Reliable Measures of Spread in High Dimensional Latent Spaces

Anonymous Authors[1]

Abstract

Understanding geometric properties of the latent spaces of natural language processing models allows the manipulation of these properties for improved performance on downstream tasks. One such property is the amount of data spread in a model's latent space, or how fully the available latent space is being used. We demonstrate that the commonly used measures of data spread, average cosine similarity and a partition function min/max ratio I(V), do not provide reliable metrics to compare the use of latent space across data distributions. We propose and examine six alternative measures of data spread, all of which improve over these current metrics when applied to seven synthetic data distributions. Of our proposed measures, we recommend one principal component-based measure and one entropy-based measure that provide reliable, relative measures of spread and can be used to compare models of different sizes and dimensionalities.

1. Introduction

The product of many neural network models is a representation of the data input in a high dimensional latent space. The distribution of data in this latent space is often used in the application of the learned model through data clustering for classification, measuring distance between data points to quantify similarity, sampling to generate synthetic data elements, or any number of other downstream tasks. For this reason, understanding and manipulating the geometric properties of models' latent spaces is an area of active research.

One such geometric property is a quantification of how evenly the data is distributed in its latent representation. It has been shown that many common neural network architectures produce highly anisotropic latent spaces (Ethayarajh, 2019; Liu et al., 2018; Mimno & Thompson, 2017), and recent research has demonstrated improved performance on benchmarking tasks using various methods for enforcing more complete use of a model's latent space (Bihani & Rayz, 2021; Kaneko & Bollegala, 2020; Liang et al., 2021; Mu et al., 2017).

The majority of existing work quantifies the spread of data in a latent space with two measures of isotropy, average cosine similarity and a measure introduced by Mu et al. (2017) based on a ratio of principal component loadings. We present seven synthetic data distributions and show that these two measures do not behave as would be expected of a reliable measure of relative data spread.

We examine six alternative ways to quantify data spread and compare the performance of these measures on the same example distributions. We consider two principal component measures, two ratios of differential entropy approximations, and two measures of relative entropy (Kullback-Leibler Divergence). We show that all proposed measures behave more intuitively on our evaluation distributions than the two commonly used measures. In particular, we find that our proposed Eigenvalue Early Enrichment score and Vasicek Ratio MSE score most closely mirror our expectations across these example distributions and various numbers of dimensions. Finally, we investigate the behavior of the best-performing of the alternative data spread metrics on real (not simulated) latent spaces produced by a pre-trained Word2Vec model.

2. Related Work

In a natural language processing (NLP) context, the expressiveness of a model can be directly tied to the dimensionality of its trained word embeddings, with model expressiveness increasing with the number of available dimensions up to the point of severe overfitting (Yin & Shen, 2018). However, this increased expressiveness as a function of latent space dimensionality depends on the model effectively using the space across all of these dimensions.

This assumption of an evenly used latent space is not guaranteed, as addressed by a growing body of work that assesses and manipulates the geometry of latent spaces in NLP models. It has been demonstrated that static word

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

embedding models do not use their latent spaces evenly, with Mimno & Thompson (2017) finding a large positive inner product between individual word vectors and a global mean in Word2Vec, and Mu et al. (2017) finding non-zero global mean vectors and non-uniform distribution of variance across dimensions for several common models. Considering contextual NLP models, Ethayarajh (2019) find a non-zero average cosine similarity between word vectors throughout all layers of ELMo, BERT, and GPT-2.

Several explanations for this uneven use of the latent space explore the relationship between word representations and word frequency. Gao et al. (2019) introduce the concept of "representation degeneration" in which common words are handled differently than rare words during training, and word frequency has also been shown to be directly correlated with word vector magnitude (Kobayashi et al., 2020) and distance traveled during training (Gong et al., 2018). Similarly, Mu et al. (2017) find that the top two PCA components of static language model latent spaces are largely dedicated to expressing word frequency information.

Attempts to "correct" this incomplete use of the latent space have resulted in improvement on common NLP benchmarking tasks. Post-training adjustments include subtracting the global mean vector and removing highly explanatory principal components (Mu et al., 2017; Liang et al., 2021; Rajaee & Pilehvar, 2021), using an autoencoder framework toward a similar goal (Kaneko & Bollegala, 2020), and learning a transformation into a more uniformly filled non-Euclidean space (Frenzel et al., 2019). Adjustments made during training include minimizing an additional loss function (Liu et al., 2018; Gao et al., 2019; Wang et al., 2019) and using an isotropic Gaussian prior in a VAE (Zhang et al., 2022) or during batch normalization (Zhou et al., 2021).

Beyond improving benchmark performance, Liao et al. (2020) remove top principal components during an iterative quantization process and Sablayrolles et al. (2018) use a nearest-neighbor entropy approximation (Kozachenko-Leonenko entropy) to enforce uniform spread before quantizing data. Both of these methods find less data loss after compression than quantization methods that do not focus on using the latent space more completely.

3. Data Spread in High Dimensions

A theory of maximizing the expressiveness of a model by maximally using the latent space of that model requires that we first define what it means to "maximally use the latent space". While this task may seem intuitive, translating these intuitions into a well-defined, ideal latent space distribution, particularly in high dimensions, presents several complications. Here we explore existing concepts and distributions that can be used to help define this ideal space.

3.1. Data Spread vs. Isotropy

One concept that comes up in much of the literature in Section 2 is that of *isotropy*. The definition of isotropy varies between scientific and mathematical fields, but broadly it is defined as "[i]dentical in all directions; invariant with respect to direction" (Houghton Mifflin Harcourt Publishing Company). In the context of data representations, this means that the distribution of data representations would be the same in any direction from the origin. Rudelson (1999) finds that a probability distribution is in isotropic position if its covariance matrix is the identity, and Zhou et al. (2021) define a latent space as isotropic if all dimensions have the same variance and are uncorrelated.

Working from these definitions, a measure of isotropy can tell us whether our data are distributed similarly in all available directions, but it will say nothing about what the distribution is in any given direction. For example, data distributed on the shell of a hypersphere are equally isotropic to data distributed in a multivariate normal distribution. Intuitively we'd expect those two data distributions to perform differently on a measure of how fully the available space is being used.

Here we have introduced the term *isotropic* as meaning that a distribution appears to be the same in all directions. In this manuscript, we will use the term *spread* to expand on this notion; a distribution shows more complete *spread* if it evenly spreads points in all directions (isotropy) as well as along each axis/direction.

3.2. Reference Distributions

By definition, the uniform distribution will maximally fill any shape in any number of dimensions, making it an obvious choice for our ideally-filled latent space. However, while data in a uniform distribution will maximally fill a hypersphere of any radius, the probability abruptly drops to zero outside of that radius. This may not make sense for data representations in a continuous latent space.

Alternatively, we might consider a multivariate normal distribution ideal for maximally filling a latent space. Like a uniformly filled hypersphere, data in a multivariate normal distribution will be isotropic, but with probability instead approaching zero gradually as distance from the origin increases. Although using a multivariate normal distribution will result in having a higher density of points around the origin than we would find in referencing a uniform distribution, this gradual decrease in probability (and point density) generally seems a realistic and desirable trait for data representations in a latent space.

4. Models for Comparison

While we could have used the actual output of popular NLP models to compare different ways of measuring data spread in high dimensional space, we don't necessarily have strong intuition about how fully the data representations in these models fill the latent space or how they should compare to each other on a sufficient measure of data spread. Therefore we developed a collection of seven structured distributions, each with $d = \{2, 10, 50, 100\}$ dimensions and $250d$ data points, and used these as intuitive benchmarks on our proposed measures of spread.

For spherical distributions, we created a *Shell*, a *Nested Shell*, a uniformly filled *Sphere*, and a *Cone*. For cluster-based distributions, we created clusters that are symmetric across the origin and identical in size (*Symmetric Clusters*), asymmetric clusters of identical size (*Shifted Clusters*), and symmetric clusters of unequal size (*Uneven Clusters*). Details of the characteristics of these distributions can be found in Appendix A.

