Low Sensitivity Hopsets

Vikrant Ashvinkumar
Rutgers University, Piscataway, NJ, USA

Aaron Bernstein
New York University, NY, USA

Chengyuan Deng
Rutgers University, Piscataway, NJ, USA

Jie Gao
Rutgers University, Piscataway, NJ, USA

Nicole Wein
University of Michigan, Ann Arbor, MI, USA

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.

2012 ACM Subject Classification Theory of computation → Shortest paths; Theory of computation → Sparsification and spanners

Keywords and phrases Hopsets, Shortcuts, Sensitivity, Differential Privacy

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.13

Related Version *Full Version*: `https://arxiv.org/abs/2407.10249` [4]

Funding *Aaron Bernstein*: Funding through Sloan Fellowship, Google Research Award, and NSF Grant 1942010.
Jie Gao: Ashvinkumar, Deng and Gao would like to acknowledge funding through NSF IIS-2229876, DMS-2220271, DMS-2311064, CCF-2208663, CCF-2118953.

Nicole Wein: This work was done while the author was at DIMACS, Rutgers University, supported by a grant to DIMACS from the Simons Foundation (820931).

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

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* [7, 12, 13, 24–26, 28, 29, 33–35]. 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* [44], 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 [16] (also informally by [35, 41]) which preserves (weighted) shortest distances in addition to reachability. Formal definitions are given below.

▶ **Definition 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 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 [2, 6, 8, 11, 20, 21, 30, 31, 36–40, 46, 47].

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

[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.

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 $\|\widehat{\mathtt{v}}\|_\infty$ and $\|\widehat{\mathtt{e}}\|_\infty$[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 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 [22] to get recently improved path-reporting distance oracles (PRDOs); see [14] 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 $\|\widehat{\mathtt{e}}\|_\infty$ 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 [15, 18, 23]. 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 [18] 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.

[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.

[4] In fact, we also look at the ℓ_1 norm in the proofs of our lower bounds in Section 5.

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.	$\beta = O(\log^2 n)$ $\Vert\widehat{\mathsf{v}}\Vert_\infty = O(\log n)$ ([23])	$\beta = O(\sqrt{n}\log n)$ $\Vert\widehat{\mathsf{v}}\Vert_\infty = O(\log n)$ (Details in full version [4])	$\beta = n^{o(1)}$ $\Vert\widehat{\mathsf{v}}\Vert_\infty = n^{o(1)}$ (Details in full version [4])
	L. B.	–	$\Vert\widehat{\mathsf{e}}\Vert_\infty \cdot \beta^2 = \Omega(n)$ (Section 5.1)	$\beta = O_k((1/\varepsilon)^k)$ $\Rightarrow \Vert\widehat{\mathsf{e}}\Vert_\infty = \Omega(n^{\frac{1}{2^k-1}-\Delta})$ (Details in full version [4])
Directed	U. B.	$\beta = O(\sqrt{n}\log^3 n)$ $\Vert\widehat{\mathsf{v}}\Vert_\infty = O(\log n)$ (Details in full version [4])	$\beta = O(\sqrt{n}\log n)$ $\Vert\widehat{\mathsf{v}}\Vert_\infty = O(\sqrt{n}\log n)$ ([15, 18])	$\beta = O(\sqrt{n}\log n)$ $\Vert\widehat{\mathsf{v}}\Vert_\infty = O(\sqrt{n}\log n)$ (Implied by directed exact hopset)
	L. B.	$\Vert\widehat{\mathsf{e}}\Vert_\infty \cdot \beta = \Omega(n^{1/3})$ (Section 5.1)	$\Vert\widehat{\mathsf{e}}\Vert_\infty \cdot \beta^2 = \Omega(n)$ (Implied by undirected hopset)	$\Vert\widehat{\mathsf{e}}\Vert_\infty \cdot \beta = \Omega(n^{1/3})$ (Implied by directed shortcut set)

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 3** (Undirected Exact Hopset Upper Bound)**.** *There exists an algorithm producing an $(O(\sqrt{n}\log n), 0)$-hopset with $\Vert\widehat{\mathsf{v}}\Vert_\infty = O(\log n)$ 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 4** (Undirected Exact Hopset Lower Bound)**.** *Any construction of $(\beta, 0)$-hopsets H must have $\Vert\widehat{\mathsf{e}}\Vert_\infty \cdot \beta^2 = \Omega(n)$ for a graph G on n vertices.*

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 34), 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 [17].

