Keep your distance: learning dispersed embeddings on $\mathbb{S}_m$

Anonymous authors
Paper under double-blind review

Abstract

Learning well-separated features in high-dimensional spaces, such as text or image *embeddings*, is crucial for many machine learning applications. Achieving such separation can be effectively accomplished through the *dispersion* of embeddings, where unrelated vectors are pushed apart as much as possible. By constraining features to be on a *hypersphere*, we can connect dispersion to well-studied problems in mathematics and physics, where optimal solutions are known for limited low-dimensional cases. However, in representation learning we typically deal with a large number of features in high-dimensional space, and moreover, dispersion is usually traded off with some other task-oriented training objective, making existing theoretical and numerical solutions inapplicable. Therefore, it is common to rely on gradient-based methods to encourage dispersion, usually by minimizing some function of the pairwise distances. In this work, we first give an overview of existing methods from disconnected literature, making new connections and highlighting similarities. Next, we introduce some new angles. We propose to reinterpret pairwise dispersion using a maximum mean discrepancy (MMD) motivation. We then propose an online variant of the celebrated Lloyd's algorithm, of K-Means fame, as an effective alternative regularizer for dispersion on generic domains. Finally, we derive a novel dispersion method that directly exploits properties of the hypersphere. Our experiments show the importance of dispersion in image classification and natural language processing tasks, and how algorithms exhibit different trade-offs in different regimes.

1 Introduction

Dispersion, sometimes referred to as spreading or uniformity, is the property that a collection of points in a domain cover the domain well, with no two points being too close or too far. When the domain is a hypersphere, a useful space for representing embeddings of text (Meng et al., 2019) and images (Liu et al., 2017; Mettes et al., 2019), optimal dispersion is challenging. Embedding clumping, *i.e.*, occurrence of semantically distant embeddings that are close to each other in terms of distance metric, is a known problem, and it has been shown before that it negatively impacts the performance of the downstream tasks, such as image classification (Wang & Isola, 2020; Liu et al., 2021; Trosten et al., 2023), image generation (Liu et al., 2021), text classification (Wang & Isola, 2020) and text generation (Tokarchuk & Niculae, 2024). Beyond data embeddings, neural network layer weights have been shown to also benefit from dispersion (Liu et al., 2018; 2021; Wang et al., 2021).

In general, the problem of dispersing n points on the surface of a m-dimensional sphere, such that the angular distance between any two points is maximal, is an open mathematical problem known as the Tammes problem (Tammes, 1930). The optimal solutions for this problem are only known for small values of m and n (Fejes Tóth, 1943; Danzer, 1986; van der Waerden & Schütte, 1951; Robinson, 1961; Musin & Tarasov, 2012; 2015). The Tammes problem can also be formulated as a problem of finding an optimal spherical code (Conway et al., 1999; Cohn, 2024). However, in machine learning applications we typically deal with a large number of dimensions and many points. Moreover, perfect dispersion may not be optimal and may hurt the task-specific learning objective, so in learning applications we prefer being able to trade off between a task loss and a dispersion regularizer. For these reasons, we cannot directly use classical results, and instead focus on hyperspherical dispersion as a gradient-based optimization of a regularization function.

Contributions. In this work, we study the regularization approach to dispersion in machine learning applications, proposing new algorithms as well as casting new light on known ones. We survey a number of dispersion objectives used in the literature and reveal relationships between them (§2.4). We then show a new and fundamental connection between kernel-based dispersion and maximum mean discrepancy (MMD, Gretton et al., 2012) with a uniform measure (§3.1). Further, we adapt Lloyd's algorithm (Lloyd, 1982) as a stochastic dispersion regularizer (§3.2), and we introduce a novel and efficient regularizer based on slicing dispersion along great circles (§3.3). We evaluate (§4) old and new methods on synthetic small and large scale problems, as well as real-world large-scale applications in computer vision and natural language processing, revealing different trade-offs and throughout confirming the importance of representation dispersion for task performance. Additionally, we highlight that using Riemannian optimization (Bonnabel, 2013; Becigneul & Ganea, 2019) on the hypersphere, rather than projecting Euclidean gradient updates, benefits dispersion and overall task performance. A reusable library for spherical dispersion will be released as open-source software.

2 Background: Dispersion on the Hypersphere

2.1 Notation

We denote by $\mathbb{S}_m$ the $m-1$-dimensional hypersphere embedded in $\mathbb{R}^m$, *i.e.*, $\mathbb{S}_m = \{x \in \mathbb{R}^m \mid \|x\| = 1\}$. For $u, v \in \mathbb{R}^m$ we denote their Euclidean inner product by $\langle u, v\rangle \coloneqq \sum_{i=1}^m u_i v_i$. The hypersphere is an embedded Riemannian submanifold of $\mathbb{R}^m$. The tangent space of the hypersphere at a point x is $T_x\mathbb{S}_m \coloneqq \{v \in \mathbb{R}^m \mid \langle x, v\rangle = 0\}$, and the Riemannian inner product on it is inherited from $\mathbb{R}^m$, *i.e.*, for $u, v \in T_x\mathbb{S}_m$, $\langle u, v\rangle_x \coloneqq \langle u, v\rangle$. The geodesic distance or angular distance on $\mathbb{S}_m$ is denoted by $d_{\mathbb{S}}(x, x') \coloneqq \arccos(\langle x, x'\rangle)$, in contrast with the Euclidean distance in ambient space $d_{\mathbb{R}}(x, x') \coloneqq \|x - x'\|$. As a special case, for circles ($m = 2$) it is more convenient to work in an isomorphic angular parametrization, *i.e.*, $\mathbb{S}_2$ is isomorphic to $\{\theta \mid -\pi \leq \theta < \pi\} \subset \mathbb{R}$ with $d_{\mathbb{S}}(\theta, \theta') = |\theta - \theta'|$, and the embedding is $\theta \mapsto (\cos\theta, \sin\theta)$. We reserve the use of Greek letters τ, θ, ϕ for 1-d angles. We denote by Π_n the set of permutations of $(1, \ldots, n)$.

We denote as $X = (x_1, \ldots, x_n)$ an (ordered) collection, configuration, or design of n points on the same sphere, *i.e.*, each $x_i \in \mathbb{S}_m$. Sans-serif capitals, *e.g.*, Y, denote random variables.

2.2 Riemannian Optimization

Since the configurations we are considering consists of points on a sphere, the natural framework for optimization is Riemannian optimization, taking into account the geometry of the sphere $\mathbb{S}_m$. We briefly describe and illustrate the method for $\mathbb{S}_m$ as an embedded submanifold of $\mathbb{R}^m$. Given a point $x \in \mathbb{S}_m$ and a function $F : \mathbb{S}_m \to \mathbb{R}$, the Riemannian gradient is the projection the standard (Euclidean) gradient onto the tangent plane at x: (Figure 1)

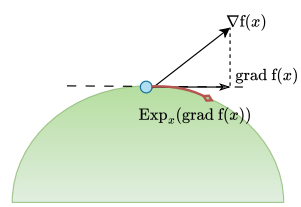


Figure 1: Euclidean vs. Riemannian optimization.

$$\operatorname{grad}_x F(x) = \operatorname{proj}_{T_x\mathbb{S}_m}(\nabla_x F(x)) = (I - xx^\top)\nabla_x F(x).$$

Vectors tangent at x can be interpreted as directions of travel for x along the manifold. This intuition gives the Riemannian gradient method:

$$x^{(t+1)} = \operatorname{retr}_{x^{(t)}}(\lambda^{(t)} \operatorname{grad}_x F(x^{(t)})),$$

where $\lambda^{(t)}$ is a sequence of step sizes, and the retraction retr_x maps tangent vectors back onto the surface. The canonical retraction is the exponential map $\operatorname{retr}_x^{(\exp)}(v) = \operatorname{Exp}_x(v) = \cos(\|v\|)x + \sin(\|v\|)\frac{v}{\|v\|}$; it moves from x along the sphere surface with constant velocity v for one unit of time. For optimization, approximate retractions such as the projection-based one also work and can be implemented slightly faster: $\operatorname{retr}_x^{(\text{proj})}(v) = (x + v)/\|x + v\|$. Riemannian gradient methods are due to Luenberger (1972); Lichnewsky (1979); Gabay (1982). Stochastic versions where the gradient is replaced by an estimation are studied by Bonnabel (2013), and accelerated versions like Riemannian Adam are due to Becigneul & Ganea (2019). While some works optimize spherical objectives ignoring the geometry and applying Euclidean gradient descent in

$\mathbb{R}^m$ interspersed with projection back onto $\mathbb{S}_m$, we note this is ill-defined: while formally it resembles the projected gradient method, $\mathbb{S}_m$ is a non-convex subset of $\mathbb{R}^m$. Projection is not defined at the origin of $\mathbb{R}^m$, and it exhibits quick jumps around the origin. Despite also being convergent for certain simple objectives (Vu et al., 2019) and being often used, we experimentally show in Appendix B that projected gradient is less well-behaved than Riemannian gradient for our purposes, and therefore stick to the natural choice of Riemannian optimization.

2.3 Measuring Dispersion

Intuitively, we say a configuration of n points on $\mathbb{S}_m$ is dispersed if the points are as well spread out, with each point being neither too close nor too far from its neighbors, and therefore covering the surface well. In coding theory, maximally-dispersed configurations form optimal *spherical codes*, and can be used to discretize continuous spherical data. The problem of finding such optimal configurations is also known, equivalently, as the sphere packing problem (Conway et al., 1999) and the Tammes problem (Tammes, 1930), and its general form is hard, with solutions found only for special values of n and m. Approximations can be found with continuous optimization methods, which also allow us to tackle multi-task objectives, where we seek to balance dispersion with some other task of interest. In order to develop dispersion regularizers that can be efficiently optimized, we will first review the traditionally used measures to quantify dispersion in the theoretical contexts listed.

Minimum distance. The Tammes problem (Tammes, 1930) is the problem of finding a configuration that maximizes the minimum distance between all pairs of points. A natural measure of dispersion would then be:

$$d_{\min}(X) = \min_{1 \leq i \neq j \leq n} d(x_i, x_j), \tag{1}$$

where d is a distance. In the sequel we will always define $d_{\min}$ in terms of the geodesic distance $d_{\mathbb{S}}$. As a metric, however, $d_{\min}$ may be less useful for comparing dispersion of approximate solutions, as it cannot distinguish between a configuration with $n-1$ well-dispersed points and one ε-close from another, from a configuration where all points are ε-close.

Spherical variance. Seeing the points X as vectors in $\mathbb{R}^m$, we may consider their average $\mu = \left(\sum_{i=1}^n x_i\right)/n$. This average is a vector lying inside the unit ball, and its length, sometimes called *mean resultant length*, gives a variance estimator studied in directional statistics (Jammalamadaka & Sengupta, 2001; Mardia, 1975):

$$\operatorname{svar}(X) = 1 - \|\mu\|, \text{ where } \mu = \frac{1}{n}\sum_{i=1}^n x_i. \tag{2}$$

If the points are highly concentrated, μ will be close to the surface of the sphere, and so $\operatorname{svar}(X)$ would be close to zero. When points are dispersed, their directions cancel out, μ is close to the origin, and so $\operatorname{svar}(X)$ is close to 1. Since μ must be in the unit ball, $\operatorname{svar}(X)$ must be between 0 and 1.

