On the Matter of Embeddings Dispersion on Hyperspheres

Evgeniia Tokarchuk [1] **Hua Chang Bakker** [1] **Vlad Niculae** [1]

Abstract

Dispersion of the embeddings on the d-dimensional hypersphere is a process of finding a configuration that preserves semantic information while pushing unrelated vectors away from each other without the need for negative examples. Such a formulation can be connected to the finding configuration of the points such that the minimum distance between two distinct points is maximal, which is a well-known open mathematical problem called the Tammes problem. When dealing with high-dimensional spaces and extremely large numbers of points, as in the text embeddings learning, there is typically no optimal solution, contrary to the Tammes problem, where the optimal solution exists for particular values of N and d. Moreover, embeddings learning is mostly done in Euclidean space, which is at odds with the goal of directional dispersion. In this work, we revisit existing algorithms and propose new ones to find a sub-optimal solution for embeddings dispersion by defining the Riemannian optimization problem on the hypersphere.

1. Introduction

Dispersion[2] of the embeddings encouraging spreading out a large amount of high-dimensional embeddings vectors on the surface of the d-dimensional unit hypersphere (Liu et al., 2021). The need for separation is motivated by the recently found issues with clumping embeddings together while using angular distances. Such clustering of the points negatively impacts the performance of the downstream tasks (Wang & Isola, 2020; Liu et al., 2021; Trosten et al., 2023; Tokarchuk & Niculae, 2023). Mettes et al. (2019) also argue that directly minimized maximum similarity of the points on the hypersphere is superior to uniformly obtained samples (Hicks & Wheeling, 1959; Muller, 1959), since it encourages separation between points. Having embeddings explicitly modeled to be on the unit hypersphere enables making use of dispersion (Liu et al., 2021) and is in line with using angular distance.

In general, the problem of spreading N points on the surface of d dimensional sphere, such that the angular distance between two distinct points is maximal, is an open mathematical problem known as the Tammes problem (Tammes, 1930). The optimal solutions for this problem are known for small values of d and N (Fejes, 1943; Danzer, 1986; Waerden van der & Schütte, 1951; Robinson, 1961; Musin & Tarasov, 2012; 2015). The Tammes problem can also be formulated as a problem of finding a spherical code (Conway et al., 1999) with minimal cosine similarity value for given d and N. There are known numerical solutions for several values of N and d for spherical codes problem (Cohn, 2024). However, we typically deal with a large number of dimensions and many points when learning, e.g., text embeddings for ML tasks. Thus, we can rely on gradient optimization methods to approximate the optimal configuration on the hypersphere.

We study several regularization terms in order to find a sub-optimal solution to the dispersion problem on the unit hypersphere. In particular, we reinterpret Maximum Mean Discrepancy (MMD, Gretton et al., 2012a) as a method for dispersing an arbitrary number of high-dimensional points, adapt Lloyd's algorithm (Lloyd, 1982), and propose sliced dispersion that directly exploits properties of the hypersphere. We showcase the performance of those regularizing by approximating optimal Tammes problem solutions and learning dispersed target embeddings for continuous-output neural machine translation (CoNMT). We chose CoNMT since recently Tokarchuk & Niculae (2023) showed that it is fairly sensitive to the dispersion of the target embeddings.

2. Dispersion on the Hypersphere

2.1. Notation and Background

We denote by $\mathbb{S}_d$ the d-dimensional hypersphere embedded in $\mathbb{R}^{d+1}$, *i.e.*, $\mathbb{S}_d = \{x \in \mathbb{R}^{d+1} \mid \|x\| = 1\}$. For $u, v \in$

[1] University of Amsterdam, Amsterdam, The Netherlands. Correspondence to: Evgeniia Tokarchuk <e.tokarchuk@uva.nl>.

Proceedings of the Geometry-grounded Representation Learning and Generative Modeling at 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR Vol Number, 2024. Copyright 2024 by the author(s).

[2] In the literature, the term "uniformity" is also used. However, to highlight the difference with samples from the Uniform distribution, we use "dispersion" instead.

