A General Reduction from Near-Additive Emulators to Near-Exact Hopsets

Julian Aeri
Department of Computer Science, University of Salzburg, Austria

Sebastian Forster
Department of Computer Science, University of Salzburg, Austria

Mara Grilnberger
Department of Computer Science, University of Salzburg, Austria

Abstract

Graph emulators and hopsets are two fundamental concepts for distance approximation. For a given graph G, an (α, β)-emulator is a sparse graph on the same vertex set that preserves the distances of G up to a multiplicative stretch α and additive stretch β. In contrast, an (α, β)-hopset is a set of additional edges that, when added to G, ensures that distances can be approximated up to a multiplicative stretch α, using paths containing at most β edges. When $\alpha = 1+\epsilon$ for arbitrarily small $\epsilon > 0$, these structures are known as near-additive emulators and near-exact hopsets, respectively. Prior work showed that there is a remarkable similarity between the constructions and guarantees of these two objects. In their survey on this topic, Elkin and Neiman [Bull. EATCS 130, 2020] explicitly asked whether one can obtain a general reduction between near-additive emulators and near-exact hopsets. Following that, Kogan and Parter [FOCS, 2022] provided a general reduction from hopsets to emulators and spanners.

In this paper, we address the reverse direction and show that any construction for a near-additive emulator for undirected unweighted graphs can be leveraged as a black box to construct a hopset for an undirected weighted graph with comparable size, stretch, and a hopbound comparable to the emulator's additive stretch. Specifically, we show that any algorithm that constructs a $(1 + \epsilon', \beta)$-emulator, with $0 \leq \epsilon' \leq 1$ and $\beta \geq 1$, of size $S_{\mathcal{A}}(n, \epsilon', \beta)$, can be used to obtain a $(1+\epsilon, O(\frac{\beta^2}{\epsilon^2} \ln(\frac{n}{\epsilon})))$-hopset of size $O((S_{\mathcal{A}}(n+m\frac{\beta}{\epsilon^2}, \frac{\epsilon}{294}, \beta)\frac{1}{\epsilon}+n)\ln(\frac{n}{\epsilon}))$, for any $0 < \epsilon \leq 1$. Therefore, our reduction answers the question of Elkin and Neiman [Bull. EATCS 130, 2020] for sparse graphs and further advances the understanding of the formal connection between these two structures. Designing a reduction resulting in a hopset size that does not depend on m remains an intriguing open question.

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

Keywords and phrases hopsets, shortest paths, emulator-to-hopset-reduction

Digital Object Identifier 10.4230/LIPIcs.ESA.2026.155

Related Version *Full Version*: `https://arxiv.org/abs/2607.07190`

Funding This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 947702).
Mara Grilnberger: This publication has been supported by the EXDIGIT (Excellence in Digital Sciences and Interdisciplinary Technologies) project, funded by Land Salzburg under grant number 20204-WISS/263/6-6022.

1 Introduction

Given a graph G, a hopset is a set of additional edges that, when added to G, ensures that for any pair of vertices, an approximate shortest path exists with a bounded number of edges (or hops). They were first formally introduced by Cohen [10] as a tool for efficient parallel computation of approximate shortest paths. And since then they have been central in parallel algorithms [10, 26, 13, 17, 7, 8, 6], dynamic distance maintenance [3, 20, 18, 25], and distributed shortest paths computations [19, 13, 9].

▶ **Definition 1** (h-hop distance). *Given a graph $G = (V, E)$. For a pair of vertices $u, v \in V$, the h-hop distance $\mathrm{d}_G^h(u, v)$ denotes the weight of the shortest path from u to v in G that contains at most h edges.*

▶ **Definition 2** ((α, β)-hopset). *Given a graph $G = (V, E)$, a multiplicative stretch $\alpha \geq 1$ and a hopbound $\beta \in \mathbb{N}$, a (α, β)-hopset $F \subseteq V \times V$ is a set of weighted edges, such that*

$$\forall u, v \in V \quad \mathrm{d}_G(u, v) \leq \mathrm{d}_H^\beta(u, v) \leq \alpha \cdot \mathrm{d}_G(u, v)$$

where $H = (V, E \cup F)$ is the hopset graph. A hopset is called exact if $\alpha = 1$, and near-exact if $\alpha = 1 + \epsilon$ for arbitrarily small $\epsilon > 0$.

Two other fundamental structures for approximating distances are spanners and emulators. Rather than bounding the number of hops, they instead aim to sparsify the graph, while preserving approximate pairwise distances. Formally, an emulator is defined as follows:

▶ **Definition 3** ((α, β)-emulator). *Given an undirected graph $G = (V, E)$, a multiplicative stretch $\alpha \geq 1$ and an additive stretch $\beta \geq 0$, the graph $G^* = (V, E^*)$ with $E^* \subseteq V \times V$ is an (α, β)-emulator of G, iff*

$$\forall_{u,v \in V} \quad \mathrm{d}_G(u, v) \leq \mathrm{d}_{G^*}(u, v) \leq \alpha \cdot \mathrm{d}_G(u, v) + \beta.$$

If in addition $E^* \subseteq E$, then G^* is a spanner of G.

Peleg and Schäffer first studied multiplicative spanners (where $\beta = 0$) in [28], and additive emulators (where $\alpha = 1$) were introduced by Dor, Halperin, and Zwick [11]. The concept of near-additive spanners and emulators was later introduced by Elkin and Peleg [16], who showed that one can construct spanners with stretch $1 + \epsilon$ for arbitrarily small $\epsilon > 0$ and near-linear size, at the cost of allowing a sufficiently large constant additive stretch. In particular, they showed that for every integer $k \geq 1$, there exists a $(1 + \epsilon, \beta)$-spanner with $O(\beta \cdot n^{1+1/k})$ edges, where $\beta = O(\log(k)/\epsilon)^{\log(k)}$. Thorup and Zwick [29] later gave different constructions for near-additive spanners and emulators on undirected, unweighted graphs. A notable property of their construction is that it is universal, i.e., the algorithm is independent of ϵ, and thus the resulting spanner applies for all $\epsilon > 0$ simultaneously. Abboud, Bodwin and Pettie [1] provided a lower bound on the trade-off between ϵ, the additive stretch and the sparsity of emulators.

In recent years, also hopsets for directed graphs have been studied intensively [23, 4, 5]. Kogan and Parter [23] showed that for any directed graph, there is a near-exact hopset with size $O(n)$ and a hopbound of $\tilde{O}(n^{2/5})$, improving upon a long-standing folklore sampling algorithm, that implied a near-exact hopset of size $O(n)$ with a hopbound of $\tilde{O}(n^{1/2})$. Bernstein and Wein [4] further improved on this result and showed that one can obtain a linear-sized near-exact hopset with a hopbound of $\tilde{O}(n^{1/3})$. Bodwin and Hoppenworth [5] showed that the folklore sampling is essentially optimal for exact-hopsets in both directed and undirected graphs.

For undirected graphs, there is an interesting and extensively studied relationship between the constructions of near-additive spanners, emulators, and near-exact hopsets [13, 12, 1, 21, 14, 2, 27]. Around the same time, Elkin and Neiman [14] and Huang and Pettie [21] independently showed that classic constructions for emulators based on [29] can be used to obtain near-exact hopsets with a constant hopbound. In particular for any integer $k \geq 1$, Elkin and Neiman [14] gave a construction, that yields a near-exact hopset with hopbound $O(k/\epsilon)^k$ and size $O(n^{1+1/(2^k-1)})$, while Huang and Pettie [21] showed a construction yielding a near-exact hopset with hopbound $O(k/\epsilon)^k$ and size $O(n^{1+1/(2^{k+1}-1)})$. These results were unified by Neiman and Shabat [27], who devised a single algorithm that can provide state-of-the-art hopsets for undirected, weighted graphs with various stretch regimes.

Elkin and Neiman [15] gave an extensive survey on the connection between these objects, where they raised the following question:

> [15] *"A very interesting open problem is to explain the relationship between near-additive spanners and near-exact hopsets rigorously, i.e., by providing a reduction between these two objects."*

Subsequently, Kogan and Parter [22] made significant progress on this question by providing a general reduction from hopsets to emulators, spanners, and distance preservers. While their reduction to distance preservers is a bit more involved, the reduction to emulators and spanners is quite simple.

▶ **Observation 4** ([22]). *Let $G = (V, E)$ be an unweighted n-vertex Graph. Let H^* be some $(1 + \epsilon, \beta)$- hopset for G and let G^* be a multiplicative spanner with stretch t for G. Then, $H^* \cup G^*$ is a $(1 + \epsilon, \beta \cdot t)$-emulator.*

This result also extends to spanners, by a standard reduction. More recently, Kogan and Parter [24] initiated progress in the reverse direction, by showing that certain classes of distance preservers can be converted into exact hopsets.

1.1 Our Results

In this work, we present a general reduction from near-additive emulators to near-exact hopsets, thereby answering the question of [15] in the affirmative for sparse graphs. As our main contribution, we present a general reduction that shows that any emulator construction can be used to obtain a hopset with similar guarantees, formalized as follows.

▶ **Theorem 5.** *Let $G = (V, E)$ be an undirected weighted graph, with edge weights in $[1, W]$ and let $0 < \epsilon \leq 1$. Suppose there exists an algorithm $\mathcal{A}$ that, for any undirected unweighted n-vertex graph, constructs an (α, β)-emulator, with $\alpha \geq 1$ and $\beta \geq 0$, of size at most $S_{\mathcal{A}}(n, \alpha, \beta)$. Then there exists a $(\alpha(1 + 24\epsilon), O(\alpha t^2) \ln(nW))$-hopset of G of size $O(S_{\mathcal{A}}(n + m\frac{t}{\epsilon}, \alpha, \beta)\frac{1}{\epsilon} \ln(nW))$, where $t = \max(\frac{1}{\epsilon}, \frac{\beta}{\epsilon})$.*

A direct consequence of Theorem 5, by setting $\alpha = 1 + \epsilon$, is the following result for near-additive emulators, thus answering the question raised in [15] for sparse graphs.

▶ **Corollary 6.** *Let $G = (V, E)$ be an undirected weighted graph, with edge weights in $[1, W]$ and let $0 < \epsilon \leq 1$. Assume there exists an algorithm $\mathcal{A}$ that, for any undirected unweighted n-vertex graph and any $0 \leq \epsilon' \leq 1$, constructs a $(1 + \epsilon', \beta)$-emulator, with $\beta \geq 1$, of size at most $S_{\mathcal{A}}(n, \epsilon', \beta)$. Then G admits a $(1 + \epsilon, O(\frac{\beta^2}{\epsilon^2}) \ln(nW))$-hopset of size $O(S_{\mathcal{A}}(n + m\frac{\beta}{\epsilon^2}, \frac{\epsilon}{49}, \beta)\frac{1}{\epsilon} \ln(nW))$.*

Finally, we can make use of the reduction from [13] to remove the maximum weight W from the log-factor and get the following overall result.