Unlike $d_{\min}$, configurations maximizing svar are not necessarily well dispersed: the spherical variance of $X = (x, x, -x, -x)$ is 1, but the minimum distance is 0 for any x. Nevertheless, its computational efficient and highly global nature make it a useful metric, in ways that can complement minimum distance.

2.4 Optimizing for Dispersion

In this section we review some of the methods for numerical optimization of dispersion in previous work. Such methods are almost exclusively pairwise in nature, leading to at least quadratic complexity. Along the way, we make some new connections between the methods.

Closest-point objectives. It may be tempting to optimize $d_{\min}(X)$ directly, as this is a definitional measure of dispersion. However, the gradient of $d_{\min}(X)$ is zero for almost all points, with the exception of the closest pair,[1] leading to slow convergence, as pointed out by Mettes et al. (2019) and Liu et al. (2018).

[1] In the case of ties, the function is not differentiable. Picking any closest pair ignoring ties gives a subgradient.

An alternative is to consider the closest point to every point in the configuration. The resulting *max-min* (MM) objective appears in several works:

$$\mathcal{L}_{\mathrm{MM}}(X) = -\frac{1}{n}\sum_{i=1}^{n}\left(\min_{j\neq i} d(x_i, x_j)\right). \tag{3}$$

Mettes et al. (2019) use cosine distance (equivalent to $d^2_{\mathbb{R}}$), while Wang et al. (MMA, 2021) use $d_{\mathbb{S}}$. Plugging in a log-distance instead of a distance gives the dispersion method proposed by Sablayrolles et al. (2019):

$$\mathcal{L}_{\mathrm{KoLeo}}(X) = -\frac{1}{n}\sum_{i=1}^{n}\log\min_{j\neq i} d(x_i, x_j), \tag{4}$$

which is—up to a constant—the Kozachenko-Leonenko estimator of differential entropy (Leonenko, 1987). Assuming the distance is the same, the Kozachenko-Leonenko estimator and the max-min distance are related by the following bound which follows from the Jensen inequality and the fact that $-e^{-t} \leq t - 1$:

$$-d_{\min}(X) \leq \mathcal{L}_{\text{Max-Min}}(X) \leq -\exp\left(-\mathcal{L}_{\mathrm{KoLeo}}(X)\right) \leq \mathcal{L}_{\mathrm{KoLeo}}(X) - 1. \tag{5}$$

We also note that technically the Kozachenko-Leonenko estimator is not applicable when the data is supported on a lower-dimensional submanifold such as $\mathbb{S}_m$: the differential entropy *w.r.t.* the Lebesgue measure on $\mathbb{R}^m$ is infinite. This issue is addressed in the estimator of Nilsson & Kleijn (2007). Nevertheless, as Eq. (5) suggests, seen simply as an optimization objective, $\mathcal{L}_{\mathrm{KoLeo}}$ has the intended dispersion effect.

Energy and kernel methods. While the Tammes problem is defined in terms of the minimum distance, a related problem is to minimize an energy of the form:

$$\mathcal{L}_{\mathrm{MHE},k}(X) = \frac{1}{n(n-1)}\sum_{1\leq i\neq j<n} k(x_i, x_j). \tag{6}$$

MHE stands for *minimum hyperspherical energy* (Liu et al., 2018).[2] When k is the Riesz 1-kernel with Euclidean distance, *i.e.*, $k(x, x') = d_{\mathbb{R}}(x, x')^{-1}$, $\mathcal{L}_{\mathrm{MHE},k}(X)$ corresponds to minimizing the electrostatic energy between n electrons, and its minimization is known as the Thomson problem (Thomson, 1904). This energy objective is minimized in a number of works on dispersion (Gautam & Vaintrob, 2013; Liu et al., 2018; 2021). Similarly, Wang & Isola (2020) propose a related objective using the RBF (also known as Gaussian) kernel $k(x_i, x_j) = \exp\left(\gamma\langle x_i, x_j\rangle\right)$ in the overall expression

$$\mathcal{L}_{\mathrm{WI}}(X) = \log\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} k(x_i, x_j). \tag{7}$$

Denoting $t(X) = \sum_i k(x_i, x_i)$, we thus have $\exp(\mathcal{L}_{\mathrm{WI}}(X)) = (1 + 1/n^2)\left(L_{\mathrm{MHE},k}(X) + t(X)\right)$. For many kernels (including RBF, but not Riesz with $s \geq 0$), $t(X)$ is a finite constant independent on X and thus, as exp is monotonic, the two objectives are optimization-equivalent. We shall revisit kernel properties in §3.1. While in general such energy minimization objectives lead to different optimal solutions, some relationships to the Tammes problem exist. For example, using the Riesz kernel with $s \to \infty$, or the RBF kernel with $\gamma \to \infty$, are equivalent to the Tammes problem in the limit. (Appendix A.1.1)

Evaluating a kernel energy requires the entire (or at least the upper triangle of the) sample Gram matrix K with $(K)_{ij} = k(x_i, x_j)$, so these methods have at least quadratic complexity, just like the closest-point methods above. Liu et al. (2021) also show that the determinant of K is a dispersion measure; nevertheless calculating determinants require diagonalization, which has cubic complexity in n.

[2]Some definitions omit the scaling term $n(n-1)$.

Non-pairwise objectives. Spherical variance (Eq. (2)) requires only linear time to evaluate. However, it is unfortunately not suitable for gradient-based optimization, as $\nabla_x \operatorname{svar}(X) = -x/2n\|\mu\|$ (it is in the direction of x), and thus $\operatorname{grad}_x \operatorname{svar}(X) = 0$ for any point x part of any configuration X. Intuitively, svar wants to shrink all points toward the origin, which can result in no optimization progress on the sphere. As a generalization of spherical variance, it is possible to define more generic manifold notions of variance using Fréchet expectations, *i.e.* $\min_x \sum_i d(x, x_i)^2$ (Pennec, 2006), or variants using kernels instead of distances (maximum polarization, Liu et al., 2021). While avoiding explicit pairwise comparisons, such measures require iterative bilevel optimization, which poses additional numerical and computational challenges; moreover, for certain symmetrical optimally-dispersed situations any point on $\mathbb{S}_m$ solves the inner optimization and thus higher-order gradients can be ill-behaved.

3 New Insights Into Optimizing for Dispersion

In this section, we give a new interpretation of the uniformity regularizer discussed in §2, in terms of (squared) MMD, and then define two novel dispersion objectives with more appealing computational properties.

3.1 Energy-based dispersion and MMD

A promising way of thinking about dispersion comes from the statistical perspective of sample-based tests of uniformity. Indeed, spherical variance appears in the Rayleigh test for uniformity on the hypersphere $\mathbb{S}_m$ (Mardia & Jupp, 1999, 10.4.1), with test statistic $mn\|1 - \operatorname{svar}(X)\|^2$. Liu et al. (2018) show further connections between Ajne's test (Mardia & Jupp, 1999, 10.4.1) and MHE with a specific choice of kernel.

An alternative statistical test for uniformity can be derived from the *maximum mean discrepancy* (MMD), which is a nonparametric test measuring the distance between two probability distributions (Gretton et al., 2012). The next result shows that on $\mathbb{S}_m$, for kernels defined in terms of angles, the squared MMD from the uniform distribution simplifies to a familiar form.

Lemma 1 (MMD^2 with the uniform distribution on $\mathbb{S}_m$.) *Let p be a distribution on $\mathbb{S}_m$, and denote by $u = \operatorname{Uniform}(\mathbb{S}_m)$ the uniform distribution on $\mathbb{S}_m$. Let k be a positive definite kernel on $\mathbb{S}_m$ such that $k(x, y) = f(\langle x, y\rangle)$ for some function $f\colon [-1, 1] \to \mathbb{R}$, and $\mathcal{F}$ be the unit ball in the reproducing kernel Hilbert space (RKHS) corresponding to k.*

Then, for i.i.d. random variables X, X' with distribution p, we have

$$\mathrm{MMD}^2[\mathcal{F}, p, u] = \mathbb{E}_{\mathsf{X},\mathsf{X}'\sim p}\left[k(\mathsf{X}, \mathsf{X}')\right] - c,$$

where the constant c only depends on the choice of k.

The assumption of expressing kernels in terms of the cosine $\langle x, y\rangle$ applies broadly and is required for rotational symmetry, and is analogous to expressing kernels in terms of distances, as the geodesic and Euclidean distances can both be given in terms of the cosine. The proof of Lemma 1 is provided in Appendix A.1. As a direct consequence, we may interpret $\mathcal{L}_{\mathrm{MHE},k}$ as an unbiased estimator of squared MMD against the uniform distribution on $\mathbb{S}_m$:

Proposition 1 *Let p be any distribution on $\mathbb{S}_m$ and k a positive definite kernel with the property from Lemma 1. Then, $\mathcal{L}_{\mathrm{MHE},k}$ as defined in Eq. (6) is an unbiased estimator of MMD^2 between p and the uniform distribution on the sphere, with respect to the unit ball in the corresponding reproducible kernel Hilbert space.*

Proof. Combine Lemma 1 with Lemma 6 of Gretton et al. (2012) and identify with Eq. (6).

For our application, we take p to be an empirical distribution supported at a finite number of points X. Since the uniform distribution is not in this family, MMD cannot reach zero, but we still have a meaningful notion of distance between X and the continuous uniform distribution. This result provides a valid interpretation of kernel-based dispersion objectives like MHE and WI without requiring taking a limit toward infinitely many points, as needed in the motivation of Wang & Isola (2020).

Choice of kernel. The most commonly-used kernels for dispersion are 1-Riesz $k_{\text{Riesz},s=1}(x,y) = d(x,y)^{-1}$ (due to its connection to the Thomson problem), and RBF $k_{\text{RBF},\gamma} = \exp(-\gamma d^2(x,x'))$. On the sphere (with $d_{\mathbb{S}}$), the RBF kernel is unfortunately not positive definite, while the Laplace kernel $k_{\text{Laplace},\gamma}(x,x') = \exp(-\gamma d(x,x'))$ is (Feragen et al., 2015). Inspired by Wang et al. (2021, Fig. 2), which shows the gradient norm of MMA is constantly 1 regardless of the angle to the nearest neighbor, we compare in Figure 2 the gradient norms for various kernels as a function of the angle to the other point. We find the geodesic Laplace kernel to be best-behaved from this perspective, as it is bounded and monotonically decreasing. All kernels that use $d_{\mathbb{R}}$, and also the RBF kernel with $d_{\mathbb{S}}$, first increase for small angles before decreasing. We give a detailed table of these kernels in Appendix A.1.1.

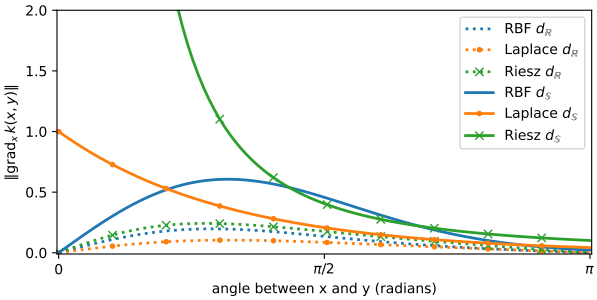


Figure 2: Length of the gradient contribution to a point by another point *w.r.t.* kernel k, as a function of their angle.

