Leveraging Hierarchical Representations for Preserving Privacy and Utility in Text

Oluwaseyi Feyisetan
Amazon
sey@amazon.com

Tom Diethe
Amazon
tdiethe@amazon.co.uk

Thomas Drake
Amazon
draket@amazon.com

***Abstract*—Guaranteeing a certain level of user privacy in an arbitrary piece of text is a challenging issue. However, with this challenge comes the potential of unlocking access to vast data stores for training machine learning models and supporting data driven decisions. We address this problem through the lens of d_χ-privacy, a generalization of Differential Privacy to non Hamming distance metrics. In this work, we explore word representations in Hyperbolic space as a means of preserving privacy in text. We provide a proof satisfying d_χ-privacy, then we define a probability distribution in Hyperbolic space and describe a way to sample from it in high dimensions. Privacy is provided by perturbing vector representations of words in high dimensional Hyperbolic space to obtain a semantic generalization. We conduct a series of experiments to demonstrate the tradeoff between privacy and utility. Our privacy experiments illustrate protections against an authorship attribution algorithm while our utility experiments highlight the minimal impact of our perturbations on several downstream machine learning models. Compared to the Euclidean baseline, we observe $\geq$ 20x greater guarantees on expected privacy against comparable worst case statistics.**

***Index Terms*—privacy; document redaction; data sanitization**

I. Introduction

In Machine Learning (ML) tasks and Artificial Intelligence (AI) systems, training data often consists of information collected from users. This data can be sensitive; for example, in conversational systems, a user can explicitly or implicitly disclose their identity or some personal preference during their voice interactions. Explicit *personally identifiable information* (PII) (such as an individual's PIN or SSN) can potentially be filtered out via rules or pattern matching. However, more subtle privacy attacks occur when seemingly innocuous information is used to discern the private details of an individual [1]. This can lead to a number of attack vectors – ranging from human annotators making deductions on user queries [2] to membership inference attacks being launched against machine learning models that were trained on such data [3]. As a result, privacy-preserving analysis has increasingly been studied in statistics, machine learning and data mining [4], [5] to build systems that provide better privacy guarantees.

Of particular interest are these implicit, subtle privacy breaches which occur as a result of an adversary's ability to leverage observable patterns in the user's data. These *tracing attacks* have been described to be akin to 'fingerprinting' [6] due to their ability to identify the presence of a user's data in the absence of explicit PII values. The work by [7] demonstrates how to carry out such tracing attacks on ML models by determining if a user's data was used to train the model. These all go to illustrate that the traditional notion of PII which is used to build anonymization systems is fundamentally flawed [8]. Essentially, any part of a user's information can be used to launch these attacks, and we are therefore in a post-PII era [1]. This effect is more pronounced in naturally generated text as opposed to statistical data where techniques such as Differential Privacy (DP) have been established as a *de facto* way to mitigate these attacks.

While providing quantifiable privacy guarantees over a user's textual data has attracted recent attention [9], [10], there is significantly more research into privacy-preserving statistical analysis. In addition, most of the text-based approaches have focused on providing protections over vectors, hashes and counts [11], [12]. The question remains: what quantifiable guarantees can we provide over the actual text? We seek to answer that question by adopting the notion of d_χ-privacy [13]–[15], an adaptation of Local DP (LDP) [16] which was designed for providing privacy guarantees over location data. d_χ-privacy generalizes DP beyond Hamming distances to include Euclidean, Manhattan and Chebyshev metrics, among others. In this work, we demonstrate the utility of the Hyperbolic distance for d_χ-privacy in the context of textual data. This is motivated by the ability to better encode hierarchical and semantic information in Hyperbolic space than in Euclidean space [17]–[19].

At a high level, our algorithm preserves privacy by providing *plausible deniability* [20] over the contents of a user's query. We achieve this by transforming selected words to a high dimensional vector representation in Hyperbolic space as defined by Poincaré word embeddings [18]. We then perturb the vector by sampling noise from the same Hyperbolic space with the amount of added noise being proportional to the privacy guarantee. This is followed by a post-processing step of *discretization* where the noisy vector is mapped to the closest word in the embedding vocabulary. This algorithm conforms to the d_χ-privacy model introduced by [13] with our transformations carried out in higher dimensions, in a different metric space, and within a different domain. To understand why this technique preserves privacy, we describe motivating examples in Sec. II-A and define how we quantify privacy loss by using a series of interpretable proxy statistics in Sec. VI.

Accepted at ICDM 2019

A. Contributions

Our contributions in this paper are summarized as follows:

1) We demonstrate that the Hyperbolic distance metric satisfies d_χ-privacy by providing a formal proof in the Lorentz model of Hyperbolic space.
2) We define a probability distribution in Hyperbolic space for getting noise values and also describe how to sample from the distribution.
3) We evaluate our approach by preserving privacy against an attribution algorithm, baselining against a Euclidean model, and preserving utility on downstream systems.

II. Privacy Requirement

Consider a user interacting freely with an AI system via a natural language interface. The user's goal is to meet some specific need with respect to an issued query x. The expected norm in this specific context would be satisfying the user's request. A privacy violation occurs when x is used to make personal inference beyond what the norm allows [21]. This generally manifests in the form of unrestricted PII present in x (including, but not restricted to locations, medical conditions or personal preferences [8]). In many cases, the PII contains more semantic information than what is required to address the user's base intent and the AI system can handle the request without recourse to the explicit PII (we discuss motivating examples shortly). Therefore, our goal is to output $\hat{x}$, a semantic preserving redaction of x that preserves the user's objective while protecting their privacy. We approach this privacy goal along two dimensions (described in Sec. VI): (i) uncertainty – the adversary cannot base their guesses about the user's *identity* and *property* on information known with certainty from x; and (ii) indistinguishability – the adversary cannot distinguish whether an observed query $\hat{x}$ was generated by a given user's query x, or another similar query x'.

To describe the privacy requirements and threat model, we defer to the framework provided by [22]. First, we set our *privacy domain* to be the non-interactive *textual* database setting where we seek to release a *sanitized* database to an internal team of *analysts* who can visually inspect the queries. We also restrict this database to the one user, one query model – i.e., for the baseline, we are not concerned with providing protections on a user's aggregate data. In this model, the analyst is a required part of the system, thus, it is impossible to provide *semantic security* where the analyst learns nothing. This is only possible in a three-party cryptographic system (*e.g.* under the Alice, Bob and Eve monikers) where the analyst is different from the attacker (in our threat model, the analyst is simultaneously Bob and Eve).

We address purely privacy issues by considering that the data was willingly made available by the user in pursuit of a specific objective (as opposed to security issues [23] where the user's data might have been stolen). Therefore, we posit that the user's query x is *published* and *observable* data. Our overall aim is to protect the user from *identity* and *property* inference *i.e.*, given x, the analyst should neither be able to infer with certainty, the user's identity, nor some unique property of the user.

A. Motivating examples

To illustrate the desired functionality, which is to infer the user's high level objective while preserving privacy, let us consider the examples in Tab. I from the Snips dataset [24]:

Intent	Sample query	New word
GetWeather	will it be colder in ohio	(that) state
PlayMusic	play techno on lastfm	music; app
BookRestaurant	book a restaurant in milladore	(the) city
RateBook	rate the firebrand one of 6 stars	product
SearchCreativeWork	i want to watch manthan	(a) movie

TABLE I: Examples from the Snips dataset

In the examples listed, the underlined terms correspond to the well defined notion of 'slot values' while the other words are known as the 'carrier phrase'. The slot values are essentially 'variables' in queries which can take on different values and are identified by an instance type. We observe therefore that, replacing the slot value with a new word along the similarity or hierarchical axis does not change the user's initial intent. As a result we would expect $\hat{x}$ = '*play (music, song) from app*' to be classified in the same way as x = '*play techno on lastfm*'. We are interested in protecting the privacy of one user, issuing one query, while correctly classifying the user's intent. This model is not designed to handle multiple queries from a user, neither is it designed for handling exact queries *e.g.* booking the 'specific restaurant in milladore'.

Our objective is to create a privacy preserving mechanism M that can carry out these slot transformations $\hat{x} = M(x)$ in a principled way, with a quantifiable notion of privacy loss.