Figure 1 shows all seven distributions in 2-dimensional space, and provides a visual basis for our intuitions about how the distributions compare in terms of how fully they use the space. Appendix B includes further discussion of how these distributions compare in high dimensions, leading to the following intuitions about how they should perform on reliable measures of data spread:

1. The *Shell*, *Nested Shell*, and *Sphere* distributions should all have similar scores that indicate well-spread distributions, particularly in high dimensions.
2. The *Cone* distribution should have a slightly lower score that still indicates a fairly well-spread distribution, particularly in high dimensions.
3. All three cluster models should have similar scores that indicate poorly spread distributions, regardless of the dimensionality of the distribution.

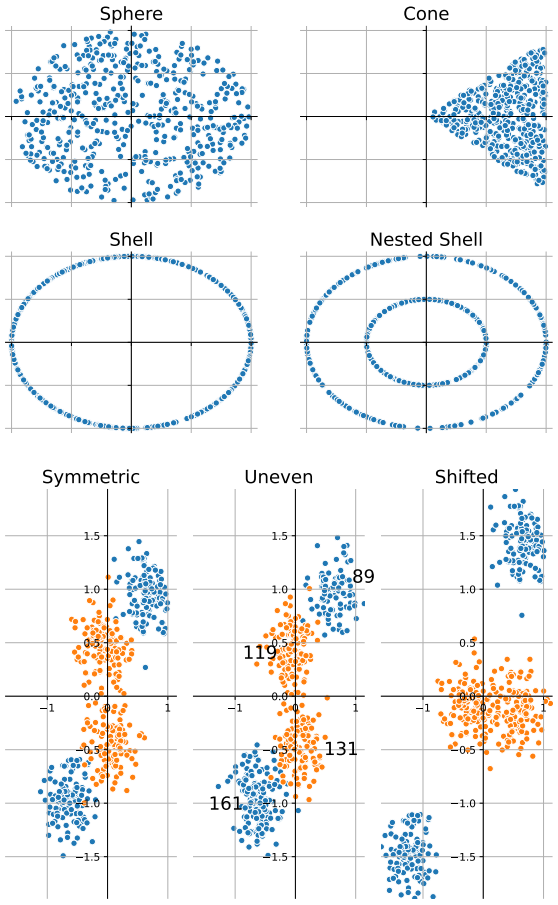

Figure 1. Example distributions visualized in 2 dimensions.

5. Common Measures of Isotropy

Research into the effect that data spread has on NLP model performance generally relies on one or two measures of spread: average cosine similarity and I(V). We compute these values across our seven example distributions and demonstrate that both of these common measures fall short as relative measures of spread.

5.1. Average Cosine Similarity (ACS)

The first and simpler of these two common measures is the average pairwise cosine similarity between word representations in an NLP model (Bihani & Rayz, 2021; Ethayarajh, 2019; Ferner & Wegenkittl, 2021; Gao et al., 2019; Liang et al., 2021). In an isotropic latent space, the expected pairwise cosine similarity between data points is zero.

5.2. I(V)

The second commonly used measure of spread, I(V) (sometimes called γ), is a min/max ratio of a principal component-based partition function. I(V) was first introduced by Mu et al. (2017) and has been used broadly in research involving latent space isotropy (Kaneko & Bollegala, 2020; Liao et al., 2020; Rajaee & Pilehvar, 2021; Wang et al., 2019; Zhang et al., 2022).

Mu et al. (2017) build on the partition function explored by Arora et al. (2015) (Equation 1), which was shown to be constant in an isotropic space over all partitions, c, where v_w is the vector representation of word w. They used the full-rank set of principal components as their partitions, C, and proposed Equation 2 as a measure of isotropy. I(V) ranges from zero to one, and holds the value one in a completely isotropic space.

$$Z_c = \sum_w \exp(c^T v_w) \tag{1}$$

$$I(V) = \frac{\min_{c \in C} Z_c}{\max_{c \in C} Z_c} \tag{2}$$

Table 1. Example distribution results on *Average Cosine Similarity (ACS)* and *I(V)* in 2 and 100 dimensions. Full results shown in Appendix D

Example Distribution	2D		100D	
	ACS	I(V)	ACS	I(V)
Shell	0.0027	0.9737	0.0007	0.9988
Nested Shell	0.0027	0.9801	0.0007	0.9982
Sphere	0.0036	0.9613	0.0007	0.9988
Cone	0.8176	0.9469	0.5119	0.9944
Symm. Clust.	0.0020	0.7888	0.0007	0.9822
Shifted Clust.	0.0109	0.9295	0.0012	0.8326
Uneven Clust.	0.0249	0.7797	0.0043	0.9819
Normal	0.0022	0.9969	0.0008	0.9988

5.3. Measure Weaknesses

I(V) and ACS both have characteristic results in an isotropic space ($ACS = 0$ and $I(V) = 1$), but this does not necessarily mean that they are good measures of *relative* spread. In particular, we find the following weaknesses after computing their values across our example distributions (Table 1):

- ACS is **insensitive to uneven data distributions that are symmetric across the origin**. As an example, all cluster distributions produce an ACS value similar to that of the *Sphere* distribution.
- I(V) is **insensitive to the difference in spread between the *Cone* and the three other spherical distributions**, particularly in high dimensions.
- I(V) shows **inconsistent results on the cluster models** across different dimension counts [1]:
- I(V) **heavily down-weights negative projections** due to the exponentiation in the partition function Z_c (Equation 1). In an ideally filled space, all positive and negative directions must be equally filled.
- I(V) is **sensitive to the sign of principal components**, which is arbitrary (Jolliffe & Cadima, 2016). Figure 2 shows two alternative (negated) projections of randomly sampled data: when the principal components for a single distribution are negated (causing the projected data to be reflected across the origin), the resulting I(V) scores can differ greatly.

6. Alternative Measures of Spread

To better address the issues described in Section 5.3, we explored a variety of new options for measuring spread in high dimensional space. Here we present two measures based on principal components/eigenvalues and four entropy-based measures.

[1]The I(V) measure is also sensitive to the random seed used to produce these distributions. The same cluster distributions with different seeds used for sampling the cluster centers and points will have I(V) scores that differ greatly.

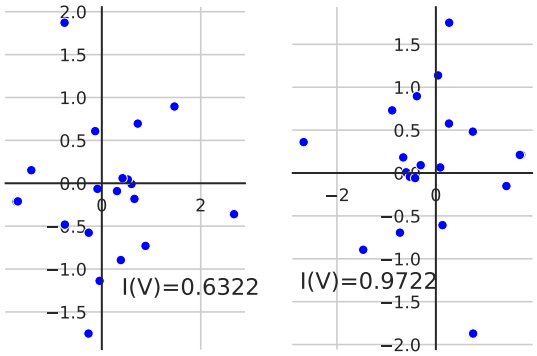

Figure 2. 2D principal component projections of identical sample data with component signs switched and I(V) calculated for each set of principal components.

In addition to the six measures proposed below, we explored a measure that considered the Euclidean distances between a distribution's first and kth nearest neighbors, and a Gaussian approximation of KL-Divergence. These measures are excluded from the main body as they did not provide new or interesting results when applied to our example distributions, but their definitions, results, and discussion can be found in Appendix C.

6.1. Principal Component Measures

Principal components (eigenvectors) describe all of the orthogonal directions for a given dataset, and the associated eigenvalues describe how much of the dataset's variance can be explained along each individual direction. These eigenvalues can be used as a measure of how evenly a latent space's variance is spread along each axis. In an ideal, well-spread space, each principal component will explain an equal amount of the dataset's variance, resulting in all eigenvalues being roughly equal.

Eigenvalue Ratio (ER) We first propose to compute the ratio of the largest and smallest eigenvalues as in Equation 3, where C is the set of all d principal components in a d-dimensional space, and λ_c is the eigenvalue associated with component c. In an ideally filled latent space, this ratio is equal to one; if one or more dimensions is poorly used, the value will approach zero. Note that this measure provides no way to differentiate between the case of one poorly utilized dimension from the case in which many (or even most) dimensions are largely unused.

$$ER = \frac{\min_{c \in C} \lambda_c}{\max_{c \in C} \lambda_c} \tag{3}$$

Eigenvalue Early Enrichment (EEE) A related approach is to instead consider the cumulative sum of the eigenvalues (which are sorted largest to smallest by convention). Figure 3 provides a visual example of the approach.

