Approximate PSI with Near-Linear Communication

Wutichai Chongchitmate*[1], Steve Lu†[2], and Rafail Ostrovsky‡ §[3]

[1]Chulalongkorn University
[2]Stealth Software Technologies, Inc.
[3]UCLA

Abstract

Private Set Intersection (PSI) is a protocol where two parties with individually held confidential sets want to jointly learn (or secret-share) the intersection of these two sets and reveal nothing else to each other. In this paper, we introduce a natural extension of this notion to approximate matching. Specifically, given a distance metric between elements, an *approximate* PSI (Approx-PSI) allows to run PSI where "close" elements match. Assuming that elements are either "close" or sufficiently "far apart", we present an Approx-PSI protocol for Hamming distance that improves the overall efficiency compared to all existing approximate PSI solutions. In particular, we achieve a near-linear $\tilde{\mathcal{O}}(n)$ communication and computation complexity, an improvement over the previously best-known $\tilde{\mathcal{O}}(n^2)$. We also show Approx-PSI protocols for Edit distance (also known as Levenstein distance), Euclidean distance and angular distance by deploying results on low distortion embeddings to Hamming distance. The latter two results further imply secure Approx-PSI for other metrics such as *cosine similarity* metric. Our Approx-PSI for Hamming distance is up to 20 times faster and communicating 30% less than best known protocols when (1) matching small binary vectors; or (2) matching large threshold; or (3) matching large input sets. We also apply our technique to analyze private approximate membership computation, which can be viewed as asymmetric case of approximate PSI, and obtain a protocol with sublinear communication.

*`wutichai.ch@chula.ac.th`

†`steve@stealthsoftwareinc.com`

‡`rafail@cs.ucla.edu`

§This research was supported in part by DARPA under Cooperative Agreement HR0011-20-2-0025, the Algorand Centers of Excellence programme managed by Algorand Foundation, NSF grants CNS-2246355, CCF-2220450, CNS-2001096, US-Israel BSF grant 2022370, Amazon Faculty Award and Sunday Group. Any views, opinions, findings, conclusions, or recommendations contained herein are those of the author(s) and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA, the Department of Defense, the Algorand Foundation, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes, notwithstanding any copyright annotation therein.

1 Introduction

Secure computation protocols enable two or more parties to engage in distributed computation while preserving the confidentiality of their inputs. Among these, private set intersection (PSI) has recently garnered significant research attention as a specialized secure computation protocol. PSI allows parties to compute the intersection, the common elements between their input sets without exposing other unrelated data. Consequently, at the end of the protocol, the parties are only aware of the shared elements, ensuring confidentiality. This characteristic has made PSI indispensable in various applications ranging from private contact discovery and business data matching to efficient data management and contact tracing. We refer the reader to recent literature [IKN+20, PRTY19, DPT20, CM20, MPR+20, RT21, RS21, GPR+21] and references therein.

In numerous applications, identifying an exact overlap between both parties' datasets might be improbable or overly restrictive. Here, discovering approximate matches – elements that share a "distance" under a specified threshold – becomes increasingly relevant. In the rapidly evolving landscape of privacy-preserving data analysis, these secure protocols adept at identifying such approximate matches are gaining traction, signifying their potential in recognizing analogous elements spanning datasets. This can be useful in various applications, such as:

- Biometric data: If two parties have databases of biometric data (like fingerprints or facial features represented as vectors), they may want to find matches, or near matches due to variations in sampling biometric data, without revealing the entirety of their databases [Dau09, UCK+21].

- Genomic data: Parties might be interested in finding genomic sequences that are close matches without revealing sensitive genomic data [MKHSO17, WHZ+15] as similarities of such data are already useful in medical diagnoses or resulting features in biology.

- Security: Traditional methods of identifying malicious network traffic rely on exact signature matches or IP addresses [MPDC19], potentially missing novel or slightly altered threats. Near match intersection allows network security tools to detect traffic patterns similar to known attack signatures and cover ranges of potentially malicious IP addresses [CSF+07, WACL10].

- Image data: In fields like computer vision and image processing, parties may need to match similar images without revealing their entire datasets. This is crucial for tasks such as object recognition, content-based image retrieval, and image classification, benefiting areas like autonomous vehicles, surveillance systems, and medical imaging analysis [KM21].

Distance-Aware PSI. Recently, Chakraborti *et al.* [CFR23] introduced a variant of PSI, called *distance-aware PSI (DA-PSI)*. In this setting, two parties jointly compute a set of pairs of elements, one from each of their individual datasets, that are within a specified threshold based on a particular distance metric. More precisely, given input sets $A, B \subseteq \mathcal{U}$ for the two parties, and a distance metric δ defined on $\mathcal{U}$, the objective of DA-PSI is to securely compute a set $S = \{(a, b) \in A \times B : \delta(a, b) \leq d\}$, with d being a pre-defined threshold.

Nevertheless, a significant challenge associated with existing DA-PSI protocols is their extensive communication complexity. This complexity limits their practicality, especially in contexts demanding fast or nearly instantaneous feedback. In particular, the communication and computation complexity of the DA-PSI protocol for the Hamming distance in [CFR23] is $\tilde{\mathcal{O}}(n^2)$, where n represents the size of sets A and B. Such scalability issues render these protocols impractical for analyzing extensive data sets.

Structure-Aware PSI. Garimella *et al.* [GRS22] introduced another related PSI variant called *structure-aware PSI (sa-PSI)*. In this setting, the receiver's set adheres to a specific structure, for instance, a union of fixed-radius balls based on a particular distance metric. The output is the same as the standard PSI for the receiver, but the efficiency (computation and communication) is only influenced by the structure of the receiver's set. In the case of the union of balls, the efficiency would depends on the number of balls present in the receiver's set, rather than the total number of individual elements.

The sa-PSI concept is broad since the sender's set structure can vary widely. Nonetheless, a primary area of interest within sa-PSI centers around the previously mentioned case of a union of disjoint balls with a fixed radius. Considering a distance metric δ and a ball radius d, sa-PSI is similar to DA-PSI with distance threshold d. However, the distinctions lies in their outputs: DA-PSI yields a set of pairs to both parties, when viewing in terminology of sa-PSI, include both the sender's elements and the centers of the receiver's balls, while sa-PSI outputs the intersection, meaning the elements in the unstructured set, to one of the parties [GGM24]. When this result made known to the party with the unstructured set, the centers of the balls are still concealed. Notably, in a semi-honest model, parties involved in an sa-PSI protocol can subsequently exchange data to discern these pairs, suggesting that sa-PSI implies DA-PSI under these conditions, but not the other way around. While the efficiency of the sa-PSI protocols in [GRS22, GRS23, GGM24] are linear in the number of balls, their construction is specifically for the ℓ_∞ norm for integral vectors.

Private Approximate Membership Computation. Kulshrestha and Mayer [KM21] introduced a variant of private membership testing (PMT), referred to as private exact membership computation (PEMC) in [KM21], where a client queries whether its input belongs to a large database. In *private approximate membership computation (PAMC)*, instead of exact matching, the protocol outputs 1 if an element in the database is "sufficiently close" to the queried element with respect to a certain metric. The authors constructed a PAMC protocol for images by first converting them to binary strings where similar images are close under the Hamming distance metric. Therefore, the protocol can also be applied to any database of binary strings assuming they are uniformly distributed. Nevertheless, the analysis of correctness of the protocol is conducted experimentally on images.

In this context, the objective is to construct a protocol that is sublinear in the database size, as the obvious approach would be securely comparing the query to each element in the database. While the PAMC protocol in [KM21] achieved sublinear communication, it has high false negative rate (FNR), as determined experimentally. Specifically, the protocol, using the parameters given in [KM21], has about 16% chance of failing to detect a match.

1.1 Approximate PSI

Here, we consider another setting of PSI where elements are from a metric space, i.e., a set $\mathcal{U}$, equipped with a distance metric δ. Instead of computing the intersection or precise matches of elements from each set, we consider approximate matches (with respect to δ), which are pairs of elements that have distance at most d. When $d = 0$, this setting is equivalent to the standard PSI. When $d > 0$ and the protocol output is the set of pairs of matches, the variant is called DA-PSI by [CFR23]. Given that both input sets have size n, the upper bound for matched pairs is n^2. This creates challenges to avoid the quadratic communication as in [CFR23].

To reduce the excessive communication costs, we introduce an additional constraint: for any pairs of elements $a \in A$ and $b \in B$, either $\delta(a, b) \leq d$ or $\delta(a, b) \geq td$ for some $t > 3$. This allows for clustering elements from A and B, that are within distance d of each other. Each cluster in each input set is represented by only one element from that cluster. By only considering the representations of the clusters, we further assume that elements $a, a' \in A$ satisfy $\delta(a, a') \geq td$, and each element in A can match with at most one element in B and vice versa. Finally, one party (or both) outputs which elements in their set near-match with elements in the other party's set.

Our proposed PSI variant offers flexibility: it can output either one party's elements (as in sa-PSI), both parties' own elements, or element pairs (as in DA-PSI). We call this problem an *approximate PSI (Approx-PSI)*, and t the *gap* distinguishing matches from non-matches. The setting in the first variant is similar to sa-PSI for the structure of the union of disjoint fixed-radius balls with center $(t - 1)d$ apart, with additional assumption on non-structure side that elements must also be far apart. The setting where one party's set is a single element is similar to that of PAMC.

Nevertheless, imposing such a restriction is difficult when honest parties are unaware of the counterpart's elements. Our approach assumes both input sets lie within a subset $\mathcal{S} \subseteq \mathcal{U}$, where every pair of elements in this subset is either near or far apart. There are various applications where such conditions may arise naturally. For instance, a set may contain a compilation of texts and their small-error-induced variants. A single base text could have close relatives with just a handful of typographical mistakes while remaining entirely distinct from other base texts within the same set. Similarly, it could be a set of ID numbers engineered with error-correcting properties or checksums. Likewise, in similar image matching, small changes in image resolution or lighting can make two images appear almost identical, even if they are not the same, while completely different images are not nearly so [KM21]. In these sets, items consist of those that are far apart together with their variants which are nearly identical.

We note that when such condition does not hold across the input sets, our protocol can be modified to remain correct when elements within each set are clustered and only represented by elements that are far apart. The false negatives only occurs for the omitted elements clustered around representatives as the transitive property of being near no longer holds.

Our goal is to find an approximate PSI protocol with linear communication complexity in n, the size of both input sets, improving the result directly implies by the DA-PSI of [CFR23].

Euclidean and angular distance. Euclidean distance measures the straight-line distance between points in a multi-dimensional space, capturing geometric relationships among continuous variables. It offers a holistic view of positional relationships between vectors, suitable for applications requiring similarity or dissimilarity assessment between multi-attribute entities.

Angular distance measures the angle between vectors, focusing on their directions rather than positions. It evaluates vector orientation or alignment, making it ideal for text similarity in natural language processing or preference analysis in recommendation systems. Unlike Euclidean distance, angular distance highlights relationships based on direction rather than distance.

Integrating Euclidean and angular distance metrics into Approx-PSI protocols holds vast potential, especially for spatial or multi-dimensional analysis. In machine learning and data science, such protocols can facilitate secure k-means clustering or nearest neighbor searches across distributed datasets. In financial analytics, Approx-PSI can enhance fraud detection by identifying the closeness of transactions in a multi-dimensional feature space. In medical research, it enables secure comparison of patient data across healthcare organizations, finding similarities in symptoms or treatment responses without compromising privacy. This expands Approx-PSI's utility beyond set intersection to more nuanced, privacy-preserving analytics in multi-dimensional data environments.

Edit distance. Edit distance, particularly *Levenshtein* distance, is crucial for assessing similarity between sequences like text strings, genetic data, or numerical time-series. It measures the minimum number of single-character edits – insertions, deletions, substitutions – needed to transform one sequence into another, providing a detailed understanding of sequence similarity or difference. This metric is essential in fields such as computational biology, linguistics, data mining, and cybersecurity for operations like sequence alignment, clustering, and anomaly detection [WHZ+15].

Approx-PSI for edit distance metrics enables secure, privacy-preserving computations that require detailed data similarity analysis. For example, in genomic research, it allows secure identification and evaluation of shared genetic markers. In natural language processing, it can facilitate secure collaborative filtering or content recommendation by considering text string edit distances. Thus, Approx-PSI with edit distance metrics significantly advances secure multi-party computations involving sequence similarity.

1.2 Related Work

PSI for approximate or near-matches using Hamming distance has been studied in securely comparing biometric or fuzzy data [OPJM10,HEKM11,UCK+21]. Secure Hamming distance comparison can be adapted into DA-PSI or Approx-PSI protocols by comparing all n^2 pairs of elements [OPJM10,HEKM11]. Uzun *et al.* [UCK+21] developed a method for comparing multiple elements simultaneously using fully homomorphic encryption (FHE). Their protocol, called *fuzzy labeled PSI*, is efficient in the client-server setting and uses a sub-sampling technique that trades off some accuracy for nearly linear communication, though the computation remains quadratic.

Chakraborti *et al.* [CFR23] formally defined and constructed the first DA-PSI for Hamming distance that the communication and computation complexity do not depend on the element size. Thus, the resulting protocol is more efficient when $d \ll \ell$. However, their protocol is quadratic in the number of elements. Additionally, they also constructed DA-PSI for integers with their difference as distance with linear communication complexity.

Garimella *et al.* [GRS22] defined and constructed the sa-PSI protocol for the case of disjoint balls of u-bit integer vectors with ℓ_∞ norm, and a more efficient one where centers of the balls are far apart. The original protocols are secure against semi-honest adversaries, and later improved in [GRS23] using derandomizable function secret sharing to be secure against malicious adversaries, and in [GGM24] using incremental function secret sharing to be more efficient, allow overlapping balls and switching the party with structured set.

Kulshrestha and Mayer [KM21] defined and constructed a PAMC protocol for matching images under simple manipulations, such as resizing, blurring, edge cropping, or small rotation. Since the protocol first maps images to 256-bit binary strings where similar images have a small Hamming distance, it can be applied to any uniformly distributed binary strings under Hamming distance metric. The constructed protocol is sublinear in communication but has a high FNR, and it has only been evaluated experimentally for some parameters.

1.3 Our Results

In this work, we present Approx-PSI for Hamming distance, which can be converted to Approx-PSI for three other distance metrics: Euclidean distance, angular distance and edit distance. We summarize our results in Table 1. Our protocols are near linear in the number of elements. Additionally, we construct a PAMC with negligible FNR which can be viewed as Approx-PSI where one set is a single element.

Hamming distance. Our main result is an Approx-PSI protocol for Hamming distance for gap $t \geq 2$ with $\tilde{\mathcal{O}}(n^{1+\frac{1}{t-1}})$ communication. For $t = \mathcal{O}(\log n)$, the protocol has near linear $\tilde{\mathcal{O}}(n)$ communication, and only gains sub-linear multiplicative factor for $t = \mathcal{O}(\log \log n)$. Our protocols are secure against semi-honest adversaries. We briefly discuss an extension to security against malicious adversaries in Appendix I.

