עבודת גמר (תזה) לתואר
**מוסמך למדעים**

מוגשת למועצה המדעית של
מכון ויצמן למדע
רחובות, ישראל

Thesis for the degree
**Master of Science**

Submitted to the Scientific Council of the
Weizmann Institute of Science
Rehovot, Israel

מאת
ניר פטרושקה

By
**Nir Petruschka**

# הכוח של שיכונים רקורסיביים עבור מטריקות $\ell_p$
# The Power of Recursive Embeddings for $\ell_p$ Metrics

מנחה:
פרופ' רוברט קראוטגמר

Advisor:
Prof. Robert Krauthgamer

תשרי התשפ"ו

October 2025

**Abstract**

Metric embedding is a powerful tool used extensively in mathematics and computer science. We devise a new method of using metric embeddings recursively, which turns out to be particularly effective in $\ell_p$ spaces, $p > 2$, yielding state-of-the-art results for Lipschitz decomposition, for Nearest Neighbor Search, and for embedding into $\ell_2$. In a nutshell, our method composes metric embeddings by viewing them as reductions between problems, and thereby obtains a new reduction that is substantially more effective than the known reduction that employs a single embedding. We in fact apply this method recursively, oftentimes using double recursion, which further amplifies the gap from a single embedding

## Acknowledgements

I want to thank my advisor, Prof. Robert Krauthgamer, for his exceptional guidance and support throughout this journey. I especially appreciate the way he always listened attentively to my ideas, many of which were admittedly quite bad, and, without ever making me feel discouraged, helped guide the path through which some good ideas eventually emerged.

I am deeply grateful to my family, who have always been there for me, supporting me through every step along the way. I am especially thankful to my partner, Daniel, whose unwavering belief in me gave me the confidence to keep going, even in the most challenging moments. Finally, a special thank you goes to my brother Asaf, who, whenever he's approached for assistance, starts with "I'm not sure I can help you with this", followed by the best advice one could receive.

# 1 Introduction[1]

Metric embeddings represent points in one metric space using another metric space, often one that is simpler or easier, while preserving pairwise distances within some distortion bounds. This mathematical tool is very powerful at transferring properties between the two metric spaces, and is thus used extensively in many areas of mathematics and computer science. Its huge impact over the past decades is easily demonstrated by fundamental results, such as John's ellipsoid theorem [Joh48], the Johnson-Lindenstrauss (JL) Lemma [JL84], Bourgain's embedding [Bou85], and probabilistic tree embedding [Bar96].

We devise a new method of using metric embeddings *recursively*, in a manner that is particularly effective for $\ell_p$ spaces, $p > 2$. Our method is based on the well-known approach of embedding $\ell_p$ into $\ell_2$ (via the so-called Mazur map), but leverages a new form of recursion that goes through intermediate spaces, to beat a direct embedding from $\ell_p$ into $\ell_2$.

Our method is inspired by the concept of reduction between (computational) problems, which is fundamental in computer science and has been used extensively to design algorithms and/or to prove conditional hardness. Many known reductions use metric embeddings in a straightforward manner, without harnessing the full power of reductions, which allow further manipulation, like employing multiple embeddings and taking the majority (or best) solution.[2] To see this gap between embeddings and reductions, consider a composition of multiple embeddings, which yields overall an embedding from the first metric space to the last one. While going through intermediate metric spaces may simplify the exposition, it can only restrict the overall embedding. In contrast, composing metric embeddings by way of reductions, can create new reductions that are substantially richer than any single direct embedding. Our method actually composes reductions *recursively*, which makes this gap even more pronounced. We emphasize that the application of this method is problem-specific, unlike a metric embedding which is very general and thus applies to many problems at once. On the flip side, tailoring our recursive method to a specific problem opens the door to embeddings that are non-oblivious to the problem/data, which is reminiscent of data-dependent space partitioning used in recent nearest neighbor search (NNS) algorithms [ANN+18a, ANN+18b, KNT21]. To the best of our knowledge, this recursive method is new, i.e., related to but different from variants that have been used in prior work.

Our method yields several state-of-the-art results: (i) Lipschitz decomposition for finite subsets of $\ell_p$ spaces, $p > 2$; (ii) consequently, also Lipschitz decomposition for $\ell_\infty^d$; and (iii) algorithms for NNS in $\ell_p$ spaces, $p > 2$. After obtaining these results, we noticed the online posting of parallel work [NR25], and realized that our method can also (iv) improve some of its results about embedding into $\ell_2$.

## 1.1 Lipschitz Decomposition

A standard approach in many metric embeddings and algorithms is to partition a metric space into low-diameter (so-called) clusters, and the following probabilistic variant is commonly used and highly studied (sometimes called a separating decomposition).

[1] The main body of this thesis is identical to a paper that was accepted for publication at FOCS 2025 [KPS25] and is a joint work with Shay Sapir. These results improve upon earlier work that was also carried out during these MSc studies and was published at SoCG 2025 [KP25]. Appendix A contains yet unpublished results that resolve one of the questions posed in this thesis.

[2] This is perhaps analogous to the difference between Cook reductions and Karp reductions. The former allows the use of a subroutine that solves the said problem, while the latter applies only a single transformation on the input, and is thereby restricted to a single subroutine call.

**Definition 1.1** (Lipschitz decomposition [Bar96]). Let $(\mathcal{M}, d_{\mathcal{M}})$ be a metric space. A distribution $\mathcal{D}$ over partitions of $\mathcal{M}$ is called a $(\beta, \Delta)$-*Lipschitz decomposition* if

- for every partition $P \in \text{supp}(\mathcal{D})$, all clusters $C \in P$ satisfy $\text{diam}(C) \le \Delta$; and
- for every $x, y \in \mathcal{M}$,
$$\Pr_{P\sim\mathcal{D}}[P(x) \neq P(y)] \le \beta \frac{d_{\mathcal{M}}(x,y)}{\Delta},$$
where $P(z)$ denotes the cluster of $P$ containing $z \in \mathcal{M}$ and $\text{diam}(C) \coloneqq \sup_{x,y\in C} d_{\mathcal{M}}(x, y)$.

Our first use of recursive embedding yields the following theorem, whose proof appears in Section 3.

**Theorem 1.2.** *Let $p \ge 2$ and $d \ge 1$. Then for every $n$-point metric $\mathcal{C} \subset \ell_p^d$ and $\Delta > 0$, there exists an $(O(p^4\sqrt{\min\{\log n, d\}}), \Delta)$-Lipschitz decomposition.*

Typically, $\Delta$ is not known in advance or one needs multiple values of $\Delta$ (e.g., every power of 2). We naturally seek the smallest possible $\beta$ in this setting, and thus define the (optimal) *decomposition parameter* of a metric space $(\mathcal{M}, \rho)$ as

$$\beta^*(\mathcal{M}) \coloneqq \inf_{\beta\ge1} \Big\{\beta : \ \forall\Delta > 0, \text{ every finite } \mathcal{M}' \subseteq \mathcal{M} \text{ admits a } (\beta, \Delta)\text{-Lipschitz decomposition}\Big\},$$

and further define $\beta_n^*(\mathcal{M}) \coloneqq \sup\big\{\beta^*(\mathcal{M}') : \ \mathcal{M}' \subseteq \mathcal{M}, |\mathcal{M}'| \le n\big\}$. The following two corollaries of Theorem 1.2 bound these quantities and delineate the asymptotic dependence on $n$ and on $d$.

**Corollary 1.3.** *For every $p \in [2,\infty)$ and $n \ge 1$, we have $\beta_n^*(\ell_p) = O(p^4\sqrt{\log n})$.*

*Proof.* It follows directly from Theorem 1.2 and the result from [Bal90], that every finite set $X \subset \ell_p$ embeds isometrically into $\ell_p^d$ for some $d$. □

This result significantly improves the previous bound $\beta_n^*(\ell_p) = O(\log^{1-1/p} n)$ from [KP25], and fully resolves [Nao17, Question 1] (see also [Nao24, Question 83]), which asked for an $O_p(\sqrt{\log n})$ bound. (Throughout, the notation $O_\alpha(\cdot)$ hides a factor that depends only on $\alpha$.) In parallel to our work, a slightly weaker bound $\beta_n^*(\ell_p) \le O(2^p\sqrt{\log n})$ was obtained in [NR25]. Both our improvement and that of [NR25] rely on the technique developed in [KP25], and essentially apply it iteratively/recursively instead of once, and ours actually applies double recursion.

**Corollary 1.4.** *For every $p \in [2,\infty]$ and $d \ge 1$, we have $\beta^*(\ell_p^d) = O((\min\{p, \log d\})^4 \cdot \sqrt{d})$.*

*Proof.* For $p \le \log d$, it follows from Theorem 1.2. For larger $p$, use Hölder's inequality to reduce the problem from $\ell_p^d$ to $\ell_{\log d}^d$ with $O(1)$ distortion.[3] □

Theorem 1.4 is slightly weaker than Naor's main result in [Nao17], which was later slightly improved in [Nao24]. Naor showed that $\beta^*(\ell_p^d) = \Theta(\sqrt{\cdot d})$ for all $p \in [2,\infty]$, matching the lower bound that follows from [CCG+98]. Our proof is fundamentally different from, and arguably simpler than, Naor's proof, which relies on a deep understanding of the geometry of $\ell_p^d$ spaces. One may hope that our proof could be enhanced to match the exact asymptotics of $\beta^*(\ell_\infty^d)$, perhaps by simply optimizing the constants in our recursion that yield the $p^4$ factor in Theorem 1.2. Unfortunately, this approach has a serious barrier. For $\ell_{\log n}$, we have $\beta_n^*(\ell_{\log n}) = \Omega(\log n)$, since every $n$-point metric embeds into $\ell_{\log n}$ with $O(1)$ distortion by [Mat97], and there is an $\Omega(\log n)$ lower bound for Lipschitz decomposition of general $n$-point metrics [Bar96]. Improving the $p^4$ factor in our analysis to $o(\sqrt{p})$ would imply that $\beta_n^*(\ell_{\log n}) = o(\log n)$, contradicting the known lower bound.

[3] A metric space $(\mathcal{M}, d_{\mathcal{M}})$ embeds into a metric space $(\mathcal{N}, d_{\mathcal{N}})$ with distortion $D \ge 1$ iff there exists $s > 0$ and a function $f : \mathcal{M} \to \mathcal{N}$ such that for all $x, y \in \mathcal{M}$, $\frac{s}{D} \cdot d_{\mathcal{M}}(x, y) \le d_{\mathcal{N}}(f(x), f(y)) \le s \cdot d_{\mathcal{M}}(x, y)$.

*Remark* 1.5. Naor [Nao17] shows that his upper bound on $\beta^*(\ell_\infty^d)$ has an important application to the Lipschitz extension problem. More precisely, he proves an infinitary variant of his upper bound, and that it implies a similar bound on $e(\ell_\infty^d)$, which is the Lipschitz extension modulus of $\ell_\infty^d$. He thus concludes that $e(\ell_\infty^d) \leq O(\sqrt{d \log d})$, which almost matches (up to lower order factors), the lower bound $e(\ell_\infty^d) \geq \Omega(\sqrt{d})$ that follows from [BB05, BB06]. We have not attempted to extend Theorem 1.4 to the infinitary variant, as Naor notes that it is required only for extension theorems into certain exotic Banach spaces [Nao17, Appendix A, Remark 4].

*Remark* 1.6. The result of Theorem 1.2 extends to a related notion of decomposition, that was introduced in [FN22] and immediately implies geometric spanners. This yields spanners for $\ell_p$ spaces, $p > 2$, whose stretch-size tradeoff is comparable to that known for $\ell_2$. Previously, weaker bounds for such decompositions, and consequently also weaker spanners for $\ell_p$, were proved in [KP25]. The details, which are similar to Theorem 1.2, are omitted.

## 1.2 Nearest Neighbor Search

The Nearest Neighbor Search (NNS) problem is to design a data structure that preprocesses an $n$-point dataset $V$ residing in a metric $\mathcal{M}$, so that given a query point $q \in \mathcal{M}$, the data structure reports a point in $V$ that is closest to $q$ (and approximately closest to $q$ in approximate NNS). The main measures for efficiency are the data structure's space complexity and the time it takes to answer a query; a secondary measure is the preprocessing time, which is often proportional to the space. The problem has a wide range of applications in machine learning, computer vision and other fields, and has thus been studied extensively, including from theoretical perspective, see e.g. the survey [AI17]. It is well known that approximate NNS reduces to solving $\text{polylog}(n)$ instances of the approximate *near* neighbor problem [IM98], hence we consider the latter.