III. Privacy mechanism overview

In this section we review d_χ-privacy as a generalization of DP over metric spaces. Next, we introduce word embeddings as a natural vector space for d_χ-privacy over text. Then, we give an overview of the privacy algorithm in Euclidean space, and finish by discussing why Hyperbolic embeddings would be a better candidate for the privacy task.

A. Broadening privacy over metric spaces

Our requirement warrants a privacy metric that confers uncertainty via randomness to an observing adversary while providing indistinguishability on the user inputs and mechanism outputs. Over the years, DP [4] has been established as mathematically well-founded definition of privacy. It mathematically guarantees that an adversary observing the result of an analysis will make essentially the same inference about any user's information, regardless of whether the user's data is or is not included as an input to the analysis. Formally, DP is defined on adjacent datasets x and x' that differ in at most a single row, i.e., the *Hamming distance* between them is at most 1 and which satisfy the following inequality: We say that a randomized mechanism $M : \mathcal{X} \to \mathcal{Y}$ satisfies ε DP if

for any $x, x' \in \mathcal{X}$ the distributions over outputs of $M(x)$ and $M(x')$ satisfy the following bound: for all $y \in \mathcal{Y}$ we have

$$\frac{\Pr[M(x) = y]}{\Pr[M(x') = y]} \leq e^{\varepsilon d(x,x')} \quad . \tag{1}$$

where $d(x, x')$ is the Hamming distance and ε is the measure of privacy loss. [14] generalized the classical definition of DP by exploring other possible distance metrics which are suitable where the Hamming distance is unable to capture the notion of closeness between datasets (see Fig. 1 for other distance metrics).

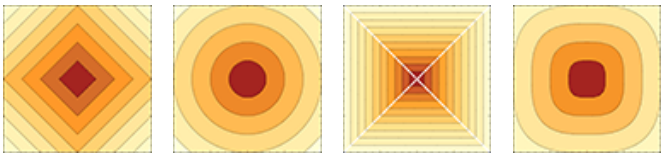


Fig. 1: Contour plots of different metrics [25]. Left to right: L_1 Manhattan distance, L_2 Euclidean distance, L_∞ Chebyshev distance, L_p Minkowski distance (L_3 shown here)

For example, a privacy model built using the Manhattan distance metric can be used to provide indistinguishability when the objective is to release the number of days from a reference point [14]. Similarly, the Euclidean distance on a $2d$ plane can be used to preserve privacy while releasing a user's longitude and latitude to mobile applications [15]. Finally, the Chebyshev distance can be adopted to to perturb the readings of smart meters thereby preserving privacy on what TV channels or movies are being watched [22].

In order to apply d_χ-privacy to the text domain, first, we need a way to organize words in a space equipped with an appropriate distance metric. One way to achieve this is by representing words using a word embedding model.

B. Word embeddings and their metric spaces

Word embeddings organize discrete words in a continuous metric space such that their similarity in the embedding space reflects their semantic or functional similarity. Word embedding models like Word2Vec [26], GloVe [27], and fastText [28] create such a mapping $\phi : \mathcal{W} \to \mathbb{R}^n$ of a set of words $\mathcal{W}$ into n-dimensional Euclidean space. The distance between words is measured by the distance function $d : \mathcal{W} \times \mathcal{W} \to \mathbb{R}_+$. This follows as $d(w, w') = d(\phi(w), \phi(w')) = \|\phi(w) - \phi(w)\|$ where $\|\cdot\|$ denotes the Euclidean norm on $\mathbb{R}^n$. The vectors $\phi(w_i)$ are generally learned by proposing a conditional probability for observing a word given its context words or by predicting the context giving the original word in a large text corpus [26].

C. The privacy mechanism in Euclidean space

Our d_χ-privacy algorithm is similar to the model introduced by [29] for privacy preserving text analysis, and [30] for author obfuscation. The algorithms are all analogous to that originally proposed by [13] and we describe it here using the Euclidean distance for word embedding vectors. In the ensuing sections, we will justify the need to use embedding models trained in Hyperbolic space (Sec. III-D) while highlighting the changes required to make the algorithm work in such space. This includes the Hyperbolic word embeddings in Sec. V-E, describing the noise distribution in Sec. V-B, and how to sample from it in Sec. V-C

Algorithm 1: Privacy Mechanism

Input: string $x = w_1 w_2 \cdots w_\ell$, privacy parameter $\varepsilon > 0$
1 **for** $i \in \{1, \ldots, \ell\}$ **do**
2 Word embedding $\phi_i = \phi(w_i)$
3 Sample noise N with density $p_N(\vec{z}) \propto \exp(-\varepsilon\|\vec{z}\|)$
4 Perturb embedding with noise $\hat{\phi}_i = \phi_i + N$
5 Discretization $\hat{w}_i = argmin_{u \in \mathcal{W}} \|\phi(u) - \hat{\phi}_i\|$
6 Insert $\hat{w}_i$ in ith position of $\hat{x}$
7 **release** $\hat{x}$

D. The case for Hyperbolic space

Even though Euclidean embeddings can model semantic similarity between discrete words in continuous space, they are not well attuned to modeling the latent hierarchical structure of words which are required for our use case. To better capture semantic similarity and hierarchical relationships between words (without exponentially increasing the dimensionality of the embeddings), recent works [18], [19], [31] propose learning the vector representation in Hyperbolic space $\phi : \mathcal{W} \to \mathbb{H}^n$. Unlike the Euclidean model, the Hyperbolic model can realize word hierarchy through the norms of the word vectors and word similarity through the distance between word vectors (see Eqn. 2). Apart from *hypernymy* relationships (e.g., LONDON $\to$ ENGLAND), Hyperbolic embeddings can also model multiple latent hierarchies for a given word (e.g., LONDON $\to$ LOCATION and LONDON $\to$ CITY). Capturing these IS-A relationships (or concept hierarchies) using Hyperbolic embeddings was recently demonstrated by [32] using data from large text corpora.

Furthermore, for Euclidean models such as [29], [30], the utility degrades badly as the privacy guarantees increase. This is because the noise injected (line 4 of Alg. 1) increases to match the privacy guarantees, resulting in words that are not semantically related to the initial word. The space defined by Hyperbolic geometry (Sec IV), in addition to the distribution of words as concept hierarchies does away with this problem while preserving privacy and utility of the user's query.

IV. HYPERBOLIC SPACE AND GEOMETRY

Hyperbolic space $\mathbb{H}^n$ is a homogeneous space with constant negative curvature [17]. The space exhibits hyperbolic geometry, characterized by a negation of the parallel postulate with infinite parallel line passing through a point. It is thus distinguished from the other two isotropic spaces: Euclidean $\mathbb{R}^n$, with zero (flat) curvature; and spherical $\mathbb{S}^n$, with constant positive curvature. Hyperbolic spaces cannot be embedded isometrically into Euclidean space, therefore embedding results in every point being a *saddle point*. In addition, the growth of the hyperbolic space area is exponential (with respect to

the curvature K and radius r), while Euclidean space grows polynomially (see Tab. II for a summary of both spaces).

Property	Euclidean	Hyperbolic
Curvature K	0	< 0
Parallel lines	1	∞
Triangles are	normal	thin
Sum of $\triangle$ angles	π	$< \pi$
Circle length	$2\pi r$	$2\pi \sinh \zeta r$
Disk area	$2\pi r^2/2$	$2\pi\ (\cosh \zeta r - 1)$

TABLE II: Properties of Euclidean and hyperbolic geometries. Parallel lines is the number of lines parallel to a line and that go through a point not on this line, and $\zeta = \sqrt{|K|}$ [17]

As a result of the unique characteristics of hyperbolic space, it can be constructed with different isomorphic models. These include: the Klein model, the Poincaré disk model, the Poincaré half-plane model, and the Lorentz (or hyperboloid) model. In this paper, we review two of the models: the Lorentz model, and the Poincaré model. We also highlight what unique properties we leverage in each model and how we can carry out transformations across them.