Approx-PSI for other distance metrics. We demonstrate how to achieve Approx-PSI for Euclidean distance, angular distance (which implies cosine similarity), and edit distance metrics using our Approx-PSI for Hamming distance. Our reductions from these other distance metrics leverage the gap setting.

PAMC and unbalanced Approx-PSI. We also apply the technique we use to construct Approx-PSI to reduce the FNR from [KM21], and analyze the result mathematically. We construct the PAMC with negligible FNR in the same setting as [KM21]: uniformly distributed binary strings under the Hamming distance and the database are not required to have a gap like our Approx-PSI. The resulting protocol has sublinear $\tilde{\mathcal{O}}(n^\epsilon)$ for some $\frac{1}{3} < \epsilon < \frac{2}{3}$ assuming efficient PIR protocol that communicates $\mathcal{O}(n^{\frac{1}{3}})$ such as [LMRSW24].

Table 1: Asymptotic communication ad computation of our protocols in comparison to existing works. The protocol in [UCK+21] has false positive and false negative, depending on parameters m, B, T with $T = \mathcal{O}(\ell)$. The protocol for the Euclidean distance assumes that all input vectors are within a ball of constant radius. We assume $\log n < \lambda < \ell$ to simplify some notations.

<table>
<tr><th>Metric</th><th>Prot.</th><th>Gap</th><th>Communication</th><th>Computation</th></tr>
<tr><td rowspan="2">ℓ_∞</td><td>[GRS23]</td><td>$\mathcal{O}(1)$</td><td>$\mathcal{O}(n\lambda^2\ell + \lambda d^\ell)$</td><td>$\mathcal{O}(nd^\ell)$</td></tr>
<tr><td>[GGM24]</td><td>$\mathcal{O}(1)$</td><td>$\mathcal{O}(n^2\lambda d\ell)$</td><td>$\mathcal{O}(nd\ell)$</td></tr>
<tr><td rowspan="5">Hamming</td><td>[UCK+21]</td><td>1</td><td>$\mathcal{O}\left(\frac{n^2T}{mB}\lambda\right)$</td><td>$\mathcal{O}\left(\frac{n^2T}{m}\lambda\right)$</td></tr>
<tr><td>[CFR23]</td><td>1</td><td colspan="2">$\mathcal{O}(n^2d^2\lambda)$</td></tr>
<tr><td rowspan="3">Ours</td><td>$\mathcal{O}(\log n)$</td><td colspan="2">$\mathcal{O}(n\lambda\ell)$</td></tr>
<tr><td>$\mathcal{O}(\log\log n)$</td><td colspan="2">$n^{1+o(1)}\lambda\ell$</td></tr>
<tr><td>$t = \mathcal{O}(1)$</td><td colspan="2">$\mathcal{O}(n^{1+\frac{1}{t-1}}\lambda\ell)$</td></tr>
<tr><td rowspan="2">Euclidean</td><td rowspan="2">Ours</td><td>$\mathcal{O}(\log n)$</td><td colspan="2">$\mathcal{O}(n(\text{polylog}\, n)\lambda^2)$</td></tr>
<tr><td>$\mathcal{O}(1)$</td><td colspan="2">$\mathcal{O}(n^{1+\epsilon}\lambda^2)$</td></tr>
<tr><td rowspan="2">Angular</td><td rowspan="2">Ours</td><td>$\mathcal{O}(\log n)$</td><td colspan="2">$\mathcal{O}(n\log^2 n\lambda^2)$</td></tr>
<tr><td>$\mathcal{O}(1)$</td><td colspan="2">$\mathcal{O}(n^{1+\epsilon}\lambda^2)$</td></tr>
<tr><td rowspan="2">Edit</td><td rowspan="2">Ours</td><td>$\mathcal{O}(\ell^\epsilon\log n)$</td><td colspan="2">$\mathcal{O}(n\lambda^2\ell^2\log\ell)$</td></tr>
<tr><td>$\mathcal{O}(\ell^\epsilon)$</td><td colspan="2">$\mathcal{O}(n^{1+\epsilon}\lambda^2\ell^2\log\ell)$</td></tr>
</table>

The protocol can naturally be extended to an unbalanced Approx-PSI, where one input set is significantly larger than the other, in a more efficient manner.

2 High-Level Overview of our Approach

In this section, we provide an informal overview of our approximate PSI protocol, starting with the protocol for Hamming distance. Inspired by the DA-PSI for the same distance metric in [CFR23], our approach involves securely comparing the Hamming distance between two binary strings and applying this comparison for each element pair across two input sets. In [CFR23], the authors introduce a subprotocol that securely compares the Hamming distance between two elements, with communication complexity depending only on the threshold d and the security parameter, not the element size ℓ. Despite this efficiency, executing the subprotocol across all n^2 pairs results in quadratic communication and computation. Circumventing this quadratic complexity is challenging in standard DA-PSI, given that the match count can reach n^2.

In order to overcome this limitation, we consider the following primary aspects:

Input Restriction. We limit the potential inputs to those resulting in at most a linear number of matches. The setting for our Approx-PSI effectively translates to scenarios where each element in one input set corresponds to just one element in its counterpart. To enforce this condition, we impose a structure for all elements in each input set: every pair of elements should be either near or far apart. This reflects real-world scenarios where legitimate texts or numbers differ significantly, while their errors deviate by only a few characters or digits. The parties are required to consolidate their elements, ensuring each cluster is represented only once within their input set. This setting caps the match count at a linear number of matches, aligning with our goal for linear communication.

Near Linear Matching. Despite the linear match limit, in order to find them, the protocol needs to compare every possible pair still results in quadratic communication and computation. Our strategy incorporates an additional phase to eliminate non-matching pairs, utilizing the random projection technique in [KOR98] to minimize comparisons to a near-linear count.

First, both parties jointly sample a random subset of positions. Then, each party calculates a set of element projections based on these agreed positions. Matched pairs are more likely to have identical projections, unlike elements that are far apart. These projections undergo an exact match evaluation using traditional PSI for security, reducing the problem of near-matching to exact matching, which can be securely and efficiently computed using standard PSI.

If two vectors differ by few positions, the probability that none of those positions are chosen is high, leading to matching short vectors. Conversely, if the number of selected positions is too small, many vectors, even non-matching ones, may project or "collide" into identical vectors. This scenario increases the potential match count, leading to near-quadratic communication.

We meticulously adjust the position selection probability to minimize collisions while preserving actual matches. The probability can be amplified by repeatedly and independently sampling positions and computing intersections of projections. Repeating this process a logarithmic number of times ensures that our protocol finds all approximate matches with negligible probability of error. This approach reduces Approx-PSI to a logarithmic number of standard PSI computations.

Information Leakage. However, the intermediate steps of the aforementioned method disclose more about the distance between elements in the input sets than the intended Approx-PSI outcome should reveal. The process of projecting and comparing projections potentially leaks information, even with PSI. For instance, vectors with identical projections that fail the Hamming distance check might inadvertently reveal some bits of a party's vector to the other party.

To address this, both the PSI subprotocol and the secure Hamming distance comparison subprotocol must output secret shares of their respective results. We use secret-share versions of both PSI and the Hamming distance comparison test to hide these intermediate results. There are known PSI protocols that output secret shares, with prominent examples being circuit-based PSI protocols like those in [RS21,RR22]. While ready-to-use PSI protocols that output secret shares of results exist, efficient Hamming distance checks outputting secret shares remain unknown. The secure Hamming distance comparison subprotocol deployed in [CFR23] is not suitable to be compiled to output secret shares. The reason is because the subprotocol in [CFR23] inherently reveals both parties' inputs when matched, which forces their DA-PSI to output the result in pairs rather than only to the owner of each matched element. As our Approx-PSI may be customized to give the result to one party, we cannot follow their approach directly.

The secret-share Hamming distance comparison test can be constructed simply from garbled circuit. The resulting subprotocol is efficient for small and median size elements. For large elements (8000 bits or more), the length-independent comparison test can be constructed by combining the ideas from [CFR23, GS19, KMWF07]. We use [CFR23] as a starting point, representing binary vectors as subsets of finite field elements, whose Hamming distance corresponds to the size of set difference. These subsets can be further encoded as matrices whose subtraction corresponds to the set difference, using the idea in [GS19]. Moreover, the dimension of the matrices corresponds to the threshold value and the size of the set difference can be tested if above or below the threshold from the determinant of the matrix difference. The parties can jointly and securely compute the determinant using additive homomorphic encryption in [KMWF07]. Further modification of the homomorphically encrypted output gives the secret shares of the result. Finally, we utilize standard secret-sharing scheme operations for addition and multiplication to manipulate the secret shares between steps of our protocol.

Other Distance Metrics. We then combine the Approx-PSI protocol for Hamming distance with low distortion embedding from edit distance by [OR07], Euclidean distance by [DM21] and angular distance by [DS18] to construct Approx-PSI protocols for these distance metrics. We take advantage of the gap to guarantee that the pairs of elements that are near or far apart remain so after the embedding. Using the relationship between Euclidean distance, angular distance and cosine similarity, we also obtain the Approx-PSI protocol for cosine similarity.

Euclidean Distance. As the Euclidean distance is one of the most used distance metrics, there is a long line of work on embedding Euclidean distance or its related metrics such as cosine distance and angular distance into the Hamming distance. The ideas follow from the Johnson-Lindenstrauss lemma [JL84]. The recent line of work [PV14, OR15, HS20, DS20, DM21] gives low distortion of balls or the unit sphere centered at the origin in $\mathbb{R}^N$ with Euclidean metric or angular metric into binary string with Hamming distance. We construct the Approx-PSI using the similar method as the one with the edit distance.

Angular Distance and Cosine Similarity. Since cosine similarity, cosine distance and angular distance can be computed from the Euclidean distance, the Approx-PSI for Euclidean distance naturally gives the Approx-PSI for these metrics as well. Many Johnson-Lindenstrauss-styled embeddings are done directly for the angular distance [PV14, OR15]. We obtain the Approx-PSI for the angular distance from these direct embeddings. This implicitly give us a second way to reach the cosine similarity as it has tighter connection to the angular distance. The result has better parameters compared to converting from Euclidean distance one.

Edit distance. Ostrovsky and Rabani [OR07] showed how to embed edit distance metric in Hamming distance metric with bounded distortion. Such embedding could not be used to construct standard DA-PSI as the distortion could turn a match into a non-match and vice versa. However, the Approx-PSI tolerates some degree of distortion. Thus, we can embed elements in Hamming distance metric, securely compute matches and look up the original elements in the result.

Private Approximate Membership Computation. A natural approach to construct a PAMC protocol is to compare the client's query to each database element to determine whether they are close with respect to a given distance metric. This leads to $\mathcal{O}(n)$ secure comparisons, and consequently $\tilde{\mathcal{O}}(n)$ communication and computation, where n is the size of the database.

To overcome the linear communication, we follow the approach in [KM21], where the server organizes the database elements into $\mathcal{O}(n)$ buckets, labeled by their hashes. Instead of using a complex perceptual hashes as in [KM21], we use simple projection, as this allows us to mathematically analyze the probability that the query is mapped to the same bucket as its matched, rather than relying on experimental analysis. The client computes the same hash on its query and executes a private information retrieval (PIR) protocol to obtain the homomorphically encrypted bucket that the query belongs to. The client masks the encrypted bucket using homomorphic addition and sends it back to the server. The server and client then securely compare the masked database elements in the bucket and the masked query. This results in a small number of comparisons.

The entire process is repeated $k = \tilde{\mathcal{O}}(n^{\epsilon})$ times to ensure that the query is in the same bucket as its match at least once, except with negligible probability. We further optimize the protocol by combining the (encrypted) buckets across all repetitions into a single database for the PIR. This method allows our protocol to use additive homomorphic encryption independently of the PIR, with any efficient PIR protocol such as one in [LMRSW24].

3 Preliminaries

We denote the set $\{1, 2, \ldots, n\}$ as $[n]$. Let $x \in \{0, 1\}^*$. The length of x is denoted by $|x|$. For $i \in [|x|]$, we denote the ith character in x by x_i. We use λ to represent the (statistical) security parameter unless specified otherwise. We follow the standard definitions of negligible functions and computational indistinguishability [GM84]. The probability of an event A over random coins r is denoted by $\Pr_r[A]$, and simply $\Pr[A]$ when r is unspecified. The expectation of a random variable X is denoted by $\mathbb{E}[X]$. For a finite set S, we denote a uniformly random selection of a from S by $a \leftarrow S$. For a randomized algorithm A, let $A(x; r)$ represent running A on input x with random coins r. If r is chosen uniformly at random and the output is y, we denote this as $y \leftarrow A(x)$. When we write $\log x$, we refer to the logarithm based 2 of x.

3.1 Approximate PSI

We consider the setting of two parties with input sets A, B whose elements are drawn from a subset $\mathcal{S}$ of the universe $\mathcal{U}$, equipped with a distance metric $\delta : \mathcal{U} \times \mathcal{U} \to \mathbb{R}_{\geq 0}$. The subset $\mathcal{S}$ has the property that any pair of elements must be either near or far from each other. More specifically, for any elements $a, b \in \mathcal{S}$, either $\delta(a, b) \leq d$ (called *matched*, close, or near) or

$\delta(a,b) \geq td$ (called *non-matched*, or far) for some integer $d > 0$ and $t > 1$. We call d the *threshold* and t the *gap*.

The approximate PSI (Approx-PSI) functionality is defined in Figure 1. The goal of the Approx-PSI is to find pairs of elements, one from each input set, that are near, i.e., approximate matches. We allow three possibilities for the output: only one party receives their matched elements; each party receives matched elements in their respective set; or both parties receive a set of matches pairs.

$$\mathcal{F}^{\mathcal{S}}_{\mathsf{Approx-PSI}}$$

Parameters. upper bound on input size n, threshold d

Functionality.

1. Upon receiving a message (inputS, A) from the sender with $A \subseteq \mathcal{S}$ and $|A| \leq n$, store A; otherwise, ignore the message.

2. Upon receiving a message (inputR, B) from the receiver with $B \subseteq \mathcal{S}$ and $|B| \leq n$, store B; otherwise, ignore the message.

3. If both A and B are stored, compute $M = \{(a,b) \in A \times B : \delta(a,b) \leq d\}$; otherwise, abort. Let $M_A = \{a : (a,b) \in M\}$ and $M_B = \{b : (a,b) \in M\}$.

4. Send M_B to the receiver. Optionally, send M_A to the sender, or send M to both parties.

Figure 1: Ideal functionality for approximate private set intersection

We note that any matched elements can be grouped by the following lemma.