**Definition 1.7** (Approximate Near Neighbor)**.** The Approximate Near Neighbor problem for a metric space $(\mathcal{M}, d_\mathcal{M})$ and parameters $c \geq 1$, $r > 0$, abbreviated $(c, r)$-*ANN*, is the following. Design a data structure that preprocesses an $n$-point subset $V \subseteq \mathcal{M}$, so that given a query $q \in \mathcal{M}$ with $d_\mathcal{M}(q, V) \leq r$,[4] it reports $x \in V$ such that

$$d_\mathcal{M}(q, x) \leq cr.$$

In a randomized data structure, the reported $x$ satisfies this with probability at least $2/3$.

We prove the following theorem, whose proof appears in Section 4 and is similar in spirit to that of Theorem 1.2. It applies our method of recursive embedding, using Mazur maps for $n$-point subsets of $\ell_p^d$.

**Theorem 1.8.** *Let $p > 2$, $d \geq 1$ and $0 < \varepsilon < 1$. Then for $c = O(p^{1+\ln 4+\varepsilon})$ and every $r > 0$, there is a randomized data structure for $(c, r)$-ANN in $\ell_p^d$, that has query time* $\text{poly}(\varepsilon^{-1} d \log n)$*, and has space and preprocessing time* $\text{poly}(d n^{\varepsilon^{-1} \log p})$.

*Remark.* Picking $\varepsilon = \frac{1}{\log p}$ is sufficient to get approximation $O(p^{1+\ln 4}) \leq O(p^{2.387})$.

Most prior work on ANN in $\ell_p$ spaces studies the case $1 \leq p \leq 2$, where $(O(1), r)$-ANN can be solved using query time $\text{poly}(d \log n)$ and space $\text{poly}(n)$ [KOR00, IM98, HIM12]. For $p > 2$, such a bound is not known, and we list in Table 1 all the known results (ours and previous ones), which are often incomparable. The results of [And09, AIK09] and of [ANRW21] are based on Indyk's [Ind01] result for $\ell_\infty$, and are most suitable for large values of $p$; note though that the

[4] If $d_\mathcal{M}(q, V) > r$, it may report anything, where as usual, $d_\mathcal{M}(q, V) \coloneqq \min_{x^* \in V} d_\mathcal{M}(x^*, q)$.

| Approximation | Query time | Space | Reference |
|---|---|---|---|
| $O(\varepsilon^{-1} \log\log d)$ | $n^{\varepsilon^p}$ | $n^{1+\varepsilon}$ | [AIK09, And09] |
| $O_\varepsilon(\log p \cdot (\log d)^{2/p})$ | $n^\varepsilon$ | $n^{1+\varepsilon}$ | [ANRW21] |
| $2^{O(p)}$ | $(d \log n)^{O(1)}$ | $n^{O(1)}$ | [BG19] |
| $p^{O(1)}$ | $(d \log n)^{O(1)}$ | $n^{O(\log p)}$ | Thm 1.8 |
| $O(p/\varepsilon)$ | $n^\varepsilon$ | $n^{1+\varepsilon}$ | [ANN+18a, ANN+18b, KNT21] |
| $c$ | $n^\varepsilon$ | $n^{O(p/c)\cdot\log(1/\varepsilon)}$ | [BBM+24] |

Table 1: Known data structures for ANN in $\ell_p$, $p > 2$. For brevity, we omit here $\mathrm{poly}(d \log n)$ factors when the complexity is polynomial in $n$. The top-listed two results are particularly suited for large values of $p$, and the others are suited for small values of $p$.

*preprocessing time* of [ANRW21] is exponential in $d$. The other results are more suited for small values of $p > 2$, and they all have different downsides: one result [BG19] has a large approximation $2^{O(p)}$; another one [ANN+18a, ANN+18b, KNT21] has a large query time $n^\varepsilon \cdot \mathrm{poly}(d \log n)$, which can be mitigated by picking $\varepsilon = \frac{1}{\log n}$, at the cost of increasing the approximation to $O(p \log n)$; ours (Theorem 1.8) has a large space $n^{O(\log p)}$; and lastly, [BBM+24] and [AIK09, And09] can achieve $O(1)$-approximation but this requires an even larger space $d \cdot n^{2^{O(p)} \log(1/\varepsilon)}$ and $n^{O(\log d)}$, respectively. The bottom line is that the regime of $p > 2$ is notoriously difficult. It remains open to bridge the gap between small $p$ and large $p$, and specifically to obtain $O(p)$-approximation using $\mathrm{poly}(d \log n)$ query time and $\mathrm{poly}(n)$ space.

Our result for ANN provides yet another illustration for the power of recursive embedding. Bartal and Gottlieb [BG19] mentioned that Assaf Naor noted, in personal communication regarding improving their $2^{O(p)}$-approximation, that all uniform embeddings of $\ell_p$ to $\ell_2$ (like Mazur maps) have distortion exponential in $p$ [Nao14, Lemma 5.2]. Our use of recursive embeddings breaks this barrier, and essentially provides a black-box reduction from $\ell_p$ to $\ell_2$, that still uses Mazur maps but achieves $\mathrm{poly}(p)$-approximation. We note that the improved approximation of [ANN+18a, ANN+18b, KNT21] uses embedding into $\ell_2$ with small average distortion, however this approach is not known to provide a black-box reduction for ANN, and its specialized solution increases the query time.

### 1.3 Low-Distortion Embeddings

After we obtained our aforementioned results for Lipschitz decomposition and NNS, we noticed the online posting of [NR25] on the distortion required for embedding $\ell_p$ space ($p > 2$) into Euclidean space, and used our technique to extend their result. The study of the distortion required for embedding metrics into Euclidean space has a decades-long history for general metrics [Joh48, Bou85, LLR95] and for $\ell_p$ space [Lee05, CGR05, ALN08, CNR24, BG14, NR25]. For an infinite metric space $(\mathcal{M}, d_\mathcal{M})$, define $c_2^n(\mathcal{M}) := \sup_{\mathcal{C}\subseteq\mathcal{M},\, |\mathcal{C}|\le n} c_2(\mathcal{C})$, where $c_2(\mathcal{C})$ denotes the minimal distortion needed to embed $\mathcal{C}$ into $\ell_2$. We prove the following in Section 5.

**Theorem 1.9.** *If $3 < p < 3\sqrt{e}$, then for every fixed $0 < \varepsilon \le 1$,*

$$c_2^n(\ell_p) \le O(\log^{\frac{1}{2}+\ln\frac{p}{3}+\varepsilon} n).$$

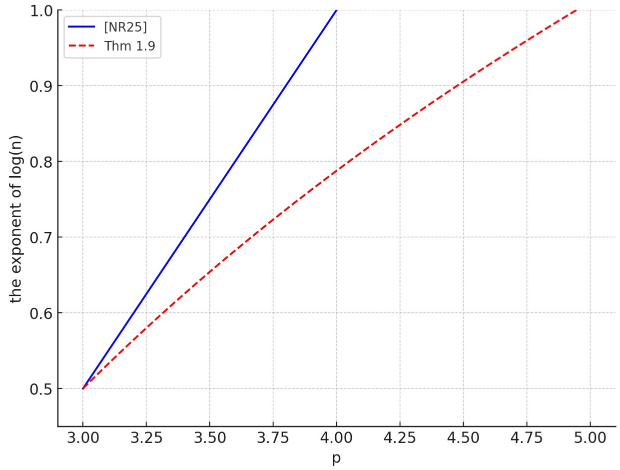


Figure 1: The distortion of embedding from $\ell_p$, $p > 3$ into $\ell_2$ shown by depicting the exponent of $\log n$ in [NR25, Theorem 1] (blue) compared with our bound in Theorem 1.9 (red).

Previously, for $p > 2$, non-trivial distortion was only known in the range $2 < p < 4$ [BG14, NR25], where non-trivial means distortion asymptotically smaller than $O(\log n)$, which holds for every $n$-point metric space [Bou85]. Bartal and Gottlieb [BG14] established that $c_2^n(\ell_p) = O(\log^{p/4} n)$ for every $p \in (2,4)$, and Naor and Ren [NR25] proved a better bound $c_2^n(\ell_p) = O(\sqrt{\log n} \cdot \log\log n)$ for $p \in (2,3]$ and $c_2^n(\ell_p) = O(\log^{p/2-1} n \cdot \log\log n)$ for $p \in (3,4)$. Theorem 1.9 improves these bounds further in the range $3 < p < 3\sqrt{e}$. Since it may not be immediate that Theorem 1.9 indeed improves the bounds on $c_2^n(\ell_p)$ for all $3 < p < 3\sqrt{e}$, we plot the corresponding exponents of the $\log n$ factor in Figure 1.

*Remark* 1.10. Every finite metric embeds isometrically in $\ell_\infty$, and thus $c_2^n(\ell_\infty) = \Theta(\log n)$ by [Bou85] and [LLR95]. For $\ell_p$, $p \in (2,\infty)$, a lower bound of

$$c_2^n(\ell_p) \geq \Omega(\log^{1/2-1/p} n)$$

follows from [LN13, Theorem 1.3].

## 2 Preliminaries

The main tool we use for recursive embeddings between $\ell_p$ spaces is a classical embedding, commonly known as the Mazur map. For every $p, q \in [1,\infty)$, the Mazur map $M_{p,q} : \ell_p^m \to \ell_q^m$ is computed by raising the absolute value of each coordinate to the power $p/q$ while preserving the original signs. The following key property of this map is central to all our results.

**Theorem 2.1** ([BL98, BG19])**.** *Let $1 \leq q < p < \infty$ and $C_0 > 0$, and let $M$ be the Mazur map $M_{p,q}$ scaled down by factor $\frac{p}{q} {C_0}^{p/q-1}$. Then for all $x, y \in \ell_p$ such that $||x||_p, ||y||_p \leq C_0$,*

$$\tfrac{q}{p}(2C_0)^{1-p/q}||x-y||_p^{p/q} \leq ||M(x) - M(y)||_q \leq ||x-y||_p.$$

## 3 Lipschitz Decomposition of $\ell_p$ Metrics

In this section, we prove Theorem 1.2. We first outline the proof. Our approach uses a double recursion, where each recursion is an instance of recursive embedding. The first recursion takes a

Lipschitz decomposition of a finite subset $\mathcal{M} \subset \ell_p^d$ with decomposition parameter $\beta$ and produces a Lipschitz decomposition with (ideally smaller) decomposition parameter $\beta_{new}$. Each iteration in this recursion is as follows. We first use the given decomposition to decompose $\mathcal{M}$ into bounded-diameter subsets, embed each subset into $\ell_q$ for $q < p$ using Mazur maps, employ Lipschitz decomposition for $\ell_q$, and pull back the solution (clusters) we found. It is natural to choose here $q = 2$, because the known Lipschitz decompositions for $\ell_2$ are tight. However, this choice leads to a decomposition parameter with an $\exp(p)$ factor, and we overcome this by picking $q = p/2$. We only then apply a second recursion, which goes from $\ell_p$ to $\ell_2$ gradually, via intermediate values $2 < q < p$.

**Lemma 3.1.** *Let $2 \leq q < p < \infty$ and let $\mathcal{M} \subset \ell_p$ be an $n$-point metric. Suppose that for every $\Delta' > 0$, there exists a $(\beta, \Delta')$-Lipschitz decomposition of $\mathcal{M}$. Then, for every $\Delta > 0$, there exists a $(\beta_{new}, \Delta)$-Lipschitz decomposition of $\mathcal{M}$, with*

$$\beta_{new} = 4(\tfrac{p}{2q})^{q/p} \ [\beta_n^*(\ell_q)]^{q/p} \ \beta^{1-q/p}.$$

Theorem 3.1 provides the recursion step for the first recursion from the outline above, and we use it with $q = p/2$. For the natural choice of $q = 2$, the expression in Theorem 3.1 equals $\beta_{new} = 4(p/4)^{2/p} \ [\beta_n^*(\ell_2)]^{2/p} \ \beta^{1-2/p}$, hence iterative applications converge to the fixpoint $\beta = \frac{p}{4}2^p \cdot \beta_n^*(\ell_2)$, which is easily found by setting $\beta = \beta_{new}$. In contrast, for $q = p/2$, the expression simplifies to $\beta_{new} = 4\sqrt{\beta_n^*(\ell_{p/2}) \cdot \beta}$, the fixpoint is now $\beta = 16\beta_n^*(\ell_{p/2})$, and recursion on $p$ introduces only a $\mathrm{poly}(p)$ factor.