▶ **Corollary 7.** *Let $G = (V, E)$ be an undirected weighted graph and let $0 < \epsilon \leq \frac{1}{2}$. Assume there exists an algorithm $\mathcal{A}$ that, for any undirected unweighted n-vertex graph and any $0 \leq \epsilon' \leq \frac{1}{2}$, constructs a $(1+\epsilon', \beta)$-emulator, with $\beta \geq 1$, of size at most $S_{\mathcal{A}}(n, \epsilon', \beta)$. Then G admits a $(1+\epsilon, O(\frac{\beta^2}{\epsilon^2}\ln(\frac{n}{\epsilon})))$-hopset of size $O((S_{\mathcal{A}}(n + m\frac{\beta}{\epsilon^2}, \frac{\epsilon}{294}, \beta)\frac{1}{\epsilon} + n)\ln(\frac{n}{\epsilon}))$.*

Compared to the "ideal result", having guarantees as close to the original emulator as possible, our reduction incurs a blowup in the size depending on the number of edges m in the input graph, the dependence on β in the hopbound is quadratic instead of linear, and our method introduces an additional log-factor in both the size and the hopbound. Although the size of the obtained hopset also depends on m, we believe that this has only a minor impact in practice, as the runtime of typical algorithmic applications usually already depends at least linearly on m and therefore adding roughly m additional edges does not change the asymptotic complexity.

1.2 Overview of Techniques

First, we introduce the notation and relevant definitions, then we give a high-level overview of the techniques used to achieve the reduction.

Notation

In this work, we consider an undirected, weighted graph $G = (V, E)$ with positive edge weights, where V is the set of vertices and E is the set of edges, with $n := |V|$ and $m := |E|$. Given an edge $(u, v) \in E$, its weight is denoted by the function $\mathrm{w}_G \colon E \to \mathbb{R}_{\geq 1}$, and all edge weights lie in the range $[1, W]$. More generally, for any weighted edge set F, we denote the weight of an edge $e \in F$ by $\mathrm{w}_F(e)$. Given a path π in G, its weight is defined as $\mathrm{w}_G(\pi) = \sum_{(u,v)\in\pi} \mathrm{w}_G((u, v))$ and $|\pi|$ is the number of edges contained in π. For any two vertices u, v on π, we denote by $\pi[u, v]$ the subpath of π between u and v. For a pair of vertices $u, v \in V$, the distance $\mathrm{d}_G(u, v)$ is defined as the minimum weight of any path from u to v in G. Such a path with minimum weight is referred to as a shortest path. If no such path exists, we define $\mathrm{d}_G(u, v) := \infty$.

Reduction

Given a weighted undirected graph $G = (V, E)$, we organize our construction around distance scales. We consider the scales $[(1+\epsilon)^i, (1+\epsilon)^{i+1})$ for some $0 < \epsilon \leq 1$ and $i \in \mathbb{N}_0$.

The core idea behind our reduction is to apply a given (α, β)-emulator construction, for unweighted undirected graphs, independently at each scale. For every scale i, we define an unweighted graph G_i that captures the distances in G at that scale. In particular, we connect pairs of vertices whose distance falls in the given range by a single edge. We then construct an emulator M_i on G_i. Since the emulators approximate the distances in the unweighted graphs, we can bound the number of hops of the resulting approximate shortest paths, and by appropriately rescaling the weight of these emulator edges, we obtain the approximate distances in the graph G. The hopset F is then formed by taking the union of these rescaled emulator edges over all distance scales.

To analyze the stretch and hopbound, we show that for any two vertices $u, v \in V$ we can find a sequence of hop-bounded alternative paths, a shortcut, each skipping a subpath of the shortest path π from u to v in G. Every shortcut reduces the remaining distance to v by a fixed fraction, ensuring that only a logarithmic number of such shortcuts are required. The idea of constructing a sequence of shortcuts follows the strategy used by Bernstein in [3] and later by Henzinger, Krinninger, and Nanongkai in [19].

Next, we consider a single such shortcut. We need to ensure that a *sufficiently long* consecutive subpath of the shortcut is contained entirely within an emulator M_i of a single scale. If we concatenate multiple shortcuts from emulators at different scales, the additive stretch β incurred by the emulator construction would appear additionally each time the path "switches" scale. Hence, preventing the need to make such "switches" allows β to be absorbed into the multiplicative stretch. To this end, a primary challenge is to find a sequence of vertices, starting from u, on the path π in G, such that:

- two consecutive vertices are connected by a single edge in G_i and
- the last vertex in the sequence is *sufficiently far* from u.

Then the corresponding emulator M_i can provide (almost) the entire required shortcut path, and the number of hops is roughly given by $\alpha k + \beta$, if the aforementioned sequence contains $k + 1$ points. This is non-trivial, as heavy edges (relative to the scale i) on the path may cause consecutive points in the sequence not to be connected in G_i. Intuitively, we would require a point of the sequence to lie "on" that edge. To address this issue, we subdivide these heavy edges in the graph G at each scale before defining the unweighted graphs G_i. We implicitly show that the error incurred by this "discretization" of large "continuous" edges is negligible. However, this approach introduces additional vertices and, as a result, the hopset F must be formally defined on the graph $G' = (V', E)$, where $V \subseteq V'$. It is sufficient to require the "hopset guarantees" to hold only for the vertices in G. This leads to the following definition.

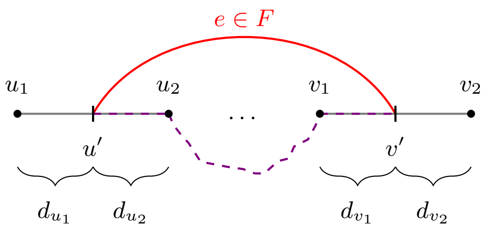
▶ **Definition 8** (T-constrained (α, β)-hopset)**.** *Given an undirected graph $G = (V, E)$ and a subset of vertices $T \subseteq V$, for a multiplicative stretch $\alpha \geq 1$, and a hopbound $\beta \in \mathbb{N}$, a T-constrained (α, β)-hopset is a set of weighted edges $F \subseteq V \times V$, such that*

$$\forall u, v \in T \quad \mathrm{d}_G(u, v) \leq \mathrm{d}_H^{\beta}(u, v) \leq \alpha \cdot \mathrm{d}_G(u, v)$$

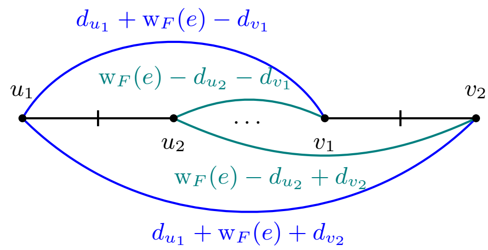
where $H = (V, E \cup F)$ is the hopset graph.

To obtain a hopset for the original graph G, we provide a projection that maps the V-constrained hopset F of G' back to a hopset defined entirely on the vertex set of G, while preserving the same stretch and hopbound guarantees. Conceptually, each subdivision vertex in $V' \setminus V$ represents a specific point "on" some original edge in E, and the idea behind the projection is to shift hopset edges that are incident to such a subdivision vertex to the nearby original vertices in V. Consider such an edge $(u', v') \in F$, suppose $u' \in V'$ lies "on" the edge $(u_1, u_2) \in E$ (where possibly $u' \in V$, in which case $u' = u_1 = u_2$), and similarly suppose $v' \in V'$ lies "on" the edge $(v_1, v_2) \in E$ (again allowing $v' = v_1 = v_2$, when $v' \in V$). We add up to four replacement edges connecting each endpoint u_1, u_2 to v_1 and v_2, to retain paths of the same total weight between vertices in V in each rescaled M_i after the removal of the subdivision vertices. Now, we need to ensure that we do not introduce any shortcuts that would lead to underestimating the distances by assigning weight to these new edges accordingly. First, we rename the nodes u_1 and u_2 such that u_2 is the node encountered (first) when traversing the shortest path in G' restricted to the relevant scale from u' to v'. We do the same for v_1 and v_2, where v_1 should be encountered (first) when traversing from the path starting from v'. For the new edge (u_i, v_j) with $i, j \in \{1, 2\}$, the weight is set such that the path u_1, u', v', v_2 in $G' \cup F$ has the same total weight as the path u_1, u_i, v_j, v_2 (consisting of one to three edges) in $G \cup \{(u_i, v_j)\}$. This process is illustrated in Figure 1.

Originally, the hopset edge e had to have weights such that $\mathrm{w}_F(e)$ was at least as large as the total weight of the shortest path from u' to v' in the subdivided graph of G within one scale. Furthermore, that path contained two edge-disjoint subpaths from u' to u_2 and from v_1 to v'. Therefore, we get that replacing e with four edges between original nodes in V, as described above, will not introduce any shortcuts.



(a) $e \in F$ between subdivision vertices.



(b) Weights of edges replacing e.

Figure 1 An example for the projection of an edge. The gray lines depict the edges of G, original vertices in V are shown as dots, whereas subdivision vertices are marked by a vertical line. The red line is a hopset edge in F. The violet path depicts the shortest path in the subdivided version of G from u' to v'. The blue and teal edges depict the replacement edges for $e = (u', v')$.

Roadmap

The proof of Theorem 5 is split into two main parts; building a hopset on an extended graph with additional vertices and converting it into a hopset on G while retaining the guarantees on hopbound and stretch. First, we describe the construction of a hopset on the extended graph in Section 2.1. Then, we analyze the resulting size of this hopset in Section 2.2 and in Section 2.3 we consider the resulting stretch and hopbound. Regarding the projection to a hopset on G, the details and the proof that the same guarantees hold are deferred to the full version of this paper due to space constraints.