3.2 Lloyd's Algorithm

An alternative formulation of dispersion can be reached by considering dispersion as *quantization* of a uniform measure. Quantization is the problem of approximating a given measure by an empirical measure supported at a few centers. Lloyd's algorithm (Lloyd, 1982), henceforth *Lloyd*, is a celebrated algorithm, introduced initially for quantization of uniform intervals in $\mathbb{R}$ for signal processing, but subsequently broadly applied, *e.g.*, in machine learning and graphics. It is an iterative algorithm that alternates between two stages. First, for each of the current centers, we find its *Voronoi cell*: the set of points in the domain closer to it than to the other centers. Afterward, all centers are replaced with the center of their corresponding Voronoi cell.

When the target measure is another empirical measure, quantization recovers *k-means clustering*; for continuous measures, it can be seen as an infinite generalization thereof. Both the discrete and continuous cases generalize readily to Riemannian manifolds with an adequate choice of distance (Le Brigant & Puechmorel, 2019). While Lloyd's algorithm and k-means are originally batch algorithms, stochastic gradient versions have been developed (Bottou & Bengio, 1995; Sculley, 2010), including, independently, in the Riemannian case (Le Brigant & Puechmorel, 2019). In general, given a domain $\mathbb{D}$, which could be a manifold or a compact subset of one (for quantization), or a discrete dataset (for clustering), the n optimal centroids are a minimizer of[3]

$$\mathcal{L}_{\text{Lloyd}}(X) = \mathbb{E}_{\mathsf{Y}\sim\text{Unif}(\mathbb{D})}\left[\min_{j\in[n]} \frac{1}{2} d^2(\mathsf{Y}, x_j)\right]. \tag{8}$$

A stochastic gradient of the Lloyd regularizer can be obtained by drawing m uniform samples on $\mathbb{D}$. Intuitively, each cluster center is pulled toward the barycenter of the uniform samples assigned to it; an approximation to the true Voronoi barycenter.

For dispersion on the sphere, we take $\mathbb{D} = \mathbb{S}_m$ and $d = d_{\mathbb{S}}$. While traditionally Lloyd's algorithm corresponds to minimizing $\mathcal{L}_{\text{Lloyd}}$ alone, we propose using $\mathcal{L}_{\text{Lloyd}}$ as a regularizer to move X closer to optimal Voronoi centers of the sphere, while also minimizing some main task-specific objective. The complexity of this regularizer is controlled by the number of samples: For efficiency, m should be much less than n. Notice that unassigned cluster centers are not updated in an iteration, but the stochastic gradient of assigned centers implicitly depends on all other centers competitively through the cluster assignment.

3.3 Sliced Dispersion

The previously discussed algorithms are generally applicable to other manifolds. We now show how using properties of the sphere we may obtain an alternative algorithm for embeddings dispersion. The key idea is that, while in three or more dimensions it is hard to find the location of n evenly distributed points, on the

[3]The target measure need not be uniform. Le Brigant & Puechmorel (2019) give general conditions for existence.

$\mathbb{S}_2$ circle this can be done efficiently. The following set of angles is one optimal configuration:

$$\Phi = (\phi_1, \ldots, \phi_n) \quad \text{where} \quad \phi_k = -\pi \frac{n+1}{n} + \frac{2\pi k}{n}.$$

Any other optimal configuration must be a rotation of this one, *i.e.*, $\tau + \Phi$ for $\tau \in (-\pi, \pi)$, followed by a permutation of these angles. Given a permutation $\sigma \in \Pi_n$ denote $\Phi_\sigma = (\phi_{\sigma(1)}, \ldots, \phi_{\sigma(n)})$. We can then write the set of all possible ordered optimally-dispersed configurations as

$$D_n\mathbb{S}_2 \coloneqq \{\tau + \Phi_\sigma \mid \tau \in (-\pi, \pi), \sigma \in \Pi_n\}. \tag{9}$$

Given an ordered configuration of angles $\Theta = (\theta_1, \ldots, \theta_n) \subset \mathbb{S}_2$, we define its (angular) distance to the maximally-dispersed set as:

$$d^2(\Theta, D_n\mathbb{S}_2) = \min_{\hat{\Theta} \in D_n\mathbb{S}_2} \sum_{i=1}^{n} \frac{1}{2}(\theta_i - \hat{\theta}_i)^2. \tag{10}$$

Projecting onto $D_n\mathbb{S}_2$ means finding the closest dispersed configuration to a given one. The following lemma shows how to efficiently compute this projection, in the time it takes to sort n angles.

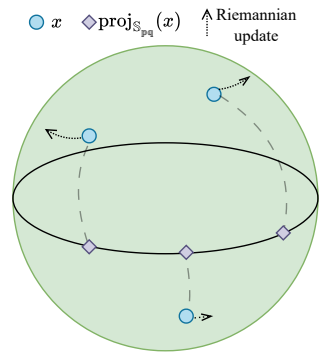


Figure 3: Sliced dispersion update along a single great circle $\mathbb{S}_{pq}$.

Lemma 2 *Optimal circular dispersion.* The projection $\arg\min_{\hat{\Theta} \in D_n\mathbb{S}_1} \sum_{i=1}^{n} 1/2\,(\theta_i - \hat{\theta}_i)^2$ is given by $\hat{\theta}_i^\star = \tau^\star + \phi_{\sigma^{-1}(i)}$, where σ is the permutation such that $\theta_{\sigma(1)} \leq \theta_{\sigma(2)} \leq \cdots \leq \theta_{\sigma(n)}$, and $\tau^\star = \sum_i \theta_i / n$.

In arbitrary dimensions, a similar construction is not possible, since the optimal configurations do not have tractable characterizations. We instead *slice* a high-dimensional spherical dataset along a great circle; similar to Bonet et al. (2023). The following result gives the geodesic projection.

Lemma 3 (Projection onto great circle.) *Let $p, q \in \mathbb{S}_m$ with $\langle p, q \rangle = 0$. Two such vectors determine a unique great circle $\mathbb{S}_{pq} \subset \mathbb{S}_m$ defined by $\mathbb{S}_{pq} \coloneqq \{\cos(\theta)p + \sin(\theta)q \mid -\pi \leq \theta < \pi\} \simeq \mathbb{S}_1$. The nearest point on $\mathbb{S}_{pq}$ to a given $x \in \mathbb{S}_m$ is given by* $\text{proj}_{\mathbb{S}_{pq}}(x) = \arctan2\left(\langle x, q \rangle, \langle x, p \rangle\right)$.

A well-dispersed configuration over $\mathbb{S}_m$ should remain fairly well-dispersed along any slice on average. If we denote $\text{proj}_{\mathbb{S}_{pq}}(X) \coloneqq (\text{proj}_{\mathbb{S}_{pq}}(x_1), \ldots \text{proj}_{\mathbb{S}_{pq}}(x_n))$, we may capture this intention by the following **sliced dispersion regularizer**:

$$\mathcal{L}_{\text{Sliced}}(X) = \mathbb{E}_{\mathsf{P},\mathsf{Q}}\left[d^2(\text{proj}_{\mathbb{S}_{\mathsf{PQ}}}(X), D_n\mathbb{S}_{\mathsf{PQ}})\right], \tag{11}$$

where d^2 is defined in Eq. (10). The joint distribution of P and Q is a distribution over great circles. A sensible choice is the uniform distribution over great circles, which can be sampled by orthogonalizing a $2 \times m$ matrix with standard normal entries. An alternative is to sample from the discrete uniform distribution over axis-aligned great circles, *i.e.*, $p = e_i, q = e_j$ with i, j drawn without replacement from $\{1, \ldots, m\}$; the latter choice allows slightly faster sampling and projection.

Note that unlike algorithms such as principal geodesic analysis (Fletcher et al., 2004), which keep X fixed and optimize for p, q to maximize variance, our intuition is the opposite: we update X to increase dispersion along *all* great circles. The following proposition efficiently computes stochastic gradients of $\mathcal{L}_{\text{Sliced}}$.

Proposition 2 *Denote $\theta_i|_{pq} = \text{proj}_{\mathbb{S}_{pq}}(x_i)$, and $\hat{\theta}_i^\star|_{pq}$ the corresponding dispersion maximizer computed using Lemma 2. The Euclidean and Riemannian gradients of $\mathcal{L}_{\text{Sliced}}$ are equal and given by:*

$$\text{grad}_{x_i}\mathcal{L}_{\text{Sliced}}(X) = \mathbb{E}_{\mathsf{P},\mathsf{Q}}\left[\left(\theta_i|_{\mathsf{PQ}} - \hat{\theta}_i^\star|_{\mathsf{PQ}}\right)\frac{\langle x_i, \mathsf{P}\rangle \mathsf{Q} - \langle x_i, \mathsf{Q}\rangle \mathsf{P}}{\langle x_i, \mathsf{Q}\rangle^2 + \langle x_i, \mathsf{P}\rangle^2}\right].$$

All proofs in this section are provided in Appendix A.2.

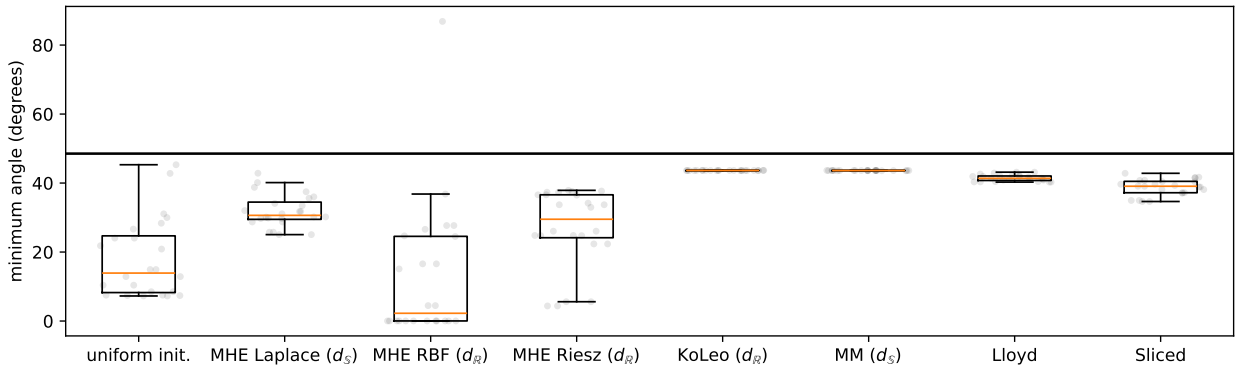


Figure 4: Minimum angles (degrees) for each of the $n = 24$ points. The horizontal line indicates the angle for known optimal solution; in an ideal solution, all angles would be on the horizontal line.

4 Applications

We demonstrate the application of dispersion objectives and provide a comparative analysis on both synthetic and real-world tasks.

4.1 Tammes problem

We evaluate the dispersion methods introduced in §2 and §3 by verifying that they can approximate the known solution to the Tammes problem for $n = 24$ in three dimensions (Robinson, 1961), by considering the minimum angle between points of the optimal configuration. Uniformly sampled points are dispersed using the regularizers described in §2 and §3. Starting from a uniform initialization, we apply 10k iterations of Riemannian Adam (Becigneul & Ganea, 2019) with learning rate 0.005. Laplace and RBF kernels use $\gamma = 1$, Riesz kernels use $s = 1$. Lloyd uses 300 samples from the uniform measure, while the sliced method uses a single random great circle per iteration.