Directed Shortcut Sets

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

▶ **Theorem 5** (Directed Shortcut Set Upper Bound)**.** *There exists an algorithm producing an $O(\sqrt{n}\log^3 n)$-shortcut set with $\Vert\widehat{\mathsf{v}}\Vert_\infty = O(\log n)$ for directed graphs.*

We complement our upper bound with a lower bound.

▶ **Theorem 6** (Directed Shortcut Set Lower Bound). *Any construction of β-shortcut sets H must have $\|\widehat{\mathrm{e}}\|_\infty \cdot \beta = \Omega(n^{1/3})$ 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 7.** *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 $\|\widehat{\mathrm{v}}\|_\infty = n^{o(1)}$.*

We complement our upper bound with a lower bound.

▶ **Theorem 8.** *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 $\|\widehat{\mathrm{e}}\|_\infty \geq n^{\frac{1}{2^k-1}-\Delta}$, $\Delta > 0$.*

This lower bound is an exact analogue of the best known lower bound on the size of approximate hopsets in [2] (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 22. For directed hopsets, the upper bounds implicit in prior work [15, 18] remain the best-known, 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 3) 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 [43] 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) [19]; 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 9** (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 hopsets have a direct application to the *All Sets Range Queries (ASRQ)* problem in differential privacy [18]. 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 [10]. 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 [18].

▶ **Theorem 10.** *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 10 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 3, we obtain the following result:

▶ **Theorem 11** (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 11 in the full version [4] of the paper. 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 10 is not available. We further discuss this problem in the full version [4] of the paper.

[5] Note that we are using "epsilon" in two different ways in this paper, but we disambiguate the symbols: ϵ for the differential privacy parameter, and ε for the approximation parameter of hopsets.

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 in Section 4, leaving the upper bound for directed exact hopsets to the full version [4] of the paper. Our upper bounds for undirected approximate hopsets can also be found in the full version [4] of the paper. In Section 5, we show lower bounds on the sensitivity-diameter tradeoff for shortcut sets and hopsets, leaving bounds in the approximate setting to the full version [4] of the paper. We finish by listing several open problems in Section 6.

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. 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 [23], 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 $\widetilde{O}(\sqrt{n})$ [15, 18]. 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 [38] 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 [15, 18] 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 12. Bodwin and Hoppenworth [11] 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 the full version [4] of the paper for more details.

Techniques for Approximate Undirected Hopsets

For traditional hopsets with $O(n)$ size, as noted in Section 1, 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 [20, 31] showed that Thorup-Zwick emulators [45] are optimal hopsets. Examining this construction, it is not clear if it has low vertex and edge sensitivity. Instead, we begin with the similar construction of Thorup-Zwick spanners (also in [45]). 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 13** (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 14** (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 15** (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 16** (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 17** (Consistency (Modification from [17])). 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 [2, 9]; 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 18** (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 19.** $\|\widehat{\mathsf{e}}_{G,\mathcal{A}(G)}\|_\infty \leq \|\widehat{\mathsf{v}}_{G,\mathcal{A}(G)}\|_\infty$.

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 19 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.

It is also useful to note that $|H| \leq n\|\widehat{\mathsf{v}}\|_\infty$ which says that small sensitivity implies hopset sizes which are not much larger than linear (i.e. none of our results produce prohibitively large hopsets). This follows, informally, since every vertex "sees" at most $\|\widehat{\mathsf{v}}\|_\infty$ hopset edges (the ones contributing to its sensitivity) and each hopset edge is "seen" at least once. Similar arguments will be described more formally in our lower bound proofs.

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$(P[v_{\lfloor \ell/2 \rfloor}..v_\ell])$

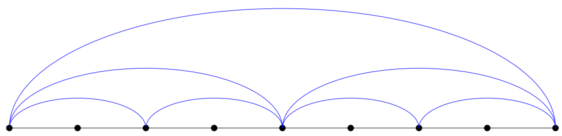

Figure 1 An example of PATH-HOPSET on a path of length 8 with vertices $v_0, \ldots, v_8$. The collection of blue edges is the shortcut/hopset output.

▶ **Observation 20.** *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 $\|\hat{\mathrm{v}}\|_\infty = O(\log n)$.*

For completeness, we provide a proof in the full version [4] of the paper.

3.3.2 Tree Shortcutting via the Heavy-light Decomposition

TREE-HOPSET, described in [23], 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 [27] (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 21.

▶ **Proposition 21** ([27])**.** *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 22.** *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 [23], but we provide a proof (in a differential-privacy-free language) in the full version [4] of the paper 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 the full version [4] of the paper 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 3. With some slight care, we apply this approach to shortcut sets on directed graphs and obtain Theorem 5.

