Fast Approximate ℓ_p Chamfer Distance via Lopsided Embeddings and Structured JL

Anonymous Author(s)
Affiliation
Address
email

Abstract

For two d-dimensional point sets A, B of size n, the Chamfer distance from A to B is defined as $\mathrm{CH}(A,B) = \sum_{a\in A} \min_{b\in B} \mathsf{dist}(a,b)$, where dist is some underlying distance measure. We present efficient algorithms for approximating the Chamfer distance under the ℓ_p norm for $1 < p \leq 2$. For the general case $1 < p < 2$, we utilize lopsided ℓ_p to ℓ_1 embeddings with weak guarantees. We show that they are sufficient for preserving the Chamfer distance. In the high-dimensional regime, we use Fast Matrix Multiplication techniques to speed up the lopsided embeddings. These lead to $\mathcal{O}(nd \cdot \mathrm{poly}(\log\log n, \log(1/\varepsilon))/\varepsilon^2)$ total time. For the specific Euclidean case ($p = 2$), we leverage the Fast Johnson-Lindenstrauss Transform based on Toeplitz matrices and re-analyze the previous ℓ_1-specific Chamfer algorithm in ℓ_2. These achieve a runtime matching the recent state-of-the-art ℓ_1 result of $\mathcal{O}(nd(\log\log n + \log(1/\varepsilon))/\varepsilon^2)$, improving upon the previous $\mathcal{O}(nd\log n/\varepsilon^2)$ bound for ℓ_2. We also give additional results for the angular distance and upper and lower bounds in the streaming setting.

1 Introduction

The Chamfer distance is a popular measurement for evaluating the similarity between two point sets, finding extensive application across various domains in machine learning and computer vision. Given two point sets $A, B \subset \mathbb{R}^d$ of size up to n, the directed Chamfer distance under the ℓ_p norm is defined as:

$$\mathrm{CH}_p(A,B) := \sum_{a\in A} \min_{b\in B} \|a-b\|_p.$$

This distance serves as a fundamental tool for quantifying dissimilarity when the exact correspondence between points is unknown or computationally intractable, making it a faster and often preferred alternative to metrics like the Earth-Mover Distance (Wasserstein Distance) [KSKW15].

In computer vision, the Chamfer distance is indispensable. It is widely used for tasks such as 3D shape reconstruction from single images [FSG17a], point cloud registration, and object detection. It often serves as a differentiable loss function within deep learning pipelines, enabling end-to-end training for geometric tasks [AS03, SMFW04, FSG17b, JSQJ18]. Beyond vision, it has gained traction in natural language processing. For instance, the Relaxed Earth Mover Distance [KSKW15, AM19], a variant of the Chamfer distance, is used to measure document similarity based on word embeddings. Furthermore, a closely related concept, the "sum of maximum similarities", where the minimum distance is replaced by the maximum inner product, underpins the efficient ColBERT passage retrieval system [KZ20].

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

The necessity for efficient computation is particularly acute in modern high-dimensional settings. Whether comparing high-dimensional feature embeddings generated by deep neural networks or analyzing large-scale 3D models, the naive $\mathcal{O}(n^2 d)$ computation time quickly becomes a bottleneck. This necessitates the development of fast approximation algorithms.

The first improvement over the naive $\mathcal{O}(dn^2)$-time algorithm was obtained in [SMFW04]. They use the fact that $\mathsf{CH}(A, B)$ can be computed by performing $|A|$ nearest neighbor queries in a data structure storing B. Using this framework and the state of the art approximate nearest neighbor algorithms designed for low-dimensional regime, it leads to an $(1+\epsilon)$-approximate algorithm with $\mathcal{O}\left(n(1/\varepsilon)^{\mathcal{O}(d)} \log n\right)$ running time [AMN$^+$98], or a similar bound that scales exponentially in the dimension (e.g [EK23]), and $\mathcal{O}\left(dn^{1+\frac{1}{2(1+\epsilon)^2-1}}\right)$ running time in high dimensions [AR15]. However, the first bound suffers from exponential dependence on the dimension, while the second bound is significantly subquadratic only for large approximation factors. Recent advancements have made significant strides in accelerating the computation. [BIJ$^+$23] provided the first near-linear time $(1+\varepsilon)$-approximation in $\mathcal{O}(nd \log(n)/\varepsilon^2)$ time for ℓ_1 and ℓ_2. Subsequently, [FI25] introduced a highly efficient $(1+\varepsilon)$-approximation algorithm specifically for the ℓ_1 Chamfer distance, achieving a runtime of $\mathcal{O}(nd(\log\log n + \log(1/\varepsilon))/\varepsilon^2)$. However, extending this efficiency gain to the ℓ_2 norm and the broader range of $1 < p \leq 2$ remained an open challenge. This is significant because the choice of p significantly impacts the properties of the distance measure and its suitability for specific tasks, and many critical applications rely on the Euclidean (ℓ_2) norm or other ℓ_p norms ($1 < p \leq 2$).

1.1 The Importance of General ℓ_p Norms.

Euclidean (ℓ_2) Norm: The ℓ_2 norm is the standard geometric distance and is fundamental in applications involving physical space, such as robotics and 3D modeling. It is also the default metric in many embedding spaces (e.g., ℓ_2-normalized embeddings in contrastive learning). Also, it is closely related to the MaxSim similarity used in the ColBERT system

ℓ_p Norm for Intermediate $p \in (0, 1)$: While ℓ_1 (Manhattan distance) promotes sparsity and is robust to outliers, and ℓ_2 is sensitive to large deviations, intermediate ℓ_p norms ($1 < p < 2$) offer a crucial trade-off. They provide a tunable balance between the robustness of ℓ_1 and the rotational invariance of ℓ_2. In tasks involving noisy sensor data or feature spaces where intermediate levels of outlier resilience are desired, optimizing the Chamfer distance under a specific $p \in (1, 2)$ can yield superior performance compared to fixed ℓ_1 or ℓ_2 metrics.

To illustrate this, we give a concrete example of applications of the Chamfer distance, and show that for this application and some naturally motivated input distribution, the Chamfer distance with $p = 1.5$ results in better performance than $p = 1, 2$. We consider the task of using the Chamfer distance as a cheap surrogate for the Euclidean Earth Mover Distance (EMD). Indeed, the Chamfer distance is widely used precisely when exact correspondence like EMD is too expensive to compute. With an extensive line-of-work studying the problem of approximating EMD (e.g., [Cha02, IT03, AIK08, SA20, AS14, AFP$^+$17, KNP19, ACRX22, AZ23, BR24]), the current best runtime is $n^{2-\tilde{\Omega}(\varepsilon^{1/3})}$ for $(1+\varepsilon)$-approximation [BCAJW25]. Unfortunately, this is only slightly sub-quadratic in n; and for large dataset size n, this can be prohibitively expensive to compute. Although the Chamfer distance does not approximate EMD in the worst-case, in practical fields such as machine learning [KSKW15, WCL$^+$19] and computer vision [SMFW04, FSG17b, JSQJ18]., there has been a long-established line of works that successfully uses (variants of) the Chamfer distance as a cheap heuristic surrogate in place of EMD.

Let CH_p denote the Chamfer distance under the ℓ_p norm. We empirically show that CH_p for $p = 1.5$ serves as a better surrogate for EMD than $p = 1, 2$ on a family of synthetic, naturally motivated point-cloud pairs. In this experiment, we generate point-clouds pairs (A, B) in $\mathbb{R}^{d=64}$, each containing $n = 40$ points. For each pair, we first sample a reference cloud B i.i.d. from the standard Gaussian $\mathcal{N}(0, I)$. We then generate A from B using one of the following three stochastic corruption approaches. In all approaches, we use a parameter $t \in [0, 1]$ to controls the corruption strength:

1. **Dense-shift.** We first draw a random unit vector u and define a shift magnitude $m = m(t)$. Then every point $a_i \in A$ is shifted from b_i by an offset vector $(m + \alpha_i)u + \beta_i$, where α_i, β_i are Gaussian noise sampled i.i.d. for each i.