*Proof.* Let $\Delta > 0, p \in (2, \infty)$, and let $\mathcal{M} \subset \ell_p$ be an $n$-point metric space. Set $a := \frac{1}{2}\big(\frac{2q\beta}{p\beta_n^*(\ell_q)}\big)^{q/p}$ and $b := \frac{\beta_n^*(\ell_q)a}{\beta}$, chosen to satisfy

$$\tfrac{\beta}{a} = \tfrac{\beta_n^*(\ell_q)}{b} \qquad \text{and} \qquad \tfrac{p}{q}(2a)^{p/q-1}b = 1. \tag{1}$$

Construct a partition of $\mathcal{M}$ in the following steps:

1. Draw a partition $P_{\text{init}} = \{K_1, \ldots, K_t\}$ from a $(\beta, a\Delta)$-Lipschitz decomposition of $\mathcal{M}$.
2. Embed each cluster $K_i \subset \ell_p$ into $\ell_q$ using the embedding $f^{K_i}$ provided by Theorem 2.1 for $C_0 := a\Delta$.
3. For each embedded cluster $f^{K_i}(K_i)$, draw a partition $P_i = \{K_i^1, \ldots, K_i^{k_i}\}$ from a $(\beta_n^*(\ell_q), b\Delta)$-Lipschitz decomposition of $f^{K_i}(K_i)$.
4. Obtain a final partition $P_{\text{out}}$ by taking the preimage of every cluster of every $P_i$.

It is easy to see that $P_{\text{out}}$ is indeed a partition of $\mathcal{M}$, consisting of $\sum_{i=1}^t k_i$ clusters. Next, consider $x, y \in \mathcal{M}$ and let us bound $\Pr[P_{\text{out}}(x) \neq P_{\text{out}}(y)]$. Observe that a pair of points can be separated only in steps 1 or 3. Therefore,

$$\begin{aligned}
&\Pr\Big[P_{\text{out}}(x) \neq P_{\text{out}}(y)\Big] \\
&\quad \leq \Pr\Big[P_{\text{init}}(x) \neq P_{\text{init}}(y)\Big] + \Pr\Big[P_i(f^{K_i}(x)) \neq P_i(f^{K_i}(y)) \mid P_{\text{init}}(x) = P_{\text{init}}(y) = K_i\Big] \\
&\quad \leq \beta \frac{\|x-y\|_p}{a\Delta} + \beta_n^*(\ell_q)\frac{\|f^{K_i}(x) - f^{K_i}(y)\|_q}{b\Delta} \\
&\quad \leq \Big(\tfrac{\beta}{a} + \tfrac{\beta_n^*(\ell_q)}{b}\Big)\frac{\|x-y\|_p}{\Delta},
\end{aligned}$$

where the last inequality is because by Theorem 2.1, each $f^{K_i}$ is a non-expanding map from $K_i \subset \ell_p$ to $\ell_q$. Using (1), we obtain $\beta_{new} = 2\frac{\beta}{a} = 4(\frac{p}{2q})^{q/p}[\beta_n^*(\ell_q)]^{q/p}\beta^{1-q/p}$.

It remains to show that the final clusters all have diameter at most $\Delta$. Let $x, y \in \mathcal{M}$ be in the same final cluster, i.e., $P_{\text{out}}(x) = P_{\text{out}}(y)$. Then $P_{\text{init}}(x) = P_{\text{init}}(y) = K_i$ and $P_i(f^{K_i}(x)) = P_i(f^{K_i}(y))$. Combining the distortion guarantees of $f^{K_i}$ from Theorem 2.1 with the diameter bound of $P_i$, we get

$$\frac{q}{p}\Big(2a\Delta\Big)^{1-p/q}\|x-y\|_p^{p/q} \le \|f^{K_i}(x) - f^{K_i}(y)\|_q \le b\Delta.$$

Rearranging this and using (1), we obtain $\|x-y\|_p^{p/q} \le \frac{p}{q}(2a)^{p/q-1}b\Delta^{p/q} = \Delta^{p/q}$, which completes the proof. □

We are now ready to prove the main theorem.

*Proof of Theorem 1.2.* Let $p \in (2,\infty)$, and let $\mathcal{M} \subset \ell_p$ be an $n$-point metric space. For ease of presentation, we assume for now that $p$ is a power of 2, and resolve this assumption at the end. Denote $\beta_0(\mathcal{M}) = O(\min\{d, \log n\})$, given by [Bar96] and [CCG+98]. We now iteratively apply Theorem 3.1 with $q = p/2$, and obtain after $k$ iterations,

$$\begin{aligned}
\beta_k(\mathcal{M}) &= 4\sqrt{\beta_n^*(\ell_{p/2}) \cdot \beta_{k-1}(\mathcal{M})} \\
&= 4\sqrt{\beta_n^*(\ell_{p/2}) \cdot 4\sqrt{\beta_n^*(\ell_{p/2}) \cdot \beta_{k-2}(\mathcal{M})}} \\
&= \cdots \\
&= 4^{(1+1/2+\ldots+1/2^{k-1})}\ [\beta_n^*(\ell_{p/2})]^{(1/2+1/4+\ldots+1/2^k)}\ \beta_0(\mathcal{M})^{1/2^k} \\
&\le 16\beta_n^*(\ell_{p/2}) \cdot \beta_0(\mathcal{M})^{1/2^k}. \qquad (2)
\end{aligned}$$

Picking $k := \lceil \log(\log p \cdot \log \beta_0(\mathcal{M}))\rceil = O(\log(\log p \cdot \log \min\{d, \log n\}))$ yields $\beta_0(\mathcal{M})^{1/2^k} \le 2^{1/\log p}$, and we obtain $\beta^*(\mathcal{M}) \le \beta_k(\mathcal{M}) \le 2^{4+1/\log p} \cdot \beta_n^*(\ell_{p/2})$. Now recursion on $p$ implies

$$\beta^*(\mathcal{M}) \le 2p^4 \cdot \beta_n^*(\ell_2).$$

Finally, by [CCG+98] and the JL Lemma [JL84] we know that $\beta_n^*(\ell_2^d) \le O(\min\{\sqrt{d}, \sqrt{\log n}\})$, which concludes the proof when $p$ is a power of 2.

Resolving the case when $p$ is not a power of 2 is straightforward. Let $q$ be the largest power of 2 that is smaller than $p$, hence $1/2 < q/p < 1$. It suffices to show that $\beta_n^*(\ell_p) = O(\beta_n^*(\ell_q))$, as then we can apply the previous argument since $q$ is a power of 2. Now apply Theorem 3.1 for $k$ iterations, analogously to (2). We may assume that $\beta_n^*(\ell_q) \le \beta_i(\mathcal{M})$ for all $i \le k$, as otherwise we can simply abort after the $i$-th iteration, hence $\beta_k(\mathcal{M}) = 4(\frac{p}{2q})^{q/p}\ [\beta_n^*(\ell_q)]^{q/p}\ \beta_{k-1}(\mathcal{M})^{1-q/p} \le 4\sqrt{\beta_n^*(\ell_q)\ \beta_{k-1}(\mathcal{M})}$. Now similarly to (2) we get $\beta_n^*(\ell_p) = O(\beta_n^*(\ell_q))$, and the theorem follows. □

*Remark* 3.2. We suspect that the factor 16 in the recursion (2) is an artifact of the analysis. First, by balancing the separation probabilities over all $k$ iterations, one can perhaps eliminate the factor 2 increase in the probabilities, and thus improve the factor in the recursion to roughly 4. Second, the Mazur maps require sets of bounded *radius*, while the construction guarantees sets of bounded *diameter*. Our proof uses the trivial bound radius $\le$ diam, which holds for every metric space, and subsets of $\ell_p$ may admit a tighter bound. Denote by $J_p \in [\frac{1}{2}, 1]$ the minimum number such that $\text{radius}(\mathcal{M}) \le J_p\,\text{diam}(\mathcal{M})$ for all $\mathcal{M} \subset \ell_p$. It is known that $J_\infty = 1/2$ and by Jung's Theorem, $J_2 = \frac{1}{\sqrt{2}}$. Then, the factor above improves to roughly $(2J_p)^2$. Keeping in mind the discussion

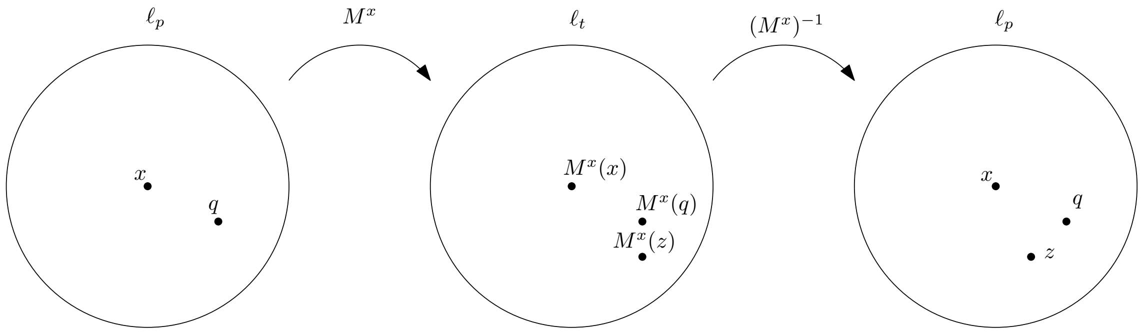


Figure 2: An illustration of Theorem 4.1. For the purpose of this illustration, the $\ell_p$ and $\ell_t$ balls are depicted using a Euclidean circle, and $x$ is assumed to lie at the origin of $\ell_p$. Given a query point $q$, an approximated solution $x$ is found in $\ell_p$ using $A_{base}$. The Mazur map $M^x$ is then applied, after which a solution $M^x(z)$ is found in $\ell_t$ using $A_x$. Finally, the inverse map is applied to obtain an improved solution $z$ in $\ell_p$.

following Theorem 1.4, and aiming for a clear presentation of the main ideas in the solution, we have omitted the above optimizations.

# 4 Nearest Neighbor Search

In this section, we design a data structure for approximate NNS in $\ell_p^d$ for $p > 2$, proving Theorem 1.8. Previously, Bartal and Gottlieb [BG19] devised a data structure that is based on embedding $\ell_p$ into $\ell_2$, for which good data structures are known (e.g., LSH), and they furthermore employ recursion to improve the approximation factor, from a large trivial factor down to $\exp(p)$. We observe that their embedding and recursion approach is actually analogous to Section 3, but using only the special case $q = 2$. We thus use our double recursion approach that goes through intermediate $\ell_q$ spaces, and obtain an improved approximation factor $\mathrm{poly}(p)$. In the rest of this section, we reserve the letter $q$ for the query point (which is standard in the NNS literature) and denote the intermediate spaces by $\ell_t$.

*Proof of Theorem 1.8.* First, we show an analogous claim to Theorem 3.1 but for the $(c, r)$-ANN problem. We take two NNS data structures, one for $\ell_p^d$ with approximation $c_p$ and one for $\ell_t^d$ (where $t < p$) with approximation $c_t$, and construct a new data structure for $\ell_p^d$ with approximation $c_{new}$ (ideally smaller than $c_p$).

Given an $n$-point dataset $V \subset \ell_p^d$, construct a $(c_p, r)$-ANN $A_{base}$ for $V$; and additionally, for every point $x \in V$, apply a Mazur map $M^x$ scaled down by $\frac{p}{t} \cdot (2rc_p)^{p/t-1}$ from $\ell_p^d$ to $\ell_t^d$ on $B_p(0, 2rc_p) \cap (V - x)$, where $B_p(x, r) \coloneqq \{y : \|x - y\|_p \le r\}$, and construct a $(c_t, r)$-ANN data structure $A_x$ for the image points. Amplify their success probabilities to 5/6 by standard amplification. Given a query $q$, with the guarantee that there exists $x^* \in V$ with $\|x^* - q\|_p \le r$, query $A_{base}$ with $q$ and obtain a point $x \in V$. Then query $A_x$ with $M^x(q - x)$, obtain a point $M^x(z - x) \in M^x(V - x)$ and output $z$ accordingly.

**Claim 4.1.** *With probability* 2/3*, we have* $\|z - q\|_p \le c_{new} r$*, where* $c_{new} = (\frac{p}{t})^{t/p}\ c_t^{t/p}\ (4c_p)^{1-t/p}$.

