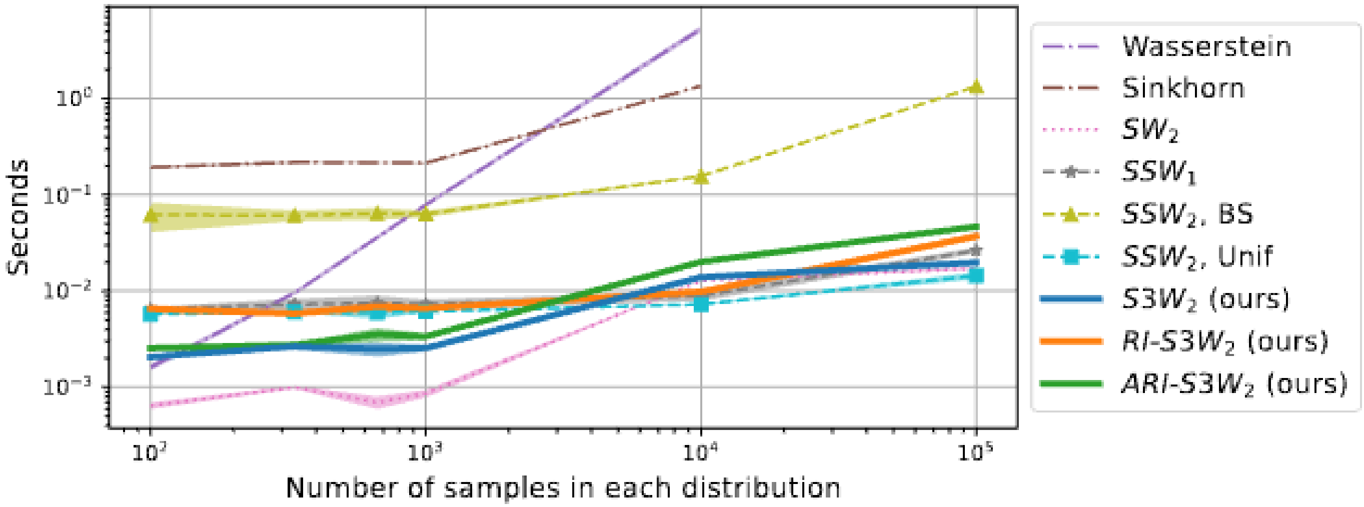
Runtime and complexity

Time Complexity

S3W has time complexity of $\mathcal{O}\big(LN(d + logN)\big)$.

RI-S3W is $\mathcal{O}\left(N_R\big(d^3 + Nd^2 + LN(d + logN)\big)\right)$.

Runtime



Experiments

Gradient flow on the sphere

Objective : matching the source and target distributions
Setup :
- 12 vMFs, 200 datapoints each
- Adam optimizer full batch and mini-batch (500)
- Metrics: runtime, NLL, $\log(W_2)$

Result :
Our methods achieve lowest loss and can be up to 10x faster

Density estimation with Earth data

Objective : estimate density of disasters on earth with NF
Setup :
- Earth datasets
- Rezende et al. 2020: Exponential map normalizing flows
- Metrics: NLL

Result :
Our methods achieve lowest NLL (and much faster)

Self-supervised learning

Objective : enforcing dispersity of learned representation
Setup :
- CIFAR-10
- Wang and Isola 2020: SSL on the sphere, 3-dimensional
- Metrics: runtime, accuracy

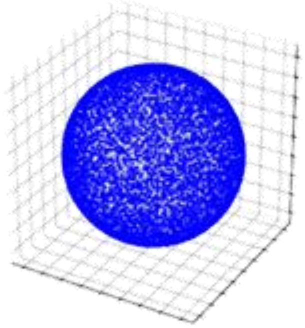
Result :
Our methods achieve slightly better acc. and are 1.5x faster

