

A Unified Framework for Light Spanners

Hung Le¹ and Shay Solomon²

¹University of Massachusetts Amherst

²Tel Aviv University

Abstract

Seminal works on *light* spanners over the years provide spanners with optimal *lightness* in various graph classes,¹ such as in general graphs [17], Euclidean spanners [26] and minor-free graphs [10]. Three shortcomings of previous works on light spanners are: (i) The runtimes of these constructions are almost always sub-optimal, and usually far from optimal. (ii) These constructions are optimal in the standard and crude sense, but not in a refined sense that takes into account a wider range of involved parameters. (iii) *The techniques are **ad hoc** per graph class, and thus can't be applied broadly.*

This work aims at addressing these shortcomings by presenting a *unified framework* of light spanners in a variety of graph classes. Informally, the framework boils down to a *transformation* from sparse spanners to light spanners; since the state-of-the-art for sparse spanners is much more advanced than that for light spanners, such a transformation is powerful. First, we apply our framework to design *fast* constructions with *optimal lightness* for several graph classes. Among various applications, we highlight the following (for simplicity assume $\epsilon > 0$ is fixed):

- In *low-dimensional Euclidean spaces*, we present an $O(n \log n)$ -time construction of $(1 + \epsilon)$ -spanners with lightness and degree both bounded by constants in the *algebraic computation tree (ACT)* (or *real-RAM*) model, which is the basic model used in Computational Geometry. The previous state-of-the-art runtime in this model for constant lightness (even for unbounded degree) was $O(n \log^2 n / \log \log n)$, whereas $O(n \log n)$ -time spanner constructions with constant degree (and $O(n)$ edges) are known for years. Our construction is optimal with respect to all the involved quality measures — runtime, lightness and degree — and it resolves a major problem in the area of geometric spanners, which was open for three decades (cf. [15, 3, 40, 53]).

Second, we apply our framework to achieve *more refined* optimality bounds for several graph classes, i.e., the bounds remain optimal when taking into account a *wider range of involved parameters*, most notably ϵ . Our new constructions are significantly better than the state-of-the-art *for every examined graph class*. Among various applications, we highlight the following (now $\epsilon > 0$ is any parameter):

- For K_r -minor-free graphs, we provide a $(1 + \epsilon)$ -spanner with lightness $\tilde{O}_{r,\epsilon}(\frac{r}{\epsilon} + \frac{1}{\epsilon^2})$, where $\tilde{O}_{r,\epsilon}$ suppresses polylog factors of $1/\epsilon$ and r , improving the lightness bound $\tilde{O}_{r,\epsilon}(\frac{r}{\epsilon^3})$ of Borradaile, Le and Wulff-Nilsen [10]. We complement our upper bound with a highly nontrivial lower bound construction, for which any $(1 + \epsilon)$ -spanner must have lightness $\Omega(\frac{r}{\epsilon} + \frac{1}{\epsilon^2})$. Interestingly, our lower bound is realized by a geometric graph in $\mathbb{R}^2$. Also, the quadratic dependency on $1/\epsilon$ that we prove is surprising, as prior work suggested that the dependency on ϵ should be around $1/\epsilon$.

¹The *lightness* is a normalized notion of weight: a graph's lightness is the ratio of its weight to the MST weight.

Contents

1	Introduction	1
1.1	Research Agenda: From Sparse to Light Spanners	3
1.2	Our Contribution	4
1.3	Our Unified Framework: Technical and Conceptual Highlights	6
2	Preliminaries	11
3	Lightness Lower Bounds	12
I	Our Unified Framework: Applications (Section 4 and Section 5)	17
4	Applications of the Unified Framework: Fast Constructions	17
4.1	Euclidean Spanners and UDG Spanners	17
4.2	General Graphs	20
4.3	Minor-free Graphs	21
5	Applications of the Unified Framework: Fine-Grained Optimality	22
5.1	General Sparse Spanner Oracles	22
5.2	Constructing General Sparse Spanner Oracles	25
5.2.1	General graphs and high dimensional metric spaces: Proof of Theorem 1.6 and Theorem 1.8	25
5.2.2	Steiner Euclidean Spanners	26
5.3	Light Spanners for Minor-Free Graphs	27
II	Our Unified Framework: The Proof (Section 6 — Section 12)	28
6	Unified Framework: Technical Setup	28
6.1	The Framework	28
6.2	Designing A Potential Function	31
6.3	Summary	34
7	Fast Construction: Proof of Theorem 1.10(1)	35
7.1	Constructing Level- $(i + 1)$ Clusters	36
7.2	Constructing H_i : Proof of Theorem 1.10(1)	38
8	Clustering: Proof of Lemma 7.3	42
8.1	Properties of $\mathbb{X}$	48
8.2	Constructing a Partition of $\mathcal{V}_i$	48
9	Light Spanners for Minor-free Graphs in Linear Time	52
10	Fine-Grained Optimal Lightness: Proof of Theorem 1.10(2)	52

11 Clustering for Stretch $t \geq 2$: Proof of Lemma 10.3(1)	55
11.1 Constructing H_i : Proof of Lemma 10.3 for $t \geq 2$	56
11.2 Clustering	58
11.2.1 Constructing $\mathbb{X}$	63
11.2.2 Constructing the partition of $\mathcal{E}_i$: Proof of Lemma 11.1	65
12 Clustering for Stretch $t = 1 + \epsilon$	67
12.1 Constructing H_i : Proof of Lemma 10.3 for $t = 1 + \epsilon$	68
12.2 Clustering	69
12.2.1 Constructing $\mathbb{X}$ and the partition of $\mathcal{E}_i$: Proof of Lemma 12.1	71

1 Introduction

For a weighted graph $G = (V, E, w)$ and a *stretch parameter* $t \geq 1$, a subgraph $H = (V, E')$ of G is called a *t-spanner* if $d_H(u, v) \leq t \cdot d_G(u, v)$, for every $e = (u, v) \in E$, where $d_G(u, v)$ and $d_H(u, v)$ are the distances between u and v in G and H , respectively. Graph spanners were introduced in two celebrated papers from 1989 [54, 55] for unweighted graphs, where it is shown that for any n -vertex graph $G = (V, E)$ and integer $k \geq 1$, there is an $O(k)$ -spanner with $O(n^{1+1/k})$ edges. We shall sometimes use a normalized notion of size, *sparsity*, which is the ratio of the size of the spanner to the size of a spanning tree, namely $n - 1$. Since then, graph spanners have been extensively studied, both for general weighted graphs and for restricted graph families, such as Euclidean spaces and minor-free graphs. In fact, spanners for Euclidean spaces—*Euclidean spanners*—were studied implicitly already in the pioneering SoCG’86 paper of Chew [19], who showed that any 2-dimensional Euclidean space admits a spanner of $O(n)$ edges and stretch $\sqrt{10}$, and later improved the stretch to 2 [20].