$\mathbb{R}^{d+1}$ we denote their Euclidean inner product by $\langle u, v\rangle := \sum_{i=1}^{d+1} u_i v_i$. The hypersphere is an embedded Riemannian submanifold of $\mathbb{R}^{d+1}$. The tangent space of the sphere at a point x is $T_x\mathbb{S}_d := \{v \in \mathbb{R}^{d+1} \mid \langle x, v\rangle = 0\} \simeq \mathbb{R}^d$, and the Riemannian inner product on it is inherited from $\mathbb{R}^{d+1}$, *i.e.*, for $u, v \in T_x\mathbb{S}_d$, $\langle u, v\rangle_x := \langle u, v\rangle$. The geodesic distance on a hypersphere is $d(x, x') = \cos^{-1}(\langle x, x'\rangle)$. As a special case, for $d = 1$ it is more convenient to work in an isomorphic angular parametrization, *i.e.*, $\mathbb{S}_1 \simeq \{\theta \mid -\pi \leq \theta < \pi\}$ with $d(\theta, \theta') = |\theta - \theta'|$: the embedding of $\mathbb{S}_1$ into $\mathbb{R}^2$ is given by $\theta \to (\cos\theta, \sin\theta)$. We reserve the use of Greek letters τ, θ, ϕ for 1-d angles. We denote by Π_n the set of permutations of $(1, \dots, n)$.

We use roman capitals, *i.e.*, $X = (x_1, \dots, x_n)$, to denote an (ordered) collection, or configuration, of n points on the same sphere, *i.e.*, each $x_i \in \mathbb{S}_d$. We use sans-serif capitals, *i.e.*, Y, to denote a random variable.

2.2. Measures of Dispersion

To measure the dispersion of the set of embeddings X on unit hypersphere, we consider two different metrics.

Minimum distance. Dispersion requires that no two points be too close, suggesting a minimum distance metric:

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

where $d(x_i, x_j)$ is the geodesic distance from §2.1.

Spherical variance. Spherical variance (Jammalamadaka & Sengupta, 2001; Mardia, 1975) originates from directional statistics and is defined for finite $X \subseteq \mathbb{S}_d$ as

$$\mathrm{svar}(X) = 1 - \overline{R}, \text{ where } \overline{R} = 1/n \sum_i x_i, \tag{2}$$

Spherical variance is a key quantity in the Raleigh test for uniformity on the hypersphere $\mathbb{S}_d$ (Mardia & Jupp, 1999, p. 206–208), which uses $(d + 1)n\overline{R}^2$ as test statistic.

The presented dispersion measures offer complementary perspectives of the dispersion of the embeddings, but are insufficient when considered in isolation. The minimum distance only depends on the two closest embeddings: embeddings can be spread out in a near perfect configuration, whilst having a minimum distance close to zero. Similarly, large spherical variance does not imply well dispersed embeddings (consider embeddings clustered around two antipodes.) In addition, neither method is well-suited for gradient optimization. The gradient of $d_{\min}$ depends only on the closest pair of points and would lead to impractically slow algorithms. As for spherical variance, since the Euclidean gradient of $\overline{R}$ is orthogonal to the surface of the hypersphere $\mathbb{S}_d$, its Riemannian gradient is null.

2.3. MMD/Kernel Entropy

The distribution of perfectly dispersed embeddings is similar to a uniform distribution on the hypersphere. Dispersing embeddings can then be seen as minimizing the 'distance' between the embedding distribution and the uniform distribution $\mathrm{Unif}(\mathbb{S}_d)$. The Raleigh test for uniformity is not well suited for this purpose as discussed in the previous section. An alternative statistical test for uniformity can be derived from the maximum mean discrepancy (MMD), which measures the distance between two probability distributions (Gretton et al., 2012b). Lemma 1 implies that the squared MMD between the distribution of the embeddings and the uniform distribution on the sphere can be computed using embeddings only, up to a constant.

Lemma 1 (MMD^2 **and spherical embeddings.)** *Let p be any distribution on $\mathbb{S}_d$ and let k be a kernel on $\mathbb{S}_d$such that $k(x, y) = f(\langle x, y\rangle)$ for some function $f\colon [-1, 1] \to \mathbb{R}$. Assume all random variables are independent. Up to a normalizing constant $c \in \mathbb{R}$, we have*

$$\mathrm{MMD}^2[p, \mathrm{Unif}(\mathbb{S}_d)] = \mathbb{E}_{\mathsf{X}, \mathsf{X}' \sim p}\left[k(\mathsf{X}, \mathsf{X}')\right] - c.$$

The proof of Lemma 1 is deferred to Appendix A.1.