2 Hopset Reduction

2.1 Construction

In this section, we describe the construction of the hopset, which is organized around distance scales of the form $[(1+\epsilon)^i, (1+\epsilon)^{i+1})$, for an integer $i \geq 0$ and $0 < \epsilon \leq 1$. Let $i_{\max}$ denote the index such that the diameter of the graph falls into the distance scale $[(1+\epsilon)^{i_{\max}}, (1+\epsilon)^{i_{\max}+1})$. We therefore only need to consider the indices $0 \leq i \leq i_{\max}$, as all higher scales are empty. Since the diameter of G is bounded by nW, we have $i_{\max} \leq \lceil \log_{1+\epsilon}(nW) \rceil$ distinct distance scales. At each scale i, we first create a subdivision graph $\tilde{G}_i$ of G with edge weights bounded relative to the current scale. Over the vertex set of $\tilde{G}_i$, we then construct an unweighted graph G_i which connects vertex pairs whose distance lies within the current scale. To bound the number of hops, we use an arbitrary (α, β)-emulator of G_i, whose weights are rescaled to approximate the distances in G. The hopset is obtained by taking the union over all distance scales i, of the rescaled emulator edges, together with the auxiliary edges introduced during the subdivision at each scale. In order to ensure the desired hopbound, we define $t := \max(\frac{1}{\epsilon}, \frac{\beta}{\epsilon})$.

Subdivision Step

Let $\tilde{G}_i := (\tilde{V}_i, \tilde{E}_i)$ be the graph, obtained from G by subdividing each edge $e \in E$ whose weight satisfies

$$\epsilon(1+\epsilon)^i < \mathrm{w}_G(e) \leq t(1+\epsilon)^{i+1} \tag{1}$$

into k-segments, with $k = \left\lceil \frac{\mathrm{w}_G(e)}{\epsilon(1+\epsilon)^i} \right\rceil$.

The edge e is replaced by a path of k edges $e_1, \ldots, e_k$, each with weight $\mathrm{w}_{\tilde{G}_i}(e_j) := \frac{\mathrm{w}_G(e)}{k}$ for $1 \leq j \leq k$, so that the total weight of the path remains $\mathrm{w}_G(e)$. Thus, for the edge $e = (v, w) = (u_0, u_k)$, we introduce $k-1$ new vertices $u_1, \ldots, u_{k-1}$, so that $e_j = (u_{j-1}, u_j)$.

Based on this subdivision, we now define the notion of a vertex's origin.

▶ **Definition 9.** *For any vertex $u \in \tilde{V}_i$, the origin of u is defined as:*

$$\mathrm{origin}(u) := \begin{cases} \{u\} & \text{if } u \in V \\ \{v, w\} & \text{if } u \in \tilde{V}_i \setminus V \text{ and } u \text{ was created in the subdivision of the edge } (v, w) \in E. \end{cases}$$

▶ **Definition 10.** *For any vertex $u \in \tilde{V}_i$ and original endpoint $x \in \mathrm{origin}(u)$, $\delta(x, u)$ denotes the distance between x and u along the subdivision path.*

- *If $u \in \tilde{V}_i \setminus V$, suppose that u was created by subdividing the edge $e = (v, w) \in E$ into k edges $e_1, \ldots, e_k$, and let $u = u_j$ with $1 \leq j < k$ be the j-th vertex along this path (with $u_0 = v$ and $u_k = w$). Then for each vertex in $\mathrm{origin}(u) = \{v, w\}$, we define*

$$\delta(u_j, v) := \sum_{l=1}^{j} \mathrm{w}_{\tilde{G}_i}(e_l) = \frac{j}{k} \mathrm{w}_G(e),\ \delta(u_j, w) := \sum_{l=j+1}^{k} \mathrm{w}_{\tilde{G}_i}(e_l) = \frac{k-j}{k} \mathrm{w}_G(e).$$

- *Otherwise, if $u \in V$, then $\delta(u, u) := 0$.*

Using these definitions we define the auxiliary edge set $\tilde{F}_i$ of $\tilde{G}_i$, as

$$\tilde{F}_i = \{(u, x) \mid u \in \tilde{V}_i \setminus V \wedge x \in \mathrm{origin}(u)\}$$

where the weight of each edge is $\mathrm{w}_{\tilde{F}_i}((u, x)) := \delta(u, x)$. Before we continue with the construction, observe the following important properties of the subdivision graph $\tilde{G}_i$. The corresponding proofs are in Appendix A.

▶ **Observation 11.** *Let π be any path from u to v in G. There exists a path $\tilde{\pi}$ in $\tilde{G}_i$ between u and v, such that $\mathrm{w}_G(\pi) = \mathrm{w}_{\tilde{G}_i}(\tilde{\pi})$.*

▷ Claim 12. For every pair of vertices $u, v \in V$ with $\mathrm{d}_G(u, v) \leq t(1+\epsilon)^{i+1}$, every edge on the shortest path from u to v in $\tilde{G}_i$ has weight at most $\epsilon(1+\epsilon)^i$.

▷ Claim 13. The graph $\tilde{G}_i$ contains at most $O(m\frac{t}{\epsilon})$ edges and at most $O(n + m\frac{t}{\epsilon})$ vertices.

▶ **Lemma 14.** *An edge $e \in E$ can be subdivided in at most $\log_{1+\epsilon}(\frac{t}{\epsilon}) + 1$ different graphs $\tilde{G}_i$.*

Hopset Definition

Let $G_i := (\tilde{V}_i, E_i)$ be an unweighted graph, defined over the vertex set $\tilde{V}_i$ of $\tilde{G}_i$, which contains an edge for every pair of vertices whose distance in $\tilde{G}_i$ lies in the i-th distance scale. Formally, the edge set is given by $E_i := \{(u, v) \in \tilde{V}_i^2 \mid (1+\epsilon)^i \leq \mathrm{d}_{\tilde{G}_i}(u, v) < (1+\epsilon)^{i+1}\}$. Let M_i be an (α, β)-emulator of G_i. Let M_i' be the graph obtained by rescaling the weights of M_i by the upper bound of the current distance scale. Specifically, for each edge $e \in \mathrm{E}(M_i)$ its weight in M_i' is given by $\mathrm{w}_{M_i'}(e) := (1+\epsilon)^{i+1} \mathrm{w}_{M_i}(e)$. Finally, we define the hopset F as the union of the edge sets of all M_i' and the auxiliary sets $\tilde{F}_i$ across distance scales:

$$F := \bigcup_{i=0}^{i_{\max}} \mathrm{E}(M_i') \cup \tilde{F}_i.$$

Extended Graph

Since we consider arbitrary emulators M_i of the unweighted graphs $G_i = (\tilde{V}_i, E_i)$, we make no assumptions about their internal structure. In particular, M_i may contain edges incident to vertices in $\tilde{V}_i \setminus V$, which were introduced in the subdivision step and do not appear in the original graph $G = (V, E)$. Consequently, a path in M_i from $u \in V$ to $v \in V$ may contain such edges, which would not exist in a hopset defined solely on the vertex set V. Hence, to ensure all paths in M_i correspond to a path in the hopset graph, we need to extend the vertex set of G by including all new vertices of each graph $\tilde{G}_i$. Let $G' := (V', E)$ be the extended graph, with

$$V' := \bigcup_{i=0}^{i_{\max}} \tilde{V}_i.$$

Thus F is a V-constrained hopset of the extended graph G', and the resulting hopset graph is $H = (V', E \cup F)$. Note that G' is obtained from G by only adding vertices. All vertices in $V' \setminus V$ are isolated in G' and only become connected through the edges in F in the graph H. Thus, the distances and all paths between vertices in V are identical in G and G'. Throughout Section 2.3, we may refer to shortest paths in G for simplicity, even though the hopset is defined on G'.

2.2 Size Analysis

Next, we analyze the size of the hopset F.

▶ **Lemma 15.** *Let $\mathcal{A}$ be an algorithm that, for any n-vertex graph, constructs an (α, β)-emulator with $S_{\mathcal{A}}(n, \alpha, \beta)$ edges. Then the construction in Section 2.1 yields a hopset $F \subseteq V' \times V'$ which consists of $O(S_{\mathcal{A}}(n + m\frac{t}{\epsilon}, \alpha, \beta) \log_{1+\epsilon}(nW) + m\frac{t}{\epsilon} \log_{1+\epsilon}(\frac{t}{\epsilon}))$ edges.*

Proof. Using the algorithm $\mathcal{A}$, each (α, β)-emulator M_i is constructed on the graph G_i. Let n_i denote the number of vertices in G_i. From Claim 13 we know that $n_i \leq n + \frac{2t}{\epsilon}m$. Thus each emulator M_i contains $O(S_{\mathcal{A}}(n + m\frac{t}{\epsilon}, \alpha, \beta))$ edges. There are $i_{\max} \leq \log_{1+\epsilon}(nW)$ different distance scales, therefore we have at most $\log_{1+\epsilon}(nW)$ of such emulators. By Lemma 14 and Claim 13, it follows that over all edges and distance scales the number of newly introduced vertices in the subdivision step is $O(m\frac{t}{\epsilon} \log_{1+\epsilon}(\frac{t}{\epsilon}))$, so the size of the auxiliary edge set $\tilde{F}$ is also $O(m\frac{t}{\epsilon} \log_{1+\epsilon}(\frac{t}{\epsilon}))$.[1] Thus the total size of the hopset F is in $O(S_{\mathcal{A}}(n + m\frac{t}{\epsilon}, \alpha, \beta) \log_{1+\epsilon}(nW) + m\frac{t}{\epsilon} \log_{1+\epsilon}(\frac{t}{\epsilon}))$. ◀

Note that the size bound stated in Lemma 15 is conservative, as it assumes that every edge gets subdivided at every distance scale i into the maximum possible number of subedges.

2.3 Hop Reduction

Let $G = (V, E)$ be the original graph, and let F be the edge set as defined in Section 2.1. In this section we show that F is indeed a V-constrained $(\alpha(1 + 24\epsilon), O(\alpha t^2) \ln(nW))$-hopset of G', for $t = \max(\frac{1}{\epsilon}, \frac{\beta}{\epsilon})$. Let π be a shortest path from u to v in G. We will show that there exists a vertex w on π and an alternative path π' from u to w in $H = (V', E \cup F)$ with the following properties:

[1] The edges in $\tilde{F}$ get removed in the projection back to a hopset on G. See the full version of this paper.

(P1) The distance from u to w in G is at least Δt, with $\Delta := \frac{d_G(u,v)}{t^2}$.
(P2) The alternative path π' consists of at most $3t(\alpha+1)$ edges of F and includes at most one additional edge from either E or F.
(P3) The weight of the alternative path π' in H is at most $\alpha(1+24\epsilon)$ times the distance from u to w in G.

Thus, by choosing to take such an alternative path π' from u to w instead of a subpath of π we take at most $3t(\alpha+1)+1$ hops, while decreasing the remaining distance to v by at least $\Delta t = \frac{d_G(u,v)}{t}$. However, this introduces an approximation error of $\alpha(1+24\epsilon)$.

