Low Sensitivity Hopsets

Vikrant Ashvinkumar* Aaron Bernstein* Chengyuan Deng* Jie Gao*
Nicole Wein†

Abstract

Given a weighted graph $G = (V, E, w)$, a (β, ε)-hopset H is an edge set such that for any $s, t \in V$, where s can reach t in G, there is a path from s to t in $G \cup H$ which uses at most β hops whose length is in the range $[\mathsf{dist}_G(s,t), (1+\varepsilon)\mathsf{dist}_G(s,t)]$. We break away from the traditional question that asks for a hopset H that achieves small $|H|$ and small diameter β and instead study the *sensitivity* of H, a new quality measure. The sensitivity of a vertex (or edge) given a hopset H is, informally, the number of times a single hop in $G \cup H$ bypasses it; a bit more formally, assuming shortest paths in G are unique, it is the number of hopset edges $(s,t) \in H$ such that the vertex (or edge) is contained in the unique st-path in G having length exactly $\mathsf{dist}_G(s,t)$. The sensitivity associated with H is then the maximum sensitivity over all vertices (or edges). The highlights of our results are:

- A construction for $(\widetilde{O}(\sqrt{n}), 0)$-hopsets on undirected graphs with $O(\log n)$ sensitivity, complemented with a lower bound showing that $\widetilde{O}(\sqrt{n})$ is tight up to polylogarithmic factors for any construction with polylogarithmic sensitivity.
- A construction for $(n^{o(1)}, \varepsilon)$-hopsets on undirected graphs with $n^{o(1)}$ sensitivity for any $\varepsilon > 0$ that is at least inverse polylogarithmic, complemented with a lower bound on the tradeoff between β, ε, and the sensitivity.
- We define a notion of sensitivity for β-shortcut sets (which are the reachability analogues of hopsets) and give a construction for $\widetilde{O}(\sqrt{n})$-shortcut sets on *directed* graphs with $O(\log n)$ sensitivity, complemented with a lower bound showing that $\beta = \widetilde{\Omega}(n^{1/3})$ for any construction with polylogarithmic sensitivity.

We believe hopset sensitivity is a natural measure in and of itself, and could potentially find use in a diverse range of contexts. More concretely, the notion of hopset sensitivity is also directly motivated by the Differentially Private All Sets Range Queries problem [Deng et al. WADS 23]. Our result for $O(\log n)$ sensitivity $(\widetilde{O}(\sqrt{n}), 0)$-hopsets on undirected graphs immediately improves the current best-known upper bound on utility from $\widetilde{O}(n^{1/3})$ to $\widetilde{O}(n^{1/4})$ in the pure-DP setting, which is tight up to polylogarithmic factors.

*Rutgers University, `{va264,aaron.bernstein,jg1555,cd751}@rutgers.edu` Ashvinkumar, Deng and Gao would like to acknowledge funding through NSF IIS-2229876, DMS-2220271, DMS-2311064, CCF-2208663, CCF-2118953. Bernstein would like to acknowledge funding through Sloan Fellowship, Google Research Award, and NSF Grant 1942010.

†University of Michigan, `nswein@umich.edu`. This work was done while the author was at DIMACS, Rutgers University, supported by a grant to DIMACS from the Simons Foundation (820931).

Contents

1 Introduction

Many fundamental graph-theoretic problems involve the computation of reachability and shortest path distances. In a variety of computation models (e.g. parallel, distributed, streaming, dynamic), the computation of shortest paths of most state-of-the-art algorithms scales with the *hop-diameter* [KS97, HKN14, HKN15, FN18, JLS19, Fin18, GWN20, BGWN20, CFR20, CFR21, KS21]. The hop-diameter of a graph refers to the maximum hop-length over reachable pairs of vertices, where the hop-length from vertex s to vertex t is the minimum number of edges in a shortest st-path. Motivated by the goal of reducing hop-diameter, Thorup introduced the notion of a *shortcut set* [Tho93], which is a set of edges H added to a given graph G such that: $G \cup H$ has the same reachability structure as G (i.e. for all vertices s and t, s can reach t in $G \cup H$ if and only if s can reach t in G) and $G \cup H$ has small hop-diameter. The shortcut set was later generalized to the *hopset* by Cohen [Coh00] (also informally by [KS97, SS99]) which preserves (weighted) shortest distances in addition to reachability. Formal definitions are given below.

Definition 1.1 (*Shortcut Set*)**.** Given a graph $G = (V, E)$, a *β-shortcut set* of G is a set of edges $H \subseteq V \times V$ such that: (1) Every edge $(s, t) \in H$ is in the transitive closure of G. (2) For every vertex pair (s, t) in the transitive closure of G, there is an st-path in $G \cup H$ with at most β edges.

Definition 1.2 (*Hopset*)**.** Given a weighted graph $G = (V, E, w)$, a *(β, ε)-hopset* of G is a set of weighted edges $H \subseteq V \times V$ such that: (1) Every edge $(s, t) \in H$ has a weight of $\mathsf{dist}_G(s, t)$, where $\mathsf{dist}_G(s, t)$ stands for the shortest distance between s, t in graph G. (2) For every vertex pair (s, t) in the transitive closure of G, there is an st-path P_{st} in $G \cup H$ with at most β edges, and the weight of P_{st} is at most $(1 + \varepsilon)\mathsf{dist}_G(s, t)$.

A $(\beta, 0)$-hopset is sometimes called an exact hopset, while (β, ε)-hopsets for $\varepsilon > 0$ are called approximate hopsets. For both shortcut sets and hopsets, a natural measure of their *cost* is their size (i.e. the number of edges added). There has been a rich literature studying the tradeoff between the size of a shortcut/hopset and the hop-diameter for both directed and undirected graphs [UY90, Hes03, HP19, EN19, LWWX22, KP22c, KP22a, KP22b, SN21, BLP20, ABP18, BW23, BH23, WXX24].

A New Problem: Low-Sensitivity Hopsets. In some settings, measuring the cost of a hopset by its total number of edges provides information that is too coarse and not descriptive enough to capture the problem. In this work, we instead consider a notion of cost that is more local to individual vertices and edges. Specifically, we define the notion of the *sensitivity* of a hopset[1].

Informally, given a graph $G = (V, E)$ and a hopset H, the sensitivity of a vertex v is the number of hopset edges that bypass v. We say a hopset edge $(s, t) \in H$ bypasses v if v is on the unique[2] st-path in G having length exactly $\mathsf{dist}_G(s, t)$. The vertex sensitivity associated with H is then the maximum sensitivity over all vertices. The edge sensitivity is defined in a similar way. We denote the hopset vertex/edge sensitivity by $\|\widehat{\mathsf{v}}\|_\infty$ and $\|\widehat{\mathsf{e}}\|_\infty$[3], and defer the formal definitions to Section 3.1.

We say a hopset H has low sensitivity if it has a vertex/edge sensitivity of polylog(n). We also define an analogous notion of a *low-sensitivity shortcut set*. Since shortcut sets pertain to reachability instead of shortest paths, and there could be many paths between a vertex pair s, t, it is not clear a priori which path should absorb the sensitivity of a shortcut edge from s to t. We

[1]To avoid redundancy, we use hopset as a general term for shortcut/approximate/hopset unless specified otherwise.

[2]We assume shortest paths are unique now. Later we formally define the conditions for non-unique shortest paths.

[3]The arc is meant to be evocative of bypassing hopset edges. When they bypass vertices, we draw the arc over the symbol v and when they bypass edges, we draw the arc over the symbol e.

allow the algorithm to specify the paths; that is, the input is a graph, and the output is a path between each pair of vertices as well as a shortcut set that has low sensitivity with respect to the chosen paths.

A closely related concept is the *support size* of a hopset, first studied in [ES23] to get recently improved path-reporting distance oracles (PRDOs); see [CZ24] for the latest on PRDOs. The support size of a hopset H is, loosely speaking, the number of edges in G that are bypassed by edges from H. This can be seen as the ℓ_0 norm of a certain vector whereas, in contrast, hopset sensitivity is the ℓ_∞ norm of the same vector (as the notation $\|\widehat{\mathsf{e}}\|_\infty$ suggests[4]).

At any rate, we investigate tradeoffs between hopset sensitivity and hop-diameter in this work. We believe hopset sensitivity is a natural measure in and of itself, and could potentially find use in a diverse range of contexts. In fact, low-sensitivity hopsets have already been implicitly studied in several prior works [DGUW23, CGK+23, FL22]. These prior works come from the area of differential privacy where sensitivity is a crucial concept since it captures the effect of the perturbation of individual data points on the output of an algorithm. Our new low-sensitivity hopset constructions directly improve the bounds from [DGUW23] for the problem of *Differentially Private Range Queries*. More details on this concrete application, as well as other applications of a more speculative nature, can be found in Section 1.2.

1.1 Our Results

We study upper and lower bounds for low-sensitivity shortcut sets, exact hopsets, and approximate hopsets, on both undirected and directed graphs.

To give context to our results, we draw parallels to the different regimes for traditional hopsets. For hopsets with $O(n)$ edges, there are essentially three diameter regimes: (1) directed shortcut/hopsets as well as undirected exact hopsets all provably require polynomial hop-diameter, (2) *approximate* hopsets for undirected graphs require only $n^{o(1)}$ hop-diameter, and (3) undirected shortcut sets trivially achieve diameter 2. Roughly speaking, these diameter regimes also hold for the low-sensitivity counterparts.

Our results focus on low-sensitivity undirected exact hopsets and directed shortcut sets from regime 1, as well as low-sensitivity approximate undirected hopsets from regime 2. For all of these, we prove both upper and lower bounds.

An overview of upper and lower bound results of all settings is shown in Table 1. Observe that the edge sensitivity of a hopset is always no more than the vertex sensitivity (formally proved in Section 3.2). Thus, it is always more desirable to have upper bounds for vertex sensitivity and lower bounds for edge sensitivity. All of our results are of this form.

Undirected Exact Hopsets. Our first result is for low-sensitivity exact hopsets on undirected graphs. We show it is possible to achieve hop-diameter $\widetilde{O}(\sqrt{n})$ and $O(\log n)$ vertex sensitivity.

Theorem 1 (Undirected Exact Hopset Upper Bound)**.** *There exists an algorithm producing an* $(O(\sqrt{n}\log n), 0)$*-hopset with* $\|\widehat{\mathsf{v}}\|_\infty = O(\log n)$ *over undirected and directed acyclic graphs.*

We complement our upper bound with a lower bound showing that a hop-diameter of $\widetilde{O}(\sqrt{n})$ is *tight* up to polylogarithmic factors, for any hopset with polylogarithmic (vertex- or edge-) sensitivity.

Theorem 2 (Undirected Exact Hopset Lower Bound)**.** *Any construction of* $(\beta, 0)$*-hopsets* H *must have* $\|\widehat{\mathsf{e}}\|_\infty \cdot \beta^2 = \Omega(n)$ *for a graph* G *on* n *vertices.*

[4]In fact, we also look at the ℓ_1 norm in the proofs of our lower bounds in Section 6.

Table 1: Our Results for sensitivity and hop-diameter for shortcut sets, exact hopsets, and $(1+\varepsilon)$ hopsets. Regarding the undirected approximate hopset, the upper bound has $\varepsilon > 0$ at least inverse polylogarithmic and, in the lower bound, k is any positive integer and Δ is any small positive constant.

		Shortcut Set	Hopset	$(1+\varepsilon)$ Hopset
Undirected	U. B.	$\beta = O(\log^2 n)$ $\lVert\widehat{\mathrm{v}}\rVert_\infty = O(\log n)$ ([FL22], Appendix C)	$\beta = O(\sqrt{n}\log n)$ $\lVert\widehat{\mathrm{v}}\rVert_\infty = O(\log n)$ (Section 4.2)	$\beta = n^{o(1)}$ $\lVert\widehat{\mathrm{v}}\rVert_\infty = n^{o(1)}$ (Section 5)
	L. B.	-	$\lVert\widehat{\mathrm{e}}\rVert_\infty \cdot \beta^2 = \Omega(n)$ (Section 6.1)	$\beta = O_k((1/\varepsilon)^k)$ $\Rightarrow \lVert\widehat{\mathrm{e}}\rVert_\infty = \Omega(n^{\frac{1}{2^k-1}-\Delta})$ (Section 6.2)
Directed	U. B.	$\beta = O(\sqrt{n}\log^3 n)$ $\lVert\widehat{\mathrm{v}}\rVert_\infty = O(\log n)$ (Section 4.3)	$\beta = O(\sqrt{n}\log n)$ $\lVert\widehat{\mathrm{v}}\rVert_\infty = O(\sqrt{n}\log n)$ ([DGUW23, CGK+23], Appendix D)	$\beta = O(\sqrt{n}\log n)$ $\lVert\widehat{\mathrm{v}}\rVert_\infty = O(\sqrt{n}\log n)$ (Implied by directed exact hopset)
	L. B.	$\lVert\widehat{\mathrm{e}}\rVert_\infty \cdot \beta = \Omega(n^{1/3})$ (Section 6.1)	$\lVert\widehat{\mathrm{e}}\rVert_\infty \cdot \beta^2 = \Omega(n)$ (Implied by undirected hopset)	$\lVert\widehat{\mathrm{e}}\rVert_\infty \cdot \beta = \Omega(n^{1/3})$ (Implied by directed shortcut set)

Our lower bound exhibits a smooth trade-off between hop-diameter and sensitivity. One could ask whether such a trade-off exists for upper bounds as well. As we show (see Remark 6.3), such a trade-off would imply the non-existence of certain *perfect path* systems, which is a big open problem in network design. The existence of such perfect path systems would imply, for example, better lower bounds for distance preservers [CE06].

Directed Shortcut Sets. For directed shortcut sets, we prove an upper bound with similar guarantees to our upper bound for undirected exact hopsets.

Theorem 3 (Directed Shortcut Set Upper Bound)**.** *There exists an algorithm producing an $O(\sqrt{n}\log^3 n)$-shortcut set with $\lVert\widehat{\mathrm{v}}\rVert_\infty = O(\log n)$ for directed graphs.*

We complement our upper bound with a lower bound.

Theorem 4 (Directed Shortcut Set Lower Bound)**.** *Any construction of β-shortcut sets H must have $\lVert\widehat{\mathrm{e}}\rVert_\infty \cdot \beta = \Omega(n^{1/3})$ for a directed graph G on n vertices.*

Unlike our bounds for exact undirected hopsets, our results for directed shortcut sets are not tight. In particular, for polylogarithmic vertex-sensitivity, there is a gap between hop-diameter $\widetilde{O}(n^{1/3})$ and $\widetilde{O}(\sqrt{n})$, which we leave as an open problem.

Undirected Approximate Hopsets. For undirected approximate hopsets, we prove that one can achieve much better hop-diameter than the polynomial bounds for the previously mentioned problems. In particular, we prove bounds of $n^{o(1)}$ for both sensitivity and hop-diameter when $\varepsilon > 0$ is at least inverse polylogarithmic.

Theorem 5. *There exists an algorithm which produces a $\big(O\big((k/\varepsilon)^k \log^2 n\big), \varepsilon\big)$-hopset H with $\lVert\widehat{\mathrm{v}}\rVert_\infty = O(kn^{1/k}\log^2 n)$ for undirected graphs. For any $\varepsilon > 0$ that is at least inverse polylogarithmic, setting $k = \Theta(\sqrt{\log n})$ gives a $(n^{o(1)}, \varepsilon)$-hopset with $\lVert\widehat{\mathrm{v}}\rVert_\infty = n^{o(1)}$.*

We complement our upper bound with a lower bound.