In Equation 4, we calculate the area between the cumulative

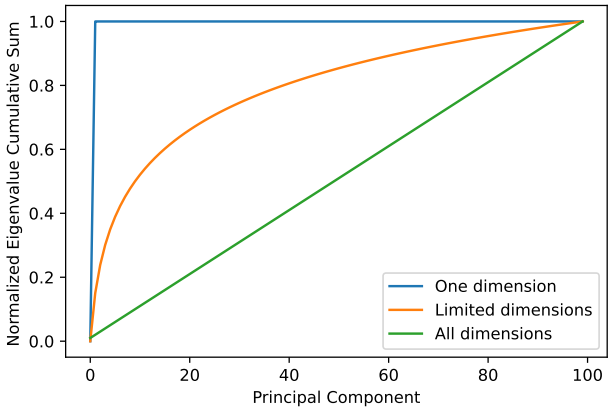

Figure 3. Cumulative Sum of Eigenvalues for latent spaces in which variance is explained on one dimension, unevenly across many dimensions, and evenly across all dimensions.

sum curve, X_{EEE}, and the ideal linear sum, Y_{ref}, as a proportion of the total space available above that linear sum line (with $v = \sum_C \lambda_c$ as the total variance in the d-dimensional distribution). Well-spread distributions show an EEE value close to zero, while poorly spread distributions approach a value of one. Unlike the previous approach, EEE is able to differentiate distributions that utilize different fractions of the available latent space.

$$EEE = \frac{AUC(X_{EEE} - Y_{ref})}{\frac{1}{2}dv} \tag{4}$$

6.2. Entropy Ratio Measures

In information theory, the concept of *entropy* quantifies the amount of uncertainty involved in predicting an outcome, which can be mapped to the concept of *spread*: with poorly spread data, certainty is high and entropy is low; with well-spread data, uncertainty is high, as is entropy. Shannon entropy (Shannon, 1951) is defined for continuous data as in Equation 5 and has been proposed, in the field of astronomy, as a measure of the isotropy of the universe (Pandey, 2016).

$$H(p) = \int p(x) \log p(x) dx \tag{5}$$

For a fixed variance, this parametric definition is maximized by the normal distribution (Arizono & Ohta, 1989; Beirlant et al., 1997), which matches our intuition (from Section 3.2) that a multivariate normal is a good candidate for a distribution that ideally fills the available space. Thus, the two empirical Shannon entropy approximations discussed below will be compared to a multivariate normal with equal variance to create relative measures of spread.

Vasicek Ratio Mean Squared Error (VRM) Our first entropy-based measure of data spread builds on the Vasicek entropy approximation (Vasicek, 1976), which is limited to univariate data and rests on the notion that ordered points will be evenly spaced in a highly entropic space. The approximation is presented in Equation 6, and considers pairs of points that are separated by a fixed interval, m, where m is a positive integer smaller than $n/2$, n is the total number of data points, $x_j = x_1$ if $j < 1$, and $x_j = x_n$ if $j > n$.

$$H_{vas} = \frac{1}{n} \sum_{i=1}^{n} \log \frac{n}{2m} (x_{i+m} - x_{i-m}) \tag{6}$$

To create a relative measure of data spread, we consider a ratio of the empirical value for a given distribution to the theoretical value for our reference normal distribution, $\ln(\sqrt{2\pi e \sigma^2})$(Arizono & Ohta, 1989). This ratio will equal one for normally distributed data points and will approach zero for poorly spread data [2]. We compute the mean squared error (MSE) from the target ratio of one to create a multi-dimensional measure (Equation 7).

$$VRM = \frac{1}{d} \sum_{i=1}^{d} (1 - \frac{H_{vas}}{\ln(\sqrt{2\pi e \sigma^2})})^2 \tag{7}$$

Nearest Neighbor Entropy Ratio (NNR) A multivariate continuous entropy approximation rests on the notion that highly entropic spaces will maximize the minimum distance between points and their nearest neighbors (Beirlant et al., 1997; Leonenko et al., 2008; Sablayrolles et al., 2018). Equation 8 presents this nearest neighbor approximation as defined by Beirlant et al. (1997), in which ρ_i is the distance of element i to its nearest neighbor, and e is the Euler constant.

$$H_{NN} = \sum_{i=1}^{n} \ln(\rho_i) + \ln(2n) + e \tag{8}$$

To create a relative measure, we simulate a d-dimensional multivariate normal distribution and empirically calculate H_{NN} for this distribution as our theoretical maximum. We use this in a ratio as in the previous section to create our proposed measure of spread[3]. A fully used space will produce a ratio close to one and a poorly used space will produce a value approaching zero.

$$NNR = \frac{H_{NN}(X)}{H_{NN}(Y)}, \quad Y \sim N(0, \sigma^2) \tag{9}$$

6.3. KL-Divergence Measures

KL-Divergence (Equation 10) provides an established framework for comparing the entropy of two distributions.

[2] This ratio is similar to Pielou's evenness index from the biodiversity literature (Pielou, 1966).

[3] This measure is similar to the Clark-Evans measure of dispersion from spatial statistics (Clark & Evans, 1979).

We compare our example distributions directly to a multivariate normal distribution to create a relative measure of spread. Below, we describe two measures of spread based on empirical KL-Divergence approximations.

$$KL = \int_x p(x) \log \frac{p(x)}{q(x)} \tag{10}$$

Discrete KL-Divergence Mean Squared Error (DKLM) KL-Divergence can be computed for finite data sets by discretizing the data as in Equation 11. We estimate univariate $p(X')$ and $q(Y')$ by binning observed and reference data (X, Y) into k bins along each dimension (with $k = 30$) and compute the MSE from a reference of $KL_{disc} = 0$ over all d dimensions, as in Equation 12.

$$KL_{disc} = \sum_{i=1}^{k} p(x_i') \log \frac{p(x_i')}{q(x_i')} \tag{11}$$

$$DKLM = \frac{1}{d} \sum_{i=1}^{d} (KL_{disc_i})^2 \tag{12}$$

Nearest Neighbor KL-Divergence (NNKL) Finally, we propose a KL-Divergence approximation based on comparing the distributions of nearest neighbor distances in our observed and reference distributions. We present Equation 13, a modified version of the approximation derived in Pérez-Cruz (2008), which relies only on nearest neighbor distances to point x_i in our observed and reference distributions, $s(x_i)$ and $r(x_i)$ respectively, the number of dimensions, d, and the number of data points, n (observed) and m (reference).

$$NNKL = \frac{d}{n} \sum_{i=1}^{n} \max(0, \log \frac{r_k(x_i)}{s_k(x_i)}) \tag{13}$$

7. Results

Here, we evaluate each of the alternative measures proposed in Section 6 as applied to the seven example distributions described in Section 4 and demonstrate that all measures better reflect our expectations as relative measures of spread. We review the results for each of our proposed measures below[4], and finally apply our two strongest measures to a pre-trained Word2Vec model.

7.1. Principal Component Measures

Our proposed ER measure considers the ratio between the largest and smallest eigenvalues of a data distribution (i.e. the explained variance for the first and last principal component), where a ratio of one indicates a well-spread distribution (Equation 3). The EEE score instead considers the area under the curve of the cumulative sum of eigenvalues as a proportion of the total area available, where a score of zero indicates a well-spread distribution (Equation 4).

[4]Patterns described in the main body hold for 10 and 50 dimensions as well, and complete results can be found in Appendix D.

Table 2. Example distribution results on *Eigenvalue Ratio (ER)* and *Eigenvalue Early Enrichment (EEE)* in 2 and 100 dimensions.