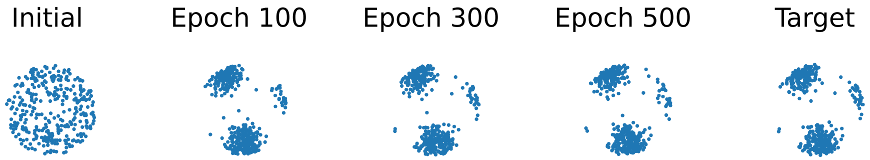
Sliced-Wasserstein Autoencoder

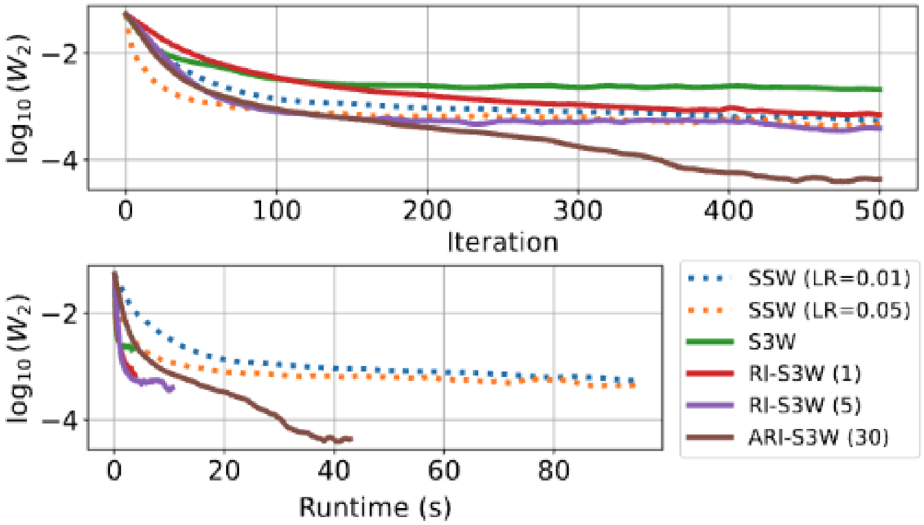
Objective : enforcing dispersity of learned representation
Setup :
- CIFAR-10, MNIST
- Kolouri et al. 2018: SSL on the sphere, 3-dimensional
- Metrics: runtime, NLL, $\log(W_2)$, BCE

