d_χ-Privacy for Text and the Curse of Dimensionality

Hassan Jameel Asghar
hassan.asghar@mq.edu.au
Macquarie University
Sydney, Australia

Robin Carpentier
robin.carpentier@mq.edu.au
Macquarie University
Sydney, Australia

Benjamin Zi Hao Zhao
ben_zi.zhao@mq.edu.au
Macquarie University
Sydney, Australia

Dali Kaafar
dali.kaafar@mq.edu.au
Macquarie University
Sydney, Australia

Abstract

A widely used method to ensure privacy of unstructured text data is the multidimensional Laplace mechanism for d_χ-privacy, which is a relaxation of differential privacy for metric spaces. We identify an intriguing peculiarity of this mechanism. When applied on a word-by-word basis, the mechanism either outputs the original word, or completely dissimilar words, and very rarely outputs semantically similar words. We investigate this observation in detail, and tie it to the fact that the distance of the nearest neighbor of a word in any word embedding model (which are high-dimensional) is much larger than the relative difference in distances to any of its two consecutive neighbors. We also show that the dot product of the multidimensional Laplace noise vector with any word embedding plays a crucial role in designating the nearest neighbor. We derive the distribution, moments and tail bounds of this dot product. We further propose a fix as a post-processing step, which satisfactorily removes the above-mentioned issue.

Keywords

Differential privacy, word embeddings, multidimensional Laplace mechanism

1 Introduction

Unstructured text is the most common method of communication in the real-world, in the form of emails, sharing reports, and entering prompts to generative artificial intelligence (AI) models to quote a more contemporary example. In many scenarios, part of the text needs to be sanitized due to privacy reasons. Examples include authorship anonymization to protect identities of whistleblowers, redacting sensitive information before releasing documents as part of freedom of information requests, and ensuring that user submitted prompts to a third-party generative AI model do not contain sensitive company information.

With the increasing use of differential privacy [14] as a principled way of releasing data with privacy in many real-world applications, it has also been proposed as an automated way to achieve privacy in the text domain. More specifically, a generalization of differential privacy to metric spaces, called d_χ-privacy [11], has been proposed as a method to sanitize sensitive text [15, 16, 22, 23, 31, 35].[1] Informally d_χ-privacy ensures that it is harder to distinguish objects in a metric space that are closer to one another under the distance metric of the metric space than objects further away. This promises better utility than ordinary differential privacy as in many use cases it suffices to provide privacy up to a certain granularity. An analogy is location data; disclosing the city or postcode one resides in is less of a concern than the exact street address.

There are a number of ways in which d_χ-privacy can be used to santize text data. One common method is word-by-word sanitization through the *word-level multidimensional Laplace* mechanism for d_χ-privacy, applied in the following manner [15, 16, 22, 31, 35]. We assume a pre-trained machine learning model is available which vectorizes words, i.e., converts them into embeddings in a high-dimensional space. This pre-trained model is public information. The words form the vocabulary, and their corresponding embeddings define the embedding space. Given a word in a sentence to be sanitized, a noise vector is generated calibrated according to the privacy parameter ϵ, and then added to the word embedding resulting in a vector in the embedding space. Almost always, this does not correspond to the embedding of any word in the vocabulary of the embedding model. We can instead find the nearest neighbor (e.g., with respect to the Euclidean distance) to this noisy embedding in the embedding space, and output the resulting word. This method can be applied independently to each word, and the complete sentence can then be used as the sanitized text.[2]

When sanitizing text through this method we encountered a confounding observation [8]. If this method is applied several times on the same word, one expects to see the original word, followed by its closest neighbors as the most frequent words output by the mechanism, with the frequency dropping smoothly but exponentially as we move away from the original word. However, we observed that through the entire spectrum of values of ϵ, the mechanism almost always outputs either the original word or words which are very far off in distance and semantic similarity to the original word. The nearest neighbors of the original word are seldom output by the mechanism [8].

One may hasten to attribute this phenomenon to the high dimensional nature of the embedding space, since the Euclidean distance

Proceedings on Privacy Enhancing Technologies 2026(1), 224–241

https://doi.org/10.56553/popets-2026-0012

[1] d_χ-privacy is also commonly known as *metric privacy*.
[2] Apart from this use case, this mechanism has also been used for author obfuscation [15], and to provide privacy where each user's input is a set of one or more words [16].

metric is known to suffer from the so-called *curse of dimensionality* [2, 5]. However, this is not true in this instance, or not true in the way we may think, since the word embedding models are trained to ensure that the Euclidean distance is an effective metric to identify similar words in the embedding space [24, 29]. See Section 8 for a detailed discussion on this topic.

In this paper, we investigate this observation in detail and identify the post-processing step of the nearest neighbor search of the perturbed embedding in a high dimensional space as the culprit. Along the way, we unravel a number of other contributions which we believe will prove useful to construct better $d_\mathcal{X}$-privacy mechanisms for text data. The multidimensional Laplace mechanism is fundamental to $d_\mathcal{X}$-privacy due to its ease of implementation, and hence wide-spread use, akin to the Laplace mechanism [14] for ordinary differential privacy.

- We highlight the above-mentioned issue in a very commonly used $d_\mathcal{X}$-privacy mechanism for text data [15, 16, 22, 23, 31, 35], whereby across different values of ϵ and different word embedding models, almost always either the word is not replaced, or is replaced by a completely dissimilar word.
- While analyzing the above phenomenon it turns out that the dot product of the noise vector against any word embedding plays a crucial role. This distribution, which we call the *noisy dot product* distribution, has a length component, which is known to follow the gamma distribution [16], and an angular component. We derive the probability density function of the angular component and its moments.
- We show that the angular component is sub-Gaussian with variance $1/n$, where n is the number of dimensions. This means that the cosine of the angle of the noise vector with any embedding is within $O(1/\sqrt{n})$, and hence the noise vector is increasingly orthogonal to any embedding regardless of the distribution of words in the embedding model. We further prove tail-bounds on the noisy dot product distribution showing that its mass is concentrated within $O(\sqrt{n}/\epsilon)$.
- Through our analysis, we show that the aforementioned observation is related to the fact that in high dimensions the nearest neighbor of a word is more distant than the relative difference of the distances of its two nearest neighbors. We prove necessary conditions on the initial word to be output by the mechanism as opposed to its close neighbors by relating it to the noisy dot product distribution. Previous works [6, 9, 12, 34] have highlighted the vulnerability of outliers in $d_\mathcal{X}$-privacy. However, their observation is related to the data distribution containing some isolated points. In contrast, our work shows that in high-dimensional settings the outlier issue is the norm rather than the exception.
- We propose a possible mitigation as a further post-processing step, and show that the resulting mechanism gives better utility and behaves as expected. The advantage of our proposed fix is that it does not amend the original mechanism, and only adds an extra step. We have released the code of our experiments, including this fix, to promote reproducibility.[3]

[3]https://github.com/r-carpentier/dx-privacy-curse

2 The Unusual Behavior of Word-Level $d_\mathcal{X}$-Privacy

The Observation. We first show an example of applying $d_\mathcal{X}$-privacy on text data with the algorithm outlined in the introduction. Formal introduction to $d_\mathcal{X}$-privacy and concrete details of this algorithm are presented in Section 3.2. Consider the following text:

> Maria Gonzalez, a patient at Riverside Clinic, was diagnosed with depression on March 5, 2023. She currently lives at 789 Oak Drive, San Francisco. Maria has been prescribed medication and is undergoing weekly therapy sessions.

To sanitize this text we first choose an embedding model. Let us say we use the `GloVe` embedding model [29] with $n = 100$ dimensions. We then take the first word of the text, pass it through the embedding model to obtain its embedding $\mathbf{w}$. We then sample a noise vector $\boldsymbol{\eta}$ according to a distribution scaled to ϵ and n. For this example, we choose $\epsilon = 10$. We thus obtain the noisy embedding $\mathbf{w}^* = \mathbf{w} + \boldsymbol{\eta}$. This does not correspond to any word in the vocabulary of `GloVe`. We therefore, search the nearest neighbor in the embedding space of `GloVe` to $\mathbf{w}^*$ and output the corresponding word as the replacement to the original word. This process is repeated for each word[4] resulting in the following sanitized text:

> maria carvalho, full patient raised bottomland clinic, was diagnosed with evanston on 8 4, 2028 . she represent lives ‘ 789 poplar drive, st. antonio. maria deeply were prescribed medication deadlock subject undergone quiz therapy approaches.

In most cases, as above, the sentence is ill-structured and grammatically incorrect. This can be corrected by another post-processing step, for example by leveraging a generative AI model. The following text is obtained by asking `ChatGPT 4o`[5] to correct the grammar of the previous text:

> Maria Carvalho, a full-time patient at Bottomland Clinic, was diagnosed in Evanston on August 4, 2028. She resides at 789 Poplar Drive, St. Antonio. Maria was prescribed medication and has undergone various therapeutic approaches, including cognitive therapy.

As we can see after this step, the sentence is coherent with some of the exact names and dates replaced, which is desirable for privacy. Now, if we use smaller values of ϵ (more privacy) we expect the original words to be replaced by their distant neighbors, i.e., words that have little to no similarity with the original word. However, as we increase ϵ (less privacy), we would expect most words to either remain unchanged or replaced by synonyms. At least this is what we expect if we apply the usual Laplace mechanism [14] of differential privacy to one-dimensional data. Figure 1 shows an example of one-dimensional data, in which we plot the result of applying the Laplace mechanism with different values of ϵ (assuming sensitivity one) with the universe of values confined to the set of positive integers. The true value is $a = 400,000$, and after adding noise we round it to the nearest integer. Any integer value within $a \pm 100$ is

[4]Commas and periods in the original text were excluded from the sanitization for readability

[5]See https://chatgpt.com.

considered a close neighbor of a and all other values considered distant neighbors. For each value of ϵ we sample 10,000 noisy answers and report the proportion of times the original, close neighbors and distant neighbors are output by the mechanism. Initially with $\epsilon = 0.001$, distant neighbors are output more frequently (Figure 1, left). This trend is quickly flipped as we increase ϵ. With $\epsilon > 1$ we see that it is either the original value or the close neighbors that are output by the mechanism and very rarely any distant neighbors (Figure 1, right).

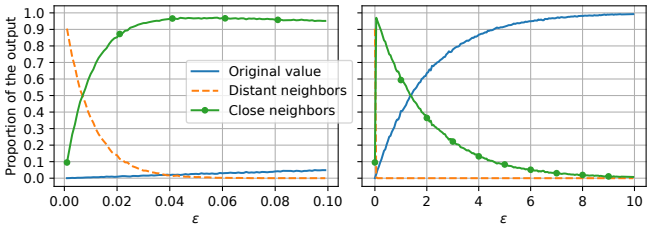


Figure 1: The proportion of times the original value, its close neighbors and distant neighbors are output by the Laplace mechanism of differential privacy. The original value is $a = 400,000$, close neighbors are all integers within $a \pm 100$, and all other integer values are distant neighbors.

However, we observed that through the text counterpart of this algorithm, i.e., the multidimensional Laplace mechanism, either the initial word is output by the mechanism, or very far off words, and the nearest neighbors are seldom encountered. This is visualized in Figure 2 where we show the proportion of times the original word is output by the mechanism, against its close neighbors, the first 100 nearest neighbors, and distant neighbors, which constitute the rest of the words in the vocabulary. These plots were obtained by randomly sampling 5,000 words.

At lower values of ϵ only far off words are selected, but as we increase ϵ, the original word dominates at the expense of its close neighbors. The end-result of course is bad privacy-utility tradeoff. The figure shows the pattern for the `GloVe-Wiki` and `fastText` embedding models, both with 300 dimensions. The result is similar for other embedding models which are detailed in Section 3.1. For a more detailed description of this observation please see [8].

These results are also consistent with the results shown in [16, 31], where the authors show the rather high frequency of unmodified words but no commentary is provided as to why this may be the case other than attributing it to the behavior of the mechanism for higher values of ϵ. Indeed, the trend is more peculiar at smaller values of ϵ.

An Illustrative Two-Dimensional Example. The issues with $d_\mathcal{X}$-privacy over high-dimensional word embeddings can be illustrated through an analogy of $d_\mathcal{X}$-privacy for location data.[6] Assume that we have a dataset containing the resident states of the inhabitants of the United States of America (USA). We exclude the state of Alaska in this example. We wish to release this dataset with privacy, meaning that an individual can plausibly deny that he/she resides in a

[6] $d_\mathcal{X}$-privacy for location data is studied in detail in [3] where the authors term the notion, geo-indistinguishability.

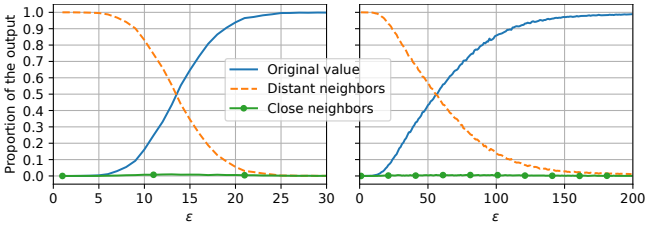


Figure 2: The proportion of times the original word, its close neighbors and distant neighbors are output by the multidimensional Laplace mechanism of $d_\mathcal{X}$-privacy on the `Glove-Wiki` (left) and `Word2Vec` (right) word embedding models. Close neighbors are the first 100 nearest neighbors, and all other words are distant neighbors.

particular state in the released dataset. To do so, we perturb each individual's location using $d_\mathcal{X}$-privacy. Assume each person's location is given as a point on the real plane (latitude and longitude, if you like). One way to satisfy $d_\mathcal{X}$-privacy is to sample a noise vector, add it to the original location, and then find the nearest state to the perturbed location. The last step is a post-processing step, and maintains $d_\mathcal{X}$-privacy. The noise is a two-dimensional vector sampled from a particular distribution [3] scaled by the privacy parameter ϵ and the distance metric is the Euclidean distance in practice which we describe in detail in Section 3. This is exactly the mechanism commonly used to provide (word-level) $d_\mathcal{X}$-privacy in text data [15, 16, 22, 31, 35], except that location coordinates are replaced by word embeddings and states are replaced by words in the vocabulary.

Now consider a resident of the state of Hawaii. For values of ϵ within a certain range, the sampled perturbed location will fall in the North Pacific Ocean with overwhelming probability. Since our universe of locations is the set of states in USA, the nearest location is Hawaii again. Compare this to a resident of Kansas. For the same value of ϵ, the noisy location in this case is likely to land on the neighboring states and beyond. Figure 3 illustrates this point. This observation has both privacy and utility implications.

Figure 3: An analogy of the issues of applying $d_\mathcal{X}$-privacy to high-dimensional word embeddings using location data. The nearest neighbor of the perturbed location of the residents of Hawaii tends to be Hawaii itself, unlike the states in mainland USA. The circles indicate the probability density of the noise vector. The map is extracted from Google Maps.

From the privacy point-of-view, the residents of mainland USA get more protection than the residents of Hawaii for the same value of ϵ. In other words, the level of privacy provided depends on the structure of the dataset. Datasets with with irregularly scattered points will end up providing less privacy for isolated points. This issue has already been highlighted in previous works [9, 34]. Here, we would like to point out that this privacy issue does not arise if we scale the noise according to the sensitivity of the distance function which makes the resulting privacy notion equivalent to ordinary ϵ-differential privacy [14]. However, applications of this $d_\mathcal{X}$-privacy mechanism to text domain do not use sensitivity to scale noise [15, 16, 22, 31, 35]; the idea being that we only need to make it hard to distinguish between nearby points. For location based applications, one specifies a radius, which serves as a proxy for sensitivity, and we say that the corresponding mechanism provides $d_\mathcal{X}$-privacy for locations within this radius [3, 21].