Theorem 6. *Fix a positive integer k and parameter $\varepsilon > 1/n^{o(1)}$. Any construction of (β, ε)-hopsets H with $\beta = O\left(\left(\frac{1}{2^{k-2}(2k-1)\varepsilon}\right)^k\right)$ has $\|\widehat{\mathsf{e}}\|_\infty \geq n^{\frac{1}{2^k-1}-\Delta}$, $\Delta > 0$.*

This lower bound is an exact analogue of the best known lower bound on the size of approximate hopsets in [ABP18] (we just divide their size bound by n to get our sensitivity bound). For $\varepsilon > 0$ exactly inverse polylogarithmic, this lower bound says that it is not possible to have sensitivity and hop-diameter be polylogarithmic simultaneously, so we can only hope to improve either the sensitivity or diameter under this regime. We leave open whether we can get sensitivity and diameter simultaneously polylogarithmic when ε is larger than polylogarithmic (for example, constant ε).

To briefly address the problems that we do not focus on (recall that we focus on undirected exact hopsets, directed shortcut sets, and approximate undirected hopsets): Low-sensitivity undirected shortcut sets are simple (but not quite as trivial as in the traditional setting) as noted by prior work and in Observation 3.10. For directed hopsets, the upper bounds implicit in prior work [DGUW23, CGK+23] remain the best-known (see Appendices C and D), and from the lower bounds side, our results for undirected exact hopsets and directed shortcut sets immediately carry over to exact and approximate directed hopsets, respectively.

1.2 Applications

Our motivation for studying low-sensitivity hopsets is two-fold. The more concrete motivation is that they are already implicitly used in the problem of differentially private range queries, and our new upper bound for low-sensitivity hopsets (Theorem 1) immediately implies improved results for this problem (details below).

More generally, we believe that sensitivity is a natural measure for evaluating hopsets, as it is one way of modeling the "robustness" of a hopset: low sensitivity means that changing a vertex/edge in the underlying graph G only affects a small number of shortcuts in H. This corresponds, for example, to the following real-world scenario. In computer networking the notion of an overlay network [Tar10] considers logical links layered on top of a physical network. The logical links support functionality in the overlay protocol but they are actually implemented along a physical path in the underlying network. Overlay networks have been widely adopted in practice (e.g. VPN, VoIP, content delivery, P2P services) due to benefits such as encapsulation, ease of deployment, and quality of service requirements. The design of an overlay network could benefit from the resilience of low sensitivity hopsets – changes in one vertex or one edge in the underlying network only affect relatively few overlay links.

1.2.1 Differentially Private Range Query

Low-sensitivity hopsets have (implicitly) found applications in differential privacy (DP) [DR14]; this is the problem setting with which we are primarily motivated by. On a high level, differential privacy protects sensitive information by introducing perturbation, such that the output stays immune to the presence or absence of any individual's data. More formally,

Definition 1.3 (*Differential Privacy*)**.** For databases Y and Y' that differ on one data entry and $\epsilon > 0, \delta \geq 0$, a randomized algorithm $\mathcal{M}$ is (ϵ, δ)-*differentially private*, if for any measurable set A in the range of possible outputs, it holds that: $\Pr[\mathcal{M}(Y) \in A] \leq e^\epsilon \Pr[\mathcal{M}(Y') \in A] + \delta$.

The algorithm $\mathcal{M}$ is called pure-DP if $\delta = 0$ and approximate-DP otherwise[5]. Low-sensitivity

[5]Note that we are using 'epsilon' in two different ways in this paper, but we disambiguate the symbols: ϵ for the

hopsets have a direct application to the *All Sets Range Queries (ASRQ)* problem in differential privacy [DGUW23]. The input to ASRQ is an undirected graph G with public topology and shortest paths, and each edge is associated with a private attribute (which can be distinct from its weight). The output of ASRQ is an $n \times n$ matrix M, where entry (u, v) contains the summation of edge attributes along a shortest path between the vertex pair u, v. The DP-ASRQ problem asks for a private mechanism to output a matrix M' with the DP guarantee, while minimizing the utility or additive error, i.e. ℓ_∞ of $M - M'$. The best-known upper bound of the additive error from prior work is $\widetilde{O}(n^{1/3})$ for the pure-DP setting and $\widetilde{O}(n^{1/4})$ for the approximate-DP setting. The latter is tight up to polylogarithmic factors [BDG+24]. As we outline next, our low-sensitivity exact hopsets imply an improvement of the $\widetilde{O}(n^{1/3})$ bound for the pure-DP setting, down to the tight bound of $\widetilde{O}(n^{1/4})$, matching the approximate-DP setting.

We are able to show the connection between low-sensitivity hopsets and the DP-ASRQ problem by the following theorem, which is shown implicitly in [DGUW23].

Theorem 7. *If there exists a $(\beta, 0)$-hopset H with edge-sensitivity $\|\widehat{\mathsf{e}}\|_\infty$ for any given undirected graph, then for any $\epsilon > 0$, there exists an ϵ-DP algorithm for the ASRQ problem such that the additive error is at most $\widetilde{O}(\frac{1}{\epsilon} \cdot \sqrt{\|\widehat{\mathsf{e}}\|_\infty \cdot \beta})$ with high probability.*

Plugging into Theorem 7 our $(\sqrt{n}\log n, 0)$-hopset with $\|\widehat{\mathsf{v}}\|_\infty = O(\log n)$ (and thus $\|\widehat{\mathsf{e}}\|_\infty = O(\log n)$) from Theorem 1, we obtain the following result:

Theorem 8 (ϵ-DP upper bound)**.** *There exists an ϵ-DP algorithm for the ASRQ problem such that the additive error is at most $\widetilde{O}(\frac{n^{1/4}}{\epsilon})$ with high probability.*

We show the proof of Theorem 8 in Appendix E. We remark that low-sensitivity hopsets are also implicitly used in another problem studied in differential privacy: All Pairs Shortest Distances. However a direct relation like Theorem 7 is not available. We further discuss this problem in Appendix E.

1.3 Organization

We first provide, in Section 2, a high-level overview of the technical challenges and ideas in our results. Formal preliminaries are then given in Section 3. Next, we show an upper bound for undirected exact hopsets and directed shortcut sets in Section 4 and undirected approximate hopsets in Section 5. In Section 6, we show lower bounds on the sensitivity-diameter tradeoff for shortcut sets and hopsets. We finish by listing several open problems in Section 7.

2 Technical Overview

In this section we outline the key technical ideas in our constructions. A natural starting point for proving upper and lower bounds for low-sensitivity hopsets is to simply look at bounds for traditional hopsets. As it turns out, these ideas can sometimes be massaged into the low-sensitivity setting, but they do not always give the best sensitivity bounds, which necessitates the development of new techniques.

We begin by discussing a technique from prior work on low-sensitivity hopsets that illuminates a key difference between the traditional and low-sensitivity settings.

Techniques from Prior Work: Heavy-Light Decomposition. We will start with a simple example: shortcut sets for undirected graphs. For traditional shortcut sets, this is a trivial problem.

differential privacy parameter, and ε for the approximation parameter of hopsets.

Simply add a star centered at an arbitrary vertex v to get a shortcut set with $n-1$ edges and diameter only 2. If we consider the sensitivity of this construction, we quickly realize that v has sensitivity $n-1$. Thus, this trivial solution does not work in the low-sensitivity setting. However, as noted implicitly in prior work [FL22], there is quite a simple construction that does work.

The solution is to use a *heavy-light decomposition* of the tree of routing paths rooted at v. A heavy-light decomposition partitions the edges of a tree into "light" edges and "heavy" paths, where any root-to-leaf path has $O(\log n)$ light edges. This is useful because we can independently shortcut each of the (vertex-disjoint) heavy paths down to diameter and sensitivity $O(\log n)$ using standard methods (see e.g. Figure 1). This results in a shortcut set with vertex-sensitivity $O(\log n)$ and diameter $O(\log^2 n)$. See Section 3.3.2 for more details.

This solution suggests a more general technique for constructing low-sensitivity hopsets: whenever a traditional hopset contains a star, simply replace it by the above heavy-light decomposition shortcutting scheme. This technique works, for instance, to translate the folklore exact hopset into the low-sensitivity setting. The folklore hopset with $\widetilde{O}(n)$ edges and $\widetilde{O}(\sqrt{n})$ hop-diameter is the following: randomly sample a set S of $\widetilde{O}(\sqrt{n})$ vertices and add a hopset edge between all pairs of reachable vertices in S (where the edge weight is the distance between its endpoints). Viewing each vertex in S as the center of a star, we can apply the heavy-light decomposition transformation to achieve both vertex-sensitivity and hop-diameter $\tilde{O}(\sqrt{n})$ [CGK+23, DGUW23]. This remains the best-known result for exact and approximate low-sensitivity directed hopsets.

We remark that it would also be natural to try to adapt Kogan and Parter's recent $\widetilde{O}(n^{1/3})$-diameter directed shortcut set [KP22c] on $\widetilde{O}(n)$ edges, to the low-sensitivity setting. In short, it is not clear how to do this. The first step of their construction picks $\widetilde{O}(n^{2/3})$ chains on $\widetilde{O}(n^{1/3})$ vertices each, and shortcuts each of them. Shortcutting a single chain could add sensitivity to $\Omega(n)$ vertices and edges (since it is a chain, not a path). So, shortcutting $\widetilde{O}(n^{2/3})$ chains could already incur vertex and edge sensitivity $\widetilde{O}(n^{2/3})$.

Our Main Technical Contribution. Although the known constructions for low-sensitivity hopsets use the strategy of starting with traditional hopsets and applying the above heavy-light decomposition transformation, it is not clear that this is the optimal strategy. In particular, the low-sensitivity setting permits us freedom not allowed in the traditional setting: we can add as many edges as we'd like, as long as the sensitivity of each individual vertex is bounded. Thus, we would like to take advantage of our ability to add many edges, while ensuring that not too many of our added edges have overlapping underlying routing paths. In light of this new goal, we examine low-sensitivity hopsets from scratch, developing a new technique that is conceptually very different from any known techniques for the traditional setting.

Our main new technique allows us to improve above exact undirected hopset of prior work [CGK+23, DGUW23] from vertex sensitivity $\widetilde{O}(\sqrt{n})$ all the way down to $O(\log n)$. Our resulting hopset with $O(\log n)$ vertex-sensitivity and $\widetilde{O}(\sqrt{n})$ hop-diameter is tight up to polylogarithmic factors in both parameters.

This technique heavily relies on the assumption that the routing paths are chosen to be *consistent*; that is, each pair of overlapping paths overlaps at one single contiguous subpath. This assumption holds for shortest paths in undirected graphs and DAGs (with a consistent tie-breaking scheme), but not for shortest paths in general directed graphs. This is why our technique does not apply to exact or approximate directed hopsets. But it does apply to directed shortcut sets when we use the technique on the DAG of SCCs, in combination with other ingredients such as carefully choosing routing paths.

Now, we outline the technique. It is a greedy algorithm for processing routing paths in a carefully chosen order. Each time we process a routing path, we shortcut the path using a simple and standard method (see Figure 1) which incurs $O(\log n)$ sensitivity. We need to be careful because if we shortcut a path that overlaps with a previously shortcutted path, the vertices in the overlap incur double the sensitivity. For this reason, when we process a path we only shortcut the segments of a path that have not been previously shortcutted. We think of it this way: every time a path is shortcutted, it cast "shadows" on other overlapping paths, and we only shortcut the non-shadowed segments of each path.

Using this method, it is immediate from the standard method for shortcutting a path, that the vertex-sensitivity is only $O(\log n)$. However, the hop-diameter is in question. The hop-diameter can be determined by roughly the number of shadows cast onto a path, since each shadow chops the path into more segments, and so the goal is to bound the number of shadows cast onto each path. Since the routing paths are consistent, we know that when we process a path it only casts at most one new shadowed segment (which may be further segmented by already existing shadows) onto every other path. This helps, but we still need to choose the order to process the paths carefully. Even with the consistency property, processing the paths in the wrong order could lead to $\Omega(n)$ vertex-sensitivity.

To choose the processing order, we carefully define a potential function and choose the path with maximum potential. The potential of a path P is rather simple to define and fast to implement: the number of vertices on P that are not on any previously chosen path. In our analysis, we show that this potential function yields an algorithm with the desired hop-diameter of $\tilde{O}(\sqrt{n})$.

Techniques for Lower Bounds. For our lower bounds, we use modifications of constructions that have previously been applied to traditional hopsets as well as spanners, emulators, distance preservers, reachability preservers, and related problems. These graph constructions are layered graphs defined by a collection of *perfect paths*, which are unique paths (or unique shortest paths, depending on the setting) that go from the first to last layer and are pairwise edge-disjoint. These graphs are useful for traditional hopsets because their properties imply that the addition of a single edge can only decrease the hop-length of one perfect path. This condition is useful for low-sensitivity hopsets too, but the situation is more nuanced. When we add a single edge to a traditional hopset it simply counts 1 towards our budget of edges, whereas in the low-sensitivity setting the total amount of sensitivity added depends on the "length" of the added edge. Thus, we need to take into account all possible edge lengths, and consider their contributions to both sensitivity and diameter.

As a separate issue, we are interested in edge sensitivity for lower bounds. For this reason we modify known perfect path constructions by replacing each vertex by an edge, which requires a slightly more careful analysis.

Our final construction begins with a known perfect path construction and optimizes the parameters for the low-sensitivity hopset setting (which results in different graph parameters than constructions for traditional hopsets). We obtain a lower bound showing that to achieve polylogarithmic vertex or edge-sensitivity for exact undirected hopsets, the hop-diameter must be $\widetilde{\Omega}(\sqrt{n})$, which is tight with our aforementioned upper bound. We also obtain lower bounds for low-sensitivity directed shortcut sets and low-sensitivity approximate undirected hopsets, both by again appealing to known constructions of layered graphs.

Remark 2.1. Bodwin and Hoppenworth [BH23] are able to prove stronger lower bounds on the number of edges in a hopset by relaxing the requirement that perfect paths are disjoint. Unfortunately, their relaxation does not immediately seem to be useful for lower bounding the *sensitivity* of a hopset; see Appendix A for more details.

Techniques for Approximate Undirected Hopsets. For traditional hopsets with $O(n)$ size, there are essentially 3 diameter regimes: (1) directed shortcut/hopsets as well as undirected exact hopsets all require polynomial hop-diameter, (2) *approximate* hopsets for undirected graphs require only $n^{o(1)}$ hop-diameter, and (3) undirected shortcut sets trivially achieve diameter 2. In the low-sensitivity setting, we have mainly discussed the polynomial diameter regime so far, but it also makes sense to consider the $n^{o(1)}$-hop-diameter regime for low-sensitivity approximate undirected hopsets. To this end, we extend known results for traditional approximate undirected hopsets to the low-sensitivity setting, achieving both vertex-sensitivity and hop-diameter $n^{o(1)}$ for any $\varepsilon > 0$ that is at least inverse polylogarithmic.