A. *Poincaré ball model*

The $n-$dimensional Poincaré ball $\mathcal{B}^n$ is a model of hyperbolic space $\mathbb{H}^n$ where all the points are mapped within the $n-$dimensional open unit ball i.e., $\mathcal{B}^n = \{x \in \mathbb{R}^n | \, \|x\| < 1\}$ where $\|\cdot\|$ is the Euclidean norm. The boundary of the ball i.e., the hypersphere $\mathbb{S}^{n-1}$ is not part of the hyperbolic space, but it represents points that are infinitely far away (see Fig. 2a). The Poincaré ball is a conformal model of hyperbolic space (i.e., Euclidean angles between hyperbolic lines in the model are equal to their hyperbolic values) with metric tensor: $g_p(x) = [2/(1 - \|x\|^2)]^2 \ g_e$ [1] where $x \in \mathcal{B}^n$ and g_e is the Euclidean metric tensor. The Poincaré ball model then corresponds to the Riemannian manifold $\mathcal{P}^n = (\mathcal{B}^n, g_p)$. Considering that the unit ball represents the infinite hyperbolic space, we introduce a distance metric by: $d\rho = 2dr/(1 - r^2)$ where ρ is the Poincaré distance and r is the Euclidean distance from the origin. Consequently, the growth in distance $d\rho \to \infty$ as $r \to 1$, which proves the infinite extent of the ball. Therefore, given 2 points (*e.g.* representing word vectors) $u, v \in \mathcal{B}^n$ we define the isometric invariant: [33]

$$\delta(u, v) = 2\frac{\|u - v\|^2}{(1 - \|u\|^2)(1 - \|v\|^2)}$$

then the distance function over $\mathcal{P}^n$ is given by:

$$\begin{aligned} d(u, v) &= \operatorname{arcosh}(1 + \delta(u, v)) \\ &= \operatorname{arcosh}\left(1 + 2\frac{\|u - v\|^2}{(1 - \|u\|^2)(1 - \|v\|^2)}\right) \end{aligned} \tag{2}$$

The main advantage of this model is that it is conformal - as a result, earlier research into Hyperbolic word embeddings have leveraged on this model [31], [34], [35]. Furthermore, there were existing artifacts such as the Poincaré embeddings by [18] built with this model that we could reuse for this work.

[1]The metric tensor (like a dot product) gives *local* notions of length and angle between tangent vectors. By integration local segments, the metric tensor allows us to calculate the *global* length of curves in the manifold

B. *Lorentz model*

The Lorentz model (also known as the hyperboloid or Minkowski model) is a model of hyperbolic space $\mathbb{H}^n$ in which points are represented by the points on the surface of the upper sheet of a two-sheeted hyperboloid in $(n + 1)$-dimensional Minkowski space. It is a combination of $n-$dimensional spatial Euclidean coordinates x_i^k for $k = 1, 2, ..., n$; and a time coordinate $x_i^0 > 0$. Therefore, given points $u, v \in \mathbb{R}^{n+1}$, the Lorentzian inner product (Minkowski bilinear form) is:

$$\langle u, v \rangle_{\mathcal{L}} = -u_0 v_0 + \sum_{i=1}^{n} u_i v_i \tag{3}$$

The product of a point with itself is -1, thus, we can compute the norm as $\|x\|_{\mathcal{L}} = \sqrt{\langle x, x \rangle_{\mathcal{L}}}$. We define the Lorentz model as a Riemannian manifold $\mathcal{L}^n = (\mathcal{H}^n, g_l)$ where: $\mathcal{H}^n = \{u \in \mathbb{R}^{n+1} : \langle u, v \rangle_{\mathcal{L}} = -1, u_0 > 0\}$ and the metric tensor $g_l = diag(+1, -1, ..., -1)$. Hence, given the vector representation of a word at the origin in Euclidean space $\mathbb{R}^n$ as $[0, 0, ...0]$, the word's corresponding vector location in the Hyperboloid model $\mathbb{R}^{n+1}$ is $[1, 0, ..., 0]$ where the first coordinate x_0 for $x = (x_0, x') \in \mathbb{R}^{n+1}$ is:

$$x_0 \in \mathcal{H}^n = \sqrt{1 + \|x'\|^2} \ \text{ where } \ x' = (x_1, ..., x_n) \tag{4}$$

The hyperbolic distance function admits a simple expression in $\mathcal{L}^n$ and it is given as:

$$d_l(u, v) = \operatorname{arcosh}\left(-\langle u, v \rangle_{\mathcal{L}}\right) \tag{5}$$

This distance function satisfies the axioms of a metric space (*i.e.* identity of indiscernibles, symmetry and the triangle inequality). Its simplicity and satisfaction of the axioms make it the ideal model for constructing our privacy proof.

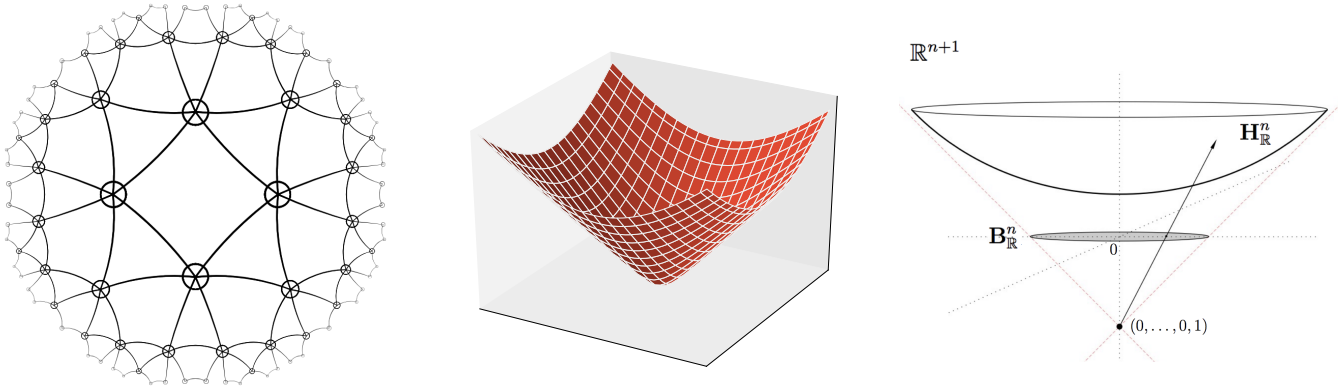
C. *Connection between the models*

Both models essentially describe the same structure of hyperbolic space characterized by its constant negative curvature. They simply represent different coordinate charts in the same metric space. Therefore, the Lorentz and Poincaré model can be related by a diffeomorphic transformation that preserves all the properties of the geometric space, including isometry. From Fig. 2c, we observe that a point $x_{\mathcal{P}}$ in the Poincaré model is a projection from the point $x_{\mathcal{L}} = (x_0, x')$ in the Lorentz model, to the hyperplane $x_0 = 0$ by intersecting it with a line drawn through $[-1, 0, ..., 0]$. Consequently, we can map this point across manifolds from the Lorentz to the Poincaré model via the transformation $x_{\mathcal{P}} : \mathcal{L}^n \to \mathcal{P}^n$ where:

$$x_{\mathcal{P}} = \frac{x'}{1 + x_0} \ \text{ where } \ x' = (x_1, ..., x_n) \tag{6}$$

In this work, we only require transformations to the Poincaré model i.e., using Eqn. 4 and 6. Mapping points back from Poincaré to Lorentz is done via:

$$x_{\mathcal{L}} = (x_0, x') = \frac{(1 + \|x\|^2, 2x)}{1 - \|x\|^2}$$



(a) (b) (c)

Fig. 2: (a) Tiling a square and triangle in the Poincaré disk $\mathcal{B}^2$ such that all line segments have identical hyperbolic length. (b) The forward sheet of a two-sheeted hyperboloid in the Lorentz model. (c) Projection [33] of a point in the Lorentz model $\mathcal{H}^n$ to the Poincaré model $\mathcal{B}^n$ (d) Embedding WebIsADb IS-A relationships in the GloVe vocabulary into the $\mathcal{B}^2$ Poincaré disk

As a result of the equivalence of the models, in this paper, we adopt the Lorentz model for constructing our d_χ-privacy proof while the word embeddings were trained in the Poincaré ball model. Consequently, the Poincaré model is also used as the basis for sampling noise from a high dimensional distribution to provide the privacy and semantic guarantees.

V. Privacy proof and Sampling mechanism

In this section, we provide the proof of d_χ-privacy for Hyperbolic space embeddings. We will be using the Lorentz model of [19] rather than the Poincaré model proposed in [18]. Then in Sec. V-B, we introduce our probability distribution for adding noise (line 4 of Alg. 1) to the word embedding vectors. Finally, we describe how to sample (line 3 of Alg. 1) from the proposed distribution in Sec. V-C. We note that whereas the privacy proof is provided in the Lorentz model, the noise distribution and the embeddings are in the Poincaré model. See Sec. IV-C for discussions on the equivalence of the models.

