Additive Spanner Lower Bounds with Optimal Inner Graph Structure

Greg Bodwin
University of Michigan, Ann Arbor, MI, USA

Gary Hoppenworth
University of Michigan, Ann Arbor, Mi, USA

Virginia Vassilevska Williams
Massachusetts Institute of Technology, Cambridge, MA, USA

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

Zixuan Xu
Massachusetts Institute of Technology, Cambridge, MA, USA

Abstract

We construct n-node graphs on which any $O(n)$-size spanner has additive error at least $+\Omega(n^{3/17})$, improving on the previous best lower bound of $\Omega(n^{1/7})$ [Bodwin-Hoppenworth FOCS '22]. Our construction completes the first two steps of a particular three-step research program, introduced in prior work and overviewed here, aimed at producing tight bounds for the problem by aligning aspects of the upper and lower bound constructions. More specifically, we develop techniques that enable the use of *inner graphs* in the lower bound framework whose technical properties are provably tight with the corresponding assumptions made in the upper bounds. As an additional application of our techniques, we improve the corresponding lower bound for $O(n)$-size additive emulators to $+\Omega(n^{1/14})$.

2012 ACM Subject Classification Theory of computation → Sparsification and spanners

Keywords and phrases Additive Spanners, Graph Theory

Digital Object Identifier 10.4230/LIPIcs.ICALP.2024.28

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://arxiv.org/abs/2404.18337

Funding *Greg Bodwin and Gary Hoppenworth*: Supported by NSF:AF 2153680.
Virginia Vassilevska Williams: Supported by NSF Grant CCF-2330048, BSF Grant 2020356 and a Simons Investigator Award.
Zixuan Xu: Partially supported by NSF Grant CCF-2330048.

1 Introduction

Suppose that we want to compute shortest paths or distances in an enormous graph G. When G is too big to store in memory, a popular strategy is to instead use a *spanner* of G, which is a much sparser subgraph H with approximately the same shortest path metric as G. This can substantially improve storage or runtime costs, in exchange for a small error in the distance information. Perhaps the most well-applied case is when the spanner is asymptotically as sparse as possible; that is, $|E(H)| = O(n)$ for an n-node input graph G (note that $\Omega(n)$ edges are needed just to preserve connectivity).

There are several ways to measure the quality of approximation of a spanner. The two most popular are as follows:

Table 1 The progression of upper and lower bounds on the additive error associated to n-node spanners on $O(n)$ edges; current state of the art bounds are highlighted in red. See also [10, 11, 6, 3, 2] for work on additive spanners of superlinear size.

	Upper Bound		Lower Bound	
$O(n)$-size Spanners			$\Omega(\log n)$	[23]
	$\widetilde{O}(n^{9/16})$	[20]	$\Omega(n^{1/22})$	[1]
	$\widetilde{O}(n^{1/2})$	[9]	$\Omega(n^{1/11})$	[16, 18]
	$O(n^{3/7+\varepsilon})$	[8]	$\Omega(n^{2/21})$	[19]
	$O(n^{\frac{15-\sqrt{54}}{19}<0.403})$	[21]	$\Omega(n^{1/7})$	[7]
			$\Omega(n^{3/17})$	this paper

▶ **Definition 1** (Multiplicative and Additive Spanners). *Given a graph G, a subgraph*[1] *$H \subseteq G$ is a* multiplicative $\cdot k$ spanner *if for all nodes s, t we have $dist_H(s,t) \leq dist_G(s,t) \cdot k$. It is an* additive $+k$ spanner *if we have $dist_H(s,t) \leq dist_G(s,t) + k$.*

The parameter k is called the (additive or multiplicative) *stretch* of the spanner. A famous paper of Althöfer, Das, Dobkin, Joseph, and Soares [4] settled the optimal *multiplicative* stretch for $O(n)$-size spanners:[2]

▶ **Theorem 2** ([4]). *Every n-node graph has a spanner H of size $|E(H)| = O(n)$ and multiplicative stretch $O(\log n)$. This stretch cannot generally be improved to $o(\log n)$.*

The goal of this paper is to make progress on the corresponding question for *additive* error. This question has been intensively studied; see Table 1 for the progression of results. Our contributions are on the lower bounds side:

▶ **Theorem 3** (Main Result). *There exists an infinite family of n-vertex undirected graphs for which any additive spanner on $O(n)$ edges has additive stretch $\Omega(n^{3/17})$.*

Our techniques also lead to progress on related questions for $O(n)$-size emulators, which we discuss further in Section 1.2. Before we explain this, we contextualize Theorem 3 by explaining in more depth the sense in which it moves the upper and lower bounds closer together.

1.1 Our Contribution and Next Steps for the Area

There are well-established frameworks in place for proving upper and lower bounds for $O(n)$-size spanners, and the current sentiment among experts is that these two frameworks *could* eventually produce near-matching (likely within n^{ε} factors) upper and lower bounds. Both frameworks can be broken down into three corresponding steps, and over the last few years, a research program has emerged in which the long-term goal is to find optimal bounds for the problem by making each of these three steps align.[3] That is, we can investigate what "should" happen in each step if a hypothetical optimal version of the upper bound framework were run on the graph from a hypothetical optimal version of the lower bound framework. This thought experiment leads to a list of three concrete features that should be realized in an ideal lower bound, which we overview at a high level in Table 2.

[1] Throughout the paper, for brevity, we write "subgraph" to specifically mean a subgraph over the same vertex set as the original graph.
[2] Although we generally treat input graphs G as undirected and unweighted in this paper, this particular theorem also extends to the setting where G is weighted.
[3] This program was made somewhat explicit in [7] (c.f. Section 2.4), but was implicit in work before that.

However, it is easier to write down this wishlist for the lower bound than it is to actually achieve the listed features in a construction; we discuss the various technical barriers in Section 2. The contribution of the current paper is to achieve the first two steps of alignment (i.e., the first two items in Table 2) simultaneously, which both have to do with optimizing properties of the so-called *inner graph* in the lower bound construction. That said, the ideal structure of an inner graph has been known since [8], and well before that Coppersmith and Elkin [12] found graph constructions achieving this ideal structure ("subset distance preserver lower bound graphs"). Our main technical contributions are not in designing new inner graphs, but rather, in improving the *outer graph* in a way that allows these previously known optimal inner graphs to be used within the framework for the first time.

This paper makes no real progress on the third and final point of alignment, which contends with optimizing certain quantitative properties of the shortest paths in the outer graph. Here there is still significant misalignment between the upper and lower bounds, which is responsible for essentially all of the remaining numeric gap between the current upper and lower bounds for $O(n)$-size spanners. Improving this third point, either on the upper bounds side or the lower bounds side, is the clear next step for the area and it may first require advances in our understanding of distance preservers [12]; see [7] for discussion.

1.2 Additional Results

The technical improvements to the construction that enable our improved spanner lower bounds also imply improvements for two nearby objects, which we overview next. First, an *emulator* is similar to a spanner, but not required to be a subgraph:

▶ **Definition 4** (Additive Emulators). *Given a graph G, a graph H on the same vertex set as G is an* additive $+k$ emulator *if for all nodes s, t we have*

$$dist_G(s,t) \leq dist_H(s,t) \leq dist_G(s,t) + k.$$

An emulator H is allowed to be weighted, even when the input graph G is unweighted. Emulators generalize spanners, and hence the upper and lower bounds known for $O(n)$-size emulators are a bit lower than the corresponding bounds for spanners. See Table 3 for the progression of results on the additive error that can be obtained for $O(n)$-size emulators.