In the traditional setting, Huang-Pettie and Elkin-Neiman [HP19, EN19] showed that Thorup-Zwick emulators [TZ06] are optimal hopsets. Examining this construction quickly reveals that it has high vertex and edge sensitivity. Instead, we begin with the similar construction of Thorup-Zwick spanners (also in [TZ06]). Recall that emulators and spanners are both sparse graphs that approximately preserve distances in graphs, but a spanner is a subgraph while an emulator can have added edges. Thus, it makes sense to consider an emulator as a hopset, but not a spanner (since it has no added edges). However, we can use the heavy-light tree decomposition shortcutting on top of the Thorup-Zwick spanner to obtain a construction that is similar to the Thorup-Zwick emulator, but now it has low sensitivity. Then, we can leverage the hopset analysis tools of Huang-Pettie to show that our $n^{o(1)}$ vertex sensitivity hopset also has hop-diameter $n^{o(1)}$ for any $\varepsilon > 0$ that is at least inverse polylogarithmic.

3 Preliminaries

3.1 Notations and Definitions

With the aim of formally defining vertex and edge sensitivity, we first introduce a couple of notions.

Definition 3.1 (*Routing Paths*)**.** Given a graph $G = (V, E)$ we call $\mathcal{P}$ a set of *routing paths* for G if for all $s, t \in V$ such that s can reach t, there is exactly one path $P \in \mathcal{P}$ such that P is a path in G with starting point s and ending point t. If G is weighted, we stipulate that the paths in $\mathcal{P}$ must be shortest paths in G.

Routing paths are meant to model paths that are deemed critical. In the context of hopsets, the routing paths must thus be shortest paths, whereas in the context of shortcut sets, any path may be chosen as a routing path.

Definition 3.2 (*Span*)**.** Given a graph $G = (V, E)$ and a tuple of a set of routing paths and a shortcut/hopset $\mathcal{A}(G) = (\mathcal{P}, H)$, we define the *span* of an edge $e = (s, t) \in H$ to be the unique path in $\mathcal{P}$ with starting point s and ending point t; we denote this with $\mathsf{span}_{\mathcal{A}(G)}(e)$. When the context is clear, we omit the subscript and write $\mathsf{span}(e)$.

We are now ready to define vertex and edge sensitivity. These are properties of (the inputs) graphs, together with (the outputs) a set of routing paths and a shortcut/hopset.

Definition 3.3 (*Vertex Sensitivity*)**.** Given a graph $G = (V, E)$ and a tuple of a set of routing paths and a shortcut/hopset $\mathcal{A}(G) = (\mathcal{P}, H)$, the *vertex sensitivity* vector, indexed by V, is denoted with $\widehat{\mathsf{v}}_{G,\mathcal{A}(G)}$. When the context is clear, we omit any subset of the subscripts: $\widehat{\mathsf{v}}$.

The value of $\widehat{\mathsf{v}}$ at index v is

$$\widehat{\mathsf{v}}(v) = |\{f \in H : v \in V(\mathsf{span}(f))\}|.$$

The worst case *vertex sensitivity* (given $G, \mathcal{A}(G)$) is

$$\|\widehat{\mathsf{v}}\|_\infty = \max_{v \in V} \widehat{\mathsf{v}}(v).$$

Definition 3.4 (*Edge Sensitivity*)**.** Given a graph $G = (V, E)$ and a tuple of a set of routing paths and a shortcut/hopset $\mathcal{A}(G) = (\mathcal{P}, H)$, the *edge sensitivity* vector, indexed by E, is denoted with $\widehat{\mathsf{e}}_{G,\mathcal{A}(G)}$. When the context is clear, we omit any subset of the subscripts: $\widehat{\mathsf{e}}$.

The value of $\widehat{\mathsf{e}}$ at index e is

$$\widehat{\mathsf{e}}(e) = |\{f \in H : e \in E(\mathsf{span}(f))\}|.$$

The worst case *edge sensitivity* (given $G, \mathcal{A}(G)$) is

$$\|\widehat{\mathsf{e}}\|_\infty = \max_{e \in E} \widehat{\mathsf{e}}(e).$$

A reason behind our choice of using vector notation to describe sensitivity is as follows: We sometimes consider the total sensitivity $\|\widehat{\mathsf{e}}\|_1 = \sum_e \widehat{\mathsf{e}}(e)$ and even the sum of sensitivities over a set $T \subseteq E$ which we write with $\mathbb{1}_T \cdot \widehat{\mathsf{e}}$ where $\mathbb{1}_T$ is the indicator vector of T and '$\cdot$' denotes the usual inner product over vectors.

Next, we define the notion of *consistency*, which will prove to be a very useful property of carefully chosen routing paths in our upper bound constructions.

Definition 3.5 (*Consistency* (Modification from [CE06]))**.** A pair of paths P, P' are said to be *consistent* if their intersection contains at most one connected component. A set of paths $\mathcal{P}$ is said to be *consistent* if every pair of paths $P, P' \in \mathcal{P}$ are consistent.

A shortest path tiebreaking function chooses for each pair of vertices $s, t \in V$, where s can reach t, exactly one shortest path from s to t. We say a shortest path tiebreaking function is *consistent* if its image is consistent.

Our lower bounds go through via constructions of graphs inspired by [Bod21, ABP18]; in fact we directly use Theorems 5 and 6 from the first paper and Theorem 4.6 from the second paper for our constructions, but our proofs and the way we invoke the theorems differ. A common object that appears in both papers, and that we use, are layered graphs.

Definition 3.6 (*Layered Graphs*)**.** A graph $G = (V, E)$ is *layered* if the vertex set can be partitioned into $V = V_1 \sqcup V_2 \sqcup \ldots \sqcup V_\ell$ such that every edge goes between two consecutive parts V_i and V_{i+1}. If G is directed, then edges are oriented towards the parts with larger indices.

3.2 Relationship Between Vertex and Edge Sensitivity

We discuss how different notions of sensitivity relate to each other. First, observe that for a fixed shorctut/hopset H, its $\|\widehat{\mathsf{v}}\|_\infty$ and $\|\widehat{\mathsf{e}}\|_\infty$ could differ drastically. For example, consider an $n/2$-tipped star with tips of length 2; any H comprising of all edges from the center c to the ends of all the tips would have $\widehat{\mathsf{v}}(c) = n/2$ but $\widehat{\mathsf{e}}(e) = 1$ for all edges e. More generally, the shortcut/hopset edges with endpoint v contribute one each to $\widehat{\mathsf{v}}(v)$ whereas their contribution to the edge sensitivity can be dispersed over the routing paths incident to v. At any rate, observe that the edge sensitivity being no more than the vertex sensitivity generalizes to any graph.

Observation 3.7. $\|\widehat{\mathsf{e}}_{G,\mathcal{A}(G)}\|_\infty \leq \|\widehat{\mathsf{v}}_{G,\mathcal{A}(G)}\|_\infty$.

Proof. Consider the edge $e = (u, v)$ where $\widehat{\mathsf{e}}(e) = \|\widehat{\mathsf{e}}\|_\infty$. Observe in particular that $\widehat{\mathsf{e}}(e) \leq \widehat{\mathsf{v}}(u)$ since every edge in $\mathcal{A}(G)$ contributing to $\widehat{\mathsf{e}}(e)$ also contributes to $\widehat{\mathsf{v}}(u)$. We therefore conclude that $\|\widehat{\mathsf{e}}_{G,\mathcal{A}(G)}\|_\infty \leq \|\widehat{\mathsf{v}}_{G,\mathcal{A}(G)}\|_\infty$. ∎

It follows from Observation 3.7 that proving upper bounds for $\|\widehat{\mathsf{v}}\|_\infty$ gets us the same bound for $\|\widehat{\mathsf{e}}\|_\infty$ and, in the contrapositive, proving lower bounds for $\|\widehat{\mathsf{e}}\|_\infty$ gets us the same bound for $\|\widehat{\mathsf{v}}\|_\infty$. We thus strive to prove upper bounds on $\|\widehat{\mathsf{v}}\|_\infty$ and lower bounds on $\|\widehat{\mathsf{e}}\|_\infty$ throughout the paper; all results presented here are of this form.

3.3 Algorithmic Building Blocks

In this section we describe primitive subroutines used in our constructions of shortcut/hopsets. The first tool addresses how to "shorten" paths; the second, which can be viewed as a generalization of the first, addresses how to shorten trees.

3.3.1 Path Shortcutting

Path-Hopset is a well-known primitive which, when given a path, reduces hop-lengths of shortest paths to $O(\log n)$ while adding no more than $O(\log n)$ sensitivity to any vertex. Intuitively, Path-Hopset adds an edge between the two endpoints of the path, then partitions the path into two equally sized subpaths and recurses into each of them. An example of the output of Path-Hopset is depicted in Figure 1.

Path-Hopset
Input: A path P from v_0 to v_ℓ

```
1. If $\ell < 2$ then return $\{\}$
2. Return $\{(v_0, v_\ell)\} \cup$ Path-Hopset$(P[v_0..v_{\lfloor \ell/2 \rfloor}]) \cup$ Path-Hopset$(P[v_{\lfloor \ell/2 \rfloor}..v_\ell])$
```

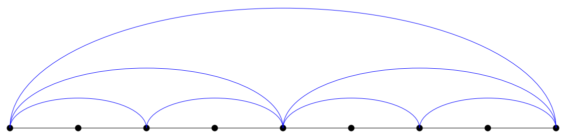

Figure 1: An example of Path-Hopset on a path of length 8 with vertices $v_0, \ldots, v_8$. The collection of blue edges is the shortcut/hopset output.

Observation 3.8. *Given a path $P = v_0, v_1, \ldots, v_n$ (possibly oriented in the (v_i, v_{i+1}) direction), Path-Hopset outputs a $O(\log n)$-shortcut/hopset with $\|\widehat{\mathsf{v}}\|_\infty = O(\log n)$.*

For completeness, we provide a proof in Appendix B.

3.3.2 Tree Shortcutting via the Heavy-light Decomposition

Tree-Hopset, described in [FL22], reduces the problem of shortcutting trees to that of shortcutting paths, where the instances passed to Path-Hopset are found via a heavy-light decomposition of the tree. A heavy-light decomposition [HT84] (also termed heavy path decomposition) of a rooted tree is a partitioning of its edges into 'heavy' paths and 'light' edges such that any root-to-leaf path passes through at most $O(\log n)$ light edges, which is captured by Proposition 3.9.

Proposition 3.9 ([HT84]). *Given a tree T of size n, there is a heavy-light decomposition such that*

- *any root-to-leaf path has at most $O(\log n)$ light edges;*
- *all heavy paths are vertex disjoint.*

Using the above proposition, Tree-Hopset is specified as follows.

Tree-Hopset
Input: A tree T rooted at v

1. Return Path-Hopset(P) for all heavy paths P in a heavy-light decomposition of T

Observation 3.10. *Given a tree T rooted at v (with possibly all edges oriented away from the root or all edges oriented towards the root), Tree-Hopset outputs an $O(\log^2 n)$-shortcut/hopset with $\|\widehat{\mathrm{v}}\|_\infty = O(\log n)$.*

This result is implicitly shown in the analysis of [FL22], but we provide a proof (in a differential-privacy-free language) in Appendix B for completeness. Elementary use of Tree-Hopset yields the following quick results. A $O(\log^2 n)$-shortcut set construction with $\|\widehat{\mathrm{v}}\|_\infty = O(\log n)$ on undirected graphs. A directed $(O(\sqrt{n}\log n), 0)$-hopset construction with $\|\widehat{\mathrm{v}}\|_\infty = O(\sqrt{n}\log n)$. See Appendices C and D for details.

4 Upper Bounds via a New Greedy Approach

This section describes a new way to construct low-sensitivity shortcut/hopsets when we are able to choose consistent routing paths; this is always possible, for example, in undirected graphs and DAGs. Using this approach, we can quite immediately get Theorem 1. With some slight care, we apply this approach to shortcut sets on directed graphs and obtain Theorem 3.

4.1 The Greedy Construction

At a schematic-level, the construction is very simple to describe: for each routing path P, processed in *some order*, apply Path-Hopset to the *uncovered pieces* of P. The details are filled in as follows.

Finding an Order to Process Paths. The order for which we process the shortest paths is described via a potential function Φ on routing paths P, which counts the number of vertices in P that have not yet been contained in a path that has thus far been processed.

Definition 4.1. Let $\mathcal{P}$ be the set of all routing paths, and let $\mathcal{P}' \subseteq \mathcal{P}$ be a subset of the set of all routing paths. We define, for all $P \in \mathcal{P}$, the potential

$$\Phi_{\mathcal{P}'}(P) = \left|\left\{v \in V(P) : v \notin V(P') \text{ for all paths } P' \in \mathcal{P}'\right\}\right|.$$

If it is clear from the context, we omit the subscript and write $\Phi(P)$.

The i^{th} routing path in the order is then the one which maximizes $\Phi_{\mathcal{P}^{(i-1)}}$, where $\mathcal{P}^{(i-1)}$ is the set of the first $i-1$ routing paths in the order (that is, the ones that have already been processed).

The following is an observation that we will make use of later; it says that the potential of a path P is weakly decreasing throughout the construction process, which can be seen by noting that if P is processed then it has 0 potential and, otherwise, its vertices that are not contained in already processed paths can only drop as more and more paths are processed.

Observation 4.2. *$\Phi_{\mathcal{P}(i)}$ is weakly decreasing in i.*

Processing a Path. The uncovered pieces of an unprocessed path P depend on the the history of the algorithm up to the point where P is processed. These are the maximal subpaths of P for which no vertex is part of an already processed path.

Definition 4.3 (*Uncovered pieces of a shortest path*)**.** Let $\mathcal{P}'$ be the set of paths processed over so far. The *uncovered pieces* of P, given $\mathcal{P}'$, are then the maximal connected components of $\{v \in V(P) : v \notin V(P') \text{ for all paths } P' \in \mathcal{P}'\}$.

A path P is processed by running Path-Hopset on each of its uncovered pieces.

Putting Things Together. The algorithm producing an $\widetilde{O}(\sqrt{n}, 0)$-hopset with $\|\widehat{\mathrm{v}}\|_\infty = O(\log n)$ is then described by Greedy-Hopset which selects the path that maximizes the potential Φ and processes it as above, repeating until every path is processed.

```
Greedy-Hopset
Input: A graph $G = (V, E, w)$ with a consistent set of routing paths $\mathcal{P}$.
  1. $\mathcal{P}' \leftarrow \emptyset, H \leftarrow \emptyset$
  2. While $|\mathcal{P}'| < \binom{n}{2}$
     (a) Select the next routing path $P^* \leftarrow \arg\max_{P \in \mathcal{P} \setminus \mathcal{P}'} \Phi_{\mathcal{P}'}(P)$, breaking ties arbitrarily
     (b) $H \leftarrow H \cup$ Path-Hopset$(p)$ for all uncovered pieces $p$ of $P^*$
     (c) $\mathcal{P}' \leftarrow \mathcal{P}' \cup \{P^*\}$
  3. Return $H$
```

One can quickly observe that Greedy-Hopset produces a low-sensitivity shortcut/hopset.

Observation 4.4. *The hopset produced by Greedy-Hopset has $\|\widehat{\mathrm{v}}\|_\infty = O(\log n)$.*

Proof. Let v be an arbitrary vertex. Its sensitivity is only ever increased in one iteration of Greedy-Hopset, namely the first time a path P is chosen such that $v \in V(P)$. We know from Observation 3.8 that the contribution to $\widehat{\mathrm{v}}(v)$ in one iteration is bounded by $O(\log n)$, completing the proof. ∎

Before we turn to the remaining claim that Greedy-Hopset is an $\widetilde{O}(\sqrt{n}, 0)$-hopset, we need a couple of definitions. Henceforth, let the paths be processed in the order $P_1, P_2, \ldots, P_{O(n^2)}$. First we define the *shadows* and *penumbras* cast onto P_i. Shadows are, loosely speaking, complementary to the uncovered pieces of P_i when it is processed; they serve two main purposes: (i) to bound the number of uncovered pieces when P_i is processed, and (ii) as shortcuts that can be used to traverse the covered pieces of P_i.