*Proof.* With probability at least $\frac{5}{6}$, $A_{base}$ outputs a point $x$ with $\|x - q\|_p \le rc_p$. By triangle inequality, $\|x^* - x\|_p \le \|x^* - q\|_p + \|q - x\|_p \le 2rc_p$, hence $\|M^x(x^*) - M^x(q)\|_t \le r$. Thus, with probability at least $\frac{5}{6}$, $A_x$ outputs a point $M^x(z)$ with $\|M^x(z) - M^x(q)\|_t \le rc_t$. By a union bound, both events hold with probability 2/3. Assume they hold. By Theorem 2.1,

$$\frac{t}{p} \cdot (4rc_p)^{1-p/t} \|z-q\|_p^{p/t} \le \|M^x(z) - M^x(q)\|_t \le r \cdot c_t,$$

rearranging this we obtain $\|z-q\|_p \le r(\frac{p}{t})^{t/p}\ c_t^{t/p}\ (4c_p)^{1-t/p} \equiv r \cdot c_{new}$. □

*Remark* 4.2. Plugging $t = 2$ into Theorem 4.1 and solving the recursion, we obtain a variation of [BG19, Lemma 11].

Now, as in the proof of Theorem 1.2, we apply the additional recursive embedding reduction that goes through intermediate $\ell_t$ spaces. To improve readability, we first provide a simpler proof with $O(p^3)$-approximation, and then explain the improvement to $O(p^{1+\ln(4)+\varepsilon})$-approximation. We assume without loss of generality that $p \le \log d$ by Hölder's inequality.

Assume for now that $p$ is a power of 2. Consider the data structure for $\ell_2^d$ given by [Cha98], with approximation $c = \mathrm{poly}(d)$, space and processing time $\tilde{O}(n \cdot \mathrm{poly}(d))$ and query time $\mathrm{poly}(d \log n)$. By Hölder's inequality, the same data structure yields $\mathrm{poly}(d)$ approximation also for $\ell_p^d$.

Now, we recursively apply Theorem 4.1 with $t = p/2$, as follows. Denote by $k$ the number of recursive steps to be determined later, and by $\hat{c}_i$ the approximation guarantee in $\ell_p$ after the $i$-th recursive step. Initially, $\hat{c}_0 = \mathrm{poly}(d)$, by using the data structure of [Cha98]. For every $i \in [k]$, we maintain data structures $\{A_x^i\}_{x \in V}$, where the Mazur map is scaled according to the current approximation guarantee (i.e., scaled down by $\frac{p}{t} \cdot (2r\hat{c}_{i-1})^{p/t-1}$). Moreover, we amplify the success probabilities to $1 - \frac{2}{3k}$ by $O(\log k)$ independent repetitions. Thus, if the $(i-1)$-th iteration is successful, i.e., it returns a point $x$ solving $(\hat{c}_{i-1}, r)$-ANN, then the Mazur maps in the $i$-th iteration are scaled correctly. Hence, by querying $A_x^i$, we get the approximation given by Theorem 4.1. By the law of total probability, with probability 2/3, all the $k$ recursive steps return a correct estimate. Therefore,

$$\begin{aligned}
\hat{c}_k(V) &\le \sqrt{8c_{p/2} \cdot \hat{c}_{k-1}(V)} \\
&\le \sqrt{8c_{p/2} \cdot \sqrt{8c_{p/2} \cdot \hat{c}_{k-2}(V)}} \\
&\le \cdots \\
&\le (8c_{p/2})^{(1/2+1/4+\ldots+1/2^k)}\ \hat{c}_0(V)^{2^{-k}} \\
&\le 8c_{p/2} \cdot \hat{c}_0(V)^{2^{-k}}. \qquad (3)
\end{aligned}$$

Picking $k := \lceil \log(\log p \cdot \log \hat{c}_0(V)) \rceil = O(\log\log d)$ yields $\hat{c}_0(V)^{2^{-k}} \le 2^{1/\log p}$, and we obtain a data structure with approximation at most $\hat{c}_k(V) \le 2^{3+1/\log p} \cdot c_{p/2}$.

Before applying a second recursion on $p$, we amplify the success probabilities to $1 - \frac{2}{3\log p}$ by $O(\log\log p) = O(\log\log\log d)$ independent repetitions. Now a second recursion on $p$ implies $\hat{c}_k(V) \le 2p^3 \cdot c_2$ with probability at least 2/3. Finally, we bound $c_2$ similarly to [BG19], namely, using the JL-lemma to reduce the dimension to $O(\log n)$ together with a $(2, r)$-ANN data structure of [KOR00, HIM12] in $\ell_2^{O(\log n)}$, which has query time $T_2 = \mathrm{polylog}\, n$, and space and preprocessing time $S_2 = Z_2 = n^{O(1)}$. Plugging this as the base case of the second recursion, and we get the desired approximation $\hat{c}_k(V) = O(p^3)$. Each level of the second recursion increases the space and preprocessing time by factor $n$, resulting in a total of $n^{O(\log p)} \cdot S_2 = n^{\log p + O(1)} \cdot d^{O(1)}$ space and preprocessing time. Answering a query goes through both recursions, but the first recursion only requires $O(k \log k) = \tilde{O}(\log \log d)$ calls to an ANN data structure for $\ell_t$, hence the overall running time is $(\log \log d)^{O(\log p)} \cdot T_2 = \mathrm{poly}(d \log n)$. Resolving the case when $p$ is not a power of 2 is straightforward and performed exactly as in the proof of Theorem 1.2, and thus omitted.

To improve the approximation, let $\varepsilon > 0$, and pick $t = (1-\varepsilon)p$ instead of $t = p/2$. We now have that

$$\hat{c}_k(V) \le (\tfrac{1}{1-\varepsilon})^{1-\varepsilon} \; c_t^{1-\varepsilon} \; (4\hat{c}_{k-1}(V))^{\varepsilon} \le \ldots \le (\tfrac{c_t}{1-\varepsilon})^{1-\varepsilon^k} 4^{\frac{\varepsilon(1-\varepsilon^k)}{1-\varepsilon}} \, (\hat{c}_0(V))^{\varepsilon^k}.$$

For sufficiently large $k = O(\log(\varepsilon^{-1}) \log(\log p \cdot \log d))$, we get $\hat{c}_k(V) \le \frac{1}{1-\varepsilon} 4^{\frac{\varepsilon}{1-\varepsilon}} c_t$. Now, a recursion on $p$ for $\log_{\frac{1}{1-\varepsilon}} p = O(\varepsilon^{-1} \log p)$ levels implies

$$\hat{c}_k(V) \le p \cdot \exp\big(\ln(4)(\tfrac{\varepsilon}{1-\varepsilon} \cdot \log_{\frac{1}{1-\varepsilon}} p)\big) c_2 \le p^{1+\ln(4)+O(\varepsilon)} c_2,$$

where the last step uses the inequalities $\frac{1}{1-\varepsilon} \ge 1+\varepsilon$ and $\ln(1+\varepsilon) \ge \frac{\varepsilon}{1+\varepsilon}$. The rest of the proof is the same, and the space and preprocessing time increase to $\mathrm{poly}(dn^{\varepsilon^{-1} \log p})$. Rescaling $\varepsilon$ concludes the proof. □

## 5 Embedding Finite $\ell_p$ Metrics into $\ell_2$

In this section, we prove Theorem 1.9 by providing embeddings of finite $\ell_p$ metrics into $\ell_2$, for $3 < p < 3\sqrt{e}$. We will need the following setup from [NR25].

**Definition 5.1** (Definition 4 in [NR25]). Given $K, D > 1$, we say that a metric space $(\mathcal{M}, d_{\mathcal{M}})$ admits a *$K$-localized weakly bi-Lipschitz embedding* into a metric space $(\mathcal{N}, d_{\mathcal{N}})$ with distortion $D$ if for every $\Delta > 0$ and every subset $\mathcal{C} \subseteq \mathcal{M}$ of diameter $\mathrm{diam}_{\mathcal{M}}(\mathcal{C}) \le K\Delta$, there exists a non-constant Lipschitz function $f_\Delta^{\mathcal{C}} : \mathcal{C} \to \mathcal{N}$ satisfying the following. For every $x, y \in \mathcal{C}$, if $d_{\mathcal{M}}(x,y) > \Delta$, then

$$d_{\mathcal{N}}\left(f_\Delta^{\mathcal{C}}(x), f_\Delta^{\mathcal{C}}(y)\right) > \frac{\|f_\Delta^{\mathcal{C}}\|_{\mathrm{Lip}}}{D} \Delta,$$

where $\|\cdot\|_{\mathrm{Lip}}$ is the Lipschitz constant.

We provide the following simple observation, that composing a localized weakly bi-Lipschitz embedding with a low-distortion embedding yields a localized weakly bi-Lipschitz embedding, as follows.

**Observation 5.2.** *Let $(\mathcal{M}, d_{\mathcal{M}}), (\mathcal{N}, d_{\mathcal{N}}), (\mathcal{Z}, d_{\mathcal{Z}})$ be metric spaces, such that $(\mathcal{M}, d_{\mathcal{M}})$ admits a $K$-localized weakly bi-Lipschitz embedding into $(\mathcal{N}, d_{\mathcal{N}})$ with distortion $D_1$ and $(\mathcal{N}, d_{\mathcal{N}})$ admits an embedding into $(\mathcal{Z}, d_{\mathcal{Z}})$ with distortion $D_2$. Then $(\mathcal{M}, d_{\mathcal{M}})$ admits a $K$-localized weakly bi-Lipschitz embedding into $(\mathcal{Z}, d_{\mathcal{Z}})$ with distortion $D_1 \cdot D_2$.*

*Proof.* Let $\Delta > 0$ and $\mathcal{C} \subseteq \mathcal{M}$ of diameter $\mathrm{diam}_{\mathcal{M}}(\mathcal{C}) \le K\Delta$. Let $f_\Delta^{\mathcal{C}} : \mathcal{C} \to \mathcal{N}$ be the function promised by Theorem 5.1, and $g : (\mathcal{N}, d_{\mathcal{N}}) \to (\mathcal{Z}, d_{\mathcal{Z}})$ be an embedding with distortion $D_2$. Consider $\tilde{f}_\Delta^{\mathcal{C}} := g \circ f_\Delta^{\mathcal{C}}$. Recall that since $g$ has distortion at most $D_2$, there exists $s > 0$ such that for every $u, v \in \mathcal{N}$, we have $\frac{s}{D_2} \cdot d_{\mathcal{N}}(u,v) \le d_{\mathcal{Z}}(g(u), g(v)) \le s \cdot d_{\mathcal{N}}(u,v)$. Since $f_\Delta^{\mathcal{C}}$ is non-constant and $g$ has bounded contraction, $\tilde{f}_\Delta^{\mathcal{C}}$ is non-constant. Let $x, y \in \mathcal{C}$ such that $d_{\mathcal{M}}(x,y) > \Delta$. Hence,

$$d_{\mathcal{Z}}\left(\tilde{f}_\Delta^{\mathcal{C}}(x), \tilde{f}_\Delta^{\mathcal{C}}(y)\right) \ge \frac{s}{D_2} \cdot d_{\mathcal{N}}\left(f_\Delta^{\mathcal{C}}(x), f_\Delta^{\mathcal{C}}(y)\right) > \frac{s \cdot \|f_\Delta^{\mathcal{C}}\|_{\mathrm{Lip}}}{D_1 \cdot D_2} \Delta,$$

where the last inequality follows since $f_\Delta^{\mathcal{C}}$ is a $K$-localized weakly bi-Lipschitz embedding with distortion $D_1$. Since $g$ expands distances by at most a factor $s$, we have $\|\tilde{f}_\Delta^{\mathcal{C}}\|_{\text{Lip}} \leq s \cdot \|f_\Delta^{\mathcal{C}}\|_{\text{Lip}}$, concluding the proof. □

**Lemma 5.3** (Generalization of Lemma 5 in [NR25]). *For every $K > 1$, if $p > q \geq 1$, then $\ell_p$ admits a $K$-localized weakly bi-Lipschitz embedding into $\ell_q$ with distortion $O_{p/q}(K^{p/q-1})$.*

*Proof.* Fixing $K, \Delta > 0$ and a subset $\mathcal{C} \subset \ell_p$ whose $\ell_p$ diameter is at most $K\Delta$, pick an arbitrary point $z \in \mathcal{C}$, and consider the Mazur map $M_{p,q}$ scaled down by $(K\Delta)^{p/q-1}$ on $\mathcal{C} - z$. The lemma follows immediately by Theorem 2.1. □

**Definition 5.4.** The Lipschitz extension modulus $e(\mathcal{M}, \mathcal{N})$ of a pair of metric spaces $\mathcal{M}, \mathcal{N}$ is the infimum over all $L \in [1, \infty)$ such that for every subset $\mathcal{C} \subseteq \mathcal{M}$, every 1-Lipschitz function $f : \mathcal{C} \to \mathcal{N}$ can be extended to an $L$-Lipschitz function $F : \mathcal{M} \to \mathcal{N}$.