Example Distribution	2D ER	2D EEE	100D ER	100D EEE
Shell	0.9021	0.0129	0.7845	0.0361
Nested Shell	0.8552	0.0195	0.7542	0.0422
Sphere	0.9118	0.0115	0.7889	0.0353
Cone	0.4173	0.1028	0.0090	0.0458
Symm. Clust.	0.1124	0.1995	0.0001	0.5394
Shifted Clust.	0.0915	0.2081	0.0007	0.6746
Uneven Clust.	0.1102	0.2004	0.0001	0.5397
Normal	0.7910	0.0292	0.7804	0.0362

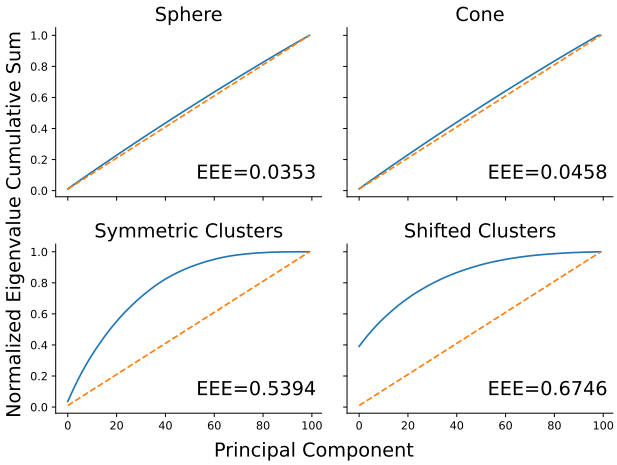

Figure 4. Eigenvalue Early Enrichment (EEE) scores for *Sphere*, *Cone*, *Symmetric Clusters*, and *Shifted Clusters* distributions

As seen in Table 2, both measures show a gradient of spread across our example distributions in 2 and 100 dimensions, with the *Sphere* and *Shell* distributions most evenly spread and the three cluster distributions least well-spread.

Although both of these measures seem to perform well on our example distributions, the EEE score better agrees with our expectations for a relative measure of spread, since it considers more than just the first and last eigenvalue. Indeed, Table 2 shows a disagreement between the ER score and the EEE score on the 100-dimensional cluster distributions which can be explained by examining the eigenvalue cumulative sum curves in Figure 4. The ER score penalizes the *Symmetric Clusters* distribution for having a very small last eigenvalue, while the EEE score more accurately reflects the more gradual decrease over all eigenvalues when compared to the *Shifted Clusters* distribution.

Table 3. Example distribution results for *Vasicek Ratio MSE (VRM)* and *Discrete KL-Divergence MSE (DKLM)* in 2 and 100 dimensions.

Example Distribution	2D VRM	2D DKLM	100D VRM	100D DKLM
Shell	0.1891	0.1349	0.0010	0.0000
Nested Shell	0.0465	0.0774	0.0014	0.0019
Sphere	0.0106	0.0169	0.0009	0.0000
Cone	0.0686	0.0266	0.0097	0.0295
Symm. Clust.	0.1545	0.1434	0.2624	0.6200
Shifted Clust.	0.2143	0.5392	0.2386	20.5810
Uneven Clust.	0.1637	0.1470	0.2574	0.6333
Normal	0.0047	0.0190	0.0010	0.0000

7.2. Univariate Entropy Measures

VRM considers a ratio of the Vasicek entropy approximation between an observed distribution and a normal distribution (Equation 6) and DKLM considers the KL-divergence between a discretized observed distribution and a normal distribution (Equation 11). We calculated the MSE over all dimensions to translate these into multivariate measures, and thus zero indicates a well-spread distribution for both measures.

Both VRM and DKLM behave as we would expect for quality relative measures of spread, particularly in high dimensions (see Table 3 and Figures 5 and 6 [5]). Both measures produce small MSE for the *Sphere*, *Shell*, and *Nested Shell* distributions, a larger MSE for the *Cone* distribution, and an even larger MSE for the cluster distributions.

The *Shifted Clusters* chart in Figure 6 shows how just a few components dominate the DKLM score for this distribution, which is possible because KL-Divergence does not have an upper limit. Indeed, our KL-divergence measures (DKLM and NNKL) are functions of the number of data points and the number of dimensions, such that the range of scores for distributions in high dimensions is much larger than in lower dimensional distributions[6]. An ideal measure of spread should allow meaningful comparison between distributions with different dimensionality.

7.3. Nearest Neighbor Entropy Measures

NNR considers a ratio between the nearest-neighbor entropy approximation of the observed data and that of a multivariate normal distribution (Equation 8), and NNKL considers a nearest-neighbor approximation of the KL-divergence between the observed data and a multivariate normal distribution. In a well-spread distribution, we would expect an NNR of one, and an NNKL of zero.

Table 4 shows that, although these measures are once again effective at identifying the less-complete spread of the cluster distributions, the *Cone* distribution receives similar scores to the *Sphere* and *Shell* distributions, while the scores for the *Nested Shell* distribution indicate a less complete spread, especially in high dimensions.

Figure 7 shows histograms of the nearest neighbor distances in our example distributions, and demonstrates that these distances are very similar for the *Sphere* and *Cone* distributions as the number of dimensions increases, while the nearest neighbor distances for the *Nested Shell* distribution

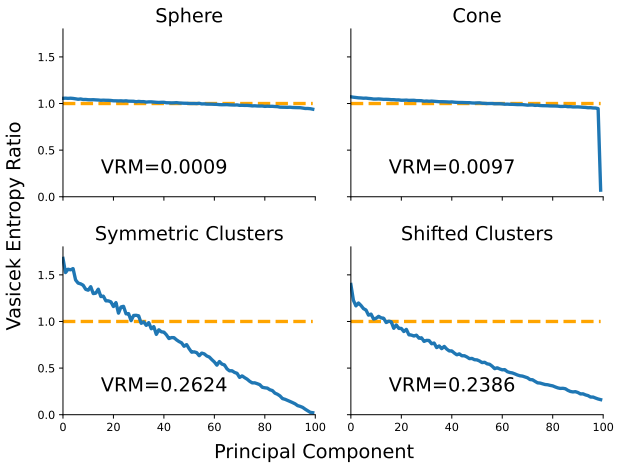

Figure 5. Vasicek Entropy Ratio for 100-dimensional distributions in blue, with the reference line in orange. In a well-spread distribution, the ratio would always be one, and the VRM score would be zero.

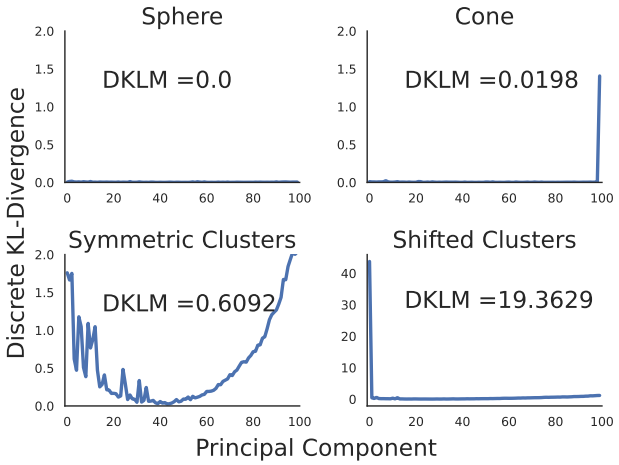

Figure 6. Discrete KL-Divergence for 100-dimensional models. In a well-spread distribution the KL-Divergence and DKLM scores would both be zero.

[5] To ensure that the measures are not influenced by covariance between dimensions, we projected our distributions onto their principal components before computing our univariate entropy measures. This is reflected in the negative slopes and u-shaped distributions in Figures 5 and 6, respectively.

[6] Table 4 provides the strongest example of this issue, with the NNKL scores for the 100-dimensional cluster distributions being almost 300x the scores of their 2-dimensional counterparts.

Table 4. Example distribution results on *Nearest Neighbor Entropy Ratio (NNR)* and *Nearest Neighbor KL-Divergence (NNKL)* in 2 and 100 dimensions.