One may also dismiss the above observation as an outlier; an issue that relates only to a few isolated points in the dataset. However, as we shall demonstrate, in higher dimensions the nearest neighbor of any word is at a considerable distance away from it, and the distance increases as we increase the size of dimension. Thus, in higher dimensions, this is more the norm than an anomaly.

The above observation also creates issues from a utility perspective. With higher values of ϵ (less privacy), the original word is returned by the nearest neighbor search most of the times. As we decrease ϵ to provide more privacy, one expects the frequency of the nearest neighbors of the original word to be selected more than distant words. However, since the variance of the noise is now larger, this happens less often than expected. The undesired result is that either the word is not changed at all, or if it is changed, it is replaced by a distant word with little to no semantic similarity with the original word. Again this issue is exacerbated in higher dimensions, as the distance between a word and its nearest neighbor is higher than the relative difference in the distances to its two successive neighbors.

3 Background and Notation

Notations. The n-dimensional real space is denoted by $\mathbb{R}^n$. A vector from $\mathbb{R}^n$ will be denoted in bold face, e.g., $\mathbf{x}$. Let $\|\mathbf{x}\| = \sqrt{\sum_{i=1}^n |x_i^2|}$ denote the Euclidean norm of $\mathbf{x}$, where x_i is the ith element of $\mathbf{x}$. The dot product between two vectors $\mathbf{x}$ and $\mathbf{y}$ is denoted as $\langle \mathbf{x}, \mathbf{y} \rangle$. The following is an elementary fact:

$$\langle \mathbf{x}, \mathbf{y} \rangle = \|\mathbf{x}\| \|\mathbf{y}\| \cos \theta_{\mathbf{x},\mathbf{y}},$$

where $0 \leq \theta_{\mathbf{x},\mathbf{y}} \leq \pi$, is the angle between $\mathbf{x}$ and $\mathbf{y}$.[7] We can interpret $\|\mathbf{y}\| \cos \theta_{\mathbf{x},\mathbf{y}}$ as the length of the projection of $\mathbf{y}$ on $\mathbf{x}$. For any two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ the Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$ is $\|\mathbf{x} - \mathbf{y}\|$. The Euclidean distance is a metric as it satisfies the following properties of (1) (positivity) $\|\mathbf{x} - \mathbf{y}\| \geq 0$ with equality if and only if $\mathbf{x} = \mathbf{y}$, (2) (symmetry) $\|\mathbf{x} - \mathbf{y}\| = \|\mathbf{y} - \mathbf{z}\|$, and (3) (triangle inequality) $\|\mathbf{x} - \mathbf{y}\| \leq \|\mathbf{x} - \mathbf{z}\| + \|\mathbf{z} - \mathbf{y}\|$, for any $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{R}^n$ [27].

[7] Keeping $\mathbf{x}$ fixed, if the direction of $\mathbf{y}$ is chosen randomly in the plane containing the two vectors, then $\theta_{\mathbf{x},\mathbf{y}}$ is the smallest of the two possible angles. This is well-defined. To see this let θ_1 and θ_2 denote the two possible angles, and let $\theta_1 > \pi$. Then $\theta_2 = 2\pi - \theta_1 < \pi$, and note that $\cos \theta_1 = \cos(2\pi - \theta_2) = \cos \theta_2$.

3.1 Vector Representation of Words

In recent years, a number of machine learning models have sprung up which produce vector representations of words, which we call word embeddings for short. These pre-trained word embeddings can be downloaded and used for natural language processing tasks. Notable examples include `Word2Vec` [24], `GloVe` [29], and `fastText` [20]. These word embeddings lie on a high-dimensional real vector space with dimensions ranging from 50 to 300, and even higher. Some embeddings, such as `GloVe`, come in different dimensions. Despite the high-dimensional space the word embeddings maintain distance-based semantic similarities. In other words, the Euclidean distance is an effective method to obtain nearest neighbors of a word in terms of semantic similarity to the target word.[8] Furthermore, these embeddings maintain linear relationships between words, e.g., king minus queen equals man minus woman [29]. For a more detailed introduction to vector representation of words and models producing word embeddings, see [36, §15]. The pre-trained word embeddings used in this paper are shown in Table 1. We will use the terms 'embedding model' and 'vocabulary' interchangeably.

Vocabulary	Dimensions	Words
`GloVe-Twitter`	25, 50, 100, 200	1,193,514
`GloVe-Wiki`	50, 100, 200, 300	400,000
`Word2Vec`	300	3,000,000
`fastText`	300	2,519,370

Table 1: Word embedding models used in this work.

3.2 $d_\mathcal{X}$-Privacy and Applications to Text Data

Let $\mathcal{D} \subseteq \mathbb{R}^n$ be the input domain, which for our purpose is the embedding space. Let $\mathbf{x} \in \mathcal{D}$ denote a word.

Definition 1 (Differential Privacy [14])**.** An algorithm $\mathcal{M} : \mathcal{D} \to R$ satisfies ϵ-(local) differential privacy if for all words $\mathbf{x}, \mathbf{y} \in \mathcal{D}$ and for all possible subsets $S \subseteq R$ we have

$$\Pr[\mathcal{M}(\mathbf{x}) \in S] \leq e^\epsilon \Pr[\mathcal{M}(\mathbf{y}) \in S]$$

Definition 2 ($d_\mathcal{X}$-privacy [11])**.** Let d be a metric on $\mathcal{D}$. An algorithm $\mathcal{M} : \mathcal{D} \to R$ satisfies $\epsilon d_\mathcal{X}$-privacy for the metric d if for all words $\mathbf{x}, \mathbf{y} \in \mathcal{D}$ and for all possible subsets $S \subseteq R$ we have

$$\Pr[\mathcal{M}(\mathbf{x}) \in S] \leq e^{\epsilon d(\mathbf{x},\mathbf{y})} \Pr[\mathcal{M}(\mathbf{y}) \in S]$$

Note that if d is the Hamming distance d_H then we recover the original definition of ϵ-differential privacy, since $d_\mathrm{H}(\mathbf{x}, \mathbf{y}) = 1$, whenever $\mathbf{x} \neq \mathbf{y}$. A key advantage of $d_\mathcal{X}$-privacy over ordinary differential privacy is in terms of utility: the former treats all inputs equally, and indistinguishability is with respect to all inputs, whereas the latter provides more privacy with respect to similar inputs than far away inputs, when measured according to the distance metric. Like ϵ-DP, $d_\mathcal{X}$-privacy enjoys the properties of immunity to post-processing, and composition of privacy guarantees [11]. In particular, applying a $d_\mathcal{X}$-privacy mechanism independently to each of the words in a sequence (sentence) of m words, makes the resulting composition of these mechanisms $m\epsilon d_\mathcal{X}$-private [21].

[8] See https://nlp.stanford.edu/projects/glove/.

Finally, we would like to point out a particular (simplified) result from the original paper by Dwork et al [14] on ϵ-DP, which relates to general metric spaces. Let d be a metric on the domain $\mathcal{D}$, and let the sensitivity of the distance metric d be defined as $\Delta d = \max_{\mathbf{x},\mathbf{y}\in\mathcal{D}} d(\mathbf{x},\mathbf{y})$, then the mechanism $\mathcal{M}$ which on input $\mathbf{x} \in \mathcal{D}$ outputs a $\mathbf{y} \in \mathcal{D}$ with probability proportional to

$$\Pr[\mathcal{M}(\mathbf{x}) = \mathbf{y}] \propto \exp\left(-\frac{\epsilon d(\mathbf{x},\mathbf{y})}{2\Delta d}\right),$$

is ϵ-differentially private, provided such a probability density function exists [14, §3.3]. We remark that the result in [14] does not contain the negative sign in the proportionality above, but it is easy to verify that the result still holds. As we shall see next, the $d_\mathcal{X}$-private mechanism samples a word embedding proportional to above except that there is no scaling according to sensitivity.

$d_\mathcal{X}$-Private Noise. In order to add multidimensional Laplace noise to a word embedding $\mathbf{w} \in \mathbb{R}^n$, the method is to add a noise vector $\boldsymbol{\eta} \in \mathbb{R}^n$ from a distribution with probability density $\propto \exp(-\epsilon\|\boldsymbol{\eta}\|)$ [14, 16].[9] To sample from this distribution, one first samples n zero-mean, unit-variance Gaussians to produce an n-dimentional vector $\mathbf{u}$ whose resulting probability density is

$$\frac{1}{2\pi^{n/2}} e^{(-\frac{\|\mathbf{u}\|}{2})} \tag{1}$$

The vector is then normalized to produce the unit vector $\hat{\mathbf{u}} = \frac{\mathbf{u}}{\|\mathbf{u}\|}$. This means that $\hat{\mathbf{u}}$ is distributed uniformly at random on the surface of a hypersphere of n-dimensions with unit radius (see for example [7, §2.5]). Next we find the "length" of the noise vector. The number of points on an n-dimensional hypersphere with radius r is proportional to r^{n-1}, with each point having density proportional to $\exp(-\epsilon\|\boldsymbol{\eta}\|)$ (the requirement above). Thus, we need a probability density function proportional to $r^{n-1}\exp(-\epsilon\|\boldsymbol{\eta}\|)$. This means that we should sample the noise as $\boldsymbol{\eta} = r\hat{\mathbf{u}}$, where r has the Gamma distribution [16], with probability density

$$f_\mathrm{G}(r) = \frac{1}{\Gamma(n)\epsilon^{-n}} r^{n-1} e^{-\epsilon r}. \tag{2}$$

Note that $\|\boldsymbol{\eta}\| = \|r\hat{\mathbf{u}}\| = r$, and hence this is the required distribution. For completeness, we provide a proof in Appendix B. Figure 4 illustrates the distribution for $n = 2$.

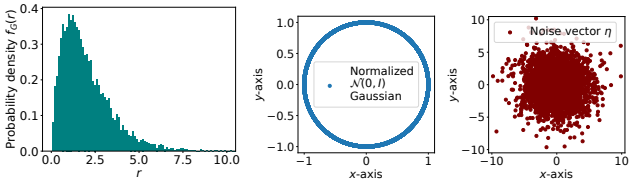
Nearest Neighbor Search. Almost always, the perturbed embedding $\mathbf{w}^* = \mathbf{w} + \boldsymbol{\eta}$ is not a member of $\mathcal{D}$. Thus, a nearest neighbor search is performed to find an embedding $\mathbf{x}^* \in \mathcal{D}$ which is closest to $\mathbf{w}^*$ in the Euclidean distance [16]. That is, we find the embedding:

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathcal{D}} \|\mathbf{w}^* - \mathbf{x}\|. \tag{3}$$

This vector is then the output of the $d_\mathcal{X}$-private mechanism.

Proof of Privacy. The above mechanism is $d_\mathcal{X}$-private since the resulting noise word embedding $\mathbf{w}^*$ is output from the distribution proportional to $\exp(-\epsilon\|\boldsymbol{\eta}\|) = \exp(-\epsilon d(\mathbf{w}^*, \mathbf{w}))$, where d is the Euclidean distance. The nearest neighbor search is a post-processing step, and the result can be extended to a sequence (sentence) of multiple words by applying the mechanism independently on each word, and then combining the results, invoking the composition property of $d_\mathcal{X}$-privacy. Feyisetan et al also provide a direct proof of this result in [16]. We note that, while most commonly employed, this is not the only method of applying differential privacy for text sanitization. Other methods include other word-level $d_\mathcal{X}$-privacy mechanisms [9, 17], sentence-level differential privacy mechanisms [19, 23], and mechanisms for more advanced natural language processing tasks [32]. We describe them in more detail in Section 8.

[9] As mentioned above, the result from [14] is for ordinary ϵ-differential privacy as opposed to $d_\mathcal{X}$-privacy.



(a) (b) (c)

Figure 4: The noise distribution for $n = 2$ with 10,000 sampled points. Subfigure a shows the distribution of the length of noise vector, i.e., r through Eq. (2). Subfigure b is the unit vector produced by normalizing the 2D Gaussian $\mathcal{N}(0, I)$ (Eq. (1)), where I is the identity matrix. Subfigure c shows the distribution of the resulting noise vector $\boldsymbol{\eta}$.

4 Conditions for Nearest Neighbor Selection and the Noisy Dot Product Distribution

We are first interested in the conditions when the perturbed embedding $\mathbf{w}^*$ is closer to $\mathbf{w}$, i.e., the original word, than any of the neighbors of $\mathbf{w}$. When this is the case, the original word will be chosen as output by the mechanism. Formalizing these conditions is necessary to understand why the original word is overwhelmingly chosen over its close neighbors (Figure 2). To this end, we have the following result.

Theorem 1. *Let $\boldsymbol{\eta}$ be a noise vector. Let $\mathbf{w} \in \mathcal{D}$ be a word embedding, and let $\mathbf{w}^* = \mathbf{w} + \boldsymbol{\eta}$ be the perturbed embedding. Let $\mathbf{x} \in \mathcal{D}$ be any embedding different from $\mathbf{w}$. Then $\mathbf{w}^*$ is closer to $\mathbf{w}$ than any of its neighbors if for all neighbors $\mathbf{x}$ of $\mathbf{w}$, we have*

$$r\cos\theta_{\boldsymbol{\eta},\mathbf{x}-\mathbf{w}} < \frac{1}{2}\|\mathbf{w} - \mathbf{x}\|$$

Proof. Let $\mathbf{x} \in \mathcal{D}$ be any neighbor of $\mathbf{w}$. We have:

$$\begin{aligned}
\|\mathbf{w}^* - \mathbf{w}\|^2 &< \|\mathbf{w}^* - \mathbf{x}\|^2 \\
\Rightarrow \|\boldsymbol{\eta}\|^2 &< \|\mathbf{w} - \mathbf{x} + \boldsymbol{\eta}\|^2 \\
\Rightarrow \|\boldsymbol{\eta}\|^2 &< \|\mathbf{w} - \mathbf{x}\|^2 + \|\boldsymbol{\eta}\|^2 + 2\langle \mathbf{w} - \mathbf{x}, \boldsymbol{\eta}\rangle \\
\Rightarrow -2\langle \mathbf{w} - \mathbf{x}, \boldsymbol{\eta}\rangle &< \|\mathbf{w} - \mathbf{x}\|^2 \\
\Rightarrow \langle \boldsymbol{\eta}, \mathbf{x} - \mathbf{w}\rangle &< \frac{1}{2}\|\mathbf{w} - \mathbf{x}\|^2 \\
\Rightarrow r\|\mathbf{x} - \mathbf{w}\|\cos\theta_{\boldsymbol{\eta},\mathbf{x}-\mathbf{w}} &< \frac{1}{2}\|\mathbf{w} - \mathbf{x}\|^2 \\
\Rightarrow r\cos\theta_{\boldsymbol{\eta},\mathbf{x}-\mathbf{w}} &< \frac{1}{2}\|\mathbf{w} - \mathbf{x}\|,
\end{aligned}$$

where in the second last step, we have used Theorem 3. □

An interpretation of the above is as follows: The noisy embedding will be closer to the original vector $\mathbf{w}$ than its nearest neigbour $\mathbf{x}$ if the length of the projection of the noise vector on $\mathbf{x}-\mathbf{w}$ is less than half the length of $\mathbf{x}-\mathbf{w}$. Figure 5a illustrates Theorem 1. This can also be explained easily via a nice geometric illustration. Suppose $0 \leq \theta_{\boldsymbol{\eta},\mathbf{x}-\mathbf{w}} \leq \frac{\pi}{2}$, as shown in Figure 5b, where we denote the vectors by their end-points in the real space. It is easy to see that if $A < B$, then $C^2 = A^2 - E^2 < B^2 - E^2 = D^2$, and when $C < D$ then $A^2 = C^2 - E^2 < D^2 - E^2 = B^2$. The line segment C is precisely the length of the projection of the noise vector on $\mathbf{x}-\mathbf{w}$. If $\frac{\pi}{2} \leq \theta_{\boldsymbol{\eta},\mathbf{x}-\mathbf{w}} \leq \pi$ then clearly $A < B$, which is supported by the fact that the projection of the noise vector on $\mathbf{x}-\mathbf{w}$ is negative.