Lemma 3.1. *Suppose $t > 3$ and $A, B \subseteq \mathcal{S}$. Then $a \in A$ and $b \in B$ are matched if and only if any $a' \in A$ near a and any $b' \in B$ near b are also matched. In particular, if A and B have no close elements within each set, then any $a \in A$ is matched with at most one $b \in B$ and vice versa.*

Proof. Suppose $A, B \subseteq \mathcal{S}$, Suppose also that $a \in A$ and $b \in B$ are matched, i.e., and $\delta(a,b) \leq d$. Let $a' \in A$ and $b' \in B$ satisfying $\delta(a,a') \leq d$ and $\delta(b,b') \leq d$. If a,b are matched, by the triangle inequality,

$$\delta(a',b') \leq \delta(a,a') + \delta(a,b) + \delta(b,b') \leq d + d + d = 3d < td.$$

Hence, a', b' are not far, and must be matched by the structure of $\mathcal{S}$. Similarly, if a,b are non-matched, by the triangle inequality,

$$td \leq \delta(a,b) \leq \delta(a,a') + \delta(a',b') + \delta(b,b') \leq \delta(a',b') + 2d.$$

Thus, $\delta(a',b') \geq (t-2)d > d$. Hence, a', b' are not close, and must be non-matched.

Now assuming that A and B have no close elements within each set. If $a \in A$ are matched with both $b, b' \in B$, then b, b' are close, contradicting the assumption. Thus, a is matched with at most one element in B. By symmetry, this property holds for $b \in B$ as well. □

When $t > 3$, this lemma implies that being matched or non-matched can be transferred between close elements in $\mathcal{S}$. Thus, we may group all $a' \in A$ within distance d from a into one class represented by a. Whenever, a and b are matched (as output by $\mathcal{F}^{\mathcal{S}}_{\mathsf{Approx-PSI}}$), then every a' in the same class are matched to b as well. We call the process of removing all $a' \in A$ within distance d from a representative $a \in A$ *clustering*, and adding the a' back if a is matched with some $b \in B$ *declustering*. By performing clustering and declustering in the beginning and at the end of an Approx-PSI protocol with semi-honest parties, we may further assume that elements of A are far apart, and so are elements of B.

3.2 Distance Metrics

In this work, we consider two distance metrics for binary strings: Hamming distance and edit distance. We let ℓ denote the length of the string, i.e., the universe $\mathcal{U} = \{0,1\}^\ell$.

For $x, y \in \{0,1\}^\ell$, the *Hamming distance* between x and y, denoted $\mathcal{H}(x,y)$, is the number of positions $i \in [\ell]$ such that $x_i \neq y_i$. We also denote $\mathcal{H}(x) = \mathcal{H}(x,0)$, the *Hamming weight* of x. The *edit distance* (also known as Levenstein distance) between x and y, denoted $\mathrm{ed}(x,y)$, is the minimum number of insert, delete and substitute operations (one character at a time) needed to convert x to y.

In many practical contexts, the distance metric most frequently employed to measure the separation between two points or vectors in space is the Euclidean distance. When the vectors are normalized, we can consider them on a unit sphere and measure the shortest path on the sphere connecting two vectors. This distance is called *angular distance*. The Euclidean distance between two vectors can be computed from their dot product or angular distance, and vice versa. We refer to Appendix A for their formulas and relationship.

4 Building Blocks: Secret-Shared Operations

Our construction requires several operations whose outputs are secret shared between two parties to hide the intermediate results. In particular, the building blocks are secret-shared PSI, secret-shared Hamming distance comparison test, and operations on secret-shared data including scalar-vector multiplication.

4.1 Secret Sharing

In this work, we consider only a 2-out-of-2 secret sharing for binary strings and elements of a finite field. For a secret $s \in \mathcal{S}$, secret sharing of s are denoted $\mathsf{Share}(s) \to ([s]_0, [s]_1)$ (or $[s]_S, [s]_R$ when the shares belong to the sender and the receiver in a 2-party protocol, respectively) where for any $s, s' \in \mathcal{S}$ and $i \in \{0,1\}$, $\{[s]_i : \mathsf{Share}(s) \to ([s]_0, [s]_1)\} = \{[s']_i : \mathsf{Share}(s') \to ([s']_0, [s']_1)\}$. The secret can then be reconstructed by $\mathsf{Recon}([s]_0, [s]_1) = s$. When it is clear from context, we may omit the subscript and only denote the shares by $[s]$ when each party operates on their own share. We also denote the process when a party

$\mathcal{F}_{\mathsf{ssPSI}}$

Parameters. element set $\mathcal{U}$, payload set $\{0,1\}^\sigma$, upper bound on set size m, output size $m' > m$

Functionality.

1. Upon receiving a message $(\mathsf{inputS}, \tilde{A})$ from the sender where $\tilde{A} = \{(a_i, \tilde{a}_i)) : a_i \in \mathcal{U}, \tilde{a}_i \in \{0,1\}^\sigma\}_{i\in[m_A]}$, $m_A \leq m$, store $\tilde{A}$.

2. Upon receiving a message $(\mathsf{inputR}, \tilde{B})$ from the receiver where $\tilde{B} = \{(b_i, \tilde{b}_i)) : b_i \in \mathcal{U}, \tilde{b}_i \in \{0,1\}^\sigma\}_{i\in[m_B]}$, $m_A \leq m$, store $\tilde{B}$.

3. If both $\tilde{A}$ and $\tilde{B}$ are stored, compute $\pi = \mathsf{Reorder}([m])$ such that

$$z_j = \begin{cases} (\tilde{a}_i \| \tilde{b}_{j'}) & \text{if } \exists a_i \in A, \text{ s.t. } a_i = b_j \\ 0^{2\sigma} & \text{otherwise} \end{cases}$$

for $j' = \pi(j)$, $j \in [m]$. Compute $\mathsf{Share}(z) \rightarrow ([z]_S, [z]_R)$. Send $[z]_S$ to the sender and $[z]_R$ to the receiver.

Figure 2: Ideal functionality for secret-shared PSI

sends (and authenticates, in the malicious setting) their share to the other party to allow the later party to reconstruct the secret as *opening*.

In the semi-honest setting and $\mathcal{S} = \{0,1\}^\ell$, $\mathsf{Share}(s)$ simply uniformly samples $[s]_0, [s]_1 \in \{0,1\}^\ell$ conditioned on $[s]_0 \oplus [s]_1 = s$ (replacing $\oplus$ by addition in mod p for the finite field $\mathbb{F}_p$.) The maliciously secure variant can be done using more complicated authenticated secret sharing [NNOB12, FKOS15].

4.2 Secret-shared PSI

Two-party PSI protocols can be constructed from various techniques resulting in different performance and properties [PRTY19, DPT20, CM20, MPR+20, RT21, RS21, GPR+21, CILO22, RR22, BPSY23]. In this work, we focus on PSI Payload variant, where each party's input consists of two sets, an elements set for intersection and a set of values associated to the elements. The output of the protocol also contains the associated values of the elements in the intersection. These associated values are called "payloads." Most PSI protocols can be configured to transfer the payloads with differing efficiency [IKN+20, RS21, CILO22, RR22].

In our Approx-PSI construction, the output of PSI Payload should be secret shared between parties. Such protocols are often constructed using circuit-based PSI techniques such as the circuit-based variant of the PSI protocol in [RS21]. They call the variant *circuit PSI*. We simplify the variant to better serve our purpose in Figure 2. See Appendix B.1 for the ideal functionality in [RS21].

$\mathcal{F}_{\mathsf{ssHamCom}}$

Parameters. element size ℓ, threshold Hamming distance d

Functionality.

1. Upon receiving a message $(\mathsf{inputS}, [a]_S, [b]_S)$ from the sender where $[a]_S, [b]_S \in \{0,1\}^\ell$, store $([a]_S, [b]_S)$.
2. Upon receiving a message $(\mathsf{inputR}, [a]_R, [b]_R)$ from the receiver where $[a]_R, [b]_R \in \{0,1\}^\ell$, store $([a]_R, [b]_R)$.
3. If both $([a]_S, [b]_S)$ and $([a]_R, [b]_R)$ are stored, compute $a = \mathsf{Recon}([a]_S, [a]_R)$ and $b = \mathsf{Recon}([b]_S, [b]_R)$. Let $out = 1$ if $\mathcal{H}(a, b) \leq d$ and $out = 0$ otherwise. Send $[out]_S$ and $[out]_R$, secret shares of out to each party.

Figure 3: Ideal functionality for secret-shared Hamming distance comparison test

Here, the intersection of m-element sets is mapped to a slightly larger set $m' > m$ (concretely $m' \approx 1.27m$ in [RS21]). In the original version each party also learns a secret shared bit indicating if each element is in the intersection, thus hiding even the intersection size. In our work, we only need the protocol to output the shares of the payloads, and not the actual PSI elements, in any order. Since the construction in [RS21] executes a garbled circuit in the last step, we simply modify the circuit to only output the payload parts.

The protocol in [RS21] is already quite efficient, and can be further improved using more recent oblivious key-value stores (OKVS) in [RR22, BPSY23] and VOLE setup [BCG$^+$22] resulting in a high-performance protocol. The communication and computation cost of the secret-shared PSI when instantiated with the protocol above is linear in the number of elements $\mathcal{O}(\lambda n)$ [RS21].

4.3 Secret-shared Hamming distance comparison test

Similar to the DA-PSI for Hamming distance in [CFR23], our Approx-PSI protocol utilizes a subprotocol for computing the Hamming distance. Unlike to one in [CFR23] that outputs both input binary strings to both parties when matched, our protocol takes secret shares of the strings as input, and outputs secret share of a single bit indicating whether the Hamming distance between the two inputs is within a certain threshold or not. We define the functionality $\mathcal{F}_{\mathsf{ssHamCom}}$ in Figure 3.

We note that the secret-share inputs of $\mathcal{F}_{\mathsf{ssHamCom}}$ can be added locally to obtain different inputs with the same Hamming distance. More specifically, $\mathcal{H}(a, b) = \mathcal{H}(a \oplus b)$ where $[a \oplus b]$ can be locally computed from $[a]$ and $[b]$ for additive secret sharing. Thus, we only need to construct one with secret shares output. However, we cannot simply convert the protocol in [CFR23] in the final step to output secret share as their immediate results reveal other party's input if they are matched.

The simplest way to realize this functionality is to through garbled circuit. However, the communication complexity of the resulting protocol will depend on the length ℓ. To obtain length-independent communication as in [CFR23], we consider a more complicated technique described in Appendix D. The resulting protocol has communication complexity $\tilde{\mathcal{O}}(d^2)$ and computation complexity of $\tilde{\mathcal{O}}(\ell + d^2)$ similar to the protocol in [CFR23].

4.4 Secret-shared vector multiplication

The final functionality used in our work is the secret-shared vector multiplication described in Figure 4. Both parties hold secret shares of a bit c and a binary vector $\vec{v}$, and would like to compute the product $c\vec{v}$, and output as secret shares between two parties.

$\mathcal{F}_{\mathsf{ssVMult}}$

Parameters. element size ℓ

Functionality.

1. Upon receiving a message $(\mathsf{inputS}, [c]_S, [\vec{v}]_S)$ from the sender, store $([c]_S, [\vec{v}]_S)$.
2. Upon receiving a message $(\mathsf{inputR}, [c]_R, [\vec{v}]_R)$ from the receiver, store $([c]_R, [\vec{v}]_R)$.
3. If both $([c]_S, [\vec{v}]_S)$ and $([c]_R, [\vec{v}]_R)$ are stored, compute $c = \mathsf{Recon}([c]_S, [c]_R)$ and $\vec{v} = \mathsf{Recon}([\vec{v}]_S, [\vec{v}]_R)$. If $c \in \{0,1\}$ and $\vec{v} \in \{0,1\}^\ell$, compute $\vec{out} = c\vec{v}$, and $\mathsf{Share}(\vec{out}) \to ([\vec{out}]_S, [\vec{out}]_R)$. Send $[\vec{out}]_S$ and $[\vec{out}]_R$ to the sender and the receiver, respectively.

Figure 4: Ideal functionality for Secret-Share Vector Multiplication

The functionality can be implemented using standard techniques for multiplication on secret shares such as using setup triples using OT or HE preprocessing in the semi-honest model, or using the standard frameworks for maliciously secure secret-shared operations such as TinyOT [NNOB12], Tinier [FKOS15] and MD-SPDZ [Kel20] in the malicious model. See Appendix E for a concrete protocol in the semi-honest model using OT. Using OT extension techniques, the (amortized) communication and computation is $\mathcal{O}(\ell)$ and $o(1)$, respectively, per multiplication. The OT can also be setup offline to improve online efficiency.

5 Gaining intuition about the problem: false starts

The starting point of our constructions is the Approx-PSI protocol for Hamming distance. We begin by constructing an insecure version using the idea from [KOR98]. First, we randomly select a subset of position $I \subseteq [\ell]$ such that each $i \in [\ell]$ is chosen independently with probability p. We project every element of $A, B \subseteq \{0,1\}^\ell$ onto the position in I. We denote the projection of individual element $a \in \{0,1\}^\ell$ by $a_I = (a_{i_1}, a_{i_2}, \ldots, a_{i_{|I|}})$ for

$I = \{i_1, \ldots, i_{|I|}\}$, and denote the sets of projections $A_I = \{a_I : a \in A\}$, and B_I defined similarly. We also denote the reverse map by $I_A^{-1}(c) = \{a \in A : a_I = c\}$ for $c \in \{0,1\}^{|I|}$. When it is clear from context, we may drop the set to $I^{-1}(a_I)$. For each $c \in A_I \cap B_I$, we can compute the probability that $\mathcal{H}(a,b) \le d$ when $c = a_I = b_I$. By repeatedly sampling I, independently, and merging all pairs (a,b) from each projection, the probability that we fail to find any matched pairs is negligible.

It is important to note that the goal of [KOR98] is to efficiently approximate the Hamming distance between vectors, without considering security. As a result, this version of the protocol is not secure, even when using PSI to compute $A_I \cap B_I$. If a projected vector is in the intersection, it reveals that both parties share the same bits at the positions in I, even if the vectors are not matched. We will address this leakage in the final version of the protocol. For now, we focus on analyzing the correctness.

For $i \in [k]$, where k is the number of repeats, the parties choose $I_i \subseteq [\ell]$ by choosing each position independently with probability p. Let $q = (1-p)^d$.

Lemma 5.1. *When $k \ge (\ln 2)\frac{\log n + \lambda}{q}$, the probability that there exists $a \in A, b \in B$ such that $\mathcal{H}(a,b) \le d$ but $a_{I_i} \ne b_{I_i}$ for all $i \in [k]$, is negligible.*

Proof. Let $a \in A, b \in B$ such that $\mathcal{H}(a,b) \le d$. We have

$$\Pr[a_I \ne b_I] = 1 - \Pr[a_I = b_I] \le 1 - (1-p)^d = 1 - q \le e^{-q}.$$

Then

$$\Pr[a_{I_i} \ne b_{I_i} \forall i \in [k]] \le e^{-kq},$$

and

$$\Pr[a_{I_i} \ne b_{I_i} \forall i \in [k], \exists (a,b) \in A \times B, \mathcal{H}(a,b) \le d] \le ne^{-kq}$$

by the union bound on A as each $a \in A$ has at most one b that is matched. When $kq \ge (\ln 2)(\log n + \lambda)$, we have negligible probability. □

This lemma guarantees that any matched pair will be found from at least one of the projections as long as the protocol repeats sufficiently many times, which is only $\mathcal{O}(\log n + \lambda)$ when q is constant.