The pairwise methods with min-max objective (MM and KoLeo) perform best, unsurprisingly, as they are closest-related to the Tammes objective; they are followed by Lloyd and Sliced. Among the MHE methods, the Laplace kernel with geodesic distance works best here. For comparison, in Appendix B we perform this experiment with Euclidean projected Adam instead of Riemannian Adam, finding a decrease in performance.

4.2 Synthetic Embeddings

In practice, we are mostly interested in dispersion of large amount of points in dimension $d \gg 3$. We evaluate the behavior of the regularizers discussed in §2 on $n = 20$k points in dimension $m = 64$, under two initial conditions: uniform samples (thus already somewhat spread out), and clumped points (sampled from a power spherical distribution with scale $\kappa = 100$) (De Cao & Aziz, 2020).

method	steps/sec ($\pm$ s.e.)
MHE (RBF, $d_{\mathbb{R}}$)	79.40 ± 0.06
KoLeo ($d_{\mathbb{R}}$)	90.52 ± 0.07
MM ($d_{\mathbb{S}}$)	91.29 ± 0.06
Lloyd	90.98 ± 0.07
Sliced	87.72 ± 0.14

Table 1: Timing (mean gradient steps per second) and standard error of the mean for the synthetic experiments.

We optimize using Riemannian Adam for 5k iterations with learning rate 0.001. Laplace and RBF kernels use $\gamma = 1$, Riesz kernels use $s = 1$, and the Sliced regularizer uses axis-aligned great circles for efficiency. We set the batch size and number of samples to result in the same order of magnitude of computations: for all pairwise objectives (MHE, MM, KoLeo) we use minibatches of 512 points. For Lloyd, we also take minibatches of size 512, and draw 512 random uniform points to estimate the expectation. For Sliced, we draw 13 random axis-aligned circles (as $13n \approx 512^2$). We report spherical variance and minimum distances in Figure 5, and additionally measure timings in Table 1.

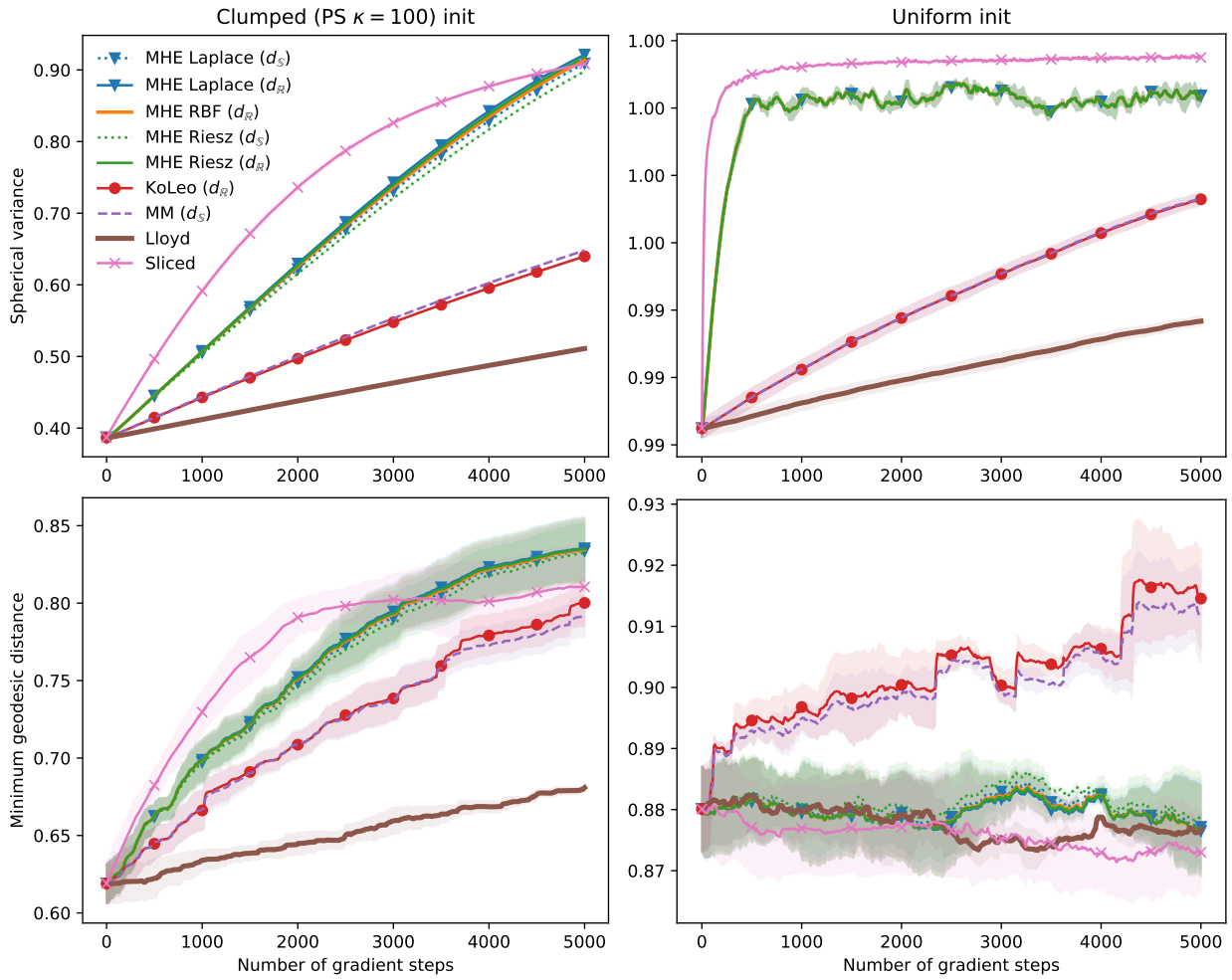


Figure 5: Convergence and performance of different dispersion objectives on synthetic data with $n = 20\,000, m = 64$. Mean and its standard error over three random initializations.

In contrast to the small-scale case of the Tammes problem, we find the choice of kernel and distance in MHE to not be empirically important in this large-scale regime. We find somewhat different trends in the clumped versus the uniform initialization scenarios. The clumped case roughly corresponds to a optimization alongside a strong downstream objective that pushes against dispersion. In this case, we find that MHE and Sliced perform best, with the latter making faster progress initially. In the uniform scenario, when the points are already well-separated, MHE and Sliced remain strong in terms of spherical variance, but the MM-style regularizers are the only ones that manage to make progress in terms of $d_{\min}$. In terms of speed (Table 1), even after balancing for amount of computation, we find MHE to be slower than the others.

Overall, these results suggest that Sliced is a contender for scenarios where initial dispersion is poor and fast progress is wanted, while MaxMin objectives seem preferable after some dispersion has already been achieved. Of course, $d_{\min}$ and svar do not necessarily correlate with improvement on downstream task metrics; the next sections are dedicated to such applications.

4.3 Image Classification with Prototypes

Mettes et al. (2019) showed that learning prototypes with dispersion encouraged by minimizing the maximum cosine similarity on a hypersphere improves classification results on ImageNet-200 (Le & Yang, 2015). In our naming system, this objective is $\mathcal{L}_{\text{MM}}$ with $d^2_{\mathbb{R}}$.

We first show in Table 2 that applying Riemannian gradient descent on the sphere, rather than projected gradient as done by Mettes et al. (2019), leads to the better class separation, and as a result better classification accuracy. Second, we compare the classification accuracy given the prototypes trained with different dispersion objectives discussed in §2 and §3. We use Riemannian gradient descent for all except the projection baseline; we use $\gamma = 1$ for RBF and $s = 1$ for Riesz kernels, 200 uniform samples in Lloyd, and a single sampled great circle for Sliced. Also, Table 2 shows that, for $m = 50$, MHE with the RBF kernel performs the best among all dispersion objectives considered, even though the minimum distance is smaller compared to other pairwise-distance based objectives. It proves that even though we can measure the dispersion using minimum distance, we cannot rely on this metric alone as a predictive factor of the downstream task accuracy.

prototypes	50		100		200	
	Acc.	$d_{\min}$	Acc.	$d_{\min}$	Acc.	$d_{\min}$
MM ($d^2_{\mathbb{R}}$, projected)	41.67	1.22	42.76	1.36	43.03	1.44
MM ($d^2_{\mathbb{R}}$)	42.59	1.46	42.96	1.52	**43.27**	1.56
KoLeo ($d_{\mathbb{R}}$)	41.78	1.37	43.12	1.44	42.37	1.49
MM ($d_{\mathbb{S}}$)	41.72	1.39	**43.47**	1.46	42.90	1.51
MHE (Riesz, $d_{\mathbb{S}}$)	43.37	1.41	42.25	1.56	34.47	1.58
MHE (RBF, $d_{\mathbb{R}}$)	**43.87**	1.22	42.73	1.57	34.53	1.58
Lloyd	41.69	1.20	42.42	1.30	43.09	1.35
Sliced	40.76	1.10	42.34	1.20	42.92	1.33

Table 2: ImageNet-200 classification accuracy. Prototypes are trained with different separation conditions. In bold we emphasise the best accuracy in a column. MM ($d^2_{\mathbb{R}}$, projected) is the objective of Mettes et al. (2019) optimized with Euclidean projected gradient; all other rows use Riemannian optimization.

Interestingly, when dimensionality is equal to the number of points, MHE results degrade noticeably, as shown in Table 2. For both kernels considered, the minimum distance and median distance is approximately 1.57 radian or 90.3°, which is close to an orthogonal solution. Orthogonality is suggested by Mettes et al. (2019) to be insufficient despite being optimally dispersed.

4.4 Neural Machine Translation

Embeddings learned with the vanilla transformer model (Vaswani et al., 2017) are known for their inefficiency in utilizing space effectively, leading to the collapse of token representations (Gao et al., 2019; Wang et al., 2020). This issue is particularly pronounced for rare tokens (Gong et al., 2018; Tokarchuk & Niculae, 2024; Zhang et al., 2022). Gong et al. (2018) proposed to alleviate the problem of rare tokens by learning frequency-agnostic embeddings, while Zhang et al. (2022) proposed to use contrastive learning. In our approach, we tackle this challenge by focusing on the concept of dispersion. Specifically, we train a neural machine translation (NMT) system and optimize the transformer weights W along with a dispersion regularizer on the decoder embeddings X (tied between the input and the output decoder layers):

$$\mathcal{L}(W, X) = \mathcal{L}_{\text{MT}}(W, X) + \lambda \mathcal{L}_{\text{disp}}(X), \tag{12}$$

where $\mathcal{L}_{\text{MT}}$ is a standard conditional log-likelihood loss of the target sentence given the source sentence accumulated over a dataset (*e.g.*, Bahdanau et al., 2015). We perform stochastic training: for every $\mathcal{L}_{\text{MT}}$ minibatch gradient calculation, we calculate an (independent) stochastic gradient of the regularizer. In particular, for MHE, we estimate the dispersion gradient over a minibatch of 1000 tokens drawn from the whole vocabulary. For Lloyd we use 100 uniform samples from the sphere, while for Sliced we sample one great circle, with no additional subsampling. We use $\gamma = 1$ for RBF. The remaining NMT training settings are standard and provided in Appendix C. We report results on two WMT translation tasks:[4] WMT 2016 Romanian→English (`ro-en`) with 612K training samples and WMT 2019 English→German (`en-de`) with 9.1M training samples (including back-translated data). We measure translation accuracy on the best checkpoint according to validation BLEU score using SacreBLEU (Papineni et al., 2002; Post, 2018) and COMET (Rei et al., 2020).