Table 1: Spearman correlation between symmetric ℓ_p-Chamfer distance and exact Euclidean EMD across 5 random seeds. Each seed uses $n = 40$, $d = 64$, and 450 generated point-cloud pairs. In every seed, the intermediate choice $p = 1.5$ attains the highest correlation.

Seed	Average Spearman ($p = 1.0$)	Average Spearman ($p = 1.5$)	Average Spearman ($p = 2.0$)
0	0.8783	**0.8817**	0.8802
1	0.8895	**0.8916**	0.8870
2	0.8898	**0.8917**	0.8889
3	0.8894	**0.8903**	0.8896
4	0.8761	**0.8786**	0.8749

2. **Many-to-one-collapse.** We start from A being a global shift of B by the same random offset vector. Then we choose a random fraction $f = f(t) + \alpha$ of points in A to collapse, and a random fraction $g = g(t) + \beta$ of points in B to collapse to, where α, β are again Gaussian noise, and $f \gg g$. For each chosen point in A, it is moved to one of the chosen points in B uniformly at random. This creates a many-to-one collapse of a subset of points toward a small set of hubs, plus a small common drift.
3. **Sparse-spikes.** We start from $A = B$. Then we choose a small random fraction $a = a(t) + \alpha$ of points in A to corrupt, and for each chosen point, we choose a small random subset of coordinates, and perturb it by a Gaussian offset whose variance is $b = b(t) + \beta$, where α, β are again Gaussian noise. Thus only a sparse subset of points is perturbed.

These three types of corruptions are intended to capture, respectively, diffuse geometric mismatch, loss of correspondence structure, and sparse outlier-like corruption. To make the three corruption types similarly representative, we do not directly use the same parameter t across families. Instead, we compute a calibration curve for $t \mapsto$ Empirical Expectation of EMD. We then choose 10 target EMD levels, equally spaced in the interval $[0.8, 2.6]$, and for each family invert the calibration curve to obtain the value of t that approximately matches that target EMD level.

For each random seed, we use: 10 target EMD levels, 3 corruption families, and 15 independent samples – a total of 450 point-cloud pairs per seed. For each generated pairs (A, B) , we compute EMD and a symmetric version of CH_p: $\mathsf{CH}_p(A, B) + \mathsf{CH}_p(B, A)$, for $p = 1, 1.5, 2$. This induces a ranking among 450 pairs for each measurement. To evaluate how well CH_p serves as a surrogate for EMD, we compute the Spearman rank correlation between the ranking induced by CH_p and the ranking induced by EMD. A larger Spearman correlation means that the corresponding Chamfer variant better preserves the ordering induced by exact EMD, which is the relevant notion of surrogate quality here. Our result is shown in Table 1.

We observe that $p = 1.5$ is the best-performing choice in every seed, i.e., it consistently achieves the strongest monotone agreement with exact EMD. We stress that this is exactly the type of effect one should expect from an intermediate norm. The ℓ_1-Chamfer distance is relatively more influenced by diffuse high-dimensional shifts, while ℓ_2-Chamfer is relatively more affected by sparse large perturbations and mismatch structure created by collapse. The intermediate choice $p = 1.5$ provides a compromise between these two behaviors, which in this benchmark leads to a consistently better surrogate for exact Euclidean transport. This experiment provides concrete evidence: there exist natural high-dimensional regimes in which an intermediate $p \in (1, 2)$ performs better than either endpoint. This motivates our study of CH_p for $p \in (1, 2)$.

1.2 Our Contributions

In this work, we present fast algorithms for approximating the Chamfer distance for the general case of $1 < p \leq 2$. We also empirically validate our results by experiments on both synthetic and real-world data. We leverage embedding and dimensionality reduction techniques tailored to different ℓ_p norms. Our results are summarized in Table 2.

General Case ($1 < p < 2$). For the general case, we utilize lopsided embeddings from ℓ_p^d to ℓ_1^m for $m = \mathrm{poly}(\log n, \varepsilon^{-1})$ [LLW23]. These embeddings guarantee non-contraction for all $\mathcal{O}(n^2)$ pairwise distances between points in sets A and B, while preserving nearest neighbor distances with bounded expected error. Focusing on the high-dimensional case where $d \geq \mathrm{poly}(\log n)$, we show that Fast Matrix Multiplication (FMM) techniques [SW26] can apply the dense embedding matrix efficiently, resulting in a runtime near-linear in the input dimension d. These lead to $\mathcal{O}(nd \cdot$

Table 2: Comparison of $(1+\varepsilon)$-approximate Chamfer Distance Algorithms for n points in d dimensions. Our work is the first to achieve near-linear time for the general ℓ_p case, extend the ℓ_1 state-of-the-art runtime to ℓ_2 and ColBERT MaxSim, and support space-efficient streaming. The time complexity of general ℓ_p norms and MaxSim assumes $d \geq \operatorname{poly}\log(n)$.

Algorithm / Reference	Supported Metric	Time Complexity	Space Complexity	Dynamic Streaming
Naive Exact	Any ℓ_p ($p \geq 1$)	$\mathcal{O}(n^2 d)$	$\mathcal{O}(nd)$	$\times$
[BIJ+23]	ℓ_1, ℓ_2	$\mathcal{O}\left(nd \log(n)/\varepsilon^2\right)$	$\mathcal{O}(nd)$	$\times$
[FI25]	ℓ_1	$\mathcal{O}\left(nd(\log\log n + \log(1/\varepsilon))/\varepsilon^2\right)$	$\mathcal{O}(nd)$	$\times$
Ours (Theorem 3.4)	**General** ℓ_p ($1 < p < 2$)	$\mathcal{O}\left(nd \cdot \operatorname{poly}(\log\log n, \log(1/\varepsilon))/\varepsilon^2\right)$	$\mathcal{O}(n \cdot \operatorname{poly}(\log n))$	✓
Ours (Theorems 4.5 & 5.2)	ℓ_2 **and MaxSim**	$\mathcal{O}\left(nd(\log\log n + \log(1/\varepsilon))/\varepsilon^2\right)$	$\mathcal{O}(n \log^2 n)$	✓
Our Lower Bound (Theorem 5.3)	**Any** ℓ_p ($p \geq 1$)	—	$\Omega(n)$ **bits**	—

$\operatorname{poly}(\log\log n, \log(1/\varepsilon))/\varepsilon^2)$ time algorithm for estimating the Chamfer distance for $p \in (1,2)$. We remark that this approach also works for $p = 2$, but by opening the black-box of the algorithm in [FI25] and combining it with appropriate dimension reduction, we can achieve an even faster runtime for $p = 2$.

Euclidean Case ($p = 2$). For the ℓ_2 norm, we give an algorithm that matches the best known ℓ_1 runtime, solving an open problem in [FI25]. To this end, we use a Fast Johnson-Lindenstrauss Transform (FJLT) based on structured Toeplitz matrices [KW11, HV11, Vyb11] to reduce the dimension from d to $m = \mathcal{O}(\log^2 n/\varepsilon^2)$, and then re-analyze the algorithmic framework of [FI25] in the reduced dimension in the ℓ_2 space. Although the algorithm in [FI25] is designed to approximate the Chamfer distance in the ℓ_1 norm, we show that it actually gives a good approximation in the reduced-dimensional ℓ_2 space. Related ideas of analyzing tree embeddings for ℓ_2 norm appeared in [LRSS25]. However, their tree construction takes quadratic time, which is in contrast to our near-linear-time algorithm framework. Combining this with the $\mathcal{O}(d \log m)$-time Toeplitz matrix multiplication, we achieve a total runtime of $\mathcal{O}(nd(\log\log n + \log(1/\varepsilon))/\varepsilon^2)$, matching the ℓ_1 result and improving the previous ℓ_2 bound.

Extensions and Theoretical Limits. Beyond geometric ℓ_p norms, we show that our techniques also apply to similarity measures used in modern LLM retrieval. In particular, we consider the ColBERT MaxSim score, defined for token embeddings on the unit sphere as

$$\operatorname{MaxSim}(A, B) = \sum_{a \in A} \max_{b \in B} \langle a, b \rangle.$$

Using the identity $\|a - b\|_2^2 = 2 - 2\langle a, b \rangle$ for normalized vectors, we show that our Fast ℓ_2 Chamfer algorithm (Theorem 4.4) yields a near-linear-time *additive* $\mathcal{O}(\varepsilon)$-approximation to $\operatorname{MaxSim}(A, B)$.