A. d_χ-privacy proof

In this section, we will show d_χ-privacy for the Hyperboloid embeddings of [19]. In the following, given $u, v \in \mathbb{R}^{n+1}$, we use the Lorentzian inner product from Eqn 3 *i.e.* $\langle u, v\rangle_{\mathcal{L}} = -u_0 v_0 + \sum_{i=1}^{n} u_i v_i$. The space $(\mathbb{H}^n, d)$, where $d(u,v) = \operatorname{arcosh}\left(-\langle u, v\rangle_{\mathcal{L}}\right) \in [0, \infty]$, is the hyperboloid model of n-dimensional (real) hyperbolic space.

Lemma 1. *If $u, v \in \mathbb{H}^n$, then $\langle u, v\rangle_{\mathcal{L}} \leq -1$ with equality only if $u = v$.*

Proof. Using the Cauchy-Schwarz inequality for the Euclidean inner product in $\mathbb{R}^n$ for the first inequality and a simple calculation for the second, we have:

$$\left(\sum_{i=1}^{n} u_i v_i\right)^2 = \left(\sum_{i=1}^{n} u_i^2\right)\left(\sum_{i=1}^{n} v_i^2\right) \text{ from Cauchy-Schwarz, then}$$

$$\begin{aligned}\langle u, v\rangle_{\mathcal{L}} &= -u_0 v_0 + \sum_{i=1}^{n} u_i v_i \leq -u_0 v_0 + \sqrt{\sum_{i=1}^{n} u_i^2}\sqrt{\sum_{i=1}^{n} v_i^2} \\ &= -u_0 v_0 + \sqrt{u_0^2 - 1}\sqrt{v_0^2 - 1} \leq -1\end{aligned}$$

Any line through the origin intersects $\mathbb{H}^n$ in at most one point, so Cauchy's inequality is an equality if and only if $u = v$ (as a consequence of using the positive roots).

□

Now, as was the case for the Euclidean metric, we can use the triangle inequality for the metric d which implies that for any $z \in \mathbb{R}^n$ we have the following inequality:

$$\begin{aligned}\exp\left(-\varepsilon d(z, \phi(w))\right) &= \frac{\exp\left(-\varepsilon d(z, \phi(w))\right)}{\exp\left(-\varepsilon d(z, \phi(w'))\right)} \exp\left(-\varepsilon d(z, \phi(w'))\right) \\ &= \exp\Big(\varepsilon d(z, \phi(w')) - \varepsilon d(z, \phi(w))\Big) \times \exp\Big(-\varepsilon d(z, \phi(w'))\Big) \\ &= \exp\left[\varepsilon \operatorname{arcosh}\left(-\left\langle z, \phi(w')\right\rangle_{\mathcal{L}}\right) - \varepsilon \operatorname{arcosh}\left(-\left\langle z, \phi(w)\right\rangle_{\mathcal{L}}\right)\right] \\ &\quad \times \exp\left(-\varepsilon d(z, \phi(w'))\right) \\ &<= \exp\left(\varepsilon \operatorname{arcosh}\left(\left\langle \phi(w), \phi(w')\right\rangle_{\mathcal{L}}\right)\right) \exp\Big(-\varepsilon d(z, \phi(w'))\Big) \\ &= \exp\Big(\varepsilon d\left(\phi(w), \phi(w')\right)\Big) \exp\Big(-\varepsilon d(z, \phi(w'))\Big)\end{aligned}$$

Thus, as before by plugging the last two derivations together and observing the the normalization constants in $p_N(z)$ and $p_{\phi(w)+N}(z)$ are the same, we obtain:

$$\frac{\mathsf{Pr}[M(w)=\hat{w}]}{\mathsf{Pr}\left[M\left(w'\right)=\hat{w}\right]} = \frac{\int_{C_{\hat{w}}} \exp(-\varepsilon \left\langle z, \phi(w)\right\rangle_{\mathcal{L}}) dz}{\int_{C_{\hat{w}}} \exp\left(-\varepsilon \left\langle z, \phi(w')\right\rangle_{\mathcal{L}}\right) dz} \leq \exp\left(\varepsilon d\left(w, w'\right)\right)$$

Thus, for $l = 1$ the mechanism M is εd_χ-privacy preserving. The proof for $l > 1$ is identical to the Euclidean case of [29].

B. Probability distribution for sampling noise

In this section we describe the Hyperbolic distribution from which we sample our noise perturbations. One option was sampling from the Hyperbolic normal distribution proposed by [36] (discussed in [37] and [38]) with pdf:

$$p(x|\mu,\sigma) = \frac{1}{Z(\sigma)} e^{\frac{d^2(x,\mu)}{2\sigma^2}} \text{ and } Z(\sigma) = 2\pi\sqrt{\frac{\pi}{2}}\sigma e^{\frac{\sigma^2}{2}} \operatorname{erf}\left(\frac{\sigma}{\sqrt{2}}\right)$$

However, our aim was to sample from the family of Generalized Hyperbolic distributions which reduce to the Laplacian distribution at particular location and scale parameters. By taking this approach, we can build on the proof proposed in the Euclidean case of d_χ-privacy where noise was sampled from a planar Laplace distribution [13], [14].

In the Poincaré model of Hyperbolic spaces, we have the following distance function defined in Eqn 2:

$$d(u,v) = \operatorname{arcosh}\left(1 + 2\frac{\|u-v\|^2}{(1-\|u\|^2)(1-\|v\|^2)}\right)$$

Now, analogous to the Euclidean distance used for the Laplace distribution, we wish to construct a distribution that matches this distance function. This will take the form:

$$p(x|\mu,\varepsilon) \propto (-\varepsilon d(x,\mu)) = \exp\left(-\varepsilon \operatorname{arcosh}\left(1 + 2\frac{\|x-\mu\|^2}{(1-\|x\|^2)(1-\|\mu\|^2)}\right)\right)$$

In all cases, our noise will be centered at x, and hence $\mu = 0$:

$$p(x|\mu=0,\varepsilon) = \frac{1}{Z}\exp\left(-\varepsilon d(x,0)\right) = \frac{1}{Z}\exp\left[-\varepsilon \operatorname{arcosh}\left(1 + 2\frac{\|x\|^2}{(1-\|x\|^2)}\right)\right]$$

Next, we set a new variable c and observe:

$$c = \frac{\|x\|^2}{(1-\|x\|^2)} \quad and \quad \operatorname{arcosh}(1+2c) = \log\left(2c + 2\sqrt{c^2+c} + 1\right)$$

$$p(x|\mu=0,\varepsilon) = \frac{1}{Z}\exp\left[-\varepsilon \log\left(1 + 2c + \sqrt{(1+2c)^2 - 1}\right)\right]$$

Reinserting the variable c and simplifying

$$= \frac{1}{Z}\exp\left[-\varepsilon \log\left(-\frac{2}{\|x\|-1} - 1\right)\right] = \frac{1}{Z}\left(-\frac{2}{\|x\|-1} - 1\right)^{-\varepsilon}$$

The normalization constant is then:

$$Z = \int_{-1}^{1}\left(-\frac{2}{\|x\|-1} - 1\right)^{-\varepsilon} dx = \frac{2\ {}_2F_1(1,\varepsilon,2+\varepsilon,-1)}{1+\varepsilon}$$

where ${}_2F_1(a,b;c,z)$ is the hypergeometric function defined for $|z| < 1$ by the power series

$${}_2F_1(a,b;c;z) = \sum_{n=0}^{\infty}\frac{(a)_n(b)_n}{(c)_n}\frac{z^n}{n!}.$$

where $(z)_n = \frac{\Gamma(z+n)}{\Gamma(z)}$ is the Pochhammer symbol (rising factorial). Note that Z does not depend on x, and hence can be computed in closed form *a-priori*. Our distribution is therefore:

$$p(x|\mu=0,\varepsilon) = \frac{1+\varepsilon}{2\ {}_2F_1(1,\varepsilon,2+\varepsilon,-1)}\left(-\frac{2}{\|x\|-1} - 1\right)^{-\varepsilon} \quad (7)$$

The result shown in Fig. 3 for different values of ε illustrates the PDF of the new distribution derived from Eqn. 7.