Table 3 shows the BLEU and COMET results on newstest2016 for `ro-en` and newstest2016 `en-de` along with the dispersion metrics. Surprisingly, simply training the transformer with Riemannian gradient updates

[4] https://www2.statmt.org/

model	ro-en BLEU	ro-en COMET	ro-en $d_{\min}$	ro-en svar	en-de BLEU	en-de COMET	en-de $d_{\min}$	en-de svar
Euclidean baseline	31.4	0.790	0.003	0.19	33.1	0.819	0.000	0.000
spherical baseline	32.2	0.793	0.001	0.57	33.7	**0.825**	0.001	0.408
+MHE (RBF, $d_{\mathbb{R}}$)	32.3	**0.795**	0.001	0.56	**33.9**	**0.825**	0.001	0.410
+Lloyd	**32.4**	0.791	0.001	0.60	33.4	0.822	0.001	0.414
+Sliced	**32.4**	**0.795**	0.435	0.99	33.5	0.820	0.222	0.999

Table 3: newstest2016 `ro-en` and `en-de` results on discrete NMT. Embeddings are 128 dim.

for X (and standard Euclidean gradients for all other parameters) alone already improves dispersion and translation performance over the baseline. Additional dispersion regularization further improves dispersion and BLEU, but not always COMET.

To shed more light into the dispersion effect of Riemannian optimization, we plot the relationship between token frequency and gradient norm (Figure 6a) / minimum pairwise distance (Figure 6b) in the late stage of training. While in the Euclidean model the embeddings of rare words can collapse close to their neighbor, the Riemannian model prevents this. The gradient norm of rare words is an order of magnitude higher with Riemannian gradients than with Euclidean gradients. We hypothesize that this increased gradient norm contributes to better dispersion of rare tokens, thereby mitigating representation collapse. The training dynamics of gradient norms and minimum distances along the earlier training stages, shown in Appendix D, support this intuition.

4.5 Continuous-Output Neural Machine Translation

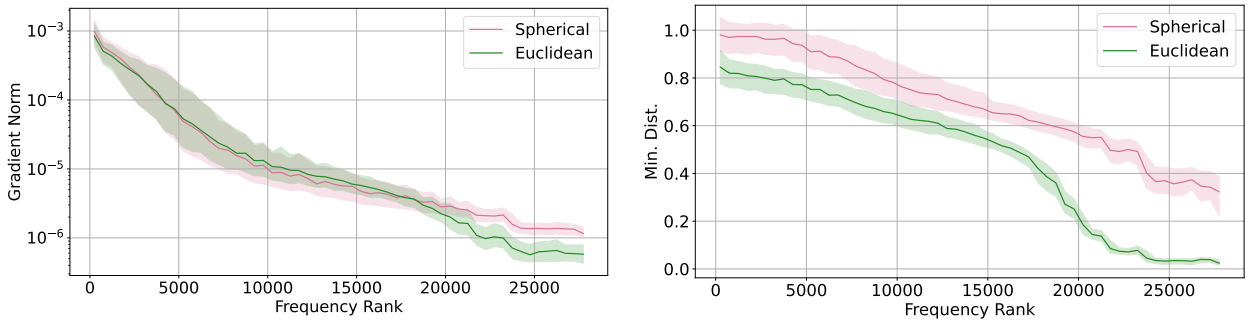
Continuous-output NMT (CoNMT, Kumar & Tsvetkov, 2019) reformulates machine translation as a sequential continuous regression problem of predicting the embedding of the next word, instead of the more usual discrete classification formulation. Tokarchuk & Niculae (2024) recently showed that dispersion plays an important role and greatly impacts performance. We follow closely their setup and apply the dispersion regularizers in order to achieve dispersion. Pre-trained embeddings come from the well-trained discrete models from §4.4 and are frozen while the continuous model is trained from scratch. We present results for WMT 2016 `ro-en` with 612k training samples. Table 4 shows the BLEU score results on `newstest2016` for CoNMT models with different target embeddings X, alongside dispersion measures defined in §2.3

embeddings	svar ↑	$d_{\min}$ ↑	BLEU↑
pretrained $\mathbb{R}^m$ (normalized)	0.191	0.014	29.0
+offline MHE (RBF, $d_{\mathbb{R}}$)	0.599	0.372	29.7
+offline Lloyd	0.585	0.004	27.7
+offline Sliced	0.979	0.106	29.6
pretrained $\mathbb{S}_m$	0.573	0.001	29.9
+MHE (RBF, $d_{\mathbb{R}}$)	0.561	0.001	30.0
+Lloyd	0.596	0.001	**30.1**
+Sliced	**0.999**	**0.435**	30.0

Table 4: Dispersion impact on CoNMT results. We report BLEU scores on the `newstest2016` for `ro-en`. Beam size is equal to 5.

As baselines, we include (normalized) target embeddings from the best Euclidean discrete model (labeled *pretrained* $\mathbb{R}^m$). For *offline* dispersion we optimize starting from the pretrained embeddings for 500 epochs, using batch size 2000 in MHE, RBF $\gamma = 1$, 100 uniform samples in Lloyd, and a single sampled great circle for Sliced. Second, as we have already trained discrete NMT models with spherical dispersion in §4.4, we use the resulting embeddings for CoNMT (rows labeled *pretrained* $\mathbb{S}_m$*)*. All CoNMT models were trained for 50k steps.

Spreading out the projected embeddings generally results in BLEU score improvements. For all dispersion regularizers, we can see that svar is increasing. However, $d_{\min}$ decreases for Lloyd, which may explain the lower BLEU score. Consistent with the synthetic results (§4.2), the Sliced regularizer is best at quickly increasing svar.



(a) Norm of the gradient of the embedding.

(b) Minimum $d_{\mathbb{S}}$ distance to the nearest embedding.

Figure 6: Gradient norms (a) and minimum distances (b) *w.r.t.* NMT embeddings, after 40k steps, for the standard (Euclidean) transformer and the spherical version where decoder embeddings are optimized on $\mathbb{S}_m$. Frequency rank is the index of the token in a sorted vocabulary: higher means less frequent.

5 Additional Related Work

A prominent related but distinct construction is that of **determinantal point processes** (Kulesza & Taskar, 2012), often used to ensure diversity or dispersion in extractive models. Given a set of n points, in the scope of our paper we study optimizing the locations of the points; in contrast, DPPs are intended for selecting subsets of $k \leq n$ such points in a way that encourages dispersion of the selected points. Technically, DPPs induce a probability distribution over all subsets. Like the pairwise regularizers we study in §2, DPPs form an $n \times n$ kernel matrix of pairwise similarities. In order to draw samples, or to evaluate the probability of a given subset, an eigendecomposition (or equivalent) of the kernel matrix is necessary, leading to $O(n^3)$ time complexity for such operations. Maximization, *i.e.*, finding an optimally-dispersed subset under a DPP, is NP-hard to solve exactly, but approximations may be worth exploring.

Dispersion is also closely connected to **contrastive learning** (Chen et al., 2020a; He et al., 2020; Hjelm et al., 2019; Chen et al., 2020b), where model outputs corresponding to different classes are pushed away from each other. Wang & Isola (2020) in particularly showed that widely used contrastive learning objective can be interpreted in terms of "alignment" (similar features for similar samples) and "uniformity" (feature distribution is close to uniform distribution). In our work we focus on parameter dispersion, which can more easily be quantified.

Marbut et al. (2023) explore other ways to quantify dispersion in Euclidean space, and studied their consequences for word embeddings. Exploring spherical extensions thereof is a direction for future work.

6 Conclusion

In this work, we evaluate several dispersion objectives on the hypersphere, including one that is equivalent to the widely used maximum mean discrepancy (MMD) method, as well as two novel approaches: Lloyd and Sliced. We compare these objectives against various pairwise distance-based methods previously explored in the literature. Our experimental results show that these methods can approximate the Tammes problem solution, and also allow improvement on few-shot image classification with prototypes, machine translation and the CoNMT tasks, which uses cosine distance both for training and decoding.

References

Kavosh Asadi and Michael L. Littman. An alternative softmax operator for reinforcement learning. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 243–252. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/asadi17a.html.

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*, January 2015.

Gary Becigneul and Octavian-Eugen Ganea. Riemannian adaptive optimization methods. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=r1eiqi09K7.

Clément Bonet, Paul Berg, Nicolas Courty, François Septier, Lucas Drumetz, and Minh Tan Pham. Spherical Sliced-Wasserstein. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=jXQ0ipgMdU.

Silvere Bonnabel. Stochastic gradient descent on Riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013.

Léon Bottou and Yoshua Bengio. Convergence properties of the KMeans algorithm. In *Proc. of NeurIPS*. 1995. URL http://leon.bottou.org/papers/bottou-bengio-95.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020a.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33: 22243–22255, 2020b.

Henry Cohn. Table of spherical codes. https://dspace.mit.edu/handle/1721.1/153543, 2024. Accessed: 2024-05-28.

John Horton Conway, Neil James Alexander Sloane, and Eiichi Bannai. *Sphere-packings, lattices, and groups*. Grundlehren der mathematischen Wissenschaften 290. Springer, New York, 3rd ed edition, 1999. ISBN 0387985859.

Ludwig Danzer. Finite point-sets on S^2 with minimum distance as large as possible. *Discrete Mathematics*, 60:3–66, 1986. ISSN 0012-365X. doi: https://doi.org/10.1016/0012-365X(86)90002-6. URL https://www.sciencedirect.com/science/article/pii/0012365X86900026.

Nicola De Cao and Wilker Aziz. The power spherical distrbution. *Proceedings of the 37th International Conference on Machine Learning, INNF+*, 2020.

László Fejes Tóth. Über eine Abschätzung des kürzesten Abstandes zweier Punkte eines auf einer Kugelfläche liegenden Punktsystems. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 53:66–68, 1943. URL http://dml.mathdoc.fr/item/GDZPPN002133873.

Aasa Feragen, Francois Lauze, and Soren Hauberg. Geodesic exponential kernels: When curvature and linearity conflict. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3032–3042, 2015.

P Thomas Fletcher, Conglin Lu, Stephen M Pizer, and Sarang Joshi. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE transactions on medical imaging*, 23(8):995–1005, 2004.

Daniel Gabay. Minimizing a differentiable function over a differential manifold. *Journal of Optimization Theory and Applications*, 37:177–219, 1982.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tieyan Liu. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SkEYojRqtm.

Simanta Gautam and Dmitry Vaintrob. A novel approach to the spherical codes problem. 2013. URL https://api.semanticscholar.org/CorpusID:12647839.

Chengyue Gong, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. Frage: Frequency-agnostic word representation. *Advances in neural information processing systems*, 31, 2018.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL http://jmlr.org/papers/v13/gretton12a.html.

Godfrey Harold Hardy, John Edensor Littlewood, and György Pólya. *Inequalities*. Cambridge University Press, 1952.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bklr3j0cKX.

S Rao Jammalamadaka and Ambar Sengupta. *Topics in circular statistics*. Series on multivariate analysis; vol. 5. World Scientific, Singapore, 2001. ISBN 9810237782.

Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Eduardo Blanco and Wei Lu (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2012. URL https://aclanthology.org/D18-2012.

Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2–3):123–286, December 2012. ISSN 1935-8237, 1935-8245. doi: 10.1561/2200000044. URL https://www.nowpublishers.com/article/Details/MAL-044. Publisher: Now Publishers, Inc.

Sachin Kumar and Yulia Tsvetkov. von Mises-Fisher loss for training sequence to sequence models with continuous outputs. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJlDnoA5Y7.

Ya Le and Xuan S. Yang. Tiny ImageNet visual recognition challenge. 2015. URL https://api.semanticscholar.org/CorpusID:16664790.

Alice Le Brigant and Stéphane Puechmorel. Quantization and clustering on riemannian manifolds with an application to air traffic analysis. *Journal of Multivariate Analysis*, 173:685–703, 2019. ISSN 0047-259X. doi: https://doi.org/10.1016/j.jmva.2019.05.008. URL https://www.sciencedirect.com/science/article/pii/S0047259X18303361.

Nikolai N Leonenko. Sample estimate of the entropy of a random vector. *Problemy Peredachi Informatsii*, 23 (2):9–16, 1987.

Alain Lichnewsky. *Minimisation des fonctionnelles définies sur une variété par la methode du gradiënt conjugué*. PhD thesis, These de Doctorat d'Etat. Paris: Université de Paris-Sud, 1979.

Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Weiyang Liu, Rongmei Lin, Z. Liu, Lixin Liu, Zhiding Yu, Bo Dai, and Le Song. Learning towards minimum hyperspherical energy. In *Neural Information Processing Systems*, 2018. URL https://api.semanticscholar.org/CorpusID:43921092.

Weiyang Liu, Rongmei Lin, Zhen Liu, Li Xiong, Bernhard Schölkopf, and Adrian Weller. Learning with hyperspherical uniformity. In Arindam Banerjee and Kenji Fukumizu (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 1180–1188. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/liu21d.html.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742, 2020. doi: 10.1162/tacl_a_00343. URL https://aclanthology.org/2020.tacl-1.47.

Stuart P Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2): 129–137, 1982. doi: 10.1109/TIT.1982.1056489.

David G. Luenberger. The gradient projection method along geodesics. *Management Science*, 18(11):620–631, 1972. ISSN 00251909, 15265501. URL http://www.jstor.org/stable/2629156.

Anna Marbut, Katy McKinney-Bock, and Travis Wheeler. Reliable measures of spread in high dimensional latent spaces. In *International Conference on Machine Learning*, pp. 23871–23885. PMLR, 2023.

Kanti V. Mardia and P. E. Jupp. *Directional statistics*. John Wiley & Sons, Inc., 1 1999. doi: 10.1002/9780470316979. URL https://doi.org/10.1002/9780470316979.

Kantilal Varichand Mardia. Statistics of directional data. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 37(3):349–371, 1975.

Yu Meng, Jiaxin Huang, Guangyuan Wang, Chao Zhang, Honglei Zhuang, Lance Kaplan, and Jiawei Han. Spherical text embedding. In *Advances in neural information processing systems*, 2019.

Pascal Mettes, Elise van der Pol, and Cees G M Snoek. Hyperspherical prototype networks. In *Advances in Neural Information Processing Systems*, 2019.

Oleg R. Musin and Alexey S. Tarasov. The strong thirteen spheres problem. *Discrete and Computational Geometry*, 48(1):128–141, 2 2012. doi: 10.1007/s00454-011-9392-2. URL https://doi.org/10.1007/s00454-011-9392-2.

Oleg R. Musin and Alexey S. Tarasov. The Tammes problem for N = 14. *Experimental Mathematics*, 24(4): 460–468, 2015. doi: 10.1080/10586458.2015.1022842. URL https://doi.org/10.1080/10586458.2015.1022842.

Mattias Nilsson and W. Bastiaan Kleijn. On the estimation of differential entropy from data located on embedded manifolds. *IEEE Transactions on Information Theory*, 53(7):2330–2341, 2007. doi: 10.1109/TIT.2007.899533.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin (eds.), *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040.

Xavier Pennec. Intrinsic statistics on Riemannian manifolds: Basic tools for geometric measurements. *J Math Imaging Vis*, 25:127–154, 2006. URL https://link.springer.com/article/10.1007/s10851-006-6228-4.

Matt Post. A call for clarity in reporting BLEU scores. In Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Lucia Specia, Marco Turchi, and Karin Verspoor (eds.), *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 186–191, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6319. URL https://aclanthology.org/W18-6319.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. COMET: A neural framework for MT evaluation. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2685–2702, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.213. URL https://aclanthology.org/2020.emnlp-main.213.

Raphael M Robinson. Arrangement of 24 points on a sphere. *Mathematische Annalen*, 144:17–48, 1961. URL http://eudml.org/doc/160873.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Spreading vectors for similarity search. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SkGuG2R5tm.

Joseph Salmon and François-David Collin. Soft-max and Soft-argmax, 2024. URL https://josephsalmon.eu/blog/softmax/.

D Sculley. Web-scale K-means clustering. In *Proc. of WWW*, 2010.

Pieter Merkus Lambertus Tammes. *On the origin of number and arrangement of the places of exit on the surface of pollen-grains*. PhD thesis, 1930. Relation: http://www.rug.nl/ Rights: De Bussy.

JJ Thomson. On the structure of the atom: an investigation of the stability and periods of oscillation of a number of corpuscles arranged at equal intervals around the circumference of a circle; with application of the results to the theory of atomic structure. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 7(39):237–265, 1904. doi: 10.1080/14786440409463107. URL https://doi.org/10.1080/14786440409463107.

Evgeniia Tokarchuk and Vlad Niculae. The unreasonable effectiveness of random target embeddings for continuous-output neural machine translation. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 653–662, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-short.56. URL https://aclanthology.org/2024.naacl-short.56.

Daniel Trosten, Rwiddhi Chakraborty, Sigurd Løkse, Kristoffer Wickstrøm, Robert Jenssen, and Michael Kampffmeyer. Hubs and hyperspheres: Reducing hubness and improving transductive few-shot learning with hyperspherical embeddings. pp. 7527–7536, 06 2023. doi: 10.1109/CVPR52729.2023.00727.

Bartel Leendert van der Waerden and Kurt Schütte. Auf welcher kugel haben 5, 6, 7, 8 oder 9 punkte mit mindestabstand eins platz ? *Mathematische Annalen*, 123:96–124, 1951. URL http://eudml.org/doc/160237.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Trung Vu, Raviv Raich, and Xiao Fu. On convergence of projected gradient descent for minimizing a large-scale quadratic over the unit sphere. In *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6. IEEE, 2019.

Lingxiao Wang, Jing Huang, Kevin Huang, Ziniu Hu, Guangtao Wang, and Quanquan Gu. Improving neural language generation with spectrum control. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ByxY8CNtvr.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9929–9939. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/wang20k.html.

Zhennan Wang, Canqun Xiang, Wenbin Zou, and Chen Xu. Mma regularization: Decorrelating weights of neural networks by maximizing the minimal angles, 2021.

Wikipedia contributors. Atan2 — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Atan2&oldid=1247664857, 2024. [Online; accessed 25-September-2024].

Tong Zhang, Wei Ye, Baosong Yang, Long Zhang, Xingzhang Ren, Dayiheng Liu, Jinan Sun, Shikun Zhang, Haibo Zhang, and Wen Zhao. Frequency-aware contrastive learning for neural machine translation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11712–11720, Jun. 2022. doi: 10.1609/aaai.v36i10.21426. URL https://ojs.aaai.org/index.php/AAAI/article/view/21426.

A Appendix

A.1 Kernels and MMD Dispersion

A.1.1 Spherical kernels and relationships

In this section we give a convenient table of the expressions of the kernels used in this paper (Table 5, and additionally show the limit-case relationship to the Tammes problem.

name	distance	hyperparameter branch	expression
RBF	$\mathbb{R}$	$\gamma \in (0, \infty)$	$k(x, y) = \exp(\gamma \langle x, y \rangle)$
RBF	$\mathbb{S}$	$\gamma \in (0, \infty)$	$k(x, y) = \exp\left(-\gamma (\arccos \langle x, y \rangle)^2\right)$
Laplace	$\mathbb{R}$	$\gamma \in (0, \infty)$	$k(x, y) = \exp\left(-\gamma \sqrt{2 - \langle x, y \rangle}\right)$
Laplace	$\mathbb{S}$	$\gamma \in (0, \infty)$	$k(x, y) = \exp(-\gamma \arccos \langle x, y \rangle)$
Riesz	$\mathbb{S}$	$s \in (0, \infty)$	$k(x, y) = (\arccos \langle x, y \rangle)^{-s}$
		$s = 0$	$k(x, y) = -\log \arccos \langle x, y \rangle$
Riesz	$\mathbb{R}$	$s \in (0, \infty)$	$k(x, y) = (2 - \langle x, y \rangle)^{-s/2}$
		$s = 0$	$k(x, y) = -1/2 \log(2 - \langle x, y \rangle)$

Table 5: Expressions of the kernels considered in this work and in related work on hyperspherical dispersion. In all cases, the domain is a sphere $\mathbb{S}^m$. The Euclidean-distance RBF kernel is up to a constant equal to the one in ambient space. Riesz kernels for negative s are sometimes defined as $k_{\text{Riesz},s}(x, y) = -k_{\text{Riesz},-s}(x, y)$.

Limit cases. First, let's consider the RBF and Laplace kernels. For simplicity let R denote the set of pairs of distinct indices from 1 to n, and z_r be the cosine similarity between the rth pair of points in X. Then, the geodesic objective of the Tammes problem can be written as $d_{\min}(X) = \arccos(\max_{r \in R} z_r)$ and since arccos is nondecreasing, the optima of the Tammes problem thus also maximize $\max_{r \in R} z_r$. The "soft maximum" operator log-sum-exp satisfies a well-known limit property (Salmon & Collin, 2024; Asadi & Littman, 2017).

Denoting $z_* = \max_{r\in R} z_r$ and assuming no ties,

$$\begin{aligned}\lim_{\gamma\to\infty} \gamma^{-1} \log \sum_{r\in R} \exp(\gamma z_r) &= \lim_{\gamma\to\infty} \gamma^{-1} \log \sum_{r\in R} \exp(\gamma(z_r - z_* + z_*)) \\ &= \lim_{\gamma\to\infty} \gamma^{-1} \log \left(\exp(\gamma z_*) \sum_{r\in R} \exp(\gamma \underbrace{(z_r - z_*)}_{<0}) \right) \\ &= z_*.\end{aligned} \tag{13}$$

Since log-sum-exp is continuous, in presence of ties, the same limit holds as the ties can be broken by some ε going to zero. Introducing or not a normalization of $\frac{1}{|R|}$ before the sum does not change this result, as $\gamma^{-1} \log Z/|R| = \gamma^{-1} \log Z - \gamma^{-1} \log |R|$ and the latter term goes to zero in the limit. Moreover, any strictly monotonic transformation of the cosine similarities also preserves the optima, which means all versions of the RBF and Laplace kernels shown in Table 5 are equivalent to Tammes in the limit of $\gamma \to \infty$. The limit case of the Riesz kernel as $s \to \infty$ can be proven similarly and is related to the definition of the $\|\cdot\|_\infty$ norm. Let's redefine now $z_r = 1/d_r$ be the inverse of a distance (so still a similarity):

$$\lim_{s\to\infty} \left(\sum_{r\in R} z_r^s \right)^{1/s} = z_* \lim_{s\to\infty} \left(1 + \sum_{r\in R} \underbrace{\left(\frac{z_r}{z_*} \right)^s}_{<1} \right)^{1/s} = z_*. \tag{14}$$

The MHE regularizer with Riesz kernel as $s \to \infty$ is unbounded, but its sth root converges to an objective equivalent to the Tammes problem.