Definition 4.5 (*Shadows, Penumbras*)**.** For any $j < i$, the *shadows* cast by P_j onto P_i are the segments of P_j that coincide with P_i after removing all shadows cast before P_j onto P_i. Formally, they are the maximal subpaths of

$$P_i^{(j)} = \{v \in P_i \mid v \in P_j \text{ and } v \notin P_k \text{ for } k < j\},$$

defined for all $j < i$.

Let S be a shadow cast onto P_i. We call the edges in $E(S, P_i \setminus S)$ the *penumbras* of S cast onto P_i, where $E(S, P_i \setminus S)$ denotes the edges with one endpoint in S and the other in $P_i \setminus S$.

Observe that a single path P_j may cast $\Omega(j)$ shadows onto P_i. Nevertheless, if P_i has many shadows cast onto it, there must be almost as many paths which cast shadows onto P_i.

Observation 4.6. *Suppose after iterating P_1 through to P_t, there are k shadows cast onto P_i for $i > t$. Then, at least $(k-1)/2$ distinct paths among P_1 through to P_t cast at least one shadow onto P_i.*

Proof. First, note that since there are k shadows cast onto P_i, there are at least $k-1$ edges in P_i which are penumbras. While the path P_j may cast many penumbras onto P_i for $j < i$, by the consistency of routing paths P_j casts at most 2 penumbra onto P_i that have not already been cast before P_j; these edges are namely the ones incident to boundary of the intersection between P_j and P_i (see Figure 2). We conclude that there are at least $(k-1)/2$ distinct paths among P_1 through to P_t that cast at least one shadow onto P_i. ∎

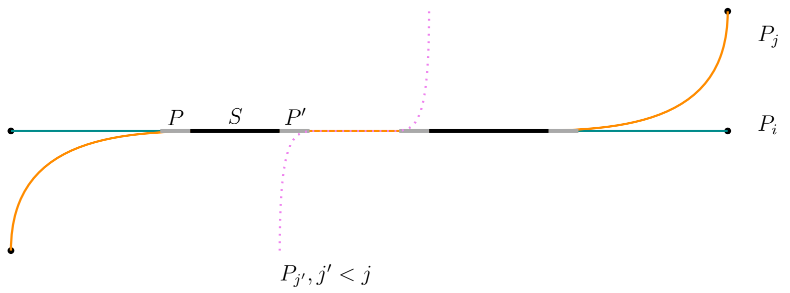

Figure 2: The penumbra P of S (a shadow cast by P_j onto P_i) is new, but the penumbra P' of S is also the penumbra of a shadow cast by $P_{j'}$ and is thus not new.

Suppose a routing path P from s to t has no more than k shadows cast onto it. Then the $G \cup$ Greedy-Hopset(G) hop-length from s to t is at most $\widetilde{O}(k)$; each shadow has shortcut-distance at most $O(\log n)$ by an application of Observation 3.8 to the time the shadow was first cast, and the remaining parts of P, of which there are $O(k)$ of them, each have shortcut-distance at most $O(\log n)$ by another application of Observation 3.8 at the time P is processed. Motivated by this, we now show that every routing path has at most $O(\sqrt{n})$ shadows cast onto it and every shadow on P has hop-length $O(\log n)$.

Lemma 4.7. *P_i has $O(\sqrt{n})$ shadows cast onto it, for all i.*

Proof. Suppose, for a contradiction, that there is a routing path P_i which has at least $4\sqrt{n}$ shadows cast onto it. We aim to show that there are many "highly disjoint" paths that cast at least one shadow onto P_i, which will lead to a contradiction since too many such paths imply G has more than n vertices.

Towards this, let P_t for $t < i$ be the first path processed after which $\Phi(P_i) < \sqrt{n}$. Such a t must exist since, otherwise, we can invoke the monotonicity of Φ, which is weakly decreasing (see Observation 4.2), and Observation 4.6, which says that there must be at least $(4\sqrt{n}-1)/2$ distinct paths casting shadows onto P_i, to show that G has $\sqrt{n} \cdot (4\sqrt{n}-1)/2 > n$ vertices which is a contradiction.

After P_t is processed, there can be at most $\sqrt{n}$ more shadows cast onto P_i since $\Phi(P_i) < \sqrt{n}$ and each shadow cast onto P_i reduces $\Phi(P_i)$ by at least 1. There are therefore at least $3\sqrt{n}$ shadows cast onto P_i by paths processed before P_t (inclusive). Then, we can repeat the same argument as before: by Observation 4.2 and Observation 4.6 G has $\sqrt{n} \cdot (3\sqrt{n} - 1)/2 > n$ vertices, leading to a contradiction. ∎

Observation 4.8. *Let P be a routing path, and let S be any shadow with endpoints s, t cast onto P. Then the $G \cup$ GREEDY-HOPSET(G) hop-length from s to t (using edges contained in S or whose span is contained in S)*[6] *is bounded by $O(\log n)$.*

Proof. Let Q be the routing path casting S onto P. The claim goes through since at the time Q is processed, S is contained in an uncovered piece of Q; GREEDY-HOPSET makes a call to PATH-HOPSET on the uncovered piece of Q containing S when Q is being processed, finishing things up by Observation 3.8. ∎

4.2 An Upper Bound for Undirected Exact Hopsets

We are able to get upper bounds for exact hopsets on undirected graphs and DAGs immediately using GREEDY-HOPSET.

Theorem 1 (Undirected Exact Hopset Upper Bound)**.** *There exists an algorithm producing an $(O(\sqrt{n}\log n), 0)$-hopset with $\|\widehat{\mathrm{v}}\|_\infty = O(\log n)$ over undirected and directed acyclic graphs.*

Proof. We show the theorem for GREEDY-HOPSET. This follows from combining Lemma 4.7 and Observation 3.8, and also Observation 4.8. To spell things out, let G be an undirected or directed acyclic graph. Any routing path in G has $O(\sqrt{n})$ shadows, each shadow can be crossed within $O(\log n)$ hops (in the correct direction since G is undirected or a DAG), and the uncovered pieces of P at the time it is being processed (of which there are $O(\sqrt{n})$ many) can be crossed within $O(\log n)$ hops since GREEDY-HOPSET makes a call to PATH-HOPSET on each of these pieces.

The sensitivity claim follows directly from Observation 4.4. ∎

4.3 An Upper Bound for Directed Shortcut Sets

Recall that shortcut sets pertain to the problem of reachability. The matter of finding low-sensitivity directed shortcut sets can be resolved by a more careful application of GREEDY-HOPSET. The crux here is that now we have to be slightly careful about our choice of routing paths in order to apply GREEDY-HOPSET effectively; directed graphs may unavoidably contain inconsistent paths (see Figure 3).

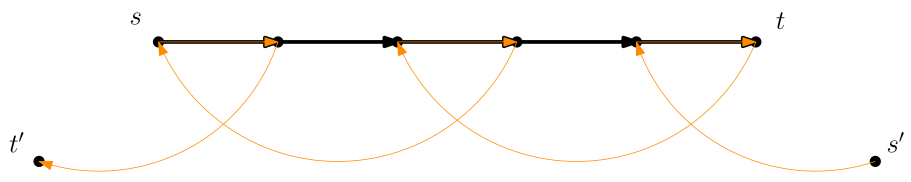

Figure 3: No choice of an st-path and an $s't'$-path yields a consistent set of routing paths.

[6]This is an important constraint, since we wish for this claim to apply to $(\beta, 0)$-hopsets

Routing Paths. Fix a directed graph $G = (V, E)$ and consider its strongly connected components (SCCs) $C^1, C^2, \dots, C^k$. Our shortcut set construction will contain two types of edges: (i) intra-SCC edges, and (ii) inter-SCC edges.

To handle the intra-SCC edges of C^i, we arbitrarily pick exactly one representative vertex $v^i \in C^i$ and use any shortest path tiebreaking function to find a shortest path arborescence of $G[C^i]$, the graph induced by C^i, towards v^i and another away from v^i. We denote these arborescences by $T^{in}_{v^i}$ and $T^{out}_{v^i}$ respectively. All paths in $T^{in}_{v^i}$ from descendent towards ancestor and in $T^{out}_{v^i}$ from ancestor towards descendent are included in the set of routing paths. Notice that the routing paths are well defined. For a pair of vertices $s, t \in C^i$ it might seem that we have mistakenly defined two distinct routing paths (one from $T^{in}_{v^i}$ and the other from $T^{out}_{v^i}$); this is, however, not possible since we have used only one shortest path tiebreaking function from which there is a unique shortest st-path.

To handle the inter-SCC edges, for any C^i and C^j with $E(C^i, C^j) \neq \emptyset$, we arbitrarily pick exactly one representative edge $e^{i,j} = (u, v) \in E(C^i, C^j)$ and call u an out-port and v an in-port, which we access with $\mathsf{out}(e^{i,j})$ and $\mathsf{in}(e^{i,j})$ respectively. The shortcut set will only ever add inter-SCC edges from out-ports to in-ports and so only routing paths for these types of edges are left to be defined (the other routing paths can be defined arbitrarily). Let D be the directed acyclic graph (DAG) formed by contracting the SCCs of G, and removing for all i, j parallel edges between the super-vertices C^i and C^j except for the representative edge $e^{i,j}$, if it exists. Let $\mathcal{P}_D$ be the set of shortest paths of D over a *consistent* shortest path tiebreaking function (see Definition 3.5). Then for any $P \in \mathcal{P}_D$, where $P = e_1, e_2, \dots, e_\ell$, we define the routing path in G from $\mathsf{out}(e_1)$ to $\mathsf{in}(e_\ell)$ by connecting, for all $i \in [\ell - 1]$, $\mathsf{out}(e_i)$ to $\mathsf{in}(e_{i+1})$ via an arbitrary simple path contained in the SCC they both belong to (see Figure 4).

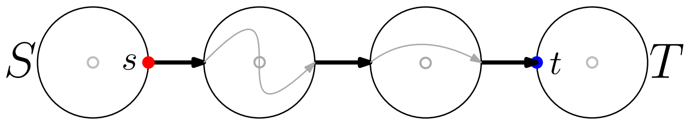

Figure 4: The ST-path in D (super-vertices, which are SCCs, are represented by large circles) yields an st-path in G where s is an out-port and t is an in-port. Gray paths contained in SCCs are chosen arbitrarily.

The Construction. Having defined the relevant routing paths to our shortcut set, the complete construction is as follows.

Greedy-Di-Shortcut-Set
Input: A directed graph $G = (V, E)$ with a representative vertex v^i for each strongly connected component C^i and a representative edge for each pair of adjacent SCCs.

```
1. $H \leftarrow \emptyset$
2. $H \leftarrow H \cup \textsc{Tree-Hopset}(T^{in}_{v^i}) \cup \textsc{Tree-Hopset}(T^{out}_{v^i})$ for every strongly connected component $C^i$ where $T^*_{v^i}$ is a shortest path arborescence of $G[C^i]$ rooted at $v^i$
3. $D \leftarrow$ contraction of SCCs of $G$, only keeping representative edges
4. For every $e \in \textsc{Greedy-Hopset}(D)$ update $H \leftarrow H \cup \{(\mathsf{out}(e_1), \mathsf{in}(e_\ell))\}$ where $e_1, e_\ell$ are defined by $\mathsf{span}_D(e) = e_1, e_2, \dots, e_\ell$
5. Return $H$
```

It remains to bound the diameter and sensitivity of GREEDY-DI-SHORTCUT-SET.

Theorem 3 (Directed Shortcut Set Upper Bound)**.** *There exists an algorithm producing an $O(\sqrt{n}\log^3 n)$-shortcut set with $\|\widehat{\mathrm{v}}\|_\infty = O(\log n)$ for directed graphs.*

Proof.
We show the theorem for GREEDY-DI-SHORTCUT-SET.
Output is a $O(\sqrt{n}\log^3 n)$-shortcut set. Let G be a directed graph and suppose the vertex s reaches the vertex t in G. Denote the SCC of s by S and that of t by T. There must be a shortest path from S to T in the directed acyclic graph D. Thus, $D \cup$ GREEDY-HOPSET(D) produces an $O(\sqrt{n}\log n)$-shortcut set for D, by Theorem 1. This translates to a path in $G \cup$ GREEDY-DI-SHORTCUT-SET(G) that uses $O(\sqrt{n}\log n)$ edges to connect S to T (not counting the edges required the move from in-port to out-port within each SCC). Moving from an in-port to an out-port of an SCC C takes $O(\log^2 n)$ by the call to TREE-HOPSET(C), and using Observation 3.10. The $G \cup$ GREEDY-DI-SHORTCUT-SET(G) hop-length from s to t is thus $O(\sqrt{n}\log n) \cdot O(\log^2 n) = O(\sqrt{n}\log^3 n)$.
Sensitivity is bounded by $O(\log n)$. Consider any vertex v. The contributions to $\widehat{\mathrm{v}}(v)$ come from the call to TREE-HOPSET(C) where $v \in C$ (the intra-SCC edges) and from the shortcut set edges added by the call to GREEDY-HOPSET(D) (the inter-SCC edges). The contribution of TREE-HOPSET(C) is $O(\log n)$ by Observation 3.10. The contribution by the latter is also $O(\log n)$ since any inter-SCC edge which contributes to $\widehat{\mathrm{v}}(v)$ has a distinct corresponding edge which contributes to $\widehat{\mathrm{v}}_D(C)$ in the call to GREEDY-HOPSET(D), and we know that $\|\widehat{\mathrm{v}}_D\|_\infty = O(\log n)$ by Observation 4.4. ∎

5 Approximate Undirected Low-Sensitivity Hopset Constructions

Here we give an $(n^{o(1)}, \varepsilon)$-hopset with vertex sensitivity $n^{o(1)}$ for any $\varepsilon > 0$ that is at least inverse polylogarithmic. As a reminder, the main theorem of this section is as follows.

Theorem 5. *There exists an algorithm which produces a $\big(O\big((k/\varepsilon)^k \log^2 n\big), \varepsilon\big)$-hopset H with $\|\widehat{\mathrm{v}}\|_\infty = O(kn^{1/k}\log^2 n)$ for undirected graphs. For any $\varepsilon > 0$ that is at least inverse polylogarithmic, setting $k = \Theta(\sqrt{\log n})$ gives a $(n^{o(1)}, \varepsilon)$-hopset with $\|\widehat{\mathrm{v}}\|_\infty = n^{o(1)}$.*

Intuition. Huang-Pettie and Elkin-Neiman showed in [HP19, EN19] that the emulators constructed by Thorup and Zwick in [TZ06] are $\big(O\big((k/\varepsilon)^k\big), \varepsilon\big)$-hopsets for every $\varepsilon > 0$. Briefly, the emulators are constructed by placing vertices in a hierarchy of roughly k levels and, for each vertex v, connecting v with a direct edge to all vertices in Ball$[v]$, where Ball$[v]$ are the vertices in v's level that are closer to v than v is to the level one higher than v's (formal details to come later).

Ideally, setting $k = \Theta(\sqrt{\log n})$, the emulator would immediately give us an $(n^{o(1)}, \varepsilon)$-hopset. But things are hardly ever ideal; for one, vertices in the last level of the hierarchy are connected directly to all of V and thus would witness high sensitivity. To circumvent this, we construct a low-sensitivity hopset heavily based on (if not entirely lifted from) [TZ05], another result of Thorup and Zwick pertaining to distance oracles (and, later, also to spanners in [TZ06]). We then use the emulator of Thorup and Zwick as an analysis tool: for any pair (u, v) in the emulator, we show that there is a shortest path using $O(\log^2 n)$ hops to get from u to v in our hopset.