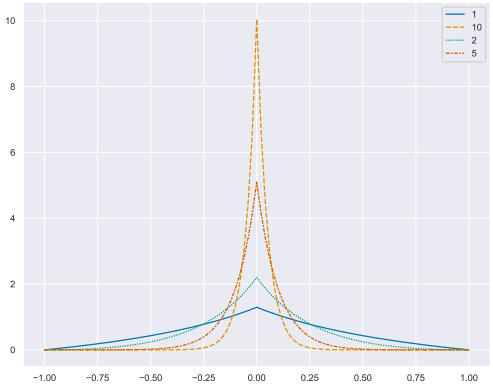


Fig. 3: $1d$ PDF of Eqn. 7 at different values of ε

C. Sampling from the distribution

Since we are unable to sample directly from the high dimensional hyperbolic distribution in Eqn. 7, we derive sampled points by simulating random walks over it using the Metropolis-Hastings (MH) algorithm. A similar approach was adopted by [37] and [38] to sample points from high dimensional Riemannian normal distributions using Monte Carlo samples. We start with $f(x,\varepsilon)$ which is our desired probability distribution as defined in Eqn. 7 where ε is the d_χ-privacy parameter. Then we choose a starting point x_0 to be the first sample. The point x_0 is set at the origin of the Poincaré model (see Fig. 2c). The sample x_0 is then updated as the current point x_t.

To select the next candidate x'_t, MH requires the point be sampled ideally from a symmetric distribution g such that $g(x_t|x'_t) = g(x'_t|x_t)$ for example, a Gaussian distribution centered at x_t. To achieve this, we sampled x'_t from the multivariate normal distribution in Euclidean space, centered at

x_t. The sampled point x_t is then translated to the $\mathcal{H}^n$ Lorentz model in $\mathbb{R}^{n+1}$ dimensional Minkowski space by setting the first coordinate using Eqn. 4. The Lorentz coordinates are then converted to the $\mathcal{B}^n$ Poincaré model in $\mathbb{R}^n$ Hyperbolic space using Eqn. 6. Therefore, the final coordinates of the sampled point x_t is in the Poincaré model.

Next, for every MH iteration, we calculate an acceptance ratio $\alpha = f(x'_t, \varepsilon)/f(x_t, \varepsilon)$ with our privacy parameter ε. If α is less than a uniform random number $u \sim \mathcal{U}([0, 1])$, we accept the sampled point by setting $x_{t+1} = x'_t$ (and sample the next point centered at this new point), otherwise, we reject the sampled point by setting x_{t+1} to the old point x_t.

Algorithm 2: Hyperbolic Noise Sampling Mechanism

Input: dimension $n > 0$, $\mu = 0$, privacy parameter $\varepsilon > 0$
Result: k results from $\mathcal{B}^n$
Let $f(x, \varepsilon)$ be the Hyperbolic noise distribution in n dimensions
set $x_0 = [1, 0, ..., 0]$
set $x_t = x_0$
set b as the initial sample burn-in period
while $i < k + b$ **do**
    sample $x' \sim \mathcal{N}(x_t, \Sigma)$
    translate $x' \rightarrow \mathcal{H}^n \rightarrow \mathcal{B}^n$
    compute $\alpha = f(x')/f(x_t)$
    sample $u \sim \mathcal{U}[(0, 1)]$
    if $u \leq \alpha$ **then**
        accept sample
        set $x_{t+1} = x'$
    else
        reject sample
        set $x_{t+1} = x_t$
release $x_i^n, ..., x_k^n$

D. Ensuring numerical stability

Sampling in high dimensional Hyperbolic spaces comes with numeric stability issues [18], [19], [37]. This occurs as the curvature and dimensionality of the space increases. This leads to points being consistently sampled at an infinite distance from the mean. Using an approach similar to [18], we constrain the updated vector to remain within the Poincaré ball by updating the noisy vectors as:

$$proj(\theta) = \begin{cases} \theta / \|\theta\| \cdot (1 + \lambda), & \text{if } \|\theta\| \geq 1 \\ \theta, & \text{otherwise} \end{cases}$$

where λ is a small constant. This occurs as a post-processing step and therefore does not affect the d_χ-privacy proof. In our experiments, we set the value to be $\lambda = 10e{-}5$ as in [18].

E. Poincaré embeddings for our analysis

The geometric structure of the Poincaré embeddings represent the metric space over which we provide privacy guarantees. By visualizing the embeddings in the Poincaré disk (see Fig. 4), we observe that higher order concepts are distributed towards the center of the disk, instances are found closer to the perimeter, and similar words are equidistant from the origin.

In this work, we train the Poincaré embeddings described in [18]. To train, we use data from WEBISADB, a large database of over 400 million hypernymy relations extracted from the CommonCrawl web corpus. We narrowed the dataset by only selecting relations of words (i.e., both the instance and the class) that occurred in the GLOVE vocabulary. To ensure we had high quality data, we further restricted the data to links that had been found at least 10 times in the CommonCrawl corpus. Finally, we filtered out stop words, offensive words and outliers (words with ≤ 2 links) from the dataset, resulting in $\approx 126,000$ extracted IS-A relations. We use the final dataset to train a $100-$dimension Poincaré embedding model. We use this model for all our analysis and experiments.

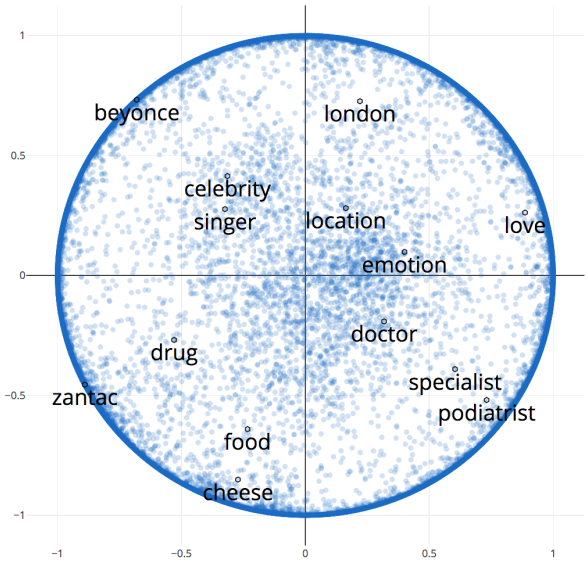


Fig. 4: Embedding WebIsADb IS-A relationships in the GloVe vocabulary into the $\mathcal{B}^2$ Poincaré disk

VI. PRIVACY CALIBRATION

In this section, we describe our approach for calibrating the values of ε for a given mechanism M. For all our discussions, $M(w) = w$ means the privacy mechanism M returns the same word, while $M(w) = \hat{w}$ represents a different random word from the algorithm. We now define the privacy guarantees that result in *uncertainty* for the adversary over the *outputs* of $M(w)$, and *indistinguishability* over the *inputs* to $M(w)$.

A. Uncertainty statistics

The uncertainty of an adversary is defined over the probability of predicting the value of the random variable $\hat{w}$ *i.e.* $\Pr[M(w) = \hat{w}]$. This follows from the definition of *Shannon entropy* which is the number of additional bits required by the adversary to reveal the user's *identity* or some secret *property*. However, even though entropy is a measure of uncertainty, there are issues with directly adopting it as a privacy metric [22] since it is possible to construct different probability distributions with the same level of entropy.

Nevertheless, we still resort to defining the uncertainty statistics by using the two extremes of the Rényi entropy [39]. The Hartley entropy H_0 is the special case of Rényi entropy with $\alpha = 0$. It depends on vocabulary size $|\mathcal{W}|$ and

is therefore a best-case scenario as it represents the perfect privacy scenario for a user as the number of words grow. It is given by $H_0 = \log_2 |\mathcal{W}|$. Min-entropy H_∞ is the special case with $\alpha = \infty$ which is a worst-case scenario because it depends on the adversary attaching the highest probability to a specific word $p(w)$. It is given by $H_\infty = -\log_2 \max_{w \in \mathcal{W}}(p(w))$.