**Theorem 5.5** (Theorem 6 in [NR25]). *There is a universal constant $\kappa > 1$ with the following property. Fix $\theta > 0$, an integer $n \geq 3$, and $\alpha > 1$. Let $(\mathcal{M}, d_\mathcal{M})$ be an $n$-point metric space such that every subset $\mathcal{C} \subseteq \mathcal{M}$ with $|C| \geq 3$ admits a $\kappa(\log |\mathcal{C}|)$-localized weakly bi-Lipschitz embedding into $\ell_2$ with distortion $\alpha(\log |\mathcal{C}|)^\theta$. Then*

$$c_2(\mathcal{M}) \leq \alpha \cdot e(\mathcal{M}; \ell_2) \cdot (\log n)^{\max\{\theta, \frac{1}{2}\}} \cdot \log \log n.$$

Next, we show a reduction that takes embeddings of finite $\ell_q$ metrics into $\ell_2$, and constructs an embedding of finite $\ell_p$ metric into $\ell_2$, for $p > q$. The proof constructs a localized weakly bi-Lipschitz embedding of $\ell_p$ into $\ell_q$ and composes it with the given embedding from $\ell_q$ into $\ell_2$. By Theorem 5.2, this yields a localized weakly bi-Lipschitz embedding from $\ell_p$ into $\ell_2$, and by Theorem 5.5, we get a low-distortion embedding into $\ell_2$.

For every $q \in [1, \infty]$, define

$$\xi_q \coloneqq \inf_{\theta \geq 0} \Big\{\theta : \exists \nu > 0, \forall n \geq 2, \quad c_2^n(\ell_q) \leq \nu \cdot \log^\theta n\Big\},$$

where $\xi_q \leq 1$ for all $q \in [1, \infty]$ by Bourgain's embedding [Bou85].

**Lemma 5.6.** *For every $2 \leq q < p$,*

$$\xi_p \leq \max\{\tfrac{1}{2}, \xi_q\} + \tfrac{p}{q} - 1.$$

*Proof.* Let $\delta > 0$ and let $\mathcal{M} \subset \ell_p$ be an $n$-point metric. If $n \leq 2$, then clearly $c_2^n(\ell_p) = 1$. Otherwise, let $\mathcal{C} \subseteq \mathcal{M}$ with $|\mathcal{C}| \geq 3$. We now construct a weakly bi-Lipschitz embedding of $\mathcal{C}$ into $\ell_2$. By Theorems 5.2 and 5.3, we have that for every $K \geq 1$, $\mathcal{C}$ admits a $K$-localized weakly bi-Lipschitz embedding into $\ell_2$ with distortion $O(K^{p/q-1} \cdot c_2^{|\mathcal{C}|}(\ell_q))$. Setting $K = \kappa(\log |\mathcal{C}|)$, where $\kappa$ is the universal constant from Theorem 5.5, and using $c_2^{|\mathcal{C}|}(\ell_q) \leq O_\delta(\log^{\xi_q+\delta} |\mathcal{C}|)$, we obtain a $\kappa(\log |\mathcal{C}|)$-localized weakly bi-Lipschitz embedding of $\mathcal{C}$ into $\ell_2$ with distortion $O_{p,\delta}(\log^{\frac{p}{q}-1+\xi_q+\delta} |\mathcal{C}|)$.

By Theorem 5.5,

$$\begin{aligned}
c_2(\ell_p) &\leq O_{p,\delta}\Big(e(\ell_p;\ell_2)(\log n)^{\max\{\frac{1}{2}, \frac{p}{q}-1+\xi_q+\delta\}} \log\log n\Big) \\
&\leq O_{p,\delta}\Big((\log n)^{\max\{\frac{1}{2}, \frac{p}{q}-1+\xi_q+\delta\}} \log\log n\Big) && e(\ell_p, \ell_2) \leq O(\sqrt{p}) \text{ by [NPSS06]} \\
&\leq O_{p,\delta}\Big((\log n)^{\max\{\frac{1}{2}, \xi_q\}+\frac{p}{q}-1+\delta} \log\log n\Big) && \text{since } \frac{p}{q} - 1 + \delta > 0 \\
&\leq O_{p,\delta}\Big((\log n)^{\max\{\frac{1}{2}, \xi_q\}+\frac{p}{q}-1+2\delta}\Big).
\end{aligned}$$

Since $\delta$ is arbitrary, the lemma follows. $\square$

The reduction given in the lemma above is a single iteration of recursive embedding, and we repeat it recursively to prove Theorem 1.9.

*Proof of Theorem 1.9.* Let $3 < p < 3\sqrt{e}$ and $\varepsilon > 0$. Consider a sequence $q_0, \dots, q_k$, where $q_0 = p$ and $\frac{q_i}{q_{i+1}} = (\frac{p}{3})^{1/k}$ for all $i \in [0, k-1]$. Therefore, $q_k = 3$. By Theorem 5.6 we have,

$$\begin{aligned}\xi_p &\le \max\{\tfrac{1}{2}, \xi_{q_1}\} + \tfrac{p}{q_1} - 1 \\ &\le \max\{\tfrac{1}{2}, \xi_{q_2}\} + \tfrac{p}{q_1} - 1 + \tfrac{q_1}{q_2} - 1 \\ &\dots \\ &\le \max\{\tfrac{1}{2}, \xi_3\} + (\tfrac{p}{q_1} - 1 + \tfrac{q_1}{q_2} - 1 + \dots + \tfrac{q_{k-1}}{q_k} - 1).\end{aligned}$$

By [NR25, Theorem 1], we have $c_2^n(\ell_3) \le O(\sqrt{\log n} \cdot \log\log n)$, and thus $\xi_3 \le \frac{1}{2}$. Therefore,

$$\begin{aligned}&= \tfrac{1}{2} - k + \sum_{i=0}^{k-1} \tfrac{q_i}{q_{i+1}} \\ &= \tfrac{1}{2} - k + k(\tfrac{p}{3})^{1/k} = \tfrac{1}{2} - k + k \cdot \exp(\tfrac{1}{k} \ln \tfrac{p}{3}).\end{aligned}$$

For a suitable choice of $k = O(\varepsilon^{-1})$, and using the useful inequality $e^x \le 1 + x + x^2$ for $x < 1.79$,

$$\begin{aligned}&\le \tfrac{1}{2} - k + k\big(1 + \tfrac{1}{k} \ln \tfrac{p}{3} + (\tfrac{1}{k} \ln \tfrac{p}{3})^2\big) \\ &< \tfrac{1}{2} + \ln \tfrac{p}{3} + \varepsilon.\end{aligned}$$

The theorem follows from the definition of $\xi_p$. $\square$

## 6 Future Directions

**Problems in $\ell_p$, $p < 2$.** Our results for $\ell_p$ spaces are all for $p > 2$. For the other case, $p < 2$, there are natural candidates for intermediate spaces, namely, $\ell_q$ for $p < q < 2$. Can recursive embedding be used in such settings?

**Problems in $\ell_\infty$.** Many problems in $\ell_\infty^d$ can be reduced to $\ell_2^d$ using John's theorem [Joh48], which incurs $O(\sqrt{d})$ multiplicative distortion and is known to be tight. Our method bypasses this limitation and reduces the Lipschitz decomposition problem from $\ell_\infty^d$ to $\ell_2^d$ at the cost of only a polylogarithmic (in $d$) factor. Indeed, the reduction in Theorem 1.2 actually proves (although not stated explicitly) that

$$\beta^*(\ell_\infty^d) \le \mathrm{polylog}(d) \cdot \beta^*(\ell_2^d). \tag{4}$$

Can other problems in $\ell_\infty^d$ be resolved similarly, i.e., through a recursive embedding to $\ell_2^d$ that bypasses the $O(\sqrt{d})$ factor of a direct embedding?

**Lower Bounds.** Our approach of reducing from $\ell_\infty^d$ to $\ell_2^d$ can also establish lower bounds for problems in $\ell_2^d$, which essentially amounts to "pulling" hard instances, from $\ell_\infty^d$ into $\ell_2^d$. For $\beta^*(\ell_2^d)$, a tight bound is already known [CCG+98], and thus (4) cannot yield a new lower bound for it. However, for the extension modulus of $\ell_2^d$, the known bounds are not tight, namely, $\Omega(d^{1/4}) \le e(\ell_2^d) \le O(\sqrt{d})$ [LN05, MN13], and it is conjectured that $e(\ell_2^d) = \Theta(\sqrt{d})$ [Nao17]. Can the known lower bound $e(\ell_\infty^d) \ge \Omega(\sqrt{d})$ be pulled to $\ell_2^d$, analogously to (4)?

**Nearest Neighbor Search.** The space and preprocessing time of our data structure in Theorem 1.8 are not polynomial in $n$ and $d$ whenever $p$ is non-constant. This increase in preprocessing time and space was somewhat mitigated in [BG19] in the special case of doubling metrics. Can this issue be avoided also in the general case?[5]

**Low-Distortion Embeddings.** There remains a gap in our understanding of the distortion required to embed finite $\ell_p$ metrics into $\ell_2$ for every $p \in (2, \infty)$. For the special case of doubling metrics, we know from [BG14, Theorem 5.5] that $c_2(\mathcal{C}) \leq O\left(\sqrt{\mathrm{ddim}(\mathcal{C})^{p/2-1} \log n}\right)$ for every $p \in (2, \infty)$ and every $n$-point metric $\mathcal{C} \subset \ell_p$, where $\mathrm{ddim}(\mathcal{C})$ denotes its doubling dimension. This upper bound above does not match the $\Omega(\log^{1/2-1/p} n)$ lower bound in Theorem 1.10, which actually holds for doubling metrics. We thus ask whether the distortion bound in the doubling case can be improved.

# References


[ABCP98] Baruch Awerbuch, Bonnie Berger, Lenore Cowen, and David Peleg. Near-linear time construction of sparse neighborhood covers. *SIAM Journal on Computing*, 28(1):263–277, 1998. doi:10.1137/S0097539794271898.

[ACP08] Alexandr Andoni, Dorian Croitoru, and Mihai Pătraşcu. Hardness of nearest neighbor under l-infinity. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008*, pages 424–433. IEEE Computer Society, 2008. doi:10.1109/FOCS.2008.89.

[AI06] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *47th Annual IEEE Symposium on Foundations of Computer Science*, pages 459–468. IEEE, 2006. doi:10.1109/FOCS.2006.49.

[AI17] A. Andoni and P. Indyk. Nearest neighbors in high-dimensional spaces. In *Handbook of Discrete and Computational Geometry*, chapter 43, pages 1135–1150. CRC Press, 3rd edition, 2017. doi:10.1201/9781315119601.

[AIK09] A. Andoni, P. Indyk, and R. Krauthgamer. Overcoming the $l_1$ non-embeddability barrier: algorithms for product metrics. In *19th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 865–874. SIAM, 2009.

[AIR19] Alexandr Andoni, Piotr Indyk, and Ilya Razenshteyn. Approximate nearest neighbor search in high dimensions. In *Proceedings of the International Congress of Mathematicians (ICM 2018)*, pages 3287–3318, 2019. doi:10.1142/9789813272880_0182.

[ALN08] S. Arora, J. R. Lee, and A. Naor. Euclidean distortion and the sparsest cut. *J. Amer. Math. Soc.*, 21(1):1–21, 2008.

[AN25] Alexandr Andoni and Negev Shekel Nosatzki. Embeddings into similarity measures for nearest neighbor search, 2025. Accepted to FOCS 2025.

[And09] Alexandr Andoni. *NN search : the old, the new, and the impossible*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2009. URL: https://hdl.handle.net/1721.1/55090.

[ANN+18a] Alexandr Andoni, Assaf Naor, Aleksandar Nikolov, Ilya P. Razenshteyn, and Erik Waingarten. Data-dependent hashing via nonlinear spectral gaps. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018*, pages 787–800. ACM, 2018. doi:10.1145/3188745.3188846.


[5] See Appendix A for a positive resolution of this question.

[ANN+18b] Alexandr Andoni, Assaf Naor, Aleksandar Nikolov, Ilya P. Razenshteyn, and Erik Waingarten. Hölder homeomorphisms and approximate nearest neighbors. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018*, pages 159–169. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00024.