The Thorup-Zwick Spanner Based Construction. For the remainder of this section, we work with an undirected weighted graph $G = (V, E, w)$. We also set the routing paths under a consistent shortest path tiebreaking function (see Definition 3.5) and refer to them as *the* shortest paths. To proceed, we first make a definition (to the extent possible, all definitions in this section coincide with those in [TZ05, TZ06, HP19]).

Definition 5.1 (*Cluster*)**.** Let $V = A_0 \supseteq A_1 \supseteq \ldots \supseteq A_{k-1}$ be a hierarchy of vertices, and $A_k = \emptyset$. The *cluster* of $u \in A_i \setminus A_{i+1}$, defined for all $i \in [0, k-1]$, is

$$\text{Cluster}(u) = \{v \in V : \mathsf{dist}_w(v,u) < \mathsf{dist}_w(v, A_{i+1})\}.$$

Note that if $u \in A_{k-1}$, then $\text{Cluster}(u) = V$. We first give an observation about clusters in advance, to be used later in the analysis of our construction.

Observation 5.2. *For all $u \in V$, there is some truncation T_u^* of the shortest path tree T_u rooted at u such that $V(T_u^*) = \mathit{Cluster}(u) \cup \{u\}$.*

Proof. Let $u \in A_i \setminus A_{i+1}$ and $v \in \text{Cluster}(u)$. We establish this claim by showing that all vertices along the shortest path from u to v also belong to $\text{Cluster}(u)$. Let v' be any such vertex, and suppose for a contradiction that $v' \notin \text{Cluster}(u)$. Then

$$\begin{aligned}
& \mathsf{dist}_w(v', A_{i+1}) \leq \mathsf{dist}_w(v', u) && (v' \notin \text{Cluster}(u)) \\
\implies & \mathsf{dist}_w(v, v') + \mathsf{dist}_w(v', A_{i+1}) \leq \mathsf{dist}_w(v, v') + \mathsf{dist}_w(v', u) \\
\implies & \mathsf{dist}_w(v, v') + \mathsf{dist}_w(v', A_{i+1}) \leq \mathsf{dist}_w(v, u) && (\mathsf{dist}_w(v, v') + \mathsf{dist}_w(v', u) = \mathsf{dist}_w(v, u)) \\
\implies & \mathsf{dist}_w(v, A_{i+1}) \leq \mathsf{dist}_w(v, u) \\
& && (\text{Triangle inequality: } \mathsf{dist}_w(v, A_{i+1}) \leq \mathsf{dist}_w(v, v') + \mathsf{dist}_w(v', A_{i+1})) \\
\implies & v \notin \text{Cluster}(u),
\end{aligned}$$

which is a contradiction. ∎

We are now ready to present the low-sensitivity hopset, given by Apx-Undirected-Hopset which runs Tree-Hopset on the truncated shortest path trees rooted at $v \in V$ whose union over all vertices forms the classic Thorup-Zwick spanner.

Apx-Undirected-Hopset
Input: A weighted undirected graph $G = (V, E, w)$

```
1. Let $V = A_0 \supseteq A_1 \supseteq \ldots \supseteq A_{k-1}$ be a hierarchy of vertices chosen according to Theorem 3.7 in [TZ05] [a].
2. $H \leftarrow \emptyset$
3. For $u \in V$
   (a) Let $T_u$ be the shortest path tree rooted at $u$, truncated to $T_u^*$ so that $V(T_u^*) = \text{Cluster}(u) \cup \{u\}$
   (b) $H \leftarrow H \cup$ Tree-Hopset($T_u^*$ rooted at $u$)
4. Return $H$
```

[a] Theorem 3.7 in [TZ05] derandomizes the process where each vertex in A_i is independently promoted to A_{i+1} with probability $n^{-1/k}$, and $A_k = \emptyset$.

Let us first see that this construction indeed has low sensitivity, before we come back to the Thorup-Zwick emulator alluded to in the preamble of this section. Towards this, we define the inverse notion of Cluster(u).

Definition 5.3 (*Bunch*)**.** Let $V = A_0 \supseteq A_1 \supseteq \ldots \supseteq A_{k-1}$ be a hierarchy of vertices, and $A_k = \emptyset$. The i *level bunch of* $v \in V$ is defined as follows:

$$\mathrm{Bunch}_i(v) = \{u \in A_i : \mathsf{dist}_w(v, u) < \mathsf{dist}_w(v, A_{i+1})\}.$$

The *bunch of* v is

$$\mathrm{Bunch}(v) = \bigcup_i \mathrm{Bunch}_i(v).$$

Observe that $v \in$ Cluster(u) if and only if $u \in$ Bunch(v). We are now ready to show that Apx-Undirected-Hopset returns a low-sensitivity hopset.

Lemma 5.4. *Apx-Undirected-Hopset outputs a hopset H such that $\|\widehat{\mathrm{v}}\|_\infty = O(kn^{1/k} \log^2 n)$.*

Proof. Observe that $\widehat{\mathrm{v}}(v)$ for any $v \in V$ is only ever increased by Apx-Undirected-Hopset(G) in two cases:

- The call to Tree-Hopset(T^*_v rooted at v);
- Calls to Tree-Hopset(T^*_u rooted at u) when $v \in T^*_u$ which holds if and only if $u \in$ Bunch(v).

Each case contributes $O(\log n)$ to $\widehat{\mathrm{v}}(v)$, using Observation 3.10. Since the first case only occurs once, what remains is to address the second case by bounding $|\mathrm{Bunch}(v)|$.

Theorem 3.7 in in [TZ05] shows that the choice of $A_0, A_1, \ldots, A_k$ results in $|\mathrm{Bunch}(v)| = O(kn^{1/k} \log n)$ for all $v \in V$.

Putting everything together, $\|\widehat{\mathrm{v}}\|_\infty$ is bounded above by $\log n \cdot \big(1 + O(kn^{1/k} \log n)\big) = O(kn^{1/k} \log^2 n)$. ∎

Setting $k = \Theta(\sqrt{\log n})$ in Lemma 5.4 gives subpolynomial sensitivity: $\|\widehat{\mathrm{v}}\|_\infty = n^{o(1)}$. Now it remains to show that what Apx-Undirected-Hopset returns is indeed a good hopset.

Using the Thorup-Zwick as an Analysis Tool. We now describe the Thorup-Zwick Emulator of [TZ06, HP19], so that we may presently show that Apx-Undirected-Hopset is a good hopset. Let us first make a few more definitions.

Definition 5.5 ($p_i(v)$ and its tiebreaking)**.** Let $V = A_0 \supseteq A_1 \supseteq \ldots \supseteq A_{k-1}$ be a hierarchy of vertices, and $A_k = \emptyset$. For $i \in [0, k-1]$, $p_i(v) \in A_i$ is a vertex such that $\mathsf{dist}_w(v, p_i(v)) = \mathsf{dist}_w(v, A_i)$.

There could be many possible candidates for $p_i(v)$. Ties for $p_{k-1}(v)$ are broken arbitrarily. If $\mathsf{dist}_w(v, A_i) = \mathsf{dist}_w(v, A_{i+1})$, then we set $p_i(v) = p_{i+1}(v)$; otherwise, we break ties for $p_i(v)$ by setting it to an arbitrary candidate.

Finally, $p_k(v)$ is undefined, but we say that $\mathsf{dist}_w(v, p_k(v)) = \infty$.

While the tiebreaking is not necessary in the emulator construction in [TZ06], it will be useful for our purposes; what it buys us is that we are ensured $p_i(v) \in$ Bunch(v), shown in Lemma 4.1 of [TZ05] which uses the same tiebreaking scheme.

Definition 5.6 (Open and Closed Balls)**.** Let $V = A_0 \supseteq A_1 \supseteq \ldots \supseteq A_{k-1}$ be a hierarchy of vertices, and $A_k = \emptyset$. Let $v \in A_i \setminus A_{i+1}$ for $i \in [0, k-1]$. We define the *open ball* of v to be

$$\text{Ball}(v) = \{u \in V_i : \mathsf{dist}_w(v, u) < \mathsf{dist}_w(v, p_{i+1}(v)\}.$$

The *closed ball* of v is

$$\text{Ball}[v] = \text{Ball}(v) \cup \{p_{i+1}(v)\}.$$

Note in particular that, since we have earlier defined $\mathsf{dist}_w(v, p_k(v)) = \infty$, it follows that for $v \in A_{k-1}$ we have $\text{Ball}(v) = \text{Ball}[v] = V$. The emulator construction is then given below by TZ-Emulator.

TZ-Emulator
Input: A weighted undirected graph $G = (V, E, w)$

1. Let $V = A_0 \supseteq A_1 \supseteq \ldots \supseteq A_{k-1}$ be a hierarchy of vertices where each vertex in A_i is independently promoted to A_{i+1} with probability $n^{-1/k}$, and set $A_k = \emptyset$
2. Return $\bigcup_{v \in V} \{(v, u) : u \in \text{Ball}[v]\}$

Huang and Pettie showed in [HP19] the following proposition with regards to TZ-Emulator constructing a hopset.

Proposition 5.7 (Theorem 1 in [HP19])**.** *Fix any weighted graph G and integer $k \geq 1$. TZ-Emulator returns a (β, ε)-hopset for G with with size $O(n^{1+\frac{1}{2^{k+1}-1}})$ and $\beta = 2\left(\frac{(4+o(1))k}{\varepsilon}\right)^k = O\big((\frac{k}{\varepsilon})^k\big)$.*

In order to ascertain that Apx-Undirected-Hopset gives an $\Big(\widetilde{O}\big((k/\varepsilon)^k\big), \varepsilon\Big)$-hopset, it therefore suffices to show that all edges in TZ-Emulator can be traversed using a polylogarithmic number of edges of Apx-Undirected-Hopset.

Lemma 5.8. *For any edge (v, u) in the set returned by TZ-Emulator, the set returned by Apx-Undirected-Hopset contains a path with the same weight as (v, u) and, moreover, has $O(\log^2 n)$ hop-length.*

Proof. Fix any edge (v, u) in the set returned by TZ-Emulator, where $v \in A_i \setminus A_{i+1}$ and $u \in \text{Ball}[v]$. We split into two cases.

Case 1: $u \in \text{Ball}(v)$. By comparing definitions, note that $\text{Ball}(v) = \text{Bunch}_i(v) \subseteq \text{Bunch}(v)$. Thus $u \in \text{Bunch}(v)$ or, equivalently, $v \in \text{Cluster}(u)$.

Case 2: $u = p_{i+1}(v)$. By Lemma 4.1 of [TZ05], we have $u \in \text{Bunch}(v)$ and so $v \in \text{Cluster}(u)$.

In either case, by Observation 5.2 and Observation 3.10 used on T_u^* (for the definition see the main loop of Apx-Undirected-Hopset), H contains a shortest path from u to v (and thus v to u since the graph is undirected) using $O(\log^2 n)$ edges. ∎

And thus we arrive at the main result of this section.

Theorem 5. *There exists an algorithm which produces a $\big(O\big((k/\varepsilon)^k \log^2 n\big), \varepsilon\big)$-hopset H with $\|\widehat{\mathrm{v}}\|_\infty = O(kn^{1/k} \log^2 n)$ for undirected graphs. For any $\varepsilon > 0$ that is at least inverse polylogarithmic, setting $k = \Theta(\sqrt{\log n})$ gives a $(n^{o(1)}, \varepsilon)$-hopset with $\|\widehat{\mathrm{v}}\|_\infty = n^{o(1)}$.*

Proof. We show the theorem for Apx-Undirected-Hopset. This follows immediately from combining Lemma 5.8 with Proposition 5.7 to show that H is a $\big(O\big((k/\varepsilon)^k \log^2 n\big), \varepsilon\big)$-hopset, and Lemma 5.4 to finish the claim on sensitivity. ∎

6 Lower Bounds: Sensitivity-Diameter Tradeoffs

In this section we show unconditional lower bounds for how low both the sensitivity and diameter of a construction can simultaneously be. Namely, we show Theorems 2 and 4 in Section 6.1 and Theorem 6 in Section 6.2.

6.1 Tradeoffs via Perfect Paths

Our lower bounds for exact hopsets and shortcut sets go through layered graphs endowed with a set of so-called perfect paths.

Definition 6.1 (Perfect Paths, Definition 8 in [Bod21])**.** Let $G = (V_1 \sqcup V_2 \sqcup \ldots \sqcup V_\ell, E)$ be a layered graph. A set of paths Π is *perfect* if each $\pi \in \Pi$ is the unique shortest path between its endpoints, each π starts in V_1 and ends in V_ℓ with exactly one node in each layer, and each $e \in E$ is in exactly one $\pi \in \Pi$.

Layered graphs with perfect paths have been used to prove lower bounds for traditional shortcut/hopsets [Hes03, CE06, HP21, KP22b], and here we repurpose these constructions, with some changes, for our lower bounds. The "engine" of this section is Theorem 9 below, for which we first give some context before proving.

Theorem 9. *If there exists an ℓ-layered graph G with n nodes in each layer, and a set of perfect paths Π, then there exists a 2ℓ-layered graph G' with n nodes in each layer such that any ℓ-shortcut set or $(\ell, 0)$-hopset H on G' must have $\|\widehat{\mathsf{e}}\|_\infty \geq \frac{|\Pi|}{2n}$.*

Simplifying Assumptions. First, observe that if G is unweighted and we are concerned with directed ℓ-shortcut sets, then for any s that reaches t, the length of any st-path is completely determined by the layers they belong to; any shortcut set on G is thus also a hopset on G (where all edges of G are given weight 1). It is therefore sufficient to show the theorem for $(\ell, 0)$-hopsets.

Definition of the graph G'. We can show a lower bound for vertex sensitivity using G from the hypothesis of Theorem 9 directly (in fact, the proof is simpler). However, to get a lower bound for *edge* sensitivity, a little more care needs to be taken: we add a 0 weight edge in the "middle" of each vertex of G, which gives us G'. We now describe G' more formally.

G' is a standard transformation of G comprising of (i) vertices v^{in}, v^{out} for all $v \in V(G)$; (ii) 0 weight edges (v^{in}, v^{out}) for all $v \in V(G)$; (iii) edges (u^{out}, v^{in}) for all $(u, v) \in E(G)$, with the same weight. The proof will only look at the set of paths Π' in G' which are "lifted" from Π (dually, which project down back to Π), formally defined as follows:

$$\Pi' = \big\{(v_1^{in}, v_1^{out}, v_2^{in}, v_2^{out}, \ldots, v_\ell^{in}, v_\ell^{out}) : (v_1, v_2, \ldots, v_\ell) \in \Pi\big\}.$$

Observe that every path $\pi' \in \Pi'$ is the unique shortest path between its endpoints (which starts in the first layer and ends in layer 2ℓ), a property inherited from Π being a set of perfect paths. From this, notice that there is no choice in selecting the routing path from s to t for any $s, t \in \pi'$ for all $\pi' \in \Pi'$; any such routing path must be the subpath of π' from s to t since π' is the unique shortest path between its endpoints, and so the span of hopset edges (s, t) are fixed. Moreover, each edge of the form (u_{out}, v_{in}) is in exactly one $\pi' \in \Pi'$ since Π is perfect. Finally, note that $|\Pi'| = |\Pi|$. We are now ready to state the proof.