A similar lower bound framework is used to achieve lower bounds for emulators, and hence our new technical machinery improves the current lower bounds for emulators as well:

▶ **Theorem 5.** *There exists an infinite family of n-vertex undirected graphs for which any additive emulator on $O(n)$ edges has additive stretch $\Omega(n^{1/14})$.*

Our numeric improvement in the lower bound for emulators is more modest than our improvement for spanners; at a high level, this is because our main improvement is to enable stronger inner graphs in the lower bound framework, but the role of the inner graph is generally less important in emulator lower bounds.

We next provide a more fine-grained overview of our lower bound framework, and we describe our technical improvements that lead to our new results in more detail.

2 Technical Overview

In this section we will give an overview of the different technical components in our lower bound graph construction. We start by reviewing the obstacle product framework in Section 2.1 and recalling some ideas from prior work in Section 2.2. Finally we will discuss the new components in our construction in Section 2.3.

Table 2 A point-by-point comparison of the frameworks used to prove upper and lower bounds. Our main technical contributions are to satisfy the first point of alignment by enabling the use of inner graphs with $\Theta(r^{4/3})$ nodes (where $+\Omega(r)$ is the desired lower bound on spanner error), and to satisfy the second point of alignment by enabling the use of subset distance preserver lower bounds for our inner graphs. Neither of these properties were fully achieved in prior work.

Step in Upper Bounds	Step in Lower Bounds	What should ideally happen when we run the upper bound framework on a lower bound graph?
Cover the input graph by clusters C of radius r each. These clusters are classified as either *small* or *large*, depending on whether their number of nodes is smaller or larger than $r^{4/3}$.	Start with an *outer graph*, and systematically replace each node with a disjoint copy of an *inner graph*.	The upper bound should select the inner graphs as its clusters. All inner graphs should have $\Theta(r^{4/3})$ nodes, since the worst case for the upper bound is when all clusters are near the large/small threshold.
Small clusters C have a node separator of size $\leq \lvert C\rvert^{1/4}$. Construct a *subset distance preserver* on each small cluster, preserving all shortest paths between separator nodes, at cost $O(\lvert C\rvert)$ [12].	The inner graphs should be selected as the union of many long unique shortest paths among nodes that form a separator for the graph, and also any two of these shortest paths may intersect on at most one node.	The inner graph should be a lower bound graph against subset distance preservers with $\Theta(\lvert C\rvert^{1/4})$ source nodes (with a large implicit constant), so that the approach of constructing a subset distance preserver is too expensive to be used in an attack against the lower bound.
Large clusters C are handled by adding some additional shortest paths in the spanner to connect far-away clusters to each other. Using the *path-buying framework* [6], we can limit the total number and length of the shortest paths we need to add.	The outer graph is selected to be the union of as many long unique shortest paths as possible, and any two of these shortest paths may intersect on at most one edge. That is, the outer graph is a slightly modified distance preserver lower bound graph.	The shortest paths added for large clusters should coincide with the shortest paths in the original outer graph (before inner graph replacement). The path-buying bounds on the number and length of these shortest paths should coincide with the number and length of these shortest paths in the outer graph.

Table 3 The progression of upper and lower bounds on the additive error associated to n-node emulators on $O(n)$ edges; current state of the art bounds are highlighted in red. See also [13].

	Upper Bound		Lower Bound	
$O(n)$-size Emulators	$O(n^{1/3+\varepsilon})$	[9]	$\Omega(\log n)$	[23]
	$O(n^{3/11+\varepsilon})$	[8]	$\Omega(n^{1/22})$	[1]
	$\widetilde{O}(n^{1/4})$	[20]	$\Omega(n^{1/18})$	[16]
	$\widetilde{O}(n^{2/9-1/1600<0.222})$	[17]	$\Omega(n^{2/29})$	[19]
	$O(n^{\frac{1}{3+\sqrt{5}}+\varepsilon<0.191})$	[15]	$\Omega(n^{1/14})$	this paper

2.1 The obstacle product framework

Similar to all previous works including [1, 16, 19, 7] on proving stretch lower bounds for linear-sized additive spanners, our construction falls under the *obstacle product* framework introduced in [1]. Any construction under this framework consists of an *outer graph* $G_O = (V_O, E_O)$ and an *inner graph* $G_I = (V_I, E_I)$ where every vertex in the outer graph is replaced by a copy of the inner graph. The desired outer graph should contain a set pairs $P_O \subseteq V_O \times V_O$ often called the *critical pairs* such that the following holds:

1. For each pair $(s, t) \in P_O$, the shortest path from s to t is *unique*. These unique shortest paths connecting between pairs in P_O are often called the *critical paths*.
2. The critical paths have roughly the same length $\Theta(k)$.
3. The critical paths are pairwise edge disjoint.

When we replace each vertex in the outer graph with a copy of the inner graph G_I, we make sure that the critical paths remain the unique shortest paths between their endpoints and pairwise edge-disjoint by attaching each incoming edge and outgoing edge to distinct vertices of the inner graph. Finally, we subdivide the edges originally in G_O into paths of length $\Theta(k)$. Now in the resulting graph denoted as $G_{obs} = (V_{obs}, E_{obs})$ with critical pairs P_{obs}, each critical path between the endpoints in P_{obs} uniquely corresponds to a critical path in G_O and it takes the form of traveling alternatingly between subdivided edges in G_O and paths in G_I. In particular, each critical path travels through $\Theta(k)$ subdivided paths of length $\Theta(k)$, and $\Theta(k)$ inner graph copies.

Now let us see how to show that any sparse spanner on G_{obs} must suffer additive distortion $+\Omega(k)$. The goal is to argue that if lots of edges are missing in the spanner $H \subseteq G_{obs}$ compared to G_{obs}, then there exists some pair $(s, t) \in P_{obs}$ whose shortest path π in H falls into one of the following two cases:

1. If π traverses the same sequence of inner graph copies as the critical path in G_{obs}, then it must use at least one *extra* edge in each inner graph copy compared to the critical path in the original graph due to missing edges. Since the critical path passes through $\Theta(k)$ inner graph copies, the path π must suffer a $+\Omega(k)$ distortion in total.
2. If π traverses a different sequence of inner graph copies, then it must traverse a different set of subdivided paths corresponding to the edges in G_O. Since the critical paths in G_O are the unique shortest paths between its endpoints, π must traverse at least one more subdivided path of length $\Theta(k)$ and thus suffer a $+\Omega(k)$ distortion.

Furthermore, note that the reason behind doing inner graph replacement is that without the inner graphs, subdividing each edge of the outer graph would significantly sparsify the graph so that even a trivial spanner including all the edges would have linear size. Adding the inner graphs helps balance the overall density of the graph so that any linear-sized spanner needs to be nontrivial. Thus, ideally we would want the inner graphs to be dense.

2.2 The outer graph: distance preservers and the alternation product

In this subsection, we review the two key components for the outer graph construction: the distance preserver lower bound graph given in [12] and the alternation product first used in [14].

2.2.1 Distance preserver lower bound graph

Given a graph $G = (V, E)$ and a set of pairs $P \subseteq V \times V$, a distance preserver H is a sparse subgraph of G that preserves the distances for every pair in P *exactly*. Previously, Coppersmith and Elkin [12] obtained a lower bound instance for distance preservers by constructing a large set of vertex pairs with pairwise *edge-disjoint* unique shortest paths that are as long as possible; the union of the edges of these paths is the lower bound instance. Following the intuition outlined in Section 2.1, it is natural to consider using the distance preserver lower bound construction of [12] as the outer graph. From now on, we will abbreviate the term "distance preserver lower bound graph" to "DP LB graph" and the term "Coppersmith-Elkin construction" to "CE construction" for convenience.