Using the radial basis function kernel $k(x, y) = \exp\left(-\lambda \left\|x - y\right\|^2\right)$ in the result of Lemma 1, we see that minimizing the estimated squared MMD of the embeddings and the uniform distribution is equivalent to minimizing

$$M(X) = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ i \neq j}}^{n} \exp\left(\gamma \langle x_i, x_j\rangle\right), \tag{3}$$

where $X \subseteq \mathbb{S}_d$ is a set of n embeddings and $\gamma := 2\lambda > 0$. The intuition for $M(X)$ is that the embeddings are pushed away from each other when minimizing $M(X)$, thereby improving the uniformity of the embedding distribution. The parameter γ determines the emphasis on the distance between embeddings, *i.e.*, a larger γ results in a larger emphasis on close embeddings.

The regularizer $M(X)$ is related to the partial loss function used by Trosten et al. (2023) to disperse image representation embeddings for few shot learning, as well as the energy-based approaches to Tammes and Thompson problem (Gautam & Vaintrob, 2013). In particular, the exponential of the energy optimized by Trosten et al. (2023); Wang & Isola (2020) differs from M by a constant. Our work thus provides a perhaps more direct justification of their objective.

2.4. Lloyd's Algorithm

An alternative formulation comes from casting maximal dispersion as *quantization* of a uniform measure. Quantization refers to the problem of approximating a given measure by an empirical measure supported at a few centers. When the given measure is uniform over some support set, the optimal centers are spread out uniformly over the support; and can be calculated by Lloyd's algorithm (Lloyd, 1982), henceforth *Lloyd*, which iteratively moves each centroid to the center of mass of its Voronoi cell. When the given measure is another empirical measure, quantization is equivalent to *k-means clustering*. When the space is Riemannian and not Euclidean, both quantization and clustering generalize readily 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]

$$L(X) = \mathbb{E}_{\mathsf{Y}\sim \mathrm{Unif}(\mathbb{D})}\left[\min_{j\in[n]} \frac{1}{2} d^2(\mathsf{Y}, x_j)\right]. \tag{4}$$

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 $\mathbb{D} = \mathbb{S}_d$. While traditionally Lloyd's algorithm corresponds to minimizing $L(X)$ alone, we propose using $L(X)$ 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, in which case most cluster centers are not updated in an iteration. However, unlike for MMD, the stochastic gradient takes into account all of X through the cluster assignment.

2.5. 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 2 or more dimensions it is hard to find the location of n evenly distributed points, on $\mathbb{S}_1$ this can be done efficiently: The following set of angles is one optimal configuration:

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

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 $\Phi_\sigma = (\phi_{\sigma(1)}, \dots, \phi_{\sigma(n)})$. We can then write the set of all possible ordered optimally-dispersed configurations as

$$D_n\mathbb{S}_1 := \{\tau + \Phi_\sigma \mid \tau \in (-\pi, \pi), \sigma \in \Pi_n\}. \tag{5}$$

Given an ordered configuration of angles $\Theta = (\theta_1, \dots, \theta_n) \subset \mathbb{S}_1$, we define its (angular) distance to the maximally-dispersed set as:

$$d^2(\Theta, D_n\mathbb{S}_1) = \min_{\hat{\Theta}\in D_n\mathbb{S}_1} \sum_{i=1}^{n} \frac{1}{2}(\theta_i - \hat{\theta}_i)^2. \tag{6}$$

Lemma 3 defined and proved in Appendix A.2 shows that any configuration of angles can be efficiently projected to its nearest maximally-dispersed configuration. We defer all proofs in this section to Appendix A.2.