We also study the Chamfer distance in the *dynamic turnstile streaming* model, where point coordinates receive additive updates. Since both of our dimensionality reduction maps (Toeplitz FJLT and lopsided embeddings) are linear, they can be maintained online under updates, allowing the algorithm to store only the low-dimensional sketches and run the downstream solver at the end of the stream, reducing space from $\mathcal{O}(nd)$ to $\mathcal{O}(n \operatorname{poly}(\log n))$. Complementing this upper bound, we prove an unconditional $\Omega(n)$ space lower bound for any randomized constant-pass streaming algorithm achieving a constant-factor approximation (for any $p \geq 1$), via a reduction from two-party set disjointness. We note that while a dynamic version of the Chamfer distance problem has been recently studied in [GJK+25], they assume that the sets undergo insertions and deletions of *points*, while our setting allows arbitrary updates to the coordinates of points.

2 Preliminaries

2.1 Fast Sketch Application via FMM

We need to efficiently apply dense sketch matrices used in lopsided embeddings. This is done using the Fast Matrix Multiplication (FMM) technique from [SW26].

Lemma 2.1 (Fast Sketch Application, Adapted from Theorem C.1 and C.3 of [SW26]). *Let $T \in \mathbb{R}^{m \times d}$ be a dense sketch matrix. There exists a constant $K \approx 11.76$ (related to the exponents of rectangular matrix multiplication) such that if $d \geq m^{2K}$, the product Tv for any vector $v \in \mathbb{R}^d$ can be computed in time $\mathcal{O}(d \cdot \operatorname{poly}\log(m))$.*

2.2 Fast Johnson-Lindenstrauss Transform (FJLT)

For the Euclidean case, we use FJLT variants over matrices in $\mathbb{R}^{m\times d}$ that provide the guarantees of the JL lemma [JL84]. We highlight that we require the FJLT runtime to be $\mathcal{O}(d\log m)$, which is achieved by Toeplitz matrices [KW11, HV11, Vyb11], instead of the more standard $\mathcal{O}(d\log d)$ bound often associated with full-dimensional FFTs. This is crucial to obtain the desired $\mathrm{poly}\log\log(n)$ overhead in the final runtime.

Lemma 2.2 (Fast Johnson-Lindenstrauss Transform, [KW11, HV11, Vyb11])**.** *Let $V\subset\mathbb{R}^d$ be a set of N points. Let $\varepsilon,\delta\in(0,1/2)$. There exists a distribution over structured matrices $T\in\mathbb{R}^{m\times d}$ where $m=\mathcal{O}(\log^2(n/\delta)/\varepsilon^2)$, such that with probability $1-\delta$, for all $u,v\in V$:*

$$(1-\varepsilon)\|u-v\|_2\le\|T(u-v)\|_2\le(1+\varepsilon)\|u-v\|_2.$$

Moreover, Tv can be computed in $\mathcal{O}(d\log m)$ time.

2.3 The Framework of [FI25]

The algorithm in [FI25] consists of three main components: QuadTrees, Tournaments, and Rejection Sampling. In particular, the QuadTrees compute crude estimates $\mathcal{D}_a$ of $\mathsf{opt}_a:=\min_{b\in B}\|a-b\|$ for all $a\in A$. And the efficiency of the overall algorithm depends on the approximation factor achieved by the QuadTree step. If QuadTree provides an F-approximation in expectation (i.e., $\mathbb{E}[\mathcal{D}_a]\le F\cdot\mathsf{opt}_a$), then the overall runtime of the algorithm is dominated by $\mathcal{O}(nd\log\log n+n(d+F/\varepsilon^2)\log(F/\varepsilon^2))$.

For the ℓ_1 metric, the authors prove $F=\mathcal{O}(\min(d,\log n))$. This leads to $\mathcal{O}(nd\log\log n/\varepsilon^2)$ total runtime. In Section 4, we will re-analyze the approximation factor F when the framework is applied to the ℓ_2 metric.

We also note that given the existence of p-stable distributions, the rest of the algorithm in [FI25] works the same regardless of the underlying p-norm. These can be summarized into the following theorem:

Theorem 2.3 (Adapted from [FI25])**.** *Given a QuadTree procedure that achieves an expected approximation factor F, there exists an algorithm that, with .9 probability, computes a $(1+\varepsilon)$-approximation to the Chamfer distance in total time: $\mathcal{O}(nd\log\log n+n(d+F/\varepsilon^2)\log(F/\varepsilon^2))$.*

3 The General Case ($1<p<2$)

For $1<p<2$, we use lopsided embeddings into ℓ_1. Unlike the ℓ_2 case where the JL Lemma guarantees two-sided strong concentration, the lopsided embeddings are weaker, providing only strong non-contraction (but not dilation) and bounded expected error. However, we show that this suffices for our need, due to the fact that the Chamfer distance is defined as the sum of the nearest neighbor distance.

3.1 Lopsided ℓ_p to ℓ_1 Embedding for Finite Sets

We utilize the results based on p-stable distributions from [LLW23].

Lemma 3.1 (Lopsided Embedding for Finite Sets, adapted from [LLW23])**.** *Suppose $p\in(1,2)$ is constant. Let $\varepsilon,\delta\in(0,1/2)$ and $V\subset\mathbb{R}^d$ with $|V|=n$. There exists a distribution over matrices $T\in\mathbb{R}^{m\times d}$, where $m=\mathrm{poly}(\log n,\varepsilon^{-1},\delta^{-1})$, satisfying:*

1. ***(Expected Error Bound):** For any fixed $v\in\mathbb{R}^d$, $\mathbb{E}[|\|Tv\|_1-\|v\|_p|]\le\varepsilon\|v\|_p$.*
2. ***(Contraction on V):** With probability $1-\delta$, for all $v\in V$, $\|Tv\|_1\ge(1-\varepsilon)\|v\|_p$.*

This is implied by the proof to Lemma 18 of [LLW23]. We note that the original Lemma 18 in [LLW23] is stated for the target space ℓ_q with $q\in(1,2)$, whereas here we need the lemma to hold for ℓ_1, i.e. the endpoint $q=1$. Thus, we cannot use the prior theorem entirely as a black box: we revisit the proof and verify that the argument extends to the target needed here. We delay the proof to Appendix A due to the page limit.

3.2 The Algorithm

The algorithm is via a reduction from the ℓ_p^d instance to the ℓ_1^m instance. We show that the weak guarantees provided by the lopsided embeddings imply good preservation of the Chamfer distance. We need to ensure non-contraction for all n^2 pairwise differences. Consequently, we will set m in the embedding (Lemma 3.1) to be polylogarithmic in n. Note that the embedding matrix is dense, thus to speed up the embedding time, we will use the FMM technique given in Lemma 2.1.

Theorem 3.2. *Let $1 < p < 2$ be constant, $A, B \subset \mathbb{R}^d$ of size n, and $\varepsilon \in (0, 1/2)$. For some $m = \mathrm{poly}(\log n, \varepsilon^{-1})$, if $d \geq m^{2K}$ (where $K \approx 11.76$), then there is an algorithm that computes a $(1+\varepsilon)$-approximation to $CH_p(A, B)$ in total time:*

$$\mathcal{O}\left(nd \cdot \mathrm{poly}\log(m) + T_1(n, m, \varepsilon)\right),$$

with .9 probability, where $T_1(n, m, \varepsilon)$ is the time to $(1+\varepsilon)$-approximation the ℓ_1 Chamfer distance in $\mathbb{R}^m$ with high probability.

Due to the page limit, we delay the proof to Theorem 3.2 to the appendix Section B.

Remark 3.3. We remark that using a standard median trick, the .9 success probability in Theorem 3.2 can be boosted to $1 - \delta$ for any $\delta \in (0, 1)$, by repeating the algorithm for $\log(1/\delta)$ time.