Proof of Theorem 9. Let G' and Π' be defined as above, and let H be any $(\ell, 0)$-hopset for G'. Let $H' \subseteq H$ be a minimal set of edges which ensures that there are ℓ hop-length shortest paths in $G' \cup H'$ between the endpoints of π' for all $\pi' \in \Pi'$. Observe that $\|\widehat{\mathsf{e}}_{G',H}\|_\infty \geq \|\widehat{\mathsf{e}}_{G',H'}\|_\infty$ since $H' \subseteq H$ and so it suffices to lower bound the latter which we now write without subscripts.

Let T be the set of all edges of the form (v_{in}, v_{out}) and $\mathbb{1}_T$ be its indicator vector; note for later that $|T| = \|\mathbb{1}_T\|_1 = n\ell$. We will lower bound the total sensitivity of edges in T and use an averaging argument to get a bound on $\|\widehat{\mathsf{e}}\|_\infty$. More specifically, we will show

$$\|\widehat{\mathsf{e}}\|_\infty \geq \mathbb{1}_T \cdot \widehat{\mathsf{e}} / \|\mathbb{1}_T\|_1 \geq x/2$$

where $\|\widehat{\mathsf{e}}\|_\infty \geq \widehat{\mathsf{e}} \cdot \mathbb{1}_T / \|\mathbb{1}_T\|_1$ comes from a simple averaging argument[7]. To this end, consider the process where we add the edges of $H' = \left\{e_1, e_2, \ldots, e_{|H'|}\right\}$ to G' one by one so that $H'_0 = \{\}$ and $H'_i = H'_{i-1} \cup \{e_i\}$, and define the potential

$$\Phi_i = \sum_{\pi' \in \Pi'} \left| E(P_{G' \cup H'_i}(\pi')) \right|$$

where $P_{G' \cup H'_i}(\pi')$ denotes a shortest path in $G' \cup H'_i$ with the smallest hop-length between the endpoints of π'. Φ_i is then the sum of said hop-lengths running over $\pi' \in \Pi'$ at stage i.

Observe that the initial value Φ_0 is $2|\Pi|\ell$ (since there are $|\Pi| = |\Pi'|$ paths in Π', each with hop-length exactly 2ℓ), and the final value $\Phi_{|H'|}$ is at most $|\Pi|\ell$ (since H' is an $(\ell, 0)$-hopset). Moreover, at most one summand of Φ_i differs from the corresponding summand of Φ_{i-1} for the following reason: $\mathsf{span}(e_i)$ is contained entirely and only in one $\pi' \in \Pi'$ by the minimality of H' and uniqueness of π'. For the same reason, $\mathsf{span}(e_i)$ alternates between edges not in T and edges in T and so twice the contribution of e_i to $\mathbb{1}_T \cdot \widehat{\mathsf{e}}$ is at least the decrease in potential $\Phi_{i-1} - \Phi_i$. That is,

$$\Phi_i + 2\mathbb{1}_T \cdot \widehat{\mathsf{e}}_{G',H'_i} > \Phi_{i-1} + 2\mathbb{1}_T \cdot \widehat{\mathsf{e}}_{G',H'_{i-1}}.$$

Using our observation about the initial and final values of Φ, we conclude that

$$\begin{aligned} & |\Pi|\ell + 2\mathbb{1}_T \cdot \widehat{\mathsf{e}} \geq \Phi_{|H'|} + 2\mathbb{1}_T \cdot \widehat{\mathsf{e}} > \Phi_0 + 2\mathbb{1}_T \cdot \widehat{\mathsf{e}}_{G',H'_0} = 2|\Pi|\ell \\ \implies & \mathbb{1}_T \cdot \widehat{\mathsf{e}} \geq \frac{|\Pi|\ell}{2} \\ \implies & \frac{\mathbb{1}_T \cdot \widehat{\mathsf{e}}}{\|\mathbb{1}_T\|_1} \geq \frac{|\Pi|}{2n}. && (\|\mathbb{1}_T\|_1 = n\ell) \end{aligned}$$

We conclude from the last line that $\|\widehat{\mathsf{e}}\|_\infty \geq \frac{|\Pi|}{2n}$. ∎

Finally, we can invoke Theorem 9 with known constructions to get unconditional lower bounds on the sensitivity-diameter tradeoff.

Proposition 6.2 (Theorem 5 of [Bod21], earlier versions of which appear in [CE06, ST83])**.** *For any integers $n, \ell \leq n$, and $x \leq n/\ell$, there is an ℓ-layered weighted graph G with n nodes in each layer and a set of perfect paths Π such that each node is in exactly x paths in Π.*

Theorem 2 (Undirected Exact Hopset Lower Bound)**.** *Any construction of $(\beta, 0)$-hopsets H must have $\|\widehat{\mathsf{e}}\|_\infty \cdot \beta^2 = \Omega(n)$ for a graph G on n vertices.*

Proof. For large enough N, we invoke Proposition 6.2, setting

[7]To be overly indulgent, from an application of Hölder's inequality.

- $n \leftarrow \frac{N}{2\beta}$
- $\ell \leftarrow 2\beta$
- $x \leftarrow \frac{N}{4\beta^2}$;

call this graph G_N. Note that $|\Pi|/(2n) = x/2$. We then pass each G_N into Theorem 9 to get

$$\|\widehat{\mathsf{e}}\|_\infty \geq \frac{x}{2} = \frac{N}{8\beta^2} \implies \|\widehat{\mathsf{e}}\|_\infty \cdot \beta^2 \geq \frac{N}{8},$$

which completes the proof. ∎

Observe that Theorem 2 shows that Theorem 1 is tight up to polylogarithmic factors, but it remains open whether we can explicitly trade between the sensitivity and diameter. For example, can we find a $(\widetilde{O}(n^{1/2-c}), 0)$-hopset with $\|\widehat{\mathsf{v}}\|_\infty = \widetilde{O}(n^{2c})$ for any c? We close the discussion on exact hopsets with a remark on a connection between the problem of packing perfect paths in layered graphs and a possible threshold effect in the upper bounds for sensitivity-diameter tradeoffs.

Remark 6.3. Observe that Theorem 9 connects the existence of layered graphs with many perfect paths to sensitivity-diameter tradeoffs, and this concretely manifests as a $\|\widehat{\mathsf{e}}\|_\infty\beta^2 = \Omega(N)$ bound by using the construction of Proposition 6.2. What are the ramifications of being able to pack even more perfect paths into an ℓ-layered graph? Let us explore one particular hypothetical scenario. Suppose it is possible to pack N perfect paths into $\sqrt{N}$-layered graphs. Then, we are able to show (in exactly the same way as Theorem 2, but with different parameters) an $\|\widehat{\mathsf{e}}\|_\infty\beta = \Omega(N)$ bound for $\beta \leq \sqrt{N}/2$. It is conceivable that such a construction may exist[8]. If such a packing does indeed exist for infinitely many values of N, this would rule out the possibility of achieving a smooth sensitivity-diameter tradeoff on the upper bound side. By Theorem 1, GREEDY-HOPSET produces a $(\widetilde{O}(\sqrt{n}), 0)$-hopset with $O(\log n)$ sensitivity, and any polynomial improvement to β would raise the sensitivity very abruptly from $O(\log n)$ to $\Omega(\sqrt{n})$. That is to say, a $(\widetilde{O}(n^{1/2-c}), 0)$-hopset construction could hope for no better than $\|\widehat{\mathsf{v}}\|_\infty = \widetilde{\Omega}(n^{1/2+c})$ if we can pack N perfect paths into $\sqrt{N}$-layered graphs. On the flip side, this means that achieving a tradeoff like a $(\widetilde{O}(n^{1/2-c}), 0)$-hopset construction with $\|\widehat{\mathsf{v}}\|_\infty$ less than $\widetilde{O}(n^{1/2+c})$ by a polynomial factor, for some constant $c > 0$, would rule out the existence of $\sqrt{N}$-layered graphs packed with N perfect paths. A similar line of reasoning also shows that any polynomial improvement to β (keeping $\|\widehat{\mathsf{v}}\|_\infty = \widetilde{O}(\sqrt{n})$) over FOLKLORE-HOPSET will rule out the existence of $\sqrt{N}$-layered graphs packed with N perfect paths. The existence of layered graphs packed with as many perfect paths as possible is an open problem related to many lower bounds in the hopset/distance preservers/etc literature [BH23, CE06]; particularly, we would get more streamlined lower bounds for hopsets matching the result of [BH23].

We next show a lower bound for reachability in directed graphs.

Proposition 6.4 (Theorem 6 of [Bod21], earlier versions of which appear in [AB17, Alo02, Beh46, CE06, HP21]). *For any integers $n, d \geq 2, \ell \leq n^{1/d}$, and*

$$x = O\left(n^{\frac{d-1}{d+1}} \ell^{-d\frac{d-1}{d+1}}\right),$$

there is an ℓ-layered unweighted graph G with n nodes in each layer and a set of perfect paths Π such that each node is in exactly x paths in Π.

[8]And such a construction does not contradict Theorem 1 since, there, $\beta \geq \sqrt{N}\log n > \sqrt{N}/2$.

Theorem 4 (Directed Shortcut Set Lower Bound)**.** *Any construction of β-shortcut sets H must have $\|\widehat{\mathsf{e}}\|_\infty \cdot \beta = \Omega(n^{1/3})$ for a directed graph G on n vertices.*

Proof. For large enough N, we invoke Proposition 6.4, setting

- $n \leftarrow \frac{N}{2\beta}$
- $d \leftarrow 2$
- $\ell \leftarrow 2\beta$
- $x \leftarrow \Theta(\frac{N^{1/3}}{\beta})$;

call this graph G_N. Note that $|\Pi|/(2n) = x/2$. We then pass G_N into Theorem 9 to get

$$\|\widehat{\mathsf{e}}\|_\infty \geq \frac{x}{2} = \Theta(\frac{N^{1/3}}{\beta}) \implies \|\widehat{\mathsf{e}}\|_\infty \cdot \beta \geq \Omega(N^{1/3}),$$

which completes the proof. ∎

Curiously, none of our upper bounds and lower bounds for directed shortcut/hopsets match each other. New ideas are probably needed to better understand sensitivity-diameter tradeoffs (for both shortcut sets and hopsets) in the directed setting.

6.2 Tradeoffs for Approximate Hopsets

In this section we give a tradeoff lower bound between ε, β, and $\|\widehat{\mathsf{e}}\|_\infty$ for approximate hopsets. While the sensitivity of a hopset is not equal to its size, we can convert some claims pertaining to the size of a hopset to a slightly weaker claim about the sensitivity of a hopset. [ABP18] gives a tradeoff between ε, β, and the size of a hopset $|H|$, which we use directly to get the aforementioned type of sensitivity tradeoff.

Proposition 6.5 (Theorem 4.6 in [ABP18])**.** *Fix a positive integer k and parameter $\varepsilon > 1/n^{o(1)}$. Any construction of (β, ε)-hopsets with size less than $n^{1+\frac{1}{2^k-1}-\Delta}$, $\Delta > 0$, has $\beta = \Omega\left(\left(\frac{1}{2^{k-2}(2k-1)\varepsilon}\right)^k\right)$.*

Using this, we get the following observation as a warmup.

Observation 6.6. *Fix a positive integer k and parameter $\varepsilon > 1/n^{o(1)}$. Any construction of (β, ε)-hopsets H with $\beta = O\left(\left(\frac{1}{2^{k-2}(2k-1)\varepsilon}\right)^k\right)$ has $\|\widehat{\mathsf{v}}\|_\infty \geq n^{\frac{1}{2^k-1}-\Delta}$, $\Delta > 0$.*

Proof. Observe that any edge in H contributes at least 1 to $\|\widehat{\mathsf{v}}\|_1$. Consequently, a hopset with size $n^{1+\frac{1}{2^k-1}-\Delta}$ contributes a total of at least $n^{1+\frac{1}{2^k-1}-\Delta}$ to $\|\widehat{\mathsf{v}}\|_1$. The average vertex sensitivity is then $n^{\frac{1}{2^k-1}-\Delta}$. Since at least one vertex must incur at least the average, it follows that $\|\widehat{\mathsf{v}}\|_\infty \geq n^{\frac{1}{2^k-1}-\Delta}$.

Taking the contrapositive of Proposition 6.5, any hopset with $\beta = O\left(\left(\frac{1}{2^{k-2}(2k-1)\varepsilon}\right)^k\right)$ must have size at least $n^{1+\frac{1}{2^k-1}-\Delta}$ which, by our earlier discussion, must have $\|\widehat{\mathsf{v}}\|_\infty \geq n^{\frac{1}{2^k-1}-\Delta}$. ∎

Observation 6.6 is not quite as strong as what we would like to prove; it bounds $\|\widehat{\mathsf{v}}\|_\infty$ but we would like to get a bound on $\|\widehat{\mathsf{e}}\|_\infty$. Averaging over the number of edges does not go through since the graph from [ABP18] contains a super-linear number of edges. To this end, we have to peek into the guts of the proof of Proposition 6.5 just a little bit.

Theorem 6. *Fix a positive integer k and parameter $\varepsilon > 1/n^{o(1)}$. Any construction of (β, ε)-hopsets H with $\beta = O\left(\left(\frac{1}{2^{k-2}(2k-1)\varepsilon}\right)^k\right)$ has $\|\widehat{\mathsf{e}}\|_\infty \geq n^{\frac{1}{2^k-1}-\Delta}$, $\Delta > 0$.*

Proof. First, we take the recursive construction G used in the proof of Proposition 6.5 and do a standard graph transformation to get G' comprising of (i) vertices v^{in}, v^{out} for all $v \in V(G)$; (ii) 0 weight edges (v^{in}, v^{out}) for all $v \in V(G)$; (iii) edges (u^{out}, v^{in}) for all $(u, v) \in E(G)$, with the same weight. Let T be the set of all edges of the form (v_{in}, v_{out}) and $\mathbb{1}_T$ be its indicator; note for later that $|T| = \|\mathbb{1}_T\|_1 = n$. Our goal is as follows: for any (β, ε)-hopset H' of G' satisying the hypothesis, we seek to lower bound $\widehat{\mathsf{e}} \cdot \mathbb{1}_T$, the sum of sensitivies over edges in T.

Inspecting the result of [ABP18], the proof of Proposition 6.5 actually gives a lower bound on the size of a subset of edges in a hopset with $\beta = O\left(\left(\frac{1}{2^{k-2}(2k-1)\varepsilon}\right)^k\right)$; these are the so-called *long and owned* edges. Long and owned edges e have the following property: $\mathsf{span}(e)$ goes *through* a layer of G. This means that such an edge "lifted" into G' must contain an edge of the form (v_{in}, v_{out}) in $\mathsf{span}(e')$ and therefore contribute at least 1 to $\widehat{\mathsf{e}} \cdot \mathbb{1}_T$.