Result :
Our methods achieve better results and are 2x faster

Gradient flow on the sphere

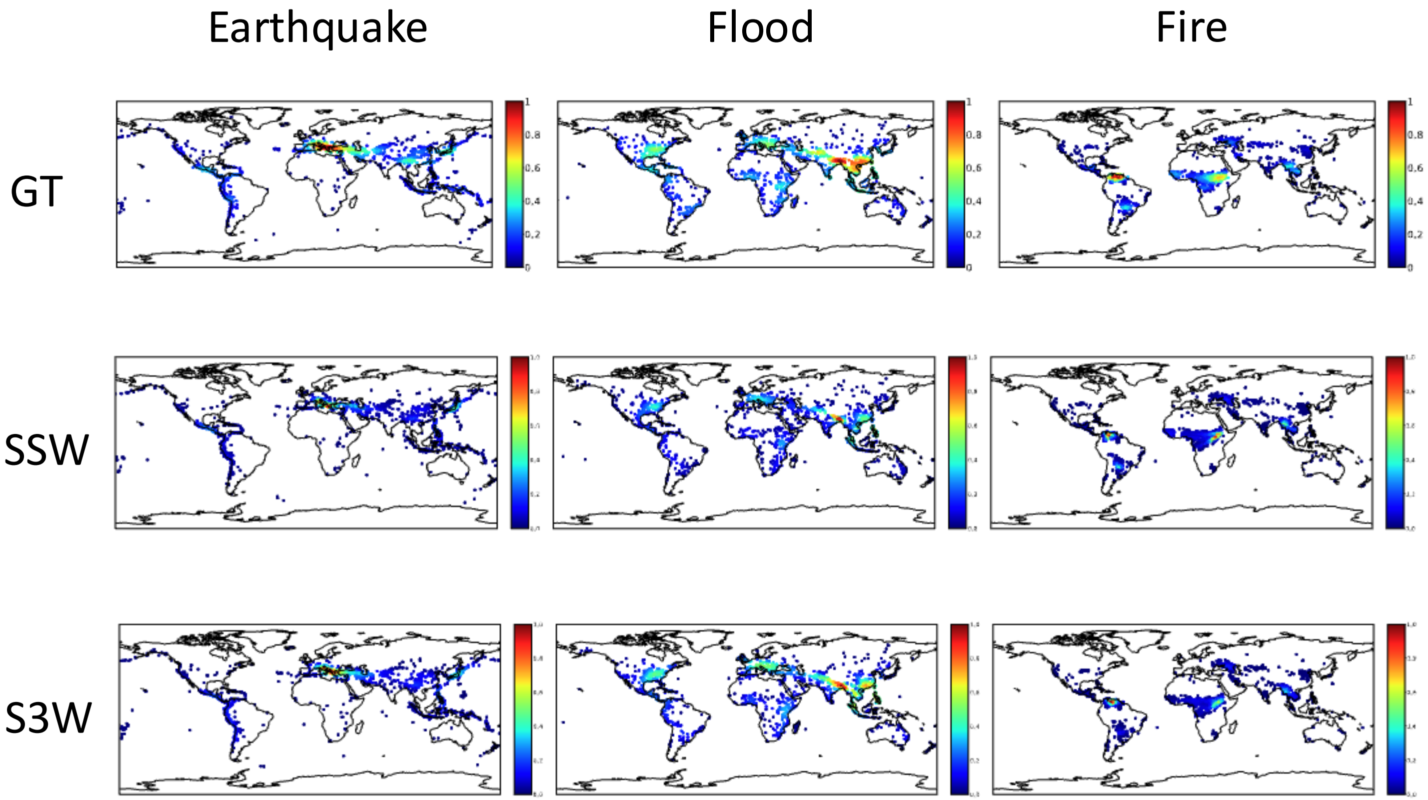






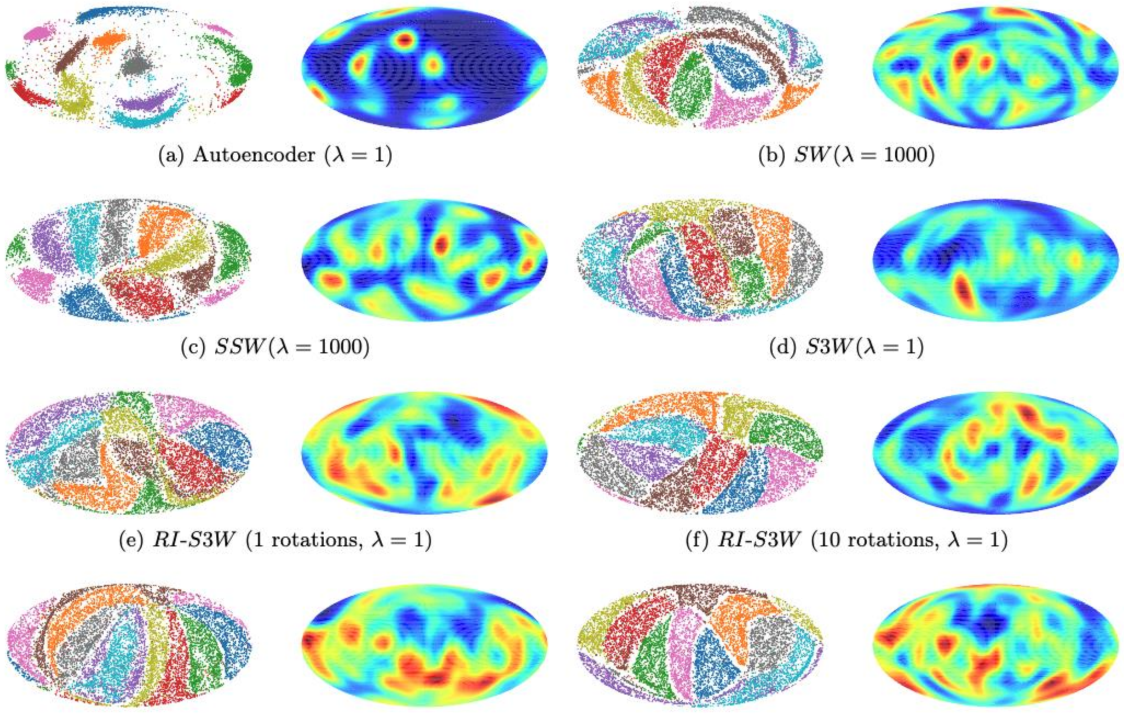
Method	Runtime(s)	NLL ↓	$\log W_2$ ↓
SSW (LR=0.01)	94.97 ± 0.96	-4974.50 ± 2.27	-3.27 ± 0.17
SSW (LR=0.05)	94.98 ± 0.27	-4974.46 ± 3.50	-3.32 ± 0.13
S3W	**3.32 ± 0.62**	-4753.81 ± 83.70	-2.57 ± 0.19
RI-S3W (1)	4.00 ± 0.83	-4957.22 ± 34.70	-3.15 ± 0.26
RI-S3W (5)	10.58 ± 0.27	-4983.56 ± 6.62	-3.50 ± 0.17
ARI-S3W (30)	42.88 ± 0.05	**-5025.37** ± 5.57	**-4.37 ± 0.21**