[ANRW21] Alexandr Andoni, Aleksandar Nikolov, Ilya P. Razenshteyn, and Erik Waingarten. Approximate nearest neighbors beyond space partitions. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021*, pages 1171–1190. SIAM, 2021. doi:10.1137/1.9781611976465.72.

[AP90] B. Awerbuch and D. Peleg. Sparse partitions. In *31st Annual IEEE Symposium on Foundations of Computer Science*, pages 503–513, 1990.

[Bal90] K. Ball. Isometric embedding in $l_p$-spaces. *European J. Combin.*, 11(4):305–311, 1990.

[Bar96] Y. Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. In *37th Annual Symposium on Foundations of Computer Science*, pages 184–193. IEEE, 1996.

[BB05] A. Brudnyi and Yu. Brudnyi. Simultaneous extensions of Lipschitz functions. *Uspekhi Matematicheskikh Nauk*, 60(6):53–72, 2005. translation in Russian Math. Surveys 60 (2005), no. 6, 1057–1076. doi:10.1070/RM2005v060n06ABEH004281.

[BB06] Alexander Brudnyi and Yuri Brudnyi. Extension of Lipschitz functions defined on metric subspaces of homogeneous type. *Revista Matemática Complutense*, 19(2):347—-359, 2006. doi:10.5209/rev_REMA.2006.v19.n2.16596.

[BBM+24] Yiqiao Bao, Anubhav Baweja, Nicolas Menand, Erik Waingarten, Nathan White, and Tian Zhang. Average-distortion sketching, 2024. arXiv preprint. arXiv:2411.05156.

[BG14] Yair Bartal and Lee-Ad Gottlieb. Dimension reduction techniques for $\ell_p$, $1 \leq p < \infty$, with applications, 2014. arXiv preprint, see version v2. arXiv:1408.1789v2.

[BG19] Yair Bartal and Lee-Ad Gottlieb. Approximate nearest neighbor search for $\ell_p$-spaces ($2 < p < \infty$) via embeddings. *Theoretical Computer Science*, 757:27–35, 2019. doi:10.1016/j.tcs.2018.07.011.

[BL98] Yoav Benyamini and Joram Lindenstrauss. *Geometric nonlinear functional analysis*, volume 48. American Mathematical Soc., 1998.

[Bou85] J. Bourgain. On Lipschitz embedding of finite metric spaces in Hilbert space. *Israel J. Math.*, 52(1-2):46–52, 1985. doi:10.1007/BF02776078.

[CCG+98] M. Charikar, C. Chekuri, A. Goel, S. Guha, and S. Plotkin. Approximating a finite metric by a small number of tree metrics. In *39th Annual Symposium on Foundations of Computer Science*, pages 379–388, 1998. doi:10.1109/SFCS.1998.743488.

[CGR05] S. Chawla, A. Gupta, and H. Räcke. Improved approximations to sparsest cut. In *16th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 102–111, 2005.

[Cha98] Timothy M. Chan. Approximate nearest neighbor queries revisited. *Discret. Comput. Geom.*, 20(3):359–373, 1998. doi:10.1007/PL00009390.

[CNR24] Alan Chang, Assaf Naor, and Kevin Ren. Random zero sets with local growth guarantees, 2024. arXiv preprint. arXiv:2410.21931.

[DIIM04] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *20th annual symposium on Computational geometry*, pages 253–262. ACM, 2004. doi:10.1145/997817.997857.

[FN22] Arnold Filtser and Ofer Neiman. Light spanners for high dimensional norms via stochastic decompositions. *Algorithmica*, 84(10):2987–3007, 2022. doi:10.1007/s00453-022-00994-0.

[HIM12] Sariel Har-Peled, Piotr Indyk, and Rajeev Motwani. Approximate nearest neighbor: Towards removing the curse of dimensionality. *Theory Comput.*, 8(1):321–350, 2012. doi:10.4086/TOC.2012.V008A014.

[IM98] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *30th Annual ACM Symposium on Theory of Computing*, pages 604–613, 1998. doi:10.1145/276698.276876.

[Ind01] Piotr Indyk. On approximate nearest neighbors under $\ell_\infty$ norm. *J. Comput. Syst. Sci.*, 63(4):627–638, 2001. doi:10.1006/JCSS.2001.1781.

[JL84] W. B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Conference in modern analysis and probability (New Haven, Conn., 1982)*, pages 189–206. Amer. Math. Soc., Providence, RI, 1984.

[Joh48] Fritz John. Extremum problems with inequalities as subsidiary conditions. In *Studies and Essays Presented to R. Courant on his 60th Birthday*. Interscience Publishers, 1948. doi:10.1007/978-3-0348-0439-4_9.

[KNT21] Deepanshu Kush, Aleksandar Nikolov, and Haohua Tang. Near neighbor search via efficient average distortion embeddings. In *37th International Symposium on Computational Geometry, SoCG 2021*, volume 189 of *LIPIcs*, pages 50:1–50:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.SOCG.2021.50.

[KOR00] E. Kushilevitz, R. Ostrovsky, and Y. Rabani. Efficient search for approximate nearest neighbor in high dimensional spaces. *SIAM J. Comput.*, 30(2):457–474, 2000. doi:10.1137/S0097539798347177.

[KP12] Michael Kapralov and Rina Panigrahy. NNS lower bounds via metric expansion for $\ell_\infty$ and EMD. In *39th International Colloquium on Automata, Languages and Programming, ICALP 2012*, volume 7391 of *Lecture Notes in Computer Science*, pages 545–556. Springer, 2012. doi:10.1007/978-3-642-31594-7_46.

[KP25] Robert Krauthgamer and Nir Petruschka. Lipschitz Decompositions of Finite $\ell_p$ Metrics. In *41st International Symposium on Computational Geometry (SoCG 2025)*, volume 332 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 66:1–66:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.SoCG.2025.66.

[KPS25] Robert Krauthgamer, Nir Petruschka, and Shay Sapir. The power of recursive embeddings for $\ell_p$ metrics, 2025. Accepted to FOCS 2025. doi:10.48550/arXiv.2503.18508.

[Lee05] James R. Lee. On distance scales, embeddings, and efficient relaxations of the cut cone. In *16th annual ACM-SIAM symposium on Discrete algorithms*, pages 92–101. SIAM, 2005.

[LLR95] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995. doi:10.1007/BF01200757.

[LN05] James R. Lee and Assaf Naor. Extending Lipschitz functions via random metric partitions. *Inventiones Mathematicae*, 160(1):59–95, 2005. doi:10.1007/s00222-004-0400-5.

[LN13] Vincent Lafforgue and Assaf Naor. A doubling subset of $l_p$ for $p > 2$ that is inherently infinite dimensional. *Geometriae Dedicata*, 172, 2013. doi:10.1007/s10711-013-9924-4.

[Mat97] Jiří Matoušek. On embedding expanders into $l_p$ spaces. *Israel J. Math.*, 102:189–197, 1997. doi:10.1007/BF02773799.

[MN13] Manor Mendel and Assaf Naor. Spectral calculus and Lipschitz extension for barycentric metric spaces. *Analysis and Geometry in Metric Spaces*, 1(2013):163–199, 2013. doi:10.2478/agms-2013-0003.

[Nao14] Assaf Naor. Comparison of metric spectral gaps. *Analysis and Geometry in Metric Spaces*, 2(1):1–52, 2014. doi:10.2478/agms-2014-0001.

[Nao17] Assaf Naor. Probabilistic clustering of high dimensional norms. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017*, pages 690–709. SIAM, 2017. doi:10.1137/1.9781611974782.44.

[Nao24] Assaf Naor. *Extension, separation and isomorphic reverse isoperimetry*, volume 11 of *Memoirs of the European Mathematical Society*. EMS, 2024. doi:10.4171/MEMS/11.

[Ngu13] Huy L. Nguyen. Approximate nearest neighbor search in $\ell_p$. *CoRR*, abs/1306.3601, 2013. URL: http://arxiv.org/abs/1306.3601.

[NPSS06] Assaf Naor, Yuval Peres, Oded Schramm, and Scott Sheffield. Markov chains in smooth Banach spaces and Gromov-hyperbolic metric spaces. *Duke Math. J.*, 134(1):165–197, 2006. doi:10.1215/S0012-7094-06-13415-4.

[NR25] Assaf Naor and Kevin Ren. $\ell_p$ has nontrivial Euclidean distortion growth when $2 < p < 4$, 2025. arXiv preprint. arXiv:2502.10543.

[YYW11] Daren Yu, Xiao Yu, and Anqi Wu. Making the nearest neighbor meaningful for time series classification. *Proceedings of the 4th International Congress on Image and Signal Processing, CISP 2011*, 5:2481–2485, 10 2011. doi:10.1109/CISP.2011.6100672.

# A Fast Nearest Neighbor Search for $\ell_p$ Metrics

## A.1 Introduction

The Nearest Neighbor Search (NNS) problem asks to design a data structure (also called a scheme) that preprocesses an $n$-point dataset $X$ lying in a metric space $\mathcal{M}$, so that given a query point $q \in \mathcal{M}$, one can quickly return a point of $X$ minimizing the distance to $q$ (or approximately minimizing it in the approximate version). The efficiency of such a data structure is evaluated primarily by the amount of space it uses and the time required to answer a query. The preprocessing time is a secondary measure and is usually comparable to the space usage. Because of its central role in areas such as machine learning, data analysis, and information retrieval, NNS has been the subject of extensive research, both practical and theoretical (see, e.g., the surveys [AI17, AIR19]).

It is well known that approximate NNS can be reduced to solving $\mathrm{polylog}(n)$ instances of the approximate *near* neighbor problem [HIM12]. For this reason, we restrict attention to the latter.

**Definition A.1.** The Approximate Near Neighbor problem for a metric space $(\mathcal{M}, d_{\mathcal{M}})$ and parameters $c \geq 1$, $r > 0$, abbreviated $(c,r)$-*ANN*, is the following. Design a data structure that preprocesses an $n$-point subset $X \subseteq \mathcal{M}$, so that given a query $q \in \mathcal{M}$ with $d_{\mathcal{M}}(q, X) \leq r$,[6] it reports $x \in X$ such that

$$d_{\mathcal{M}}(q, x) \leq cr.$$

In a randomized data structure, the reported $x \in X$ satisfies this with probability at least $2/3$.

We focus on the fast query-time regime, which is crucial for modern large-scale applications where datasets are massive and queries must be processed online, and is often modeled by query time $\mathrm{poly}(d \log n)$. In $\ell_p$ spaces, ANN in this regime is well understood for $1 \leq p \leq 2$ [IM98, KOR00, HIM12] and for $p = \infty$ [Ind01, ACP08, KP12]. For $2 < p < \infty$, the situation is less clear: there exists a handful of data structures, each suitable for a different range of $p$, as detailed in Table 2. We present a new scheme for ANN in $\ell_p$, $p > 2$, with fast query time, that offers an improved tradeoff between approximation and space, as follows.

[6] If $d_{\mathcal{M}}(q, X) > r$, it may report anything, where as usual, $d_{\mathcal{M}}(q, X) \coloneqq \min_{x^* \in X} d_{\mathcal{M}}(x^*, q)$.

| Approximation | Space | Reference |
|---|---|---|
| $O(\log^{1/p} n \log\log d)$ | $\mathrm{poly}(nd)$ | [AIK09, And09] |
| $2^{O((\log d)^{2/3}(\log\log d)^{1/3})}$ | $\mathrm{poly}(nd)$ | [ANN+18b] |
| $2^{O(p)}$ | $\mathrm{poly}(nd)$ | [BG19] |
| $p^{O(1)}$ | $n^{O(\log p)}$ | [KPS25] |
| $c$ | $n^{O(p/c)\cdot \log\log n}$ | [BBM+24] |
| $p^{O(1)+\log\log p}$ | $\mathrm{poly}(nd)$ | Theorem A.2 |

Table 2: Known data structures for ANN in $\ell_p$, $p > 2$, in the fast query time regime.

**Theorem A.2.** *Let $p > 2$, $d \geq 1$. Then for every $r > 0$, there is a randomized data structure for $(c, r)$-ANN in $\ell_p^d$, where $c = p^{O(1)+\log\log p}$, that has query time* $\mathrm{poly}(d \log n)$ *and preprocessing (both space and time)* $\mathrm{poly}(dn)$.