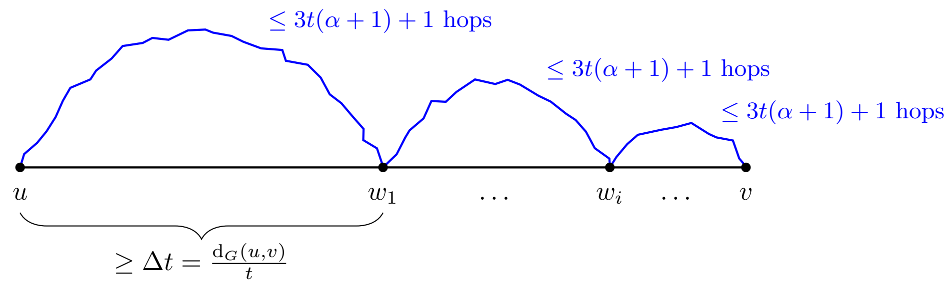


Figure 2 Instead of following the shortest path from u to v in G (black), we take a sequence of alternative paths in H (blue). Each alternative path consists of at most $3t(\alpha+1)+1$ hops and skips at least a fraction $\frac{1}{t}$ of the remaining distance to v.

Through repeated application of this strategy with decreasing Δ, and concatenating the alternative paths, we obtain an alternative path π'' in H from u to v (see Figure 2) that has at most $\ln(nW)(3t^2(\alpha+1)+t)$ hops. This follows from (P1), which guarantees that with each shortcut we skip a fixed fraction $\frac{1}{t}$ on the remaining path to v and (P2), which limits the number of hops in each shortcut. Property (P3) ensures that every alternative path has a multiplicative stretch of at most $\alpha(1+24\epsilon)$, and thus the whole path π'' also has the same multiplicative stretch. The idea of using such alternative paths to shortcut portions of the shortest path is similar to the approach used in [3, 19].

Before we proceed with the analysis as explained above, we prove two important structural properties of our construction from Section 2.1. For a fixed distance scale i, Claim 16 ensures the existence of a vertex u_p along the shortest path from u to v in the subdivided graph $\tilde{G}_i$, for which we can guarantee that there exists path from u to u_p in the unweighted graph G_i with a bounded number of edges. For any path in G_i, Claim 17 provides a bound on the weight and number of edges of the corresponding shortest path in the emulator of this distance scale and hence in H.

▷ Claim 16. Let $\Delta \geq 1$ and let i be such that $(1+\epsilon)^i \leq \Delta < (1+\epsilon)^{i+1}$. Consider any pair of vertices $u, v \in V$ with $d_G(u,v) < \infty$ and $d_G(u,v) \geq \Delta t$. Then there exists a vertex $u_p \in \tilde{V}_i$ on the shortest path from u to v in $\tilde{G}_i$, for some $0 \leq p \leq t + t\epsilon$, such that $d_{\tilde{G}_i}(u, u_p) \leq \Delta t$ and there is a path of p edges from u to u_p in G_i. Hence u_p satisfies $p(1+\epsilon)^i \leq d_{\tilde{G}_i}(u, u_p) \leq p(1+\epsilon)^{i+1}$. Moreover, there is no vertex u'_p on the remaining subpath from u_p to v, such that $d_{\tilde{G}_i}(u_p, u'_p) \geq (1+\epsilon)^i$ and $d_{\tilde{G}_i}(u, u'_p) \leq \Delta t$.

Proof. Let $\tilde{\pi}$ be the shortest path from u to v in $\tilde{G}_i$, which by Observation 11 has weight equal to $d_G(u,v)$. We iteratively construct a sequence of vertices $u_0, u_1, \dots, u_p$ along $\tilde{\pi}$ starting at $u_0 := u$. For each integer $j \geq 0$ we define u_{j+1} to be the first vertex on $\tilde{\pi}$ after u_j that satisfies $d_{\tilde{G}_i}(u_j, u_{j+1}) \geq (1+\epsilon)^i$. If no such vertex exists or $d_{\tilde{G}_i}(u, u_{j+1}) > \Delta t$ we stop and set $u_p := u_j$. Otherwise, we continue the iteration.

We now show by induction on j that for every $0 \leq j \leq p$, there exists a path of j edges from u to u_j in G_i.

Base Case. $j = 0$. Since $u_0 = u$, the trivial path from u to u_0 of length 0 exists in G_i.

Induction Step. Assume that for any $j < p$, there exists a path of j edges from u to u_j in G_i. Let x denote the last vertex on $\tilde{\pi}$ with $\mathrm{d}_{\tilde{G}_i}(u_j, x) < (1+\epsilon)^i$, and let u_{j+1} be its successor. By the choice of x, u_{j+1} must satisfy $\mathrm{d}_{\tilde{G}_i}(u_j, u_{j+1}) \geq (1+\epsilon)^i$. Since $\mathrm{d}_{\tilde{G}_i}(u, u_p) \leq \Delta t \leq t(1+\epsilon)^{i+1}$ Claim 12 guarantees that every edge e on the subpath of $\tilde{\pi}$ from u to u_p has weight $\mathrm{w}_{\tilde{G}_i}(e) \leq \epsilon(1+\epsilon)^i$. Thus, the distance between u_j and u_{j+1} can be upper bounded by $\mathrm{d}_{\tilde{G}_i}(u_j, u_{j+1}) = \mathrm{d}_{\tilde{G}_i}(u_j, x) + \mathrm{w}_{\tilde{G}_i}((x, u_{j+1})) < (1+\epsilon)^i + \epsilon(1+\epsilon)^i = (1+\epsilon)^{i+1}$. Combining these bounds, we have $(1+\epsilon)^i \leq \mathrm{d}_{\tilde{G}_i}(u_j, u_{j+1}) < (1+\epsilon)^{i+1}$. Therefore, by definition, G_i contains the edge (u_j, u_{j+1}) and and hence there exists a path of $j+1$ edges from u to u_{j+1} in G_i.

Note that, since there is a path of p edges from u to u_p in G_i , we have $p(1+\epsilon)^i \leq \mathrm{d}_{\tilde{G}_i}(u, u_p) \leq p(1+\epsilon)^{i+1}$.

Since the constructed final vertex u_p also satisfies $\mathrm{d}_{\tilde{G}_i}(u, u_p) \leq \Delta t < t(1+\epsilon)^{i+1}$, we have $p(1+\epsilon)^i < t(1+\epsilon)^{i+1}$ and it follows directly that $p < t(1+\epsilon) = t + t\epsilon$. ◁

By the definition of G_i, the guarantees of the emulator M_i and the definition of weights in M'_i we get the following claim. The full proof is in Appendix B.

▷ Claim 17. Consider a distance scale $i \geq 0$. If there exists a path of l edges between two vertices u,v in the graph G_i, then there exists a path π_i in $H = (V', E \cup F)$, with $|\pi_i| \leq \alpha l + \beta$ and $\mathrm{w}_H(\pi_i) \leq (1+\epsilon)^{i+1}(\alpha l + \beta)$.

With Claim 16 and Claim 17, we now show that we can construct a shortcut π' with the desired properties.

► **Lemma 18.** *Let $\Delta > 0$, for every pair of vertices $u, v \in V$ with $\mathrm{d}_G(u, v) < \infty$ and $\mathrm{d}_G(u, v) \geq \Delta t$, we can find a vertex w on the shortest path from u to v in G, and a path π' from u to w in $H = (V', E \cup F)$, such that $\mathrm{d}_G(u, w) \geq \Delta t$, $|\pi'| \leq 3t(\alpha+1) + 1$ and $\mathrm{w}_H(\pi') \leq \alpha(1 + 24\epsilon) \cdot \mathrm{d}_G(u, w)$.*

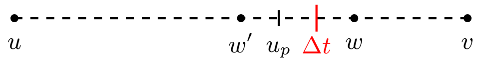
Proof. Let π be the shortest path from u to v in G. Let w' be the closest vertex to v on π for which $\mathrm{d}_G(u, w') < \Delta t$ still holds, and let w be its successor on π.

In the following proof, we assume $\Delta \geq 1$, as otherwise the claim follows immediately: Given that all edge weights are at least 1 and $\mathrm{d}_G(u, w') < \Delta t < t$, the shortest path between u and w' in G already consists of at most t edges. We define the alternative path π' to be the subpath of π from u to w. Then the guarantees for the number of hops and the stretch hold, since $|\pi'| = t + 1 < 3t(\alpha+1) + 1$ and $\mathrm{w}_H(\pi') = \mathrm{d}_G(u, w) \leq \alpha(1 + 24\epsilon)\,\mathrm{d}_G(u, w)$.

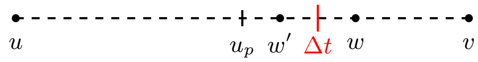
For the remainder of the proof, consider i such that $(1+\epsilon)^i \leq \Delta < (1+\epsilon)^{i+1}$, that is Δ lies in the i-th distance scale. Let $\tilde{\pi}$ be the shortest path from u to v in $\tilde{G}_i$, which by Observation 11 has weight equal to $\mathrm{d}_G(u, v)$ and consists of the vertices of π, together with any intermediate vertices introduced during the subdivision step. The strategy to construct the path π' with the desired properties from u to the vertex w in this case is as follows: First, we identify a vertex u' on $\tilde{\pi}$ and a path π_{pre} in H from u to u' with at most $3t(\alpha+1)$ hops and a bounded stretch. Then we show that there exists a single edge from u' to w in either E or in F, and by extending π_{pre} with this edge, we obtain π', whose stretch is at most $\alpha(1 + 24\epsilon)$. Let u_p be the vertex from Claim 16, so that $\mathrm{d}_{\tilde{G}_i}(u, u_p) \leq \Delta t$ and there exists a path of $p \leq t + t\epsilon$ edges from u to u_p in G_i. Let π_i be the path from u to u_p in H guaranteed by Claim 17, with $\mathrm{w}_H(\pi_i) \leq (1+\epsilon)^{i+1}(\alpha p + \beta)$ and $|\pi_i| \leq \alpha p + \beta \leq \alpha(t + t\epsilon) + \beta$. Note that π_i may be the trivial path of length 0. We will address this case separately when relevant.

The definition of u' and the path π_{pre} depend on the position of u_p relative to w' along $\tilde{\pi}$ (see Figure 3). Intuitively, we choose u' to be a vertex that is at least as far along $\tilde{\pi}$ as w', so that we can reach w with a single edge. Consider the following two cases:

- Case 1: If $\mathrm{d}_{\tilde{G}_i}(u, u_p) \geq \mathrm{d}_{\tilde{G}_i}(u, w')$, we define $u' := u_p$.
- Case 2: If $\mathrm{d}_{\tilde{G}_i}(u, u_p) < \mathrm{d}_{\tilde{G}_i}(u, w')$, we define $u' := w'$.