Taking the contrapositive of Proposition 6.5, any hopset with $\beta = O\left(\left(\frac{1}{2^{k-2}(2k-1)\varepsilon}\right)^k\right)$ must have size at least $n^{1+\frac{1}{2^k-1}-\Delta}$ which, by our earlier discussion, must mean that $\widehat{\mathsf{e}} \cdot \mathbb{1}_T \geq n^{1+\frac{1}{2^k-1}-\Delta}$. By an averaging argument, we get

$$\|\widehat{\mathsf{e}}\|_\infty \geq \frac{\widehat{\mathsf{e}} \cdot \mathbb{1}_T}{\|\mathbb{1}_T\|_1} \geq n^{\frac{1}{2^k-1}-\Delta},$$

completing the proof. ∎

Observe that Theorem 6 shows that, in the regime where ε is exactly inverse polylogarithmic, having a polylogarithmic β (which is of the form $\beta = O_k\big((1/\varepsilon)^k\big)$ for some constant k) necessitates at least a polynomial edge sensitivity of roughly $n^{1/2^k}$. That is to say, at least one of β or $\|\widehat{\mathsf{e}}\|_\infty$ needs to be superpolylogarithmic when ε is inverse polylogarithmic. We leave open whether we can say something similar for other ranges of ε (particularly, for ε constant) which, if true, would show that our upper bound of $(n^{o(1)}, \varepsilon)$-hopsets with $\|\widehat{\mathsf{v}}\|_\infty = n^{o(1)}$ is not too far from the best one could hope for: we could hope to improve one of β or $\|\widehat{\mathsf{v}}\|_\infty$ to polylogarithmic, but not both simultaneously. Finally, note that a similar gap is present in the traditional hopset literature: it is not known whether for a constant $\varepsilon > 0$ if we can get a $(O(\log^{k_1} n), 0)$-hopset with $O(n \log^{k_2} n)$ size, for constants k_1, k_2. In fact, the lower bound we have here matches the best known traditional hopset lower bound [ABP18], where a size s there corresponds to a sensitivity s/n here.

7 Open Problems

In this section we collect a list of problems left open by our results. First and foremost, the picture pertaining to *directed* graphs is least clear.

- The bounds for shortcut sets (see Theorem 3 and Theorem 4) on directed graphs are not tight. Can we provide tight results for any specific value of sensitivity (say $O(\log n)$) if not for all values?

- The best known upper bound for directed exact hopsets is based on the folklore algorithm for traditional hopsets. It had been a long standing open problem whether this was the best we could do, and it is only within the last year that the folklore algorithm has been shown to be optimal [BH23]. Can similarly fresh ideas close the gap between the folklore algorithm (see Lemma D.1) and lower bounds here (see Theorem 2)?

- We know essentially nothing insightful about the situation for directed approximate hopsets, besides the fact that our shortcut set lower bound in Theorem 4 applies (since a directed approximate hopset is a directed shortcut set). Find a non-trivial upper bound.

While more is known for undirected graphs, there are certainly gaps left to fill.

- The upper bound given by Greedy-Hopset only works for a specific sensitivity regime (i.e. it completes the picture for only one point on the tradeoff curve). Find a non-trivial algorithm, that outputs hopsets close to the curve of Theorem 2, where the sensitivity is tunable to values larger than polylogarithmic. See Remark 6.3 for more context surrounding this problem.
- Our lower bounds for undirected approximate hopsets in Theorem 6 are not tight; they do not address the regime when ε is a constant well. For a constant $\varepsilon > 0$, do there exist (β, ε)-hopsets with β and $\|\widehat{\mathrm{v}}\|_\infty$ simultaneously polylogarithmic, or must one of the quantities necessarily be superpolylogarithmic?

Finally, we believe that hopset sensitivity is a natural notion which should have concrete algorithmic applications beyond differential privacy. Where else can low-sensitivity hopsets can be used as a primitive?

Acknowledgements

The authors would like to thank Greg Bodwin for helpful discussions about hopsets.

References

[AB17] Amir Abboud and Greg Bodwin. The 4/3 additive spanner exponent is tight. *Journal of the ACM (JACM)*, 64(4):1–20, 2017. 22

[ABP18] Amir Abboud, Greg Bodwin, and Seth Pettie. A hierarchy of lower bounds for sublinear additive spanners. *SIAM Journal on Computing*, 47(6):2203–2236, 2018. 1, 4, 9, 23, 24

[Alo02] Noga Alon. Testing subgraphs in large graphs. *Random Structures & Algorithms*, 21(3-4):359–370, 2002. 22

[BDG$^+$24] Greg Bodwin, Chengyuan Deng, Jie Gao, Gary Hoppenworth, Jalaj Upadhyay, and Chen Wang. The discrepancy of shortest paths, 2024. 5, 34

[Beh46] Felix A Behrend. On sets of integers which contain no three terms in arithmetical progression. *Proceedings of the National Academy of Sciences*, 32(12):331–332, 1946. 22

[BGWN20] Aaron Bernstein, Maximilian Probst Gutenberg, and Christian Wulff-Nilsen. Near-optimal decremental SSSP in dense weighted digraphs. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1112–1122. IEEE, 2020. 1

[BH23] Greg Bodwin and Gary Hoppenworth. Folklore sampling is optimal for exact hopsets: Confirming the $\sqrt{n}$ barrier. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 701–720. IEEE, 2023. ii, 1, 7, 22, 25, 30

[BLP20] Uri Ben-Levy and Merav Parter. New (α, β) spanners and hopsets. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1695–1714. SIAM, 2020. 1

[Bod21] Greg Bodwin. New results on linear size distance preservers. *SIAM Journal on Computing*, 50(2):662–673, 2021. 9, 20, 21, 22

[BW23] Aaron Bernstein and Nicole Wein. Closing the gap between directed hopsets and shortcut sets. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 163–182. SIAM, 2023. 1

[CE06] Don Coppersmith and Michael Elkin. Sparse sourcewise and pairwise distance preservers. *SIAM Journal on Discrete Mathematics*, 20(2):463–501, 2006. 3, 9, 20, 21, 22

[CFR20] Nairen Cao, Jeremy T Fineman, and Katina Russell. Efficient construction of directed hopsets and parallel approximate shortest paths. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 336–349, 2020. 1

[CFR21] Nairen Cao, Jeremy T Fineman, and Katina Russell. Brief announcement: An improved distributed approximate single source shortest paths algorithm. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing*, pages 493–496, 2021. 1

[CGK$^+$23] Justin Y. Chen, Badih Ghazi, Ravi Kumar, Pasin Manurangsi, Shyam Narayanan, Jelani Nelson, and Yinzhan Xu. Differentially private all-pairs shortest path distances: Improved algorithms and lower bounds. In *SODA*, 2023. 2, 3, 4, 6, 31, 35

[Coh00] Edith Cohen. Polylog-time and near-linear work approximation scheme for undirected shortest paths. *Journal of the ACM (JACM)*, 47(1):132–166, 2000. 1

[CSS11] T-H Hubert Chan, Elaine Shi, and Dawn Song. Private and continual release of statistics. *ACM Transactions on Information and System Security (TISSEC)*, 14(3):1–24, 2011. 32

[CZ24] Shiri Chechik and Tianyi Zhang. Path-reporting distance oracles with logarithmic stretch and linear size. In *51st International Colloquium on Automata, Languages, and Programming (ICALP 2024)*, pages 42:1–42:18. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2024. 2

[DGUW23] Chengyuan Deng, Jie Gao, Jalaj Upadhyay, and Chen Wang. Differentially private range query on shortest paths. In *Algorithms and Data Structures Symposium*, pages 340–370. Springer, 2023. 2, 3, 4, 5, 6, 31, 34

[DMNS16] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. *Journal of Privacy and Confidentiality*, 7(3):17–51, 2016. 33

[DR14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014. 4

[EN19] Michael Elkin and Ofer Neiman. Hopsets with constant hopbound, and applications to approximate shortest paths. *SIAM Journal on Computing*, 48(4):1436–1480, 2019. 1, 8, 16

[ES23] Michael Elkin and Idan Shabat. Path-reporting distance oracles with logarithmic stretch and size o (n log log n). In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 2278–2311. IEEE, 2023. 2

[Fin18] Jeremy T Fineman. Nearly work-efficient parallel algorithm for digraph reachability. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 457–470, 2018. 1

[FL22] Chenglin Fan and Ping Li. Distances release with differential privacy in tree and grid graph. In *2022 IEEE International Symposium on Information Theory (ISIT)*, pages 2190–2195. IEEE, 2022. 2, 3, 6, 10, 11

[FN18] Sebastian Forster and Danupon Nanongkai. A faster distributed single-source shortest paths algorithm. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 686–697. IEEE, 2018. 1

[GWN20] Maximilian Probst Gutenberg and Christian Wulff-Nilsen. Decremental SSSP in weighted digraphs: Faster and against an adaptive adversary. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2542–2561. SIAM, 2020. 1

[Hes03] William Hesse. Directed graphs requiring large numbers of shortcuts. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '03, page 665–669, USA, 2003. Society for Industrial and Applied Mathematics. 1, 20

[HKN14] Monika Henzinger, Sebastian Krinninger, and Danupon Nanongkai. Sublinear-time decremental algorithms for single-source reachability and shortest paths on directed graphs. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 674–683, 2014. 1

[HKN15] Monika Henzinger, Sebastian Krinninger, and Danupon Nanongkai. Improved algorithms for decremental single-source reachability on directed graphs. In *Automata, Languages, and Programming: 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I 42*, pages 725–736. Springer, 2015. 1

[HP19] Shang-En Huang and Seth Pettie. Thorup–Zwick emulators are universally optimal hopsets. *Information Processing Letters*, 142:9–13, 2019. 1, 8, 16, 17, 18, 19

[HP21] Shang-En Huang and Seth Pettie. Lower bounds on sparse spanners, emulators, and diameter-reducing shortcuts. *SIAM Journal on Discrete Mathematics*, 35(3):2129–2144, 2021. 20, 22

[HT84] Dov Harel and Robert Endre Tarjan. Fast algorithms for finding nearest common ancestors. *siam Journal on Computing*, 13(2):338–355, 1984. 10, 11

[JLS19] Arun Jambulapati, Yang P Liu, and Aaron Sidford. Parallel reachability in almost linear work and square root depth. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1664–1686. IEEE, 2019. 1

[KP22a] Shimon Kogan and Merav Parter. Beating matrix multiplication for $n^{1/3}$-directed shortcuts. In *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*, pages 82:1–82:20. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022. 1

[KP22b] Shimon Kogan and Merav Parter. Having hope in hops: New spanners, preservers and lower bounds for hopsets. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 766–777. IEEE, 2022. 1, 20

[KP22c] Shimon Kogan and Merav Parter. New diameter-reducing shortcuts and directed hopsets: Breaking the barrier. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1326–1341. SIAM, 2022. 1, 6

[KS97] Philip N Klein and Sairam Subramanian. A randomized parallel algorithm for single-source shortest paths. *Journal of Algorithms*, 25(2):205–220, 1997. 1

[KS21] Adam Karczmarz and Piotr Sankowski. A deterministic parallel APSP algorithm and its applications. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 255–272. SIAM, 2021. 1

[LWWX22] Kevin Lu, Virginia Vassilevska Williams, Nicole Wein, and Zixuan Xu. Better lower bounds for shortcut sets and additive spanners via an improved alternation product. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3311–3331. SIAM, 2022. 1

[Sea16] Adam Sealfon. Shortest paths and distances with differential privacy. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 29–41, 2016. 34

[SN21] Idan Shabat and Ofer Neiman. A unified framework for hopsets and spanners. *arXiv preprint arXiv:2108.09673*, 2021. 1

[SS99] Hanmao Shi and Thomas H Spencer. Time–work tradeoffs of the single-source shortest paths problem. *Journal of algorithms*, 30(1):19–32, 1999. 1

[ST83] Endre Szemerédi and William T. Trotter. Extremal problems in discrete geometry. *Combinatorica*, 3(3):381–392, 1983. 21

[Tar10] Sasu Tarkoma. *Overlay Networks: Toward Information Networking*. CRC Press, February 2010. 4

[Tho93] Mikkel Thorup. On shortcutting digraphs. In *Graph-Theoretic Concepts in Computer Science: 18th International Workshop, WG'92 Wiesbaden-Naurod, Germany, June 18–20, 1992 Proceedings 18*, pages 205–211. Springer, 1993. 1

[TZ05] Mikkel Thorup and Uri Zwick. Approximate distance oracles. *Journal of the ACM (JACM)*, 52(1):1–24, 2005. 16, 17, 18, 19

[TZ06] Mikkel Thorup and Uri Zwick. Spanners and emulators with sublinear distance errors. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 802–809, 2006. 8, 16, 17, 18

[UY90] Jeffery Ullman and Mihalis Yannakakis. High-probability parallel transitive closure algorithms. In *Proceedings of the second annual ACM symposium on Parallel algorithms and architectures*, pages 200–209, 1990. 1, 31

[Wai19] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019. 32

[WXX24] Virginia Vassilevska Williams, Yinzhan Xu, and Zixuan Xu. Simpler and higher lower bounds for shortcut sets. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2643–2656. SIAM, 2024. 1

A On Applying [BH23] to Sensitivity-Diameter Lower Bounds

Here we continue Remark 2.1 in a little more detail.

A recent breakthrough of Bodwin and Hoppenworth [BH23] was able to prove tight lower bounds for the *number of edges* in a directed exact hopset by using a different kind of construction that relaxes the property that perfect paths (see Definition 6.1) are internally disjoint. Unfortunately, their relaxation does not work for lower bounding the *sensitivity* of a hopset.

Loosely speaking, in the construction of Bodwin and Hoppenworth, perfect paths are allowed to intersect, but their pairwise intersection is always a segment with a small number of edges. This means that a shortcut edge (s, t) can shortcut multiple perfect paths only if the shortcut is low-hop – i.e. only if s and t are in nearby layers. Since low-hop shortcuts only make limited progress in reducing the hop-distance between the first and last layer of the overall graph, the authors are able to conclude that it would require a large number of low-hop shortcuts to reduce the number of hops on every perfect path. But in our setting we do not limit the number of shortcut edges, so by using a large number of these low-hop shortcuts, one can reduce the diameter of their construction while incurring low sensitivity; for this reason, their approach does not lead to an improved lower bound in our setting.

B Proofs of Path-Hopset and Tree-Hopset Guarantees

For completeness, we prove the guarantees of Path-Hopset and Tree-Hopset here.