Next we are interested in knowing when the nearest neighbor $\mathbf{x}$ of the original word $\mathbf{w}$ is closer to the noisy embedding $\mathbf{w}^*$ than any other neighbor $\mathbf{y}$ of $\mathbf{w}$.

Theorem 2. *Let $\boldsymbol{\eta}$ be a noise vector. Let $\mathbf{w} \in \mathcal{D}$ be a word embedding, and let $\mathbf{w}^* = \mathbf{w} + \boldsymbol{\eta}$ be the perturbed embedding. Let $\mathbf{x} \in \mathcal{D}$ be the nearest neighbor of $\mathbf{w}$. Let $\mathbf{y} \in \mathcal{D}$ be any other embedding different from $\mathbf{w}$ and $\mathbf{x}$. Then $\mathbf{w}^*$ is closer to $\mathbf{x}$ than $\mathbf{y}$ if*

$$\|\mathbf{w}-\mathbf{x}\| \cos\theta_{\mathbf{w}-\mathbf{x},\mathbf{y}+\mathbf{x}} + r\cos\theta_{\boldsymbol{\eta},\mathbf{y}-\mathbf{x}} < \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|$$

Proof. See Appendix A. □

An interpretation of the above is as follows: If the sum of projections of the noise vector and that of $\mathbf{w}-\mathbf{x}$ on $\mathbf{y}-\mathbf{x}$ is less than half its length, then $\mathbf{x}$ will be closer to the noisy embedding than $\mathbf{y}$. Figure 5c shows a graphical illustration of this result. The result of the theorem can also be explained in a manner similar to Figure 5b. We will revisit these results in Section 6 to see how often they are true for word embeddings. For now we notice that both the results of Theorem 1 and 2 involve the product of the length r of the noise vector and the cosine of its angle with a vector in $\mathbb{R}^n$. The following theorem characterizes this distribution.

Theorem 3. *Let $n \geq 2$. Let $\boldsymbol{\eta}$ be a noise vector. Let $\mathbf{w} \in \mathbb{R}^n$ be a non-zero vector. Then*

$$\langle \boldsymbol{\eta}, \mathbf{w} \rangle = r\|\mathbf{w}\| \cos\theta_{\boldsymbol{\eta},\mathbf{w}}, \tag{4}$$

where $r \sim f_G$ given in Eq. (2) and $k = \cos\theta_{\boldsymbol{\eta},\mathbf{w}} \sim f_B$ given as

$$f_B(k) = \frac{1}{B(\frac{n-1}{2}, \frac{1}{2})}(1-k^2)^{\frac{n-1}{2}-1}, \quad k \in [-1,1], \tag{5}$$

where $B(\cdot,\cdot)$ is the beta function defined as:

$$B(a,b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)},$$

for all real numbers $a, b > 0$.

Proof. From the definition of the dot product $\langle \boldsymbol{\eta}, \mathbf{w} \rangle$ equals

$$\|\boldsymbol{\eta}\|\|\mathbf{w}\| \cos\theta_{\boldsymbol{\eta},\mathbf{w}} = \|r\hat{\mathbf{u}}\|\|\mathbf{w}\| \cos\theta_{\boldsymbol{\eta},\mathbf{w}} = r\|\mathbf{w}\| \cos\theta_{\boldsymbol{\eta},\mathbf{w}}.$$

Assuming $\|\mathbf{w}\| \neq 0$, expanding the left hand side, we have:

$$\sum_{i=1}^{n} \eta_i w_i = r\|\mathbf{w}\| \cos\theta_{\boldsymbol{\eta},\mathbf{w}}$$

$$\sum_{i=1}^{n} r\hat{u}_i w_i = r\|\mathbf{w}\| \cos\theta_{\boldsymbol{\eta},\mathbf{w}}$$

$$\sum_{i=1}^{n} \hat{u}_i \frac{w_i}{\|\mathbf{w}\|} = \cos\theta_{\boldsymbol{\eta},\mathbf{w}}$$

$$\langle \hat{\mathbf{u}}, \hat{\mathbf{w}} \rangle = \cos\theta_{\boldsymbol{\eta},\mathbf{w}},$$

where $\hat{\mathbf{w}}$ is a unit vector obtained from $\mathbf{w}$ by dividing it by its norm. Being unit vectors, the two vectors are on the surface of the n-dimensional hypersphere of radius one. Furthermore, $\hat{\mathbf{u}}$ is uniformly distributed on the surface of this hypersphere by construction. Since the dot product is rotationally invariant, we can align $\hat{\mathbf{w}}$ to align with the unit vector $\hat{\mathbf{e}}_1$ whose first coordinate is 1 and all other coordinates are 0. Since $\hat{\mathbf{u}}$ is uniformly distributed on the surface of the hypersphere, the rotated vector is still uniformly distributed. Thus, without fear of ambiguity, let us also call this vector $\hat{\mathbf{u}}$. Then we see that:

$$\cos\theta_{\boldsymbol{\eta},\mathbf{w}} = \langle \hat{\mathbf{u}}, \hat{\mathbf{e}}_1 \rangle = \hat{u}_1 = \frac{u_1}{\|\mathbf{u}\|}, \tag{6}$$

where $u_1 \sim \mathcal{N}(0,1)$ and $\|\mathbf{u}\|$ is the norm of an n-dimensional vector each element of which is independently distributed as $\mathcal{N}(0,1)$. We now find the PDF of the distribution in Eq. (6).[10]

Let U_i be random variables distributed as $\mathcal{N}(0,1)$, for $1 \leq i \leq n$. We are interested in the distribution of

$$K = \frac{U_1}{\sqrt{U_1^2 + U_2^2 + \cdots + U_n^2}}$$

The range of this variable is in the interval [-1, 1]. Let $-1 < k < 0$. Then we see that

$$\begin{aligned}
&\Pr\left[\frac{U_1}{\sqrt{U_1^2 + U_2^2 + \cdots + U_n^2}} \leq k\right] \\
&= \Pr\left[U_1^2 \geq k^2(U_1^2 + U_2^2 + \cdots + U_n^2) \mid U_1 < 0\right] \\
&= \Pr\left[\frac{U_1^2}{U_2^2 + \cdots + U_n^2} \geq \frac{k^2}{1-k^2} \mid U_1 < 0\right] \\
&= \Pr\left[\frac{(n-1)U_1^2}{U_2^2 + \cdots + U_n^2} \geq \frac{(n-1)k^2}{1-k^2} \mid U_1 < 0\right] \\
&= \frac{1}{2}\Pr\left[\frac{(n-1)U_1^2}{U_2^2 + \cdots + U_n^2} \geq \frac{(n-1)k^2}{1-k^2}\right],
\end{aligned} \tag{7}$$

where in the last step we have used the fact that U_1 is symmetric. Now U_1^2 is a chi-squared variable with 1 degree of freedom and $U_2^2 + \cdots + U_n^2$ is a chi-squared variable with $n-1$ degrees of freedom [26, §4.3]. Thus, the ratio $\frac{(n-1)U_1^2}{U_2^2+\cdots+U_n^2}$ is an F-distributed random variable with degrees of freedom 1 and $n-1$ [26, §4.4]. The CDF of the F-distributed random variable X with 1 and $n-1$ degrees of freedom is given by:[11]

$$F_X(x; 1, n-1) = I_{\frac{x}{n-1+x}}\left(\frac{1}{2}, \frac{n-1}{2}\right), \tag{8}$$

where $I_y(a,b)$ is the regularized incomplete beta function given as

$$I_y(a,b) = \frac{B_y(a,b)}{B(a,b)} = \frac{1}{B(a,b)}\int_0^y t^{a-1}(1-t)^{b-1}\,dt,$$

[10] The derivation is taken from https://math.stackexchange.com/questions/185298/random-point-uniform-on-a-sphere. We reproduce it here to add missing details.
[11] See for example: https://mathworld.wolfram.com/F-Distribution.html

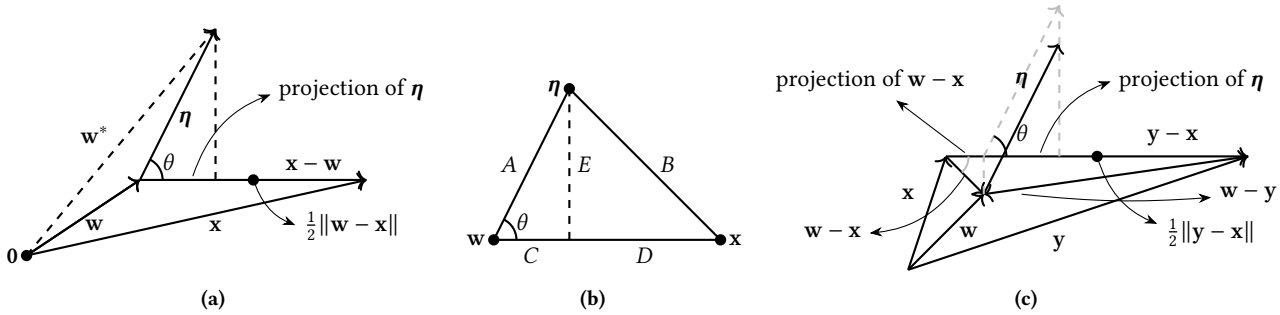


Figure 5: Illustrations of Theorems 1 and 2. Subfigure a illustrates Theorem 1, Subfigure b shows a geometric interpretation of the result of Theorem 1, and Subfigure c illustrates Theorem 2.

for $a, b > 0$. The function satisfies the relation [30, §6.4]:

$$I_{1-y}(b, a) = 1 - I_y(a, b). \tag{9}$$

Now, substituting $x = ((n-1)k^2)/(1-k^2)$ in $x/(n-1+x)$ we get

$$\frac{x}{n-1+x} = \frac{(n-1)k^2}{1-k^2} \frac{1-k^2}{(n-1)(1-k^2)+(n-1)k^2} = k^2$$

Thus, combining this result and using Eqs. (8) and (9), we get for $-1 < k < 0$

$$\begin{aligned}
\Pr[K \le k] &= \Pr\left[\frac{U_1}{\sqrt{U_1^2 + U_2^2 + \cdots + U_n^2}} \le k\right] \\
&= \frac{1}{2}\Pr\left[\frac{(n-1)U_1^2}{U_2^2 + \cdots + U_n^2} \ge \frac{(n-1)k^2}{1-k^2}\right] \\
&= \frac{1}{2}\left(1 - \Pr\left[\frac{(n-1)U_1^2}{U_2^2 + \cdots + U_n^2} \le \frac{(n-1)k^2}{1-k^2}\right]\right) \\
&= \frac{1}{2}\left(1 - F_X\left(x; 1, n-1\right)\right) \\
&= \frac{1}{2}\left(1 - I_{k^2}\left(\frac{1}{2}, \frac{n-1}{2}\right)\right) \\
&= \frac{1}{2} I_{1-k^2}\left(\frac{n-1}{2}, \frac{1}{2}\right)
\end{aligned} \tag{10}$$

Now taking the derivative of the integrand in $I_{1-k^2}((n-1)/2, 1/2)$ with respect to k with $-1 < k < 0$, we get:

$$\begin{aligned}
&\frac{d}{dk}\int_0^{1-k^2} t^{(n-1)/2-1}(1-t)^{1/2-1}\,dt \\
&= (1-k^2)^{\frac{n-1}{2}-1}(1-(1-k^2))^{1/2}(-2k) \\
&= -2(1-k^2)^{\frac{n-1}{2}-1}\frac{k}{\sqrt{k^2}} \\
&= -2(1-k^2)^{\frac{n-1}{2}-1}\frac{k}{|k|} \\
&= 2(1-k^2)^{\frac{n-1}{2}-1}
\end{aligned}$$

Thus, from Eq. (10), if we denote the PDF of K by f_{B}, we get for $-1 < k < 0$,

$$f_B(k) = \frac{1}{2}\frac{2(1-k^2)^{\frac{n-1}{2}-1}}{B(\frac{n-1}{2}, \frac{1}{2})} = \frac{1}{B(\frac{n-1}{2}, \frac{1}{2})}(1-k^2)^{\frac{n-1}{2}-1}.$$

Finally, since K is symmetric, the above is the PDF for $k \in [-1, 1]$. □

Remark. We call the above, the *noisy dot product* distribution. In light of the above theorem, we call the random variable R distributed as f_{G} the *length component* of this distribution and the random variable $K = \cos\theta_{\boldsymbol{\eta},\mathbf{w}}$ distributed as f_{B} with respect to *any* word embedding or vector in $\mathbb{R}^n$ as the *angular component* of the distribution, with $Z = RK$ denoting the overall distribution.

5 Moments and Tail Bounds of the Noisy Dot Product Distribution

In order to rule out any unusual behavior of the noisy dot product distribution $Z = RK$ in higher dimensions, we explore its properties in detail. These properties include its probability density function (PDF), cumulative distribution function (CDF), expectation, variance, and tail bounds of its components, i.e., R and K. In the process, we also find an expression for all moments of the component K.

5.1 CDF and PDF of the Distribution

We are interested in:

$$F_Z(z) = \Pr(Z \le z) = \Pr(RK \le z)$$

Now, R and K are independent. Also, the density function of R is non-zero on positive values of r and that of K is non-zero on $-1 \le k \le 1$. Furthermore, if $K = z/r$, then $r \ge |z|$, for K to be less than or equal to 1. Therefore, we get:

$$\begin{aligned}
F_Z(z) &= \Pr[RK \le z] \\
&= \int_{-\infty}^{\infty} \Pr[K \le z/R \mid R = r] f_G(r) dr \\
&= \int_{|z|}^{\infty} \Pr[K \le z/R \mid R = r] f_G(r) dr \\
&= \int_{|z|}^{\infty} \Pr[K \le z/r] f_G(r) dr \\
&= \int_{|z|}^{\infty} \left(\int_{-1}^{z/r} f_B(k) dk\right) f_G(r) dr \\
&= \int_{|z|}^{\infty} f_G(r) \left(\int_{-1}^{z/r} f_B(k) dk\right) dr
\end{aligned}$$

$$= \frac{1}{\Gamma(n)\epsilon^{-n}B(\frac{n-1}{2},\frac{1}{2})}\int_{|z|}^{\infty} r^{n-1}e^{-\epsilon r} \times\left(\int_{-1}^{z/r}\left(1-k^2\right)^{\frac{n-1}{2}-1}dk\right)dr \tag{11}$$

Taking the derivative of the above with respect to z using the fundamental theorem of calculus gives us the PDF of this distribution:

$$\begin{aligned} f_Z(z) &= \frac{d}{dz}\int_{|z|}^{\infty} f_G(r)\left(\int_{-1}^{z/r} f_B(k)dk\right)dr \\ &= \frac{d}{dz}\int_{0}^{\infty} f_G(r)\left(\int_{-1}^{z/r} f_B(k)dk\right)dr \\ &= \int_0^\infty f_G(r)f_B(z/r)\frac{1}{r}dr \qquad (12) \\ &= \int_{|z|}^\infty f_G(r)f_B(z/r)\frac{1}{r}dr \qquad (13) \end{aligned}$$

The last equality follows since $f_B(z/r) = 0$ for $r < |z|$. Now, for any $\delta > 0$, we have that

$$\begin{aligned} f_Z(\delta) &= \int_{|\delta|}^\infty f_G(r)f_B(\delta/r)\frac{1}{r}dr \\ &= \int_{|-\delta|}^\infty f_G(r)f_B(-\delta/r)\frac{1}{r}dr = f_Z(-\delta), \end{aligned}$$

where the second step follows since f_B is symmetric around 0. Thus, the distribution is symmetric around 0. As we shall see in Section 5.2, the expected value of Z is 0. Thus, the distribution is symmetric around its mean. Unfortunately, the integral above does not have an easy analytical solution. We can, however, numerically evaluate it or through Monte Carlo simulations by repeatedly sampling the noise vector.