As with the sparsity parameter, its weighted variant—lightness—has been extremely well-studied; the *lightness* is the ratio of the weight of the spanner to $w(MST(G))$. Seminal works on *light* spanners over the years provide spanners with optimal *lightness* in various graph classes, such as in general graphs [17], Euclidean spanners [26] and minor-free graphs [10]. **Despite the large body of work on light spanners, the stretch-lightness tradeoff is not nearly as well-understood as the stretch-sparsity tradeoff**, and the intuitive reason behind that is clear: Lightness seems inherently more challenging to optimize than sparsity, since different edges may contribute disproportionately to the overall lightness due to differences in their weights. The three shortcomings of light spanners that emerge, when considering the large body of work in this area, are: (i) The runtimes of these constructions are usually far from optimal. (ii) These constructions are optimal in the standard and crude sense, but not in a refined sense that takes into account a wider range of involved parameters, most notably ϵ , but also other parameters, such as the dimension (in Euclidean spaces) or the minor size (in minor-free graphs). (iii) **The techniques are ad hoc per graph class, and thus can’t be applied broadly** (e.g., some require large stretch and are thus suitable to general graphs, while others are naturally suitable to stretch $1 + \epsilon$).

In this work, we are set out to address these shortcomings by presenting a *unified framework* of light spanners in a variety of graph classes. Informally, the framework boils down to a *transformation* from sparse spanners to light spanners; since the state-of-the-art for sparse spanners is much more advanced than that for light spanners, such a transformation is powerful.

Our ultimate goal is to bridge the gap in the understanding between light spanners and sparse spanners. This gap is prominent when considering (i) the construction time of light versus sparse spanners, and (ii) a *fine-grained* optimality of the lightness. In terms of (ii), the state-of-the-art spanner constructions for general graphs, as well as for most restricted graph families, incur a (multiplicative) $(1 + \epsilon)$ -factor slack on the stretch with a *suboptimal* dependence on ϵ as well as other parameters in the lightness bound. In this work, we present new spanner constructions, all of which are derived as applications and implications from a unified framework developed in this paper.

- In terms of (i), i.e., runtime, our constructions are significantly faster than the state-of-the-art for every examined graph class; moreover, our runtimes are near-linear or linear and usually optimal. Our main result in this context is an $O(n \log n)$ time algorithm in the ACT model for constructing a Euclidean spanner with constant lightness and degree.
- In terms of (ii), i.e., fine-grained optimality, our constructions are significantly better than the state-of-the-art for every examined graph class; our main result in this context is for minor-free graphs, where we achieve tight dependencies on both ϵ and the minor size – the upper bound follows as an application of the unified framework, and the lower bound is obtained by different means.

We now highlight three completely different yet well-studied settings to which our framework applies.

Fast construction of Euclidean spanners in the algebraic computation tree (ACT) model.

Spanners have had special success in geometric settings, especially in low-dimensional Euclidean spaces. The reason Euclidean spanners have been extensively studied over the years — in both theory and practice — is that one can achieve stretch arbitrarily close to 1. The *algebraic computation tree (ACT)* introduced by Ben-Or [6] model is used extensively in computational geometry, and in the area of Euclidean spanners in particular; it is intimately related to the real RAM model. (The reader can refer to [6] and Chapter 3 in the book [53] for a detailed description of ACT model.) Computing $(1 + \epsilon)$ -spanners for point sets in $\mathbb{R}^d$, $d = O(1)$, in ACT model requires $\Omega(n \log n)$ time [18, 32]. Despite a large body of work on light Euclidean spanners [51, 15, 23, 26, 24, 25, 3, 56, 40, 53, 31, 50] since the late 80s, the following problem has been open for nearly three decades:

Question 1 (Question 22 in [53]). *Can one construct a Euclidean $(1 + \epsilon)$ -spanner with constant lightness and degree (and thus constant sparsity) in optimal time $O(n \log n)$ in the ACT model, for any fixed $\epsilon < 1$?*

While Question 1 asks for both constant lightness and degree, it is even not known how to achieve constant lightness only in $O(n \log n)$ time in the ACT model. The best-known algorithm has running time $O(n \frac{\log^2 n}{\log \log n})$ [40]. If one assumes *indirect addressing*, then there is an algorithm with running time $O(n \log n)$ [40]. But indirect addressing is a very strong operation: the lower bound of $\Omega(n \log n)$ in ACT model for Euclidean spanners mentioned earlier no longer applies. Some applications of light spanners [56, 22] require that they can be computed in $O(n \log n)$ time. Euclidean spanners of bounded degree have applications in designing routing schemes (see, e.g., [14, 37, 12]), and more generally, the degree of the spanner determines the local memory constraints when using spanners also for other purposes, such as constructing network synchronizers and efficient broadcast protocols.

Fast construction of general weighted graphs. Althöfer et al [2] shown that for every n -vertex *weighted* graph $G = (V, E, w)$ and integer $k \geq 1$, there is a *greedy* algorithm for constructing a $(2k - 1)$ -spanner with $O(n^{1+1/k})$ edges, which is optimal under Erdős’ girth conjecture. Moreover, there is an $O(m)$ -time algorithm for constructing $(2k - 1)$ -spanners in unweighted graphs with sparsity $O(n^{\frac{1}{k}})$ [41]. Therefore, not only is the stretch-sparsity tradeoff in general graphs optimal (up to Erdős’ girth conjecture), one can achieve it in optimal time. For weighted graphs, one can construct $(2k - 1)$ -spanners with sparsity $O(kn^{\frac{1}{k}})$ within time $O(km)$ [5, 57]. On the other hand, the best running time for achieving lightness bound $O(n^{1/k})$ for stretch $(2k - 1)(1 + \epsilon)$ for a *fixed* ϵ is super-quadratic in n : $O(n^{2+1/k+\epsilon'})$ [1] for any fixed constant $\epsilon' < 1$. Other faster constructions have a worst dependency on n and k [30].