We now describe proxies for the Hartley and Min-entropy. First, we observe that the mechanism $M(w)$ at $\varepsilon = (0, \infty)$ has full support over the entire vocabulary $\mathcal{W}$. However, empirically, the effective number of new words returned by the mechanism $M(w)$ over multiple runs approaches a finite subset. As a result, we can expect that $|\hat{\mathcal{W}}|_{\varepsilon \to 0} > |\hat{\mathcal{W}}|_{\varepsilon \to \infty}$ for a finite number of successive runs of the mechanism $M(w)$. We define this effective number $|\hat{\mathcal{W}}|$ at each value of ε for each word as S_w. Therefore, our estimate of the Hartley entropy H_0 becomes $H_0 = \log_2 |\mathcal{W}| \approx \log_2 S_w$.

Similarly, we expect that over multiple runs of the mechanism $M(w)$, as $\varepsilon \to \infty$, the probability $\mathsf{Pr}[M(w) = w]$ increases and approaches 1. As a result, we can expect that $\mathsf{Pr}[M(w) = w]_{\varepsilon \to 0} < \mathsf{Pr}[M(w) = w]_{\varepsilon \to \infty}$ for a finite number of successive runs of the mechanism $M(w)$. We define this number $\mathsf{Pr}[M(w) = w]$ at each value of ε, and for each word as N_w. Therefore, our estimate of the Min-entropy H_∞ becomes $H_\infty = -\log_2 \max_{w \in \mathcal{W}}(p(w)) \approx -\log_2 N_w$.

We estimated the quantities N_w and S_w empirically by running the mechanism $M(w)$ $1,000$ times for a random population ($10,000$ words) of the vocabulary $\mathcal{W}$ at different values of ε. The results are presented in Fig. 5a and 5b for N_w and Fig. 5c for S_w.

B. Indistinguishability statistics

Indistinguishability metrics of privacy indicate denote the ability of the adversary to distinguish between two items of interest. d_χ-privacy provides degrees of indistinguishability of *outputs* bounded by the privacy loss parameter ε. For example, given a query $x =$ 'send the package to *London*', corresponding outputs of $\hat{x} =$ 'send the package to *England*' or '...to *Britain*' provide privacy by *output* indistinguishability for the user. This is captured as uncertainty over the number of random new words as expressed in the S_w metric.

However, we also extend our privacy guarantees to indistinguishability over the *inputs*. For example, for an adversary observing the output $\hat{x} =$ 'send the package to *England*', they are unable to infer the input x that created the output $\hat{x}$ because, for the permuted word $\hat{w} =$ *England*, the original word w could have been any member of the set $\{w : \forall w \in \hat{\mathcal{W}}$ if $w \prec \hat{w}\}$, where $a \prec b$ implies that a is lower than b in the embedding hierarchy. For example, $\{\textsc{London}, \textsc{Manchester}, \textsc{Birmingham}, \ldots\} \prec \{\textsc{England}, \textsc{Britain}, \ldots\}$. Since this new statistic derives from S_w, we expect it to vary across ε in the same manner. Hence, we replace $\hat{\mathcal{W}}$ with S_w and define the new statistic K_w as:

$$K_w = \min |\{w : \forall w \in S_w \text{ if } w \prec \hat{w}\}|$$

This input indistinguishability statistic can be thought of formally in terms of *plausible deniability* [20]. In [20], plausible deniability states that an adversary cannot deduce that a particular *input* was significantly more responsible for an observed *output*. This means, there exists a set of inputs that could have generated the given output with about the same probability. Therefore, given a vocabulary size $|\mathcal{W}| > k$ and mechanism M such that $\hat{w} = M(w_1)$, we get $k-$indistinguishability over the inputs with probability γ if there are at least $k-1$ distinct words $w_2, ..., w_k \in \mathcal{W} \quad \backslash\{d_1\}$ such that:

$$\gamma^{-1} \leq \frac{\mathsf{Pr}[M(w_i)] = \hat{w}}{\mathsf{Pr}[M(w_j)] = \hat{w}} \leq \gamma \text{ for any } i, j \in \{1, 2, ..., k\}$$

We also estimated the values of K_w empirically by running the mechanism $1,000$ times for a random population ($10,000$ words) of the vocabulary $\mathcal{W}$ at different values of ε. The results are presented in Fig. 5d.

C. Selecting a value of ε

To set the value of ε for a given task, we propose following the guidelines offered by [15] in the context of location privacy by providing appropriate reformulations. They suggest mapping ε to a desired radius of *high protection* within which, all points have the same distinguishability level. We can achieve a corresponding calibration using results in Fig. 5.

The worst-case guarantees highlighted by the upper bound of the N_w statistic (see Fig. 5b) equips us with a way to fix an equivalent 'radius of *high protection*'. This 'radius' corresponds to the upper bound on the probability $\mathsf{Pr}[M(w) = w]$ which sets the guarantee on the likelihood of changing *any* word in the embedding vocabulary. Consequently, the words which are provided with the 'same distinguishability level' can be interpreted by the size of the results in Fig. 5c, and by extension, Fig. 5d. In the following sections, we investigate the impact of setting varying values of ε on the performance of downstream ML models, and how the privacy guarantees of our Hyperbolic model compares to the Euclidean baseline.

VII. Experiments

We carry out 3 experiments to illustrate the tradeoff between privacy and utility. The first two are *privacy* experiments, while the third is a set of *utility* experiments against ML tasks.

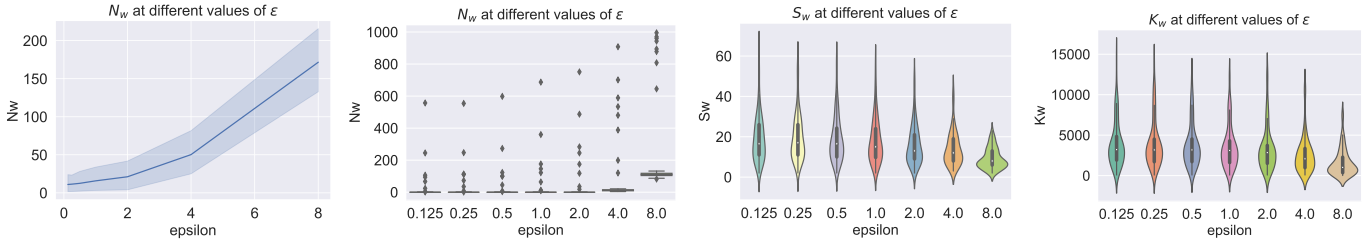
A. Evaluation metrics

- *Author predictions*: the number of authors that were re-identified in a dataset. Lower is better for privacy.
- N_w: number of times (of $1,000$) where the mechanism returned the original word. Lower is better for privacy.
- *Accuracy*: is the percentage of predictions the downstream model got right. Higher is better for utility

B. Privacy experiments I

In this section, we describe how we carried out privacy evaluations using an approach similar to [30].

1) ***Task****:* The task is to carry out author *obfuscation* against an authorship *attribution* algorithm.



(a) N_w average case (b) N_w worst case (c) S_w uncertainty statistics (d) K_w indistinguishability

Fig. 5: Privacy statistics – (a) N_w statistics: avg count of $M(w) = w$ (b) N_w statistics: max count of $M(w) = w$ (c) S_w statistics: distinct outputs for $M(w)$ (d) K_w statistics: count of different words $|\{w, w'\}|$ which resolve to same output $\hat{w}$

*2) **Baselines**:* Just as in [30], the 'adversarial' attribution algorithm is Koppel's algorithm [40]. Evaluation datasets were selected from the PAN@Clef tasks as follows:

- **PAN11** [41] the *small* dataset contained $3,001$ documents by 26 authors while the *large* set had $9,337$ documents by 72 authors. Both were derived from the Enron emails.
- **PAN12** [42] unlike the short email lengths per author in PAN11, this dataset consisted of dense volumes per author. Set-A had 3 authors with between $1,800$ and $6,060$ words; C and D had 8 authors with $\approx 13,000$ words each; while set-I consisted of 14 authors of novels with word counts ranging between $40,000$ to $170,000$.

*3) **Experiment setup**:* We ran each dataset against Koppel's algorithm [40] to get the baseline. Each dataset was then passed through our d_χ-privacy algorithm to get a new text output. This was done line by line in a manner similar to [30] *i.e.* all non stop words were considered. We evaluated our approach at the following values of $\varepsilon = 0.5, 1, 2$ and 8.