5.2 Moments of the Angular Component

Let K denote the random variable distributed as Eq. (5). We are interested in the moment $\mathbb{E}[K^j]$ of this distribution with $j \geq 0$. We parameterize this distribution by using K_n to denote the random variable K with a given value of $n \geq 2$.

Theorem 4. *Let $K \sim f_B$ as defined in Eq. (5). Let K_n denote K for a particular value of n. Let $\mu(j,n)$ denote the jth moment of K_n. Then, for all $n \geq 2$*

$$\mu(j,n) = \begin{cases} 1, & \text{if } j = 0, \\ 0, & \text{if } j \text{ is odd}, \\ \frac{(n-2)!!(j-1)!!}{(n-2+j)!!}, & \text{if } j \text{ is even} \end{cases} \tag{14}$$

In particular, $\mathbb{E}[K] = 0$ and $Var[K] = \frac{1}{n}$.

Proof. See Appendix A. □

5.3 Tail Bounds, Expectation and Variance

The following definition and the follow-up theorem are taken from [33, §2].

Definition 3 (Sub-Gaussian Random Variable)**.** A random variable X with mean $\mu = \mathbb{E}[X]$ is sub-Gaussian if there is a positive number σ such that

$$\mathbb{E}\left[e^{\lambda(X-\mu)}\right] \leq e^{\sigma^2\lambda^2/2}, \quad \text{for all } \lambda \in \mathbb{R}.$$

Here σ is called the sub-Gaussian parameter.

Theorem 5 (Sub-Gaussian Tail Bound). *A sub-Gaussian random variable X with mean $\mu = \mathbb{E}[X]$ and sub-Gaussian parameter σ satisfies*

$$\Pr\left[X-\mu \geq t\right] \leq e^{-\frac{t^2}{2\sigma^2}}, \text{ and } \Pr\left[X-\mu \leq -t\right] \leq e^{-\frac{t^2}{2\sigma^2}},$$

and combining the two

$$\Pr\left[|X-\mu| \geq t\right] \leq 2e^{-\frac{t^2}{2\sigma^2}}, \quad \text{for all } t \in \mathbb{R}.$$

We prove the following result to be used for the next theorem.

Lemma 1. *Let $K \sim f_B$. Then for any $\lambda \in \mathbb{R}$ we have*

$$\mathbb{E}\left[\sum_{j=0}^{\infty}\frac{(\lambda K)^j}{j!}\right] = \sum_{j=0}^{\infty}\frac{\lambda^j\mathbb{E}[K^j]}{j!}$$

Proof. Since $-1 \leq K \leq 1$, we have $|K| \leq 1$. Therefore $\mathbb{E}[|K|^j] \leq 1$. We have, for any $j \geq 0$:

$$\mathbb{E}\left[\frac{\lambda^j|K|^j}{j!}\right] = \frac{\lambda^j}{j!}\mathbb{E}[|K|^j] \leq \frac{\lambda^j}{j!}.$$

For any integer m, let

$$S_m = \mathbb{E}\left[\sum_{j=0}^{m}\frac{\lambda^j|K|^j}{j!}\right] = \sum_{j=0}^{m}\frac{\lambda^j}{j!}\mathbb{E}[|K|^j] \leq \sum_{j=0}^{m}\frac{\lambda^j}{j!},$$

where the last inequality follows from the result above. Furthermore, the sequence S_m is monotonically increasing if $\lambda \geq 0$ or monotonically decreasing if $\lambda < 0$, since $\mathbb{E}[|K|^j] \geq 0$ as $|K| \geq 0$. Thus, S_m is a monotone sequence. From the Taylor series expansion of the exponential function, we have for any $\lambda \in \mathbb{R}$

$$e^{|\lambda|} = \sum_{j=0}^{\infty}\frac{|\lambda|^j}{j!} \geq \sum_{j=0}^{m}\frac{|\lambda|^j}{j!} \geq \sum_{j=0}^{m}\frac{\lambda^j}{j!} \geq S_m.$$

Thus, S_m is bounded. From the monotone convergence theorem [1, §2.4], S_m converges. Therefore, the statement of the theorem follows as expectation is linear in this case [25, §2.1.1]. □

Theorem 6. *Let $K \sim f_B$ where f_B is as defined in Eq. 5. Then K is sub-Gaussian with parameter $\sigma = \frac{1}{\sqrt{n}}$.*

Proof. See Appendix A. □

Corollary 1. *Let $K \sim f_B$. Then for any $c \in \mathbb{R}$,*

$$\Pr\left[|K| \geq \frac{c}{\sqrt{n}}\right] \leq 2e^{-\frac{c^2}{2}}.$$

Next we prove generic lower and upper tail bounds for the gamma distributed random variable R.[12]

Theorem 7. *Let $R \sim f_G$. Then for any real number $c > 1$,*

$$\Pr\left[R \geq \frac{cn}{\epsilon}\right] \leq \left(\frac{c}{e^{c-1}}\right)^n, \text{ and } \Pr\left[R \leq \frac{n}{c\epsilon}\right] \leq \frac{1}{(ce^{(1-c)/c})^n}$$

Proof. See Appendix A. □

Combining the result from Corollary 1 and Theorem 7, we have the following bound on the overall noise distribution.

[12] These results are generalizations of the result for the upper tail bound with $c = 2$ in https://math.hawaii.edu/~grw/Classes/2013-2014/2014Spring/Math472_1/Solutions01.pdf.

Theorem 8. *Let $Z = RK$. Then, for all $c_1, c_2 \in \mathbb{R}$, where $c_2 > 1$, we have*

$$\Pr\left[|Z| \leq \frac{c_1 c_2 \sqrt{n}}{\epsilon}\right] \geq 2\left(1 - e^{\frac{-c_1^2}{2}}\right)\left(1 - \left(\frac{c_2}{e^{c_2-1}}\right)^n\right) - 1$$

Proof. We have

$$\begin{aligned}
\Pr\left[Z \leq \frac{c_1 c_2 \sqrt{n}}{\epsilon}\right] &= \Pr\left[RK \leq \frac{c_1 c_2 \sqrt{n}}{\epsilon}\right] \\
&= \Pr\left[RK \leq \frac{c_1}{\sqrt{n}} \frac{c_2 n}{\epsilon}\right] \\
&\geq \Pr\left[K \leq \frac{c_1}{\sqrt{n}}\right] \Pr\left[R \leq \frac{c_2 n}{\epsilon}\right] \\
&\geq \left(1 - e^{\frac{-c_1^2}{2}}\right)\left(1 - \left(\frac{c_2}{e^{c_2-1}}\right)^n\right),
\end{aligned}$$

where the last inequality follows from Corollary 1 and Theorem 7. From Section 5.1, Z is symmetric, and hence the above bound is also true for $\Pr[Z \geq -c_1 c_2 \sqrt{n}/\epsilon]$. Through Bonferroni's inequality

$$\begin{aligned}
&\Pr\left[|Z| \leq \frac{c_1 c_2 \sqrt{n}}{\epsilon}\right] \\
&= \Pr\left[Z \leq \frac{c_1 c_2 \sqrt{n}}{\epsilon} \text{ and } Z \geq -\frac{c_1 c_2 \sqrt{n}}{\epsilon}\right] \\
&\geq \Pr\left[Z \leq \frac{c_1 c_2 \sqrt{n}}{\epsilon}\right] + \Pr\left[Z \geq -\frac{c_1 c_2 \sqrt{n}}{\epsilon}\right] - 1 \\
&\geq 2\left(1 - e^{\frac{-c_1^2}{2}}\right)\left(1 - \left(\frac{c_2}{e^{c_2-1}}\right)^n\right) - 1,
\end{aligned}$$

as required. □

As an illustration of this inequality, with $n = 100$ and $\epsilon = 10$, more than 99 percent of the probability mass of Z lies within the interval $\pm 4.8\sqrt{n}/\epsilon = 4.8$ (with $c_1 \approx 3.46$ and $c + 2 \approx 1.39$). On the other hand, with $n = 10$, more than 99 percent of the probability mass of Z lies within the interval $\pm 8.74\sqrt{n}/\epsilon \approx 2.76$ (put for example $c_1 \approx 3.46$ and $c_2 \approx 2.53$).[13] Thus, the majority of the mass of Z is concentrated within $O(\sqrt{n}/\epsilon)$.

Finally we have the following theorem on the expected value and variance of Z, together with the convergence to the expected value if we sample a large number of instances of Z.

Theorem 9. *Let $Z = RK$. Then $\mathbb{E}[Z] = 0$ and $Var[Z] = \frac{n+1}{\epsilon^2}$.*

Proof. We know that the gamma distributed random variable R has $\mathbb{E}[R] = n/\epsilon$ and $\mathrm{Var}[R] = n/\epsilon^2$ Since R and K are independent, we have $\mathbb{E}[Z] = \mathbb{E}[RK] = \mathbb{E}[R]\mathbb{E}[K] = \frac{n}{\epsilon} \cdot 0 = 0$. Now using this result, and again because R and K are independent, we have:

$$\begin{aligned}
\mathrm{Var}[Z] &= \mathbb{E}[Z^2] - (\mathbb{E}[Z])^2 = \mathbb{E}[Z^2] \\
&= \mathbb{E}[R^2K^2] = \mathbb{E}[R^2]\mathbb{E}[K^2] \\
&= (\mathrm{Var}[R] + (\mathbb{E}[R])^2)\mathrm{Var}[K] \\
&= \left(\frac{n}{\epsilon^2} + \frac{n^2}{\epsilon^2}\right)\left(\frac{1}{n}\right) = \frac{n+1}{\epsilon^2}
\end{aligned}$$

□

[13] These values were obtained by fixing the bound from Theorem 8 at 0.99 and numerically finding the constants c_1 and c_2 by keeping the two terms in the product equal to each other.

Corollary 2. *Let $Z_1, Z_2, \ldots, Z_m$ be m independent samples of the random variable $Z = RK$. Let $\bar{Z} = \frac{1}{m}\sum_{i=1}^m Z_i$. Then for any $\delta > 0$*

$$\Pr\left[|\bar{Z}| > \delta\right] \leq \frac{n+1}{m\epsilon^2\delta^2} \tag{15}$$

Proof. The result is obtained by putting the expected value and variance of Z in Chebyshev's inequality [33, §2]. □

The above result shows that the higher the dimension and/or the lower the values of ϵ the slower will be the convergence of the average of Z to 0, the expectation. Thus, the behaviour of the noisy dot product distribution is as we would expect: it is symmetric with mean 0, and it converges to this expectation inversely proportional to ϵ, the privacy parameter.

6 The Loss Function and Consequences

Equipped with the results of the last section, we can now explore in depth how the nearest neighbor is obtained via the post-processing step. Recall the objective function from Eq. (3). Taking its square, we get

$$\begin{aligned}
\|\mathbf{w}^* - \mathbf{x}\|^2 &= \|\mathbf{w} + \boldsymbol{\eta} - \mathbf{x}\|^2 \\
&= \|\mathbf{w} - \mathbf{x}\|^2 + \|\boldsymbol{\eta}\|^2 + 2\langle \mathbf{w} - \mathbf{x}, \boldsymbol{\eta}\rangle \\
&= \|\mathbf{w} - \mathbf{x}\|^2 + r^2 + 2\langle \mathbf{w} - \mathbf{x}, \boldsymbol{\eta}\rangle \\
&= \|\mathbf{w} - \mathbf{x}\|^2 + r^2 + 2\langle \mathbf{w}, \boldsymbol{\eta}\rangle - 2\langle \mathbf{x}, \boldsymbol{\eta}\rangle
\end{aligned}$$

Now, the terms $\langle \mathbf{w}, \boldsymbol{\eta}\rangle$ and r^2 are the same for all $\mathbf{x} \in \mathcal{D}$, and therefore we can ignore them when finding the minimum. Let us define the *loss function* containing the remaining terms for all $\mathbf{x} \in \mathcal{D}$ as

$$L(\mathbf{x}) = \|\mathbf{w} - \mathbf{x}\|^2 - 2\langle \mathbf{x}, \boldsymbol{\eta}\rangle = \|\mathbf{w} - \mathbf{x}\|^2 + 2\|\mathbf{x}\| r \cos\theta_{\boldsymbol{\eta},\mathbf{x}} \tag{16}$$

Indeed, using partial derivatives we can see that the solution that minimizes the loss function L is $\mathbf{x} = \mathbf{w} + \boldsymbol{\eta} = \mathbf{w}^*$, but we know that with overwhelming probability this vector is not part of the embedding space. We have the following result.

Theorem 10. *Let L be defined as in Eq. (16). Then for any $\mathbf{x} \in \mathcal{D}$*

$$\mathbb{E}[L(\mathbf{x})] = \|\mathbf{w} - \mathbf{x}\|^2.$$

In particular, $\mathbb{E}[L(\mathbf{w})] = 0$, and $\mathbb{E}[L(\mathbf{x})] > 0$ iff $\mathbf{x} \neq \mathbf{w}$. Furthermore, for any $\mathbf{x}, \mathbf{y} \in \mathcal{D}$ with $\mathbf{x} \neq \mathbf{y}$ and $\|\mathbf{w} - \mathbf{x}\| < \|\mathbf{w} - \mathbf{y}\|$, we have

$$\Pr\left[L(\mathbf{x}) < L(\mathbf{y})\right] > 1/2.$$

Proof. From Eq. (16) and Theorem 9, we see that

$$\begin{aligned}
\mathbb{E}[L(\mathbf{x})] &= \mathbb{E}\left[\|\mathbf{w} - \mathbf{x}\|^2 + 2\|\mathbf{x}\| r \cos\theta_{\boldsymbol{\eta},\mathbf{x}}\right] \\
&= \|\mathbf{w} - \mathbf{x}\|^2 + 2\|\mathbf{x}\|\mathbb{E}[r\cos\theta_{\boldsymbol{\eta},\mathbf{x}}] \\
&= \|\mathbf{w} - \mathbf{x}\|^2,
\end{aligned}$$

from which it follows that the expectation is 0 only if $\mathbf{x} = \mathbf{w}$, and otherwise it is greater than 0. For the second part of the theorem, we see that

$$\begin{aligned}
L(\mathbf{y}) - L(\mathbf{x}) &= \|\mathbf{w} - \mathbf{y}\|^2 - 2\langle \mathbf{y}, \boldsymbol{\eta}\rangle - \|\mathbf{w} - \mathbf{x}\|^2 + 2\langle \mathbf{x}, \boldsymbol{\eta}\rangle \\
&= \|\mathbf{w} - \mathbf{y}\|^2 - \|\mathbf{w} - \mathbf{x}\|^2 + 2\langle \mathbf{x} - \mathbf{y}, \boldsymbol{\eta}\rangle \\
&= \|\mathbf{w} - \mathbf{y}\|^2 - \|\mathbf{w} - \mathbf{x}\|^2 + 2\|\mathbf{x} - \mathbf{y}\| r \cos\theta_{\boldsymbol{\eta},\mathbf{x}-\mathbf{y}}
\end{aligned}$$

Define the random variable S as

$$S = \|\mathbf{w}-\mathbf{y}\|^2 - \|\mathbf{w}-\mathbf{x}\|^2 + 2\|\mathbf{x}-\mathbf{y}\|Z,$$

which follows from the fact that $r\cos\theta_{\boldsymbol{\eta},\mathbf{x}-\mathbf{y}}$ is distributed as Z. Note that from the analysis above, we have $\mathbb{E}[S] = \|\mathbf{w}-\mathbf{y}\|^2 - \|\mathbf{w}-\mathbf{x}\|^2$. Now, since $\|\mathbf{w}-\mathbf{x}\| < \|\mathbf{w}-\mathbf{y}\|$, from Eq. (11), we have that

$$\begin{aligned}
&\Pr[L(\mathbf{x}) < L(\mathbf{y})] = \Pr[S > 0] \\
&= \Pr\left[Z > \frac{\|\mathbf{w}-\mathbf{x}\|^2 - \|\mathbf{w}-\mathbf{y}\|^2}{2\|\mathbf{x}-\mathbf{y}\|}\right] \qquad (17)\\
&> \Pr[Z \geq 0] \\
&= \frac{1}{\Gamma(n)\epsilon^{-n}B(\frac{n-1}{2},\frac{1}{2})}\int_{|0|}^{\infty} r^{n-1}e^{-\epsilon r} \\
&\times\left(\int_{-1}^{0/r}\left(1-k^2\right)^{\frac{n-1}{2}-1}dk\right)dr \\
&= \frac{1}{2}\frac{1}{\Gamma(n)\epsilon^{-n}}\int_0^{\infty} r^{n-1}e^{-\epsilon r}dr = \frac{1}{2},
\end{aligned}$$

as required. The second-to-last equality follows because the distribution f_{B} is symmetric around 0. □

This theorem shows that, ignoring the actual difference in probabilities, there is nothing unusual in the nearest neighbor search for any value of ϵ and n: the original word is expected to be output more often, followed by its nearest neighbor, and so on. However, the issue is with the exact values of these probabilities, as we shall see next.