Density estimation with Earth data



Method	Quake ↓	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**

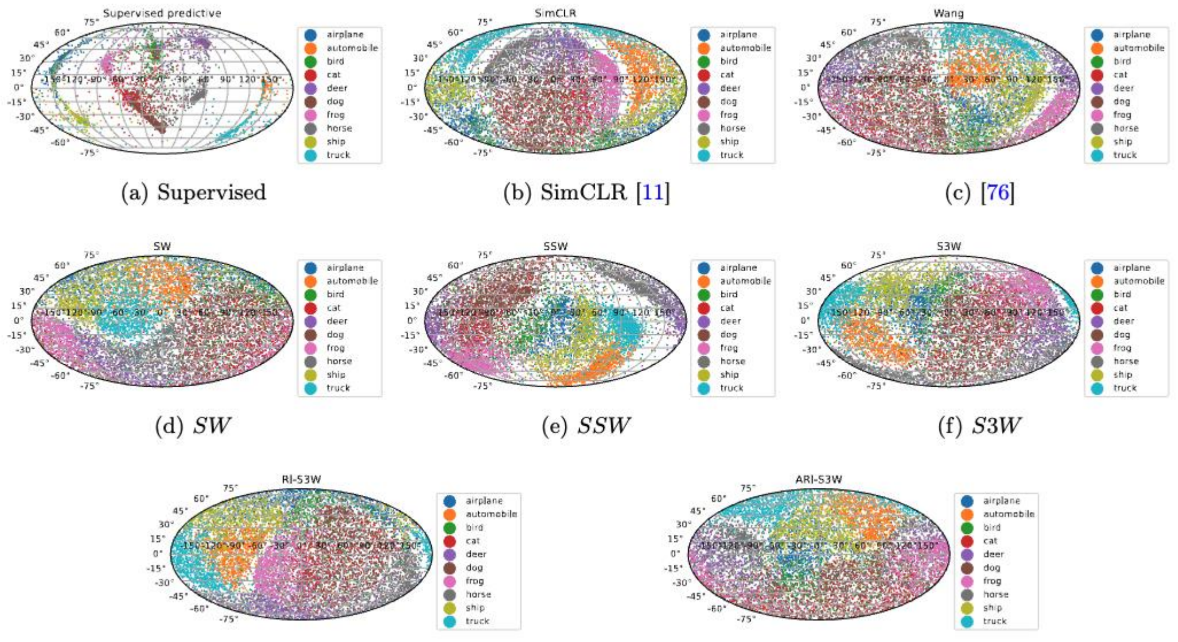
Self-supervised learning



(a) Autoencoder ($\lambda = 1$) (b) SW ($\lambda = 1000$) (c) SSW ($\lambda = 1000$) (d) $S3W$ ($\lambda = 1$) (e) $RI\text{-}S3W$ (1 rotations, $\lambda = 1$) (f) $RI\text{-}S3W$ (10 rotations, $\lambda = 1$) (g) $ARI\text{-}S3W$ (10 rotations, $\lambda = 1$) (h) $ARI\text{-}S3W$ (30 rotations, $\lambda = 1$)

Method	$\log W_2 \downarrow$	NLL $\downarrow$	BCE $\downarrow$	Time (s/ep.)
Supervised	-0.5132	0.0060	0.6319	**5.3243**
SSW	-2.1949	0.0052	0.6323	15.4651
SW	-3.3229	-0.0007	0.6348	5.4661
S3W	-3.3381	0.0025	**0.6318**	5.7511
RI-S3W (5)	-3.1424	**-0.0043**	0.6376	7.5443
ARI-S3W (5)	**-3.3853**	0.0028	0.6332	5.8316

Sliced-Wasserstein Autoencoder



(a) Supervised (b) SimCLR [11] (c) [76] (d) SW (e) SSW (f) $S3W$ (g) $RI\text{-}S3W$ (h) $ARI\text{-}S3W$

Method	Acc.(%) E/P ↑	Time(s/ep.)
Supervised	92.38 / 91.77	—
Hypersphere [76]	79.76 / 74.57	24.28
SimCLR [11]	79.69 / 72.78	**20.94**
SSW [5]	70.46 / 64.52	33.14
SW	74.45 / 68.35	21.09
S3W	78.54 / 73.84	21.36
RI-S3W (5)	**79.97** / 74.27	21.59
ARI-S3W (5)	79.92 / **75.07**	21.51