Example Distribution	2D NNR	2D NNKL	100D NNR	100D NNKL
Shell	0.1323	7.4573	1.0024	0.2532
Nested Shell	0.3996	5.2696	0.9887	13.2125
Sphere	0.9749	0.9267	1.0023	0.2558
Cone	0.9350	1.2659	1.0022	0.2853
Symm. Clust.	0.8971	1.4874	0.7135	394.5467
Shifted Clust.	0.8312	1.9226	0.6882	429.2692
Uneven Clust.	0.9015	1.4215	0.7134	394.7038
Normal	0.9973	0.7260	1.0001	1.1751

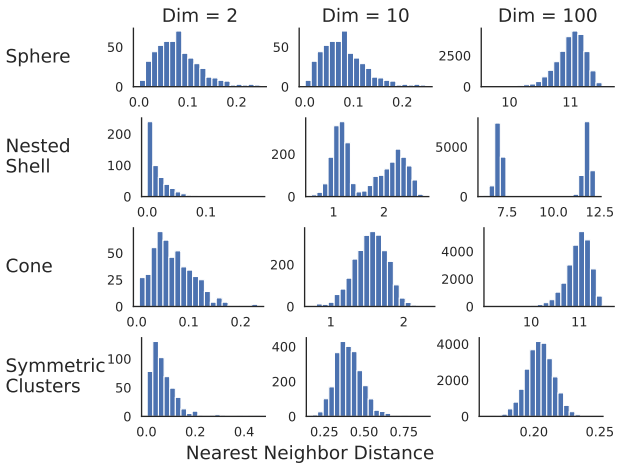

Figure 7. Histograms of nearest neighbor distances for our *Sphere*, *Nested Shell*, *Cone*, and *Symmetric Clusters* example distributions in 2, 10, and 100 dimensions.

become increasingly separated for the two "rings" in the distribution. This suggests that measures that rely wholly on Euclidean distances may not be as robust as some of the other measures we have examined.

7.4. Word2Vec Embedding Performance

The work in this paper is largely motivated by creating better metrics for comparing the spread in natural language processing models' latent spaces. Thus, as a final exploration, we compare the performance of a pre-trained Word2Vec model (300-dimensions trained on the Google News corpus, Mikolov, 2013; Mikolov et al., 2013) on our two strongest candidate measures, EEE and VRM, along with the current common measures, ACS and I(V). We follow the methods used in generating our example distributions and randomly sample 75000 ($250d$) word embeddings from this pre-trained model to use in calculating our measures of data spread. To examine whether these measures can capture a gradual increase in data spread when applied to a real (not simulated) latent space, we add random uniform noise to the pre-trained embeddings and calculate the EEE, VRM, ACS, and I(V) scores at each addition.

Table 5. Pre-trained Word2Vec Results on Early Eigenvalue Enrichment (EEE), Vasicek Ratio MSE (VRM), Average Cosine Similarity (ACS), and I(V) measures with gradual addition of noise sampled from uniform distributions with the listed range.

Noise	EEE	VRM	ACS	I(V)
None	0.4058	0.1525	0.1317	0.9251
± 0.001	0.4058	0.1525	0.1319	0.9251
± 0.01	0.4049	0.1514	0.1280	0.9257
± 0.05	0.3861	0.1320	0.1147	0.9359
± 0.1	0.3375	0.0961	0.0933	0.9534
± 0.3	0.1492	0.0216	0.0346	0.9883
± 0.5	0.0800	0.0066	0.0192	0.9952
± 1	0.0441	0.0016	0.0062	0.9991
± 3	0.0367	0.0010	0.0022	0.9996

Table 5 shows that the original pre-trained Word2Vec model falls somewhere between our example *Cone* distribution and our *Symmetric Clusters* distribution according to all four of these measures of spread. Additionally, we see that all four measures successfully reflect the gradual increase in data spread as random noise is added to the pre-trained word embeddings. This does not discount the issues with ACS and I(V) that we raised in Section 5, but further supports the use of EEE and VRM as relative measures of data spread that do not suffer from the same weaknesses.

8. Conclusion

In this work, we have examined methods for quantifying how completely data fills a latent space. We demonstrated that the metrics commonly being used to quantify this usage are insufficient as relative measures of data spread, and we proposed six alternative measures of data spread. Of our proposed measures, all improved upon the commonly used measures when applied to seven synthetic data distributions, and we present one principal component measure and one entropy-based measure, EEE (Early Eigenvalue Enrichment) and VRM (Vasicek Ratio MSE) respectively, as our strongest proposed measures.

Future work that builds on our findings includes the reassessment of previous methods and the development of new methods for increasing data spread in NLP models using these two proposed measures. We expect that the application of reliable measures of data spread in this manner will contribute to the general understanding of NLP and other neural network models, by further defining the geometric properties associated with improved benchmarking performance.

References

Arizono, I. and Ohta, H. A test for normality based on kullback—leibler information. *The American Statistician*, 43(1):20–22, 1989.

Arora, S., Li, Y., Liang, Y., Ma, T., and Risteski, A. Random walks on context spaces: Towards an explanation of the mysteries of semantic word embeddings. *arXiv preprint arXiv:1502. 03520*, pp. 385–399, 2015.

Beirlant, J., Dudewicz, E. J., Györfi, L., Van der Meulen, E. C., and Others. Nonparametric entropy estimation: An overview. *International Journal of Mathematical and Statistical Sciences*, 6(1):17–39, 1997.

Bihani, G. and Rayz, J. T. Low anisotropy sense retrofitting (laser): Towards isotropic and sense enriched representations. *arXiv preprint arXiv:2104.10833*, 2021.

Clark, P. J. and Evans, F. C. Generalization of a nearest neighbor measure of dispersion for use in k dimensions. *Ecology*, 60(2):316–317, 1979.

Doersch, C. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

Duchi, J. Derivations for linear algebra and optimization. *Berkeley, California*, 3(1):2325–5870, 2007.

Ethayarajh, K. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. *arXiv preprint arXiv:1909.00512*, 2019.

Ferner, C. and Wegenkittl, S. Isotropic contextual representations through variational regularization. September 2021.

Filippi, S., Cappé, O., and Garivier, A. Optimism in reinforcement learning and kullback-leibler divergence. In *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 115–122. IEEE, 2010.

Frenzel, M. F., Teleaga, B., and Ushio, A. Latent space cartography: Generalised metric-inspired measures and measure-based transformations for generative models. *arXiv preprint arXiv:1902.02113*, 2019.

Gao, J., He, D., Tan, X., Qin, T., Wang, L., and Liu, T.-Y. Representation degeneration problem in training natural language generation models. *arXiv preprint arXiv:1907.12009*, 2019.

Gong, C., He, D., Tan, X., Qin, T., Wang, L., and Liu, T.-Y. Frage: Frequency-agnostic word representation. *Advances in neural information processing systems*, 31, 2018.

Houghton Mifflin Harcourt Publishing Company. *isotropic, adj.* The American Heritage Science Dictionary. URL https://www.dictionary.com/browse/isotropy.

Jolliffe, I. T. and Cadima, J. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.

Kaneko, M. and Bollegala, D. Autoencoding improves pre-trained word embeddings. *arXiv preprint arXiv:2010.13094*, 2020.

Kobayashi, G., Kuribayashi, T., Yokoi, S., and Inui, K. Attention is not only a weight: Analyzing transformers with vector norms. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Leonenko, N., Pronzato, L., and Savani, V. A class of rényi information estimators for multidimensional densities. *The Annals of Statistics*, 36(5):2153–2182, 2008.

Liang, Y., Cao, R., Zheng, J., Ren, J., and Gao, L. Learning to remove: Towards isotropic pre-trained bert embedding. *International Conference on Artificial Neural Networks*, pp. 448–459, 2021.

Liao, S., Chen, J., Wang, Y., Qiu, Q., and Yuan, B. Embedding compression with isotropic iterative quantization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8336–8343, 2020.

Liu, W., Lin, R., Liu, Z., Liu, L., Yu, Z., Dai, B., and Song, L. Learning towards minimum hyperspherical energy. *Advances in neural information processing systems*, 31, 2018.