Now we analyze the probability for non-matched pairs. Our goal is rule out as many non-matched pairs as possible to ensure that the number of remaining pairs to be checked is near-linear. To increase the probability that the pair a, b such that $\mathcal{H}(a,b) \ge td$ are not projected to the same vector, we first consider the case when $t = \log n$. In other words, for any $a \in A$ and $b \in B$, either $\mathcal{H}(a,b) \le d$ or $\mathcal{H}(a,b) \ge td$ where $t = \log n$. We will ease this assumption in the later sections.

5.1 First attempt (that does not work for most parameters)

We first observe that, under the condition $\lambda = \mathcal{O}(\log n)$, any pair (a,b) with $\mathcal{H}(a,b) \ge td$ will not be projected to the same element with high probability.

Lemma 5.2. *Assuming $\lambda = \mathcal{O}(\log n)$ and $\frac{1}{q} = 2^{\lambda/\log n + 2}$, the probability that there exists a, b such that $\mathcal{H}(a,b) \ge td$ and $a_{I_i} = b_{I_i}$ for some $i \in [k]$ is negligible.*

Proof. Let $a \in A, b \in B$ such that $\mathcal{H}(a,b) \geq td$. We have

$$\Pr[a_I = b_I] \leq (1-p)^{td} = q^t.$$

When $t = \log n$, we have $q^t = q^{\log n} = 2^{\log q \log n} = n^{\log q}$. Then

$$\Pr[a_{I_i} = b_{I_i} \exists i \in [k]] \leq kn^{\log q},$$

and

$$\Pr[a_{I_i} = b_{I_i} \exists i \in [k], \exists (a,b) \in A \times B, \mathcal{H}(a,b) \geq td]$$
$$\leq n^2 kn^{\log q} = \frac{k}{n^{\log(1/q)-2}}.$$

Assuming $\lambda = \mathcal{O}(\log n)$, we may choose $\frac{1}{q} = 2^{\frac{\lambda}{\log n}+2} = \mathcal{O}(1)$. Then $k = \mathcal{O}(\log n) = \mathcal{O}(\lambda)$, and the above probability is $\frac{k}{2^\lambda}$, which is negligible. □

In this case, by projecting A and B onto the coordinates in I_i for $i \in [k]$ with probability $p = 1 - q^{\frac{1}{d}} = 1 - 2^{-\frac{\frac{\lambda}{\log n}+2}{d}}$, and computing the intersection $A_I \cap B_I$, each party learns the elements that matched with another party's elements with overwhelming probability without additional direct comparison. We could construct a secure Approx-PSI protocol by merging the result of the intersection from each round.

The assumption $\lambda = \mathcal{O}(\log n)$ is probable in some cases as λ is a statistical security parameter. For example, we may choose $\lambda = 40$ and $n = 2^{20}$, which gives $q = \frac{1}{16}$. When $d = 4$, each position is chosen with probability 0.5. However, in the general case when λ is much larger than $\log n$, the number of rounds k will be exponential in $\frac{\lambda}{\log n}$ as k is proportional to $\frac{1}{q}$, so is the communication from computing the intersections. Thus, we need a different method to separate the non-matched pairs.

In term of security, we note that when I is jointly chosen uniformly, and the intersection is computed using a PSI protocol, the resulting protocol is secure as the intermediate result C_j can be computed from the projection of each party's output.

5.2 Second attempt (that is too complicated to obtain security)

Now we consider a more complex solution when $\lambda \gg \mathcal{O}(\log n)$. In this case, projections alone cannot completely rule out false positives, where $\mathcal{H}(a,b) \geq td$ but project to the same vector, while keeping the number of rounds poly-logarithmic. Each party needs to run a 2PC protocol to compare every pair of $a \in A$ and $B \in B$ (such as the one in [CFR23]) that project to the same c. This process is repeated k times with independently sampled I.

Unfortunately, this method may not result in a linear number of comparisons, as the number of possible pairs for each projection can be super-linear. To resolve this, parties must select a "good" projection that results in a linear number of comparisons. However, revealing the "good" projection also reveals the structure of the set. Thus, we consider a special-purpose PSI that outputs $\perp$ (privately as secret shares) when the projection produces too many collisions.

A "good" projection $I \subseteq [\ell]$ is defined such that, for all $a \in A$, $|I^{-1}(a_I)| \leq \tau$ for a fixed constant $\tau \geq 2$. The PSI is modified to a special-purpose private *multiset* intersection that only outputs when I is good. Instead of having I as an additional input to PSI, both parties' input sets must be a multiset, where each element is associated with an integer representing its repetitions. In this case, the number associating to a_I is $|I^{-1}(a_I)|$. The protocol may only output $\perp$ when there exists a_I in the intersection where $|I^{-1}(a_I)| \geq \tau$.

Both parties will perform the special-purpose PSI on the projected sets for k rounds. We can construct a secure Approx-PSI protocol by privately compare all pairs that project to the same elements in the output of the special-purpose PSI in each round, and then merging the results from all rounds.

This solution requires a special-purpose variant of PSI for multisets. While theoretically feasible, constructing it efficiently may be challenging. Additionally, intermediate results, especially from bad projections, may reveal information not inferable from the final result. Thus, the output may need to be secret-shared, further complicating the construction.

6 Approx-PSI Protocol

In this section, we build on the previous two ideas to construct an efficient and secure approximate PSI protocol. The independently repeating projection from the first idea already identifies matches, provided the number of rounds is sufficiently large. The "good" or "bad" projection from the second idea, however, can be improved to ensure that the projections are not dropped entirely. Therefore, we redefine a "bad" projection as the union of conditions from both the first and second ideas, where the latter applied for both sets.

Specifically, for a close pair (a, b), we call $I \subseteq [\ell]$ *bad* for (a, b) if one of the following holds: (1) $a_I \neq b_I$, (2) $|I^{-1}(a_I)| \geq 2$, or (3) $|I^{-1}(b_I)| \geq 2$. When a projection is bad for (a, b), a_I is dropped from A_I if condition (2) holds, or b_I is dropped from B_I if condition (3) holds. In case of (1), they do not appear in the intersection anyway (unless as a different pair, like (a, b') or (a', b)). This method allows other pairs to proceed and be discovered without dropping the entire projection, as was done in the second idea.

For $i \in [k]$, where k will be determined later, we choose $I_i \subseteq [\ell]$ by choosing each position to be in I_i independently with probability p. Both parties project their sets to coordinates in I_i, denoted A_{I_i} and B_{I_i}, respectively. Let $q = (1-p)^d$. For $a \in A, b \in B$ such that $\mathcal{H}(a, b) \leq d$, we define

$$\mathsf{BAD}(a, b) = \{I \subseteq [\ell] : a_I \neq b_I \text{ or } |I_A^{-1}(a_I)| \geq 2 \text{ or } |I_B^{-1}(b_I)| \geq 2\}$$

We assume that for any $a \in A$ and $b \in B$, either $\mathcal{H}(a, b) \leq d$ or $\mathcal{H}(a, b) \geq td$, and for any $a, a' \in A$ and $b, b' \in B$, $\mathcal{H}(a, a') \geq td$ and $\mathcal{H}(b, b') \geq td$.

Lemma 6.1. *When $k \approx \frac{(nt)^{\frac{1}{t-1}}(\lambda + \log n)}{1 - \frac{1}{t}}$, the probability that there exists $a \in A, b \in B$ such that $\mathcal{H}(a, b) \leq d$ but $I_i \in \mathsf{BAD}(a, b)$ for all $i \in [k]$ is negligible.*

Proof. Let $a \in A, b \in B$ such that $\mathcal{H}(a, b) \leq d$. We have

$$\Pr[a_I \neq b_I] = 1 - \Pr[a_I = b_I] \leq 1 - (1-p)^d = 1 - q.$$

Fix an element $a \in A$ that is projected to a_I. Let $X_{a'}$ be an indicator that $a'_I = a_I$. Then

$$\mathbb{E}[|I^{-1}(a_I)|] = \sum_{a' \in A} \mathbb{E}[X_{a'}] = n(1-p)^{td} = nq^t.$$

By the Markov's inequality,

$$\Pr[|I^{-1}(a_I)| \geq 2] \leq \frac{nq^t}{2}.$$

Similarly,

$$\Pr[|I^{-1}(b_I)| \geq 2] \leq \frac{nq^t}{2}.$$

By the Union bound, we have the probability that I is bad for (a, b) is at most

$$1 - q + nq^t.$$

Now we consider this probability as a function of q, $f(q) = 1 - q + nq^t$. When $t > 1$, the function takes the minimum value when $f'(q) = -1 + ntq^{t-1} = 0$. Solving the equation above gives $q = \frac{1}{(nt)^{\frac{1}{t-1}}}$. In this case, the above probability becomes

$$\alpha(n, t) = 1 - \frac{1}{(nt)^{\frac{1}{t-1}}} + \frac{n}{(nt)^{\frac{t}{t-1}}} = 1 - \frac{\beta(t)}{n^{\frac{1}{t-1}}}$$

where $\beta(t) = \frac{1}{t^{\frac{1}{t-1}}}\left(1 - \frac{1}{t}\right)$. Then the probability that I_i are bad for all $i \in [k]$ is at most $\alpha(n, t)^k$. Thus, the probability that for some close pair (a, b), all I_i's are bad is at most

$$n\alpha(n, t)^k = \frac{1}{2^\lambda}$$

when $k = \frac{\lambda + \log n}{\log(1/\alpha(n,t))}$. Using an approximation $\log(1 - x) \approx -x$, we have

$$k \approx \frac{n^{\frac{1}{t-1}}(\lambda + \log n)}{\beta(t)} = \frac{(nt)^{\frac{1}{t-1}}(\lambda + \log n)}{1 - \frac{1}{t}}$$

□

We note that the second and third conditions for BAD imply that the projections are sets, not multisets, and the number of pairs in the intersection is at most n. This means the number of comparisons is at most nk.

Now we analyze this result for different asymptotic cases of t.

- When $t = \mathcal{O}(\log n)$:
  $\log(nt)^{\frac{1}{t-1}} = \mathcal{O}\left(\frac{\log n + \log\log n}{\log n - 1}\right) = \mathcal{O}(1)$. Thus, $k = \mathcal{O}(\lambda + \log n)$.
- When $t = \mathcal{O}(\log\log n)$:
  $\log(nt)^{\frac{1}{t-1}} = \mathcal{O}\left(\frac{\log n + \log\log\log n}{\log\log n - 1}\right) = \mathcal{O}(\log n / \log\log n)$.
  Thus, $k = \mathcal{O}(n^{\frac{1}{\log\log n}}(\lambda + \log n)) = n^{o(1)}(\lambda + \log n)$.
- When $t = \mathcal{O}(1)$:
  $k = \mathcal{O}(n^{\frac{1}{t-1}}(\lambda + \log n))$.

```
Algorithm 1: Approx-PSI
Input  : Sets $A, B \subseteq \{0,1\}^\ell$, $|A|, |B| \leq n$
Output: $\{a \in A : \exists b \in B, \mathcal{H}(a,b) \leq d\}$ and $\{b \in B : \exists a \in A, \mathcal{H}(a,b) \leq d\}$
Each party replaces their input set by a representation of each cluster. We still
  denote their clustered inputs by $A$ and $B$ ;
for $j = 1$ to $k$ do
    The parties jointly sample $I_j \subseteq [\ell]$ such that each $i \in [\ell]$ has probability
      $p = 1 - \frac{1}{(nt)^{\frac{1}{d(t-1)}}}$ to be in $I_j$;
    The parties project every element in their sets into coordinates in $I_j$. If more
      than one element shares the same projection, randomly pick one of them. The
      original elements are attached to its projection as payload. The projection sets
      are denoted as $\tilde{A}_{I_j} = \{(a_{I_j}, a) : a \in A\}$ and $\tilde{B}_{I_j} = \{(b_{I_j}, b) : b \in B\}$;
    Each party sends $\tilde{A}_{I_j}$ and $\tilde{B}_{I_j}$ to $\mathcal{F}_{\mathsf{ssPSI}}$ and receives shares of the intersection
      $[z] \in (\{0,1\}^{2|I|})^{n'}$ ;
    foreach $i \in [n']$ do
        Each party sends shares $[z_i]$ to $\mathcal{F}_{\mathsf{ssHamCom}}$ and receives shares of $[out_i]$. ;
        Both parties send the shares of $[out_i]$ and $[z_i]$ to $\mathcal{F}_{\mathsf{ssVMult}}$, and obtains shares
          $[\tilde{z}_i]$;
    end
    Each party stores all shares of $[\tilde{z}_i]$ in $\tilde{Z}_j$ (separately as $\tilde{Z}_j^A$ and $\tilde{Z}_j^B$)
end
For each $j \in [k]$ and for each $[\tilde{z}] = ([a], [b]) \in \tilde{Z}_j$, open $[a]$ to the sender and $[b]$ to
  the receiver; let $A'_j$ and $B'_j$ denoted the opened values ;
The party computes $\{a : a \neq 0^\ell \in A'_j, \exists j \in [k]\}$ and $\{b : b \neq 0^\ell \in B'_j, \exists j \in [k]\}$, and
  outputs the elements in the set and their cluster in the original input set ;
```

We obtain the following Approx-PSI protocol, described in Algorithm 1, assuming the following functionalities: the secret-shared PSI, the secret-shared Hamming distance comparison, and the secret-shared vector multiplication. The correctness of the protocol follows from Lemma 6.1.

Here we proof the security of our main protocol through simulator construction. Suppose an adversary corrupting the receiver. For each $j \in [k]$, the simulator jointly samples I_j and compute the projections honestly. It simulates $\mathcal{F}_{\mathsf{ssPSI}}$ receiving $\tilde{B}_{I_j}$ from the adversary and returning shares of 0. The simulator uses $\tilde{B}_{I_j}$ to reconstruct B' and sends to $\mathcal{F}^{\mathcal{S}}_{\mathsf{Approx-PSI}}$ and obtain the output $P \subseteq A \times B$. We may assume that the comparison is done after finishing all intersection first. The simulator simulates $\mathcal{F}_{\mathsf{ssHamCom}}$ by using P to compute *out* for each $b \in B'$ and simulates secure multiplication to create shares of correct output.