Indeed, all prior work uses some version of the CE construction of DP LB graph as the outer graph, and so do we. The CE construction is a geometric construction where the vertex set corresponds to a d-dimensional integer grid $[n]^d$ and edges are added corresponding to a d-dimensional convex set $B_d(r)$ defined to be the vertices of the convex hull of integer points contained in a ball of radius $r > 0$. More specifically, the vertices corresponding to the points $\vec{x}, \vec{y}$ are connected by an edge if $\vec{y} - \vec{x} \in B_d(r)$. Then the critical paths are defined to be the paths corresponding to the straight lines starting from a "start zone" passing through the grid, i.e. the paths that repeatedly take the edge corresponding to the same vector in $B_d(r)$. By convexity of the set $B_d(r)$, one can show that these critical paths are edge-disjoint and they are the unique shortest paths between their endpoints.

Prior to the work of [7], works including [16, 19] all considered a *layered* version of the DP LB graph as the outer graph. Namely the graph contains $\ell + 1$ layers where each layer corresponds to a d-dimensional integer grid $[n]^d$ and edges are added between adjacent layers corresponding to the convex set $B_d(r)$ similarly as defined in [12]. Then the critical paths are defined to be the paths that start in the first layer and end in the last layer that repeatedly take the edge corresponding to the same vector in $B_d(r)$. This layering simplifies the stretch analysis for additive spanner lower bounds because it is easy to argue that all the critical paths have length exactly ℓ and the shortest path should not take any backward edges as it will then need to traverse more layers. However, the layered version resulted in worse bounds compared to the original unlayered version but it was unclear at the time how to analyze an unlayered outer graph. Most recently, Bodwin and Hoppenworth [7] developed a new analysis framework and successfully analyzed an obstacle product graph with a modified version of the unlayered DP LB graph as the outer graph. As a result, they improved the lower bound to $\Omega(n^{1/7})$ from $\Omega(n^{1/10.5})$ where the former remains the current best known lower bound before this work. We use the unlayered outer graph construction in [7] as an ingredient in our construction.

2.2.2 The alternation product

Another important idea that goes in to the outer graph construction is the alternation product first used in [14]. Subsequent works including [1, 16, 19] all use the alternation product in the outer graph construction. Consider two copies G_1, G_2 of the same 2-dimensional layered DP LB graph with $\ell + 1$ layers and convex set $B_2(r)$. Namely, each layer corresponds to the $[n]^2$ grid and the edges correspond to the 2-dimensional convex set $B_2(r)$ of radius r. The original implementation of the alternation product graph G_{alt} used in [14, 1, 16] of G_1 and G_2 is a graph on $2\ell + 1$ layers with each layer corresponding to the 4-dimensional grid $[n]^4$. Each vertex in G_{alt} corresponds the pair (v_1, v_2) where $v_1 \in G_1, v_2 \in G_2$ and the edges are added alternatingly between adjacent layers according to G_1 and G_2, respectively. Specifically, between layer i and $i+1$ for i odd, we connect the vertex $(\vec{x}, \vec{y})$ for $\vec{x}, \vec{y} \in [n]^2$ to $(\vec{x} + \vec{w}, \vec{y})$ for $\vec{w} \in B_2(r)$; for i even, we connect the vertex $(\vec{x}, \vec{y})$ to $(\vec{x}, \vec{y} + \vec{w})$ for $\vec{w} \in B_2(r)$. In other words, G_{alt} keeps track of G_1 using the first two coordinates and G_2 using the last two coordinates. Then a critical path π in G_{alt} corresponds to a pair of critical paths π_1 in G_1 and π_2 in G_2 by taking alternating steps from π_1 and π_2. So the main advantage of the alternation product for us is that it gives an extra product structure over the set of critical paths that we want in our construction.

Unlike in our construction, prior works including [14, 1, 16, 19] apply the alternation product in order to obtain a different relative count between the number of vertices and the number of critical pairs rather than to obtain the extra product structure. However, these changes in parameters are in fact unfavorable to the construction for linear-sized spanner lower bounds. To see this, notice that one can equivalently think of G_{alt} as 4-dimensional CE construction graph using the smaller convex set $\{(\vec{w}_1, \vec{w}_2) \mid \vec{w}_1, \vec{w}_2 \in B_2(r)\}$ instead of $B_4(r)$, which means that G_{alt} has fewer critical pairs (see Section 2.3 for a more detailed discussion). In fact, in [16], Huang and Pettie gave an $\Omega(n^{1/11})$ lower bound construction without the alternation product that improved on their own construction that uses the alternation product which gave a bound of $\Omega(n^{1/13})$ in the same paper. Later in [19], Lu, Vassilevska Wiliams, Wein and Xu improved on the alternation product that reduces the loss in the number of critical pairs compared to the CE construction, thereby obtaining an $\Omega(n^{1/10.5})$ lower bound that improved on the previous best bound of $\Omega(n^{1/11})$. Most recently, Vassilevska Wiliams, Xu and Xu implicitly constructed an alternation product graph in their $O(m)$-shortcut lower bound construction in [22] that asymptotically matches the number critical pairs in the CE construction. Unfortunately, their construction is under a different setting so we cannot directly apply their technique to our construction as a blackbox. However, by isolating a main observation implied in their work, we were able to integrate such an alternation product into our construction (see Section 2.3).

2.3 Our construction: optimal unlayered alternation product and optimal inner graph structure

Our main technical contribution is a linear-sized additive spanner lower bound construction that carefully combines the following ideas:

1. An unlayered DP LB graph as the outer graph, as in [7].
2. An optimal alternation product implicit in [22].
3. An optimal subset DP LB graph as the inner graphs, as motivated in Table 2.

We start with comparing our construction with the previously known lower bound constructions in Table 4.

Table 4 All known lower bound constructions.

Citation	Lower bound	Outer graph	Inner graph
Woodruff [23]	$\Omega(\log n)$	Butterfly	Biclique
Abboud, Bodwin [1]	$\Omega(n^{1/22})$	Layered DP LB + Alt Product	Biclique
Huang, Pettie [16]	$\Omega(n^{1/13})$	Layered DP LB + Alt Product	Biclique
Huang, Pettie [16]	$\Omega(n^{1/11})$	Layered DP LB	Layered DP LB
Lu, Vassilevska W., Wein, Xu [19]	$\Omega(n^{1/10.5})$	Layered DP LB + Improved Alt Product	Biclique
Bodwin, Hoppenworth [7]	$\Omega(n^{1/7})$	Unlayered DP LB	DP LB
This work	$\Omega(n^{3/17})$	Unlayered DP LB + Optimal Alt Product	Subset DP LB

In the following, we will discuss the main components of our construction.

2.3.1 Outer Graph: Unlayered DP LB graph with optimal alternation product

As mentioned in Section 2.2, we would like to be able to apply the implicit alternation product in [22] to unlayered DP LB graphs. By isolating the main idea that one can use the set $\{(x, y, x^2 + y^2) \mid x, y \in [r]\}$ as the convex set in the alternation product graph, we are able to apply the implicit alternation product in [22] on unlayered DP LB graphs successfully after certain modifications (See Section 4.1 for more details). In the following, we give a more detailed discussion of the informal intuition behind why the alternation product we use is more desirable than the alternation product used in prior works including [1, 16, 19].