A.1.2 MMD^2 and spherical embeddings: Proof of Lemma 1

The squared MMD of two probability distributions p and q is equal to (Gretton et al., 2012, Lemma 6)

$$\mathrm{MMD}^2[p, q] = \mathbb{E}_{\mathsf{X},\mathsf{X}'\sim p}[k(\mathsf{X}, \mathsf{X}')] - 2\mathbb{E}_{\mathsf{X}\sim p,\mathsf{Y}\sim q}[k(\mathsf{X}, \mathsf{Y})] + \mathbb{E}_{\mathsf{Y},\mathsf{Y}'\sim q}[k(\mathsf{Y}, \mathsf{Y}')].$$

We show that the last two expectations are constant, when p is a distribution on the hypersphere $\mathbb{S}_m$ and q is $\mathrm{Unif}(\mathbb{S}_m)$. Let $z, z' \in \mathbb{S}_m$ and let Q be a rotation matrix such that $Qz = z'$. Note that $\mathsf{Y} \sim \mathrm{Unif}(\mathbb{S}_m)$ if and only if $Q^\top \mathsf{Y} \sim \mathrm{Unif}(\mathbb{S}_m)$, and $\langle Qz, z\rangle = \langle z, Q^\top z\rangle$. It then follows that

$$\mathbb{E}_{\mathsf{Y}\sim\mathrm{Unif}(\mathbb{S}_m)}[k(z, \mathsf{Y})] = \mathbb{E}_{\mathsf{Y}\sim\mathrm{Unif}(\mathbb{S}_m)}[k(z', \mathsf{Y})],$$

since $k(x, y) = f(\langle x, y\rangle)$. Hence, there exists a $c \in \mathbb{R}$ such that for all $z \in \mathbb{S}_m$ we have

$$\mathbb{E}_{\mathsf{Y}\sim\mathrm{Unif}(\mathbb{S}_m)}[k(z, \mathsf{Y})] = c.$$

Consequently, $\mathbb{E}_{\mathsf{X}\sim p,\mathsf{Y}\sim\mathrm{Unif}(\mathbb{S}_m)}[k(\mathsf{X}, \mathsf{Y})] = c$ and $\mathbb{E}_{\mathsf{Y},\mathsf{Y}'\sim\mathrm{Unif}(\mathbb{S}_m)}[k(\mathsf{Y}, \mathsf{Y}')] = c$. The desired result follows immediately.

A.2 Sliced Dispersion: Proofs

A.2.1 Optimal Circle Dispersion

We aim to prove the assertion that the projection

$$\operatorname*{arg\,min}_{\hat{\Theta}\in D_n\mathbb{S}_2} \sum_{i=1}^{n} \frac{1}{2}(\theta_i - \hat{\theta}_i)^2$$

is given by $\hat{\theta}_i^\star = \tau^\star + \phi_{\sigma^{-1}(i)}$, where σ is the permutation such that $\theta_{\sigma(1)} \le \theta_{\sigma(2)} \le \cdots \le \theta_{\sigma(n)}$, and $\tau^\star = \frac{\sum_i \theta_i}{n}$.

By definition, per Eq. (9), $\hat{\Theta} = \tau + \Phi_\sigma$ and thus we may write the problem equivalently as

$$\operatorname*{arg\,min}_{\tau\in[-\pi,\pi),\sigma\in\Pi_n} \sum_i \frac{1}{2}\big(\theta_i - \phi_{\sigma(i)} - \tau\big)^2.$$

Finding the permutation. In terms of σ the objective takes the form $-\sum_i \theta_i \phi_{\sigma(i)} + \text{const}$, so we must find the permutation that maximizes $\sum_i \theta_i \phi_{\sigma(i)} = \sum_i \theta_{\sigma^{-1}(i)} \phi_i$. By the rearrangement inequality (Hardy et al., 1952, Thms. 368–369), since ϕ_i is in ascending order, this sum is maximized when $\theta_{\sigma^{-1}(i)}$ is in ascending order; so the optimal σ must be the inverse of the permutation that sorts Θ.

Finding τ. Ignore the constraints momentarily, and set the gradient of the objective to zero:

$$\frac{\partial}{\partial \tau} \sum_i \frac{1}{2}(\theta_i - \phi_{\sigma(i)} - \tau)^2 = \sum_i (\tau + \phi_{\sigma(i)} - \theta_i) = 0, \quad \text{implying} \quad n\tau = \sum_i \theta_i - \sum_i \phi_i = \sum_i \theta_i,$$

the last equality by choice of the zero-centered reference configuration Φ. Since all $\theta_i \in [-\pi, \pi)$, so is their average, and thus the constraints are satisfied, concluding the proof.

A.2.2 Projection onto a great circle

The projection we seek to compute is

$$\text{proj}_{\mathbb{S}_{pq}}(x) \coloneqq \underset{-\pi \le \theta < \pi}{\arg\min}\, d_{\mathbb{S}}^2(\cos(\theta)p + \sin(\theta)q, x).$$

Since the geodesic distance satisfies $d_{\mathbb{S}}(\cdot, \cdot) = \arccos \langle \cdot, \cdot \rangle$ and arccos is strictly decreasing on $(-1, 1)$, we have

$$\text{proj}_{\mathbb{S}_{pq}}(x) \coloneqq \underset{-\pi \le \theta < \pi}{\arg\max}\, \langle \cos(\theta)p + \sin(\theta)q, x \rangle.$$

As a side note, this shows that it doesn't matter whether we use geodesic or Euclidean distance to define this projection. Setting the gradient to zero yields

$$\cos(\theta)\langle q, x \rangle = \sin(\theta)\langle p, x \rangle,$$

or equivalently $\tan(\theta) = \langle q, x \rangle / \langle p, x \rangle$. The unique solution on $[-\pi, \pi)$ is given by the two-argument arctangent function (arctan2), also known as the argument of complex number $\langle p, x \rangle + i\langle q, x \rangle$ (Wikipedia contributors, 2024).

A.2.3 Gradient of sliced distance

We first compute the Euclidean gradient of the desired expression:

$$\nabla_{x_i} \mathcal{L}_{\text{Sliced}}(X) = \nabla_{x_i} \mathbb{E}_{p,q} \left[d_{\mathbb{S}}^2(\text{proj}_{\mathbb{S}_{pq}}(X), D_n \mathbb{S}_{pq}) \right]. \tag{15}$$

First, by writing

$$d_{\mathbb{S}}^2(\Theta, D_n \mathbb{S}_{pq}) = \min_{\hat{\Theta}} \sum_i \frac{1}{2}(\theta_i - \hat{\theta}_i)^2$$

we see this may be interpreted as a Euclidean projection and

$$\frac{\partial}{\partial \theta_i} d_{\mathbb{S}}^2(\Theta, D_n \mathbb{S}_{pq}) = (\theta_i - \theta_i^\star).$$

But $\theta_i = \text{proj}_{\mathbb{S}_{pq}}(x_i)$ and we can write

$$\begin{aligned} \frac{\partial \theta_i}{\partial x_i} &= \frac{\partial}{\partial x_i} \text{proj}_{\mathbb{S}_{p,q}}(x_i) \\ &= \frac{\partial \theta_i}{\partial x_i} \tan^{-1}\left(\frac{\langle q, x \rangle}{\langle p, x \rangle} \right) \\ &= \frac{\langle p, x \rangle q - \langle q, x \rangle p}{\langle q, x \rangle^2 + \langle p, x \rangle^2}. \end{aligned}$$

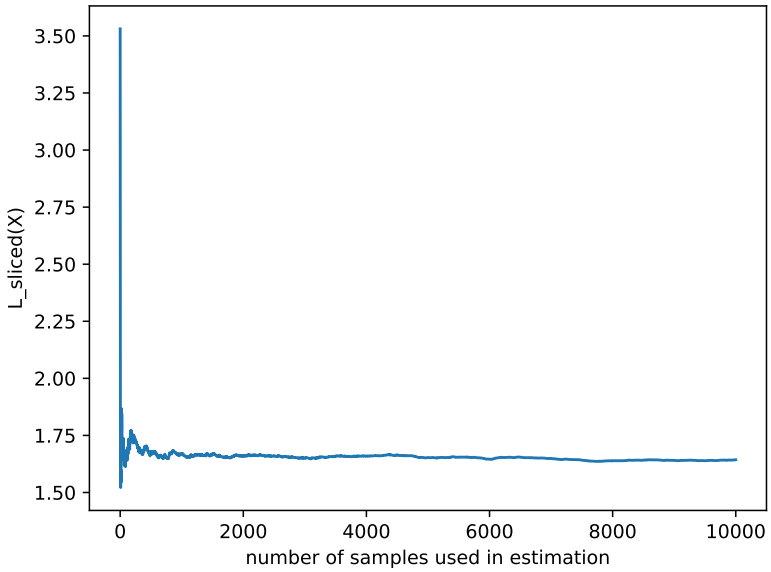


Figure 7: Convergence of the sliced regularizer.

Putting the two together via the chain rule yields

$$\nabla_{x_i} \mathcal{L}_{\text{Sliced}}(X) = \left(\theta_i|_{pq} - \hat{\theta}_i|_{pq}\right) \frac{\langle p, x_i\rangle q - \langle q, x_i\rangle p}{\langle q, x_i\rangle^2 + \langle p, x_i\rangle^2}. \tag{16}$$

Notice that the second term is a vector in $\mathbb{R}^m$ that is orthogonal to x_i because:

$$\langle x_i, \langle p, x_i\rangle q - \langle q, x_i\rangle p\rangle = \langle p, x_i\rangle\langle q, x_i\rangle - \langle q, x_i\rangle\langle p, x_i\rangle = 0.$$

Therefore,

$$\text{grad}_{x_i} \mathcal{L}_{\text{Sliced}}(X) = \nabla_{x_i} \mathcal{L}_{\text{Sliced}}(X).$$

A.3 Sample Convergence of the Sliced Regularizer

For better insight into the number of great-circle samples required to estimate $\mathcal{L}_{\text{Sliced}}$ in high dimensions, we perform a numerical experiment. Fixing X to be a uniformly-sampled configuration of $n = 10k$ points in dimension $m = 128$, we draw up to $10k$ uniformly-sampled great circles and report the Monte Carlo estimated average objective in Figure 7. The value converges relatively quickly and 1000 samples seem more than enough. In experiments, we find it sufficient to draw a single sample per update, because we perform sufficiently many updates.

B Riemannian vs Euclidean optimization

We compare the results of optimal angle approximation using projected gradient Adam $\mathbb{R}^m$ against Riemannian Adam over $\mathbb{S}_m$. All other parameters are the same as described in §4.1. Results are shown in Figure 8. However, for all other methods except KoLeo we can see that Riemannian optimization performs better overall.