Question 2. *Can one construct a $(2k - 1) \cdot (1 + \epsilon)$ -spanner in general weighted graphs with lightness $O(n^{1/k})$, within (nearly) linear time for any fixed $\epsilon < 1$?*

Fine-grained lightness bound for minor-free graphs. The gap between sparsity and lightness is prominent in minor-free graphs, for stretch $1 + \epsilon$. Indeed, minor-free graphs are sparse to begin with, and the sparsity is trivially $\tilde{\Theta}(r)$. On the other hand, for lightness, bounds are much more interesting. Borradaile, Le, and Wulff-Nilsen [10] showed that the greedy $(1 + \epsilon)$ -spanners of K_r -minor-free graphs have lightness $\tilde{O}_{r,\epsilon}(\frac{r}{\epsilon^3})$, where the notation $\tilde{O}_{r,\epsilon}(\cdot)$ hides polylog factors of r and $\frac{1}{\epsilon}$. Moreover, this is the state-of-the-art lightness bound also in some sub-classes of minor-free graphs, particularly bounded treewidth graphs. Past works provided strong evidence that the dependence of lightness on $1/\epsilon$ of $(1 + \epsilon)$ -spanners should be *linear*: $O(\frac{1}{\epsilon})$ in planar graphs by Althöfer et al. [2], $O(\frac{g}{\epsilon})$ in bounded genus graphs by Grigni [39], and $\tilde{O}_r(\frac{r \log n}{\epsilon})$ in K_r -minor-free graphs by Grigni and Sissokho [38].

Question 3. *Is there a $(1 + \epsilon)$ -spanner of lightness $O(1/\epsilon)$ for K_r -minor-free graphs for a fixed r ?*

Other open problems. There are several settings where the *fast constructions* of light spanners for *fixed* ϵ remains open.

- **Unit disk graph.** There is a significant gap between the fastest constructions of sparse versus light spanners in UDGs. Fürer and Kasiviswanathan [35] showed that *sparse* $(1 + \epsilon)$ -spanners for UDGs can be built in nearly linear time when $d = 2$, and in subquadratic time when d is a constant of value at least 3. However, no $o(n^2)$ -time $(1 + \epsilon)$ -spanner construction for UDGs with a nontrivial lightness bound is known, even for $d = 2$. Can we construct a light $(1 + \epsilon)$ -spanner for UDGs in $O(n \log n)$ time for $d = 2$, and in truly subquadratic time for general d ?
- **Minor-free graphs.** The fastest algorithm for constructing light spanners in K_r -minor-free graphs is greedy [2] with quadratic running time $\tilde{O}_r(n^2 r^2)$. Can we construct a light $(1 + \epsilon)$ -spanner for K_r -minor-free graphs in nearly linear time?

For *fine-grained lightness bounds*, there are two additional settings where the lightness bounds are not well-understood.

- **General graphs.** While the stretch-sparsity tradeoff for spanners of general graphs is resolved up to the girth conjecture, the stretch-lightness tradeoff, on the other hand, is still far from being resolved. A long line of research [2, 15, 30, 17, 34] over the past three decades leads to a $(2k - 1) \cdot (1 + \epsilon)$ -spanner with lightness $O(n^{1/k}(1/\epsilon)^{3+2/k})$ [17, 34]. While the dependence on n and k are optimal assuming Erdős' girth conjecture, the dependence on $1/\epsilon$ is super-cubic. Can we reduce the dependency of the lightness on ϵ to linear?
- **Euclidean Steiner spanners in low dimensional spaces.** Le and Solomon [50] studied *Steiner spanners*, namely, spanners that are allowed to use *Steiner points*, which are additional points that are not part of the input point set. It was shown there that Steiner points can be used to improve the sparsity quadratically, i.e., to $O(\epsilon^{-\frac{d+1}{2}})$, which was shown to be tight for dimension $d = 2$ in [50], and for any $d = O(1)$ by Bhore and Tóth [8]. An important question left open in [50] is whether one could use Steiner points to improve the lightness bound *quadratically* to $O(\epsilon^{-d/2})$ for any dimension d . Previous results either have a dependency on the *spread* of the metric [47] which could be huge, or only work for $d = 2$ [7].

1.1 Research Agenda: From Sparse to Light Spanners

Thus far we exemplified the statement that the stretch-lightness tradeoff is not as well-understood as the stretch-sparsity tradeoff. As we showed, this lack of understanding is prominent when considering (i) the *construction time*, and (ii) fine-grained dependencies. This statement is not to underestimate in any way the exciting line of work on light spanners, but rather to call for attention to the important research agenda of narrowing this gap and ideally closing it.

All questions regarding fast constructions of light spanners ask the same thing: Can one achieve *fast constructions* of *light* spanners that match the analogous results for *sparse* spanners?

Goal 1. *Achieve fast constructions of light spanners that match the corresponding constructions of sparse spanners. In particular, achieve (nearly) linear-time constructions of spanners with optimal lightness for basic graph families, such as the ones covered in the aforementioned questions.*

A fine-grained optimization of the stretch-lightness tradeoff, which takes into account the exact dependencies on ϵ and the other involved parameters, is a highly challenging goal.

Goal 2. *Achieve fine-grained optimality for light spanners in basic graph families.*

Some of the papers on light spanners employ inherently different techniques than others, e.g., the technique of [17] requires large stretch while others are naturally suitable to stretch $1 + \epsilon$. Since the techniques in this area are ad hoc per graph class, they can't be applied broadly. A unified framework for light spanners would be of both theoretical and practical merit.

Goal 3. *Achieve a unified framework of light spanners.*

1.2 Our Contribution

Our work aims at meeting the above goals (Goal 1—Goal 3) by presenting a unified framework for optimal and fast constructions of light spanners in a variety of graph classes. Basically, we strive to translate results — in a unified manner — from sparse spanners to light spanners, without significant loss in any of the parameters. Our paper achieves Goal 1 and Goal 3 or achieves Goal 2 and Goal 3; achieving all three goals simultaneously is left open by our work.

We also answer almost all the aforementioned open problems, either positively or negatively. In particular, we answer Question 1 and Question 2 positively, and Question 3 negatively. For other open problems, we completely resolve them in the affirmative.