*4) **Privacy experiment results**:* The results in Table III show that our algorithm provides tunable privacy guarantees against the authorship model. It also extends guarantees to authors with thousands of words. As ε increases, the privacy guarantees decrease as clearly evidenced by the PAN11 tasks. We only show results for Kopell's algorithm because other evaluations perform worse on the baselines. *e.g.* PAN18 identifies only 65 and 50 authors, while PAN19-SVM identifies 54 and 35 authors in the PAN11 small and large datasets.

	PAN-11		PAN-12			
ε	small	large	set-A	set-C	set-D	set-I
0.5	36	72	4	3	2	5
1	35	73	3	3	2	5
2	40	78	4	3	2	5
8	65	116	4	5	4	5
∞	147	259	6	6	6	12

TABLE III: Correct author predictions (lower is better)

C. Privacy experiments II

We now describe how we evaluate the privacy guarantees of our Hyperbolic model against the Euclidean baseline.

*1) **Task and baselines**:* The objective was to compare the expected privacy guarantees for our Hyperbolic *vs.* the Euclidean baseline, given the same worst case guarantees. We evaluated against $100d$, $200d$ and $300d$ GloVe embeddings.

*2) **Experiment setup**:* We designed the d_χ-privacy algorithm in the Euclidean space as follows: (a) the embedding model was GloVe, using the same vocabulary as in the Poincaré embeddings described in Sec. V-E; (b) we sampled using the multivariate Laplacian distribution, by extending the planar Laplacian in [13] using the technique in [29]; (c) we calibrate Euclidean ε values by computing the privacy statistics N_w for a given Hyperbolic ε value.

To run the experiment, we repeat the following for the Hyperbolic and Euclidean embeddings: (1) first, we select a value of ε, (2) we empirically compute the worst case guarantee, *i.e.* the largest maximum number of times we get *any* word $\max_{w \in \mathcal{W}}[M(w) = w]$ rather than selecting a new word after our noise perturbation, (3) we compute the expected guarantee, *i.e.* the average number of times we get *all* words each time we perturb the word $\operatorname{avg}_{w \in \mathcal{W}}[M(w) = w]$.

*3) **Privacy experiment results**:* The results for the comparative privacy analysis are presented in Tab. IV. The results clearly demonstrate that for identical worst case guarantees, the expected case for the Hyperbolic model is significantly better than the Euclidean across all Euclidean dimensions. Combining this with the superior ability of the Hyperbolic model to encode both similarity and hierarchy even at lower dimensions provides a strong argument for adopting it as a d_χ-privacy preserving mechanism for the motivating examples described in Sec. II-A.

		expected value N_w			
ε	worst-case N_w	HYP-100	EUC-100	EUC-200	EUC-300
0.125	134	**1.25**	38.54	39.66	39.88
0.5	148	**1.62**	42.48	43.62	43.44
1	172	**2.07**	48.80	50.26	53.82
2	297	**3.92**	92.42	93.75	90.90
8	960	**140.67**	602.21	613.11	587.68

TABLE IV: Privacy comparisons (lower N_w is better)

D. Utility experiments

Having established Hyperbolic embeddings as being better than the Euclidean baseline for d_χ-privacy, we now demonstrate its effects on the utility of downstream models (*i.e.* we conduct utility experiments *only* on Hyperbolic embeddings).

*1) **ML Tasks**:* we ran experiments on 8 tasks (5 classification and 3 natural language tasks) to highlight the tradeoff between privacy and utility for a broad range of tasks. See Tab. V for a summary of the tasks and datasets.

name	samples	task	classes	example(s)
MR [43]	$10,662$	sentiment (movies)	2	neg, pos
CR [44]	$3,775$	product reviews	2	neg, pos
MPQA [45]	$10,606$	opinion polarity	2	neg, pos
SST-5 [46]	$12,000$	sentiment (movies)	5	0
TREC-6 [47]	$5,452$	question-type	6	LOC:city
SICK-E [48]	$10,000$	natural language inference	3	contradiction
MRPC [49]	$5,801$	paraphrase detection	2	paraphrased
STS14 [50]	$4,500$	semantic textual similarity	$[0, 5]$	4.6

TABLE V: Classification and natural language tasks

*2) **Task baselines**:* the utility results were baselined using SentEval [51], an evaluation toolkit for sentence embeddings. We evaluated the utility of our algorithm against an upper and lower bound.

To set an upper bound on utility, we ran each task using the original datasets. Each task was done on the following embedding representations: (1) InferSent [52], (2) SkipThought [53] and (3) fastText [28] (as an average of word vectors).

To set a lower bound on the utility scores, rather than replacing words using our algorithm, we replaced them with *random* words from the embedding vocabulary.

*3) **Experiment setup**:* unlike intent classification datasets such as [24] and [54], most datasets do not come with 'slot values' to be processed by a privacy preserving algorithm (see motivating examples in Sec. II-A). As a result, we pre-processed all the datasets to identify phrases with *low transition probabilities* using the privacy preserving algorithm proposed by [55].

The output from [55] yields a sequence of high frequency sub-phrases. As a result, for every query in each dataset, we are able to (inversely) select a set of low transition phrases which act as slot values to be fed into our algorithm.

For a given dataset, the output from processing each query using our algorithm is then fed into the corresponding task.

*4) **Utility experiment results**:* We evaluated our algorithm at values of $\varepsilon = 0.125, 1$ and 8. Words were sampled from the metric space defined by the $100d$ Poincaré embeddings described in Sec V-E.

The results are presented in Tab. VI. The evaluation metric for all tasks was accuracy on the test set. Across all the experiments, our algorithm yielded better results that just replacing with random words. In addition, and as expected, at lower values of $\varepsilon = 0.125$, we record lower values of utility across all tasks. Conversely at the higher value of $\varepsilon = 8$, the accuracy scores get closer to the baselines. All the results illustrate the tradeoff between privacy and utility. It also shows that we can achieve tunable privacy guarantees with minimal impact on the utility of downstream ML models.

VIII. Related work

There are two sets of research most similar to ours. The recent work by [29] and [30] applies d_χ-privacy to text using similar techniques to ours. However, their approach was done in Euclidean space while ours used word representations in, and noise sampled from Hyperbolic space. As a result, our approach can better preserve semantics at smaller values of ε by selecting hierarchical replacements.

The next set include research by [56], [57], and [58]. These all work by identifying sensitive terms in a document and replacing them by some generalization of the word. This is similar to what happens when we sample from Hyperbolic space towards the center of the Poincaré ball to select a hypernym of the current word. The difference between these and our work is the mechanism for selecting the words and the privacy model used to describe the guarantees provided.

IX. Conclusion

This paper is the first to demonstrate how hierarchical word representations in Hyperbolic space can be deployed to satisfy d_χ-privacy in the text domain. We presented a theoretical proof of the privacy guarantees in addition to defining a probability distribution for sampling privacy preserving noise from Hyperbolic space. Our experiments illustrate that our approach preserves privacy against an author attribution model and utility on several downstream models. Compared to the Euclidean baseline, we observe $>$ 20x greater guarantees on expected privacy against comparable worst case statistics. Our results significantly advance the study of d_χ-privacy, making generalized differential privacy with provable guarantees closer to practical deployment in the text domain.

References

[1] C. Dwork, A. Smith, T. Steinke, and J. Ullman, "Exposed! a survey of attacks on private data," *Annual Rev. of Stats and Its Application*, 2017.

[2] O. Feyisetan, T. Drake, B. Balle, and T. Diethe, "Privacy-preserving active learning on sensitive data for user intent classification," *arXiv preprint arXiv:1903.11112*, 2019.

[3] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *SP*. IEEE, 2017.

[4] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *TCC*. Springer, 2006.

[5] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *SIGSAC CCS*. ACM, 2015, pp. 1310–1321.

[6] M. Bun, J. Ullman, and S. Vadhan, "Fingerprinting codes and the price of approximate differential privacy," *SIAM Journal on Computing*, 2018.

[7] C. Song and V. Shmatikov, "Auditing data provenance in text-generation models," in *ACM SIGKDD*, 2019. [Online]. Available: https://arxiv.org/pdf/1811.00513.pdf

[8] P. M. Schwartz and D. J. Solove, "The PII problem: Privacy and a new concept of personally identifiable information," *NYUL rev.*, 2011.