Combining this with the known Chamfer algorithm under the ℓ_1 norm, we get the following result:

Theorem 3.4. *Let $1 < p < 2$ be constant, $A, B \subset \mathbb{R}^d$ of size n, and $\varepsilon \in (0, 1/2)$. For some $m = \mathrm{poly}(\log n, \varepsilon^{-1})$, if $d \geq m$, then there is an algorithm that computes a $(1+\varepsilon)$-approximation to $CH_p(A, B)$ in total time $\mathcal{O}\left(nd \cdot \mathrm{poly}(\log\log n + \log(1/\varepsilon))/\varepsilon^2\right)$ with .9 probability.*

4 The Euclidean Case ($p = 2$)

We now analyze the Euclidean case by combining the FJLT with a modified analysis of [FI25]. As outlined in the introduction, the strategy is to first reduce the dimension using a Toeplitz structure, and then analyze the performance of the QuadTree estimator in the reduced dimension.

In this section, we let m denote the target dimension obtained after the FJLT projection, where the point sets now reside. We apply Lemma 2.2 with the failure probability δ set to .99 and approximation factor set to 1. Thus we can set $m = \mathcal{O}(\log^2 n)$.

4.1 QuadTree Analysis in ℓ_2

We first analyze the approximation factor F_{ℓ_2} achieved by the QuadTree procedure of [FI25] when applied directly to the ℓ_2 metric in dimension m. A quadtree of depth t is defined by a random offset vector $z \sim [0, 2^t]^m$. It assigns every point in $\mathbb{R}^m$ with a sequence of t hash values, using t nested grids shifted by z. Concretely, for any $a \in \mathbb{R}^m$ and any integer k such that $0 \leq k \leq t$, it hashes

$$h_k(a) := (\lceil \frac{a_1 + z_1}{2^k} \rceil, \lceil \frac{a_2 + z_2}{2^k} \rceil, \cdots, \lceil \frac{a_m + z_m}{2^k} \rceil).$$

[FI25] used two independent quadtrees to estimate the nearest neighbor distance for all $a \in A$ in ℓ_1, i.e. estimating $\min_{b \in A} \|a - b\|_1$ for all $a \in A$. We follow a similar proof structure and analyzes the behavior of quadtrees in ℓ_2. Due to the page limit, we delay the proofs to the appendix Section C.

Lemma 4.1. *For all $a, b \in \mathbb{R}^m$ and $0 \leq k \leq t$, if $h_k(a) = h_k(b)$, then $\|a - b\|_2 \leq 2^k \sqrt{m}$.*

Lemma 4.2. *With probability $1 - \mathcal{O}(1/n)$, the following holds simultaneously for all a, b, k: If $h_k(a) = h_k(b)$, then $\|a - b\|_2 \leq 2^k \cdot 3 \log n$.*

With the above lemmas, we can show that in expectation, two quadtrees output good estimation to $\mathsf{opt}_a := \min_{b \in A} \|a - b\|_2$ for all $a \in A$. The estimator $\mathcal{D}_a$ is defined as:

1. Identifying the smallest $\tilde{k}$ such that $h_{\tilde{k}}(a) = h_{\tilde{k}}(b)$ for some $b \in B$ across the two quadtrees.
2. Assigning $\mathcal{D}_a := \|a - b\|_2$ for arbitrary b such that $h_{\tilde{k}}(a) = h_{\tilde{k}}(b)$.

Lemma 4.3. *With probability $1 - \mathcal{O}(1/n)$, it holds for all $a \in A$ that $\mathbb{E}\mathcal{D}_a \leq F \cdot \mathsf{opt}(a)$, where $F = \mathcal{O}(\min(m, \sqrt{m} \log n))$.*

4.2 Algorithm for ℓ_2 Chamfer Distance

As mentioned in the introduction, it can be verified that when shifting from the ℓ_1 norm to ℓ_2, the rest of the algorithm in [FI25] works the same. Thus all it remains is the combine the FJLT with the analysis above.

Theorem 4.4 (Fast ℓ_2 Chamfer Distance)**.** *Given $A, B \subset \mathbb{R}^d$ of size n, and $\varepsilon \in (0, 1/2)$. There is an algorithm that computes a $(1+\varepsilon)$-approximation to $CH_2(A, B)$ with .9 probability in time $\mathcal{O}\left(nd(\log\log n + \log(1/\varepsilon))/\varepsilon^2\right)$.*

Proof. When the dimension $d \leq \mathcal{O}(\log^2 n)$, we skip the FJLT and directly plug in $m = d$ to Lemma 4.3. This tells us that the Quadtree step of the algorithm outputs estimators with expected distortion $F = \mathcal{O}(\min(d, \sqrt{d}\log n)) = \mathcal{O}(\log^2 n)$. Applying Theorem 2.3 on this F, we get a total runtime of $\mathcal{O}(nd(\log\log n + \log(1/\varepsilon))/\varepsilon^2)$ as claimed.

On the other hand, in the high-dimension regime ($d \geq \Omega(\log^2 n)$), we choose appropriate $m = \mathcal{O}(\log^2 n)$ and runs the FJLT from Lemma 2.2. This takes $\mathcal{O}(nd\log\log n)$ time. Then we are back in the low-dimensional case, where we spend an additive $\mathcal{O}(nd(\log\log n + \log(1/\varepsilon))/\varepsilon^2)$ time to complete running the algorithm.

□

5 Extensions and Theoretical Limits

To substantiate the broad applicability of our techniques and to map the theoretical boundaries of the approximation, we introduce the application to non-geometric norms (LLM Retrieval metrics) and prove conditional lower bounds for the memory required by streaming models. Due to the page limit, we delay the proofs to the appendix Section D.

5.1 MaxSim and the ColBERT

In Natural Language Processing, the ColBERT model utilizes a similarity measure intrinsically related to the Chamfer distance.

Definition 5.1 (ColBERT MaxSim)**.** For embedded document tokens $A, B \subset \mathbb{S}^{d-1}$ on the unit hypersphere, the ColBERT score is defined as:

$$\text{MaxSim}(A, B) = \sum_{a \in A} \max_{b \in B} \langle a, b \rangle. \tag{1}$$

Theorem 5.2 (Fast Approximate ColBERT Retrieval)**.** *For normalized sets $A, B \subset \mathbb{S}^{d-1}$, the Fast ℓ_2 Chamfer distance algorithm (Theorem 4.4) can be modified to output an additive $\mathcal{O}(\varepsilon n)$-approximation to MaxSim$(A, B)$, and for $d \geq \text{poly}\log(n)$, it runs in $\mathcal{O}\left(nd(\log\log n + \log(1/\varepsilon))/\varepsilon^2\right)$ time.*

5.2 Streaming Algorithm and Space Complexity Lower Bound

In large-scale data settings, the input sets A and B are often not static but arrive as a data stream with coordinate updates. We analyze the Chamfer distance under the dynamic turnstile stream model, where points receive additive updates to their coordinates.

Upper Bound via Linear Sketching: Our dimensionality reduction techniques are naturally suited for this environment. Since the projection matrices T used in our algorithms (both Toeplitz FJLT and Lopsided embeddings) are linear, data-oblivious maps, they trivially support additive updates: $T(v+\Delta) = Tv + T\Delta$. This allows the algorithm to maintain the low-dimensional embeddings online. Once the stream terminates, the solver of [FI25] is applied black-box to the sketches. This dynamic algorithm reduces the memory footprint from $\mathcal{O}(nd)$ raw coordinates to $\mathcal{O}(nm) = \mathcal{O}(n \cdot \text{poly}(\log n))$ space.

We formalize that this $\tilde{\mathcal{O}}(n)$ space complexity is near-optimal by proving an unconditional $\Omega(n)$ space lower bound via communication complexity.