Mikolov, T. Word2vec archive, Jul 2013. URL https://code.google.com/archive/p/word2vec/.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

Mimno, D. and Thompson, L. The strange geometry of skip-gram with negative sampling. *Empirical Methods in Natural Language Processing*, 2017.

Mu, J., Bhat, S., and Viswanath, P. All-but-the-top: Simple and effective postprocessing for word representations. *arXiv preprint arXiv:1702.01417*, 2017.

Muller, M. E. A note on a method for generating points uniformly on n-dimensional spheres. *Communications of the ACM*, 2(4):19–20, 1959.

Pandey, B. A new method for testing isotropy with shannon entropy. *Monthly Notices of the Royal Astronomical Society*, 462(2):1630–1641, 2016.

Pérez-Cruz, F. Kullback-leibler divergence estimation of continuous distributions. In *2008 IEEE international symposium on information theory*, pp. 1666–1670. IEEE, 2008.

Pielou, E. C. The measurement of diversity in different types of biological collections. *Journal of theoretical biology*, 13:131–144, 1966.

Rajaee, S. and Pilehvar, M. T. How does fine-tuning affect the geometry of embedding space: A case study on isotropy. *arXiv preprint arXiv:2109.04740*, 2021.

Rudelson, M. Random vectors in the isotropic position. *Journal of Functional Analysis*, 164(1):60–72, 1999.

Sablayrolles, A., Douze, M., Schmid, C., and Jégou, H. Spreading vectors for similarity search. *arXiv preprint arXiv:1806.03198*, 2018.

Shannon, C. E. Prediction and entropy of printed english. *Bell system technical journal*, 30(1):50–64, 1951.

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Vasicek, O. A test for normality based on sample entropy. *Journal of the Royal Statistical Society: Series B (Methodological)*, 38(1):54–59, 1976.

Wang, L., Huang, J., Huang, K., Hu, Z., Wang, G., and Gu, Q. Improving neural language generation with spectrum control. *International Conference on Learning Representations*, 2019.

Yin, Z. and Shen, Y. On the dimensionality of word embedding. *Advances in neural information processing systems*, 31, 2018.

Zhang, L., Buntine, W., and Shareghi, E. On the effect of isotropy on vae representations of text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 694–701, 2022.

Zhou, W., Lin, B. Y., and Ren, X. Isobn: Fine-tuning bert with isotropic batch normalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 14621–14629, 2021.

A. Distribution Definitions

Shell

A relatively simple way to sample points from the surface of a d-dimensional unit hypersphere is to sample from a multivariate standard normal and normalize the length of the associated vectors (Muller, 1959). Here, X_i^{shell} is the vector for one of the $250d$ points sampled from our d-dimensional shell.

$$X_i^{shell} = \frac{X_i}{\parallel X_i \parallel}, \quad X_i \sim N(0, I_d) \tag{14}$$

Nested Shell

To create a nested shell, that is, a shell within a shell, we adjust the radius of half of our shell points to be $\frac{1}{2}$ instead of 1.

$$r_i^{nest} = \begin{cases} 1 & i \leq 250d/2 \\ 1/2 & i > 250d/2 \end{cases} \tag{15}$$

$$X_i^{nest} = r_i^{nest} X_i^{shell} \tag{16}$$

Sphere

We can sample from a uniformly filled unit hypersphere by taking the points from our *Shell* distribution and randomizing their distance from the origin between 0 and the radius of the hypersphere, r. However, due to the exponential relationship between the radius and the volume of a d-dimensional hypersphere (as seen in Equations 25 and 26, $V_d = f(r^d)$)[7], we cannot sample the distance of each point directly from a $U(0, r)$ distribution without causing points to be more densely concentrated around the origin. Thus, we invert this exponent when sampling our distance from the origin, as seen in Equation 17.

$$l_i^{sphere} = l_i^{\frac{1}{d}}, \quad l_i \in L \sim U(0, r) \tag{17}$$

$$X_i^{sphere} = l_i^{sphere} X_i^{shell} \tag{18}$$

Cone

A 3-dimensional cone can be described as a continuous series of circles (2-dimensional spheres), stacked along a third dimension (the cone's height), where the radius of each circle is a function of the distance from the origin, l, and the angle/width of the cone, θ: as we move further from the origin (as l grows), the radius of each circle increases according to the width of the cone.

Analogously, a d-dimensional hypercone can be imagined as a series of $(d-1)$-dimensional hyperspheres that are continuously stacked along the dth dimension. Again, the

[7] This exponential relationship is directly related to the discussion in Section B.2, since points that are distributed uniformly within a high-dimensional hypersphere will end up largely in the neighborhood of the hypersphere's radius.

radius of the ith stacked sphere, r_i^{stack}, is a function of l_i^{cone} (the sphere's distance from the origin) and θ (the angle/width of the cone)[8]. As in the case of the *Sphere* distribution, we invert the exponential relationship between volume and distance from the origin to uniformly sample within the cone as in Equation 19.

$$l_i^{cone} = l_i^{\frac{1}{d}}, \quad l_i \in L \sim U(0, r) \tag{19}$$

$$r_i^{stack} = l_i^{cone} \tan(\theta) \tag{20}$$

We then sample from an $(d-1)$-dimensional sphere of radius r_i^{stack} as in Equations 17 and 18. The points sampled from these $(d-1)$-dimensional spheres (X_i^{stack}) are concatenated with the distance from the origin (l_i^{cone}) to produce d-dimensional vectors as in Equation 21.

$$X_i^{cone} = (l_i^{cone}, X_i^{stack}) \tag{21}$$

Clusters

Cluster distributions were created by randomly sampling d cluster centers, mirroring these centers over the origin to create a total of $2d$ cluster centers, and randomly sampling around each center μ_j to create clusters. In Equation 23 we use the floor function to ensure integer division so that the *Symmetric Clusters* distribution sampled 125 points around each cluster center.

$$\mu_j^{symm} = \begin{cases} \mu_j & j \leq d \\ -\mu_{j/2} & j > d \end{cases}, \quad \mu_j \sim U(-1, 1) \tag{22}$$

$$X_i^{symm} \sim N(\mu_{\lfloor i/125 \rfloor}^{symm}, \min(1/d, 0.2)) \tag{23}$$

To break symmetry, we amend Equation 23 in two ways. For the *Shifted Clusters* distribution, we randomly shifted each cluster center and again sampled 125 points around each cluster center.

$$\mu_i^{shift} = \mu_i^{symm} + S_i, \quad S_i \sim U(0, 1) \tag{24}$$

For the *Uneven Clusters* distribution, we randomized the number of points in each symmetric cluster, such that the clusters in each mirrored cluster pair, (μ_i^{symm}, μ_{i+d}^{symm}), have k and $(250-k)$ points, respectively, and k is sampled from a $U(0, 250)$ distribution.

B. Example Distributions in High Dimensions

B.1. Geometry

While not central to our work, there are several non-intuitive geometric characteristics that will come up when discussing data spread in high dimensional spaces. Here we provide a brief description of two particular characteristics that are often included in defining the *curse of dimensionality*.

First, the volume and surface area of a hypersphere (with a fixed radius) approach zero as the number of dimensions grows above ∼7. Second, the volume approaches zero more quickly than the surface area, which results in almost all of the points within a uniformly filled hypersphere being concentrated on a very thin shell of that hypersphere. The equations for surface area ($S_d(r)$) and volume ($V_d(r)$) in d dimensions are shown in Equations 25 and 26 respectively.

$$S_d(r) = \frac{2\pi^{d/2}}{\Gamma(d/2)} r^{d-1} \tag{25}$$

$$V_d(r) = S_d \frac{r}{d} \tag{26}$$

In Equation 25 the gamma function in the denominator will dominate as d grows, bringing the surface area to zero as the number of dimensions increases. The additional d factor in the denominator of the volume function causes $V_d(r)$ to approach zero even more quickly than $S_d(r)$ as d grows, with the result that data points are increasingly concentrated in the neighborhood of a hypersphere shell with a fixed radius as the number of dimensions grows [9]

B.2. Visualizing the Distributions

Figure 8 shows density plots of the raw data values along each dimension in the 10-dimensional distributions. Even in just ten dimensions, the distribution of data among the three spherical distributions (*Shell*, *Nested Shell*, *Sphere*) are quite similar, and even the *Cone* distribution is fairly similar with the exception of the dth dimension along which the cone points. As discussed in Section B.1, this is a consequence of the data points being more densely concentrated on the shell of a fixed-radius hypersphere as the number of dimensions grows.