Two of our results are particularly surprising. First, we show that the optimal lightness bound of $(1 + \epsilon)$ for K_r -minor-free graphs is $\tilde{\Theta}_{r,\epsilon}(\frac{r}{\epsilon} + \frac{1}{\epsilon^2})$ (Theorem 1.5). That is, the lightness dependency on $1/\epsilon$ is quadratic, despite ample evidence [2, 39, 38] of a linear dependency on $1/\epsilon$ in subclasses of minor-free graphs. In particular, our result negatively settles Question 3. Second, we construct light spanners in general graphs with near-optimal lightness in $O(m\alpha(m, n))$ time (Theorem 1.2); our algorithm is significantly faster than the best algorithms for sparse spanners with the same sparsity bound.

Fast constructions. We present a spanner construction that achieves constant lightness and degree, within optimal time of $O(n \log n)$ in ACT model; this proves the following theorem, which affirmatively resolves Question 1 which was open for three decades.

Theorem 1.1. *For any set P of n points in R^d , any $d = O(1)$ and any fixed $\epsilon > 0$, one can construct in the ACT model a $(1 + \epsilon)$ -spanner for P with constant degree and lightness within optimal time $O(n \log n)$.*

For general graphs we provide a nearly linear-time spanner construction with optimal lightness, assuming Erdős' girth conjecture (and up to the ϵ -dependency), thus answering Question 2.

Theorem 1.2. *For any edge-weighted graph $G(V, E)$, a stretch parameter $k \geq 2$ and an arbitrary small fixed $\epsilon < 1$, there is a deterministic algorithm that constructs a $(2k - 1)(1 + \epsilon)$ -spanner of G with lightness $O(n^{1/k})$ in $O(m\alpha(m, n))$ time, where $\alpha(\cdot, \cdot)$ is the inverse-Ackermann function.*

We remark that $\alpha(m, n) = O(1)$ when $m = \Omega(n \log^* n)$. Thus, the running time in Theorem 1.2 is linear in m in almost the entire regime of graph densities, i.e., except for very sparse graphs. The previous state-of-the-art runtime for the same lightness bound is super-quadratic [1]. Surprisingly, the result of Theorem 1.2 outperforms the analog result for sparse spanners in weighted graphs: for stretch $2k - 1$, the only spanner construction with sparsity $O(n^{1/k})$ is the greedy spanner, whose runtime is $O(mn^{1+\frac{1}{k}})$. Other results [1, 28] with stretch $(2k - 1)(1 + \epsilon)$ have (nearly) linear running time, but the sparsity is $O(n^{1/k} \log(k))$, which is worse than our lightness bound by a factor of $\log(k)$.

Subsequent work. In a subsequent and consequent follow-up to this work, the authors [48] used our framework here to present a fast construction of spanners with near-optimal *sparsity* and *lightness* for general graphs [48]. We also adapted and simplified our construction here to construct a *sparse* spanner (with unbounded lightness) in $O(m\alpha(m, n) + \text{SORT}(m))$ time in the pointer-machine model,

where $\text{SORT}(m)$ is the time to sort m integers. Even in a stronger Word RAM model, the best known algorithm for sorting m integers takes $O(m\sqrt{\log \log m})$ [42] expected time. Thus, the running time of the sparse spanner algorithm is still inferior to our running time in Theorem 1.2. In the Word RAM model, a linear time algorithm for constructing a sparse spanner was presented; we do not consider this model in our work here.

Our framework also resolves two open problems regarding fast constructions of light spanners in two different settings. In particular, we get an $O(n \log n)$ time algorithm for UDGs; the running time is optimal in the ACT model. For minor-free graphs, we get the first linear time algorithm, which significantly improves over the best known algorithm for this problem.

Theorem 1.3. *For any set P of n points in $\mathbb{R}^d$, any $d = O(1)$ and any fixed $\epsilon > 0$, one can construct a $(1 + \epsilon)$ -spanner of the UDG for P with constant sparsity and lightness. For $d = 2$, the construction runtime is $O(n \log n)$ in the ACT model; for $d = 3$, the runtime is $\tilde{O}(n^{4/3})$; and for $d \geq 4$, the runtime is $O(n^{2 - \frac{2}{\lceil d/2 \rceil + 1} + \delta})$ for any constant $\delta > 0$.*

Theorem 1.4. *For any K_r -minor-free graph G and any fixed $\epsilon > 0$, one can construct a $(1 + \epsilon)$ -spanner of G with lightness $O(r\sqrt{\log r})$ in $O(nr\sqrt{\log r})$ time.*

Fine-grained lightness bounds. The most important implication of our framework in terms of fine-grained lightness bounds is to minor-free graphs, where we obtain a tight dependence on ϵ in the lightness.

Theorem 1.5. *Any K_r -minor-free graph admits a $(1 + \epsilon)$ -spanner with lightness $\tilde{O}_{r,\epsilon}(\frac{r}{\epsilon} + \frac{1}{\epsilon^2})$ for any $\epsilon < 1$ and $r \geq 3$.*

Furthermore, for any fixed $r \geq 6$, any $\epsilon < 1$ and $n \geq r + (\frac{1}{\epsilon})^{\Theta(1/\epsilon)}$, there is an n -vertex graph G excluding K_r as a minor for which any $(1 + \epsilon)$ -spanner must have lightness $\Omega(\frac{r}{\epsilon} + \frac{1}{\epsilon^2})$.

The $\tilde{O}_{\epsilon,r}(\cdot)$ notation in Theorem 1.5 hides a poly-logarithmic factor of $1/\epsilon$ and r . Theorem 1.5 resolves Question 3 negatively. We remark that, in Theorem 1.5, the exponential dependence on $1/\epsilon$ in the lower bound on n is unavoidable since, if $n = \text{poly}(1/\epsilon)$, the result of [38] yields a lightness of $\tilde{O}_r(\frac{r}{\epsilon} \log(n)) = \tilde{O}_{r,\epsilon}(\frac{r}{\epsilon})$.

Interestingly, our lower bound applies to a geometric graph, where the vertices correspond to points in $\mathbb{R}^2$ and the edge weights are the Euclidean distances between the points. The construction is recursive. We start with a basic gadget and then recursively “stick” many copies of the same basic gadgets in a fractal-like structure. We use geometric considerations to show that any $(1 + \epsilon)$ -spanner must take every edge of this graph, whose total edge weight is $\Omega(1/\epsilon^2)w(\text{MST})$.