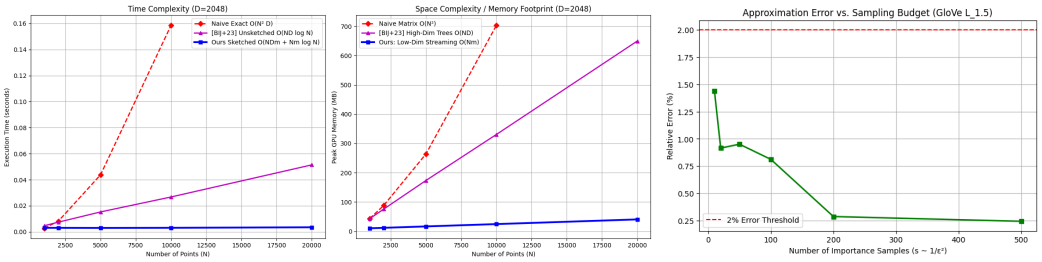

Figure 1: **Left:** Runtime scaling ($D = 2048$). Our method remains flat near the GPU noise-floor due to efficient sketching. **Middle:** Peak GPU memory ($D = 2048$). Our method validates the $\tilde{\mathcal{O}}(nm)$ space claim, using less than 45 MB while previous methods exceed 600 MB. **Right:** Approximation error on the GloVe dataset for the $\ell_{1.5}$ Chamfer distance, showing sub-1% error with very few importance samples.

Theorem 5.3 ($\Omega(n)$ Space Lower Bound for Streaming Chamfer)**.** *Any randomized constant-pass streaming algorithm that computes a constant-factor approximation to the ℓ_p Chamfer distance ($p \geq 1$) for dynamically updated point sets requires $\Omega(n)$ bits of space.*

6 Experiments

We evaluate the performance of our algorithm on both synthetic high-dimensional data and real-world natural language embeddings. We focus on three dimensions of performance: time complexity, space complexity, and approximation accuracy (validating the combination of lopsided embeddings / FJLT with the Chamfer algorithm. [1]).

6.1 Experimental Setup

Baselines: We compare our method against two baselines. First, the **Naive Exact** method, which computes the full $\mathcal{O}(n^2 d)$ distance matrix. Second, the unsketched Chamfer algorithm from [BIJ+23], which runs QuadTrees and uses the outputted estimator to perform importance sampling in the original high-dimensional space without our random projections, theoretically taking $\mathcal{O}(nd \log n)$ time.

Datasets: For computational scaling experiments, we generate synthetic sets $A, B \subset \mathbb{R}^d$ with $d = 2048$ to accentuate the high-dimensional bottlenecks. For accuracy validation, we use the 300-dimensional GloVe dataset [KSKW15] consisting of word embeddings, a standard benchmark for NLP metrics like the ColBERT retrieval score (Definition 5.1). All experiments were conducted on a single NVIDIA T4 GPU. Our algorithm projects to a target dimension of $m = 64$ and uses $s \approx \mathcal{O}(1/\varepsilon^2)$ importance samples for the importance sampling step of [BIJ+23].

6.2 Computational Efficiency (Time Scaling)

We evaluate the wall-clock execution time as n increases from $1,000$ to $20,000$ with a fixed dimension $d = 2048$. As shown in Figure 1 (Left), the exact method scales quadratically, running out of memory beyond $n = 10,000$.

The unsketched [BIJ+23] baseline scales linearly but incurs a high constant overhead due to operating directly in $\mathbb{R}^{2048}$, taking 51ms at $n = 20,000$. In contrast, our algorithm runs in 5ms for $n = 20,000$.

6.3 Space Complexity and Streaming Feasibility

In Section 5.2, we claimed our algorithm supports dynamic streams via a low-memory footprint of $\tilde{\mathcal{O}}(nm)$. Figure 1 (Middle) validates this claim by tracking Peak GPU Memory allocation.

[1] In our implementations, instead of using [FI25] as the black-box Chamfer algorithm as in our theoretical proofs, we use [BIJ+23], because efficient implementation of [FI25] requires optimizing the efficiency of bit-level operations, which is beyond the focus of our experiments. This also demonstrates that our embeddings and dimension reduction approach accelerates any downstream ℓ_1 solver.

Table 3: Mean relative error (%) for [BIJ+23] and our method across different values of N. Results are reported as mean $\pm$ standard deviation.

N	[BIJ+23] Mean Relative Error (%)	Ours Mean Relative Error (%)
1000	0.055 ± 0.023	0.614 ± 0.412
2000	0.066 ± 0.062	0.559 ± 0.058
5000	0.067 ± 0.040	0.622 ± 0.249
10000	0.027 ± 0.025	0.397 ± 0.264

The exact distance matrix requires $\mathcal{O}(n^2)$ space, immediately exhausting memory. Also, [BIJ+23] must retain the high-dimensional vectors A and B to construct the quadtrees, consuming 649 MB at $n = 20,000$. Our method operates strictly on the sketches. By discarding the high-dimensional inputs after the oblivious random projection, our peak memory consumption is just 41 MB at $n = 20,000$, which is roughly $15\times$ reduction over the linear unsketched baseline. This empirically confirms the efficiency of our algorithm in memory-constrained environments.

6.4 Accuracy and Lopsided Embedding Validation

In Table 3, we record the relative approximation error of our algorithm against the unsketched baseline [BIJ+23]. As expected, the unsketched baseline achieves smaller error because it operates directly in the ambient space $\mathbb{R}^{2048}$, whereas our method introduces distortion through dimensionality reduction. At the same time, the distortion remains small in practice: in these experiments, the relative error of our method stays below 1%, with the largest mean error peaking at just 0.622%.

We also validate the accuracy of combining lopsided $\ell_p \to \ell_1$ embeddings with the Chamfer algorithm (Theorem 3.2, we compute the $\ell_{1.5}$ Chamfer distance on GloVe word embeddings.

Figure 1 (Right) plots the relative approximation error against the importance sampling budget s. Following our theoretical guarantees, the error decays rapidly. With a sample budget of just $s = 10$, the error is 1.44%. Increasing the budget to $s = 100$ drops the error to 0.81%, and at $s = 500$, the error reaches 0.24%.

This confirms one of the theoretical novelties of our work, that the weak guarantees of the lopsided embeddings (non-contraction and bounded first moment, Lemma 3.1) are sufficient to preserve the Chamfer distance from ℓ_p to ℓ_1.

7 Conclusion

We give efficient algorithms for approximating the Chamfer distance. For $1 < p < 2$, we use lopsided ℓ_p to ℓ_1 embeddings and show that they reduce the ℓ_p Chamfer distance instance to ℓ_1 with small distortion. This lead to $\mathcal{O}(nd \cdot \mathrm{poly}(\log\log n, \log(1/\varepsilon))/\varepsilon^2)$ time algorithm for estimating the Chamfer distance. For $p = 2$, we successfully resolved the open problem by employing the FJLT using Toeplitz matrices [KW11] and integrating it with the framework of [FI25]. This yields a runtime of $\mathcal{O}(nd(\log\log n + \log(1/\varepsilon))/\varepsilon^2)$, matching the state-of-the-art ℓ_1 result.

We also connect our fast ℓ_2 Chamfer algorithms to MaxSim-style LLM retrieval (e.g., ColBERT), obtaining a near-linear-time additive approximation. Finally, in the dynamic turnstile model, the linearity of our embeddings gives $\tilde{\mathcal{O}}(n)$-space streaming algorithms, and we prove a matching $\Omega(n)$ space lower bound for any randomized constant-pass constant-factor approximation (for all $p \geq 1$).

Regarding the limitations of our work, in the plain model, our result leaves a factor of $\log\log n$ between our upper bound and the naïve $\Omega(nd)$ lower bound. It would be interesting to remove this overhead, or else prove a matching lower bound. Empirically, our evaluation is promising but limited: it uses synthetic data plus GloVe, a single GPU setup. It would be interesting to evaluate our algorithm on a broader range of large-scale point-cloud and retrieval datasets.

Impact Statement This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

[ACRX22] Pankaj K Agarwal, Hsien-Chih Chang, Sharath Raghvendra, and Allen Xiao. Deterministic, near-linear ϵ-approximation algorithm for geometric bipartite matching. In *Proceedings of the 54nd ACM Symposium on the Theory of Computing (STOC '2022)*, 2022.