Theorem 6.2. *The protocol in Algorithm 1 is secure in the $\mathcal{F}_{\mathsf{ssPSI}}$, $\mathcal{F}_{\mathsf{ssHamCom}}$ and $\mathcal{F}_{\mathsf{ssVMult}}$ hybrid model.*

Proof. By symmetry, it suffices to construct a simulator $\mathcal{S}$ for the case when an adversary corrupting the receiver. For each $j \in [k]$, $\mathcal{S}$ follows the protocol to jointly sample I. It simulates $\mathcal{F}_{\mathsf{ssPSI}}$ to learn $\tilde{B}_I$ and outputs a secret share of $0^{2|I|n'}$, instead of z, to $\mathcal{S}$. $\mathcal{S}$ stores $\tilde{B}_I$. It also simulates $\mathcal{F}_{\mathsf{ssHamCom}}$ and $\mathcal{F}_{\mathsf{ssVMult}}$, and outputs a random secret share of 0 and $0^{2|I|n'}$, instead of *out* and $\tilde{z}$, to $\mathcal{S}$, respectively. After k rounds, $\mathcal{S}$ uses the stored $\tilde{B}_I$'s to reconstruct the receiver's set B^*. It sends B^* to $\mathcal{F}^{\mathcal{S}}_{\mathsf{Approx-PSI}}$ to learn the set of Hamming close pairs. Finally, $\mathcal{S}$ computes openings for each $[\tilde{z}] \in \tilde{Z}_I$'s that gives the Hamming close pairs for each I.

We prove the indistinguishability through the following hybrids:

- H_0: This is the real world interaction.

- H_1: Same as H_0 except $\mathcal{S}$ simulates the functionalities honestly. This hybrid is identical to H_0.

- H_2: Same as H_1 except $\mathcal{S}$ outputs shares of $0^{2|I|n'}$ instead of the correct output of $\mathcal{F}_{\mathsf{ssPSI}}$. It then replaces the adversary's input for $\mathcal{F}_{\mathsf{ssHamCom}}$ with the correct one from the adversary's input to $\mathcal{F}_{\mathsf{ssPSI}}$. This hybrid is identical to H_1 as single shares of $0^{2|I|n'}$ and z are identically distributed.

- H_3: Same as H_2 except $\mathcal{S}$ outputs shares of 0 instead of the correct output of $\mathcal{F}_{\mathsf{SS-Ham-Compare}}$. It then replaces the adversary's input for $\mathcal{F}_{\mathsf{ssVMult}}$ with the correct shares of the output of $\mathcal{F}_{\mathsf{ssHamCom}}$. This hybrid is identical to H_2 as a single share of 0 and out_i are identically distributed.

- H_4: Same as H_3 except $\mathcal{S}$ outputs random shares instead of the correct output of $\mathcal{F}_{\mathsf{ssVMult}}$. When $\mathcal{S}$ opens the shares in the final step, it opens to the correct outputs of $\mathcal{F}_{\mathsf{ssVMult}}$. This hybrid is identical to H_3 as each share of $\tilde{c}'$'s is uniformly random.

- H_5: Same as H_4 except $\mathcal{S}$ uses $\tilde{B}_I$ to reconstruct B^* from the payload and fill the rest with the special element $\perp$. It uses B^* to compute outputs in each step instead of B. Note that B^* may be smaller than B when there is an element that always collides with others when projected to coordinates in I in every round. We show that such elements occurs with negligible probability.

Claim. *Except with negligible probability, $B^* = B$.*

Proof. Clearly, $B^* \subseteq B$. We need to show that except with negligible probability, every element of B appears in B^*. Note that $b \in B$ does not appear in B^* only when its projection collides with another element in every round. In each round, the probability of such event is at most $\frac{nq^t}{2}$ by the proof of Lemma 6.1. Thus, the probability that $B^* \neq B$ is at most $\left(\frac{nq^t}{2}\right)^k$ which is negligible for the choice of k in the lemma.

□

- H_6: Same as H_5 except $\mathcal{S}$ sends B^* to $\mathcal{F}^{\mathcal{S}}_{\mathsf{Approx-PSI}}$ and no longer interact with the sender. It uses B^* to compute openings for each $\tilde{C}_I$'s.

□

Table 2: Communication and computation complexity of each subprotocol in Approx-PSI for Hamming distance.

Step	Subprotocol	Comm.	Comp.
1.	Clustering data	-	$\mathcal{O}(n(\log n)\ell)$
2.	Repeat k times		
2.1	Sampling projections	$\mathcal{O}(\ell)$	$\mathcal{O}(\ell)$
2.2	Projecting vectors	-	$\mathcal{O}(n\ell)$
2.3	SS-PSI	$\mathcal{O}(n(\ell+\lambda))$	$\mathcal{O}(n(\ell+\lambda))$
2.4	Repeat $n' = \mathcal{O}(n)$ times		
2.4.1	SS Ham. comp. test	$\mathcal{O}(\ell)$	$\mathcal{O}(\ell)$
2.4.2	SS Vector Mult.	$\mathcal{O}(\ell)$	$\mathcal{O}(\ell)$
2.5	Opening share	$\mathcal{O}(n\ell)$	-
3.	Combining result	-	$\mathcal{O}(nk\ell)$
	Total	$\mathcal{O}(nk(\ell+\lambda))$	$\mathcal{O}(nk(\ell+\lambda))$

6.1 Communication and Computation

In this section, we analyze the performance of our Approx-PSI protocol. The protocol consists of one instance of $\mathcal{F}_{\mathsf{ssPSI}}$ in each of the k rounds. n' instance of $\mathcal{F}_{\mathsf{ssHamCom}}$ and $\mathcal{F}_{\mathsf{ssVMult}}$ in each of the k rounds. We instantiate the functionalities used to construct the Approx-PSI in Algorithm 1 as we discussed in Section 4, and compute theoretical communication complexity and computation complexity of the protocol.

Clustering can be done via BK tree [BK73] or VP tree [Yia93]. The communication and computation of $\mathcal{F}_{\mathsf{ssPSI}}$ when instantiated with circuit PSI of [RS21], with or without later improvement in [RR22], is $\mathcal{O}(n(\ell+\lambda))$. Here the PSI output size is $n' = \mathcal{O}(n)$. The communication and computation of $\mathcal{F}_{\mathsf{ssHamCom}}$ instantiated using garbled circuit is $\mathcal{O}(\ell)$. The communication and computation of $\mathcal{F}_{\mathsf{ssVMult}}$ instantiated using OT as described in Appendix E are $\mathcal{O}(\ell)$ when amortized. Other subprotocols are simply sending data or local computations as shown in Table 2. We remark that replacing the secret-shared Hamming distance comparison test by ones with communication independent of ℓ such as the one in Appendix D does not improve the asymptotic complexity of the overall protocol.

Here, the number of rounds k depends on the gap t as proved in Lemma 6.1. We conclude the following corollary.

Corollary 6.3. *The protocol in Algorithm 1 when $\mathcal{F}_{SS-PSI}$,
$\mathcal{F}_{\mathsf{ssHamCom}}$ and $\mathcal{F}_{\mathsf{ssVMult}}$ are instantiated as described above has the communication and computation complexity $\mathcal{O}(\gamma(t)n^{1+\frac{1}{t-1}}(\lambda+\log n)(\ell+\lambda))$ where $\gamma(t) = \frac{t^{\frac{1}{t-1}}}{1-\frac{1}{t}}$.*

When $t = \log n$, $\frac{\log n}{\log\log n}$ or $\log\log n$, the above communication is $\mathcal{O}(n(\lambda+\log n)(\ell+\lambda))$, $\mathcal{O}(n\,\mathrm{polylog}(n)(\lambda+\log n)(\ell+\lambda))$ or $n^{1+o(1)}(\lambda+\log n)(\ell+\lambda)$, respectively.

7 Other Distance Metrics

We construct Approx-PSI for different distance metrics by embedding the set $(\mathcal{U}, \delta)$ into the set of binary strings equipped with the Hamming distance $(\{0,1\}^{\ell'}, \mathcal{H})$. This approach leverages the gap between matched and non-matched pairs, ensuring the two cases remain separate after the embedding. However, this method does not work for standard DA-PSI (where there is no gap, meaning $t = 1$), as the distance distortion from embedding could cause matched pairs to become non-matched pairs, or vice versa.

We focus on three main distance metrics: edit distance, Euclidean distance, and angular distance. As we discussed in Appendix A, Euclidean distance relates to cosine similarity, cosine distance and angular distance. However, directly embedding into the angular metric gives better results. We refer to Appendix F for more details.

8 Private Approximate Membership Computation

Private Approximate Membership Computation (PAMC) allows a client to check if their input is sufficiently close to elements in a server's database (see Appendix B.2 for a formal definition). PAMC can be viewed as an asymmetric case of Approx-PSI where one party holds a single element. While our Approx-PSI protocol can handle this by padding the smaller set, such solution is inefficient for PAMC, where a simpler solution is to securely comparing the query to each database element.

In [KM21], a PAMC protocol was constructed for Hamming distance using private information retrieval (PIR) (see Appendix C for more information and references) and secure Hamming distance comparison. Here, the server divides the database into a large number of buckets using perceptual hashes, where elements that are Hamming close are more likely to be hashed to the same binary string, thus belong to the same bucket. The client uses the same hashes on the query, and uses the result to query encrypted buckets via PIR, and securely compares each bucket element's Hamming distance to the query against a given threshold.

The protocol in [KM21] has an experimentally derived false negative rate (FNR), meaning it may fails to recognize approximate matches due to the query and the matched element being mapped to different buckets. Because of the complexity of perceptual hashes, this probability cannot be evaluated mathematically, although empirical results show perceptual hashes (FNR 16.8%) outperforming simple projection (FNR 37.88%).

We improve the PAMC accuracy using random projection instead of the perceptual hashes, allowing mathematical analysis. Using the same idea as in our Approx-PSI, we repeat the process $k = n^{\epsilon}\lambda$ times to ensure the query and its match appear in the same bucket at least once except with negligible probability. We then combine the encrypted buckets from all rounds of projections into one new database for PIR, optimizing efficiency since the communication of the PIR is sublinear. The full protocol is described in Algorithm 2.

Now we analyze the protocol to find appropriate parameters k and t. First, to have buckets of small size, we need the number of buckets be $\mathcal{O}(n)$. We set $t = \log n$ to have the same number of buckets as the size of the database, similar to the protocol in [KM21].

Lemma 8.1. *Let $\beta = \mathcal{O}(\log n)$. Assume that each element in the server database is sampled*

Algorithm 2: Private Approximate Membership Computation

```
Parameter: Element size $\ell$, distance threshold $d$, bucket size bound $\beta$
Input     : Client $x \in \{0,1\}^\ell$, Server $A \subseteq \{0,1\}^\ell$, $|A| = n$.
Output    : $b \in \{0,1\}$ to one of the parties.
For $i = 1, \dots, k$, Server uniformly samples $I_i \subseteq [\ell]$ of size $|I_i| = t$, and computes a
  bucket list $B_i$ such that $B_i[p] = \{a \in A : a_{I_i} = p\}$ for $p \in \{0,1\}^t$. If there is a
  bucket with $|B_i[p]| > \beta$, resamples $I_i$. Otherwise, Server fills the bucket to size $\beta$
  with random elements. Server encrypts each element in every bucket, denoted $\tilde{B}_i$.
  Let $\mathcal{B} = \{\tilde{B}_i\}_{i \in [k]}$ be a database for PIR indexed by $(i,p)$. Server sends $\{I_i\}_{i \in [k]}$ to
  Client. ;
For $i = 1, \dots, k$, Client computes $x_i = x_{I_i}$ and queries $\mathcal{B}$ on all indices in $\{(i, x_i)\}$.
  Client receives $\tilde{B}_i[x_i] = \{c_j\}$. For each $j = 1, \dots, \beta$, Client samples $r_j \leftarrow \{0,1\}^\ell$
  and homomorphically adds $r_j$ to $c_j$. Let $\tilde{c}_j$ denoted the result and $\tilde{B}'_i = \{\tilde{c}_j\}$.
  Client sends $\{\tilde{B}'_i\}$ back to Server. ;
For each $i = 1, \dots, k$ and $j = 1, \dots, \beta$, Server decrypts $\tilde{c}_j \in \tilde{B}'_i$ to $a_{ij}$. Server and
  Client runs $\mathcal{F}_{\mathsf{HamCompare}}$ on input $a_{ij}$ and $x + r_j$. Party $\mathcal{P}$ receives output
  $b_{ij} \in \{0,1\}$. They output $\bigwedge_{i,j} b_{ij}$. ;
```

uniformly from $\{0,1\}^\ell$. Then, except with negligible probability in λ, the server can find a projection I_i such that each $|B_i[p]| \leq \beta$ by resampling I_i for $\mathcal{O}(\lambda)$ times.

Proof. Let Y_p be an indicator that $|B_i[p]| > \beta$, and $Y = \sum_{p \in \{0,1\}^t} Y_p$. Since $\mathbb{E}[Y] \geq \Pr[Y > 0]$, we will bound the probability that there exists a bucket with more than β elements by

$$\mathbb{E}[Y] = \sum_{p \in \{0,1\}^t} \mathbb{E}[Y_p] = n\mathbb{E}[Y_p] = n \Pr[Y_p = 1].$$

For $a \in A$, let X_a be an indicator that $a \in B_i[p]$, and $X = \sum_{a \in A} X_a$. Under the assumption that $a \leftarrow \{0,1\}^\ell$, X_a's are independent and $\mathbb{E}[X_a] = \Pr[a \in B_i[p]] = \frac{1}{2^t} = \frac{1}{n}$. Thus, $\mathbb{E}[X] = 1$. By the Chernoff bound and using $\beta = \mathcal{O}(\log n)$,

$$\Pr[Y_p = 1] = \Pr[X > \beta] \leq e^{-\frac{\beta^2}{2+\beta}} \leq \mathcal{O}(\frac{1}{n}).$$

Thus, $\Pr[Y > 0]$ is bounded by a constant $c < 1$. Since $Y > 0$ means there is a bucket with more than β elements, resulting in the server resamples I_i, such resampling can occur at most $\mathcal{O}(\lambda)$ except with probability $2^{-\lambda}$. □

Lemma 8.2. *Let $k = \mathcal{O}(n^\epsilon \lambda)$ for some $0 < \epsilon < \frac{1}{2}$. If a query x has a match in A, then the protocol in Algorithm 2 will output* 1 *except with negligible probability.*

Proof. For any input $x \in \{0,1\}^\ell$ with a match $y \in A$, we have $\mathcal{H}(x,y) \leq d$. Thus, the probability that they do not collide when projected to coordinates in I is at most

$$1 - \frac{\binom{\ell-d}{t}}{\binom{\ell}{t}} = 1 - \left(1 - \frac{d}{\ell}\right)\left(1 - \frac{d}{\ell-1}\right)\cdots\left(1 - \frac{d}{\ell-t+1}\right) \leq 1 - \left(1 - \frac{d}{\ell-t}\right)^t$$

By similar analysis as in Lemma 5.1, the probability that they do not collide for all k rounds is at most e^{-kq} where $q = \left(1 - \frac{d}{\ell - t}\right)^t$. Thus, we can choose $k = \mathcal{O}\left(\frac{\lambda}{q}\right)$ to make the failure probability negligible in λ.

Now we let $t = \log n$. We have

$$q = 2^{\log(1 - \frac{d}{\ell - t}) \log n} = n^{-\epsilon}$$

where $\epsilon = -\log(1 - \frac{d}{\ell - t})$. Thus, the number of rounds $k = \mathcal{O}(n^\epsilon \lambda)$. □

For example, using the parameters in [KM21], $\ell = 256$, $t = 20$ and $d = 25$, gives $\epsilon \approx 0.17$. Using an efficient PIR such as [LMRSW24] that communicates $\mathcal{O}(n^{\frac{1}{3}})$ bits, we obtain an efficient PAMC.