A prominent application of light spanners for K_r -minor-free graphs is to the Traveling Salesperson Problem (TSP). Theorem 1.5 implies a PTAS (polynomial time approximation scheme) with approximation $1 + \epsilon$ and running time $2^{1/\epsilon^3} n^{O(1)}$, improving upon the algorithm by Borradaile, Le, and Wulff-Nilsen [10] with running time $2^{O(1/\epsilon^4)} n^{O(1)}$. Our lower bound of Theorem 1.5 implies that to further improve the runtime for TSP one has to significantly deviate from the standard technique [27] that relies on light spanners.

Using our framework, we obtain near-optimal lightness bounds in two different settings: general graphs (Theorem 1.6) and Steiner Euclidean spanners (Theorem 1.7). Both results resolve two open problems mentioned above.

Theorem 1.6. *Given an edge-weighted graph $G(V, E)$ and two parameters $k \geq 1, \epsilon < 1$, there is a $(2k - 1)(1 + \epsilon)$ -spanner of G with lightness $O(g(n, k)/\epsilon)$ where $g(n, k)$ is the minimum sparsity of n -vertex graphs with girth $2k + 1$. As $g(n, k) = O(n^{1/k})$, the lightness is $O(n^{1/k}/\epsilon)$.*

The Erdős' girth conjecture implies that $g(n, k) = \Omega(n^{1/k})$. While the conjecture is very commonly used in the computer science community as evidence for spanners' optimality, the combinatorics community is quite skeptical about it [9, 13, 21, 44]; in particular, a bipartite version of the conjecture was refuted [13, 21]. Consequently, the fact that our Theorem 1.6 gives a near-optimal lightness bound that does not rely on Erdős' girth conjecture is a significant advantage. We are not aware of any prior work showing the existence of a near optimal spanner without the Erdős' girth conjecture.

Theorem 1.7. *For any n -point set $P \in \mathbb{R}^d$ and any $d \geq 3$, $d = O(1)$, there is a Steiner $(1 + \epsilon)$ -spanner for P with lightness $\tilde{O}(\epsilon^{-(d+1)/2})$ that is constructable in polynomial time.*

We also obtain improved lightness bounds for light spanners in high dimensional Euclidean spaces. The literature on spanners in high-dimensional Euclidean spaces is surprisingly sparse. Har-Peled, Indyk and Sidiropoulos [43] showed that for any set of n -point Euclidean space (in any dimension) and any parameter $t \geq 2$, there is an $O(t)$ -spanner with sparsity $O(n^{1/t^2} \cdot (\log n \log t))$. Filtser and Neiman [33] gave an analogous but weaker result for lightness, achieving a lightness bound of $O(t^3 n^{\frac{1}{t^2}} \log n)$. They also generalized their results to any ℓ_p metric, for $p \in (1, 2]$, achieving a lightness bound of $O(\frac{t^{1+p}}{\log^2 t} n^{\frac{\log^2 t}{t^p}} \log n)$. Our results improve all of these results.

Theorem 1.8. • *For any n -point set P in a Euclidean space and any given $t \geq 2$, there is an $O(t)$ -spanner for P with lightness $O(tn^{\frac{1}{t^2}} \log n)$ that is constructible in polynomial time.*

- *For any n -point ℓ_p normed space (X, d_X) with $p \in (1, 2]$ and any $t \geq 2$, there is an $O(t)$ -spanner for (X, d_X) with lightness $O(tn^{\frac{\log^2 t}{t^p}} \log n)$.*

1.3 Our Unified Framework: Technical and Conceptual Highlights

In this section, we give a high-level overview of our framework for constructing light spanners with stretch $t(1 + \epsilon)$, for some parameter t that depends on the examined graph class; e.g., for Euclidean spaces $t = 1 + \epsilon$, while for general graphs $t = 2k - 1$. We shall construct spanners with stretch $t(1 + O(\epsilon))$ and assume w.l.o.g. that ϵ is sufficiently smaller than 1; a stretch of $t(1 + \epsilon)$, for any $0 \leq \epsilon \leq 1$, can be achieved by scaling.

Let L be a positive parameter, and let $H_{<L}$ be a $t(1 + \gamma\epsilon)$ -spanner for all edges in $G = (V, E, w)$ of weight $< L$, for some constant $\gamma \geq 1$. That is, $V(H_{<L}) = V$ and for any edge $(u, v) \in E$ with $w(u, v) < L$:

$$d_{H_{<L}}(u, v) \leq t(1 + \gamma\epsilon)w(u, v). \quad (1)$$

Note that by the triangle inequality, $H_{<L}$ is also a $t(1 + \gamma\epsilon)$ -spanner for every pair of vertices of distance $< L$. Our framework relies on the notion of a *cluster graph*, defined as follows.

Definition 1.9 ($(L, \epsilon, \beta, \Upsilon)$ -Cluster Graph). *An edge-weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \omega)$ is called an (L, ϵ, β) -cluster graph with respect to spanner $H_{<L}$, for positive parameters $L, \epsilon, \beta, \Upsilon > 1$, if it satisfies the following conditions:*

1. *Each node $\varphi_C \in \mathcal{V}$ corresponds to a subset of vertices $C \in V$, called a cluster, in the original graph G . For any pair $\varphi_{C_1}, \varphi_{C_2}$ of distinct nodes in $\mathcal{V}$, we have $C_1 \cap C_2 = \emptyset$.*
2. *Each edge $(\varphi_{C_1}, \varphi_{C_2}) \in \mathcal{E}$ corresponds to an edge $(u, v) \in E$, such that $u \in C_1$ and $v \in C_2$. Furthermore, $\omega(\varphi_{C_1}, \varphi_{C_2}) = w(u, v)$.*
3. *$L \leq \omega(\varphi_{C_1}, \varphi_{C_2}) < \Upsilon L$, for every edge $(\varphi_{C_1}, \varphi_{C_2}) \in \mathcal{E}$.*

4. $\text{Dm}(H_{<L}[C]) \leq \beta\epsilon L$, for any cluster C corresponding to a node $\varphi_C \in \mathcal{V}$.