Recall that in Section 2.2, one may view an alternation product graph as a CE construction with a different convex set that determines the set of edges of the graph. In addition, a vector from the convex set and a vertex in the "start zone" determines a critical path, so we get more critical pairs if we use a larger convex set. More precisely, we want to use a convex set that is "large" with respect to the total number of integer points contained in the convex hull of the set. We recall from [5] that $|B_d(r)| = \Theta(r^{d\cdot\frac{d-1}{d+1}})$. Let us compare the convex sets used in the various alternation product graphs against $B_d(r)$ in the same number of dimension that is scaled to contain the same number of points in its convex hull asymptotically in Table 5. Then we can see from Table 5 that all prior constructions use a convex set that contains less points than the respective $B_d(r)$ while the convex set we use in this work matches the quality of $B_3(r)$, which is optimal in 3-dimensions (see [5] for more details). That is, the construction that we use is as good as the CE construction in 3-dimensions.

Table 5 Comparison between known constructions of the alternation product and the CE construction. The top row in each pair is the corresponding CE construction in the same number of dimensions and scaled to contain the same number of points in its convex hull. The bottom row in each pair indicates the alternation product construction used in the work cited.

Citation	Convex Set Used	Convex Set Size	Convex Hull Size
4-dim [12]	$B_4(r)$	$\Theta(r^{12/5})$	$\Theta(r^4)$
[1, 16]	$\{(\vec{x}, \vec{y}) \mid \vec{x}, \vec{y} \in B_2(r)\}$	$\Theta(r^{4/3})$	$\Theta(r^4)$
3-dim [12]	$B_3(r)$	$\Theta(r^{3/2})$	$\Theta(r^3)$
[19]	$\{(x_1, x_2 + y_1, y_2) \mid \vec{x}, \vec{y} \in B_2(r)\}$	$\Theta(r^{4/3})$	$\Theta(r^3)$
3-dim [12]	$B_3(r^{4/3})$	$\Theta(r^2)$	$\Theta(r^4)$
This work (based on [22])	$\{(x, y, x^2 + y^2) \mid x, y \in [r]\}$	$\Theta(r^2)$	$\Theta(r^4)$

One may wonder why we do not simply use the CE construction as our outer graph. The reason is that the alternation product has extra structure that is crucial for allowing us to use our desired inner graph. The CE construction lacks these properties. We elaborate on this below.

2.3.2 Inner graph: Optimal subset DP LB graph

For our inner graphs, we use the CE construction of subset DP LB graphs in the regime where the pairs $S \times S$ has size $|S| = \Theta(n^{1/4})$ where n denotes the number of vertices in the graph. In fact, this construction is tight in the sense that it has $\Omega(n)$ edges while on the other hand it is known that there exists subset distance preservers of size $O(n)$ for every set of sources S of size $O(n^{1/4})$. So not only are we using an inner graph structure that aligns with the upper bound algorithm as illustrated in Table 2, we are in fact using a tight construction of the desired structure.

The main reason why we are able to use subset DP LB graphs as inner graphs in our construction is that we have an alternation product graph as our outer graph. We discuss below why an alternation product is necessary for using subset DP LB graphs as inner graphs. In the inner graph replacement step under the obstacle product framework, we need to attach the incoming edges and outgoing edges adjacent to a vertex v to vertices in the corresponding inner graph copy so that each critical path passing through v in the outer graph will pass through a unique critical path in the inner graph copy as well. Since the subset DP LB graph has critical pairs of the form $S \times S$ for some subset S of the vertex set, it is required that the critical paths passing through v in the outer graph also be equipped with a product structure. In DP LB graphs, we have no such product structure over the critical paths. However, notice that applying an alternation product would exactly give us a product structure over the critical paths as desired.

3 Preliminaries

We use the following notations:

- We use $\texttt{Conv}(\cdot)$ to denote the convex hull of a set.
- We use $\langle \cdot, \cdot \rangle$ to denote the standard Euclidean inner product, $\| \cdot \|$ the Euclidean norm, and $\texttt{proj}_{\vec{w}}(\cdot)$ the Euclidean scalar projection onto $\vec{w}$.
- We use $[x]$, where x is a positive integer, to denote the set $\{1, \dots, x\}$.

4 Outer Graph G_O

The goal of this section will be to construct the outer graph G_O of our additive spanner and emulator lower bound constructions. The key properties of G_O are summarized in Theorem 6.

▶ **Theorem 6** (Properties of Outer Graph). *For any $a, r > 0 \in \mathbb{Z}$, there exists a graph $G_O(a, r) = (V_O, E_O)$ with a set Π_O of critical paths in G_O that has the following properties:*

1. *The number of nodes, edges, and critical paths in G_O is:*

$$\begin{aligned} |V_O| &= \Theta(a^3 r), \\ |E_O| &= \Theta(a^3 r^2), \\ |\Pi_O| &= \Theta(a^2 r^4). \end{aligned}$$

2. *Every critical path $\pi \in \Pi_O$ is a unique shortest path in G_O of length at least $|\pi| \geq \frac{a}{4r}$.*
3. *Every pair of distinct critical paths $\pi_1, \pi_2 \in \Pi_O$ intersect on at most two nodes.*
4. *Every edge $e \in E_O$ lies on some critical path in Π_O.*

The rest of the section is devoted to constructing the graph $G_O(a, r)$ and paths Π_O that satisfy Theorem 6.

4.1 Convex Set of Vectors

Before specifying the construction of the graph G_O, we begin by specifying our construction of a set of vectors $W \subseteq \mathbb{R}^3$ that is crucial to the construction of G_O. Set W will be parameterized by a positive integer r, i.e., $W = W(r)$. The vectors in W will satisfy a certain strict convexity property that we will use to ensure the unique shortest paths property of paths Π_O in G_O.

▶ **Definition 7** ($W(r)$). *Given a positive integer r, let*

$$W_1(r) := \{(x, 0, x^2) \mid x \in \{r/2, \dots, r\}\} \quad \textit{and} \quad W_2(r) := \{(0, y, y^2) \mid y \in \{r/2, \dots, r\}\}.$$

We define $W(r)$ to be the sumset

$$W(r) := W_1(r) + W_2(r) = \{(x, y, x^2 + y^2) \mid x, y \in \{r/2, \dots, r\}\}.$$

We now verify that sets of vectors $W_1(r), W_2(r), W(r)$ have the necessary convexity property to ensure that graph G_O has unique shortest paths (Property 2 of Theorem 6). The convexity property of W stated in Lemma 8 is roughly similar to the notion of "strong convexity" in [7], but is in fact stronger.

▶ **Lemma 8** (Convexity property). *Let W_1, W_2, W be the sets defined in Definition 7 for some positive integer r. Let W' be the set*

$$W' = W \cup (-W) \cup (W_1 - W_2) \cup (W_2 - W_1).$$

Then each vector $\vec{w} \in W$ is an extreme point of the convex hull $\texttt{Conv}(W')$ of W'.

We omit the proof as it simply follows from checking that every vector in W' is an extreme point of $\texttt{Conv}(W')$.

4.2 Construction of G_O

Let $a, r > 0 \in \mathbb{Z}$ be the input parameters for our construction of outer graph $G_O = (V_O, E_O)$. Let $W_1 = W_1(r)$, $W_2 = W_2(r)$, and $W = W(r)$ be the sets of vectors constructed in Definition 7 and parameterized by our choice of r.