Eq. (17) gives us another insight. Suppose $\mathbf{x}_1$ and $\mathbf{x}_2$ are the first two neighbors of $\mathbf{w}$. Then first we see that for $i = 1, 2$

$$\begin{aligned}
\Pr[L(\mathbf{w}) < L(\mathbf{x}_i)] &= \Pr\left[Z > \frac{\|\mathbf{w}-\mathbf{w}\|^2 - \|\mathbf{w}-\mathbf{x}_i\|^2}{2\|\mathbf{w}-\mathbf{x}_i\|}\right] \\
&= \Pr\left[Z > -\frac{\|\mathbf{w}-\mathbf{x}_i\|}{2}\right] \\
&= \Pr\left[Z \leq \frac{\|\mathbf{w}-\mathbf{x}_i\|}{2}\right], \qquad (18)
\end{aligned}$$

where the last equality follows from the fact that Z is symmetric around 0 (see Section 5.1). Compare this to the following where we also use symmetry of the distribution:

$$\begin{aligned}
\Pr[L(\mathbf{x}_1) < L(\mathbf{x}_2)] &= \Pr\left[Z > \frac{\|\mathbf{w}-\mathbf{x}_1\|^2 - \|\mathbf{w}-\mathbf{x}_2\|^2}{2\|\mathbf{x}_1-\mathbf{x}_2\|}\right] \\
&= \Pr\left[Z \leq \frac{\|\mathbf{w}-\mathbf{x}_2\|^2 - \|\mathbf{w}-\mathbf{x}_1\|^2}{2\|\mathbf{x}_1-\mathbf{x}_2\|}\right] \qquad (19)
\end{aligned}$$

Eqs (18) and (19) are the same equations we derived in Theorem 1 and Eq. (22) in Theorem 2, but this time with the loss function formulation.

6.1 Probabilistic Interpretation

Eqs (18) and (19) can be interpreted as upper bounds on the probability that a particular word will be chosen as the nearest neighbor. To see this let N denote the size of the vocabulary. Given a word $\mathbf{w}$, we index the vocabulary, as follows: $\mathbf{x}_0 = \mathbf{w}$ and $\mathbf{x}_i$ denotes the ith nearest neighbor of $\mathbf{w}$ where ties are broken arbitrarily. Let $C_{i,j}$ be the event that word embedding $\mathbf{x}_i$ is closer to the noisy embedding $\mathbf{w}^*$ than word embedding $\mathbf{x}_j$. Furthermore, let C_i be the event that word embedding $\mathbf{x}_i$ is the nearest neighbor of $\mathbf{w}^*$. From Eq. (18), we have the following condition:

$$\begin{aligned}
\Pr[C_i] &= 1 - \Pr[\bar{C}_i] \\
&= 1 - \Pr\left[\bigcup_{j=0, j\neq i}^{N} \bar{C}_{i,j}\right] \\
&\leq 1 - \Pr[\bar{C}_{i,0}] \\
&= 1 - \Pr[L(\mathbf{w}) < L(\mathbf{x}_i)] \\
&= 1 - \Pr\left[Z \leq \frac{\|\mathbf{w}-\mathbf{x}_i\|}{2}\right] \\
&\leq 1 - \Pr\left[Z \leq \frac{\|\mathbf{w}-\mathbf{x}_1\|}{2}\right], \qquad (20)
\end{aligned}$$

where we have used the fact that $\bar{C}_{i,0} \subseteq \cup_{j=0,j\neq i}^{N}\bar{C}_{i,j}$, and therefore the probability of former is less than or equal to the probability of the latter. Likewise, we have

$$\begin{aligned}
\Pr[C_i] &\leq 1 - \Pr[\bar{C}_{i,j}] = \Pr[C_{i,j}] = \Pr[L(\mathbf{x}_i) < L(\mathbf{x}_j)] \\
&= \Pr\left[Z \leq \frac{\|\mathbf{w}-\mathbf{x}_j\|^2 - \|\mathbf{w}-\mathbf{x}_i\|^2}{2\|\mathbf{x}_i-\mathbf{x}_j\|}\right], \qquad (21)
\end{aligned}$$

for $i \neq 0$. The probabilities in Eq. (20) and (21) are upper bounds on the necessary condition for a word $\mathbf{x}_i$ to be chosen as the nearest neighbor to the noisy embedding.

6.2 Results on Word Embeddings

To show how likely the original word $\mathbf{w}$ is chosen over its nearest neighbor $\mathbf{x}_1$, and contrast it with how likely the nearest neighbor $\mathbf{x}_1$ is chosen over the second nearest neighbor $\mathbf{x}_2$, we take the right hand quantities bounding the probability mass of the random variable Z in Eqs (18) and (19). To ease notation, we denote these quantities by $z_{\mathbf{w},\mathbf{x}_1}$ and $z_{\mathbf{x}_1,\mathbf{x}_2}$, respectively. For each vocabulary, i.e., embedding model, we take the average of these quantities by taking 5,000 random words. The results are shown in Table 2. The quantity $z_{\mathbf{x}_1,\mathbf{x}_{101}}$ is the right hand quantity in Eq (19) for the nearest neighbor and 101th neighbor of $\mathbf{w}$.

Vocabulary	n	$z_{\mathbf{w},\mathbf{x}_1}$	$z_{\mathbf{x}_1,\mathbf{x}_2}$	$z_{\mathbf{x}_1,\mathbf{x}_{101}}$
GloVe-Twitter	25	1.078	0.132	0.692
	50	1.571	0.169	0.747
	100	2.166	0.197	0.764
	200	2.708	0.220	0.683
GloVe-Wiki	50	1.369	0.172	0.757
	100	1.403	0.152	0.616
	200	2.15	0.243	0.823
	300	2.896	0.282	0.999
Word2Vec	300	0.763	0.058	0.253
fastText	300	1.493	0.239	0.912

Table 2: The right hand side terms $z_{\mathbf{w},\mathbf{x}_1}$, $z_{\mathbf{x}_1,\mathbf{x}_2}$ and $z_{\mathbf{x}_1,\mathbf{x}_{101}}$ as they appear in Eqs. (18) **and** (19) **for different vocabularies.**

The first thing to notice is that for any given dimension $z_{\mathbf{w},\mathbf{x}_1}$ is considerably greater than $z_{\mathbf{x}_1,\mathbf{x}_2}$. This is because the distance of any word to its first neighbor in high dimensions is considerably higher than the (relative) difference in distances to its first two neighbors.

The other observation is more of an illusion. The value of $z_{\mathbf{w},\mathbf{x}_1}$ is increasing as we increase the dimensions, by looking at the GloVe vocabularies. Thus it may appear that the problem is exacerbated as we increase the dimensions from say 25 to 200. However, this is not exactly correct. Recall from Theorem 8 that the bulk of the mass of Z is within $O(\sqrt{n}/\epsilon)$. Thus, under two different dimensions n_1 and n_2, we expect $F_Z(z_1) \approx F_Z(z_2)$ if $\frac{z_1}{z_2} \approx \sqrt{\frac{n_1}{n_2}}$. By looking at the table, we see that this ratio across the two types of GloVe is more or less obeyed by the quantities $z_{\mathbf{w},\mathbf{x}_1}$ as we cycle through the dimensions, and slightly less so by the quantities $z_{\mathbf{x}_1,\mathbf{x}_2}$. Thus, we might see the problem as being slightly more worsened in higher dimensions but not by much.

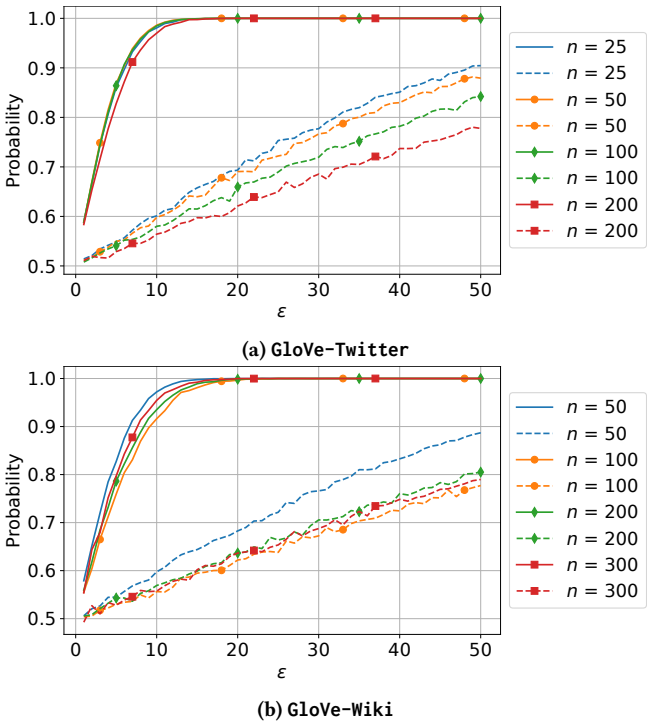
This is more obvious as we plot the probabilities $F_Z(z_{\mathbf{w},\mathbf{x}_1})$ and $F_Z(z_{\mathbf{x}_1,\mathbf{x}_2})$ in Figure 6. Although the CDF can be obtained by numerically integrating the integral in Eq. (11), however, we found the integration to be slow using for example SymPy.[14] We therefore obtained the CDF of Z using Monte Carlo simulations by randomly sampling the noise vector 10,000 times and then finding the proportion of times it falls below $z_{\mathbf{w},\mathbf{x}_1}$ or $z_{\mathbf{x}_1,\mathbf{x}_2}$. This can be done by sampling R via the Gamma distribution f_{G}, and then sampling K by checking the cosine of the angle of the noise vector against a fixed vector, e.g., $\hat{\mathbf{e}}_1 = (1, 0, 0, \ldots, 0)$.

As is evident from the figure, $F_Z(z_{\mathbf{w},\mathbf{x}_1})$ dominates when $\epsilon > 1$. This means, that the probability of choosing the original word increases, and from Eq. (20), the probability that the nearest neighbor would be output decreases. Furthermore, around $\epsilon = 10$, where the original word is still not entirely certain to be output, the probability $F_Z(z_{\mathbf{x}_1,\mathbf{x}_{101}})$ is not overwhelming enough (not plotted) to ensure that the nearest neighbor would be output more than the 101th neighbor. Hence far away neighbors are still being output at these ϵ values, until the original word completely dominates as we further increase ϵ.

7 Proposed Fix

In light of the discussion in the previous section, if we can somehow make the distance of the original word to its nearest neighbor similar to distances between its consecutive neighbors, the issue could be resolved. To do so, one may be tempted to employ the following mechanism. Let $\mathbf{w}$ be the original word. We find all nearest neighbors of $\mathbf{w}$ according to the Euclidean distance, and assign the function $d_{\mathrm{NN}}(\mathbf{w}, \mathbf{x}) = i$ if $\mathbf{x}$ is the ith nearest neighbor of $\mathbf{w}$. We have $d_{\mathrm{NN}}(\mathbf{w}, \mathbf{w}) = 0$. We can then sample a word $\mathbf{x}$ proportional to $\exp(-\epsilon d_{\mathrm{NN}}(\mathbf{w}, \mathbf{x}))$. However, d_{NN} is not a metric as it does not satisfy the properties of symmetry and triangle inequality as illustrated in Figure 7. In the figure we consider $\mathbb{R}^2$. We have $d_{\mathrm{NN}}(\mathbf{x}_1, \mathbf{x}_2) = 2$, however, $d_{\mathrm{NN}}(\mathbf{x}_2, \mathbf{x}_1) = 4$, violating symmetry. Furthermore, $d_{\mathrm{NN}}(\mathbf{x}_2, \mathbf{w}) = 1$ and $d_{\mathrm{NN}}(\mathbf{w}, \mathbf{x}_1) = 2$, implying that $d_{\mathrm{NN}}(\mathbf{x}_2, \mathbf{x}_1) > d_{\mathrm{NN}}(\mathbf{x}_2, \mathbf{w}) + d_{\mathrm{NN}}(\mathbf{w}, \mathbf{x}_1)$, violating the triangle inequality.[15] Thus, the resulting mechanism cannot be $d_{\mathcal{X}}$-private.

Theorem 10 says that on average the nearest neighbor of the noisy embedding $\mathbf{w}^*$ is the original word $\mathbf{w}$, followed by $\mathbf{w}$'s nearest neighbor, then its second nearest neighbor, and so on. Due to large probability differences, as shown in Figure 6, the original word is



(a) GloVe-Twitter

(b) GloVe-Wiki

Figure 6: The probabilities $F_Z(z_{\mathbf{w},\mathbf{x}_1})$ (solid) and $F_Z(z_{\mathbf{x}_1,\mathbf{x}_2})$ (dashed) where $z_{\mathbf{w},\mathbf{x}_1}$ and $z_{\mathbf{x}_1,\mathbf{x}_2}$ are as given in Table 2 for the two GloVe vocabularies.

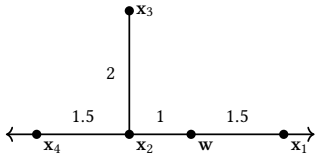


Figure 7: Example used to illustrate the fact that d_{NN} is not a metric. The numbers show the distance of the corresponding segments.

chosen overwhelmingly. To mitigate this, our idea is to *not* select the nearest neighbor of $\mathbf{w}^*$ every time, and instead occasionally sample other neighbors. This could be used as a post-processing step *after* we have found the nearest word $\mathbf{x}^*$ to the noisy embedding $\mathbf{w}^*$ through Eq. (3). In other words, we sort the nearest neighbors of $\mathbf{x}^*$ and output a neighbor proportional to $\exp(-\epsilon d_{\mathrm{NN}}(\mathbf{x}^*, \mathbf{x}))$. More specifically any word $\mathbf{x} \in \mathcal{D}$ is output with probability:

$$\frac{\exp(-c\epsilon d_{\mathrm{NN}}(\mathbf{x}^*, \mathbf{x}))}{\sum_{\mathbf{x}\in\mathcal{D}} \exp(-c\epsilon d_{\mathrm{NN}}(\mathbf{x}^*, \mathbf{x}))},$$

where c is a constant to control how many neighbors are likely to be selected. Note that ϵ is not used here for privacy protection but rather to ensure that the mechanism behaves as expected, e.g., only the original word output with very high values of ϵ. A higher value such as $c > 1$ means that the mechanism will output the first few neighbors with high probability, and a lower value such

[14] See https://www.sympy.org/.