(a) Case 1.



(b) Case 2.

Figure 3 The dashed line represents the shortest path from u to v in $\tilde{G}_i$. The original vertices $u, v, w', w \in V$ are a marked by a dot, while the vertex $u_p \in \tilde{V}_i \subseteq V'$ is marked by a vertical line.

Case 1: Suppose $\mathrm{d}_{\tilde{G}_i}(u, u_p) \geq \mathrm{d}_{\tilde{G}_i}(u, w')$. In this case, the vertex u_p already lies at or beyond w' along the path $\tilde{\pi}$. We therefore have $u' = u_p$ and define $\pi_{\text{pre}} := \pi_i$, as illustrated in Figure 4. Thus, we have $\mathrm{w}_H(\pi_{\text{pre}}) = \mathrm{w}_H(\pi_i) \leq (1+\epsilon)^{i+1}(\alpha p + \beta)$ and $|\pi_{\text{pre}}| = |\pi_i| \leq \alpha(t + t\epsilon) + \beta = \alpha t + \alpha t\epsilon + \beta \leq \alpha t + \alpha t \epsilon + t\epsilon \leq 2t(\alpha + 1)$, since $\beta \leq t\epsilon$ and $\epsilon \leq 1$.

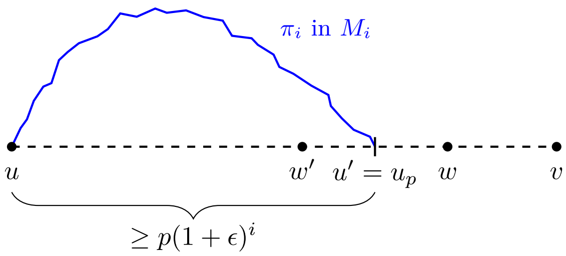


Figure 4 The dashed line represents the shortest path from u to v in $\tilde{G}_i$. The vertices u, u', and v (all in V) are marked by dots. The solid blue line is the alternative path π_{pre} from u to u' in H, which in this case only consists of π_i.

By comparing the weight of the path π_{pre} in H to the exact distance from u to u' in $\tilde{G}_i$, we can bound the multiplicative stretch λ_1 which the path π_{pre} introduces:

$$\lambda_1 := \frac{\mathrm{w}_H(\pi_{\text{pre}})}{\mathrm{d}_{\tilde{G}_i}(u, u')} \leq \frac{(1+\epsilon)^{i+1}(\alpha p + \beta)}{\mathrm{d}_{\tilde{G}_i}(u, u')}.$$

The vertex u' satisfies $\mathrm{d}_{\tilde{G}_i}(u, u') \geq p(1+\epsilon)^i$ by construction of u_p. Substituting this yields

$$\lambda_1 = \frac{\mathrm{w}_H(\pi_{\text{pre}})}{\mathrm{d}_{\tilde{G}_i}(u, u')} \leq \frac{(1+\epsilon)^{i+1}(\alpha p + \beta)}{p(1+\epsilon)^i} = \frac{(1+\epsilon)(\alpha p + \beta)}{p} = (1+\epsilon)\left(\alpha + \frac{\beta}{p}\right).$$

Case 2: Suppose that $\mathrm{d}_{\tilde{G}_i}(u, u_p) < \mathrm{d}_{\tilde{G}_i}(u, w')$. In this situation, u_p lies strictly before w' along $\tilde{\pi}$. We have defined $u' = w'$ and to construct π_{pre}, we extend the path π_i with an additional shortcut path from u_p to w', as shown in Figure 5. Since $\mathrm{d}_{\tilde{G}_i}(u, w') < \Delta t$, the construction of u_p in Claim 16 implies that the remaining distance from u_p to w' is $\mathrm{d}_{\tilde{G}_i}(u_p, w') < (1+\epsilon)^i$.

Case 2.1: We first consider the trivial case where $u_p = u$, thus π_i is the empty path. Since $\mathrm{d}_{\tilde{G}_i}(u, u') = \mathrm{d}_{\tilde{G}_i}(u_p, w') < (1+\epsilon)^i$, we know that there exists a smaller distance scale $j < i$, such that $\mathrm{d}_G(u, u')$ falls into this scale, i.e. $(1+\epsilon)^j \leq \mathrm{d}_G(u, u') < (1+\epsilon)^{j+1}$. Hence, by definition, we have an edge from u to u' in the graph G_j. Let π_{pre} be defined as the path from u to u' in H with $\mathrm{w}_H(\pi_{\text{pre}}) \le (1+\epsilon)^{j+1}(\alpha+\beta)$ and $|\pi_{\text{pre}}| = \alpha+\beta$, as provided by Claim 17. To bound the stretch of this shortcut, we compare the weight of π_{pre} to the exact distance between u and u' in G. Since $(1+\epsilon)^j \le \mathrm{d}_G(u,u')$, this yields a stretch $\lambda_{2.1}$ of

$$\lambda_{2.1} := \frac{\mathrm{w}_H(\pi_{\text{pre}})}{\mathrm{d}_{\tilde{G}_i}(u,u')} \le \frac{(1+\epsilon)^{j+1}(\alpha+\beta)}{(1+\epsilon)^j} = (1+\epsilon)(\alpha+\beta).$$

Case 2.2: Next, we consider the case where $u \neq u_p$. Let x be the vertex in $\text{origin}(u_p)$ that succeeds u_p on $\tilde{\pi}$, i.e., x is the first vertex in V on the subpath $\tilde{\pi}[u_p, v]$.

We construct the path π_{pre} by combining the path π_i from u to u_p with a path π_j from x to u'. If $u_p = x$, we can directly concatenate π_i and π_j. Otherwise if $u_p \neq x$, by definition, there exists an edge $(u_p, x) \in F$ with weight $\mathrm{d}_{\tilde{G}_i}(u_p, x)$ and we extend π_i by (u_p, x) before concatenating it with π_j.

The definition of π_j is based on the following cases:

- If $x = u'$: Let π_j be the empty path.
- If $x \neq u'$: We know that the remaining distance from x to u' is $\mathrm{d}_{\tilde{G}_i}(x,u') < (1+\epsilon)^i$, since $\mathrm{d}_{\tilde{G}_i}(u,x) \ge \mathrm{d}_{\tilde{G}_i}(u,u_p)$ and $\mathrm{d}_{\tilde{G}_i}(u,u') - \mathrm{d}_{\tilde{G}_i}(u,u_p) < (1+\epsilon)^i$. Hence $\mathrm{d}_{\tilde{G}_i}(x,u')$ falls into some distance scale j with $j < i$ and since $x, u' \in V$ there is, by definition, a single edge from x to u' in G_j. Let π_j denote the path in H with $\mathrm{w}_H(\pi_j) \le (1+\epsilon)^{j+1}(\alpha+\beta)$ from x to u', guaranteed by Claim 17.

Thus the number of edges in π_{pre} is $\pi_{\text{pre}} \le |\pi_i| + |\pi_j| + 1$. Its weight is $\mathrm{w}_H(\pi_{\text{pre}}) \le \mathrm{w}_H(\pi_i) + \mathrm{w}_H(\pi_j) + \mathrm{d}_{\tilde{G}_i}(u_p, x)$.

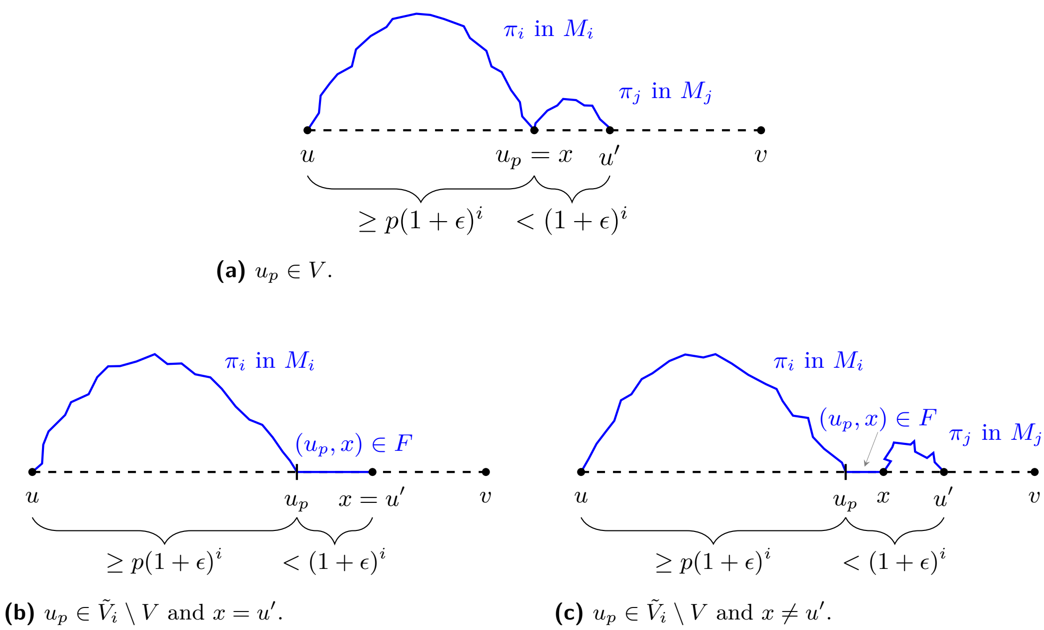


Figure 5 The dashed line represents the shortest path from u to v in $\tilde{G}_i$. Vertices in V are marked by a dot, while vertices in $\tilde{V}_i \setminus V$ are marked by a vertical line. The solid blue line is the alternative path π_{pre} from u to u' in H, consisting of subpaths π_i and π_j. Furthermore, if $u_p \in \tilde{V}_i \setminus V$ the path π_{pre} is formed by extending the path π_i with the edge $(u_p, x) \in F$, and π_j. This can be seen in (b) and (c) (the subpath π_j in (b) is empty).