Here $\text{Dm}(X)$ denotes the diameter of a graph X , i.e., the maximum pairwise distance in X .

Condition (1) asserts that clusters corresponding to nodes of $\mathcal{G}$ are vertex-disjoint. Condition (3) asserts that edges in $\mathcal{E}$ have the same weight up to a factor of Υ , which is always at most 2 in our construction.. Furthermore, Condition (4) asserts that they induce subgraphs of low diameter in $H_{<L}$. In particular, if β is constant, then the diameter of clusters is roughly ϵ times the weight of edges in the cluster graph. That is, the diameter of the clusters is much smaller than the weight of the edges when ϵ is sufficiently small.

In our framework, we use the cluster graph to compute a subset of edges in G of weights in $[L, \Upsilon L]$ to add to the spanner $H_{<L}$, so as to obtain a spanner, denoted by $H_{<\Upsilon L}$, for all edges in G of weight less than ΥL . As a result, we extend the set of edges whose endpoints' distances are preserved (to within the required stretch bound) by the spanner. By repeating the same construction for edges of higher and higher weights, we eventually obtain a spanner that preserves all pairwise distances in G .

There are two values that Υ can take, depending on whether we wish to optimize the running time or the fine-grained dependence on ϵ and other parameters such as the size of the excluded minor. In the former case we set $\Upsilon = 1 + \epsilon$, whereas in the latter we set $\Upsilon = 2$.

Note that a single edge of $\mathcal{G}$ may correspond to multiple edges of G ; to facilitate the transformation of edges of $\mathcal{G}$ to edges of G , we assume access to a function $\text{source}(\cdot)$ that supports the following operations in $O(1)$ time: (a) given a node φ_C , $\text{source}(\varphi_C)$ returns a vertex $r(C)$ in cluster C , called the *representative* of C , (b) given an edge $(\varphi_{C_1}, \varphi_{C_2})$ in $\mathcal{E}$, $\text{source}(\varphi_{C_1}, \varphi_{C_2})$ returns the corresponding edge (u, v) of $(\varphi_{C_1}, \varphi_{C_2})$, which we refer to as the *source edge* of (u, v) , where $u \in C_1$ and $v \in C_2$; we note that u (resp., v) need not be $r(C_1)$ (resp., $r(C_2)$). Constructing the function $\text{source}(\cdot)$ efficiently is straightforward; the details are in Section 7.

For *optimizing the construction time*, our framework assumes the existence of the following algorithm, hereafter the *sparse spanner algorithm (SSA)*, which computes a subset of edges in $\mathcal{G}$, whose source edges are added to $H_{<L}$. Recall that the parameter Υ is set as $\Upsilon = 1 + \epsilon$ in this case.

SSA: Given an $(L, \epsilon, \beta, \Upsilon = 1 + \epsilon)$ -cluster graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \omega)$ and function $\text{source}(\cdot)$ as defined above, the SSA outputs a subset of edges $\mathcal{E}^{\text{pruned}} \subseteq \mathcal{E}$ such that:

1. **(Sparsity)** $|\mathcal{E}^{\text{pruned}}| \leq \chi|\mathcal{V}|$ for some $\chi > 0$.
2. **(Stretch)** For each edge $(\varphi_{C_u}, \varphi_{C_v}) \in \mathcal{E}$, $d_{H_{<(1+\epsilon)L}}(u, v) \leq t(1 + s_{\text{SSA}}(\beta)\epsilon)w(u, v)$ where $(u, v) = \text{source}(\varphi_{C_u}, \varphi_{C_v})$ and $s_{\text{SSA}}(\beta)$ is some constant that depends on β only, and $H_{<(1+\epsilon)L}$ is the graph obtained by adding the source edges of $\mathcal{E}^{\text{pruned}}$ to $H_{<L}$.

Let $\text{Time}_{\text{SSA}} = O((m' + n')\tau(m', n'))$ be the running time of the SSA, where τ is a monotone non-decreasing function, $n' = |\mathcal{V}|$ and $m' = |\mathcal{E}|$.

Intuitively, the SSA can be viewed as an algorithm that constructs a *sparse spanner for an unweighted graph*, as edges of $\mathcal{G}$ have the same weights up to a factor of $(1 + \epsilon)$ and the only requirement from the edge set $\mathcal{E}^{\text{pruned}}$ returned by the SSA, besides achieving small stretch, is that it would be of small size. While the interface to the SSA remains the same across all graphs, its exact implementation may change from one graph class to another; informally, for each graph class, the SSA is akin to the state-of-the-art unweighted spanner construction for that class, and this part of the framework is pretty simple.

For *optimizing the fine-grained dependence on ϵ and other parameters (such as minor size) in the lightness bound*, our framework assumes the existence of the following algorithm, called *sparse spanner*

oracle (SSO), which computes a subset of edges in G to add to $H_{<L}$. Recall that the parameter Υ is set as $\Upsilon = 2$ in this case.

SSO: Given an $(L, \epsilon, \beta, \Upsilon = 2)$ -cluster graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \omega)$, the SSO outputs a subset of edges $F \subseteq E$ in polynomial time such that:

1. **(Sparsity)** $w(F) \leq \chi|\mathcal{V}|L$ for some $\chi > 0$.
2. **(Stretch)** For each edge $(\varphi_{C_u}, \varphi_{C_v}) \in \mathcal{E}$, $d_{H_{<2L}}(u, v) \leq t(1 + s_{\text{SSO}}(\beta)\epsilon)w(u, v)$ where (u, v) is the corresponding edge of $(\varphi_{C_u}, \varphi_{C_v})$ and $s_{\text{SSO}}(\beta)$ is some constant that depends on β only, and $H_{<2L}$ is the graph obtained by adding F to $H_{<L}$.

We can interpret the SSO as a construction of a *sparse spanner* in the following way: If F contains only edges of G corresponding to a subset of $\mathcal{E}$, say $\mathcal{E}^{\text{pruned}} \subseteq \mathcal{E}$, then, $w(e) \geq L$ for every $e \in F$; in this case $|F| \leq \chi|\mathcal{V}|$. The edges in the set F produced by the SSO may not correspond to edges in $\mathcal{E}$ of $\mathcal{G}$. This allows for more flexibility in choosing the set of edges to add to $H_{<L}$, and is the key to obtaining a fine-grained optimal dependencies on ϵ and the other parameters, such as the Euclidean dimension or the minor size. Importantly, for all classes of graphs considered in this paper, the implementation of SSO is very simple, as we show in Section 5.