[AFP+17] Pankaj Agarwal, Kyle Fox, Debmalya Panigrahi, Kasturi Varadarajan, and Allen Xiao. Faster algorithms for the geometric transportation problem. In *Proceedings of the 33rd International Symposium on Computational Geometry (SOCG '2017)*, 2017.

[AIK08] Alexandr Andoni, Piotr Indyk, and Robert Krauthgamer. Earth mover distance over high-dimensional spaces. In *Proceedings of the 19th ACM-SIAM Symposium on Discrete Algorithms (SODA '2008)*, pages 343–352, 2008.

[AM19] Kubilay Atasu and Thomas Mittelholzer. Linear-complexity data-parallel earth mover's distance approximations. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 364–373. PMLR, 09–15 Jun 2019.

[AMN+98] Sunil Arya, David M. Mount, Nathan S. Netanyahu, Ruth Silverman, and Angela Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *J. ACM*, 45(6):891–923, November 1998.

[AR15] Alexandr Andoni and Ilya Razenshteyn. Optimal data-dependent hashing for approximate near neighbors. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 793–801, 2015.

[AS03] Vassilis Athitsos and Stan Sclaroff. Estimating 3d hand pose from a cluttered image. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pages II–432. IEEE, 2003.

[AS14] Pankaj K Agarwal and R Sharathkumar. Approximation algorithms for bipartite matching with metric and geometric costs. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 555–564, 2014.

[AZ23] Alexandr Andoni and Hengjie Zhang. Sub-quadratic (1+ ϵ)-approximate euclidean spanners, with applications. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 98–112. IEEE, 2023.

[BCAJW25] Lorenzo Beretta, Vincent Cohen-Addad, Rajesh Jayaram, and Erik Waingarten. Approximating high-dimensional earth mover's distance as fast as closest pair, 2025.

[BIJ+23] Ainesh Bakshi, Piotr Indyk, Rajesh Jayaram, Sandeep Silwal, and Erik Waingarten. A near-linear time algorithm for the chamfer distance, 2023.

[BR24] Lorenzo Beretta and Aviad Rubinstein. Approximate earth mover's distance in truly-subquadratic time. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 47–58, 2024.

[Cha02] Moses Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the 34th ACM Symposium on the Theory of Computing (STOC '2002)*, pages 380–388, 2002.

[EK23] Yury Elkin and Vitaliy Kurlin. A new near-linear time algorithm for k-nearest neighbor search using a compressed cover tree. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 9267–9311. PMLR, 23–29 Jul 2023.

[FI25] Ying Feng and Piotr Indyk. Even faster algorithm for the chamfer distance. In *International Colloquium on Automata, Languages and Programming*, 2025.

[FSG17a] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017.

[FSG17b] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017.

[GJK+25] Gramoz Goranci, Shaofeng H-C Jiang, Peter Kiss, Eva Szilagyi, and Qiaoyuan Yang. Fully dynamic algorithms for chamfer distance. In *The 39th Annual Conference on Neural Information Processing Systems*, 2025.

[HV11] Aicke Hinrichs and Jan Vybíral. Johnson–lindenstrauss lemma for circulant matrices. *Random Structures & Algorithms*, 39(3):391–398, 2011.

[IT03] Piotr Indyk and Nitin Thaper. Fast color image retrieval via embeddings. In *Workshop on Statistical and Computational Theories of Vision (at ICCV)*, 2003.

[JL84] William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206), 1984.

[JSQJ18] Li Jiang, Shaoshuai Shi, Xiaojuan Qi, and Jiaya Jia. Gal: Geometric adversarial loss for single-view 3d-object reconstruction. In *Proceedings of the European conference on computer vision (ECCV)*, pages 802–816, 2018.

[KNP19] Andrey Boris Khesin, Aleksandar Nikolov, and Dmitry Paramonov. Preconditioning for the geometric transportation problem. In *Proceedings of the 35th International Symposium on Computational Geometry (SoCG '2019)*, 2019.

[KSKW15] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966. PMLR, 2015.

[KW11] Felix Krahmer and Rachel Ward. New and improved johnson–lindenstrauss embeddings via the restricted isometry property. *Journal on Mathematical Analysis*, 43(3):1269–1281, 2011.

[KZ20] Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48, 2020.

[LLW23] Yi Li, Honghao Lin, and David P. Woodruff. ℓ_p-regression in the arbitrary partition model of communication, 2023.

[LRSS25] Nathaniel Lahn, Sharath Raghvendra, Emma Saarinen, and Pouyan Shirzadian. Scalable approximation algorithms for p-Wasserstein distance and its variants. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 32188–32205. PMLR, 13–19 Jul 2025.

[SA20] R. Sharathkumar and Pankaj K. Agarwal. A near-ϵ-approximation algorithm for bipartite geometric matching. *Journal of the ACM*, 67(3):18:1–18:19, 2020.

[SMFW04] Erik B Sudderth, Michael I Mandel, William T Freeman, and Alan S Willsky. Visual hand tracking using nonparametric belief propagation. In *2004 Conference on Computer Vision and Pattern Recognition Workshop*, pages 189–189. IEEE, 2004.

[SW26] William Swartworth and David P Woodruff. Sketching faster than dimension times update time. *Manuscript*, 2026.

[Vyb11] Jan Vybíral. A variant of the johnson–lindenstrauss lemma for circulant matrices. *Journal of Functional Analysis*, 260(4):1096–1105, 2011.

[WCL+19] Ziyu Wan, Dongdong Chen, Yan Li, Xingguang Yan, Junge Zhang, Yizhou Yu, and Jing Liao. Transductive zero-shot learning with visual structure constraint. *Advances in neural information processing systems*, 32, 2019.

A Proof to Lemma 3.1

Lemma A.1 (Repeating Lemma 3.1)**.** *Suppose $p \in (1,2)$ is constant. Let $\varepsilon, \delta \in (0,1/2)$ and $V \subset \mathbb{R}^d$ with $|V| = n$. There exists a distribution over matrices $T \in \mathbb{R}^{m \times d}$, where $m = \mathrm{poly}(\log n, \varepsilon^{-1}, \delta^{-1})$, satisfying:*

1. ***(Expected Error Bound):** For any fixed $v \in \mathbb{R}^d$, $\mathbb{E}[|\|Tv\|_1 - \|v\|_p|] \le \varepsilon \|v\|_p$.*
2. ***(Contraction on V):** With probability $1-\delta$, for all $v \in V$, $\|Tv\|_1 \ge (1-\varepsilon)\|v\|_p$.*

Proof. In Lemma 18 of [LLW23], this is proven for $\|Tv\|_1$ replaced with $\|Tv\|_q$ for any $q \in (1,p)$. We trace the proof of Lemma 18 and verify that setting $q = 1$ does not violates any step of the proof.

Step 1. Moment Existence. We define T to be a matrix whose entries are i.i.d. p-stable random variables. It is a fundamental property of p-stable distributions that the q-th moment $\mathbb{E}[|X|^q]$ is finite if and only if $q < p$. This implies Lemma 25 of [LLW23] for $q = 1$, that

$$\mathbb{E}[\|Tv\|_1] = \mathbb{E}[\sum_{i=1}^{m} |(Tv)_i|] = m\alpha_p \|v\|_p$$

for some constant α_p depending only on p. Scaling T by $1/(m \cdot \alpha_p)$ gives $\mathbb{E}[|\sum_{i=1}^m (Tv)_i|] = \|v\|_p$. Similarly, it gives $\mathbb{E}[|(Tv)_i|^r] = \beta_{p,r}\|v\|_p^r / m^r$ for every i for some constant α_p depending only on p, r.