Using $|\pi_i| \le \alpha(t + t\epsilon) + \beta$ and $|\pi_j| \le \alpha + \beta$, the number of edges can be bounded as follows

$$\begin{aligned}|\pi_{\text{pre}}| \le |\pi_i| + |\pi_j| + 1 \le \alpha(t + t\epsilon) + \beta + \alpha + \beta + 1 &= \alpha t + \alpha t\epsilon + \alpha + 2\beta + 1\\ &\le \alpha t + \alpha t\epsilon + \alpha t + 2t\epsilon + t \le 3t(\alpha + 1).\end{aligned}$$

To analyze the stretch of this shortcut, we compare the weight of π_{pre} to the exact distance from u to u' in $\tilde{G}_i$. We know that, if present, the edge (u_p, x) has weight $\mathrm{d}_{\tilde{G}_i}(u_p, x)$, which does not contribute to the stretch of the shortcut path. Otherwise $u_p = x$ and therefore $\mathrm{d}_{\tilde{G}_i}(u_p, x) = 0$. Hence, to obtain the stretch $\lambda_{2.2}$, it suffices to bound the ratio

$$\lambda_2 := \frac{\mathrm{w}_H(\pi_{\text{pre}}) - \mathrm{d}_{\tilde{G}_i}(u_p, x)}{\mathrm{d}_{\tilde{G}_i}(u, u') - \mathrm{d}_{\tilde{G}_i}(u_p, x)}.$$

Since $\mathrm{d}_{\tilde{G}_i}(u, u') - \mathrm{d}_{\tilde{G}_i}(u_p, x) > \mathrm{d}_{\tilde{G}_i}(u, u_p)$ and $\mathrm{d}_{\tilde{G}_i}(u, u_p) > p(1 + \epsilon)^i$, this yields:

$$\begin{aligned}\lambda_{2.2} = \frac{\mathrm{w}_H(\pi_{\text{pre}}) - \mathrm{d}_{\tilde{G}_i}(u_p, x)}{\mathrm{d}_{\tilde{G}_i}(u, u') - \mathrm{d}_{\tilde{G}_i}(u_p, x)} &\le \frac{\mathrm{w}_H(\pi_i) + \mathrm{w}_H(\pi_j) + \mathrm{d}_{\tilde{G}_i}(u_p, x) - \mathrm{d}_{\tilde{G}_i}(u_p, x)}{\mathrm{d}_{\tilde{G}_i}(u, u') - \mathrm{d}_{\tilde{G}_i}(u_p, x)}\\ &\le \frac{\mathrm{w}_H(\pi_i) + \mathrm{w}_H(\pi_j)}{p(1 + \epsilon)^i}.\end{aligned}$$

Further since $\mathrm{w}_H(\pi_i) \le (1+\epsilon)^{i+1}(\alpha p+\beta)$ and $\mathrm{w}_H(\pi_j) \le (1+\epsilon)^{j+1}(\alpha+\beta) < (1+\epsilon)^{i+1}(\alpha+\beta)$, we have

$$\mathrm{w}_H(\pi_i) + \mathrm{w}_H(\pi_j) \le (1 + \epsilon)^{i+1}(\alpha p + \beta) + (1 + \epsilon)^{i+1}(\alpha + \beta) = (1 + \epsilon)^{i+1}(\alpha p + \alpha + 2\beta).$$

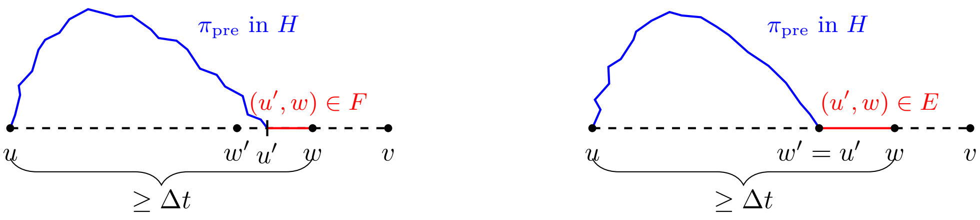
This allows us to bound $\lambda_{2.2}$ by

$$\begin{aligned}\lambda_{2.2} &\le \frac{\mathrm{w}_H(\pi_i) + \mathrm{w}_H(\pi_j)}{p(1 + \epsilon)^i} \le \frac{(1 + \epsilon)^{i+1}(\alpha p + \alpha + 2\beta)}{p(1 + \epsilon)^i}\\ &= \frac{(1 + \epsilon)(\alpha p + \alpha + 2\beta)}{p} = (1 + \epsilon)\left(\alpha + \frac{\alpha}{p} + 2\frac{\beta}{p}\right).\end{aligned}$$

Having defined u' and π_{pre}, it remains to show that we can reach the vertex w on π with a single edge from u' to w.

- If $u' \ne w'$: This only occurs in Case 1 where $u' = u_p$ and we have $\mathrm{d}_{\tilde{G}_i}(u, w') < \mathrm{d}_{\tilde{G}_i}(u, u_p)$. Since $\mathrm{d}_{\tilde{G}_i}(u, u_p) \le \Delta t \le \mathrm{d}_{\tilde{G}_i}(u, w)$, the vertex u_p must have been introduced during the subdivision of the edge $(w', w) \in E$. Therefore, in our construction, we have inserted the edge (u', w) into $\tilde{F}_i \subseteq F$, with weight $\mathrm{w}_F((u', w)) = \delta(u', w) = \mathrm{d}_{\tilde{G}_i}(u', w)$.
- If $u' = w'$: This can occur in either Case 1 or Case 2. We simply take the edge $(u', w) = (w', w) \in E$.

Let π' be the path in H obtained by extending π_{pre} with the edge $(u', w) \in E$ or $(u', w) \in F$, as illustrated in Figure 6.



(a) $u' \in \tilde{V}_i \setminus V$ (b) $u' = w' \in V$

Figure 6 The dashed line represents the shortest path from u to v in $\tilde{G}_i$, while the solid line is the alternative path π' from u to w in H. The original vertices in V are marked by dots, whereas nodes introduced during the subdivision step are represented by vertical lines. The blue line shows the path π_{pre} and the red line is the edge (u', w) in E or F depending on the case.

We have previously shown that in all cases $|\pi_{\text{pre}}| \leq 3t(\alpha + 1)$. Thus the path $|\pi'|$ consists of at most $3t(\alpha + 1) + 1$ edges as claimed. Depending on the case, we showed that $\mathrm{w}_H(\pi_{\text{pre}}) \leq \lambda \cdot \mathrm{d}_{\tilde{G}_i}(u, u')$ for $\lambda \in \{\lambda_1, \lambda_{2.1}, \lambda_{2.2}\}$. Since the last edge (u', w) of the path π is exact, it introduces no additional error, and the weight of π' can be expressed as

$$\begin{aligned}\mathrm{w}_H(\pi') &= \mathrm{w}_H(\pi_{\text{pre}}) + \mathrm{w}_H((u', w)) \leq \lambda \cdot \mathrm{d}_{\tilde{G}_i}(u, u') + \mathrm{d}_{\tilde{G}_i}(u', w) \\ &= \lambda \cdot \mathrm{d}_{\tilde{G}_i}(u, u') + (\mathrm{d}_{\tilde{G}_i}(u, w) - \mathrm{d}_{\tilde{G}_i}(u, u')) = \mathrm{d}_{\tilde{G}_i}(u, w) + (\lambda - 1) \cdot \mathrm{d}_{\tilde{G}_i}(u, u').\end{aligned}$$

This allows us to bound the stretch of π' by comparing its weight to the exact distance from u to w in $\tilde{G}_i$ (which coincides with the distance in G). Since $\mathrm{d}_{\tilde{G}_i}(u, w) \geq \Delta t \geq t(1 + \epsilon)^i$, we obtain the following:

$$\frac{\mathrm{w}_H(\pi')}{\mathrm{d}_{\tilde{G}_i}(u, w)} \leq \frac{\mathrm{d}_{\tilde{G}_i}(u, w) + (\lambda - 1) \cdot \mathrm{d}_{\tilde{G}_i}(u, u')}{\mathrm{d}_{\tilde{G}_i}(u, w)} \leq 1 + (\lambda - 1) \frac{\mathrm{d}_{\tilde{G}_i}(u, u')}{\mathrm{d}_{\tilde{G}_i}(u, w)} \leq 1 + (\lambda - 1) \frac{\mathrm{d}_{\tilde{G}_i}(u, u')}{t(1 + \epsilon)^i}$$

Now, it remains to show that in all cases (Case 1, 2.1 and 2.2) for the corresponding λ and an appropriate upper bound for $\mathrm{d}_{\tilde{G}_i}(u, u')$, we can bound the stretch of π' with $\alpha(1 + 24\epsilon)$. For the exact calculations, we defer to Appendix B. ◀

By repeatedly applying the shortcuts of Lemma 18 and concatenating them, we construct the alternative path π'' to get the following Lemma. The proof is in Appendix B.

▶ **Lemma 19.** *For every pair of vertices $u, v \in V$ with $\mathrm{d}_G(u, v) < \infty$, there exists a path π'' from u to v in the graph $H = (V', E \cup F)$, which consists of at most $(3t^2(\alpha + 1) + t) \ln(nW)$ edges, and has weight $\mathrm{w}_H(\pi'') \leq \alpha(1 + 24\epsilon) \cdot \mathrm{d}_G(u, v)$.*

Proof. Let π be the shortest path between u and v in G. The goal is to find a sequence of vertices $w_0, w_1, \ldots, w_k$ on π and a sequence of alternative paths $\pi'_0, \pi'_1, \ldots, \pi'_{k-1}$, where each $\pi'_i \in H$ is a path from w_i to w_{i+1}. Let π'' denote the path obtained by concatenating all paths π'_i. Let $w_0 = u$ and let w_i for $1 \leq i \leq k$ be the vertex on π, found as described in Lemma 18, with k chosen such that $w_k = v$, and Δ defined as $\Delta = \frac{\mathrm{d}_G(w_i, v)}{t^2}$ in order to find the vertex w_{i+1}.