Studying ANN in $\ell_p$, $p > 2$, is important both practically and theoretically. Real-world data and applications may motivate norms that emphasize outliers, e.g., for anomaly detection, or alter the presence of "hubs", e.g., to affect classification; such data structures were indeed used for time-series classification [YYW11], see [BG19] for additional references. From a theoretical perspective, the geometry of $\ell_p$ spaces undermines existing algorithmic techniques and requires developing new ones. A key challenge is to bridge between $p = 2$ and $p = \infty$. In the fast query-time regime, this means interpolating between the classical $(O(1), r)$-ANN in $\ell_2$ [HIM12] and the $(O(\log\log d), r)$-ANN in $\ell_\infty$ [Ind01], which both use only $\mathrm{poly}(dn)$ space. It natural to conjecture that $2 < p < \infty$ exhibits an interpolation between these two guarantees, and since $\ell_\infty^d$ is $O(1)$-equivalent to $\ell_{\log d}^d$ by Hölder's inequality, this interpolated data structure is conjectured to achieve $O(\log p)$ approximation using $\mathrm{poly}(nd)$ space. The first step towards this conjecture, in [AIK09, And09], devised a reduction from $\ell_p$ to $\ell_\infty$, and obtained a data structure with $O(\log^{1/p} n \log\log d)$-approximation, which is mainly suited for large values of $p$. A nontrivial reduction, devised in [BG19], reduced $\ell_p$ to $\ell_2$ and obtains $2^{O(p)}$-approximation, which is a major improvement for small values of $p$, although it is doubly-exponentially worse than the conjecture. A more sophisticated reduction, that was devised recently in [KPS25], achieves approximation $\mathrm{poly}(p)$, which is an exponential improvement. However, it goes through multiple intermediate $\ell_t$ spaces $(2 \leq t < p)$ via a recursive argument that increases the space to $n^{O(\log p)}$, much higher than conjectured. Our Theorem A.2 essentially completes the improvement of [KPS25], by decreasing the space complexity back to the conjectured $\mathrm{poly}(dn)$, albeit slightly increasing the approximation to $p^{\log\log p}$. In particular, it resolves a question posed in [KPS25], of whether the recursion can avoid this higher space complexity.

The proof of Theorem A.2 is based on a simple yet powerful enhancement of the known NNS schemes from [BG19, KPS25], which utilizes classical results from [AP90, ABCP98] about sparse covers. We provide an overview of the algorithms of [BG19, KPS25] in Section A.4, along with an intuitive explanation of our approach at the beginning of Section A.7.

### A.2 Related Work

Many of the existing results on approximate nearest neighbor search in $\ell_p$ spaces focus on the case $1 \leq p \leq 2$ [KOR00, IM98, HIM12, DIIM04, AI06, And09, Ngu13]. In this setting, $O(1)$-approximation can be achieved with $\mathrm{poly}(dn)$ preprocessing (space and time), and query time polynomial in $d \log n$ [KOR00, HIM12].

In recent years, significant progress has been made for the case $p \in (2, \infty)$, and current results can be broadly divided into three regimes. The first regime consists of data structures that achieve moderate approximation and query time using near-linear space [And09, ANN+18a, ANN+18b, ANRW21, KNT21, AN25]. The second regime has small approximation factor, say $O(1)$ or even $1 + \varepsilon$, in which case both the query time and the preprocessing requirements (space and time) are typically very large [BG19, BBM+24]. The third regime, which is the focus of our work, has fast query time, namely, $\mathrm{poly}(d \log n)$, and existing results either achieve $2^{O(p)}$-approximation with $\mathrm{poly}(dn)$ space [BG19], or better approximation $o(2^p)$ at the cost of a much bigger $n^{\omega(1)}$ space [BBM+24, KPS25]. Our result is the first to obtain both $o(2^p)$ approximation and $\mathrm{poly}(dn)$ space.

We point out that the techniques used here have been applied successfully also to other problems involving $\ell_p$ spaces, such as the construction of Lipschitz decompositions [KP25, KPS25, NR25], geometric spanners [KP25, KPS25], and low-distortion embeddings [KPS25, NR25].

### A.3 Preliminaries

Given a metric space $\mathcal{M} = (X, d_{\mathcal{M}})$, we denote by $B_{d_{\mathcal{M}}}(x, r) \coloneqq \{y \in X : d_{\mathcal{M}}(x, y) \leq r\}$ the ball of radius $r > 0$ centered at a point $x \in \mathcal{M}$.

For every $p, q \in [1, \infty)$, the Mazur map $M_{p,q} : \ell_p^d \to \ell_q^d$ is computed by taking, in each coordinate, the absolute value raised to power $p/q$, but keeping the original sign. Our algorithm crucially relies on the following property of this map.

**Theorem A.3** ([BL98, BG19])**.** *Let $1 \leq q < p < \infty$ and $C_0 > 0$, and let $M$ be the Mazur map $M_{p,q}$ scaled down by factor $\frac{p}{q} {C_0}^{p/q-1}$. Then for all $x, y \in \ell_p^d$ such that $x, y \in B_{\ell_p}(0, C_0)$,*

$$\tfrac{q}{p}(2C_0)^{1-p/q} ||x - y||_p^{p/q} \leq ||M(x) - M(y)||_q \leq ||x - y||_p.$$

### A.4 Overview of Algorithms from [BG19, KPS25]

In this section, we review the algorithms of [BG19] and [KPS25] for ANN in $\ell_p$, $p > 2$. For the rest of the section, fix a dataset $X \subset \ell_p^d$ with $|X| = n$ for some $p > 2$.

### A.5 $(2^{O(p)}, r)$-ANN with $\mathrm{poly}(dn)$ Space [BG19]

In the preprocessing stage, consider a set of $k = \frac{p}{2} \cdot O(\log \log d)$ possible approximation factors $\hat{\mathcal{C}} = \{\hat{c}_i\}_{i=0}^k$, where $\mathrm{poly}(d) = \hat{c}_0 \geq \hat{c}_1 \geq \cdots \geq \hat{c}_k = 2^{O(p)}$. First, compute for $X$ an initial NNS data structure $A_{\mathrm{init}}$ using [Cha98], which provides approximation $\hat{c}_0 = \mathrm{poly}(d)$ using query time $\mathrm{poly}(d \log n)$ and space $\mathrm{poly}(d)\tilde{O}(n)$.[7] Then, for every data point $x \in X$ and every approximation $\hat{c} \in \hat{\mathcal{C}}$, compute a (scaled) Mazur map $M_{x,\hat{c}} : \ell_p^d \to \ell_2^d$ for the points set $B_{\ell_p}(0, \hat{c}r) \cap (X - x)$. Finally, compute for the image points in $\ell_2^d$ the $(2, r)$-ANN data structure $A_{x,\hat{c}}$ from [HIM12], which uses $\mathrm{poly}(d \log n)$ query time and $\mathrm{poly}(dn)$ space. We have this data structure for each point $x \in X$ and each approximation factor $\hat{c} \in \hat{\mathcal{C}}$, and clearly $|B_{\ell_p}(x, \hat{c}r)| \leq n$, hence the total space requirement is $O(p \cdot \log \log d)n \cdot \mathrm{poly}(dn) = \mathrm{poly}(dn)$.

At query time, given a query point $q \in \ell_p^d$, find a $\hat{c}_0$-approximate solution $x_0$ using $A_{\mathrm{init}}$. The crucial observation is that since the Mazur map ensures a distortion that depends on the diameter of the point set (Theorem A.3), the answer from $A_{\hat{c}_0, x_0}$ is a $\hat{c}_1$-approximate solution $x_1$. Applying this procedure iteratively, the approximation factor decreases even faster than geometrically, roughly as

[7] Throughout, the notation $\tilde{O}(f)$ hides factors that are logarithmic in $f$.

$\hat{c}_i = \hat{c}_{i-1}^{1-2/p}$. Hence, after $k = O(p \log\log c_0)$ iterations we obtain an approximate solution $x_k$ with $\hat{c}_k = 2^{O(p)}$, where the approximation factor does not improve further.

### A.6 $(\mathrm{poly}(p), r)$-ANN with $\mathrm{poly}(d)n^{O(\log p)}$ Space [KPS25]

In [KPS25], the image space of the Mazur map is changed from $\ell_2$ to $\ell_t$ for general $1 \leq t < p$. Generalizing the results from [BG19], a $(c_t, r)$-ANN data structure in $\ell_t^d$ with query time $Q(n)$ and space $S(n)$ is used to construct a $(2^{O(p/t)} \cdot c_t, r)$-ANN data structure for $\ell_p^d$ with query time $O(d) + \frac{p}{t} O(\log\log d) Q(n)$ and space $\frac{p}{t} O(\log\log d) n \cdot S(n)$. Using the above result with $t = p/2$, and applying it recursively to decrease $p$ to 2 (which is actually a double recursion, because we also iterate over the $\hat{c}_i$'s), yields a $(\mathrm{poly}(p), r)$-ANN data structure with query time $\mathrm{poly}(d \log n)$. The caveat is that every application of the recursive step multiplies the space of the data structure by factor $n$, which yields a data structure with space $\mathrm{poly}(d)n^{O(\log p)}$.

### A.7 $(2^{\tilde{O}(\log p)}, r)$-ANN with $\mathrm{poly}(dn)$ Space

In this section, we give the proof of Theorem A.2. We first explain the intuition, and for simplicity we restrict this discussion to reducing the space requirement of [BG19]; reducing the space requirements of [KPS25] is similar in spirit, although more technical.

Revisiting Section A.5, the final solution $x_k$ is obtained by finding iteratively a sequence of intermediate solutions $x_0, x_1, \ldots, x_{k-1}$. Each iteration $i < k$ makes progress by finding a point $x_i$ and restricting the search region to $B_{\ell_p}(x_i, \hat{c}_i r)$, which has bounded diameter, and thus applying a Mazur map on this region has distortion guarantees. It follows that querying the data structure $A_{x_i, \hat{c}_i}$ (computed over $B_{\ell_p}(x_i, \hat{c}_i r) \cap X$) finds a point $x_{i+1}$ and we can restrict the search region even further, to diameter $\hat{c}_{i+1} r$.

The preprocessing phase prepares for the possibility that each point $x \in X$ will serve (at query time) the $\hat{c}_i$-approximate solution, i.e., the search region will be restricted to $B_{\ell_p}(x, \hat{c}_i r)$. To make progress and restrict the search region even further, a data structure $A_{x,\hat{c}_i}$ is constructed for (the points in) this region. Our key idea in Theorem A.2 is that, rather than preparing a *separate* data structure for each search region, the algorithm constructs one *global* collection of data structures that together cover all the possible search regions. For every $\hat{c}_i$, the algorithm constructs a set of ANN data structures computed on a collection of subsets $\mathcal{S} \subseteq 2^X$, such that for every point $x \in X$ there is some $S \in \mathcal{S}$ that contains the search region $B_{\ell_p}(x, \hat{c}_i r) \cap X$. In addition, every $S \in \mathcal{S}$ has diameter at most $\beta \hat{c}_i r$ for some $\beta > 1$. We also want the total number of points in $\mathcal{S}$ (counting repetitions) to be small. The preprocessing algorithm simply stores for every $x \in X$ a reference to a set $S_x \in \mathcal{S}$ with $B_{\ell_p}(x, \hat{c}_i r) \cap X \subseteq S_x$, and at query time, if $x$ serves as a $\hat{c}_i$-approximate solution, the algorithm queries the ANN data structure constructed for $S_x$. Since $S_x$ has a diameter at most $\beta \hat{c}_i r$, this will still cause the search region's diameter to shrink in the next iteration (although by a slightly smaller factor). Since the total number of points in $\mathcal{S}$ is small, the total memory used by all the ANN data structures will be small too.

It remains to show that the preprocessing phase can indeed find efficiently a collection of subsets of $X$ with the above properties. Fortunately, this was shown to be possible in [AP90, ABCP98], and has become a fundamental algorithmic tool with numerous applications in distributed computing, network design, routing, graph algorithms, and metric embeddings.

**Definition A.4** (Sparse Neighborhood Cover [AP90, ABCP98])**.** A $(\beta, r)$-*sparse cover* of a metric space $\mathcal{M} = (X, d_{\mathcal{M}})$ is a collection of subsets (called clusters) $\mathcal{S} \subseteq 2^X$, each of diameter at most $\beta r$, such that for every $x \in X$ there exists $S \in \mathcal{S}$ with $B_{d_{\mathcal{M}}}(x, r) \subseteq S$. The total number of points $\sum_{S \in \mathcal{S}} |S|$ is called the *sparsity* of $\mathcal{S}$.