The highly nontrivial part of the framework is given by the following theorem, which provides a *black-box transformation* (i) from an SSA to an efficient (in terms of running time) *meta-algorithm* for constructing light spanners and (ii) from an SSO to an efficient (in terms of fine-grained dependencies) *meta-algorithm* for constructing light spanners. We note that this transformation remains the same across all graphs.

Theorem 1.10. *Let $L, \epsilon, t, \gamma, \beta$ be non-negative parameters where $\gamma, \beta \geq 1$ only take on constant values, and $\epsilon \ll 1$. Let $\mathcal{F}$ be an arbitrary graph class. If, for any graph G in $\mathcal{F}$:*

- (1) *the SSA can take any $(L, \epsilon, \beta, \Upsilon = 1 + \epsilon)$ -cluster graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \omega)$ corresponding to G as input and return as output a subset of edges $\mathcal{E}^{\text{pruned}} \subseteq \mathcal{E}$ satisfying the aforementioned two properties of (Sparsity) and (Stretch), then for any graph in $\mathcal{F}$ we can construct a spanner with stretch $t(1 + (s_{\text{SSA}}(O(1)) + O(1))\epsilon)$, lightness $O((\chi\epsilon^{-3} + \epsilon^{-4})\log(1/\epsilon))$, and in time $O(m\epsilon^{-1}(\alpha(m, n) + \tau(m, n) + \epsilon^{-1})\log(1/\epsilon))$.*
- (2) *the SSO can take any $(L, \epsilon, \beta, \Upsilon = 2)$ -cluster graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \omega)$ corresponding to G as input and return as output a subset of edges F of G satisfying the aforementioned two properties of (Sparsity) and (Stretch), then for any graph in $\mathcal{F}$ we can construct a spanner with stretch $t(1 + (2s_{\text{SSO}}(O(1)) + O(1))\epsilon)$, lightness $\tilde{O}_\epsilon((\chi\epsilon^{-1} + \epsilon^{-2}))$ when $t = 1 + \epsilon$, and lightness $\tilde{O}_\epsilon((\chi\epsilon^{-1}))$ when $t \geq 2$.*

See Figure 1 for an illustration of how Theorem 1.10 is used to derive various results in our paper. We remark the following regarding Theorem 1.10.

Remark 1.11. *If the SSA can be implemented in the ACT model with the stated running time, then the construction of light spanners provided by Theorem 1.10 can also be implemented in the ACT model in the stated running time.*

In the implementations of SSA for Euclidean spaces and UDGs, we rely on the condition that $H_{<L}$ preserves distances smaller than L within a factor of $t(1 + \gamma\epsilon)$. However, we do not need this condition to hold for general graphs and minor-free graphs; for them all we need is Condition 4 in Definition 1.9.

For fast constructions, the transformation provided by Theorem 1.10 — from sparsity in almost unweighted graphs (as captured by the SSA) to lightness — has a constant loss on lightness (for constant

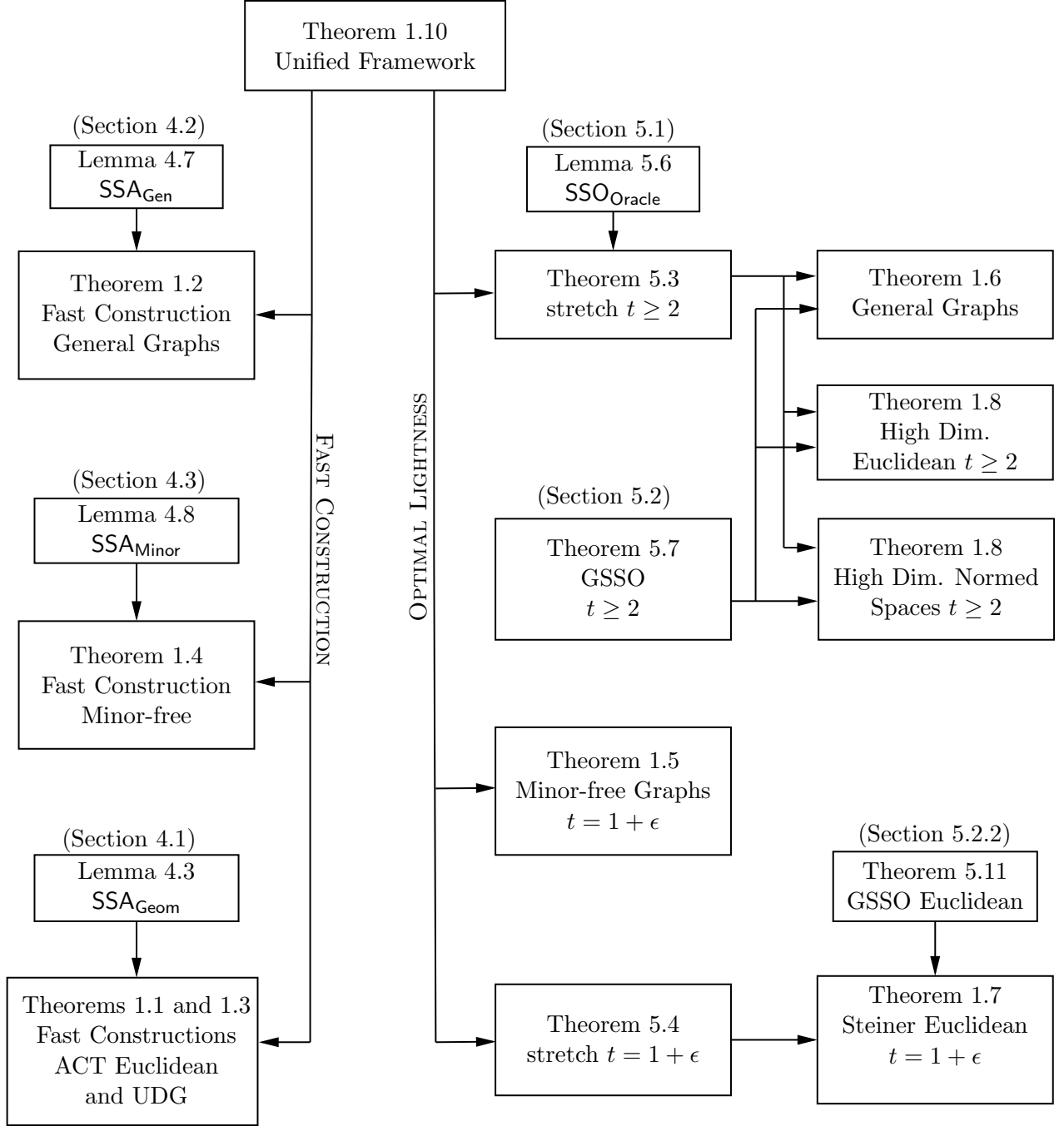


Figure 1: Applications of our framework in obtaining fast constructions of light spanners (on the left) and spanners with truly optimal lightness (on the right). The notion of *general sparse spanner oracle* (GSSO) is another abstraction that we will formally introduce in Section 5.1.