The density plots for the cluster distributions help to demonstrate the effect of the transformations between the *Symmetric Clusters* distribution and the *Shifted* and *Uneven Clusters* distributions. They also make clear the stark difference between the data uniformity of the spherical distributions and the cluster based distributions.

Figure 9 shows histograms of vector norms for our *Sphere*, *Cone*, and *Symmetric Clusters* distributions in 2, 10, and 100 dimensions, with a reference normal distribution for comparison[10]. The first item of note is that, as the number of dimensions grows, the shape of the *Sphere* and *Cone*

[8] We chose to define $\theta = 1/\sqrt{d}$ as it produced distributions that more consistently demonstrated the strengths and weaknesses of common and proposed measures of spread across different numbers of dimensions.

[9] It should be noted that this effect is only observed when considering a bounded space, such as a hypersphere, as can be seen by comparing the histograms of the vector norms of a hypersphere and a normal distribution in Figure 9.

[10] The *Cone* distribution's vector norms are longer than the other

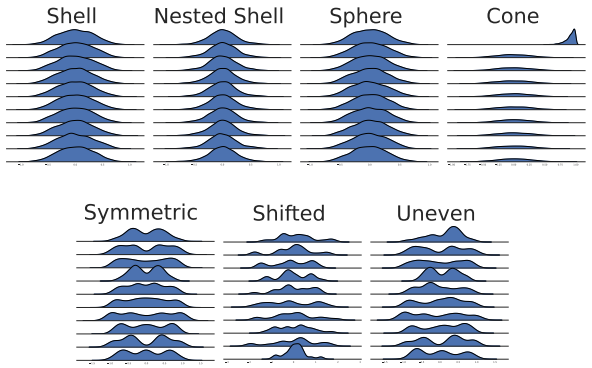

Figure 8. Density plots for 10-Dimensional example distributions show distribution of data values along each dimension; Top row shows the four spherical models; Bottom row shows the three cluster models.

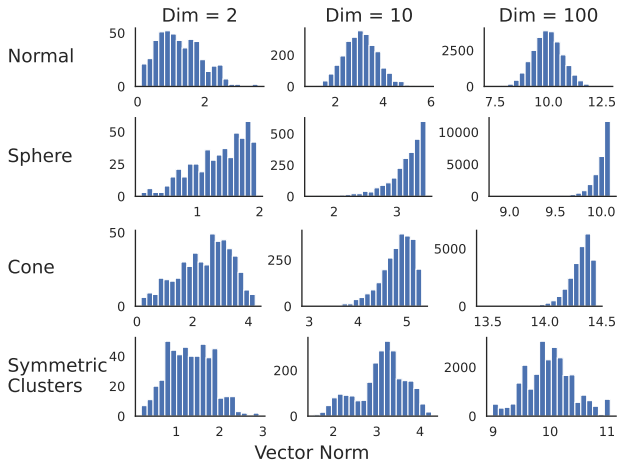

Figure 9. Histograms of vector norms for our reference *normal* distribution and our *Sphere*, *Cone*, and *Symmetric Clusters* example distributions in 2, 10, and 100 dimensions

histograms becomes increasingly similar to the expectation for the *Shell* histogram (in which all vectors have the same norm by definition). Again, this follows from the characteristic described in Section B.1, in that points within a hypersphere are forced onto a thin shell of that hypersphere in high dimensions.

Our cluster distributions are not affected by this characteristic of high-dimensional geometry, since each cluster is defined by a mean and variance, rather than a strict radius. This is apparent in the histograms for the *Symmetric Clusters* distribution, which is largely unchanged in shape as the dimensions grow.

distributions across all dimension counts. This is an artifact of holding the variance equal across all distributions, which we do to provide a more accurate comparison to the normal distribution as described in Section 6.2

C. Additional Proposed Measures of Spread

C.1. KNN Overlap

One (computationally expensive) way to simulate a fully used latent space is to maximize the smallest pairwise Euclidean distance between points in a distribution. Indeed, there are several works that build on this concept by creating loss functions designed to maximize the distance of each point to its nearest neighbors during training (Liu et al., 2018; Sablayrolles et al., 2018). We sought to develop a relative measure based on this concept.

In visualizing the effect of their nearest-neighbor-based loss function, Sablayrolles et al. (2018) include a chart showing histograms for the first and 100th nearest neighbor distances for a sample of their data. The motivation behind this approach is that, for an evenly filled space, the distribution of Euclidean distances of the first nearest neighbors for all points will not overlap with (will be smaller than) the distribution of the distances to the 100th nearest neighbors. Alternatively, in an unevenly used space, these distributions will overlap.

Although Sablayrolles et al. (2018) did not quantify this overlap, we developed a measure based on the proportion of sampled data points falling in the intersection of the distributions for the first and kth nearest neighbor distances, where $k = \min(5d, 100)$ (for d dimensions). This measure is visualized in Figure 10. Values range from zero (the expected value when for a well-spread distribution) to one (for a poorly spread distribution).

This was the only measure that we explored that didn't clearly improve on commonly used measures. As seen in Table 6, this measure is very small for almost all example

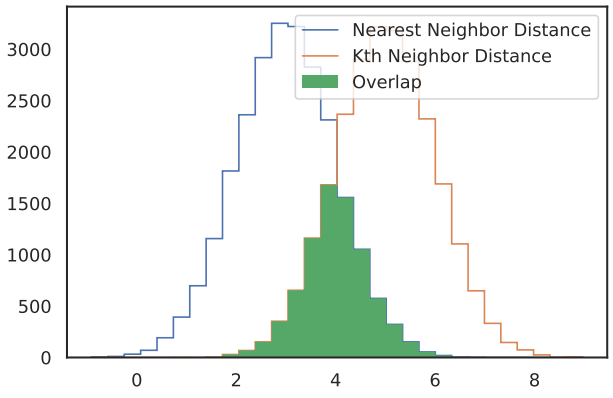

Figure 10. Based on findings from Sablayrolles et al. (2018), histograms of Euclidean distance to first and kth nearest neighbors should have less overlap in a well-spread distribution than in a poorly spread distribution.

Table 6. Example distribution results for KNN Overlap.

EXAMPLE DIST.	2D	10D	50D	100D
Shell	0.0120	0.0000	0.0000	0.0000
Nested Shell	0.0480	0.0924	0.0000	0.0000
Sphere	0.0240	0.0000	0.0002	0.0001
Cone	0.0100	0.0008	0.0004	0.0001
Symm. Clust.	0.1560	0.0508	0.0061	0.0078
Shifted Clust.	0.1700	0.0516	0.0078	0.0067
Uneven Clust.	0.1960	0.0896	0.8468	0.8491
Normal	0.1960	0.1580	0.1940	0.2260

distributions. And although the 2-dimensional distribution values reflect a reasonable relative measure of spread, these relative differences become smaller as the number of dimensions increases. We did try adjusting the value for k to account for this difference, but found that the scores for the cluster distributions were particularly sensitive to the choice of k.

Figure 11 shows that the *Uneven Clusters* distribution has very high overlap with $k = 100$. In the *Uneven Clusters* distribution, the randomization of cluster size will cause many of the clusters to have more than the standard 125 data points used in the *Symmetric Clusters* distribution. However, the randomly sized clusters still share the same tight distribution around a point, causing the 100th nearest neighbor to very frequently be quite close. Alternatively, increasing k causes the *KNN Overlap* score for the *Symmetric Clusters* and *Shifted Clusters* distributions to drop to zero, since their clusters are are all exactly 125 points (meaning that a point's 126th nearest neighbor is almost guaranteed to be much farther away than its first nearest neighbor).