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 23.** 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

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

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 24.** *$\Phi_{\mathcal{P}^{(i)}}$ 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 25** (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 26.** *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 20 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 27** (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 28.** *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. ◀

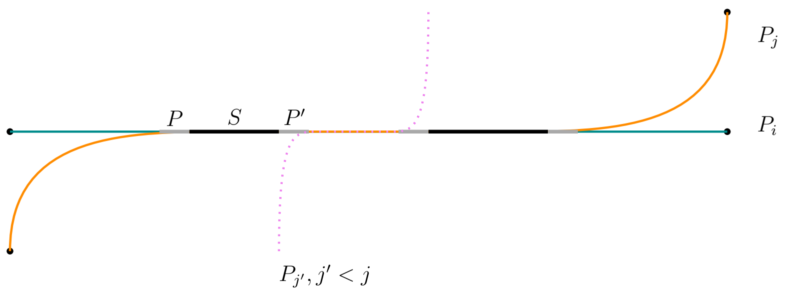

Figure 2 The penumbra P of S (a shadow cast by P_j onto P_i) is new, but the penumbra P' of S is also the penumbra of a shadow cast by $P_{j'}$ and is thus not new.

Suppose a routing path P from s to t has no more than k shadows cast onto it. Then the $G \cup \text{GREEDY-HOPSET}(G)$ 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 20 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 20 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 29.** *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 24), and Observation 28, 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 24 and Observation 28 G has $\sqrt{n} \cdot (3\sqrt{n}-1)/2 > n$ vertices, leading to a contradiction. ◀

▶ **Observation 30.** *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 20. ◀

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

Proof. We show the theorem for Greedy-Hopset. This follows from combining Lemma 29 and Observation 20, and also Observation 30. 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 26. ◀

We leave our upper bound for directed exact hopsets, proving Theorem 5, to the full version [4] of the paper.

5 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 4 and 6 in Section 5.1 and leave the proof of Theorem 8 to the full version [4] of the paper.

5.1 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 31** (Perfect Paths, Definition 8 in [9])**.** *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 [17,30,32,37], and here we repurpose these constructions, with some changes, for our lower bounds. The "engine" of this section is Theorem 32 below, for which we first give some context before proving.

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

▶ **Theorem 32.** *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 32 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:

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

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 32. 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 $H' = \{e_1, e_2, \ldots, e_{|H'|}\}$ 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.

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

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,

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

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

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

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

Finally, we can invoke Theorem 32 with known constructions to get unconditional lower bounds on the sensitivity-diameter tradeoff.

▶ **Proposition 33** (Theorem 5 of [9], earlier versions of which appear in [17, 42])**.** *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 4** (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 33, setting
- $n \leftarrow \frac{N}{2\beta}$
- $\ell \leftarrow 2\beta$
- $x \leftarrow \frac{N}{4\beta^2}$;

call this graph G_N. Note that $|\Pi|/(2n) = x/2$. We then pass each G_N into Theorem 32 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 4 shows that Theorem 3 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 34. Observe that Theorem 32 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 33. 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 4, 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 3, 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 [11, 17]; particularly, we would get more streamlined lower bounds for hopsets matching the result of [11].

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

▶ **Proposition 35** (Theorem 6 of [9], earlier versions of which appear in [1, 3, 5, 17, 32]). *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.

▶ **Theorem 6** (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 35, 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 32 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.

To see our lower bounds for approximate hopsets, proving Theorem 8, see the full version [4] of the paper.

[8] And such a construction does not contradict Theorem 3 since, there, $\beta \geq \sqrt{N} \log n > \sqrt{N}/2$.

6 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 5 and Theorem 6) 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 [11]. Can similarly fresh ideas close the gap between the folklore algorithm and lower bounds here?
- We know essentially nothing insightful about the situation for directed approximate hopsets, besides the fact that our shortcut set lower bound in Theorem 6 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 4, where the sensitivity is tunable to values larger than polylogarithmic. See Remark 34 for more context surrounding this problem.
- Our lower bounds for undirected approximate hopsets in Theorem 8 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?

References

1 Amir Abboud and Greg Bodwin. The 4/3 additive spanner exponent is tight. *Journal of the ACM (JACM)*, 64(4):1–20, 2017. doi:10.1145/3088511.

2 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. doi:10.1137/16M1105815.

3 Noga Alon. Testing subgraphs in large graphs. *Random Structures & Algorithms*, 21(3-4):359–370, 2002. doi:10.1002/RSA.10056.