[15] The triangle inequality seems less of an issue, as we might be okay with $d_{\mathrm{NN}}(\mathbf{x}_2, \mathbf{x}_1)$ being bounded by a function of $d_{\mathrm{NN}}(\mathbf{x}_2, \mathbf{w})$ and $d_{\mathrm{NN}}(\mathbf{w}, \mathbf{x}_1)$. See [11, 15].

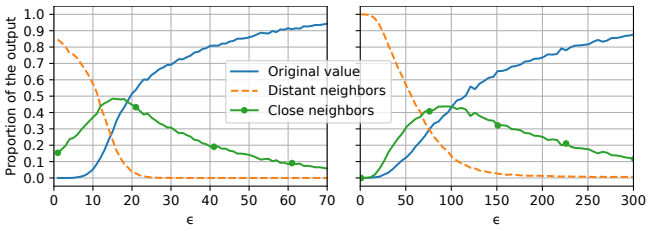


Figure 8: The proportion of times the original word, its close neighbors and distant neighbors are output by the fixed mechanism on the `GloVe-Wiki` (300 dimensions) with $c = 0.04$ (left) and `Word2Vec` with $c = 0.007$ (right). Close neighbors are the first 100 nearest neighbors, and all other words are distant neighbors.

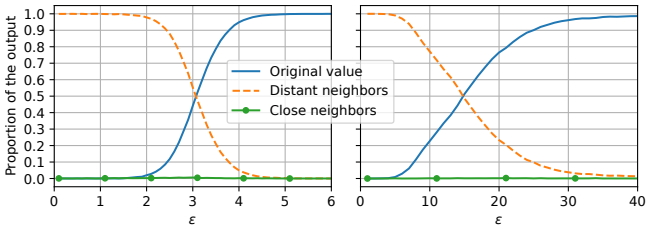


Figure 9: The proportion of times the original word, its close neighbors and distant neighbors are output by the exponential mechanism of [35] on the `GloVe-Wiki` (300 dimensions) (left) and `Word2Vec` (right). Close neighbors are the first 100 nearest neighbors, and all other words are distant neighbors.

as $c = 0.01$ means that more neighbors will likely to be output, of course, with probability exponentially decreasing as we move away from the original word. This is the same as the temperature variable in the softmax function. Note that we cannot use the original word $\mathbf{w}$ in the above expression in place of $\mathbf{x}^*$, as the resulting fix will no longer be a post-processing step (it uses the knowledge of the original word $\mathbf{w}$).

Figure 8 shows the result of applying this fix to the `GloVe-Wiki` and `Word2Vec` vocabularies. The value of c is chosen by trying different values and choosing the one that gives the best result in terms of the proportion of the times the original word, its 100 nearest neighbors and distant neighbors are output by the mechanism. This value can be pre-computed for each vocabulary. Compare this to Figure 2, where between $\epsilon = 10$ and 20 for `GloVe-Wiki` and between $\epsilon = 50$ and 60 for `Word2Vec`, either the original word or its distant neighbors are output by the original mechanism. Our mechanism in comparison ensures a more equitable proportion for the original word, its close and distant neighbors, in line with the one-dimensional case shown in Figure 1. A drawback of this mechanism, of course, is that the value of c needs to be empirically determined for each vocabulary.

Comparison with the Exponential Mechanism. Our fix looks similar to the exponential mechanism proposed by Yue et al. [35, Algorithm 1]. They pre-compute a matrix containing the probability that each token of the vocabulary is replaced by any other token. The probabilities are computed as an exponential of the Euclidean distance. More precisely, the probability that a token $\mathbf{w}$ is replaced by a token $\mathbf{x}$ is proportional to $\exp(-\frac{1}{2}\epsilon \cdot \|\mathbf{w} - \mathbf{x}\|)$. The authors argue that the mechanism has better performance by avoiding the nearest neighbor search in an n-dimensional space during sanitization. However, in Figure 9 we show that it still suffers from the problem exposed by us, namely that close neighbors are almost never sampled. The main difference between this mechanism and our fix is that we use the rank instead of a distance metric to compute probabilities. We leave it as an open problem to construct a distance metric that "flattens out" the distance between an embedding and its nearest neighbor, and distances between its consecutive neighbors. That is, it should behave similarly to the nearest neighbor function d_{NN}, while still satisfying the properties of a distance metric.

8 Related Work

In addition to the multidimensional Laplace mechanism for word-level $d_\mathcal{X}$-privacy from [15, 16, 22, 31, 35] detailed in this paper, a few other word-level $d_\mathcal{X}$-privacy methods have also been proposed in the literature. In [17], the authors propose a $d_\mathcal{X}$-private mechanism using a distance metric in the hyperbolic space, which according to the authors, better preserves hierarchical relationships between words. For instance, the hierarchical relationship of the city of London to England. Xu et al [34] propose the Mahalanobis mechanism, which instead of the spherical noise via the multidimensional Laplace mechanism, samples elliptical noise. The authors note that this mechanism provides better privacy for isolated points (see Section 2) since the elliptical nature of the noise results in sampling points other than the original word with higher probability. Another mechanism for word-level $d_\mathcal{X}$-privacy is the truncated exponential mechanism (TEM) [9], proposed to remove the drawback of the multidimensional Laplace mechanism which adds the same amount of noise regardless of whether a word is in a dense or sparse region of space. Applying their mechanism to a data domain would add more noise to points in low density areas. Although the mechanisms from [9] and [34] may solve the issue with isolated embeddings, they are unlikely to resolve the issue raised by us. As we show in Section 6, the problem stems from the large difference in probabilities of sampling the same word versus its nearest neighbors in the nearest neighbor search. The Mahalanobis mechanism [34] also contains nearest neighbor search as a post-processing step, and the TEM from [9] uses a distance metric to sample a word. Thus the problem is likely to persist in these mechanisms.

Instead of word-level differential privacy, several works focus on providing differential privacy at the sentence level [19, 23]. More specifically, these works calculate the conditional probabilities of the next word, given a sequence of previous words. We note that both the mechanisms in [23] and [19] are for ordinary differential privacy and not $d_\mathcal{X}$-privacy. Finally, differential privacy mechanisms for more advanced downstream tasks such as paraphrasing have also been proposed [32], but again using ordinary differential privacy. The exponential mechanism in [23] and [32] is similar to our fix for multidimensional Laplace mechanism for $d_\mathcal{X}$-privacy with the difference being that our fix is a post-processing step.

There have been some prior works showing the weaknesses of word-level $d_{\mathcal{X}}$-privacy. The aforementioned work in [23] criticizes several aspects of word-level $d_{\mathcal{X}}$-privacy, including introduction of grammatical errors and only making changes to individual words, rather than changing lengths of sentences. The work in [28] demonstrates attacks on word and sentence-level differential privacy on text data, by reconstructing santized prompts where the original prompts are taken from the training data of the language model. The issues and attacks mentioned in these two works are tangential to the problem addressed in our paper.

The fact that $d_{\mathcal{X}}$-privacy, in particular, the multidimensional Laplace mechanism does not provide adequate protection for isolated points in the input domain has also been highlighted in the case of location privacy [6, 12]. We discuss here why we think these solutions are not readily applicable to fix our issue in a high-dimensional space. The works in [6, 12] use the concept of *elastic distinguishability* introduced in [12]. The idea is that in addition to the usual requirement in $d_{\mathcal{X}}$-privacy that nearby points should have higher probability of being sampled than far away points, the mechanism should also sample a point proportional to its *probability mass* or density. For instance, an isolated location such as an island has a lower probability mass than a location in a dense urban area [12]. The mechanisms in [6, 12] are essentially weighted versions of the exponential mechanism, where the probability of sampling a point is also weighted by its probability mass. Location has a natural candidate for assigning a probability mass to points in space: the number of people at a given location.

First, we emphasize that their observation is related to the (location) data distribution, with the assumption that the bulk of data is concentrated with only a few isolated points. This is unlike our finding where we show that in essence every embedding in high dimensions is isolated. Second, in the context of location data, even isolated points have a non-zero probability of being output by the mechanism, as they are valid locations. Consequently, there is no need for a nearest neighbor search. Indeed the mechanisms in [6, 12] do not use it. This is unlike the word embedding space, where a random vector is overwhelmingly not an actual word embedding, and hence the need to perform the nearest neighbor search which, as we saw, is what causes the issue identified in this paper. Third, if we were to apply a similar mechanism to our setting, we need to determine how to assign probability mass to regions in the embedding space. One way to do this is to divide the embedding space into equal-volume hypercubes, count the number of embeddings falling in each hypercube and assign the fractional count as the probability mass. This is analogous to how the two-dimensional location map is divided into a grid in [6, 12]. However, dividing the embedding space in hypercubes is not trivial. On the one hand, dividing each of the n dimensions into m shares results in m^n hypercubes which is unfit for storage (e.g., $n = 300$ in some of our vocabularies). On the other hand, choosing a subset of the n dimensions to be divided requires an analysis of which dimension to divide or not. Thus it is unclear whether the notion of elastic distinguishability offers a solution let alone a feasible one.

The work in [4] also looks at the distinguishability of isolated points in the *shuffle model* of differential privacy. In the shuffle model, an intermediate shuffler permutes (local) differentially private inputs of end users before submitting them to the central aggregator. This amplifies privacy compared to the local model as the server cannot tie an output to a particular user. In the $d_{\mathcal{X}}$-privacy equivalent version of the shuffle model, the server can still identify isolated inputs (even when perturbed) as they stand out from the rest. To remove this drawback, one of the techniques proposed in [4] is to encode an input in unary and then apply differentially private noise (randomized response) to each bit before sending it to the shuffler. It is not clear how these techniques can be applied to the case of text embeddings. We would need to convert embeddings into bit vectors and then combine multiple embeddings together. Moreover, once again, the isolated point issue is endemic in high dimensional embeddings as opposed to being just a few bad apples.

Lastly, researchers have identified issues with Euclidean distance in high dimensions. Beyer et al [5] show that the distance of the nearest neighbor approaches that of the farthest neighbor as the number of dimensions increases, thus making nearest neighbor search using the Euclidean distance meaningless, a result known as *concentration of distances* [13]. The authors in [13], expand on this to show that the concentration of distances is not applicable as long as the number of 'relevant' dimensions are on par with the actual data dimensions, and hence in these cases nearest neighbor search is still meaningful. This seems to be the case with word embedding models as is backed up by the empirical results in this paper and the fact that word embedding models are trained to ensure that Euclidean distance can be used to find similar words even with high dimensions [24, 29]. For more information on this topic, we refer the reader to the survey [37].

9 Conclusion

The multidimensional Laplace mechanism for $d_{\mathcal{X}}$-privacy is widely used as a method to provide privacy for sensitive text data due to its ease of implementation. We have shown that the method behaves unexpectedly under common word embedding models. More specifically, it almost never outputs semantically related words, as it is expected to do for utility. We have extensively analyzed the noise generated through this mechanism and ruled out any issues with the original mechanism. Instead, we have identified the post-processing step of the nearest neighbor search as the culprit, causing each word embedding to behave as an outlier in high dimensions. We have provided an easy-to-use fix which makes the mechanism behave more expectedly. The reader is invited to investigate alternative remedies.

Acknowledgments

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

[1] Stephen Abbott et al. 2001. *Understanding analysis*. Vol. 2. Springer.
[2] Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. 2001. On the surprising behavior of distance metrics in high dimensional space. In *Database theory—ICDT 2001: 8th international conference London, UK, January 4–6, 2001 proceedings 8*. Springer, 420–434.
[3] Miguel E Andrés, Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. 2013. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC conference on Computer &*

communications security. 901–914.
[4] Andreas Athanasiou, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. 2025. Enhancing metric privacy with a shuffler. In *PETS 2025-25th Privacy Enhancing Technologies Symposium*.
[5] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. 1999. When is "nearest neighbor" meaningful?. In *Database Theory—ICDT'99: 7th International Conference Jerusalem, Israel, January 10–12, 1999 Proceedings 7*. Springer, 217–235.
[6] Sayan Biswas and Catuscia Palamidessi. 2023. PRIVIC: A privacy-preserving method for incremental collection of location data. *Proceedings on Privacy Enhancing Technologies* 2024, 1 (2023), 582–596.
[7] Avrim Blum, John Hopcroft, and Ravindran Kannan. 2020. *Foundations of data science*. Cambridge University Press.
[8] Robin Carpentier, Benjamin Zi Hao Zhao, Hassan Jameel Asghar, and Dali Kaafar. 2024. Preempting Text Sanitization Utility in Resource-Constrained Privacy-Preserving LLM Interactions. *arXiv preprint arXiv:2411.11521* (2024).
[9] Ricardo Silva Carvalho, Theodore Vasiloudis, Oluwaseyi Feyisetan, and Ke Wang. 2023. Tem: High utility metric differential privacy on text. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*. SIAM, 883–890.
[10] George Casella and Roger Berger. 2024. *Statistical inference*. CRC Press.
[11] Konstantinos Chatzikokolakis, Miguel E Andrés, Nicolás Emilio Bordenabe, and Catuscia Palamidessi. 2013. Broadening the scope of differential privacy using metrics. In *Privacy Enhancing Technologies: 13th International Symposium, PETS 2013, Bloomington, IN, USA, July 10-12, 2013. Proceedings 13*. Springer, 82–102.
[12] Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Marco Stronati. 2015. Constructing elastic distinguishability metrics for location privacy. *Proceedings on Privacy Enhancing Technologies* (2015).
[13] Robert J Durrant and Ata Kabán. 2009. When is 'nearest neighbour'meaningful: A converse theorem and implications. *Journal of Complexity* 25, 4 (2009), 385–397.
[14] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*. Springer, 265–284.
[15] Natasha Fernandes, Mark Dras, and Annabelle McIver. 2019. Generalised differential privacy for text document processing. In *Principles of Security and Trust: 8th International Conference, POST 2019, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2019, Prague, Czech Republic, April 6–11, 2019, Proceedings 8*. Springer International Publishing, 123–148.
[16] Oluwaseyi Feyisetan, Borja Balle, Thomas Drake, and Tom Diethe. 2020. Privacy- and utility-preserving textual analysis via calibrated multivariate perturbations. In *Proceedings of the 13th international conference on web search and data mining*. 178–186.
[17] Oluwaseyi Feyisetan, Tom Diethe, and Thomas Drake. 2019. Leveraging hierarchical representations for preserving privacy and utility in text. In *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 210–219.
[18] Russell L Herman. 2015. Introduction to partial differential equations. *North Carolina, NC, USA: RL Herman* (2015).
[19] Timour Igamberdiev and Ivan Habernal. 2023. DP-BART for Privatized Text Rewriting under Local Differential Privacy. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.
[20] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, Mirella Lapata, Phil Blunsom, and Alexander Koller (Eds.). Association for Computational Linguistics, Valencia, Spain, 427–431. https://aclanthology.org/E17-2068
[21] Parameswaran Kamalaruban, Victor Perrier, Hassan Jameel Asghar, and Mohamed Ali Kaafar. 2020. Not all attributes are created equal: dx-private mechanisms for linear queries. *Proceedings on Privacy Enhancing Technologies* (2020).
[22] Yansong Li, Zhixing Tan, and Yang Liu. 2023. Privacy-preserving prompt tuning for large language model services. *arXiv preprint arXiv:2305.06212* (2023).
[23] Justus Mattern, Benjamin Weggenmann, and Florian Kerschbaum. 2022. The Limits of Word Level Differential Privacy. In *Findings of the Association for Computational Linguistics: NAACL 2022*, Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (Eds.). Association for Computational Linguistics, Seattle, United States, 867–881. https://doi.org/10.18653/v1/2022.findings-naacl.65
[24] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*. http://arxiv.org/abs/1301.3781
[25] Michael Mitzenmacher and Eli Upfal. 2017. *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge university press.
[26] Alexander McFarlane Mood. 1950. Introduction to the Theory of Statistics. (1950).
[27] Mícheál O'Searcoid. 2006. *Metric spaces*. Springer Science & Business Media.
[28] Shuchao Pang, Zhigang Lu, Haichen Wang, Peng Fu, Yongbin Zhou, Minhui Xue, and Bo Li. 2024. Reconstruction of Differentially Private Text Sanitization via Large Language Models. *arXiv preprint arXiv:2410.12443* (2024).
[29] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.
[30] William H Press, William T Vetterling, Saul A Teukolsky, and Brian P Flannery. 1988. *Numerical recipes*. Cambridge University Press, London, England.
[31] Chen Qu, Weize Kong, Liu Yang, Mingyang Zhang, Michael Bendersky, and Marc Najork. 2021. Natural language understanding with privacy-preserving bert. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1488–1497.
[32] Saiteja Utpala, Sara Hooker, and Pin-Yu Chen. 2023. Locally Differentially Private Document Generation Using Zero Shot Prompting. In *Conference on Empirical Methods in Natural Language Processing*.
[33] Martin J Wainwright. 2019. *High-dimensional statistics: A non-asymptotic viewpoint*. Vol. 48. Cambridge university press.
[34] Zekun Xu, Abhinav Aggarwal, Oluwaseyi Feyisetan, and Nathanael Teissier. 2020. A Differentially Private Text Perturbation Method Using Regularized Mahalanobis Metric. In *Proceedings of the Second Workshop on Privacy in NLP*. 7–17.
[35] Xiang Yue, Minxin Du, Tianhao Wang, Yaliang Li, Huan Sun, and Sherman S. M. Chow. 2021. Differential Privacy for Text Analytics via Natural Text Sanitization. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online, 3853–3866. https://doi.org/10.18653/v1/2021.findings-acl.337
[36] Aston Zhang, Zachary C Lipton, Mu Li, and Alexander J Smola. 2023. *Dive into deep learning*. Cambridge University Press.
[37] Arthur Zimek, Erich Schubert, and Hans-Peter Kriegel. 2012. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal* 5, 5 (2012), 363–387.