C Neural Machine Translation: Experimental Setup

For subword tokenization we used the same SentencePiece (Kudo & Richardson, 2018) model, specifically the one used in the MBart multilingual model (Liu et al., 2020). This choice allows for unified preprocessing

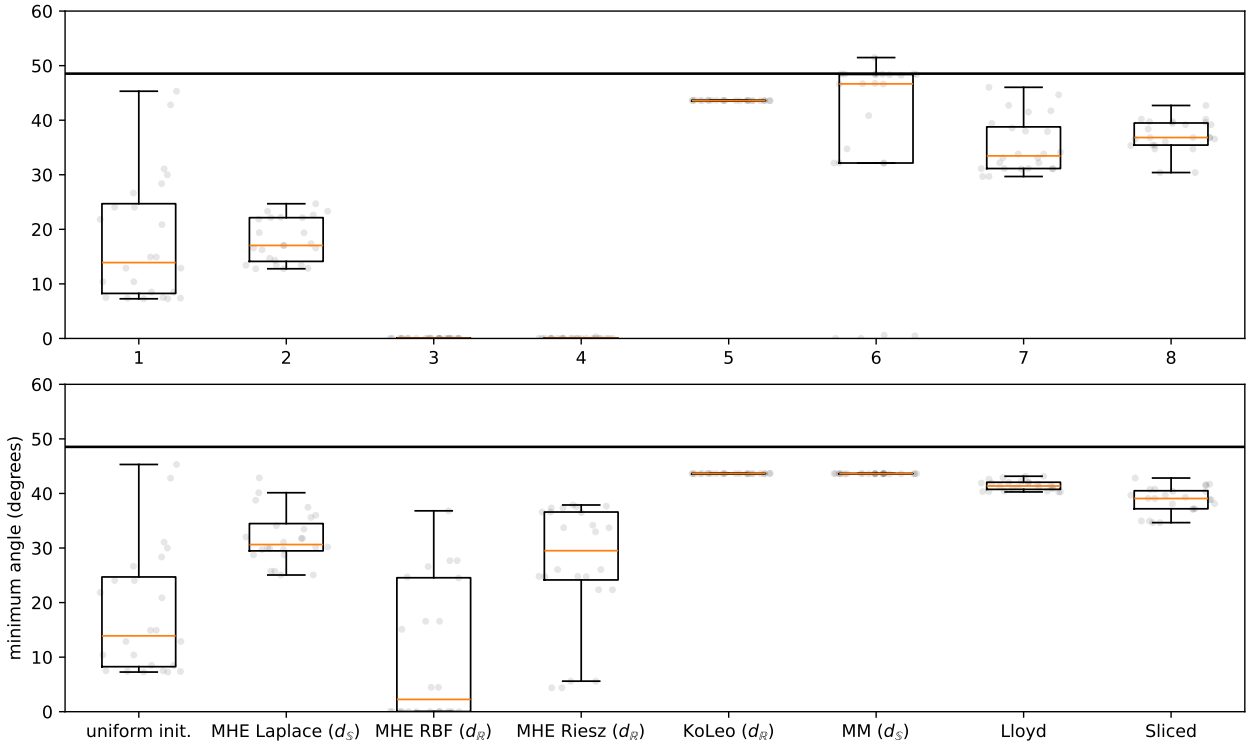


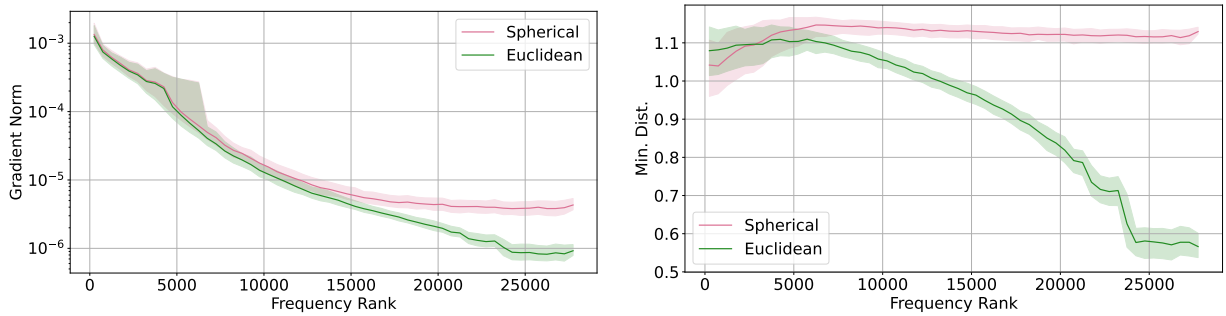
Figure 8: Minimum angles (degrees) for each of the N=24 points with respect to optimization methods. Top: projected gradient in Euclidean space. Bottom: Riemannian gradient on the sphere.

for all languages we cover. We used `fairseq` (Ott et al., 2019) framework for training our models. Baseline discrete models (Euclidean baseline) are trained with cross-entropy loss, label smoothing equal to 0.1 and effective batch size 65.5K tokens. All models are trained with learning rate $5 \cdot 10^{-4}$ and 10k warm-up steps for 50k steps in total. Spherical baseline and models with dispersion regularizer are trained by defining decoder's embeddings layer as a manifold parameter and using Riemannian Adam (Becigneul & Ganea, 2019) with learning rate $5 \cdot 10^{-3}$. We used SacreBLEU (Post, 2018) with the following signature `nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.3.1` and COMET (Rei et al., 2020) with `unbabel-comet` library version 2.2.2[5] and `Unbabel-wmt22-comet-da` model.

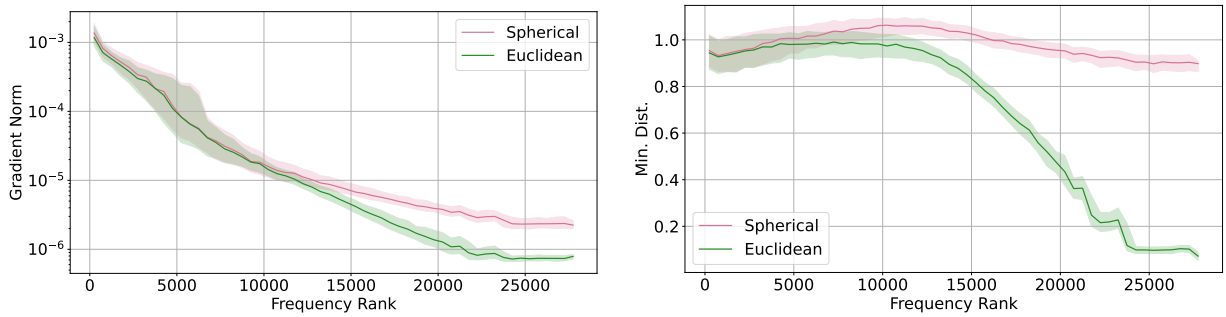
D Neural Machine Translation: Gradient Norms

We show in Figure 9 how gradient norms and minimum distances of target language embeddings vary throughout the training process. Note that at the step=0, the norms and minimum distances are the same.

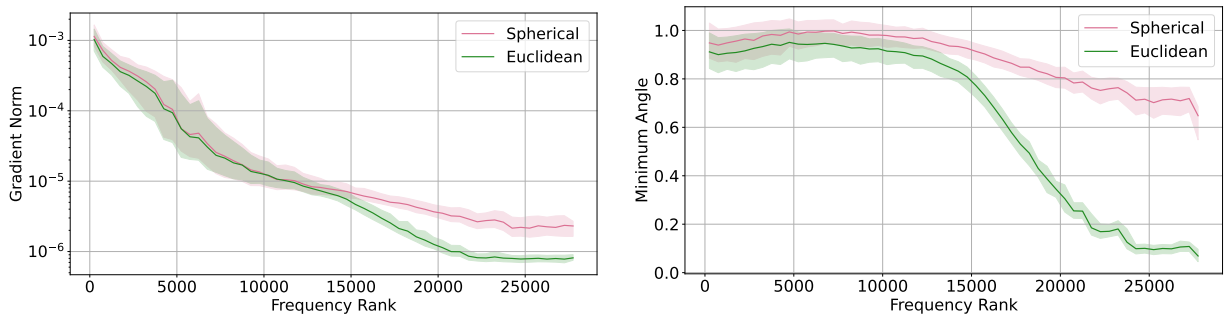
[5] https://github.com/Unbabel/COMET



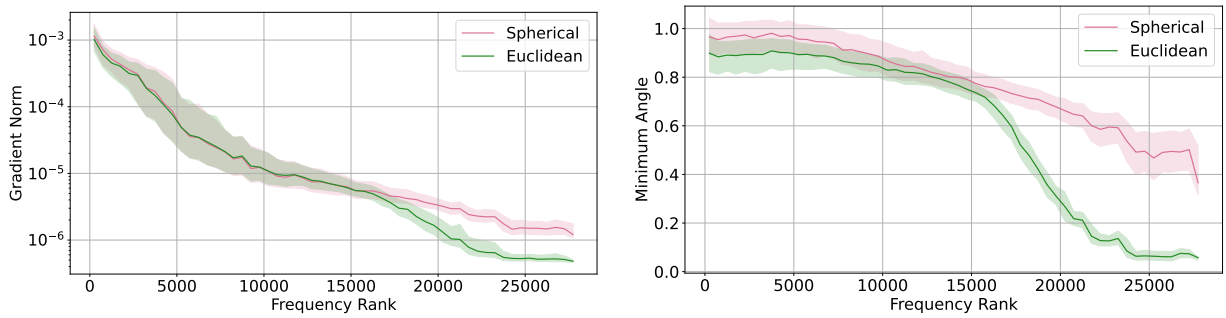
(a) Gradient norms and minimum distance for step=4000



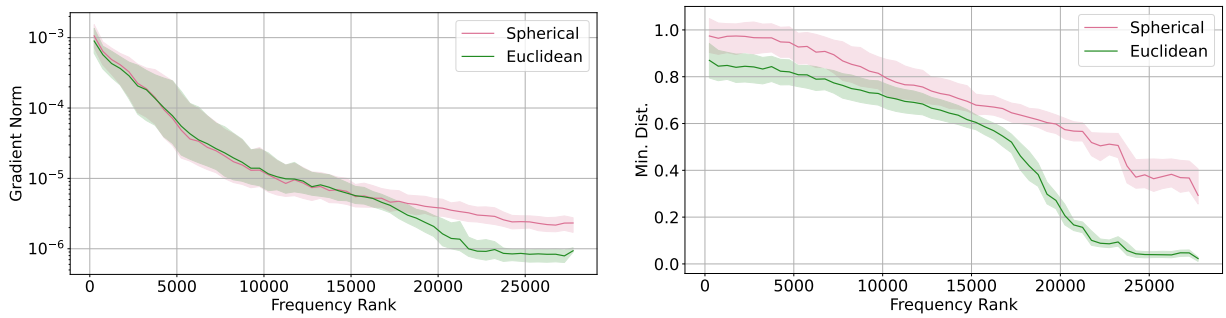
(b) Gradient norms and minimum distance for step=8000



(c) Gradient norms and minimum distance for step=12000



(d) Gradient norms and minimum distance for step=20000



(e) Gradient norms and minimum distance for step=32000

Figure 9: Training dynamic of gradient norms and minimum distances of the target language embeddings.