Theorem 8.3. *There exists a PAMC with negligible FNR with communication complexity $\mathcal{O}(n^{\frac{1+\epsilon}{3}} \ell \lambda)$ and computation complexity $\mathcal{O}(n^{1+\epsilon} \ell \lambda)$ for some $0 < \epsilon < 1$.*

We remark that, unlike the Approx-PSI protocol, our PAMC protocol does not require any specific structure on inputs.

9 Implementation and Benchmarks

In this section, we discuss implementations of our Approx-PSI protocol for Hamming distance. The exact numbers of rounds are shown in Appendix H based on the theoretical analysis in Section 6 as a function of security parameter, gap and the number of elements. When the gap is $t = \log n$, the protocol needs to run for about 80 rounds to ensure that the probability that protocol would fail is at most 2^{-40}. This is the number of times the underlying PSI protocol is executed. For the smallest possible gap $t = 4$, the number of rounds approximately double whenever the input sizes quadruple. When the underlying PSI is linear time, for this fixed $t = 4$, the Approx-PSI protocol is $\tilde{\mathcal{O}}(n^{1.33})$.

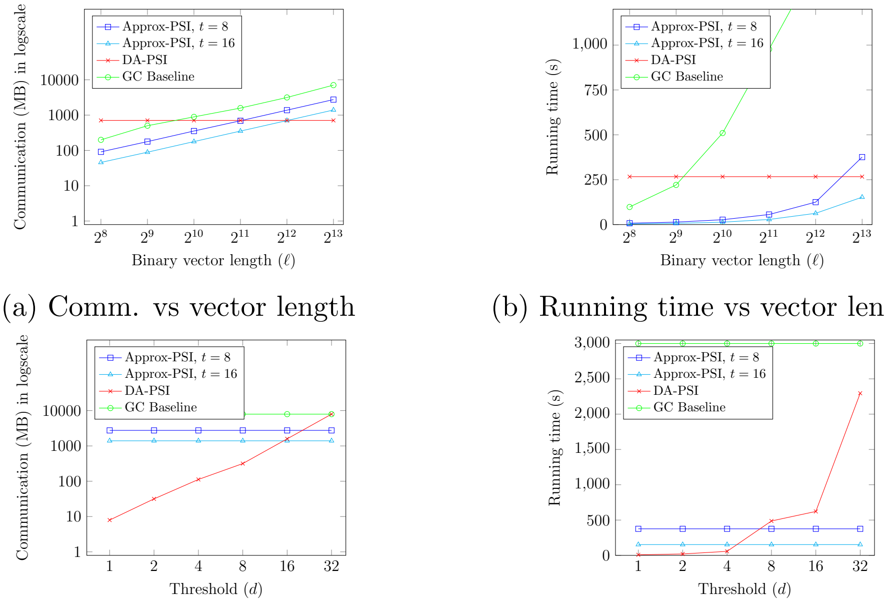
9.1 Performance

We implement our Approx-PSI in C++ using EMP-Toolkit[1] for communications, OTs (for secret-shared operations) and garbled circuits. We use volepsi[2] for the underlying OPPRF protocol in the circuit PSI from [RS21]. We benchmarked our protocol on a virtual machine with 8 vCPUs and 8GB of RAM (all of our implementations are singled-threaded). When comparing to the result in [CFR23], we match their bandwidth of 60MB/s. For comparing between our own result, we use the LAN setting.

First, we compare our Approx-PSI protocol with DA-PSI from [CFR23] for Hamming distance by matching their FNR of 0.05, which corresponds to our security parameter of $\lambda = 5$, and a garbled circuit baseline. Since we do not have access to the DA-PSI code, we try to match their resources usage as closely as possible and use the numbers reported in [CFR23]. Therefore, the comparison is only a rough estimate.

[1] https://github.com/emp-toolkit

[2] https://github.com/Visa-Research/volepsi



(a) Comm. vs vector length

(b) Running time vs vector length

(c) Comm. vs threshold

(d) Running time vs threshold

Figure 5: Running Time in seconds and Communication in logscale of MB of Approx-PSI, DA-PSI and garbled circuit baseline for set size $n = 100$, and (a) - (b) fixed threshold $d = 6$; (c) - (d) fixed element size $\ell = 8192$. Our Approx-PSI uses gap $t = 8, 16$ and security parameter $\lambda = 5$ to match the error rate of 0.05.

From Figure 5 (a) and (b), our protocol outperforms DA-PSI in both communication and running time for short binary vectors, up to $\ell = 2048$ for communication and up to $\ell = 4096$ for running time, with a gap of $t = 8$ and without parallel computation. Even for $\ell = 1024$, our protocol is 10-20 times faster depending on the gap. From Figure 5 (c) and (d), our protocol also outperforms DA-PSI when the matching threshold is above 4 bits for running time and 16 bits for communication – both of which are minuscule relative to the total length of the vectors. For instance, in the image matching application from [KM21], the distance threshold is around 10% of the total vector length. We also note that the element size in Figure 5 (c) and (d) is the largest reported in [CFR23] ($\ell = 8192$). For shorter vectors, such as $\ell = 256$ in [KM21], our protocol demonstrates an even more significant performance improvement. This further highlights the efficiency of our approach when dealing with smaller input sizes.

Second, we demonstrate the performance of our Approx-PSI protocol for Hamming distance under various parameters. Table 3 shows the communication and running time of our protocol as the input size increases, broken down by the main steps, using a much larger security parameter of $\lambda = 40$ and a gap of $t = \log n$. Both communication and running time are nearly linear in relation to the input size, making our protocol scale better with large input sets compared to the quadratic complexity seen in previous works.

We note that the secret-shared Hamming distance comparison test step dominates both the communication and running time, followed by the secret-shared PSI step. These steps are therefore the primary targets for further optimization.

Table 3: Communication and Running time of Approx-PSI for Hamming distance with element size $\ell = 128$, threshold $d = 4$, gap $t = \log n$ and security parameter $\lambda = 40$ for various set size $n = 256, 1024, 4096$, resulting in number of rounds $k = 96, 89, 86$, respectively.

Step	communication (MB)			running time (s)		
n	256	1024	4096	256	1024	4096
Projection	0.01	0.01	0.01	0.004	1.45	4.859
SS-PSI	214.57	848.65	3279.3	15.13	59.28	226.78
SS Ham. comp.	243.58	902.85	3483.4	22.54	83.58	324.9
SS Vector Mult.	4.52	16.71	64.41	0.659	2.285	8.652
Open & output	3	11.12	42.92	0.365	1.256	4.71
Total	465.68	1779.3	6870	38.7	147.85	569.9

While a direct comparison is not possible due to different distance metrics, the sa-PSI protocol in [GRS23] communicates around 30-100 GB for $n = 2700$ ℓ_∞-balls under different conditions. Based on this, our protocol is expected to be quite efficient.

References

[BCG$^+$19] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Efficient pseudorandom correlation generators: Silent ot extension and more. In *Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part III 39*, pages 489–518. Springer, 2019.

[BCG$^+$22] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, Nicolas Resch, and Peter Scholl. Correlated pseudorandomness from expand-accumulate codes. In *Annual International Cryptology Conference*, pages 603–633. Springer, 2022.

[BK73] Walter A. Burkhard and Robert M. Keller. Some approaches to best-match file searching. *Communications of the ACM*, 16(4):230–236, 1973.

[BPSY23] Alexander Bienstock, Sarvar Patel, Joon Young Seo, and Kevin Yeo. Near-Optimal oblivious Key-Value stores for efficient PSI, PSU and Volume-Hiding Multi-Maps. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 301–318, Anaheim, CA, August 2023. USENIX Association.

[CFR23] Anrin Chakraborti, Giulia Fanti, and Michael K Reiter. {Distance-Aware} private set intersection. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 319–336, 2023.

[CGK20] Henry Corrigan-Gibbs and Dmitry Kogan. Private information retrieval with sublinear online time. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 44–75. Springer, 2020.

[CILO22] Wutichai Chongchitmate, Yuval Ishai, Steve Lu, and Rafail Ostrovsky. Psi from ring-ole. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 531–545, 2022.

[CKGS98] Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan. Private information retrieval. *Journal of the ACM (JACM)*, 45(6):965–981, 1998.

[CM20] Melissa Chase and Peihan Miao. Private set intersection in the internet setting from lightweight oblivious prf. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020*, pages 34–63, Cham, 2020. Springer International Publishing.

[CSF$^+$07] M Patrick Collins, Timothy J Shimeall, Sidney Faber, Jeff Janies, Rhiannon Weaver, Markus De Shon, and Joseph Kadane. Using uncleanliness to predict future botnet addresses. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 93–104, 2007.

[Dau09] John Daugman. How iris recognition works. In *The essential guide to image processing*, pages 715–739. Elsevier, 2009.

[DM21] Sjoerd Dirksen and Shahar Mendelson. Non-gaussian hyperplane tessellations and robust one-bit compressed sensing. *Journal of the European Mathematical Society*, 23(9):2913–2947, 2021.

[DMS22] Sjoerd Dirksen, Shahar Mendelson, and Alexander Stollenwerk. Sharp estimates on random hyperplane tessellations. *SIAM Journal on Mathematics of Data Science*, 4(4):1396–1419, 2022.

[DPT20] Thai Duong, Duong Hieu Phan, and Ni Trieu. Catalic: Delegated psi cardinality with applications to contact tracing. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 870–899. Springer, 2020.

[DS18] Sjoerd Dirksen and Alexander Stollenwerk. Fast binary embeddings with gaussian circulant matrices: improved bounds. *Discrete & Computational Geometry*, 60:599–626, 2018.

[DS20] Sjoerd Dirksen and Alexander Stollenwerk. Binarized johnson-lindenstrauss embeddings. *arXiv preprint arXiv:2009.08320*, 2020.

[FKOS15] Tore Kasper Frederiksen, Marcel Keller, Emmanuela Orsini, and Peter Scholl. A unified approach to mpc with preprocessing using ot. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 711–735. Springer, 2015.

[GGM24] Gayathri Garimella, Benjamin Goff, and Peihan Miao. Computation efficient structure-aware psi from incremental function secret sharing. In *Annual International Cryptology Conference*, pages 309–345. Springer, 2024.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of computer and system sciences*, 28(2):270–299, 1984.

[GPR$^+$21] Gayathri Garimella, Benny Pinkas, Mike Rosulek, Ni Trieu, and Avishay Yanai. Oblivious key-value stores and amplification for private set intersection. In *Annual International Cryptology Conference*, pages 395–425. Springer, 2021.

[GRS22] Gayathri Garimella, Mike Rosulek, and Jaspal Singh. Structure-aware private set intersection, with applications to fuzzy matching. In *Annual International Cryptology Conference*, pages 323–352. Springer, 2022.

[GRS23] Gayathri Garimella, Mike Rosulek, and Jaspal Singh. Malicious secure, structure-aware private set intersection. In *Annual International Cryptology Conference*, pages 577–610. Springer, 2023.

[GS19] Satrajit Ghosh and Mark Simkin. The communication complexity of threshold private set intersection. In *Annual International Cryptology Conference*, pages 3–29. Springer, 2019.

[HEKM11] Yan Huang, David Evans, Jonathan Katz, and Lior Malka. Faster secure {Two-Party} computation using garbled circuits. In *20th USENIX Security Symposium (USENIX Security 11)*, 2011.

[HS20] Thang Huynh and Rayan Saab. Fast binary embeddings and quantized compressed sensing with structured matrices. *Communications on Pure and Applied Mathematics*, 73(1):110–149, 2020.

[IKN$^+$20] Mihaela Ion, Ben Kreuter, Ahmet Erhan Nergiz, Sarvar Patel, Shobhit Saxena, Karn Seth, Mariana Raykova, David Shanahan, and Moti Yung. On deploying secure computing: Private intersection-sum-with-cardinality. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 370–389. IEEE, 2020.

[IKOS06] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Cryptography from anonymity. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 239–248. IEEE, 2006.

[JL84] William Johnson and Joram Lindenstrauss. Extensions of lipschitz maps into a hilbert space. *Contemporary Mathematics*, 26:189–206, 01 1984.

[Kel20] Marcel Keller. Mp-spdz: A versatile framework for multi-party computation. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 1575–1590, 2020.

[KM21] Anunay Kulshrestha and Jonathan Mayer. Identifying harmful media in {End-to-End} encrypted communication: Efficient private membership computation. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 893–910, 2021.

[KMWF07] Eike Kiltz, Payman Mohassel, Enav Weinreb, and Matthew Franklin. Secure linear algebra using linearly recurrent sequences. In *Theory of Cryptography: 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007. Proceedings 4*, pages 291–310. Springer, 2007.

[KO97] Eyal Kushilevitz and Rafail Ostrovsky. Replication is not needed: Single database, computationally-private information retrieval. In *Proceedings 38th annual symposium on foundations of computer science*, pages 364–373. IEEE, 1997.

[KOR98] Eyal Kushilevitz, Rafail Ostrovsky, and Yuval Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 614–623, 1998.

[LMRSW24] Baiyu Li, Daniele Micciancio, Mariana Raykova, and Mark Schultz-Wu. Hintless single-server private information retrieval. In *Annual International Cryptology Conference*, pages 183–217. Springer, 2024.

[MKHSO17] Mina Mohammadi-Kambs, Kathrin Hölz, Mark M Somoza, and Albrecht Ott. Hamming distance as a concept in dna molecular recognition. *ACS omega*, 2(4):1302–1308, 2017.

[MPDC19] Luca Melis, Apostolos Pyrgelis, and Emiliano De Cristofaro. On collaborative predictive blacklisting. *ACM SIGCOMM Computer Communication Review*, 48(5):9–20, 2019.

[MPR$^+$20] Peihan Miao, Sarvar Patel, Mariana Raykova, Karn Seth, and Moti Yung. Two-sided malicious security for private intersection-sum with cardinality. In *Annual International Cryptology Conference*, pages 3–33. Springer, 2020.

[NNOB12] Jesper Buus Nielsen, Peter Sebastian Nordholt, Claudio Orlandi, and Sai Sheshank Burra. A new approach to practical active-secure two-party computation. In *Annual Cryptology Conference*, pages 681–700. Springer, 2012.

[OPJM10] Margarita Osadchy, Benny Pinkas, Ayman Jarrous, and Boaz Moskovich. Scifi-a system for secure face identification. In *2010 IEEE Symposium on Security and Privacy*, pages 239–254. IEEE, 2010.

[OR07] Rafail Ostrovsky and Yuval Rabani. Low distortion embeddings for edit distance. *Journal of the ACM (JACM)*, 54(5):23–es, 2007.

[OR15] Samet Oymak and Ben Recht. Near-optimal bounds for binary embeddings of arbitrary sets. *arXiv preprint arXiv:1512.04433*, 2015.