A Proofs

Proof of Theorem 2.

Proof.

$$
\begin{aligned}
&\|\mathbf{w}^* - \mathbf{y}\|^2 > \|\mathbf{w}^* - \mathbf{x}\|^2 \\
&\implies \|\mathbf{w} - \mathbf{y} + \boldsymbol{\eta}\|^2 > \|\mathbf{w} - \mathbf{x} + \boldsymbol{\eta}\|^2 \\
&\implies \|\mathbf{w} - \mathbf{y}\|^2 + \|\boldsymbol{\eta}\|^2 + 2\langle \mathbf{w} - \mathbf{y}, \boldsymbol{\eta}\rangle \\
&> \|\mathbf{w} - \mathbf{x}\|^2 + \|\boldsymbol{\eta}\|^2 + 2\langle \mathbf{w} - \mathbf{x}, \boldsymbol{\eta}\rangle \\
&\implies \frac{1}{2}(\|\mathbf{w} - \mathbf{y}\|^2 - \|\mathbf{w} - \mathbf{x}\|^2) > \langle \mathbf{w} - \mathbf{x}, \boldsymbol{\eta}\rangle - \langle \mathbf{w} - \mathbf{y}, \boldsymbol{\eta}\rangle \\
&= \langle \mathbf{w}, \boldsymbol{\eta}\rangle - \langle \mathbf{x}, \boldsymbol{\eta}\rangle - \langle \mathbf{w}, \boldsymbol{\eta}\rangle + \langle \mathbf{y}, \boldsymbol{\eta}\rangle \\
&= \langle \mathbf{y} - \mathbf{x}, \boldsymbol{\eta}\rangle \\
&= r\|\mathbf{y} - \mathbf{x}\| \cos\theta_{\boldsymbol{\eta}, \mathbf{y}-\mathbf{x}} \qquad (22) \\
&\implies \frac{1}{2}(\|\mathbf{w} - \mathbf{y} + \mathbf{x} - \mathbf{x}\|^2 - \|\mathbf{w} - \mathbf{x}\|^2) > r\|\mathbf{y} - \mathbf{x}\| \cos\theta_{\boldsymbol{\eta}, \mathbf{y}-\mathbf{x}} \\
&\implies \frac{1}{2}(\|\mathbf{w} - \mathbf{x}\|^2 + \|\mathbf{y} - \mathbf{x}\|^2 - 2\langle \mathbf{w} - \mathbf{x}, \mathbf{y} - \mathbf{x})\rangle - \|\mathbf{w} - \mathbf{x}\|^2) \\
&> r\|\mathbf{y} - \mathbf{x}\| \cos\theta_{\boldsymbol{\eta}, \mathbf{y}-\mathbf{x}} \\
&\implies \frac{1}{2}(\|\mathbf{y} - \mathbf{x}\|^2 - 2\langle \mathbf{w} - \mathbf{x}, \mathbf{y} - \mathbf{x})\rangle) > r\|\mathbf{y} - \mathbf{x}\| \cos\theta_{\boldsymbol{\eta}, \mathbf{y}-\mathbf{x}} \\
&\implies \frac{1}{2}(\|\mathbf{y} - \mathbf{x}\|^2 - 2\|\mathbf{w} - \mathbf{x}\|\|\mathbf{y} - \mathbf{x})\| \cos\theta_{\mathbf{w}-\mathbf{x}, \mathbf{y}-\mathbf{x}}) \\
&> r\|\mathbf{y} - \mathbf{x}\| \cos\theta_{\boldsymbol{\eta}, \mathbf{y}-\mathbf{x}} \\
&\implies \frac{1}{2}\|\mathbf{y} - \mathbf{x}\| > \|\mathbf{w} - \mathbf{x}\| \cos\theta_{\mathbf{w}-\mathbf{x}, \mathbf{y}-\mathbf{x}} + r \cos\theta_{\boldsymbol{\eta}, \mathbf{y}-\mathbf{x}}
\end{aligned}
$$

as required. □

Proof of Theorem 4.

Proof. Let $B_n = B((n-1)/2, 1/2)$. For any $n \geq 4$, we note that:

$$
\frac{B_{n-2}}{B_n} = \frac{\Gamma(\frac{n-3}{2})\Gamma(\frac{1}{2})}{\Gamma(\frac{n-3}{2} + \frac{1}{2})} \frac{\Gamma(\frac{n-1}{2} + \frac{1}{2})}{\Gamma(\frac{n-1}{2})\Gamma(\frac{1}{2})}
$$

$$
\begin{aligned}
&= \frac{\Gamma(\frac{n-3}{2})}{\Gamma(\frac{n}{2}-1)} \frac{\Gamma(\frac{n}{2})}{\Gamma(\frac{n-1}{2})} \\
&= \frac{\Gamma(\frac{n-3}{2})}{\Gamma(\frac{n}{2}-1)} \frac{\Gamma(\frac{n}{2}-1+1)}{\Gamma(\frac{n-1}{2}-1+1)} \\
&= \frac{\Gamma(\frac{n-3}{2})}{\Gamma(\frac{n}{2}-1)} \frac{(\frac{n}{2}-1)\Gamma(\frac{n}{2}-1)}{(\frac{n-3}{2})\Gamma(\frac{n-3}{2})} \\
&= \frac{n-2}{n-3}, \qquad (23)
\end{aligned}
$$

where we have used the fact that $\Gamma(x+1) = x\Gamma(x)$ for any real number $x > 0$. We first consider the cases of $n = 2$ and 3. When $n = 2$, we have:

$$
\begin{aligned}
\mu(j,2) &= \int_{-1}^{+1} k^j \frac{1}{B_2} (1-k^2)^{\frac{0}{2}-1} dk \\
&= \frac{1}{\pi} \int_{-1}^{+1} \frac{k^j}{\sqrt{1-k^2}} dk,
\end{aligned}
$$

where we have used the fact that $B_2 = B(1/2, 1/2) = \Gamma(1/2)\Gamma(1/2)/\Gamma(1) = \sqrt{\pi} \cdot \sqrt{\pi}/(1) = \pi$. Let $k = \sin t$. Then $dk = \cos t \; dt$. Therefore,

$$
\begin{aligned}
\mu(j,2) &= \frac{1}{\pi} \int_{-\pi/2}^{\pi/2} \frac{\sin^j t}{\sqrt{1-\sin^2 t}} \cos t \; dt \\
&= \frac{1}{\pi} \int_{-\pi/2}^{\pi/2} \frac{\sin^j t}{|\cos t|} \cos t \; dt \\
&= \frac{1}{\pi} \int_{-\pi/2}^{\pi/2} \sin^j t \; dt,
\end{aligned}
$$

where we have used the fact that $\cos t$ is positive in the interval $(-\pi/2, \pi/2)$. With $j = 0$, it is easy to see that $\mu(0,2) = 1$. And for $j = 1$, we see that $\mu(1,2) = 0$ since the integral of $\sin t$ is $-\cos t$. Consider therefore $j \geq 2$. Integrating by parts, we have

$$
\begin{aligned}
\mu(j,2) &= \frac{1}{\pi} \int_{-\pi/2}^{\pi/2} \sin t \; \sin^{j-1} t \; dt, \\
&= -\frac{1}{\pi} \cos t \; \sin^{j-1} t \Big|_{-\pi/2}^{\pi/2} \\
&\quad - \frac{1}{\pi} \int_{-\pi/2}^{\pi/2} -\cos t \; (j-1) \; \sin^{j-2} t \; \cos t \; dt, \\
&= \frac{j-1}{\pi} \int_{-\pi/2}^{\pi/2} \cos^2 t \; \sin^{j-2} t \; dt \\
&= \frac{j-1}{\pi} \int_{-\pi/2}^{\pi/2} (1-\sin^2 t) \; \sin^{j-2} t \; dt \\
&= \frac{j-1}{\pi} \int_{-\pi/2}^{\pi/2} \sin^{j-2} t \; dt - \frac{j-1}{\pi} \int_{-\pi/2}^{\pi/2} \sin^j t \; dt \\
&= (j-1)\mu(j-2,2) - (j-1)\mu(j,2) \\
\Rightarrow \mu(j,2) &= \frac{j-1}{j} \mu(j-2,2) \qquad (24)
\end{aligned}
$$

From this recurrence relation, we see that for odd j we have $\mu(j,2) = 0$, since $\mu(1,2) = 0$. And for even j, by using $\mu(0,2) = 1$ in the recurrence relation, it is easy to show that

$$
\mu(j,2) = \frac{(j-1)!!}{j!!},
$$

where $x!! = x(x-2)(x-4)\cdots 4 \cdot 2$ is the double factorial. Thus,

$$
\mu(j,2) = \begin{cases} 1, & \text{if } j = 0, \\ 0, & \text{if } j \text{ is odd}, \\ \frac{(j-1)!!}{j!!}, & \text{if } j \text{ is even} \end{cases} \qquad (25)
$$

Next consider $n = 3$. We have

$$
\begin{aligned}
\mu(j,3) &= \int_{-1}^{+1} k^j \frac{1}{B_3} (1-k^2)^{\frac{2}{2}-1} \; dk \\
&= \frac{1}{2} \int_{-1}^{+1} k^j \; dk, \\
&= \frac{1}{2} \frac{k^{j+1}}{j+1} \Big|_{-1}^{+1} \\
&= \frac{1-(-1)^{j+1}}{2(j+1)},
\end{aligned}
$$

where we have used the fact that $B_3 = B(2/2, 1/2) = B(1, 1/2) = \Gamma(1)\Gamma(1/2)/\Gamma(1+1/2) = \Gamma(1/2)/((1/2)(\Gamma(1/2)) = 2$. Thus,

$$
\mu(j,3) = \begin{cases} 1, & \text{if } j = 0, \\ 0, & \text{if } j \text{ is odd}, \\ \frac{1}{j+1}, & \text{if } j \text{ is even} \end{cases} \qquad (26)
$$

Finally, consider $n \geq 4$. We have through integration by parts:

$$
\begin{aligned}
\mu(j,n) &= \int_{-1}^{+1} k^j \frac{1}{B_n} (1-k^2)^{\frac{n-1}{2}-1} \; dk \\
&= \frac{1}{B_n} \int_{-1}^{+1} k^j (1-k^2)^{\frac{n-1}{2}-1} \; dk \\
&= \frac{1}{B_n} \frac{k^{j+1}}{j+1} (1-k^2)^{\frac{n-1}{2}-1} \Big|_{-1}^{+1} \\
&\quad - \frac{1}{B_n} \int_{-1}^{+1} \frac{k^{j+1}}{j+1} \left(\frac{n-3}{2}\right) (1-k^2)^{\frac{n-1}{2}-2} (-2k) \; dk \\
&= \frac{n-3}{B_n} \frac{1}{j+1} \int_{-1}^{+1} k^{j+2} (1-k^2)^{\frac{n-1}{2}-2} \; dk \\
&= \frac{n-3}{B_n} \frac{1}{j+1} \int_{-1}^{+1} k^j (1-(1-k^2))(1-k^2)^{\frac{n-1}{2}-2} \; dk \\
&= \frac{n-3}{B_n} \frac{1}{j+1} \int_{-1}^{+1} k^j (1-k^2)^{\frac{n-1}{2}-2} \; dk \\
&\quad - \frac{n-2}{B_n} \frac{1}{j+1} \int_{-1}^{+1} k^j (1-k^2)^{\frac{n-1}{2}-1} \; dk \\
&= \frac{n-2}{B_{n-2}} \frac{1}{j+1} \int_{-1}^{+1} k^j (1-k^2)^{\frac{n-1}{2}-2} \; dk \\
&\quad - \frac{n-3}{B_n} \frac{1}{j+1} \int_{-1}^{+1} k^j (1-k^2)^{\frac{n-1}{2}-1} \; dk \\
&= \frac{(n-2)}{(j+1)} \mu(j,n-2) - \frac{(n-3)}{(j+1)} \mu(j,n)
\end{aligned}
$$

where we have used Eq. (23). Thus,

$$
\mu(j,n) = \frac{n-2}{n+j-2} \mu(j,n-2) \qquad (27)
$$

If $j = 0$, then $\mu(0,n)$ is the integral of the PDF of the distribution, and hence $\mu(0,n) = 1$, for all $n \geq 2$. Consider now, odd j. From Eqs. (25), (26) and (27) we see that $\mu(j,n) = 0$, for all $n \geq 2$. Thus, consider even j. First consider even n, starting from $n = 4$. We have

$\mu(j, n-2) = \mu(j, 2)$, for $n = 4$. Putting the result from Eq. (25) into Eq. (27), we get

$$\begin{aligned}\mu(j,n) &= \frac{n-2}{n+j-2}\frac{n-4}{n+j-4}\frac{n-6}{n+j-6}\cdots\frac{4-2}{4+j-2}\frac{(j-1)!!}{j!!}\\ &= \frac{n-2}{n+j-2}\frac{n-4}{n+j-4}\frac{n-6}{n+j-6}\cdots\frac{2}{j+2}\frac{(j-1)!!}{j!!}\\ &= \frac{(n-2)!!(j-1)!!}{(n-2+j)!!}.\end{aligned}$$

Lastly, consider odd n starting from $n = 5$. We have $\mu(j, n-2) = \mu(j, 3)$, for $n = 5$. Putting the result from Eq. (26) into Eq. (27), we get

$$\begin{aligned}\mu(j,n) &= \frac{n-2}{n+j-2}\frac{n-4}{n+j-4}\frac{n-6}{n+j-6}\cdots\frac{5-2}{5+j-2}\frac{1}{j+1}\\ &= \frac{n-2}{n+j-2}\frac{n-4}{n+j-4}\frac{n-6}{n+j-6}\cdots\frac{3}{j+3}\frac{1}{j+1}\\ &= \frac{n-2}{n+j-2}\frac{n-4}{n+j-4}\frac{n-6}{n+j-6}\cdots\frac{3}{j+3}\frac{1}{j+1}\frac{(j-1)!!}{(j-1)!!}\\ &= \frac{(n-2)!!(j-1)!!}{(n-2+j)!!}\end{aligned}$$

Putting the results together, we get for $n \geq 2$

$$\mu(j,n) = \begin{cases}1, & \text{if } j = 0,\\ 0, & \text{if } j \text{ is odd},\\ \frac{(n-2)!!(j-1)!!}{(n-2+j)!!}, & \text{if } j \text{ is even}\end{cases}$$

From the above equation we see that $\mathbb{E}[K] = \mu(1, n) = 0$ and $\mathrm{Var}[K] = \mathbb{E}[K^2] - (\mathbb{E}[K])^2 = \mathbb{E}[K^2] = \mu(2, n) = \frac{1}{n}$. □

Proof of Theorem 6.