From Lemma 18, we know that each alternative path π'_i has at most $3t(\alpha + 1) + 1$ hops. Thus, the concatenation of all alternative paths consists of at most $k(3t(\alpha + 1) + 1)$ edges. Likewise, Lemma 18 ensures that when using a shortcut π'_i we skip a distance of $\Delta t = \frac{\mathrm{d}_G(w_i, v)}{t^2} t = \frac{\mathrm{d}_G(w_i, v)}{t}$, and we know that the remaining distance is:

$$\mathrm{d}_G(w_{i+1}, v) \leq \mathrm{d}_G(w_i, v) - \frac{\mathrm{d}_G(w_i, v)}{t} = \left(1 - \frac{1}{t}\right) \mathrm{d}_G(w_i, v).$$

Further, we can say that for every $0 \le i \le k$:

$$\begin{aligned} \mathrm{d}_G(w_i, v) &\le \left(1 - \frac{1}{t}\right) \mathrm{d}_G(w_{i-1}, v) \\ &\le \left(1 - \frac{1}{t}\right)\left(1 - \frac{1}{t}\right) \mathrm{d}_G(w_{i-2}, v) \le \cdots \le \left(1 - \frac{1}{t}\right)^i \mathrm{d}_G(u, v). \end{aligned}$$

To analyze how many of such shortcuts are needed in order to reach v, we need to determine for which k it is ensured that the remaining distance satisfies $\mathrm{d}_G(w_k, v) \le (1-\frac{1}{t})^k \mathrm{d}_G(u, v) < 1$. Since $\mathrm{d}_G(u, v) \le (n-1)W < nW$, it follows that

$$1 > \frac{1}{nW} \mathrm{d}_G(u, v) = \left(\frac{1}{e}\right)^{\ln(nW)} \mathrm{d}_G(u, v).$$

Noting that $(1 - \frac{1}{t})^t \le \frac{1}{e}$, this yields

$$\left(\frac{1}{e}\right)^{\ln(nW)} \mathrm{d}_G(u, v) \ge \left(1 - \frac{1}{t}\right)^{t \cdot \ln(nW)} \mathrm{d}_G(u, v).$$

Thus $\mathrm{d}_G(w_k, v) < 1$ holds, for every $k \ge t \cdot \ln(nW)$. Therefore the path π'' contains at most $k(3t(\alpha + 1) + 1) = \ln(nW)t(t(\alpha + 1) + 1) = \ln(nW)(3t^2(\alpha + 1) + t)$ many edges.

We can bound the weight of the path π'' by $\mathrm{w}_H(\pi'') \le \alpha(1 + 24\epsilon) \cdot \mathrm{d}_G(u, v)$ as follows. Since Lemma 18 guarantees that each alternative path π'_i has weight $\mathrm{w}_H(\pi'_i) \le \alpha(1 + 24\epsilon) \cdot \mathrm{d}_G(w_i, w_{i+1})$, and all w_i lie on the shortest path between u and v we have:

$$\begin{aligned} \mathrm{w}_H(\pi'') = \sum_{i=0}^{k-1} \mathrm{w}_H(\pi'_i) &\le \sum_{i=0}^{k-1} \alpha(1 + 24\epsilon) \cdot \mathrm{d}_G(w_i, w_{i+1}) \\ &= \alpha(1 + 24\epsilon) \cdot \sum_{i=0}^{k-1} \mathrm{d}_G(w_i, w_{i+1}) = \alpha(1 + 24\epsilon) \cdot \mathrm{d}_G(u, v). \end{aligned}$$ ◀

References

1. 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.
2. Uri Ben-Levy and Merav Parter. New (α, β) spanners and hopsets. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1695–1714. SIAM, 2020. doi:10.1137/1.9781611975994.104.
3. Aaron Bernstein. Fully dynamic (2+ ε) approximate all-pairs shortest paths with fast query and close to linear update time. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 693–702. IEEE, 2009. doi:10.1109/FOCS.2009.16.
4. 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.
5. 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. doi:10.1109/FOCS57990.2023.00046.
6. Nairen Cao and Jeremy T Fineman. Parallel exact shortest paths in almost linear work and square root depth. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4354–4372. SIAM, 2023. doi:10.1137/1.9781611977554.CH166.
7. 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. ACM, 2020. doi:10.1145/3357713.3384270.

8 Nairen Cao, Jeremy T Fineman, and Katina Russell. Improved work span tradeoff for single source reachability and approximate shortest paths. In *Proceedings of the 32nd ACM Symposium on Parallelism in Algorithms and Architectures*, pages 511–513. ACM, 2020. doi:10.1145/3350755.3400222.

9 Keren Censor-Hillel, Michal Dory, Janne H. Korhonen, and Dean Leitersdorf. Fast approximate shortest paths in the congested clique. *Distributed Comput.*, 34(6):463–487, 2021. doi:10.1007/s00446-020-00380-5.

10 Edith Cohen. Polylog-time and near-linear work approximation scheme for undirected shortest paths. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of Computing*, pages 16–26, 1994. doi:10.1145/195058.195089.

11 Dorit Dor, Shay Halperin, and Uri Zwick. All-pairs almost shortest paths. *SIAM J. Comput.*, 29(5):1740–1759, 2000. doi:10.1137/S0097539797327908.

12 Michael Elkin and Ofer Neiman. Efficient algorithms for constructing very sparse spanners and emulators. In *Proceedings of the 2017 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 652–669, 2017. doi:10.1137/1.9781611974782.41.

13 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.

14 Michael Elkin and Ofer Neiman. Linear-size hopsets with small hopbound, and constant-hopbound hopsets in rnc. In *The 31st ACM on Symposium on Parallelism in Algorithms and Architectures, SPAA 2019, Phoenix, AZ, USA, June 22-24, 2019*, SPAA '19, pages 333–341, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3323165.3323177.

15 Michael Elkin and Ofer Neiman. Near-additive spanners and near-exact hopsets, A unified view. *Bull. EATCS*, 130, 2020. URL: http://bulletin.eatcs.org/index.php/beatcs/article/view/608/624.

16 Michael Elkin and David Peleg. $(1+\epsilon,\beta)$-spanner constructions for general graphs. *SIAM Journal on Computing*, 33(3):608–631, 2004. doi:10.1137/S0097539701393384.

17 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. ACM, 2018. doi:10.1145/3188745.3188926.

18 Maximilian Probst Gutenberg and Christian Wulff-Nilsen. Deterministic algorithms for decremental approximate shortest paths: Faster and simpler. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2522–2541. SIAM, 2020. doi:10.1137/1.9781611975994.154.

19 Monika Henzinger, Sebastian Krinninger, and Danupon Nanongkai. A deterministic almost-tight distributed algorithm for approximating single-source shortest paths. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, STOC '16, pages 489–498. ACM, June 2016. doi:10.1145/2897518.2897638.

20 Monika Henzinger, Sebastian Krinninger, and Danupon Nanongkai. Decremental single-source shortest paths on undirected graphs in near-linear total update time. *J. ACM*, 65(6), November 2018. doi:10.1145/3218657.

21 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.

22 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.

23 Shimon Kogan and Merav Parter. New diameter-reducing shortcuts and directed hopsets: Breaking the $\sqrt{n}$ barrier. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1326–1341, 2022. doi:10.1137/1.9781611977073.55.

24 Shimon Kogan and Merav Parter. Having hope in missing spanners: New distance preservers and light hopsets. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4352–4374. SIAM, 2025. doi:10.1137/1.9781611978322.148.

25 Jakub Łącki and Yasamin Nazari. Near-optimal decremental hopsets with applications. In *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, Paris, France, July 4-8, 2022*, LIPIcs, pages 86:1–86:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ICALP.2022.86.

26 Gary L Miller, Richard Peng, Adrian Vladu, and Shen Chen Xu. Improved parallel algorithms for spanners and hopsets. In *Proceedings of the 27th ACM Symposium on Parallelism in Algorithms and Architectures*, pages 192–201. ACM, 2015. doi:10.1145/2755573.2755574.

27 Ofer Neiman and Idan Shabat. A unified framework for hopsets. In *30th Annual European Symposium on Algorithms (ESA 2022)*, pages 81–1. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ESA.2022.81.

28 David Peleg and Alejandro A Schäffer. Graph spanners. *Journal of graph theory*, 13(1):99–116, 1989. doi:10.1002/JGT.3190130114.

29 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.

A Omitted Proofs from Section 2.1

We start by giving the proof for Observation 11.

Proof. This follows immediately from the construction of $\tilde{G}_i$, where each edge of G satisfying Eq. 1 is replaced by a sequence of new edges and intermediate vertices in $\tilde{G}_i$. Thus, the path $\tilde{\pi}$ can be obtained from π by replacing each such edge with its corresponding sequence of new edges and vertices. Therefore, $\tilde{\pi}$ connects u to v in $\tilde{G}_i$ and the vertex set of $\tilde{\pi}$ consists of the vertices of π, together with any intermediate vertices introduced during the subdivision step.

Each sequence of new edges in $\tilde{G}_i$, replacing an edge, has total weight equal to that of the original edge in G. Thus $\mathrm{w}_{\tilde{G}_i}(\tilde{\pi}) = \mathrm{w}_G(\pi)$ holds. ◀

Next, we will prove Claim 12.

Proof. Let $\tilde{\pi}$ be the shortest path from u to v in $\tilde{G}_i$, which by Observation 11 has weight equal to $\mathrm{d}_G(u,v)$. Consider an edge $\tilde{e}$ on $\tilde{\pi}$. If $\tilde{e} \notin E$, then by construction of $\tilde{G}_i$, $\tilde{e}$ is one of $k = \left\lceil \frac{\mathrm{w}_G(e)}{\epsilon(1+\epsilon)^i} \right\rceil$ edges in the path that replaces some edge $e \in E$, that satisfied Eq. 1. Each of these k edges has weight $\frac{\mathrm{w}_G(e)}{k} \leq \epsilon(1+\epsilon)^i$. Otherwise, if $\tilde{e} \in E$ we know that $\tilde{e}$ doesn't satisfy Eq. 1. This implies $\mathrm{w}_{\tilde{G}_i}(\tilde{e}) \leq \epsilon(1+\epsilon)^{i+1}$ or $\mathrm{w}_{\tilde{G}_i}(\tilde{e}) > t(1+\epsilon)^{i+1}$. However, since $\mathrm{d}_{\tilde{G}_i}(u,v) \leq t(1+\epsilon)^{i+1}$, the weight of $\tilde{e}$ is bounded by $\mathrm{w}_{\tilde{G}_i}(\tilde{\pi}) \leq t(1+\epsilon)^{i+1}$, and thus we know that $\tilde{e}$ has weight at most $\epsilon(1+\epsilon)^i$. ◁

The following proves Claim 13.

Proof. An edge $e \in E$ is subdivided if its weight satisfies Eq. 1, i.e. $\epsilon(1+\epsilon)^i < \mathrm{w}_G(e) \leq t(1+\epsilon)^{i+1}$. In this case, it is replaced by $k = \left\lceil \frac{\mathrm{w}_G(e)}{\epsilon(1+\epsilon)^i} \right\rceil$ new edges. Since $\mathrm{w}_G(e) \leq t(1+\epsilon)^{i+1}$, e is divided into at most $\left\lceil \frac{t(1+\epsilon)^{i+1}}{\epsilon(1+\epsilon)^i} \right\rceil = \left\lceil \frac{t(1+\epsilon)}{\epsilon} \right\rceil \leq \frac{t}{\epsilon}(1+\epsilon) + 1 \leq \frac{2t}{\epsilon} + 1$ edges in $\tilde{E}_i$. Hence each edge in E contributes to $O(\frac{t}{\epsilon})$ edges, so the total number of edges in $\tilde{G}_i$ is bounded by $|\tilde{E}_i| = O(m\frac{t}{\epsilon})$, and the number of vertices by $|\tilde{V}_i| = O(n + m\frac{t}{\epsilon})$. ◁

We conclude this section by providing the proof of Lemma 14.