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 2 (Projection onto great circle.) *Let $p, q \in \mathbb{S}_d$ with $\langle p, q\rangle = 0$. Two such vectors determine a unique great circle $\mathbb{S}_{pq} \subset \mathbb{S}_d$ defined by:*

$$\mathbb{S}_{pq} := \{\cos(\theta)p + \sin(\theta)q \mid -\pi \le \theta < \pi\} \simeq \mathbb{S}_1.$$

The nearest point on $\mathbb{S}_{pq}$ to a given $x \in \mathbb{S}_d$ is:

$$\mathrm{proj}_{\mathbb{S}_{pq}}(x) = \mathrm{arctan2}\left(\langle x, q\rangle, \langle x, p\rangle\right). \tag{7}$$

A well-dispersed configuration over $\mathbb{S}_d$ should remain fairly well-dispersed along any slice on average. If we denote $\mathrm{proj}_{\mathbb{S}_{pq}}(X) := (\mathrm{proj}_{\mathbb{S}_{pq}}(x_1), \dots \mathrm{proj}_{\mathbb{S}_{pq}}(x_n))$, we may capture this intention by the following measure:

$$S(X) = \mathbb{E}_{p,q}\left[d^2(\mathrm{proj}_{\mathbb{S}_{pq}}(X), D_n\mathbb{S}_{pq})\right], \tag{8}$$

where d^2 is defined in eq. (6), and the expectation is over orthogonal pairs p, q. The following proposition efficiently computes stochastic gradients of S.

Proposition 1 *Denote $\theta_i^{pq} = \mathrm{proj}_{\mathbb{S}_{pq}}(x_i)$, and $\hat{\theta}_i^{\star pq}$ the corresponding dispersion maximizer computed using Lemma 1. The Riemannian gradient of S is given by:*

$$\mathrm{grad}_{x_i} S(X) = \mathbb{E}_{p,q}\left[(\theta_i^{pq} - \hat{\theta}_i^{\star pq})\frac{\langle x_i, p\rangle q - \langle x_i, q\rangle p}{\langle x_i, q\rangle^2 + \langle x_i, p\rangle^2}\right].$$

[3] More generally, the target measure need not be uniform. Le Brigant & Puechmorel (2019) discuss more general conditions for the existence of a minimizer.

3. Experimental Results

3.1. Approximate Solution for Tammes Problem

We evaluate our proposed dispersion methods by approximating the known solution to the Tammes problem for $N = 14$ (Musin & Tarasov, 2015) in three dimensions, by considering the minimum angle between points of the optimal configuration. Uniformly sampled points were dispersed using the proposed in Section §2 regularizers using Riemannian Adam for 2.5k epochs. Parameters for individual regularizers can be found in Appendix B.1.

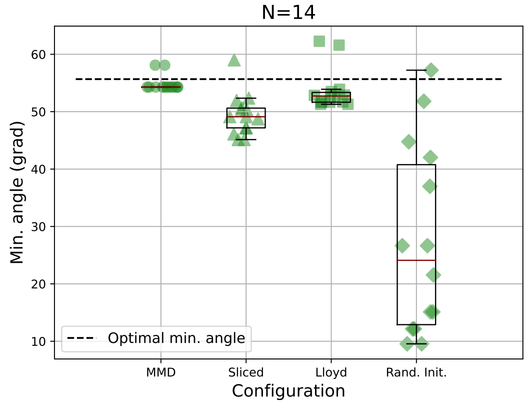

Figure 1. Minimum angles distributions for various points arrangements with d=3 and N=14. `Optimal Solution` shows the angle for known optimal solution equal to 55.6705700°. `Random Init.` represents points drawn from the uniform distribution on the sphere. `MMD`, `Sliced` and `LLoyd` are performed over the `Random Init.`

The minimum angles of the points distributed using the MMD regularizer are close to the optimal minimum angle as shown in Figure 1. The Lloyd regularizer follows closely, but seems to approximate the solutions less accurately. The sliced dispersion regularizer, however, seems to approximate the solutions worse than the other two regularizers for $N = 14$. Results of approximation on other values of N and d can be found in Appendix B.1

3.2. 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 (2023) 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 good dispersion. Pre-trained embeddings comes from the well-trained discrete model. We present results for WMT 2016 `ro-en` with 612k training samples. Table 1 shows the BLEU score results on `newstest2016` for CoNMT models with different target embeddings $\mathbf{E_Y}$, alongside dispersion measures defined in §2.2

We conduct two types of experiments. First we train a vanilla transformer model (Vaswani et al., 2017). Resulting embeddings are in Euclidean space, so we project it onto the sphere by dividing to the norms of embeddings. To spread out the embeddings we then use Riemannian optimization on the sphere with `geoopt` (Kochurov et al., 2020) using three different regularizers. We refer to this as 'offline' methods in Table 1. Second, we train transformer model with embeddings explicitly modeled to be on the sphere using Riemannian optimization. In this case, we can apply dispersion regularizers directly during optimization. Appendix B.2.1 contains further embedding training details.