Proof. First note that $\mu = \mu(1, n) = 0$ from Eq. (14). Therefore, using the Taylor series expansion of the exponential function, Lemma 1 and Eq. (14), we have:

$$\begin{aligned}\mathbb{E}[e^{\lambda(K-\mu)}] &= \mathbb{E}[e^{\lambda K}]\\ &= \mathbb{E}\left[\sum_{j=0}^{\infty}\frac{(\lambda K)^j}{j!}\right]\\ &= \sum_{j=0}^{\infty}\frac{\lambda^j}{j!}\mathbb{E}[K^j]\\ &= \sum_{j=0,\, j \text{ even}}^{\infty}\frac{\lambda^j}{j!}\mathbb{E}[K^j]\\ &= \sum_{j=0}^{\infty}\frac{(\lambda^2)^j}{(2j)!}\mathbb{E}[K^{2j}]\\ &= 1 + \sum_{j=1}^{\infty}\frac{(\lambda^2)^j}{(2j)!}\mathbb{E}[K^{2j}]\\ &= 1 + \sum_{j=1}^{\infty}\frac{(\lambda^2)^j}{(2j)!}\frac{(n-2)!!(2j-1)!!}{(n-2+2j)!!}\end{aligned}$$

Now, note that

$$\begin{aligned}\frac{(2j-1)!!}{2j!} &= \frac{(2j-1)(2j-3)(2j-5)\cdots 3\cdot 1}{2j(2j-1)(2j-2)(2j-3)\cdots 3\cdot 2\cdot 1}\\ &= \frac{1}{2j!!}\\ &= \frac{1}{2j(2j-2)(2j-4)\cdots 4\cdot 2}\\ &= \frac{1}{2^j\cdot j(j-1)(j-2)\cdots 2\cdot 1}\\ &= \frac{1}{2^j j!}\end{aligned}$$

Therefore, the above becomes,

$$\begin{aligned}&\mathbb{E}[e^{\lambda(K-\mu)}]\\ &= 1 + \sum_{j=1}^{\infty}\frac{(\lambda^2)^j}{2^j j!}\frac{(n-2)!!}{(n-2+2j)!!}\\ &= 1 + \sum_{j=1}^{\infty}\frac{(\lambda^2)^j}{2^j j!}\frac{1}{(n+2j-2)(n+2j-4)\cdots(n+2)(n)}\\ &\leq 1 + \sum_{j=1}^{\infty}\frac{(\lambda^2)^j}{2^j j!}\frac{1}{n^j}\\ &= 1 + \sum_{j=1}^{\infty}\frac{1}{j!}\left(\frac{\lambda^2}{2n}\right)^j\\ &= \sum_{j=0}^{\infty}\frac{1}{j!}\left(\frac{\lambda^2}{2n}\right)^j\\ &= e^{-\sigma^2\lambda^2/2},\end{aligned}$$

where $\sigma^2 = \frac{1}{n} = \mathrm{Var}[K]$. □

Proof of Theorem 7

Proof. Recall first that Markov's inequality states that if X is a non-negative random variable and $a > 0$, then

$$\Pr[X \geq a] \leq \frac{\mathbb{E}[X]}{a}$$

Also, let $M_X(t) = \mathbb{E}[e^{tX}]$ be the moment generating function of X, where we assume the expectation to exist within $-h < t < h$, for some real number $h > 0$ [10, §2.3].

Let us define another random variable $Y = e^{tX}$. Since e^{tx} is an increasing function of x if $t > 0$, we have that

$$\Pr[X \geq a] = \Pr[e^{tX} \geq e^{ta}] \leq \frac{\mathbb{E}[e^{tX}]}{e^{ta}} = e^{-ta}M_X(t), \qquad (28)$$

where we have used Markov's inequality on the random variable Y with $0 < t < h$. Next e^{tx} is a decreasing function of x if $t < 0$. Therefore

$$\Pr[X \leq a] = \Pr[e^{tX} \geq e^{ta}] \leq \frac{\mathbb{E}[e^{tX}]}{e^{ta}} = e^{-ta}M_X(t), \qquad (29)$$

with $-h < t < 0$. The moment generating function M_G of R is [10, §2.3]

$$M_\mathrm{G}(t) = (1 - t/\epsilon)^{-n}, \quad t < \epsilon. \qquad (30)$$

Let $c > 1$, $a = cn/\epsilon$ and $R = X$, then for $0 < t < \epsilon$ through Eqs. (30) and (28) we get

$$\Pr\left[R \geq \frac{cn}{\epsilon}\right] \leq e^{-cnt/\epsilon}\left(1 - \frac{t}{\epsilon}\right)^{-n} = y.$$

To find the minimum value of y in the interval $0 < t < \epsilon$, we take the derivative of y, which gives

$$\begin{aligned}
y' &= -\frac{cn}{\epsilon} e^{-cnt/\epsilon} \left(1 - \frac{t}{\epsilon}\right)^{-n} \\
&\quad + e^{-cnt/\epsilon}(-n)\left(1 - \frac{t}{\epsilon}\right)^{-n-1}\left(-\frac{1}{\epsilon}\right) \\
&= -\frac{cn}{\epsilon} e^{-cnt/\epsilon} \left(1 - \frac{t}{\epsilon}\right)^{-n} + \frac{n}{\epsilon} e^{-cnt/n\epsilon} \left(1 - \frac{t}{\epsilon}\right)^{-n-1} \\
&= \frac{n}{\epsilon} e^{-cnt/\epsilon} \left(1 - \frac{t}{\epsilon}\right)^{-n-1} \left(-c\left(1 - \frac{t}{\epsilon}\right) + 1\right) \\
&= \frac{n}{\epsilon} e^{-cnt/\epsilon} \left(1 - \frac{t}{\epsilon}\right)^{-n-1} \left(\frac{ct}{\epsilon} - c + 1\right).
\end{aligned}$$

We see that the derivative is 0 if $t = \frac{(c-1)\epsilon}{c}$, which is in the interval $(0, \epsilon)$ if $c > 1$. If we take the second derivative of y, we get

$$\begin{aligned}
y'' &= -\frac{cn^2}{\epsilon^2} e^{-cnt/\epsilon} \left(1 - \frac{t}{\epsilon}\right)^{-n-1} \left(\frac{ct}{\epsilon} - c + 1\right) \\
&\quad + \frac{cn}{\epsilon} e^{-cnt/\epsilon} \left(1 - \frac{t}{\epsilon}\right)^{-n-1} \\
&\quad + \frac{n(n+1)}{\epsilon^2} e^{-cnt/\epsilon} \left(1 - \frac{t}{\epsilon}\right)^{-n-2} \left(\frac{ct}{\epsilon} - c + 1\right).
\end{aligned}$$

At $t = \frac{(c-1)\epsilon}{c}$, the first and the last term in the above expression is 0. The remaining term is positive since the exponential function is greater than 0 for any value of t, and $1 - t/\epsilon$ is also greater than 0 if $0 < t < \epsilon$. Thus, $y'' > 0$. Hence the function y is minimized at this value of t. Putting this value of t into the expression for y, we get the first statement of the theorem.

For the second statement, again let $c > 1$, $a = n/c\epsilon$ and $R = X$, then for $-h < t < 0$ through Eqs. (30) and (29) we get

$$\Pr\left[R \le \frac{n}{c\epsilon}\right] \le e^{-nt/c\epsilon} \left(1 - \frac{t}{\epsilon}\right)^{-n} = y.$$

Again, taking the derivative of y to find the value of t that minimizes y in the intervale $-h < t < 0$, we get

$$y' = \frac{n}{\epsilon} e^{-nt/c\epsilon} \left(1 - \frac{t}{\epsilon}\right)^{-n-1} \left(\frac{t}{c\epsilon} - \frac{1}{c} + 1\right).$$

We see that $y' = 0$ if $t = (1 - c)\epsilon$, which indeed satisfies $t < 0 < \epsilon$, with $c > 1$. Taking the second derivative of y, we get:

$$\begin{aligned}
y'' &= -\frac{n^2}{c\epsilon^2} e^{-nt/c\epsilon} \left(1 - \frac{t}{\epsilon}\right)^{-n-1} \left(\frac{t}{c\epsilon} - \frac{1}{c} + 1\right) \\
&\quad + \frac{n}{c\epsilon} e^{-nt/c\epsilon} \left(1 - \frac{t}{\epsilon}\right)^{-n-1} \\
&\quad + \frac{n(n+1)}{\epsilon^2} e^{-nt/c\epsilon} \left(1 - \frac{t}{\epsilon}\right)^{-n-2} \left(\frac{t}{c\epsilon} - \frac{1}{c} + 1\right).
\end{aligned}$$

Once again at $t = (1 - c)\epsilon$, the first and the last term in the above expression vanishes. The middle term is positive since the exponential function is positive for any value of t and $1 - t/\epsilon = c > 1$ for this value of t. Thus, $y'' > 0$, meaning that y is minimized at this value of t. Putting this value of t in the expression for y gives us the second statement of the theorem. □

B Noise Length Follows the Gamma Distribution

Let $f(r, S)$ denote the joint probability density function of the noise distribution, which samples a noise vector $\boldsymbol{\eta} = r\hat{\mathbf{u}}$ where $\hat{\mathbf{u}}$ is sampled uniformly over the surface of the n-dimensional hypersphere of unit radius, and where $r = \|\boldsymbol{\eta}\|$, whose distribution we seek to determine. Let $f_R(r)$ denote the probability density function of this distribution. This is given by the marginal PDF

$$f_R(r) = \int_{S(r)} f(r, S)\, dS, \tag{31}$$

where the integration is over $S(r)$, i.e., the surface area of the n-dimensional hypersphere of radius r. Now, $S(r) \propto r^{n-1}$ and we want the distribution at r to be proportional to $\exp(-\epsilon\|\boldsymbol{\eta}\|) = \exp(-\epsilon r)$. Therefore, $f_R(r) \propto r^{n-1}\exp(-\epsilon r)$. To make this into a probability density function, we must have:

$$\int_0^\infty c r^{n-1} e^{-\epsilon r}\, dr = c \int_0^\infty r^{n-1} e^{-\epsilon r}\, dr = 1$$

Let $I_{n-1} = \int_0^\infty r^{n-1} e^{-\epsilon r}\, dr$. Then

$$\begin{aligned}
I_{n-1} &= \int_0^\infty r^{n-1} e^{-\epsilon r}\, dr \\
&= \frac{e^{-\epsilon r}}{-\epsilon} r^{n-1}\Big|_0^\infty - \int_0^\infty \frac{e^{-\epsilon r}}{-\epsilon}(n-1) r^{n-2}\, dr \\
&= \frac{n-1}{\epsilon} \int_0^\infty e^{-\epsilon r} r^{n-2}\, dr \\
&= \frac{n-1}{\epsilon} I_{n-2}.
\end{aligned}$$

Now,

$$I_1 = \int_0^\infty e^{-\epsilon r}\, dr = \frac{e^{-\epsilon r}}{-\epsilon}\Big|_0^\infty = -\frac{1}{\epsilon}(0 - 1) = \frac{1}{\epsilon}.$$

Therefore,

$$\begin{aligned}
I_{n-1} &= \frac{(n-1)}{\epsilon}\frac{(n-2)}{\epsilon} \cdots \frac{(n-(n-1))}{\epsilon}\frac{1}{\epsilon} \\
&= \frac{(n-1)!}{\epsilon^n} = \frac{\Gamma(n)}{\epsilon^n}
\end{aligned}$$

Thus, $c = \epsilon^n/\Gamma(n)$, and $f_R(r) = f_{\mathrm{G}}(r)$, i.e., the gamma distribution given in Eq. (2).

As an illustration of this, suppose we want to sample a point uniformly at random on and inside a circle with radius ρ. We first sample a point uniformly at random on the circumference of the circle (e.g., via the method of zero mean, unit variance Gaussians as explained in Section 3). To now select a point uniformly at random within the circle (including its circumference), we need to find the length r of the point, where $0 \le r \le \rho$. From Eq. (31) we see that at radius r the marginal PDF of the random variable R representing the length of the point is given by

$$f_R(r) = \int_0^{2\pi r} f(r, S) dS,$$

where $2\pi r$ is the circumference of the circle with radius r. See Figure 10.

Since the point needs to have a uniform distribution, we have that $f_R(r) \propto 2\pi r$. To make it into a probability density function, we

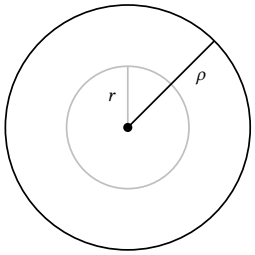


Figure 10: Sampling a point uniformly at random on and inside a circle of radius ρ. At radius r we need to integrate over the circumference of the inner circle of length $2\pi r$.

get

$$\int_0^\rho c2\pi r \; dr = 1 \Rightarrow c = \frac{1}{\pi\rho^2}.$$

Thus $f_R(r) = \frac{2\pi r}{\pi\rho^2} = \frac{2r}{\rho^2}$. Indeed, we see that if we let $f_Z(\mathbf{z})$ denote the PDF of a point $\mathbf{z}$ inside or on the circle of radius ρ, i.e., $\|\mathbf{z}\|^2 \leq \rho^2$, and let $f_U(\mathbf{u})$ denote the uniform distribution on the circumference of the unit circle which is $1/2\pi$ if $\mathbf{u}$ is on the circumference of the unit circle, and 0 otherwise, we get from Eq. (12):

$$\begin{aligned} f_Z(\mathbf{z}) &= \int_0^\rho f_R(r) f_U(\mathbf{z}/r) \frac{1}{r} \; dr \\ &= \int_0^\rho \frac{2r}{\rho^2} f_U(\mathbf{z}/r) \frac{1}{r} \; dr \\ &= \frac{2}{\rho^2} \int_0^\rho f_U(\mathbf{z}/r) \; dr \\ &= \frac{2}{\rho^2} \int_0^\rho \frac{1}{2\pi} \delta(r - \|\mathbf{z}\|) \; dr \\ &= \frac{1}{\pi\rho^2} \int_0^\rho \delta(r - \|\mathbf{z}\|) \; dr \end{aligned}$$

where δ is the Dirac delta function [18, §9.4]. Define $t = r - \|\mathbf{z}\|$, which gives $dt = dr$. Therefore, the above becomes

$$f_Z(\mathbf{z}) = \frac{1}{\pi\rho^2} \int_{-\|\mathbf{z}\|}^{\rho - \|\mathbf{z}\|} \delta(t) \; dt = \frac{1}{\pi\rho^2} \cdot 1 = \frac{1}{\pi\rho^2},$$

where the integral is 1 because $0 \in [-\|\mathbf{z}\|, \rho - \|\mathbf{z}\|]$ [18, §9.4]. This is precisely the area of the circle with radius ρ, and hence the distribution is uniform as required.