[PRTY19] Benny Pinkas, Mike Rosulek, Ni Trieu, and Avishay Yanai. Spot-light: Lightweight private set intersection from sparse ot extension. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019*, pages 401–431, Cham, 2019. Springer International Publishing.

[PV14] Yaniv Plan and Roman Vershynin. Dimension reduction by random hyperplane tessellations. *Discrete & Computational Geometry*, 51(2):438–461, 2014.

[RR22] Srinivasan Raghuraman and Peter Rindal. Blazing fast psi from improved okvs and subfield vole. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2505–2517, 2022.

[RS21] Peter Rindal and Phillipp Schoppmann. Vole-psi: Fast oprf and circuit-psi from vector-ole. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 901–930. Springer, 2021.

[RT21] Mike Rosulek and Ni Trieu. Compact and malicious private set intersection for small sets. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 1166–1181, 2021.

[Sch18] Peter Scholl. Extending oblivious transfer with low communication via key-homomorphic prfs. In *Public-Key Cryptography–PKC 2018: 21st IACR International Conference on Practice and Theory of Public-Key Cryptography, Rio de Janeiro, Brazil, March 25-29, 2018, Proceedings, Part I 21*, pages 554–583. Springer, 2018.

[UCK$^+$21] Erkam Uzun, Simon P Chung, Vladimir Kolesnikov, Alexandra Boldyreva, and Wenke Lee. Fuzzy labeled private set intersection with applications to private {Real-Time} biometric search. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 911–928, 2021.

[WACL10] Andrew G West, Adam J Aviv, Jian Chang, and Insup Lee. Spam mitigation using spatio-temporal reputations from blacklist history. In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 161–170, 2010.

[WHZ$^+$15] Xiao Shaun Wang, Yan Huang, Yongan Zhao, Haixu Tang, XiaoFeng Wang, and Diyue Bu. Efficient genome-wide, privacy-preserving similar patient query based on private edit distance. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 492–503, 2015.

[YCP15] Xinyang Yi, Constantine Caramanis, and Eric Price. Binary embedding: Fundamental limits and fast algorithm. In *International Conference on Machine Learning*, pages 2162–2170. PMLR, 2015.

[Yia93] Peter N Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *Soda*, volume 93, pages 311–21, 1993.

A Euclidean distance, Cosine Similarity and Angular distance

Here we give formulas for the distances in Euclidean space, and their relationship. For any $x, y \in \mathbb{R}^N$, with $x = (x_1, \dots, x_N)$ and $y = (y_1, \dots, y_N)$, we have

- **Euclidean distance**:
$$\|x - y\|_2 = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2};$$

- **cosine distance**:
$$\delta_{\cos}(x, y) = 1 - \frac{x \cdot y}{\|x\|_2 \|y\|_2};$$
We note that $1 - \delta_{\cos}(x, y) = \dfrac{x \cdot y}{\|x\|_2 \|y\|_2}$ is called the *cosine similarity* between x and y. In numerous analytical and computational contexts, cosine similarity serves as a prevalent metric to determine the degree of similarity or alignment between two data sets.

- **angular distance**:
$$\delta_\theta(x, y) = \frac{\arccos\left(\frac{x \cdot y}{\|x\|_2 \|y\|_2}\right)}{\pi}.$$
When x, y are unit vectors, i.e., in the unit sphere S^{N-1}, this distance is also called *geodesic* distance as it is the length of the shortest path on the sphere connecting x and y.

The cosine distance has values between 0 and 2 inclusive while the angular distance has values between 0 and 1 inclusive. Clearly,

$$\delta_\theta(x, y) = \frac{\arccos(1 - \delta_{\cos}(x, y))}{\pi},$$

and

$$\begin{aligned}\|x - y\|_2^2 &= \|x\|_2^2 + \|y\|_2^2 - 2(x \cdot y) \\ &= \|x\|_2^2 + \|y\|_2^2 - 2\|x\|_2 \|y\|_2 (1 - \delta_{\cos}(x, y)).\end{aligned}$$

The cosine distance and the angular distance do not concern the length of x, y when they are nonzero vectors. In this case, we may assume that $x, y \in S^{N-1}$, a unit sphere in $\mathbb{R}^N$. Under this condition,

$$\|x - y\|_2 = \sqrt{2\delta_{\cos}(x, y)}.$$

Thus, secure computation of the Euclidean distance implies secure computation of the cosine distance and cosine similarity as well.

B Ideal Functionalities

Here we provide various ideal functionalities for completion.

B.1 Circuit PSI

We describe the ideal functionality for circuit PSI from [RS21] in Figure 6.

$\mathcal{F}_{\mathsf{cPSI}}$

Parameters. element set $\mathcal{U}$, payload set $\{0,1\}^\sigma$, set size m, a map $\mathsf{Reorder} : \mathcal{U}^m \to \{\pi : [m] \to [m'], \text{injective}\}$ with $m' > m$

Functionality.

1. Upon receiving a message $(\mathsf{inputS}, A, \tilde{A})$ from the sender where $A = \{a_1, \ldots, a_m\} \subseteq \mathcal{U}$ and $\tilde{A} = \{\tilde{a}_1, \ldots, \tilde{a}_m\} \in \{0,1\}^\sigma\}$, store $(A, \tilde{A})$.
2. Upon receiving a message $(\mathsf{inputR}, B, \tilde{B})$ from the receiver where $B = \{b_1, \ldots, b_m\} \subseteq \mathcal{U}$ and $\tilde{B} = \{\tilde{b}_1, \ldots, \tilde{b}_m\} \in \{0,1\}^\sigma\}$, store $(B, \tilde{B})$.
3. If both $(A, \tilde{A})$ and $(B, \tilde{B})$ are stored, compute $\pi = \mathsf{Reorder}(B)$, and uniformly samples $c^0, c^1 \leftarrow \{0,1\}^{m'}$ and $z^0, z^1 \leftarrow (\{0,1\}^{2\sigma})^{m'}$ conditioned on
   - $c^0_{j'} \oplus c^1_{j'} = 1, z^0_{j'} \oplus z^1_{j'} = (\tilde{a}_{j'} \| \tilde{b}_{j'})$ if $\exists a_i \in A$ s.t. $a_i = b_j$
   - $c^0_{j'} \oplus c^1_{j'} = 0, z^0_{j'} \oplus z^1_{j'} = 0^{2\sigma}$, otherwise

   for $j' = \pi(j)$. Send c^0, z^0 to the sender and c^1, z^1, π to the receiver.

Figure 6: Ideal functionality for circuit PSI [RS21]

B.2 Private Approximate Match Computation

In [RS21], private approximate match computation (PAMC) was not defined as a functionality, but the paper described the security requirement for each party instead. Here, we describe the ideal functionality for PAMC in Figure 7.

C Private Information Retrieval

Private information retrieval (PIR) [CKGS98,KO97] allows a client to query an entry from a database managed by a server without revealing which entry is being queried. An straightforward method for this would be for the server to send the entire database to the client. However, this approach is communication inefficient, especially when the database is large. Non-trivial PIR protocols enable sublinear communication between the client and the server.

$\mathcal{F}_{\mathsf{PAMC}}$

Parameters. element set $\mathcal{U}$ with distance metric δ, database size n, distance threshold d, party receiving the output $P \in \{\mathsf{server}, \mathsf{client}\}$.

Functionality.

1. Upon receiving a message (DB, A) from the server where $A = \{a_1, \ldots, a_n\} \subseteq \mathcal{U}$, store A.
2. Upon receiving a message (query, x) from the client where $x \in \mathcal{U}$, store x.
3. If both A and x are stored, compute $b \in \{0, 1\}$ where $b = 1$ if and only if there exists $a \in A$ such that $\delta(x, a) \leq d$. Send b to party P.

Figure 7: Ideal functionality for PAMC

It is important to note, though, that PIR does not prevent the client from learning about entries they did not query.

Different PIR constructions vary in terms of computation, communication, memory requirements, and the number of duplicated databases needed. Traditionally, single-database PIR protocols are constructed using homomorphic encryption or oblivious RAM (ORAM) [KO97, IKOS06]. To reduce server running time, an offline/online model is sometimes used [CGK20, LMRSW24]. In this model, the server preprocesses the database using linear time before the query. When the client sends a query, the server only needs sublinear time to respond.

In this work, we use the hintless single-server PIR from [LMRSW24] as each runtime-constructed database is queried only once. The $\mathsf{TensorPIR}$ protocol achieves $\tilde{\mathcal{O}}(1)$ offline communication and $\tilde{\mathcal{O}}(n^{\frac{1}{3}})$ online communication, with $\tilde{\mathcal{O}}(n)$ computation for a database of size n. For more details on the construction and security proof, we refer to [LMRSW24].

D Length-independent Secret-Shared Hamming Distance Comparison

We give more details on the protocol realizing $\mathcal{F}_{\mathsf{ssHamCom}}$ which has communication complexity $\mathcal{O}(\lambda d^2)$, independent of the length of an element. The protocol combines the ideas from three different protocols from [CFR23, GS19, KMWF07].

As in the beginning of the protocol in [CFR23], we consider a binary vector of length ℓ as an ℓ-subset whose elements indicated by each bit of the vector. The Hamming distance can then be computed from the set difference. We use the idea from [GS19] that transforms a subset into a sparse polynomial and then evaluates the polynomial at various points to form a $d \times d$ matrix. The matrices has the property that the size of the corresponding set difference is below d if and only if the subtraction of the matrices is singular. This property can be checked securely using a secure determinant computation from [KMWF07]. The last step of the protocol in [KMWF07] uses an additive homomorphic encryption. We can then modify it to output a secret share of the indicator result.

We obtain the protocol in Algorithm 3. Let $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be an additive homomorphic encryption. Let p be a prime integer such that $p > 2\ell$ and $p > (4d^2 + 2d)2^\lambda$.

```
Algorithm 3: Secret-Shared Hamming Distance Comparison Test
  Input  : $a, b \subseteq \{0,1\}^\ell$
  Output: $[out]_S, [out]_R \in \{0,1\}$ where $out = [out]_S \oplus [out]_R = 1$ if $\mathcal{H}(a,b) \leq d$ and
          $out = 0$ otherwise
  Parties compute $P_a = \sum_{i \in \{0,1,\ldots,\ell-1\}} x^{2i+a_i}$ and $P_b$ defined similarly;
  Parties jointly sample $u \leftarrow \mathbb{F}_p$;
  Parties compute an $(2d+1) \times (2d+1)$ matrix
      $H_a = \begin{bmatrix} P_a(u^0) & \cdots & P_a(u^{2d}) \\ \vdots & & \vdots \\ P_a(u^{2d}) & \cdots & P_a(u^{4d}) \end{bmatrix}$
    and $H_b$ defined similarly;
  Sender generates $\mathsf{KeyGen} \rightarrow (pk, sk)$ and sends $(pk, \mathbf{H}_a = \mathsf{Enc}_{pk}(H_a))$ to Receiver;
  Receiver computes $\mathbf{H} = \mathbf{H}_a - \mathsf{Enc}_{pk}(H_b)$ (denote $H = H_a - H_b$), samples
    $\vec{u}, \vec{v} \leftarrow \mathbb{F}_p^{2d+1}$;
  Parties interactively compute $\mathbf{H}^k \vec{v}$ for $k = 1, \ldots, 4d+1$;
  Receiver computes $\vec{u}^T \mathbf{H}^k \vec{v}$ for $k = 1, \ldots, d$, and $\mathbf{M}_H$, and encryption of the minimal
    polynomial $m_H$ of $H$.;
  Parties evaluate garbled circuit that decrypts and secret shares an indicator that the
    constant term of $\mathbf{M}_H$ is zero.
```

The correctness of the protocol follows from the fact analyzed in [GS19] that $\det(H_a - H_b) = 0$ if and only if $\mathcal{H}(a,b) \leq d$. We utilize the homomorphic encryption method in [KMWF07] to compute the determinant from the minimal polynomial of the matrix, which can be computed from $\vec{u}^T H^k \vec{v}$.

From the analysis in [KMWF07], the protocol above has communication and computation complexity of $\mathcal{O}(\lambda d^2 \mathsf{polylog}\, d)$. When adding the local transformation in the first step, the computation complexity is $\mathcal{O}(\lambda(\ell + d^2 \mathsf{polylog}\, d))$.

E Secret-Shared Scalar-Vector Multiplication from OT

Using OT, we can easily realize $\mathcal{F}_{\mathsf{ssVMult}}$ in the semi-honest model. Let $\mathcal{F}_{\mathsf{OT}}$ be the 1-out-of-2 OT functionality. We describe the protocol in Algorithm 4.

As $w_S = r_R \oplus ([c]_S \cdot [v]_R)$, we have $[out]_S = ([c]_S \cdot [v]_S) \oplus ([c]_S \cdot [v]_R) = [c]_S \cdot ([v]_S \oplus [v]_R)$. Similarly, $[out]_R = [c]_R \cdot ([v]_S \oplus [v]_R)$. Using OT extension techniques, the (amortized) communication and computation can be reduced to $o(1)$ [Sch18, BCG$^+$19].

Algorithm 4: Secret-Shared Vector Multiplication

```
Input  : $[v]_S, [v]_R \subseteq \{0,1\}^\ell$, $[c]_S, [c]_R \in \{0,1\}$
Output: $[out]_S, [out]_R \in \{0,1\}^\ell$ where
        $out = [out]_S \oplus [out]_R = ([c]_S \oplus [c]_R) \cdot ([v]_S \oplus [v]_R)$
1 Sender samples $r_S \leftarrow \{0,1\}^\ell$ and sends $(\mathsf{inputS}, r_S, r_S \oplus [v]_S)$ to $\mathcal{F}_{\mathsf{OT}}$. Receiver sends
    $(\mathsf{inputR}, [c]_R)$ to $\mathcal{F}_{\mathsf{OT}}$ and receives $w_R$;
2 Receiver samples $r_R \leftarrow \{0,1\}^\ell$ and sends $(\mathsf{inputS}, r_R, r_R \oplus [v]_R)$ to $\mathcal{F}_{\mathsf{OT}}$. Sender sends
    $(\mathsf{inputR}, [c]_S)$ to $\mathcal{F}_{\mathsf{OT}}$ and receives $w_S$;
3 Sender outputs $[out]_S = ([c]_S \cdot [v]_S) \oplus r_S \oplus w_S$, and Receiver outputs
    $[out]_R = ([c]_R \cdot [v]_R) \oplus r_R \oplus w_R$.
```

F Reduction to Other Distance Metrics

Here we give more details about the embedding from three other distance metric into Hamming distance metric.

F.1 Edit Distance

Our protocol relies on the low distortion embedding by Ostrovsky and Rabani [OR07].