Tgt. Emb. $\mathbf{E_Y}$	$\mathrm{svar}(\mathbf{E_Y}) \uparrow$	$d_{\min}(\mathbf{E_Y}) \uparrow$	BLEU↑
euclidean (proj.)	0.191	0.014	27.8
+offline MMD	0.599	0.372	29.7
+offline Lloyd	0.585	0.004	27.7
+offline Sliced	0.979	0.106	29.6
spherical	0.797	0.014	29.8
+MMD	0.799	0.014	**29.9**
+Lloyd	0.799	0.026	29.8
+Sliced	**0.999**	**0.471**	**29.9**

Table 1. Impact of the dispersion of the target embeddings on the CoNMT results. We report BLEU scores on the `newstest2016` for `ro-en`. Beam size is equal to 5.

Spreading out the projected embeddings results into the BLEU score improvement with MMD and Sliced dispersion. For all dispersion regularizers, we can see that $\mathrm{svar}(\mathbf{E_Y})$ is increasing. However, $d_{\min}(\mathbf{E_Y})$ decreases for the Lloyd regularizer, which seemingly also impacts the BLEU score.

We also highlight that training target embeddings on the hypersphere with Riemannian optimization, even without any regularizer, gives better dispersion "by default" according to spherical variance. When adding dispersion regularizers, there are no significant fluctuations in $\mathrm{svar}(\mathbf{E_Y})$, except for the Sliced regularizer. We leave thorough investigation of the observed behaviour for the future work.

4. Conclusion

Our work presents and compares three regularizers for dispersing points on the sphere; one equivalent to a popular method (MMD) and two novel ones (Lloyd and Sliced). Our experimental results show that these methods can approximate the Tammes problem solution, and also allow improvement on the CoNMT task, which uses cosine distance both for training and decoding. We want to extend the variety of the task in the future.

Acknowledgements

We thank all the members of the UvA Language Technology Lab for their valuable feedback on our work. We also thank SURF (www.surf.nl) for the support in using the National Supercomputer Snellius. This work was partly supported by the Dutch Research Council (NWO) via VI.Veni.212.228 and the European Union's Horizon Europe research and innovation programme via UTTER 101070631.

References

Bonet, C., Berg, P., Courty, N., Septier, F., Drumetz, L., and Pham, M. T. Spherical sliced-wasserstein. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=jXQ0ipgMdU.

Bottou, L. and Bengio, Y. Convergence properties of the kmeans algorithm. In *Proc. of NeurIPS*. 1995. URL http://leon.bottou.org/papers/bottou-bengio-95.

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

Conway, J. J. H., Sloane, N. N. J. A., and Bannai, E. *Sphere-packings, lattices, and groups*. Grundlehren der mathematischen Wissenschaften 290. Springer, New York [etc, 3rd ed edition, 1999. ISBN 0387985859.

Danzer, L. Finite point-sets on s2 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.

Fejes, L. 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.

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

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *J. Mach. Learn. Res.*, 13(null):723–773, mar 2012a. ISSN 1532-4435.

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

Hardy, G. H., Littlewood, J. E., , and Pólya, G. *Inequalities*. Cambridge University Press, 1952.

Hicks, J. S. and Wheeling, R. F. An efficient method for generating uniformly distributed points on the surface of an n-dimensional sphere. *Commun. ACM*, 2(4):17–19, apr 1959. ISSN 0001-0782. doi: 10.1145/377939.377945. URL https://doi.org/10.1145/377939.377945.

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

Kochurov, M., Karimov, R., and Kozlukov, S. Geoopt: Riemannian optimization in pytorch, 2020.

Kudo, T. and Richardson, J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Blanco, E. and Lu, W. (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.

Kumar, S. and Tsvetkov, Y. 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.

Le Brigant, A. and Puechmorel, S. 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.

Liu, W., Lin, R., Liu, Z., Xiong, L., Schölkopf, B., and Weller, A. Learning with hyperspherical uniformity. In Banerjee, A. and Fukumizu, K. (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.

Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., and Zettlemoyer, L. 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.

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

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

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

Mettes, P., van der Pol, E., and Snoek, C. G. M. Hyperspherical prototype networks. In *Advances in Neural Information Processing Systems*, 2019.

Muller, M. E. A note on a method for generating points uniformly on n-dimensional spheres. *Commun. ACM*, 2(4):19–20, apr 1959. ISSN 0001-0782. doi: 10.1145/377939.377946. URL https://doi.org/10.1145/377939.377946.

Musin, O. R. and Tarasov, A. S. 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.

Musin, O. R. and Tarasov, A. S. 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.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In Isabelle, P., Charniak, E., and Lin, D. (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.

Post, M. A call for clarity in reporting BLEU scores. In Bojar, O., Chatterjee, R., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Huck, M., Yepes, A. J., Koehn, P., Monz, C., Negri, M., Névéol, A., Neves, M., Post, M., Specia, L., Turchi, M., and Verspoor, K. (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.

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

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

Tammes, P. *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.

Tokarchuk, E. and Niculae, V. The unreasonable effectiveness of random target embeddings for continuous-output neural machine translation, 2023.

Trosten, D., Chakraborty, R., Løkse, S., Wickstrøm, K., Jenssen, R., and Kampffmeyer, M. 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.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Waerden van der, B. and Schütte, K. 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.

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

A. Appendix

A.1. MMD Dispersion: Proofs

A.1.1. 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., 2012b, 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 $\mathbb{S}_d$ and q is $\mathrm{Unif}(\mathbb{S}_d)$. Let $z, z' \in \mathbb{S}_d$ and let Q be a rotation matrix such that $Qz = z'$. Note that $\mathsf{Y} \sim \mathrm{Unif}(\mathbb{S}_d)$ if and only if $Q^\top \mathsf{Y} \sim \mathrm{Unif}(\mathbb{S}_d)$, and $\langle Qz, z\rangle = \langle z, Q^\top z\rangle$. It then follows that

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

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

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

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

A.2. Sliced Dispersion: Proofs

A.2.1. OPTIMAL 1-D DISPERSION

Lemma 3 *Optimal 1-d dispersion. The projection*

$$\operatorname*{arg\,min}_{\hat\Theta\in D_n\mathbb{S}_1} \sum_{i=1}^{n} \frac{1}{2}(\theta_i - \hat\theta_i)^2$$

is given by $\hat\theta_i^\star = \tau^\star + \phi_{\sigma^{-1}(i)}$, where σ is the permutation s.t. $\theta_{\sigma(1)} \leq \theta_{\sigma(2)} \leq \ldots \leq \theta_{\sigma(n)}$, and $\tau^\star = \frac{\sum_i \theta_i}{n}$. The projection can be calculated in $O(n \log n)$, the dominating cost being sorting the angles.

We aim to prove the assertion that the projection

$$\operatorname*{arg\,min}_{\hat\Theta\in D_n\mathbb{S}_1} \sum_{i=1}^{n} \frac{1}{2}(\theta_i - \hat\theta_i)^2$$

is given by $\hat\theta_i^\star = \tau^\star + \phi_{\sigma^{-1}(i)}$, where σ is the permutation st $\theta_{\sigma(1)} \leq \theta_{\sigma(2)} \leq \ldots \leq \theta_{\sigma(n)}$, and $\tau^\star = \frac{\sum_i \theta_i}{n}$.

By definition, per eq. (5), $\hat\Theta = \tau + \Phi_\sigma$ and thus we may write the problem equivalently as

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

Finding the permutation. In terms of σ the objective takes the form $-\sum_i \theta_i\phi_{\sigma(i)}$ +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

$$\mathrm{proj}_{\mathbb{S}_{pq}}(x) := \underset{-\pi\leq\theta<\pi}{\arg\min}\, d^2((\cos(\theta)p + \sin(\theta)q, x).$$

Since the geodesic distance satisfies $d^2(\cdot,\cdot) = \arccos\langle\cdot,\cdot\rangle$ and $\arccos$ is strictly decreasing on $(-1, 1)$, we have

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

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 arctan2 function.