Step 2. Concentration via von Bahr-Esseen. To show concentration, the proof selects a higher moment $r \in (1,p)$. The proof then considers the random variables $X_i = |(Tv)_i| - \mathbb{E}[|(Tv)_i|]$, which are independent and mean-zero. We apply the von Bahr-Esseen inequality (Lemma 27 of [LLW23]), which bounds $\mathbb{E}[|\sum X_i|^s]$ for $1 \le s \le 2$. Here, the exponent is chosen to be $s = r/1 = r \le 2$. The von Bahr-Esseen inequality applies perfectly and provides:

$$\mathbb{E}\left[\left||\sum_{i=1}^{m}(Tv)_i| - \mathbb{E}[|\sum_{i=1}^{m}(Tv)_i|]\right|^r\right] \le 2\sum_{i=1}^{m}\mathbb{E}\left[\left||(Tv)_i| - \frac{1}{m}\|v\|_p\right|^r\right]$$

Step 3. Bounding the Central Moments. Proposition 26 of [LLW23] gives

$$\mathbb{E}||X| - \mathbb{E}|X||^r \le 2^r \mathbb{E}|X|^r.$$

Applying this with $X = (Tv)_i$, the RHS is $2^r\mathbb{E}[|(Tv)_i|^r] = \mathcal{O}(m)^{-r}\|v\|_p^r)$. Summing over the m terms gives:

$$\mathbb{E}[|\|Tv\|_1 - \mathbb{E}[\|Tv\|_1]|^r] \le \mathcal{O}(m)^{1-r}\|v\|_p^r.$$

Then using standard moment inequalities, we can bound the first moment by higher moments: for any $r > 1$,

$$\mathbb{E}[|S - \mathbb{E}[S]|] \le (\mathbb{E}[|S - \mathbb{E}[S]|^r])^{1/r}.$$

Plugging in $S = \|Tv\|_1$ and $m \ge \mathrm{poly}(\varepsilon^{-1})$, we see that the error $\mathbb{E}[|\|Tv\|_1 - \|v\|_p|]$ is bounded by $\mathcal{O}(\varepsilon\|v\|_p)$ as desired. This concludes the proof to property (1).

Moving to property (2), we note that this part of the original proof to Lemma 18 is independent of the choice of q. Therefore, the only difference is that: property (2) of Lemma 18 in [LLW23] argues for an entire subspace by union bounding over a γ-net, while in our case this is simplified to a union bound over a finite set V. This is supported as long as m is chosen to be appropriate $\mathrm{poly}(\log n, \varepsilon^{-1}, \delta^{-1})$.

□

B Proof to Theorem 3.2

Theorem B.1 (Repeating Theorem 3.2)**.** *Let $1 < p < 2$ be constant, $A, B \subset \mathbb{R}^d$ of size n, and $\varepsilon \in (0,1/2)$. For some $m = \mathrm{poly}(\log n, \varepsilon^{-1})$, if $d \ge m^{2K}$ (where $K \approx 11.76$), then there is an algorithm that computes a $(1+\varepsilon)$-approximation to $CH_p(A,B)$ in total time:*

$$\mathcal{O}\left(nd \cdot \mathrm{poly}\log(m) + T_1(n,m,\varepsilon)\right),$$

with .9 probability, where $T_1(n,m,\varepsilon)$ is the time to $(1+\varepsilon)$-approximation the ℓ_1 Chamfer distance in $\mathbb{R}^m$ with high probability.

Proof. As outlined earlier, we sample a lopsided embedding matrix T from Lemma 3.1. The algorithm proceeds by (1) applying the embedding matrix T from ℓ_p^d to ℓ_1^m using FMM, and (2) running the ℓ_1 Chamfer algorithm on the embedded points.

Runtime: The runtime of step (2) is clearly $T_1(n, m, \varepsilon)$. We thus only need to argue that step (1) runs in time $\mathcal{O}(nd \cdot \operatorname{poly}\log(m))$, which follows immediately from Lemma 2.1.

Correctness: It suffices to show that $\mathrm{CH}_1(TA, TB) = (1 \pm \mathcal{O}(\varepsilon))\mathrm{CH}_p(A, B)$. We apply Lemma 3.1 on set $V := \{a - b \mid a \in A, b \in B\}$, which has size $\mathcal{O}(n^2)$. We show the lower and upper bound separately:

(Lower Bound). By the second guarantee of T in Lemma 3.1, we can argue that with .99 chance,

$$\begin{aligned}\mathrm{CH}_1(TA, TB) &= \sum_{a\in A} \min_{b\in B} \|T(a-b)\|_1 \\ &\geq (1-\varepsilon) \sum_{a\in A} \min_{b\in B} \|a-b\|_p \\ &= (1-\varepsilon)\mathrm{CH}_p(A, B)\end{aligned}$$

where the second step follows from the fact that for any minimizer b^* of $\min_{b\in B} \|T(a-b)\|_1$, it must hold that

$$\|T(a-b^*)\|_1 \geq (1-\varepsilon)\|a-b^*\|_p \geq (1-\varepsilon) \min_{b\in B} \|a-b\|_p.$$

(Upper Bound). We now show that the embedded distance does not blow up. We use b_a^* to denote $\arg\min_{b\in B} \|a-b\|_p$ (when there are multiple minimizers, we fix an arbitrary one of them). Using the first guarantee of Lemma 3.1 and the linearity of expectation:

$$\begin{aligned}&\mathbb{E}\Big[\Big|\sum_{a\in A} \|T(a-b_a^*)\|_1 - \mathrm{CH}_p(A, B)\Big|\Big] \\ =&\mathbb{E}\Big[\Big|\sum_{a\in A} \|T(a-b_a^*)\|_1 - \sum_{a\in A} \|a-b_a^*\|_p\Big|\Big] \\ \leq&\sum_{a\in A} \mathbb{E}\Big[\Big|\|T(a-b_a^*)\|_1 - \|a-b_a^*\|_p\Big|\Big] \\ \leq&\varepsilon \cdot \sum_{a\in A} \|a-b_a^*\|_p \\ =&\varepsilon \cdot \mathrm{CH}_p(A, B)\end{aligned}$$

This bounds the expectation of the approximation error. By Markov's inequality, the probability of this error term exceeding 100ε is only .01. Therefore, with probability at least .99, the complementary event holds, i.e.

$$|\sum_{a\in A} \|T(a-b_a^*)\|_1 - \mathrm{CH}_p(A, B)| \leq 100\varepsilon \cdot \mathrm{CH}_p(A, B) = \mathcal{O}(\varepsilon) \cdot \mathrm{CH}_p(A, B).$$

On the other hand, since

$$\begin{aligned}\mathrm{CH}_1(TA, TB) &= \sum_{a\in A} \min_{b\in B} \|T(a-b)\|_1 \\ &\leq \sum_{a\in A} \|T(a-b_a^*)\|_1,\end{aligned}$$

it follows that $\mathrm{CH}_1(TA, TB) \leq \mathrm{CH}_p(A, B) + \mathcal{O}(\varepsilon) \cdot \mathrm{CH}_p(A, B)$, which concludes the proof.

□

C Delayed Proofs from Section 4.1

In this section, we repeat the statements of the lemmas and give the proofs.

Lemma C.1 (Repeating Lemma 4.1)**.** *For all $a, b \in \mathbb{R}^m$ and $0 \le k \le t$, if $h_k(a) = h_k(b)$, then $\|a - b\|_2 \le 2^k\sqrt{m}$.*

Proof. If $h_k(a) = h_k(b)$, then a, b lie in the same m-dimensional cube of side-length 2^k. The ℓ_2 diameter of such a cube is $2^k\sqrt{m}$. □

Lemma C.2 (Repeating Lemma 4.2)**.** *With probability $1 - \mathcal{O}(1/n)$, the following holds simultaneously for all a, b, k: If $h_k(a) = h_k(b)$, then $\|a - b\|_2 \le 2^k \cdot 3\log n$.*

Proof. Lemma 3.4 of [FI25] shows that $\forall a, b, k : \|a-b\|_1 \le 2^k \cdot 3\log n$ with probability $1-\mathcal{O}(1/n)$. Combining this with the fact that $\|a - b\|_1 \ge \|a - b\|_2$ concludes the proof. □

Lemma C.3 (Repeating Lemma 4.3)**.** *With probability $1 - \mathcal{O}(1/n)$, it holds for all $a \in A$ that $\mathbb{E}\mathcal{D}_a \le F \cdot \mathsf{opt}(a)$, where $F = \mathcal{O}(\min(m, \sqrt{m}\log n))$.*

Proof. We need the following fact:

Fact C.4 ([BIJ+23])**.** $\Pr[h_k(a) \ne h_k(b)] \le \frac{\|a-b\|_1}{2^k}$.