Observation 3.8. *Given a path $P = v_0, v_1, \ldots, v_n$ (possibly oriented in the (v_i, v_{i+1}) direction), Path-Hopset outputs a $O(\log n)$-shortcut/hopset with $\|\widehat{\mathrm{v}}\|_\infty = O(\log n)$.*

Proof.
Output is a $O(\log n)$-shortcut/hopset. Let $H =$ Path-Hopset(P) and u be any vertex along P. By an induction argument, there is always an edge $(v_0, v) \in H$ where v_0 reaches v which reaches u, and the P hop-length from v_0 to v is at least as much as that from v to u. Thus, the $P \cup H$ hop-length from v_0 to u is at most $O(\log n)$. A similar argument shows that the $P \cup H$ hop-length from u to v_n is at most $O(\log n)$. Finally, let u and v be any two vertices in P where u reaches v. Consider the first time u and v are separated into different recursive instances with endpoints x, y and y, z respectively. By the above reasoning, the $P \cup H$ hop-lengths from u to y and from y to v are both at most $O(\log n)$, and by concatenating them we can see that the $P \cup H$ hop-length from u to v is at most $O(\log n)$.
Sensitivity is bounded by $O(\log n)$. Consider the recursion tree of Path-Hopset, where each node is labelled with its recursive input. For example, the root node is labelled with $P = v_0, v_1, \ldots, v_n$, its left child is labelled with $v_0, v_1, \ldots, v_{n/2}$ and its right child is labelled with $v_{n/2}, v_{n/2+1}, \ldots, v_n$, and so on. A vertex is contained in at most two labels per level. Since each appearance of a vertex v in a label contributes 1 to $\widehat{\mathrm{v}}(v)$ and there are $O(\log n)$ levels, $\widehat{\mathrm{v}}(v) = O(\log n)$ for all $v \in V$ and therefore $\|\widehat{\mathrm{v}}\|_\infty = O(\log n)$. ∎

Observation 3.10. *Given a tree T rooted at v (with possibly all edges oriented away from the root or all edges oriented towards the root), Tree-Hopset outputs an $O(\log^2 n)$-shortcut/hopset with $\|\widehat{\mathrm{v}}\|_\infty = O(\log n)$.*

Proof.
Output is a $O(\log^2 n)$-shortcut/hopset. Let $H =$ Tree-Hopset(T rooted at v). For any pair of vertices s, t, consider the unique path in T, which by Proposition 3.9 is comprised of at most

$O(\log n)$ heavy (sub)paths and $O(\log n)$ light edges. For each heavy path P, any pair of vertices in P have a $P \cup H$ hop-length of $O(\log n)$ by Observation 3.8. We can stitch together light edges and the aforementioned $O(\log n)$ hop-length paths for each heavy path to show the $T \cup H$ hop-length from s to t is at most $O(\log^2 n)$.
Sensitivity is bounded by $O(\log n)$. For any heavy path P, Observation 3.8 asserts that PATH-HOPSET(P) contributes $O(\log n)$ to $\widehat{\mathrm{v}}(v)$ for all $v \in V(P)$. By Proposition 3.9, heavy paths are vertex disjoint and, consequently, the only contribution to $\widehat{\mathrm{v}}(v)$ is from the call to PATH-HOPSET(P) where $v \in V(P)$. Thus, $\|\widehat{\mathrm{v}}\|_\infty = O(\log n)$. ∎

C Undirected Shortcut Sets Upper Bound

A direct application of TREE-HOPSET gives us undirected low sensitivity shortcut sets.

```
UNDIRECTED-SHORTCUT-SET
Input: An undirected graph $G = (V, E)$

  1. $H \leftarrow \emptyset$
  2. For each connected component $C$ of $G$
      (a) Select a representative $c \in V(C)$
      (b) $H \leftarrow H \cup$ TREE-HOPSET(the union of all routing paths rooted at $c$)
  3. Return $H$
```

Observation C.1. *UNDIRECTED-SHORTCUT-SET produces an $O(\log^2 n)$-shortcut set with $\|\widehat{\mathrm{v}}\|_\infty = O(\log n)$.*

Proof. This follows immediately from Observation 3.10 and observing that reachability is preserved. ∎

D Directed Exact Hopsets Upper Bound

Using ideas from [CGK+23, DGUW23], we can modify a folklore randomized algorithm (attributed to [UY90]), which gives a directed $(\widetilde{O}(\sqrt{n}), 0)$-hopset with linear size, to get a directed $(\widetilde{O}(\sqrt{n}), 0)$-hopset with $\|\widehat{\mathrm{v}}\|_\infty = \widetilde{O}(\sqrt{n})$.

```
FOLKLORE-HOPSET
Input: A directed graph $G = (V, E, w)$

  1. Take each $v \in V$ into the set $X$ independently with probability $n^{-1/2}$
  2. $H \leftarrow \emptyset$
  3. For each $v \in X$
      (a) $H \leftarrow H \cup$ TREE-HOPSET(shortest path arborescence rooted at $v$)
  4. Return $H$
```

Lemma D.1. *Folklore-Hopset produces a directed* $(O(\sqrt{n}\log n), 0)$*-hopset with* $\mathbb{E}[\|\widehat{\mathrm{v}}\|_\infty] = O(\sqrt{n}\log n)$.

Proof.
With high probability, all shortest path hop-lengths are bounded by $(O(\sqrt{n}\log n)$**.** Let P be a G shortest path from u to v with $\Omega(\sqrt{n}\log n)$ hop-length. Then, with high probability at least one of the first $\Theta(\sqrt{n}\log n)$ vertices of P is sampled into X; call any one of these vertices x and let its shortest path arborescence be denoted with T_x. By Observation 3.10, Tree-Hopset gives an $O(\log^2 n)$ hop-length $G \cup T_x$ shortest path from x to v and so a shortest path from u to v can be taken using $O(\sqrt{n}\log n)$ edges in G from u to x and $O(\log^2 n)$ edges from x to v using edges in $G \cup T_x$, which totals to $(O(\sqrt{n}\log n)$ edges. By a union bound, this holds with high probability for all $O(n^2)$ shortest paths using $\Omega(\sqrt{n}\log n)$ edges.
Sensitivity is bounded by $O(\sqrt{n}\log n)$ **in expectation.** The expected size of X is $\sqrt{n}$, and by Observation 3.10 each call to Tree-Hopset(T_x) for $x \in X$ contributes at most $O(\log n)$ to $\widehat{\mathrm{v}}(v)$ for all $v \in V$, hence the claim follows. ∎

A simple application of the probabilistic method on Lemma D.1 then shows the existence of $(O(\sqrt{n}\log n), 0)$-hopsets with $\|\widehat{\mathrm{v}}\|_\infty = O(\sqrt{n}\log n)$.

E Applications to Differential Privacy

In this section, we present the application of low-sensitivity hopsets in the problem of All Sets Range Queries (ASRQ) with differnetial privacy introduced in Section 1.2. We first show the proof of Theorem 8 for the ASRQ problem and discuss the connection between low-sensitivity hopsets and All Pairs Shortest Distances (APSD) problem.

E.1 Technical Tools

The main technical tool we use for the proof is the Laplace Mechanism. For completeness we provide the necessary definitions.

Definition E.1 (*Laplace distribution*)**.** We say a zero-mean random variable X follows the Laplace distribution with parameter b (denoted by $X \sim \mathsf{Lap}(b)$) if the probability density function of X follows

$$p(x) = \mathsf{Lap}(b)(x) = \frac{1}{2b} \cdot \exp\left(-\frac{|x|}{b}\right).$$

Laplace random variables exhibit a nice concentration property, formally as follows.

Lemma E.2 (Sum of Laplace random variables, [CSS11, Wai19])**.** Let $\{X_i\}_{i=1}^m$ be a collection of independent random variables such that $X_i \sim \mathsf{Lap}(b_i)$ for all $1 \le i \le m$. Then, for $\nu \ge \sqrt{\sum_i b_i^2}$ and $0 < \lambda < \frac{2\sqrt{2}\nu^2}{b}$ for $b = \max_i\{b_i\}$,

$$\Pr\left[\left|\sum_i X_i\right| \ge \lambda\right] \le 2 \cdot \exp\left(-\frac{\lambda^2}{8\nu^2}\right).$$

We move forward to sensitivity as known in DP, then we are able to define the Laplace mechanism, a standard DP mechanism that adds noise sampled from Laplace distribution with scale dependent on the ℓ_1-sensitivity.

Definition E.3 (*Sensitivity*)**.** Let $p \geq 1$. For any function $f : \mathcal{X} \to \mathbb{R}^k$ defined over a domain space $\mathcal{X}$, the ℓ_p-sensitivity of the function f is defined as

$$\Delta_{f,p} = \max_{\substack{w,w' \in \mathcal{X} \\ w \sim w'}} \|f(w) - f(w')\|_p,$$

Here, $\|\mathbf{x}\|_p := \left(\sum_{i=1}^{d} |\mathbf{x}[i]|^p\right)^{1/p}$ is the ℓ_p-norm of the vector $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{x}[i]$ is the i-th coordinate.

Definition E.4 (*Laplace mechanism*)**.** For any function $f : \mathcal{X} \to \mathbb{R}^k$, the Laplace mechanism on input $w \in \mathcal{X}$ samples $Y_1, \ldots, Y_k$ independently from $\mathsf{Lap}(\frac{\Delta_{f,1}}{\varepsilon})$ and outputs

$$M_\varepsilon(f) = f(w) + (Y_1, \ldots, Y_k).$$

The last piece we need is the basic composition theorem.

Proposition E.5 (Composition theorem [DMNS16])**.** *For any $\varepsilon > 0$, the composition of k ε-differentially private algorithms is $k\varepsilon$-differentially private.*

E.2 Proof of Theorem 8

For completeness we define the DP-ASRQ formally.

Definition E.6 (*Range Queries with Neighboring Attributes*)**.** Let $(\mathcal{R} = (X, \mathcal{S}), f)$ be a system of range queries, and let $w, w' : X \to \mathbb{R}^{\geq 0}$ be attribute functions that map each element in X to a non-negative real number. we say w, w' are neighboring, denoted as $w \sim w'$ if $\sum_{e \in E} |w(e) - w'(e)| \leq 1$.

Definition E.7 (*Differentially Private Range Queries*)**.** Let $(\mathcal{R} = (X, \mathcal{S}), f)$ be a system of range queries and $w, w' : X \to \mathrm{R}^{\geq 0}$ be neighboring attribute functions. Let $\mathcal{A}$ be an algorithm that takes $(\mathcal{R}, f, w)$ as input. Then $\mathcal{A}$ is (ϵ, δ)-differentially private on $\mathcal{G}$ if, for neighboring attribute functions $w \sim w'$ and all sets of possible outputs $\mathcal{C}$, we have: $\Pr[\mathcal{A}(\mathcal{R}, f, w) \in \mathcal{C}] \leq e^\epsilon \cdot \Pr[\mathcal{A}(\mathcal{R}, f, w') \in \mathcal{C}] + \delta$.

We show an algorithm named as Hopset-ASRQ that calls the $(O(\sqrt{n}\log n), 0)$-hopset construction with $O(\log n)$ sensitivity as a subroutine. The high-level idea is to apply Greedy-Hopset to the given graph G and construct H using the true attributes. Now each hopset edge in H has an attribute with the true summation. To guarantee DP we add Laplace noise to the attributes on hopset edges, the noise magnitude is tempered by the sensitivity (as defined in DP), which is the hopset edge sensitivity (as defined in this paper). In addition we also add Laplace noise for each edge in the original graph. For a fixed shortest path, we report the query output along a path of at most $O(\sqrt{n}\log n)$ hops with edges in $E \cup H$, adding up the perturbed attributes of the edges. The additive error is at most the summation of all noises added. By concentration of Laplace random variables we get the claimed bound.

Hopset-ASRQ**: An ϵ-DP algorithm to release all sets range queries**
Input: Undirected graph $G = (V, E, w)$, shortest paths $\mathcal{P}$ and privacy parameter $\epsilon > 0$

```
1. $H \leftarrow$ Greedy-Hopset$(G, \mathcal{P})$
2. Let the new graph be $G' = (V, E + H, w_H)$, and shortest paths using hop edges be $\mathcal{P}'$
3. For each hop edge $e \in H$
```

   (a) Add an independent Laplace noise $\mathsf{Lap}(2\log n/\epsilon)$ to $w_H(e)$, denoted as $f'(e)$

4. For each edge $e \in E$

   (a) Add an independent Laplace noise $\mathsf{Lap}(2/\epsilon)$ to $w_H(e)$, denoted as $f'(e)$

5. For each $(u, v) \in V \times V$

   (a) Report the query output as $\hat{f}(u, v) = \sum_{e \in E(P_{uv})} f'(e)$

The following lemma immediately proves Theorem 8.

Lemma E.8. *For any undirected graph G and any $\epsilon > 0$,* Hopset-ASRQ *is an ϵ-DP algorithm that outputs all sets range queries with additive error at most $\widetilde{O}(\frac{n^{1/4}}{\epsilon})$ with high probability.*

Proof.
ϵ-DP guarantee. The privacy proof directly comes from the Laplace mechanism, which is applied twice in Hopset-ASRQ. By Theorem 1 the sensitivity of each hop edge $e \in H$ is $\log n$, and the sensitivity of individual edges in E is simply 1. Therefore by Definition E.4 Step 3 and 4 in Hopset-ASRQ both satisfy $\epsilon/2$-DP. By the composition theorem Proposition E.5, we arrive at ϵ-DP for Hopset-ASRQ.
Analysis of additive error. Let H be the hospet given by Hopset-ASRQ, the new graph $G' = G \cup H$ now has at most $\widetilde{O}(\sqrt{n})$ hop diameter. For edges any shortest paths P'_{uv} in G', there could be hop edges with large perturbation from step 3, and edges with small perturbation from step 4. However, the number of edges involved on a shortest path is always bounded by $\widetilde{O}(\sqrt{n})$. Therefore the maximum additive error can be decomposed into $\widetilde{O}(\sqrt{n})$ independent noises sampled from $\mathsf{Lap}(2\log n/\epsilon)$. Applying Lemma E.2 with a standard union bound argument, we get the upper bound of $\widetilde{O}(\frac{n^{1/4}}{\epsilon})$. ∎

Recall prior results in [DGUW23], we are able to improve the ϵ-DP additive error upper bound from $\widetilde{O}(n^{1/3})$ to $\widetilde{O}(n^{1/4})$, matching the $\widetilde{O}(n^{1/4})$ upper bound for the (ϵ, δ)-DP setting, which is tight up to polylogarithmic factors according to a recent work [BDG+24]. Another remark is that other exact hopset constructions can also be applied to the framework of Hopset-ASRQ by just replacing Greedy-Hopset. Theorem 7 therefore holds universally as the connection between exact hopsets and the DP-ASRQ problem.

E.3 Discussion on All Pairs Shortest Distances

The problems of APSD and ASRQ share many similarities. Given a graph with public topology and private weights, the APSD problem aims to output weighted shortest distances under DP, and minimize the additive error, which is again the ℓ_∞ norm of the difference between the true distances and the perturbed ones. We define the problem formally.

Definition E.9 (*Differentially Private APSD [Sea16]*)**.** Let $w, w' : E \to R^{\geq 0}$ be weight functions, and $\mathcal{A}$ be an algorithm taking a graph $G = (V, E)$ and w as input. The algorithm A is (ε, δ)-differentially private on G if for any neighboring weights $w \sim w'$ and all sets of possible output $\mathcal{C}$, we have: $\Pr[\mathcal{A}(G, w) \in \mathcal{C}] \leq e^{\varepsilon} \cdot \Pr[\mathcal{A}(G, w') \in \mathcal{C}] + \delta$.

One may wonder why we cannot obtain a similar connection as Theorem 7 between the DP-APSD problem and low-sensitivity hopsets. There is a crucial yet subtle difference between two problem that falls into the privacy notion: the weights that determine the shortest paths are private in the APSD problem, but are public knowledge in the ASRQ problem. In fact, the latter does not concern weights but other attributes irrelevant to the shortest paths, which are assumed to be given to algorithms. In other words, algorithms for DP-APSD cannot use the true shortest paths as a white-box however the other is fine.

The discussion above implies that we cannot use Greedy-Hopset to construct a low-sensitivity hopset because the ground-truth shortest paths are used therein. The currently state-of-the-art algorithms [CGK+23] use the folklore hopset construction by sampling a set of vertices S uniformly at random. This procedure does not involve using the shortest paths.

On the other hand, this subtlety does not refrain from having another low-sensitivity hopset construction that does not involve the shortest paths (or edge weights). If we do have an algorithm as such, which outputs a $(\beta, 0)$-hopset with $\|\widehat{\mathsf{e}}\|_\infty$ sensitivity, a similar theorem as Theorem 8 still holds for the DP-APSD problem. Knowing that the current best upper bound for the ϵ-DP setting is $\widetilde{O}(n^{2/3})$, if we have $\beta \cdot \|\widehat{\mathsf{e}}\|_\infty = o(n^{4/3})$, we improve the previous best ϵ-DP result!