C.2. Gaussian KL-Divergence (GKL)

A common method for calculating KL-Divergence is to estimate continuous parameters from the observed distribution and then calculate the closed-form of Equation 10. Here, μ_p and σ_p are the mean and covariance matrix of the observed distribution, and d is the number of dimensions. We propose a measure based on the closed-form KL-divergence between two multivariate Gaussians in Equation 27, where the $tr()$ function is the sum of the diagonal elements of the matrix (Duchi, 2007). Although we don't expect our example distributions to be well-approximated by a normal distribution, this method shows up frequently in machine learning frameworks (e.g. Variational Autoencoders (Doersch, 2016), t-SNE (Van der Maaten & Hinton, 2008), Reinforcement Learning (Filippi et al., 2010)).

$$GKL = \frac{1}{2}(\mu_p^\top \mu_p + tr(\sigma_p) - d - \log|\sigma_p|) \qquad (27)$$

With this measure, we were surprised to find that Table 7 indicates that data spread becomes less complete as we move from the three spherical distributions, to the *Cone* distribution, and to the cluster distributions. However, GKL does not produce results that differ greatly from our other measures, and GKL similarly suffers from the weakness of KL-Divergence measures described in Section 7.2, in that the range for high dimensional distributions is much larger than it is for lower dimensional distributions.

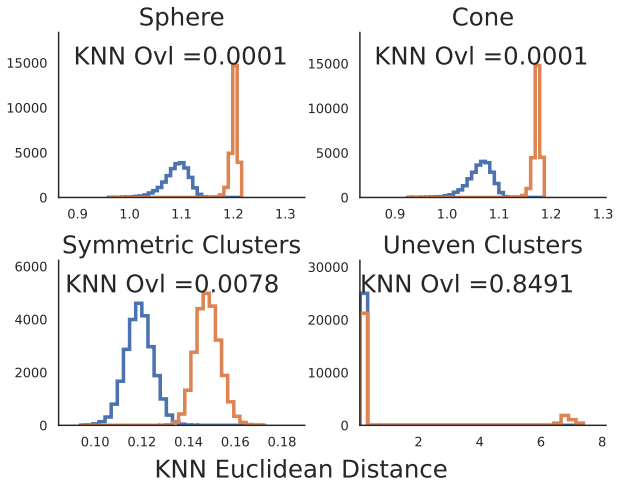

Figure 11. KNN Overlap Proportion for 100D *Sphere*, *Cone*, *Symmetric Clusters*, and *Shifted Clusters* distributions, with the nearest neighbor distances in blue, and the 100th nearest neighbor distances in orange

Table 7. Example distribution results for Gaussian KL-Divergence (GKL).

EXAMPLE DIST.	2D	10D	50D	100D
Shell	0.0013	0.0078	0.0478	0.0985
Nested Shell	0.0031	0.0122	0.0662	0.1344
Sphere	0.0011	0.0086	0.0500	0.0945
Cone	0.0926	0.7479	1.4945	1.8939
Symm. Clust.	0.5064	3.5500	18.5039	18.4996
Shifted Clust.	0.5902	4.3510	18.5038	18.4996
Uneven Clust.	0.5141	3.6196	18.5039	18.4996
Normal	0.0069	0.0130	0.0516	0.0993

D. Extended Results

Table 8. Example distribution results on Average Cosine Similarity (ACS) and I(V).

Example Distribution	2D		10D		50D		100D	
	ACS	I(V)	ACS	I(V)	ACS	I(V)	ACS	I(V)
Shell	0.0027	0.9737	0.0008	0.9916	0.0008	0.9977	0.0007	0.9988
Nested Shell	0.0027	0.9801	0.0008	0.9881	0.0008	0.9964	0.0007	0.9982
Sphere	0.0036	0.9613	0.001	0.9905	0.0008	0.9976	0.0007	0.9988
Cone	0.8176	0.9469	0.5728	0.9471	0.5201	0.9890	0.5119	0.9944
Symm. Clust.	0.0020	0.7888	0.0004	0.8857	0.0007	0.9643	0.0007	0.9822
Shifted Clust.	0.0109	0.9295	0.0108	0.8251	0.0023	0.8360	0.0012	0.8326
Uneven Clust.	0.0249	0.7797	0.0183	0.8972	0.0076	0.9664	0.0043	0.9819
Normal	0.0022	0.9969	0.0007	0.9934	0.0008	0.9974	0.0008	0.9988

Table 9. Example distribution results on Eigenvalue Ratio (ER) and Eigenvalue Early Enrichment (EEE).

Example Distribution	2D		10D		50D		100D	
	ER	EEE	ER	EEE	ER	EEE	ER	EEE
Shell	0.9021	0.0129	0.8380	0.0307	0.7943	0.0354	0.7845	0.0361
Nested Shell	0.8552	0.0195	0.7991	0.0381	0.7550	0.0417	0.7542	0.0422
Sphere	0.9118	0.0115	0.8210	0.0323	0.7845	0.0363	0.7889	0.0353
Cone	0.4173	0.1028	0.0807	0.1155	0.0185	0.0536	0.0090	0.0458
Symm. Clust.	0.1124	0.1995	0.0126	0.4987	0.0003	0.5503	0.0001	0.5394
Shifted Clust.	0.0915	0.2081	0.0070	0.5792	0.0021	0.6558	0.0007	0.6746
Uneven Clust.	0.1102	0.2004	0.0134	0.5067	0.0003	0.5506	0.0001	0.5397
Normal	0.7910	0.0292	0.7917	0.0398	0.7839	0.0368	0.7804	0.0362

Table 10. Example distribution results for Vasicek Ratio MSE (VRM) and Discrete KL-Divergence MSE (DKLM).

Example Distribution	2D		10D		50D		100D	
	VRM	DKLM	VRM	DKLM	VRM	DKLM	VRM	DKLM
Shell	0.1891	0.1349	0.0012	0.0003	0.0010	0.0000	0.0010	0.0000
Nested Shell	0.0465	0.0774	0.0013	0.0027	0.0014	0.0025	0.0014	0.0019
Sphere	0.0106	0.0169	0.0011	0.0003	0.001	0.0000	0.0009	0.0000
Cone	0.0686	0.0266	0.0603	0.0735	0.0172	0.0521	0.0097	0.0295
Symm. Clust.	0.1545	0.1434	0.1983	0.6150	0.2655	0.7538	0.2624	0.6200
Shifted Clust.	0.2143	0.5392	0.2401	0.6748	0.2398	22.5188	0.2386	20.5810
Uneven Clust.	0.1637	0.1470	0.1990	0.8154	0.2637	0.8012	0.2574	0.6333
Normal	0.0047	0.0190	0.0012	0.0017	0.0010	0.0001	0.0010	0.0000

Table 11. Example distribution results on Nearest Neighbor Entropy Ratio (NNR) and Nearest Neighbor KL-Divergence (NNKL).

Example Distribution	2D		10D		50D		100D	
	NNR	NNKL	NNR	NNKL	NNR	NNKL	NNR	NNKL
Shell	0.1323	7.4573	0.9981	0.8837	1.0028	0.3357	1.0024	0.2532
Nested Shell	0.3996	5.2696	0.9950	1.0005	0.9899	5.3053	0.9887	13.2125
Sphere	0.9749	0.9267	0.9994	0.8082	1.0026	0.3703	1.0023	0.2558
Cone	0.9350	1.2659	0.9939	1.1920	1.0022	0.4667	1.0022	0.2853
Symm. Clust.	0.8971	1.4874	0.8505	14.1981	0.7458	160.3722	0.7135	394.5467
Shifted Clust.	0.8312	1.9226	0.8111	17.5756	0.7188	177.1980	0.6882	429.2692
Uneven Clust.	0.9015	1.4215	0.8495	13.9207	0.7457	160.4552	0.7134	394.7038
Normal	0.9973	0.7260	0.9996	0.7083	0.9999	0.9290	1.0001	1.1751