4 Vikrant Ashvinkumar, Aaron Bernstein, Chengyuan Deng, Jie Gao, and Nicole Wein. Low sensitivity hopsets. *arXiv preprint*, 2024. doi:10.48550/arXiv.2407.10249.

5 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.

6 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. doi:10.1137/1.9781611975994.104.

7 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. doi:10.1109/FOCS46700.2020.00107.

8 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. doi:10.1137/1.9781611977554.CH7.

9 Greg Bodwin. New results on linear size distance preservers. *SIAM Journal on Computing*, 50(2):662–673, 2021. doi:10.1137/19M123662X.

10 Greg Bodwin, Chengyuan Deng, Jie Gao, Gary Hoppenworth, Jalaj Upadhyay, and Chen Wang. The discrepancy of shortest paths. In Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson, editors, *51st International Colloquium on Automata, Languages, and Programming (ICALP 2024)*, volume 297 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 27:1–27:20, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPICS.ICALP.2024.27.

11 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.

12 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. doi:10.1145/3357713.3384270.

13 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. doi:10.1145/3465084.3467945.

14 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. doi:10.4230/LIPICS.ICALP.2024.42.

15 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 *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 5040–5067, 2023. doi:10.1137/1.9781611977554.CH184.

16 Edith Cohen. Polylog-time and near-linear work approximation scheme for undirected shortest paths. *Journal of the ACM (JACM)*, 47(1):132–166, 2000. doi:10.1145/331605.331610.

17 Don Coppersmith and Michael Elkin. Sparse sourcewise and pairwise distance preservers. *SIAM Journal on Discrete Mathematics*, 20(2):463–501, 2006. doi:10.1137/050630696.

18 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.

19 Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014. doi:10.1561/0400000042.

20 Michael Elkin and Ofer Neiman. Linear-size hopsets with small hopbound, and distributed routing with low memory. *arXiv preprint*, 2017. arXiv:1704.08468.

21 Michael Elkin and Ofer Neiman. Hopsets with constant hopbound, and applications to approximate shortest paths. *SIAM Journal on Computing*, 48(4):1436–1480, 2019. doi:10.1137/18M1166791.

22 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. doi:10.1109/FOCS57990.2023.00141.

23 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. doi:10.1109/ISIT50566.2022.9834836.

24 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. doi:10.1145/3188745.3188926.

25 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. doi:10.1109/FOCS.2018.00071.

26 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. doi:10.1137/1.9781611975994.155.

27 Dov Harel and Robert Endre Tarjan. Fast algorithms for finding nearest common ancestors. *siam Journal on Computing*, 13(2):338–355, 1984. doi:10.1137/0213024.

28 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. doi:10.1145/2591796.2591869.

29 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. doi:10.1007/978-3-662-47672-7_59.

30 William Hesse. Directed graphs requiring large numbers of shortcuts. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '03, pages 665–669, USA, 2003. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=644108.644216.

31 Shang-En Huang and Seth Pettie. Thorup–Zwick emulators are universally optimal hopsets. *Information Processing Letters*, 142:9–13, 2019. doi:10.1016/J.IPL.2018.10.001.

32 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. doi:10.1137/19M1306154.

33 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. doi:10.1109/FOCS.2019.00098.

34 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. doi:10.1137/1.9781611976465.17.

35 Philip N Klein and Sairam Subramanian. A randomized parallel algorithm for single-source shortest paths. *Journal of Algorithms*, 25(2):205–220, 1997. doi:10.1006/JAGM.1997.0888.

36 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. doi:10.4230/LIPICS.ICALP.2022.82.

37 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. doi:10.1109/FOCS54457.2022.00078.

38 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. doi:10.1137/1.9781611977073.55.

39 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. doi:10.1137/1.9781611977073.131.

40 Idan Shabat and Ofer Neiman. A unified framework for hopsets and spanners. *arXiv preprint arXiv:2108.09673*, 2021. arXiv:2108.09673.

41 Hanmao Shi and Thomas H Spencer. Time–work tradeoffs of the single-source shortest paths problem. *Journal of algorithms*, 30(1):19–32, 1999. doi:10.1006/JAGM.1998.0968.

42 Endre Szemerédi and William T. Trotter. Extremal problems in discrete geometry. *Combinatorica*, 3(3):381–392, 1983. doi:10.1007/BF02579194.

43 Sasu Tarkoma. *Overlay Networks: Toward Information Networking*. CRC Press, February 2010.

44 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.

45 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. URL: http://dl.acm.org/citation.cfm?id=1109557.1109645.

46 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.

47 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. doi:10.1137/1.9781611977912.94.