**Theorem A.5** ([ABCP98]). *There is an algorithm that, given a metric space $\mathcal{M}$ and parameters $\beta > 1$ and $r > 0$, outputs a $(\beta, r)$-sparse cover of $\mathcal{M}$ of sparsity $O(n^{1+1/\beta})$, and runs in $O(n^{2+2/\beta})$ time.*

We are now ready to prove Theorem A.2, largely following the proof structure of [KPS25, Theorem 1.8].

*Proof of Theorem A.2.* Let $X \subset \ell_p^d$ be an $n$-point dataset for some $p \in (2, \infty)$. For clarity of exposition, we assume that $p$ is a power of 2, which can be easily resolved, see [KPS25, Theorem 1.2]. Also, by an application of Hölder's inequality, we may assume that $p \leq \log d$.

We construct the ANN scheme using a doubly-recursive procedure. The first recursion assumes access to an ANN scheme for $\ell_p^d$ that achieves approximation factor $c_{\text{base}}$, and provides a new ANN scheme for $\ell_p^d$ that achieves improved (smaller) approximation $c_{\text{new}}$. This step crucially relies on access to yet another ANN data structure, for $\ell_t^d$, for $t = p/2$, that is actually constructed by the same method. This leads to a second recursion, of constructing ANN data structures for intermediate spaces $\ell_p^d, \ell_{p/2}^d, \ldots, \ell_2^d$, where the space $\ell_2^d$ is known to have ANN data structures with $O(1)$ approximation.

We next describe the first recursion, i.e., how to construct an improved $(c_{\text{new}}, r)$-ANN scheme for $\ell_p^d$ given a $(c_{\text{base}}, r)$-ANN scheme for $\ell_p^d$ and a $(c_t, r)$-ANN scheme for $\ell_t^d$, where $t < p$. In the preprocessing phase, use Theorem A.5 to construct for $X$ a $(\beta, 2c_{\text{base}}r)$-cover $\mathcal{S}$ with sparsity $\tilde{O}(n^{1+1/\beta})$ for $\beta = \log p$. During the construction of $\mathcal{S}$, store for every $x \in X$ a reference to a set $S_x \in \mathcal{S}$ that "covers" it, i.e., $B_{\ell_p}(x, 2c_{\text{base}}r) \cap X \subseteq S_x$, which is guaranteed to exist in a sparse cover. In addition, for every $S \in \mathcal{S}$ designate (arbitrarily) a center point $y \in S$, apply a Mazur map $M^y : \ell_p^d \to \ell_t^d$ scaled down by factor $\frac{p}{t} \cdot (2\beta c_p r)^{p/t-1}$ on $B_{\ell_p}(0, 2\beta c_{\text{base}}r) \cap (X - y)$, and construct for these image points a $(c_t, r)$-ANN scheme $A_{\mathcal{S}}$. Finally, construct a $(c_{\text{base}}, r)$-ANN scheme $A_{\text{base}}$ for $X$, and amplify the success probabilities of both data structures to $5/6$ by the standard method of independent repetitions. Given a query $q$ that is guaranteed to have $x^* \in X$ with $\|x^* - q\|_p \leq r$, query $A_{\text{base}}$ for the point $q$ and obtain an answer $x_{\text{base}} \in X$. Then find its cluster $S_{x_{\text{base}}}$ and this cluster's designated center $y$, query $A_{S_{x_{\text{base}}}}$ for the point $M^y(q - y) \in \ell_t^d$, and use its answer $M^y(z_{\text{out}} - y) \in M^y(X - y)$ to output the corresponding $z_{\text{out}} \in X$.

The next claim is analogous to [KPS25, Claim 4.1], and the main difference is using the sparse cover.

**Claim A.6.** *With probability $2/3$, we have $\|z_{out} - q\|_p \leq c_{new} r$, where $c_{new} = (\frac{p}{t})^{t/p}\ c_t^{t/p}\ (4\beta c_{base})^{1-t/p}$.*

*Proof.* With probability at least $\frac{5}{6}$, the data structure $A_{base}$ outputs a point $x_{\text{base}}$ with $\|x_{\text{base}} - q\|_p \leq c_{\text{base}}r$. Let $S_{x_{\text{base}}}$ be the set in the cover referenced by $x_{\text{base}}$, and let $y \in S_{x_{\text{base}}}$ be its designated center point. Since

$$\|x^* - x_{\text{base}}\|_p \leq \|x^* - q\|_p + \|q - x_{\text{base}}\|_p \leq 2c_{\text{base}}r,$$

we get that $x^* \in B_{\ell_p}(x_{\text{base}}, 2c_{\text{base}}r) \cap X \subseteq S_{x_{\text{base}}}$. Observe that by Theorem A.3, $\|M^y(x^*) - M^y(q)\|_t \leq r$, and thus with probability at least $\frac{5}{6}$, querying $A_{S_y}$ finds a point $M^y(z)$ with $\|M^y(z) - M^y(q)\|_t \leq c_t r$. Applying a union bound, we see that with probability at least $2/3$, both events hold. In this case, we have by Theorem A.3, that

$$\frac{t}{p} \cdot (4\beta c_{\text{base}}r)^{1-p/t} \cdot \|z_{\text{out}} - q\|_p^{p/t} \leq \|M^y(z_{\text{out}}) - M^y(q)\|_t \leq c_t r,$$

and by rearranging, we obtain $\|z_{\text{out}} - q\|_p \leq (\frac{p}{t})^{t/p}\ c_t^{t/p}\ (4\beta c_{\text{base}})^{1-t/p} r = c_{\text{new}} r$. ∎

Denote by $\hat{c}_i$ the approximation of the ANN scheme obtained by $i$ applications of Theorem A.6, where the initial ANN scheme is the one from [Cha98], with approximation $\hat{c}_0 = \mathrm{poly}(d)$. We also denote by $c_t$ the approximation of an ANN scheme for $\ell_t$, that is constructed by the same method (i.e., recursively), except that for $\ell_2^d$ we use a $(2, r)$-ANN scheme from [HIM12] with $\mathrm{poly}(d \log n)$ query time and $\mathrm{poly}(dn)$ space and preprocessing time. Using Theorem A.6 with $t = p/2$, and furthermore applying this recursively $k = \lceil \log(\log \hat{c}_0) \rceil = O(\log \log d)$ times, we obtain

$$\hat{c}_k \leq \sqrt{8\beta c_{p/2} \hat{c}_{k-1}} \leq \sqrt{8\beta c_{p/2} \sqrt{8\beta c_{p/2} \hat{c}_{k-2}}} \leq \ldots \leq 8\beta c_{p/2} \cdot \hat{c}_0^{1/2^k} \leq 16\beta c_{p/2},$$

i.e., this scheme has approximation $c_p = \hat{c}_k \leq 16\beta c_{p/2}$. We can amplify the success probability of this scheme to $1 - \frac{1}{3\log p}$. by the standard method of $O(\log \log p) = O(\log \log \log d)$ independent repetitions. Now by recursion over $p$ for $\log p$ levels, we get that

$$c_p \leq (16\beta)^{\log p} = (16 \log p)^{\log p} = p^{4 + \log \log p},$$

and the overall success probability is at least $\frac{2}{3}$ by a union bound.

We are left to analyze the query time of the algorithm, and its space and preprocessing time. Each level of the second recursion makes a total of $k \cdot O(\log \log p) = \tilde{O}(\log \log d)$ calls to an ANN scheme for $\ell_t$, for different intermediate values of $t$. Since the $(2, r)$-ANN for $\ell_2$ from [HIM12] has query time $\mathrm{poly}(d \log n)$, and recalling that $p \leq \log d$, the overall query time is $\tilde{O}(\log \log d)^{\log p} \cdot \mathrm{poly}(d \log n) = \mathrm{poly}(d \log n)$.

To analyze the space and preprocessing time, we prove the following claim.

**Claim A.7.** *There exists an absolute constant $D > 1$ such that when the data structure for $\ell_p^d$, $p = 2^i$, is computed on $m$ points, it uses total space $\tilde{O}\big((\log \log d)^i\big) \cdot \mathrm{poly}(d) m^{D(1+1/\log p)^{i+1}}$, and preprocessing time $\tilde{O}\big((\log \log d)^i + d \cdot i\big) \cdot \mathrm{poly}(d) m^{2D(1+1/\log p)^{i+1}}$.*

*Proof.* We only analyze the space usage of the data structure; the analysis of the preprocessing time follows similarly, as it takes $O(d)$ time to compute a Mazur map and $O(m^{2(1+1/\log p)})$ time to compute a sparse cover.

The proof proceeds by induction on $i \geq 0$. For $i = 0$, the claim follows because the $(2, r)$-ANN from [HIM12], when computed on $m$ points, uses at most $\mathrm{poly}(d) m^D$ space for some absolute constant $D > 1$. Now, assume the claim holds for $i - 1 \geq 0$. The ANN scheme at level $i$ of the recursion consists of two types of ANN schemes. The first type is an ANN scheme from [Cha98] computed on all $m$ points, which uses $\mathrm{poly}(d) \tilde{O}(m) \leq \mathrm{poly}(d) m^{D(1+1/\log p)}$ space. The second type are multiple ANN schemes at level $i - 1$ that are computed on different subsets of the $m$ points. For every $0 \leq j \leq k = \tilde{O}(\log \log d)$, let $\mathcal{S}^j$ be the $(\log p, 2c_j r)$-cover of sparsity $O(m^{1+1/\log p})$ computed for the points at the $j$-th level of the first recursion. For every level $j$ of the first recursion and cluster $S \in \mathcal{S}^j$, the algorithm computes a data structure of level $i - 1$ on $S$. By the induction hypothesis, the space of this data structure is $\tilde{O}\big((\log \log d)^{i-1}\big) \mathrm{poly}(d) |S|^{D(1+1/\log p)^i}$. Observe that the function $f : x \mapsto x^{D(1+1/\log p)^i}$ satisfies that $f(a) + f(b) \leq f(a + b)$ for all $a, b \geq 0$. Thus, the total memory of the recursive data structure is at most

$$\begin{aligned}
&\mathrm{poly}(d) m^{1+1/\log p} + \sum_{j=0}^{\tilde{O}(\log\log d)} \sum_{S \in \mathcal{S}^j} \tilde{O}\big((\log\log d)^{i-1}\big)\,\mathrm{poly}(d) |S|^{D(1+1/\log p)^i} \\
&\qquad \le \mathrm{poly}(d)\tilde{O}\big((\log\log d)^{i-1}\big) \cdot \sum_{j=0}^{\tilde{O}(\log\log d)} \Big[f(m) + \sum_{S\in\mathcal{S}^j} f(|S|)\Big] \\
&\qquad \le \mathrm{poly}(d)\tilde{O}\big((\log\log d)^{i-1}\big) \cdot \sum_{j=0}^{\tilde{O}(\log\log d)} f\Big(m + \sum_{S\in\mathcal{S}^j} |S|\Big) \\
&\qquad \le \mathrm{poly}(d)\tilde{O}\big((\log\log d)^{i}\big) \cdot O(m)^{D(1+1/\log p)^{i+1}},
\end{aligned}$$

and the bound $O(1)^{D(1+1/\log p)^{i+1}} \le O(1)^{2eD} \le O(1)$ completes the proof. □

Finally, we use Theorem A.7 for $m = n$ and $i = \log p$, and obtain that the space usage of the data structure is bounded by

$$O(\log\log d)^{\log p}\,\mathrm{poly}(d) \cdot n^{D(1+1/\log p)^{\log p}} \le \mathrm{poly}(d) \cdot n^{De} = \mathrm{poly}(dn),$$

which completes the proof of Theorem A.2. □

*Remark* A.8. Modifying the parameter $\beta$ of the sparse cover in the proof of Theorem A.2 from $\log p$ to $\frac{\log p}{\delta}$ for $0 < \delta < 1$ yields a data structure with a slightly larger approximation factor $p^{O(1)+\log(1/\delta)+\log p}$, but with a space requirement that matches that of [HIM12] up to subpolynomial factors in $d$ and an additional $n^{O(\delta)}$ term.

*Remark* A.9. The same technique used in the proof of Theorem A.2, namely applying Theorem A.5 to construct covers in the preprocessing phase, can also be used to improve the space requirements of the ANN for general normed spaces from [ANN+18b, Theorem 3]. More specifically, for every $0 < \delta < 1$, one can shave an $\Omega(n^{1-\delta})$ factor from the space of the data structure, at the cost of an additional $O(\delta^{-1})$ factor in the approximation.