[9] M. Coavoux, S. Narayan, and S. B. Cohen, "Privacy-preserving neural representations of text," in *EMNLP*, 2018.

[10] B. Weggenmann and F. Kerschbaum, "Syntf: Synthetic and differentially private term frequency vectors for privacy-preserving text mining," in *ACM SIGIR*. ACM, 2018.

[11] Ú. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," in *SIGSAC CCS*, 2014.

[12] T. Wang, J. Blocki, N. Li, and S. Jha, "Locally differentially private protocols for frequency estimation," in *USENIX*, 2017, pp. 729–745.

[13] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *ACM SIGSAC CCS*. ACM, 2013, pp. 901–914.

[14] K. Chatzikokolakis, M. E. Andrés, N. E. Bordenabe, and C. Palamidessi, "Broadening the scope of differential privacy using metrics," in *Intl. Symposium on Privacy Enhancing Technologies Symposium*, 2013.

[15] K. Chatzikokolakis, C. Palamidessi, and M. Stronati, "Constructing elastic distinguishability metrics for location privacy," *PETS*, 2015.

		HYP-100d			*original*		
dataset	*random*	$\varepsilon = \mathbf{0.125}$	$\varepsilon = \mathbf{1}$	$\varepsilon = \mathbf{8}$	InferSent	SkipThought	fastText-BoV
MR	58.19	58.38	63.56	74.52	81.10	79.40	78.20
CR	77.48	83.21^{**}	83.92^{**}	85.19^{**}	86.30	83.1	80.20
MPQA	84.27	88.53^{*}	88.62^{*}	88.98^{*}	90.20	89.30	88.00
SST-5	30.81	41.76	42.40	42.53	46.30	–	45.10
TREC-6	75.20	82.40	82.40	84.20^{*}	88.20	88.40	83.40
SICK-E	79.20	81.00^{**}	82.38^{**}	82.34^{**}	86.10	79.5	78.9
MRPC	69.86	74.78^{*}	75.07^{*}	75.01^{*}	76.20	–	74.40
STS14	0.17/0.16	0.44/0.45	$0.45/0.46^{*}$	$0.52/0.53^{*}$	0.68/0.65	0.44/0.45	0.65/0.63

TABLE VI: Accuracy scores on classification tasks. * indicates results better than 1 baseline, ** better than 2 baselines

[16] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What can we learn privately?" *SIAM J. on Computing*, 2011.

[17] D. Krioukov, F. Papadopoulos, M. Kitsak, A. Vahdat, and M. Boguná, "Hyperbolic geometry of complex networks," *Physical Review E*, vol. 82, no. 3, p. 036106, 2010.

[18] M. Nickel and D. Kiela, "Poincaré embeddings for learning hierarchical representations," in *NeurIPS*, 2017.

[19] ——, "Learning continuous hierarchies in the lorentz model of hyperbolic geometry," *arXiv preprint arXiv:1806.03417*, 2018.

[20] V. Bindschaedler, R. Shokri, and C. A. Gunter, "Plausible deniability for privacy-preserving data synthesis," *VLDB Endowment*, 2017.

[21] H. Nissenbaum, "Privacy as contextual integrity," *Washington Law Review*, vol. 79, p. 119, 2004.

[22] I. Wagner and D. Eckhoff, "Technical privacy metrics: a systematic survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 3, p. 57, 2018.

[23] D. E. Bambauer, "Privacy versus security," *Journal of Criminal Law & Criminology*, vol. 103, p. 667, 2013.

[24] A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier *et al.*, "Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces," *arXiv:1805.10190*, 2018.

[25] C. Ruegg, M. Cuda, and J. V. Gael, "Distance metrics," 2009. [Online]. Available: https://numerics.mathdotnet.com/Distance.html

[26] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NeurIPS*, 2013.

[27] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *EMNLP*, 2014.

[28] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *TACL*, vol. 5, 2017.

[29] O. Feyisetan, B. Balle, T. Diethe, and T. Drake, "Privacy- and utility-preserving textual analysis via calibrated multivariate perturbations," in *ACM WSDM*, 2020.

[30] N. Fernandes, M. Dras, and A. McIver, "Generalised differential privacy for text document processing," *Principles of Security and Trust*, 2019.

[31] O. Ganea, G. Becigneul, and T. Hofmann, "Hyperbolic entailment cones for learning hierarchical embeddings," in *ICML*.

[32] M. Le, S. Roller, L. Papaxanthos, D. Kiela, and M. Nickel, "Inferring concept hierarchies from text corpora via hyperbolic embeddings," *arXiv preprint arXiv:1902.00913*, 2019.

[33] M. D. Staley, "Models of hyperbolic geometry," 2015.

[34] G. Alanis-Lobato, P. Mier, and M. A. Andrade-Navarro, "Efficient embedding of complex networks to hyperbolic space via their laplacian," *Scientific reports*, vol. 6, p. 30108, 2016.

[35] A. Tifrea, G. Bécigneul, and O.-E. Ganea, "Poincaré glove: Hyperbolic word embeddings," *arXiv preprint arXiv:1810.06546*, 2018.

[36] S. Said, L. Bombrun, and Y. Berthoumieu, "New riemannian priors on the univariate normal model," *Entropy*, 2014.

[37] E. Mathieu, C. L. Lan, C. J. Maddison, R. Tomioka, and Y. W. Teh, "Hierarchical representations with poincaré variational auto-encoders," *arXiv preprint arXiv:1901.06033*, 2019.

[38] I. Ovinnikov, "Poincaré wasserstein autoencoder," *arXiv preprint arXiv:1901.01427*, 2019.

[39] A. Rényi, "On measures of entropy and information," HUNGARIAN ACADEMY OF SCIENCES Budapest Hungary, Tech. Rep., 1961.

[40] M. Koppel, J. Schler, and S. Argamon, "Authorship attribution in the wild," *LREC*, vol. 45, no. 1, pp. 83–94, 2011.

[41] S. Argamon and P. Juola, "Overview of the international authorship identification competition at PAN-2011." in *CLEF*, 2011.

[42] P. Juola, "An overview of the traditional authorship attribution subtask." in *CLEF (Online Working Notes/Labs/Workshop)*, 2012.

[43] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *ACL*, 2005.

[44] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *ACM SIGKDD*, 2004, pp. 168–177.

[45] J. Wiebe, T. Wilson, and C. Cardie, "Annotating expressions of opinions and emotions in language," *LREC*, 2005.

[46] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *EMNLP*, 2013, pp. 1631–1642.

[47] X. Li and D. Roth, "Learning question classifiers," in *COLING*, 2002.

[48] M. Marelli, S. Menini, M. Baroni, L. Bentivogli, R. Bernardi, and R. Zamparelli, "A sick cure for the evaluation of compositional distributional semantic models," in *LREC*, 2014.

[49] B. Dolan, C. Quirk, and C. Brockett, "Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources," in *COLING*, 2004.

[50] E. Agirre, C. Banea, C. Cardie, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, R. Mihalcea, G. Rigau, and J. Wiebe, "Semeval-2014 task 10: Multilingual semantic textual similarity," in *SemEval*, 2014.

[51] A. Conneau and D. Kiela, "Senteval: An evaluation toolkit for universal sentence representations," *arXiv preprint arXiv:1803.05449*, 2018.

[52] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," in *EMNLP*, 2017.

[53] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler, "Skip-thought vectors," in *NeurIPS*, 2015.

[54] G. Tur, D. Hakkani-Tür, and L. Heck, "What is left to be understood in atis?" in *2010 IEEE Spoken Language Technology Workshop*, 2010.

[55] R. Chen, G. Acs, and C. Castelluccia, "Differentially private sequential data publication via variable-length n-grams," in *SIGSAC CCS*, 2012.

[56] C. M. Cumby and R. Ghani, "A machine learning based system for semi-automatically redacting documents." in *IAAI*, 2011.

[57] D. Sánchez and M. Batet, "C-sanitized: A privacy model for document redaction and sanitization," *JAIST*, vol. 67, no. 1, pp. 148–163, 2016.

[58] B. Anandan, C. Clifton, W. Jiang, M. Murugesan, P. Pastrana-Camacho, and L. Si, "t-plausibility: Generalizing words to desensitize text," *Trans. Data Privacy*, vol. 5, no. 3, pp. 505–534, 2012.