Vertex Set V_O

- Our vertex set V_O will correspond to two copies of integer points arranged in a grid in $\mathbb{R}^3$. These two copies will be denoted as V_O^L and V_O^R. For a point $p \in \mathbb{R}^3$, we will use p_L to denote the copy of point p in V_O^L, and p_R to denote the copy of point p in V_O^R. Likewise, for a set of points $P \subseteq \mathbb{R}^3$, we will use P_L to denote the copy of set P in V_O^L, and P_R to denote the copy of set P in V_O^R. Then we define V_O^L and V_O^R as:

  $$V_O^L = ([a] \times [a] \times [ar])_L, \qquad \text{and} \qquad V_O^R = ([a] \times [a] \times [ar])_R.$$

  When denoting a node v_L or v_R in V_O, we will drop the subscript and simply denote this node as v when its membership in V_O^L and V_O^R is clear from the context or otherwise irrelevant.

Edge Set E_O

- The edges E_O in G_O will pass between V_O^L and V_O^R, so that $E_O \subseteq V_O^L \times V_O^R$.
- Just as the nodes in V_O are integer points in $\mathbb{R}^3$, we will identify the edges in E_O with integer vectors in $\mathbb{R}^3$. Specifically, for each edge (x_L, y_R) in E_O, we identify $x_L \to y_R$ with the vector $y - x \in \mathbb{R}^3$. Note that $y - x$ corresponds to the orientation $x_L \to y_R$ of edge (x_L, y_R); we would use vector $x - y \in \mathbb{R}^3$ to denote $y_R \to x_L$.
- For each node $v_L \in V_O^L$ and each vector $\vec{w} \in W_1$, if $(v + \vec{w})_R \in V_O^R$, then add edge $(v_L, (v + \vec{w})_R)$ to E_O. Likewise, for each node $v_R \in V_O^R$ and each vector $\vec{w} \in W_2$, if $(v + \vec{w})_L \in V_O^L$, then add edge $(v_R, (v + \vec{w})_L)$ to E_O.

Critical Paths Π_O

- Let $S \subseteq \mathbb{R}^3$ denote the set of points $S = [a] \times [a] \times [r^2/8]$.
- Let s be a point in S. Additionally, let $\vec{w}_1$ be a vector in W_1, and let $\vec{w}_2$ be a vector in W_2, where $\vec{w}_1 + \vec{w}_2 \in W$.[4] If $s + \vec{w}_1 \notin S$, then we define a corresponding path in G_O starting from $s_L \in S_L$ as

$$s_L \to (s + \vec{w}_1)_R \to (s + \vec{w}_1 + \vec{w}_2)_L \to (s + 2\vec{w}_1 + \vec{w}_2)_R \to (s + 2\vec{w}_1 + 2\vec{w}_2)_L \to \cdots \to (s + i \cdot \vec{w}_1 + i \cdot \vec{w}_2)_L,$$

  where i is the largest integer i such that node $(s + i \cdot \vec{w}_1 + i \cdot \vec{w}_2)_L \in V_O^L$. Let $t = (s + i \cdot \vec{w}_1 + i \cdot \vec{w}_2)_L$ be the endpoint of this path, and add this $s \rightsquigarrow t$ path to our set of critical paths Π_O.
- Note that every critical path $\pi \in \Pi_O$ constructed this way is uniquely specified by a start node $s_L \in S_L$ and vectors $\vec{w}_1 \in W_1$ and $\vec{w}_2 \in W_2$.
- For every critical path $\pi \in \Pi_O$ where $|\pi| < \frac{a}{4r}$, remove π from Π_O.

As a final step in our construction of G_O, we remove all edges in G_O that do not lie on some critical path $\pi \in \Pi_O$.

4.3 Properties of G_O

We will now verify that G_O and Π_O satisfy the properties specified in Theorem 6 via the following claims. We omit the proofs as most of them follow from definition and Lemma 8.

▷ Claim 9. Every critical path $\pi \in \Pi_O$ is the unique shortest path between its endpoints in G_O. Moreover, $|\pi| \geq \frac{a}{4r}$.

▷ Claim 10. Every pair of distinct critical paths $\pi_1, \pi_2 \in \Pi_O$ can intersect on either a single vertex or a single edge.

▷ Claim 11. Every edge in G_O is used by at most $r/2$ critical paths $\pi \in \Pi_O$.

▷ Claim 12. The number of nodes, edges, and critical paths in $G_O(a, r)$ is:

$$|V_O| = \Theta(a^3 r),$$
$$|E_O| = \Theta(a^3 r^2),$$
$$|\Pi_O| = \Theta(a^2 r^4).$$

Proof of Theorem 6. Note that graph G_O and critical paths Π_O satisfy Properties 1, 2, and 3 of Theorem 6 by Claim 9, Claim 10, and Claim 12. Moreover, Property 4 of Theorem 6 follows immediately from the final step in our construction of G_O. This completes the proof of Theorem 6. ◀

5 Spanner Lower Bound Construction

In this section we present our lower bound construction for additive spanners. This construction will have a similar structure to the obstacle product graph G constructed for our emulator lower bound, but with several complications. We now describe these modifications to the obstacle product argument:

[4] Note that $\vec{w}_1 + \vec{w}_2 \in W$ for all $\vec{w}_1 \in W_1$ and $\vec{w}_2 \in W_2$, since $W = W_1 + W_2$. However, in Section 5.1, we will modify W so that W is a strict subset of $W_1 + W_2$, which will make this requirement relevant.

- **Convex Sets W_1, W_2, and W.** In Section 5.1, we modify the convex sets of vectors W_1, W_2, and W_3 that we defined in Section 4. The purpose of this modification is technical, but it has to do with the projection argument we employ in our analysis. Our new convex sets of vectors $W_1(r,c), W_2(r,c), W(r,c)$ will now be parameterized by an additional integer $c > 0$. These new sets of vectors will roughly resemble the outer graph vectors in Lemma 8 of [7] and will play a similar role in our analysis of G.
- **Inner Graph G_I.** We choose our inner graphs to be the sourcewise distance preserver lower bound graphs constructed in [12]. Lemma 19 specifies the exact properties of these new inner graphs G_I that we require in our analysis. See Subsections 1.1 and 2.2 for an overview of why we make this design choice.

5.1 Modifying Convex Sets $W_1, W_2,$ and W in G_O

In this section, we modify the definitions of the convex sets of vectors W_1, W_2, and W used to construct outer graph G_O. Let $r, c > 0 \in \mathbb{Z}$ be the input parameters to our convex sets of vectors W_1, W_2, and W.

We define I_i be the interval

$$I_i := \left[\frac{r}{2} + \frac{(2i-2)\cdot r}{4c}, \quad \frac{r}{2} + \frac{(2i-2)\cdot r}{4c} + \frac{r}{16c^3}\right],$$

for $i \in [1, c]$. The following claim is immediate from the definition of intervals I_i.

▷ Claim 13. Intervals $\{I_i\}_{i\in[1,c]}$ satisfy the following properties:
- $I_i \subseteq [r/2, r]$,
- $|I_i| = \frac{r}{16c^3}$
- if $x, y \in I_i$, then $|x - y| \leq \frac{r}{16c^3}$, and
- if $x \in I_i$ and $y \in I_j$, where $i \neq j$, then $|x - y| \geq r/(2c)$.

We will use intervals $\{I_i\}_{i\in[1,c]}$ to construct our sets of vectors $W_1(r,c)$ and $W_2(r,c)$.