A.2.3. Gradient of sliced distance

We first compute the Euclidean gradient of the desired expression:

$$\nabla_{x_i} S(X) = \nabla_{x_i}\mathbb{E}_{p,q}\left[d^2(\mathrm{proj}_{\mathbb{S}_{pq}}(X), D_n\mathbb{S}_{pq})\right]. \tag{9}$$

First, by writing

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

we see this may be interpreted as an Euclidean projection and

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

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

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

Putting the two together via the chain rule yields

$$\nabla_{x_i} S(X) = (\theta_i^{pq} - \hat{\theta}_i^{\star pq})\frac{\langle p, x_i\rangle q - \langle q, x_i\rangle p}{\langle q, x_i\rangle^2 + \langle p, x_i\rangle^2}. \tag{10}$$

Notice that the second term is a vector in $\mathbb{R}^{d+1}$ 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,

$$\mathrm{grad}_{x_i} S(X) = \nabla_{x_i} S(X).$$

B. Applications

B.1. Approximation of the Tammes problem solution

The MMD regularizer was minimized using maximum batch size, learning rate $5 \cdot 10^{-2}$; $\gamma = 20$. The sliced dispersion regularizer used a single randomly generated axis during each epoch. The Lloyd regularizer was used with 200 samples. Both the sliced dispersion regularizer and the Lloyd regularizer were used with learning rate $5 \cdot 10^{-3}$.

We report results of approximating the Tammes problem solution in 3-dimensions with $N = (13, 14, 24)$. As shown in Figure 2.

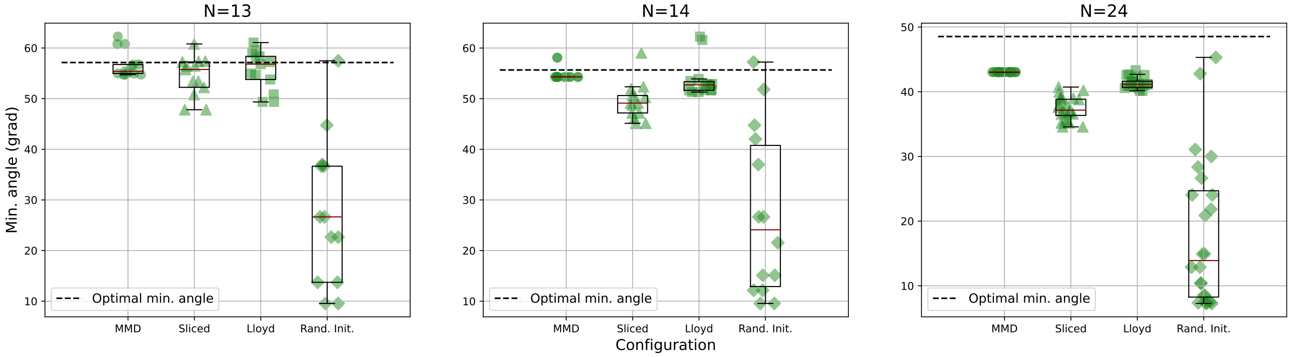

Figure 2. Minimum angles distributions for various points arrangements with d=3 and N={13,14,24}. `Optimal Solution` shows the angle for known optimal solution equal to 57.1367031° for N=13, 55.6705700° for N=14 and 48.53529763° for N=24. `Rand. Init.` represents the points generated uniformly at random on the surface of the sphere. `MMD`, `Sliced` and `LLoyd` are performed over the `Random Init.`

B.2. CoNMT

B.2.1. Experimental Setup

Results are reported on WMT[4]: 2016 Romanian→English (`ro-en`) translation task with 612K training samples. For subword tokenization we used the same SentencePiece (Kudo & Richardson, 2018) model for all language pairs, specifically the one used in the MBart multilingual model (Liu et al., 2020).

We used `fairseq` (Ott et al., 2019) framework for training our models. Baseline discrete models are trained with cross-entropy loss, label smoothing equal to 0.1 and effective batch size 65.5K tokens. Both discrete and continuous models are trained with learning rate $5 \cdot 10^{-4}$, 10k warm-up steps.

We measure translation accuracy using SacreBLEU[5] (Papineni et al., 2002; Post, 2018). All models are trained for 50k steps, and we report results on the best checkpoint according to validation BLEU score.

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

[5] nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.3.1