Proof. By definition, an edge $e \in E$ is subdivided at scale i if its weight satisfies Eq. 1, i.e. $\epsilon(1+\epsilon)^i < \mathrm{w}_G(e) \le t(1+\epsilon)^{i+1}$. For a fixed edge $e \in E$, we want to bound the number of integer values i for which this inequality holds. Rewriting the bounds we obtain:

$$\epsilon(1+\epsilon)^i < \mathrm{w}_G(e) \iff i < \log_{1+\epsilon}\left(\frac{\mathrm{w}_G(e)}{\epsilon}\right)$$
$$\mathrm{w}_G(e) \le t(1+\epsilon)^{i+1} \iff i \ge \log_{1+\epsilon}\left(\frac{\mathrm{w}_G(e)}{t}\right) - 1.$$

Thus the number of integer values of i satisfying Eq. 1 is less than $\log_{1+\epsilon}(\frac{\mathrm{w}_G(e)}{\epsilon}) - (\log_{1+\epsilon}(\frac{\mathrm{w}_G(e)}{t}) - 1) = \log_{1+\epsilon}(\frac{t}{\epsilon}) + 1$. Therefore, each edge is subdivided in at most $\log_{1+\epsilon}(\frac{t}{\epsilon}) + 1$ distinct distance scales. ◀

B Omitted Proofs from Section 2.3

First, we provide the full proof of Claim 17.

Proof. Let π_i be the shortest path from u to v in the emulator M_i, which is by construction also in F. By definition, this path has weight $\mathrm{w}_{M_i}(\pi_i) = \mathrm{d}_{M_i}(u,v) \le \alpha \cdot \mathrm{d}_{G_i}(u,v) + \beta \le \alpha l + \beta$. Since all edges have weight at least 1, this yields

$$|\pi_i| \le \alpha l + \beta.$$

Recall that in M_i' all edge weights are scaled by the factor $(1+\epsilon)^{i+1}$. Therefore the weight of the path π_i in H is $\mathrm{w}_H(\pi_i) = \mathrm{w}_{M_i'}(\pi_i) = (1+\epsilon)^{i+1}\, \mathrm{w}_{M_i}(\pi_i)$ and can be bounded by

$$\mathrm{w}_H(\pi_i) \le (1+\epsilon)^{i+1}(\alpha l + \beta).$$ ◁

Now, we give the part omitted from the proof of Lemma 18, where it remained to show that in all cases the path π' achieved a stretch of at most $\alpha(1+24\epsilon)$.

Proof. First, we consider the simple Case 2.1. Here we derived the stretch $\lambda_{2.1} = (1+\epsilon)(\alpha+\beta)$ for the path π_{pre}. Because in this case, we have $\mathrm{d}_{\tilde{G}_i}(u,u') < (1+\epsilon)^i$, the stretch of π' can be bounded as follows

$$\begin{aligned}\frac{\mathrm{w}_H(\pi')}{\mathrm{d}_{\tilde{G}_i}(u,w)} &\le 1 + ((\lambda_{2.1} - 1)\,\frac{(1+\epsilon)^i}{t(1+\epsilon)^i} \le 1 + ((1+\epsilon)(\alpha+\beta) - 1)\,\frac{(1+\epsilon)^i}{t(1+\epsilon)^i}\\ &= 1 + ((1+\epsilon)\alpha + (1+\epsilon)\beta - 1)\,\frac{1}{t} = 1 + ((1+\epsilon)\alpha - 1)\,\frac{1}{t} + (1+\epsilon)\beta\frac{1}{t}\end{aligned}$$

As $t = \max(\frac{1}{\epsilon}, \frac{\beta}{\epsilon})$, we have $\frac{1}{t} \le \epsilon \le 1$ and $\frac{\beta}{t} \le \epsilon$. This yields

$$\begin{aligned}\frac{\mathrm{w}_H(\pi')}{\mathrm{d}_{\tilde{G}_i}(u,w)} &\le 1 + (1+\epsilon)\alpha - 1 + (1+\epsilon)\epsilon = \alpha + \alpha\epsilon + \epsilon + \epsilon^2 \le \alpha(1+\epsilon+\epsilon+\epsilon^2)\\ &\le \alpha(1+3\epsilon).\end{aligned}$$

Next, we examine the remaining cases. In Case 1 we know that $\mathrm{d}_{\tilde{G}_i}(u,u') \le p(1+\epsilon)^{i+1}$ and derived the stretch $\lambda_1 = (1+\epsilon)(\alpha + \frac{\beta}{p})$. In Case 2.2 we have $\mathrm{d}_{\tilde{G}_i}(u,u') \le p(1+\epsilon)^{i+1} + (1+\epsilon)^i$ and the stretch $\lambda_{2.2} = (1+\epsilon)(\alpha + \frac{\alpha}{p} + 2\frac{\beta}{p})$. Since $\lambda_{2.2} > \lambda_1$ and $\mathrm{d}_{\tilde{G}_i}(u,u')$ in Case 2.2 is larger than in Case 1, we can bound the stretch of π' for these cases as follows:

$$\begin{aligned}\frac{\mathrm{w}_H(\pi')}{\mathrm{d}_{\tilde{G}_i}(u,w)} &\leq 1 + (\lambda_1 - 1)\,\frac{p(1+\epsilon)^{i+1}}{t(1+\epsilon)^i} \leq 1 + (\lambda_{2.2} - 1)\frac{p(1+\epsilon)^{i+1} + (1+\epsilon)^i}{t(1+\epsilon)^i}\\ &= 1 + (\lambda_{2.2} - 1)\left(\frac{p}{t}(1+\epsilon) + \frac{1}{t}\right) = 1 + (\lambda_{2.2} - 1)\frac{p}{t}(1+\epsilon) + (\lambda_{2.2} - 1)\frac{1}{t}.\end{aligned}$$

The term $(\lambda_2 - 1)\frac{p}{t}(1+\epsilon)$, can be expressed as

$$\begin{aligned}(\lambda_2 - 1)\frac{p}{t}(1+\epsilon) &= \left((1+\epsilon)(\alpha + \frac{\alpha}{p} + 2\frac{\beta}{p}) - 1\right)\frac{p}{t}(1+\epsilon)\\ &= \left(\alpha(1+\epsilon) + \frac{\alpha}{p}(1+\epsilon) + 2\frac{\beta}{p}(1+\epsilon) - 1\right)\frac{p}{t}(1+\epsilon)\\ &= (\alpha(1+\epsilon) - 1)\frac{p}{t}(1+\epsilon) + \frac{\alpha}{p}\frac{p}{t}(1+\epsilon)^2 + 2\frac{\beta}{p}\frac{p}{t}(1+\epsilon)^2.\end{aligned}$$

Given that $\frac{p}{t} \leq \frac{t+t\epsilon}{t} = 1+\epsilon$, $\frac{\alpha}{p}\frac{p}{t} \leq \alpha\epsilon$ and $\frac{\beta}{p}\frac{p}{t} \leq \epsilon$, we get

$$\begin{aligned}(\lambda_2 - 1)\frac{p}{t}(1+\epsilon) &\leq (\alpha(1+\epsilon) - 1)(1+\epsilon)^2 + \alpha\epsilon(1+\epsilon)^2 + 2\epsilon(1+\epsilon)^2\\ &= \alpha(1+\epsilon)^3 - (1+\epsilon)^2 + \alpha\epsilon(1+\epsilon)^2 + 2\epsilon(1+\epsilon)^2\\ &= \alpha(1+\epsilon)^3 - 1 - 2\epsilon - \epsilon^2 + \alpha\epsilon(1+\epsilon)^2 + 2\epsilon + 4\epsilon^2 + 2\epsilon^3\\ &= -1 + \alpha(1+\epsilon)^3 + \alpha\epsilon(1+\epsilon)^2 + 3\epsilon^2 + 2\epsilon^3.\end{aligned}$$

Since $\frac{1}{p} \leq 1$, $\frac{\beta}{t} \leq \epsilon$ and $\frac{1}{t} \leq \epsilon$, the term $(\lambda_2 - 1)\frac{1}{t}$ is bounded by

$$\begin{aligned}(\lambda_2 - 1)\frac{1}{t} &\leq \lambda_2\frac{1}{t} = (1+\epsilon)\left(\alpha + \frac{\alpha}{p} + 2\frac{\beta}{p}\right)\frac{1}{t}\\ &\leq (1+\epsilon)(2\alpha + 2\beta)\frac{1}{t} = (1+\epsilon)\left(2\alpha\frac{1}{t} + 2\frac{\beta}{t}\right)\\ &\leq (1+\epsilon)(2\alpha\epsilon + 2\epsilon) = 2\alpha\epsilon + 2\epsilon + 2\alpha\epsilon^2 + 2\epsilon^2.\end{aligned}$$

Using these two bounds, we obtain

$$\begin{aligned}\frac{\mathrm{w}_H(\pi')}{\mathrm{d}_{\tilde{G}_i}(u,w)} &\leq 1 + (\lambda_2 - 1)\frac{p}{t}(1+\epsilon) + (\lambda_2 - 1)\frac{1}{t}\\ &\leq 1 - 1 + \alpha(1+\epsilon)^3 + \alpha\epsilon(1+\epsilon)^2 + 3\epsilon^2 + 2\epsilon^3 + 2\alpha\epsilon + 2\epsilon + 2\alpha\epsilon^2 + 2\epsilon^2\\ &\leq \alpha\left((1+\epsilon)^3 + \epsilon(1+\epsilon)^2 + 4\epsilon + 7\epsilon^2 + 2\epsilon^3\right)\\ &= \alpha\left(1 + 3\epsilon + 3\epsilon^2 + \epsilon^3 + \epsilon + 2\epsilon^2 + \epsilon^3 + 4\epsilon + 7\epsilon^2 + 2\epsilon^3\right)\\ &\leq \alpha(1+24\epsilon).\end{aligned}$$

Hence, the path π' achieves the claimed stretch of at most $\alpha(1+24\epsilon)$ in all cases. ◀