▶ **Definition 14** ($W_1(r,c)$ and $W_2(r,c)$). *Let r, c be positive integers. We define $W_1(r,c)$ and $W_2(r,c)$ as*

$$W_1(r,c) := \left\{(x, 0, x^2) \mid x \in I_i, i \in [1,c]\right\} \quad \textit{and} \quad W_2(r,c) := \left\{(0, y, y^2) \mid y \in I_i, i \in [1,c]\right\}.$$

Now we partition the vectors in $W_1(r,c)$ into c sets $\mathcal{S}_1^1, \ldots, \mathcal{S}_c^1$ we call *stripes*. We define the ith stripe $\mathcal{S}_i^1$ of $W_1(r,c)$ as $\{(x, 0, x^2) \mid x \in I_i\}$. Likewise, we define the ith stripe $\mathcal{S}_i^2$ of $W_2(r,c)$ as $\{(0, y, y^2) \mid y \in I_i\}$. The key properties of our stripes are summarized in Claim 15, which follows immediately from Claim 13.

▷ Claim 15. Stripes $\{\mathcal{S}_i^1\}_{i\in[1,c]}$ satisfy the following properties:
- $\mathcal{S}_i^1 \subseteq [r/2, r]$,
- $|\mathcal{S}_i^1| = \frac{r}{16c^3}$,
- if $(x, 0, x^2), (y, 0, y^2) \in \mathcal{S}_i^1$, then $|x - y| \leq \frac{r}{16c^3}$, and
- if $(x, 0, x^2) \in \mathcal{S}_i^1$ and $(y, 0, y^2) \in \mathcal{S}_j^1$, where $i \neq j$, then $|x - y| \geq \frac{r}{2c}$.

Moreover, stripes $\{\mathcal{S}_i^2\}_{i\in[1,c]}$ satisfy analogous properties.

Roughly, Claim 15 states that vectors in the same stripe in $W_1(r,c)$ or $W_2(r,c)$ are "close" to each other in some sense, and vectors in different stripes in $W_1(r,c)$ and $W_2(r,c)$ are "far" from each other in some sense. This notion of partitioning a set of vectors into stripes satisfying these properties was introduced in the spanner lower bound construction of [7]. We are now ready to define our set of vectors $W(r,c)$.

▶ **Definition 16** ($W(r,c)$). *Let r, c be positive integers. Unlike in Definition 7, we will define $W(r,c)$ to be a subset of the sumset $W_1(r,c) + W_2(r,c)$. In particular, for $\vec{w}_1 \in W_1(r,c)$ and $\vec{w}_2 \in W_2(r,c)$, we only add $\vec{w}_1 + \vec{w}_2$ to $W(r,c)$ if $\vec{w}_1$ and $\vec{w}_2$ share the same stripe index $i \in [1,c]$. Formally,*

$$W(r,c) := \left\{(x, y, x^2 + y^2) \mid x, y \in I_i, i \in [1,c]\right\} \subset W_1(r,c) + W_2(r,c).$$

The following claim is immediate from the definitions of $W_1(r,c)$, $W_2(r,c)$ and $W(r,c)$.

▷ Claim 17. Sets $W_1(r,c)$, $W_2(r,c)$ and $W(r,c)$ satisfy the following properties:

- $|W_1(r,c)| = |W_2(r,c)| = \Theta\left(\frac{r}{c^2}\right)$,
- $|W(r,c)| = \Theta\left(\frac{r^2}{c^5}\right)$,
- $W_1(r,c) \subset W_1(r)$, $W_2(r,c) \subset W_2(r)$, and $W(r,c) \subset W(r)$, so $W(r,c)$ satisfies the convexity property stated in Lemma 8.

We modify the construction of outer graph G_O in Section 4 by replacing sets $W_1(r), W_2(r)$, and $W(r)$ defined in Section 4.1 with the new sets $W_1(r,c), W_2(r,c)$, and $W(r,c)$. Note that our new choice of sets $W_1(r,c)$ and $W_2(r,c)$ changes the the set of vectors E_O in G_O, while our new choice of set $W(r,c)$ changes the set of critical paths Π_O (see Footnote 4).

By inserting convex sets $W_1(r,c), W_2(r,c)$, and $W(r,c)$ into G_O in place of the sets $W_1(r), W_2(r)$, and $W(r)$, we obtain the following theorem about our modified outer graph $G_O = G_O(a,r,c)$.

▶ **Theorem 18** (Properties of Modified Outer Graph). *For any $a, r, c > 0 \in \mathbb{Z}$, there exists a graph $G_O(a,r,c) = (V_O, E_O)$ with a set of critical paths Π_O that has the following properties:*

1. *The number of nodes, edges, and critical paths in G_O is:*
$$|V_O| = \Theta(a^3 r), \quad |E_O| = \Theta\left(\frac{a^3 r^2}{c^2}\right), \quad |\Pi_O| = \Theta\left(\frac{a^2 r^4}{c^5}\right).$$
2. *Every critical path $\pi \in \Pi_O$ is a unique shortest path in G_O of length at least $|\pi| \geq \frac{a}{4r}$.*
3. *Every pair of distinct critical paths π_1 and π_2 intersect on at most two nodes.*
4. *Every edge $e \in E_O$ lies on some critical path in Π_O.*

Just like in the original construction of G_O in Section 4, every critical path $\pi \in \Pi_O$ corresponds to a unique vector $\vec{w} \in W(r,c)$. Specifically, by the definition of $W_1(r,c), W_2(r,c)$, and $W(r,c)$, path π is constructed using vectors $\vec{w}_1 \in W_1(r,c)$ and $\vec{w}_2 \in W_2(r,c)$, where

- $\vec{w} = \vec{w}_1 + \vec{w}_2$, and
- $\vec{w}_1 \in \mathcal{S}_i^1$ and $\vec{w}_2 \in \mathcal{S}_i^2$, for some $i \in [1,c]$.

Critically, $\vec{w}_1$ and $\vec{w}_2$ both lie in the ith stripe $\mathcal{S}_i^1$ and $\mathcal{S}_i^2$, respectively.

5.2 Inner Graph G_I

In this subsection, we formally state the properties of the family of graphs we choose for our inner graphs G_I when constructing the obstacle product graph G. We will choose our inner graphs to be the sourcewise distance preserver lower bound graphs constructed in [12]. Lemma 19 formally captures the exact properties of this family of graphs that we need for spanner lower bound argument. We will defer our proof of Lemma 19 to the appendix, as it largely follows from the proof of Theorem 5.10 in [12].