Similar to Theorem 3.5 in [FI25], we condition on Lemma 4.2 and fix $a \in A$. We also fix $b^* := \arg\min_{b\in B} \|a - b\|_2$ and define $k^* := \lceil \log(\|a - b^*\|_1)\rceil$.

Let $\mathcal{E}_k$ be the event that $\tilde{k} = k$ (recall that we identified a unique $\tilde{k}$ when defining the estimator). If $\mathcal{E}_k$ occurs, the estimator $\mathcal{D}_a$ is bounded by $2^k \cdot \tilde{F}$ for $\tilde{F} = \mathcal{O}(\min(\sqrt{m}, \log n))$. Thus

$$\mathbb{E}\mathcal{D}_a \le \sum_{0\le k\le t} \Pr[\mathcal{E}_k] \cdot 2^k \cdot \tilde{F}.$$

We split the sum based on $k \le k^*$ and $k > k^*$. The the first part by

$$\tilde{F}2^{k^*} \sum_{k\le k^*} \Pr[\mathcal{E}_k] \le \tilde{F}\|a - b^*\|_1.$$

For the second part,

$$\begin{aligned}\sum_{k>k^*} \Pr[\mathcal{E}_k] \cdot 2^k \cdot \tilde{F} &\le \tilde{F} \sum_{k>k^*} (\frac{\|a - b^*\|_1}{2^{k-1}})^2 \cdot 2^k \\ &\le 4\tilde{F} \cdot \|a - b^*\|_1 \sum_{k>k^*} \frac{\|a - b^*\|_1}{2^k} \\ &= \mathcal{O}(\tilde{F}\|a - b^*\|_1).\end{aligned}$$

Thus, $\mathbb{E}\mathcal{D}_a = \mathcal{O}(\tilde{F}) \cdot \|a - b^*\|_1 \le \mathcal{O}(\tilde{F}\sqrt{m}) \cdot \mathsf{opt}_a$. The approximation factor is $F = \mathcal{O}(\min(m, \sqrt{m}\log n))$.

□

D Delayed Proofs from Section 5

In this section, we repeat the statements of the theorems and give the proofs.

Theorem D.1 (Repeating Theorem 5.2)**.** *For normalized sets $A, B \subset \mathbb{S}^{d-1}$, the Fast ℓ_2 Chamfer distance algorithm (Theorem 4.4) can be modified to output an additive $\mathcal{O}(\varepsilon n)$-approximation to MaxSim$(A, B)$, and for $d \ge \mathrm{poly}\log(n)$, it runs in $\mathcal{O}\left(nd(\log\log n + \log(1/\varepsilon))/\varepsilon^2\right)$ time.*

Proof. For any $a, b \in \mathbb{S}^{d-1}$, the geometric identity $\|a-b\|_2^2 = \|a\|_2^2 + \|b\|_2^2 - 2\langle a, b\rangle = 2 - 2\langle a, b\rangle$ holds. Maximizing the inner product is strictly equivalent to minimizing the squared Euclidean distance. Therefore, we can output an approximation $|A| - \frac{1}{2}\tilde{\mathsf{CH}}_2(A, B)$, where $\tilde{\mathsf{CH}}_2(A, B)$ is an $(1+\varepsilon)$-approximation to $\sum_{a\in A} \min_{b\in B} \|a-b\|_2^2$. Note that this quantity corresponds to the Chamfer distance with the underlying metric being the squared ℓ_2 norms (as opposed to ℓ_2).

To approximate this, we revisit Lemma 4.3, the analysis of the approximation factor F for the Quadtree estimator $\mathcal{D}_a$. We observe that using the same analysis, by shifting from the ℓ_2 norm to the squared ℓ_2 norm, F can be bounded by $\mathcal{O}(\min(m^2, m\log^2 n))$, where $m = \mathcal{O}(\log^2 n)$ as we have in the ℓ_2 case. The rest of the algorithm from Theorem 4.4, including the FJLT remains unchanged. Therefore, the total runtime is $\mathcal{O}(nd \log\log n)$ from FJLT, plus $\mathcal{O}(n(d + F/\varepsilon^2)\log(F/\varepsilon^2))$ as shown in Theorem 2.3. For $d \geq \operatorname{poly}\log(n)$, this is at most $\mathcal{O}\left(nd(\log\log n + \log(1/\varepsilon))/\varepsilon^2\right)$ as claimed. The relative error bound from Theorem 4.4 translates to an additive $\mathcal{O}(\varepsilon n)$ error bound in the cosine similarity space, following the geometric identity and the unit sphere assumption.

□

Theorem D.2 (Reapeating Theorem 5.3)**.** *Any randomized constant-pass streaming algorithm that computes a constant-factor approximation to the ℓ_p Chamfer distance ($p \geq 1$) for dynamically updated point sets requires $\Omega(n)$ bits of space.*

Proof. We reduce from the 2-Player Set Disjointness (DISJ) problem, which is a core hardness result in communication complexity. Alice and Bob receive indicator vectors $x, y \in \{0,1\}^n$ respectively. We assume the standard promise that the Hamming weights are $|x|_1 = |y|_1 = n/4$. DISJ asks whether $x \cap y = \emptyset$ (i.e., whether there exists some index i such that $x_i = y_i = 1$). It is a well-known result that solving DISJ requires $\Omega(n)$ communication.

We map this to a dynamic point stream in $\mathbb{R}^2$. Let the target set be a fixed multiset B containing $n/4$ copies of $(1, 0)$, $n/4$ copies of $(0, 1)$, and $n/2$ copies of $(0, 0)$. Note that $|B| = n$. Let the source set $A = \{a_1, \ldots a_n\}$ be initialized with n points at the origin $(0, 0)$.

In the stream, Alice receives x and applies additive updates to the first coordinate of the i-th point of A, setting $a_i = (x_i, 0)$. She then transmits the memory state of the streaming data structure to Bob. Bob receives y and applies additive updates to the second coordinate, resulting in the final point set $A = \{(x_i, y_i) \mid i \in [n]\}$.

We analyze the ℓ_p Chamfer distance $\mathrm{CH}_p(A, B)$:

- **If the sets are Disjoint ($x \cap y = \emptyset$):** The condition $x_i \cdot y_i = 0$ holds for all i. Due to the fixed Hamming weights, the resulting set A contains exactly $n/4$ points at $(1, 0)$, $n/4$ points at $(0, 1)$, and $n/2$ points at $(0, 0)$. Thus, A and B are identical multisets, and the distance $\mathrm{CH}_p(A, B) = 0$.
- **If the sets Intersect ($x \cap y \neq \emptyset$):** There exists at least one index i where $(x_i, y_i) = (1, 1)$. The ℓ_p distance from the point $(1, 1)$ to any point in the reference set B is $\min(\|(1, 1) - (0, 0)\|_p, \|(1, 1) - (1, 0)\|_p, \|(1, 1) - (0, 1)\|_p) = \min(2^{1/p}, 1, 1) = 1$. Thus, $\mathrm{CH}_p(A, B) \geq 1$.

Since distinguishing between a distance of 0 and a distance ≥ 1 allows one to decide the Set Disjointness problem, the space required to store the streaming data structure (transmitted from Alice to Bob) must be at least the communication complexity of DISJ. Thus, the space required is $\Omega(n)$ bits.

□

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: See our contribution section.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See the conclusion section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: All theorems and lemmas are accompanied with proofs (some are delayed to the appendix). Assumptions, if any, are all stated in the statements of theorems and lemmas.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Our experiment section describes all necessary information for reproducing the results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The main contribution of this paper is theory-focused. We include experiments as proof-of-concepts, but the code are not yet organized with sufficient instructions. We plan to organize and share it after the paper is made public.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Our experiment section describes all necessary information for reproducing the results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: See table 3.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Our experiment section describes all necessary information for reproducing the results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: N/A

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See the impact statement section.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: N/A

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: N/A

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: N/A

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: N/A

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: N/A

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: This is declared in the submission registration.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.