Theorem F.1 ([OR07]). *There exists a polynomial time algorithm ϕ that for every $\delta > 0$, $\phi = \phi(\cdot, \ell, \delta) : \{0,1\}^\ell \to \{0,1\}^{\ell'}$ such that $\ell' = \mathcal{O}(\ell^2 \log(\ell/\delta))$ satisfying for any $x, y \in \{0,1\}^\ell$*

$$\Gamma(\ell)^{-1}\mathrm{ed}(x,y) \leq \mathcal{H}(\phi(x), \phi(y)) \leq \Gamma(\ell)\mathrm{ed}(x,y),$$

where $\Gamma(\ell) = 2^{\mathcal{O}(\sqrt{\log \ell \log\log \ell})}$, with probability at least $1 - \delta$.

We note that asymptotically $\log^M \ell < \Gamma(\ell) < \ell^\epsilon$ for any large constant M and small constant $\epsilon > 0$. Applying the embedding and Approx-PSI for Hamming distance gives the following corollary.

Corollary F.2. *There exists a Approx-PSI for edit distance with communication and computation $\mathcal{O}(n^{1+\frac{1}{t-1}}\ell^2(\log n + \lambda)^2)$ for gap $t' = 2^{\mathcal{O}(\sqrt{\log \ell \log\log \ell})}t$*

Proof. Let d, t be the threshold and the gap of the underlying Approx-PSI for Hamming distance, respectively. By Theorem F.1, for any $a \in A$ and $b \in B$ such that $\mathrm{ed}(a,b) \leq d'$,

$$\mathcal{H}(\phi(a), \phi(b)) \leq 2^{\mathcal{O}(\sqrt{\log \ell \log\log \ell})}\mathrm{ed}(a,b) \leq 2^{\mathcal{O}(\sqrt{\log \ell \log\log \ell})}d'.$$

For any $a \in A$ and $b \in B$ such that $\mathrm{ed}(a,b) \geq t'd'$,

$$\mathcal{H}(\phi(a), \phi(b)) \geq 2^{-\mathcal{O}(\sqrt{\log \ell \log\log \ell})}\mathrm{ed}(a,b) \geq 2^{-\mathcal{O}(\sqrt{\log \ell \log\log \ell})}t'd'.$$

Setting the right hand side of each inequality as d and td, respectively, gives $t' = 2^{\mathcal{O}(\sqrt{\log \ell \log\log \ell})}t$. Here we set $\delta = \frac{1}{n^2 2^\lambda}$. Thus, $\ell' = \mathcal{O}(\ell^2(\log n + \lambda))$.

The communication and computation of the resulting protocol is that of Approx-PSI for Hamming distance where element size is $\ell' = \mathcal{O}(\ell^2(\log n + \lambda))$.

□

F.2 Euclidean Distance

Similar to the Edit distance, there are embedding from the Euclidean distance to the Hamming distance [PV14, HS20, DS20, DM21, DMS22]. Unlike the embedding for the edit distance, which is complicated, the embeddings for Euclidean following the simple ideas from the Johnson-Lindenstrauss lemma [JL84]. The original lemma concerns the dimension reduction of vectors in $\mathbb{R}^N$. However, the technique can be used to constructed an embedding into binary strings, represented by $\{-1, 1\}$ instead of $\{0, 1\}$.

A hyperplane in $\mathbb{R}^N$ is chosen randomly to cut $\mathbb{R}^N$ into two halves. Vectors in one half is mapped to -1 while the other half is mapped to 1. This can be computed by the sign of inner product between the vectors and the normal vector of the hyperplane. The process is repeated multiple times with independently chosen hyperplanes to obtain a binary vector. The idea has been improved with better methods of chosing the hyperplanes and the analysis of the resulting distortion. Here we choose the most recent results for our construction.

Theorem F.3 ([DM21]). *There exists a polynomial time algorithm ϕ that for every $0 < \rho < R$ and $T \subseteq \mathbf{B}(R)$ with $|T| = n$ where $\mathbf{B}(R)$ is a Euclidean ball of radius R, $\phi : \mathbb{R}^N \to \{0, 1\}^\ell$ such that $\ell = \mathcal{O}(\frac{R \log(eR/\rho) \log n}{\rho^3})$, satisfying for any $x, y \in T$ such that $\|x - y\|_2 \geq \rho$*

$$\mathcal{O}(\frac{\ell}{R})\|x - y\|_2 \leq \mathcal{H}(\phi(x), \phi(y)) \leq \mathcal{O}(\frac{\ell\sqrt{\log(eR/\rho)}}{R})\|x - y\|_2$$

with probability at least $1 - e^{-\mathcal{O}(\ell\rho/R)}$.

While this multiplicative bound is easy to use, the condition $\|x - y\|_2 \geq \rho$ can be problematic as the protocol cannot check this condition efficiently. Thus, we consider the following additive bound which is a special case of the result in [DMS22].

Theorem F.4 ([DMS22]). *There exists a polynomial time algorithm ϕ that for $R > 0$, $0 < \delta < R/2$, $\rho = \mathcal{O}(R\sqrt{\log(R/\delta)})$ and $T \subseteq \mathbf{B}(R)$ with $|T| = n$ where $\mathbf{B}(R)$ is a Euclidean ball of radius R, $\phi : \mathbb{R}^N \to \{0, 1\}^\ell$ such that $\ell = \mathcal{O}\left(\frac{\rho^2(\log n + \lambda)}{\delta^2}\right)$, satisfying for any $x, y \in T$,*

$$\left|\frac{\sqrt{2\pi}\rho}{\ell}\mathcal{H}(\phi(x), \phi(y)) - \|x - y\|_2\right| \leq \delta$$

with probability at least $1 - e^{-\mathcal{O}(\delta^2\ell/\rho^2)}$.

Corollary F.5. *There exists a Approx-PSI for Euclidean distance with gap t with communication and computation complexity*
$\mathcal{O}(n^{1+\frac{1}{t-1}}t^2 \log t(\log n + \lambda)^2)$.

Proof. Let d_0, t_0 be the threshold and the gap for the underlying Approx-PSI for Hamming distance, respectively. From Theorem F.6, for a pair $x, y \in \mathbb{R}^N$ with $d_\theta(x, y) \leq d$, we have $\mathcal{H}(\phi(x), \phi(y)) \leq (d + \delta)\ell \leq d_0$. For a pair $x, y \in \mathbb{R}^N$ with $d_\theta(x, y) \geq td$, we have $\mathcal{H}(\phi(x), \phi(y)) \geq (td - \delta)\ell \geq t_0 d_0$. Thus, t_0 must satisfy $t_0(d + \delta) \leq td - \delta$.

Since any two vectors in the Euclidean ball of radius R has distance at most $2R$, we may assume that $\frac{R}{d} = \mathcal{O}(t)$. Let $\delta = \mathcal{O}(d)$, $t_0 \leq \frac{td-\delta}{d+\delta} = \mathcal{O}(t)$. We can choose $\rho = \mathcal{O}(R\sqrt{\log t})$. We have $\ell = \mathcal{O}(t^2 \log t(\log n + \lambda))$. This give the communication and computation complexity of the Approx-PSI for Euclidean distance $\mathcal{O}(n^{1+\frac{1}{t-1}t^2 \log t(\log n+\lambda)^2)}$ □

For example, when $t = \sqrt{\log\left(\frac{\lambda}{\log n}\right)\log n}$, the communication is $\mathcal{O}(n(\log n+\lambda)^2 R\sqrt{\log\left(\frac{\lambda}{\log n}\right)})$. When $t = \sqrt{\log\left(\frac{\lambda}{\log n}\right)}$, the communication is $\mathcal{O}(n^{1+\epsilon}(\log n + \lambda)^2 R\sqrt{\log\left(\frac{\lambda}{\log n}\right)})$ where $\epsilon = \frac{1}{t_0-1}$ and t_0 is the constant gap of the underlying Approx-PSI for Hamming distance.

When the vectors are in S^{N-1}, we can obtain the result for cosine similarity and cosine distance by transformation

$$\delta_{\cos}(x, y) = \frac{\|x-y\|_2^2}{2}.$$

In this case, the gap is $t = \log\left(\frac{\lambda}{\log n}\right) t_0^2$.

F.3 Angular Distance

The same hyperplane technique above also gives results for angular distance [PV14, YCP15, OR15, DS18]. We consider the embedding described by Dirksen and Stollenwerk [DS18] as its embedding size is smaller, and more concrete parameters are provided. Unlike the first two distance metrics, the angular distance for any pair of vectors are bounded between 0 and 1.

Theorem F.6 ([DS18])**.** *There exists a polynomial time algorithm ϕ that for every $T \subseteq S^{N-1}$ with $|T| = n$, $\phi : S^{N-1} \to \{0,1\}^\ell$ such that $\ell = \mathcal{O}(\log\left(\frac{n}{\eta}\right)/\delta^2)$, satisfying for any $x, y \in T$*

$$\left|\frac{\mathcal{H}(\phi(x), \phi(y))}{\ell} - \delta_\theta(x, y)\right| \le \delta$$

with probability at least $1 - \eta$.

We combine the embedding and the Approx-PSI for Hamming distance to get the following result.

Corollary F.7. *There exists a Approx-PSI for the angular distance where matching vectors have angular distance at most t_1 and non-matching vectors have angular distance at least t_2 with communication and computation complexity $\mathcal{O}(n^{1+\frac{1}{t-1}} t^2(\log n+\lambda)^2)$ where $t = \mathcal{O}(t_2/t_1)$.*

Proof. Let $\delta > 0$ and ℓ as in Theorem F.6. For a pair $x, y \in \mathbb{R}^N$ with $d_\theta(x, y) \le t_1$, we have $\mathcal{H}(\phi(x), \phi(y)) \le (t_1 + \delta)\ell \le d$. For a pair $x, y \in \mathbb{R}^N$ with $d_\theta(x, y) \ge t_2$, we have $\mathcal{H}(\phi(x), \phi(y)) \ge (t_2 - \delta)\ell \ge td$. Thus, t must satisfy $t(t_1 + \delta) \le t_2 - \delta$. That is $(t + 1)\delta \le t_2 - tt_1$. Thus, we need $t_2 - tt_1 > 0$ and $\delta \le \frac{t_2 - tt_1}{t+1} < \frac{1}{t+1}$ as $t_2 < 1$.

Setting $\eta = 2^\lambda$ and $1/\delta = \mathcal{O}(t)$ gives $\ell = \mathcal{O}(t^2(\log n + \lambda))$ and $t = \mathcal{O}(t_2/t_1)$. This gives the communication and computation complexity of Approx-PSI for Angular distance $\mathcal{O}(n^{1+\frac{1}{t-1}} t^2(\log n + \lambda)^2)$. □

Here we consider t_2 as a constant while $t_1 < t_2/t$ becomes smaller as t increases.

G When inputs do not conform to the structure

Now we discuss what happens when the input sets are not conform to the structure. This means there exists some $a, b \in A \cup B$ such that $d < \delta(a, b) < td$. We further divide the situations and how to deal with them in two cases: the mild case when every pair of elements within each input set still conforms to the structure but not across the sets, and the extreme case when the structure is not hold even within each set.

1. Every pair of elements within each input set conforms to the structure, meaning that every pair of elements in A is either near or far, and the same holds for every pair in B. This scenario can be verified by semi-honest parties since the condition is local. Each party can check their own set during the clustering step. In this case, Lemma 3.1 does not imply that if representatives of clusters in A and B are matched, then every pair of elements, one from each cluster, will also be matched. This introduces the possibility of false positives in the existing algorithm. Additionally, when representatives do not match, it is possible that members of their cluster could still match, leading to false negatives.

   To address this, we can modify the protocol to prevent false positives by checking every pair of elements from each matched cluster using $\mathcal{F}_{\mathsf{ssHamCom}}$. To prevent false negatives, in each round, the parties uniformly select a new representative for each cluster to be projected and sent to $\mathcal{F}_{ssPSI}$. The number of rounds must increase to ensure that if there is a matched pair in the clusters, the matched representatives are chosen in some rounds. When every cluster has constant size, both modifications increase communication and computation by a constant factor. Therefore, the resulting protocol remains near-linear with negligible probability of either false positives or false negatives.

2. Each pair of element, whether from the same input set or across sets, can be at any distance apart. In this case, the projection method and the analysis in Lemma 6.1 still hold. Here, the clustering is no longer unique, meaning there could be multiple ways to cluster elements through a randomized clustering algorithm. Since Lemma 5.1 is no longer hold, it is possible that the party's elements could collide more often than we previously analyzed. This could result in some matched pairs being undiscovered.

   Using the same modification as in the previous case, we could reduce false negatives by increasing the number of rounds. However, the analysis becomes more complicated and highly dependent on the structure of the distances between elements. We leave this case for further exploration in future work.

H Exact Number of Rounds in Approx-PSI for Hamming Distance

Here we calculate the exact number of rounds shown in Table 4 using the calculation from Section 6.

Table 4: Number of rounds for each value of gap t and number of elements in input sets when the security parameter is $\lambda = 40$

gap t	number of elements						
	128	256	512	1024	4096	16384	65536
4	331	431	558	722	1203	1994	3293
5	189	232	285	349	521	773	1142
6	131	156	186	221	309	429	593
7	101	118	138	160	215	286	378
8	83	96	110	126	164	212	272
9	71	81	92	104	133	167	210
10	63	71	80	89	112	139	171
11	57	63	71	79	97	119	145
12	52	58	64	71	86	104	126
13	48	53	58	64	78	93	111
14	45	49	54	59	71	85	100
15	42	46	51	55	66	78	91
16	40	44	48	52	61	72	84
17	38	41	45	49	57	67	78
18	36	39	43	46	54	63	73

I Extension to the Malicious Setting

Throughout this work, we have focused on the approximate PSI in the semi-honest setting for simplicity. In this section, we briefly explain how our Approx-PSI protocol for Hamming distance can be extended to remain secure in the malicious setting as well, and so are the ones for other distances. In the Algorithm 1, a malicious party may deviate from the protocol by (1) clustering the elements incorrectly or the elements in their set do not conform to the structure $\mathcal{S}$; (2) providing incorrect element-projection pairs to $\mathcal{F}_{\mathsf{ssPSI}}$; (3) modify their shares output from $\mathcal{F}_{\mathsf{ssPSI}}$, $\mathcal{F}_{\mathsf{ssHamCom}}$ or $\mathcal{F}_{\mathsf{ssVMult}}$.

The deviation (3), as mentioned in Section 4, can be prevented using various authenticated secret sharing techniques. When $\mathcal{F}_{\mathsf{ssPSI}}$ is instantiated using the protocol in [RS21], we may need to modify the circuit PSI to accommodate the secret sharing scheme we may use.

For (1), as discussed in Appendix G, it would result in false positive for elements in the cluster or false negative for the elements not conforming to the structure $\mathcal{S}$ in the adversary's set. Neither of which would leak information on the honest party's elements. In this case, however, we need to modify the functionality to allow such mistakes.

Unlike the other two, the deviation (2) requires further machinery to fix. In particular, the parties need to provide a zero-knowledge proof that their computed projection is correct. Since the projection is publicly known linear map, an efficient ZKP can be incorporated into the OKVS technique in the protocol in [RS21].

Since each of the above solution does not require more asymptotic communication, the resulting maliciously secure protocol has the same asymptotic communication complexity as the original protocol.