▶ **Lemma 19** (cf. Theorem 5.10 of [12]). *For any $a, c > 0 \in \mathbb{Z}$, there exists a graph $G_I(a,c) = (V_I, E_I)$ with a set $S_I \subseteq V_I$ of sources, a set $T_I \subseteq V_I$ of sinks, and a set $P_I \subseteq S_I \times T_I$ of critical pairs that has the following properties:*

1. *The number of nodes, edges, sources, sinks, and critical pairs in* G_I *is:*

$$|V_I| = \Theta(a^2),$$
$$|E_I| = \Theta(a^2 c),$$
$$|S_I| = \Theta(a^{1/2} c^{11/4}),$$
$$|T_I| = \Theta(a^{1/2} c^{11/4}),$$
$$|P_I| = \Theta(ac^{5/2}).$$

2. *Every path* $\pi_{s,t}$*, where* $(s,t) \in P_I$*, contains* $\Theta(a/c^{3/2})$ *edges that do not lie on any other path* $\pi_{s',t'}$*, where* $(s',t') \in P_I$*.*
3. *For every source* $s \in S_I$ *and sink* $t \in T_I$*, the distance between* s *and* t *in* G_I *satisfies the following:*

$$dist_{G_I}(s,t) = \Theta(ac^{1/2}).$$

4. *The set of sources* S_I *can be partitioned into* $b = \Theta(c^3)$ *sets* $S_I^1, \dots, S_I^b$*, where* $|S_I^i| = \Theta(a^{1/2} c^{-1/4})$ *for all* $i \in [b]$*. Let* $T_I^i = \{t \in T_I \mid (S_I^i \times \{t\}) \cap P_I \neq \emptyset\}$ *be the set of all sinks that belong to a critical pair with a source in* S_I^i*. Then for all* $i \in [b]$ *the following properties hold:*
   - $|T_I^i| = \Theta(a^{1/2} c^{-1/4})$ *for all* $i \in [b]$*,*
   - $S_I^i \times T_I^i \subseteq P_I$*, and*
   - *for all* $(s,t) \in P_I$ *such that* $s \in S_I^i$ *and* $t \in T_I^i$*,*

$$dist_{G_I}(s,t) \leq dist_{G_I}(S_I^i, T_I^i),$$

   where $dist_{G_I}(S_I^i, T_I^i)$ *denotes the minimum distance between* S_I^i *and* T_I^i *in* G_I*.*

5.3 Construction of Obstacle Product Graph G

Let $a, r, c > 0 \in \mathbb{Z}$ be the input parameters of an instance of outer graph $G_O = G_O(a, r, c)$. Let $W_1 = W_1(r, c)$, $W_2 = W_2(r, c)$, and $W = W(r, c)$ be the sets of vectors constructed in Section 5.1. Additionally, let $a', c' > 0 \in \mathbb{Z}$ be the input parameters of an instance of inner graph $G_I = G_I(a', c')$. We will specify the precise values of a, r, c, a', and c' later, as needed. Roughly, our choices of parameters a, r, and a' will grow with the size of our final graph G, while parameters c and c' will be (sufficiently large) integer constants.

We will construct our final graph G by performing the obstacle product. The obstacle product is performed in two steps: the edge subdivision step and the inner graph replacement. In the inner graph replacement step, we will need to carefully define two functions, $\phi_1 : W_1 \mapsto S_I \times T_I$ and $\phi_2 : W_2 \mapsto S_I \times T_I$ between vectors in W_1 and W_2 and pairs of nodes in $S_I \times T_I$ in inner graph G_I.

Edge Subdivision

We subdivide each edge in G_O into a path of length ψ. Denote the resulting graph as G'_O. For any edge $e = (u, v) \in E_O$, let P_e denote the resulting $u \rightsquigarrow v$ path of length ψ. We will take $\psi = \Theta\left(\frac{r^3}{c^{29/3}}\right)$.

Inner Graph Replacement

We now perform the inner graph replacement step of the obstacle product.

- For each node v in $V(G'_O)$ originally in G_O, replace v with a copy of G_I. We refer to this copy of G_I as G_I^v. Likewise, refer to the sources and sinks S_I and T_I in G_I^v as S_I^v and T_I^v.
- After applying the previous operation, the endpoints of the subdivided paths P_e in G'_O no longer exist in the graph. If $e = (u, v) \in E_O$, then P_e will have endpoints u and v. We will replace the endpoints u and v of P_e with nodes in G_I^u and G_I^v, respectively.
- In order to precisely define this replacement operation, it will be helpful to define two functions, $\phi_1 : W_1 \mapsto S_I \times T_I$ and $\phi_2 : W_2 \mapsto S_I \times T_I$. For ease of understanding, we will first assume the existence functions ϕ_1 and ϕ_2. We will specify our choices of ϕ_1 and ϕ_2 later.
- Let $e = (u, v) \in E_O$. If $v - u \in W_1$, then let $\phi_1(v - u) = (x, y) \in S_I \times T_I$. We will replace the endpoints u and v of P_e with nodes $y \in T_I^u$ in G_I^u and $x \in S_I^v$ in G_I^v, respectively. Otherwise, if $v - u \in W_2$, then let $\phi_2(v - u) = (x, y) \in S_I \times T_I$, and replace the endpoints u and v of P_e with nodes $y \in T_I^u$ in G_I^u and $x \in S_I^v$ in G_I^v, respectively. We repeat this operation for each $e \in E_O$ to obtain the obstacle product graph G.
- Note that after performing the previous operation, every subdivided path P_e, where $e = (u, v)$, will have a start node in T_I^u and an end node in S_I^v. We will use t_e to denote the start node of P_e in T_I^u and s_e to the end node of P_e in S_I^v.

This completes the construction of the obstacle product graph G (up to defining functions ϕ_1 and ϕ_2).

Defining functions ϕ_1 and ϕ_2

Let $\mathcal{S}_1^1, \dots, \mathcal{S}_c^1$ be the stripes of W_1, and let $\mathcal{S}_1^2, \dots, \mathcal{S}_c^2$ be the stripes of W_2. Let $S_I^1, \dots, S_I^b$ and $T_I^1, \dots, T_I^b$ be the partition of sources S_I and sinks T_I as described in Lemma 19, where $b = \Theta(c'^4)$. In order to construct our desired functions, we will require the following relations to hold:

- $b \geq c$,
- $|S_I^i| \geq |\mathcal{S}_i^j|$, for all $i \in [1, c]$ and $j \in \{1, 2\}$, and
- $|T_I^i| \geq |\mathcal{S}_i^j|$, for all $i \in [1, c]$ and $j \in \{1, 2\}$.

This can be achieved by setting

$$c' = \Theta(c^{1/3}) \quad \text{and} \quad a' = \Theta\left(\frac{r^2}{c^{35/6}}\right),$$

using the fact that $|\mathcal{S}_i^j| = \Theta(r/c^3)$, $|S_I^i| = \Theta(a'^{1/2}c'^{-1/4})$, and $|T_I^i| = \Theta(a'^{1/2}c'^{-1/4})$ by Claim 15 and Lemma 19.

We are now ready to define our functions ϕ_1 and ϕ_2. Let $\vec{w}_{i,j}^k$ denote the jth vector of $\mathcal{S}_i^k$, where $i \in [1, c]$, $j \in [1, |\mathcal{S}_i^k|]$, and $k \in \{1, 2\}$. Let s_j^i denote the jth node of S_I^i, where $i \in [1, c]$ and $j \in [1, |S_I^i|]$. Likewise, let t_j^i denote the jth node of T_I^i, where $i \in [1, c]$ and $j \in [1, |T_I^i|]$. We define ϕ_1 and ϕ_2 as follows:

$$\phi_1(\vec{w}_{i,j}^1) = (s_j^i, t_j^i) \qquad \text{and} \qquad \phi_2(\vec{w}_{i,j}^2) = (s_j^i, t_j^i) \qquad \text{for } i \in [1, c],\ j \in [1, |\mathcal{S}_j^1|].$$

The key properties of functions ϕ_1 and ϕ_2 are summarized in Claim 20.

▷ Claim 20. Functions ϕ_1 and ϕ_2 satisfy the following properties:

1. Our choice of ϕ_1 and ϕ_2 imply that for each node $u \in S_I \cup T_I$ in an inner graph copy G_I^v, there is at most one subdivided path P_e incident to u in G.
2. $\phi_k(\mathcal{S}_i^1) \subseteq S_I^i \times T_I^i$ for $k \in \{1, 2\}$ and $i \in [1, c]$, where $\phi_k(\mathcal{S}_i^1)$ denotes the image of $\mathcal{S}_i^1$ under ϕ_k.

Critical Paths Π

- Fix a critical path $\pi_O \in \Pi_O$ with associated vectors $\vec{w}_1 \in W_1$ and $\vec{w}_2 \in W_2$.
- By our construction of G_O, there exists $\vec{w} \in W$ such that $\vec{w} = \vec{w}_1 + \vec{w}_2$ (see Footnote 4). Then by the construction of W there exists some index $i \in [1, c]$ such that every edge $(u, v) \in \pi_O$ satisfies $v - u \in \{\vec{w}_1, \vec{w}_2\} \subseteq \mathcal{S}_i^1 \cup \mathcal{S}_i^2$. Let $\chi \in [1, c]$ denote this index.
- Let e_i denote the ith edge of π_O for $i \in [1, k]$. Note that by Property 2 of Claim 20, it follows that $s_{e_i} \in S_I^\chi$ and $t_{e_i} \in T_I^\chi$ for $i \in [1, k]$. Then by Property 5 of Lemma 19, $(s_{e_i}, t_{e_{i+1}}) \in P_I$. By Property 2 of Lemma 19, path $\pi_{s_{e_i}, t_{e_{i+1}}}$ is a unique shortest $s_{e_i} \rightsquigarrow t_{e_{i+1}}$ path in G_I.
- We now define a corresponding path π in G:

  $$\pi = P_{e_1} \circ \pi_{s_{e_1}, t_{e_2}} \circ P_{e_2} \circ \cdots \circ P_{e_{k-1}} \circ \pi_{s_{e_{k-1}}, t_{e_k}} \circ P_{e_k},$$

  Note that if $e_i = (x, y)$ and $e_{i+1} = (y, z)$, then $\pi_{s_{e_i}, t_{e_{i+1}}}$ corresponds to the unique shortest path between $s_{e_i} \in S_I^\chi$ and $t_{e_i} \in T_I^\chi$ in inner graph copy G_I^y. We add path π to our set of critical paths Π.
- We repeat this process for all critical paths in Π_O to obtain our set of critical paths Π in G. Each critical path $\pi \in \Pi$ is uniquely constructed from a critical path $\pi_O \in \Pi_O$, so $|\Pi| = |\Pi_O|$. We will use $\phi : \Pi \mapsto \Pi_O$ to denote the bijection between Π and Π_O implicit in the construction.

As the final step in our construction of obstacle product graph G, we remove all edges in G that do not lie on some critical path $\pi \in \Pi$. Note that this will only remove edges in G that are inside copies of the inner graph G_I. Theorem 21 summarizes some of the key properties of obstacle product graph G. The proof of Theorem 21 follows from straightforward calculations and arguments similar to those in Section 5.2 in the full paper.

▶ **Theorem 21** (Properties of Obstacle Product Graph). *For any $a, r, c > 0 \in \mathbb{Z}$, there exists a graph $G(a, r, c) = (V, E)$ with a set of critical paths Π that has the following properties:*

1. *The number of nodes, edges, and critical paths in G is:*

$$|V| = \Theta(a^3 r^5 c^{-23/2}),$$
$$|E| = \Theta\left(c^{1/4} \cdot |V|\right),$$
$$|\Pi| = \Theta\left(\frac{a^2 r^4}{c^5}\right).$$

2. *Every path $\pi \in \Pi$ that passes through inner graph copy G_I^v contains $\Theta(a'/c'^{3/2})$ edges that do not lie on any other path $\pi' \in \Pi$, for all $v \in V_O$.*

For the full analysis of our spanner construction, please refer to the full version of the paper on ArXiv.

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.
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.
3. Donald Aingworth, Chandra Chekuri, Piotr Indyk, and Rajeev Motwani. Fast estimation of diameter and shortest paths (without matrix multiplication). *SIAM Journal on Computing*, 28(4):1167–1181, 1999.

4 Ingo Althöfer, Gautam Das, David Dobkin, Deborah Joseph, and José Soares. On sparse spanners of weighted graphs. *Discrete & Computational Geometry*, 9(1):81–100, 1993.

5 Imre Bárány and David G. Larman. The convex hull of the integer points in a large ball. *Math. Ann.*, 312(1):167–181, 1998. doi:10.1007/s002080050217.

6 Surender Baswana, Telikepalli Kavitha, Kurt Mehlhorn, and Seth Pettie. Additive spanners and (α, β)-spanners. *ACM Transactions on Algorithms (TALG)*, 7(1):1–26, 2010.

7 Greg Bodwin and Gary Hoppenworth. New additive spanner lower bounds by an unlayered obstacle product. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 – November 3, 2022*, pages 778–788. IEEE, 2022. doi:10.1109/FOCS54457.2022.00079.

8 Greg Bodwin and Virginia Vassilevska Williams. Better distance preservers and additive spanners. *ACM Transactions on Algorithms (TALG)*, 17(4):1–24, 2021.

9 Gregory Bodwin and Virginia Vassilevska Williams. Very sparse additive spanners and emulators. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pages 377–382, 2015.

10 Béla Bollobás, Don Coppersmith, and Michael Elkin. Sparse distance preservers and additive spanners. *SIAM Journal on Discrete Mathematics*, 19(4):1029–1055, 2005.

11 Shiri Chechik. New additive spanners. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, pages 498–512. SIAM, 2013.

12 Don Coppersmith and Michael Elkin. Sparse sourcewise and pairwise distance preservers. *SIAM J. Discret. Math.*, 20(2):463–501, February 2006. doi:10.1137/050630696.

13 Dorit Dor, Shay Halperin, and Uri Zwick. All-pairs almost shortest paths. *SIAM Journal on Computing*, 29(5):1740–1759, 2000.

14 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.

15 Gary Hoppenworth. Simple linear-size additive emulators. In *2024 Symposium on Simplicity in Algorithms (SOSA)*, pages 1–8. SIAM, 2024.

16 Shang-En Huang and Seth Pettie. Lower Bounds on Sparse Spanners, Emulators, and Diameter-reducing shortcuts. In David Eppstein, editor, *16th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2018)*, volume 101 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 26:1–26:12, Dagstuhl, Germany, 2018. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SWAT.2018.26.

17 Shimon Kogan and Merav Parter. New additive emulators. In *50th International Colloquium on Automata, Languages, and Programming (ICALP 2023)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023.

18 Kevin Lu. *New methods for approximating shortest paths*. PhD thesis, Massachusetts Institute of Technology, 2019.

19 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 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3311–3331, 2022.

20 Seth Pettie. Low distortion spanners. *ACM Transactions on Algorithms (TALG)*, 6(1):1–22, 2009.

21 Zihan Tan and Tianyi Zhang. Almost-optimal sublinear additive spanners. *arXiv preprint*, 2023. arXiv:2303.12768.

22 Virginia Vassilevska Williams, Yinzhan Xu, and Zixuan Xu. Simpler and higher lower bounds for shortcut sets. In David P. Woodruff, editor, *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024, Alexandria, VA, USA, January 7-10, 2024*, pages 2643–2656. SIAM, 2024. doi:10.1137/1.9781611977912.94.

23 David P Woodruff. Lower bounds for additive spanners, emulators, and more. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 389–398. IEEE, 2006.