ϵ) and a small running time overhead. In Section 4, we provide simple implementations of the SSA for several classes of graphs in time $O(m + n)$, for a constant ϵ ; Theorem 1.10 thus directly yields a running time of $O((m + n)\alpha(m, n))$. For minor-free graphs, with an additional effort, we remove the factor $\alpha(m, n)$ from the runtime. For Euclidean spaces and UDGs, we apply the transformation not on the input space but rather on a sparse spanner, with $O(n)$ edges, hence the runtime $O((m + n)\alpha(m, n))$

of the transformation is not the bottleneck, as it is dominated by the time $\Theta(n \log n)$ needed for building Euclidean spanners.

For obtaining fine-grained lightness bounds, the transformation from sparsity to lightness in Theorem 1.10 only loses a factor of $1/\epsilon$ for stretch $t \geq 2$, and, in addition, another additive term of $+\frac{1}{\epsilon^2}$ is lost for stretch $t = 1 + \epsilon$. Later, we complement this upper bound by a lower bound (Section 3) showing that for $t = 1 + \epsilon$, the additive term of $+\frac{1}{\epsilon^2}$ is unavoidable in the following sense: There is a graph class — the class of bounded treewidth graphs — where we can implement an SSO with $\chi = O(1)$ for stretch $(1 + \epsilon)$, and hence the lightness of the transformed spanner is $O(1/\epsilon^2)$ due to the additive term of $+\frac{1}{\epsilon^2}$, but any light $(1 + \epsilon)$ -spanner for this class of graphs must have lightness $\Omega(1/\epsilon^2)$. (We modify this construction to obtain the lower bound for minor-free graphs in Theorem 1.5.)

Despite the clean conceptual message behind Theorem 1.10 — in providing a transformation from sparse to light spanners — its proof is technical and highly intricate. This should not be surprising as our goal is to have a single framework that can be applied to basically any graph class. The applicability of our framework goes far beyond the specific graph classes considered in the current paper, which merely aim at capturing several very different conceptual and technical hurdles, e.g., complete vs. non-complete graphs, geometric vs. non-geometric graphs, stretch $1 + \epsilon$ vs. large stretch, etc. The heart of our framework is captured by Theorem 1.10, whose proof appears in Part II. The starting point of our proof of Theorem 1.10 is a basic hierarchical partition, which dates back to the early 90s [4, 15], and was used by most if not all of the works on light spanners (see, e.g., [29, 30, 17, 10, 11, 50]). The current paper takes this hierarchical partition approach to the next level, by proposing a unified framework that reduces the problem of *efficiently* constructing a *light* spanner to the conjunction of two problems: (1) efficiently constructing a hierarchy of clusters with several *carefully chosen properties*, and (2) efficiently constructing a *sparse spanner*; these two problems are intimately related, in the sense that the “carefully chosen properties” of the clusters are set so that we can efficiently apply the sparse spanner construction.

To minimize the dependency on ϵ in the transformation in Theorem 1.10, we construct clusters in such a way that (1) a cluster at a higher level should contain as many clusters as possible, called subclusters, at lower levels, and (2) the augmented diameter of the cluster must be within a restricted bound. Condition (1) implies that each cluster has a large potential change, which is used to “pay” for spanner edges that the algorithm adds to the spanner, while condition (2) implies that the constructed spanner has the desired stretch. The two conditions are in conflict with each other, since the more subclusters we have in a single cluster, the larger the diameter of the cluster gets. Achieving the right balance between these two conflicting conditions is the main technical contribution of this paper.

Another significant technical contribution of our paper in this context is in introducing the notion of *augmented diameter* of a cluster. The definition of augmented diameter appears in Section 2, but at a high level, the idea is to consider weights on both nodes and edges in a cluster, where the node weights are determined by the *potential values* of clusters computed (via simple recursion) in previous levels of the hierarchy. The main advantage of augmented diameter over the standard notion of diameter is that it can be computed efficiently, while the computation of diameter is much more costly. Informally, the augmented diameter can be computed efficiently since (i) we can upper bound the hop-diameter of clusters, and (ii) the clusters at each level are computed on top of some underlying tree; roughly speaking, that means that all the distance computations are carried out on top of subtrees of bounded hop-diameter (or depth), hence the source of efficiency.

We next argue that our approach is inherently different than previous ones. First, the very fact that our approach is unified makes it inherently different than previous approaches, which, as mentioned, are ad hoc per graph class. Second, our approach is not just a unified framework for reproving known results — we employ it to break through the state-of-the-art. To this end, we highlight one concrete result — on Euclidean spanners in the ACT model — which breaks a longstanding barrier in the area of geometric spanners, by using an inherently *non-geometric* approach. All the previous algorithms for

light Euclidean spanners were achieved via the greedy and approximate-greedy spanner constructions. The greedy algorithm is non-geometric but slow, whereas the approximate-greedy algorithm is geometric and can be implemented much more efficiently. The analysis of the lightness in both algorithms is done via the so-called *leapfrog property* [23, 26, 24, 25, 40, 53], which is a geometric property. The fast spanner construction of GLN [40] implements the approximate-greedy algorithm by constructing a hierarchy of clusters with $O(\frac{\log n}{\log \log n})$ levels and, for each level, Dijkstra's algorithm is used for the construction of clusters for the next level. The GLN construction incurs an additional $O(n \log n)$ factor for each level to run Dijkstra's algorithm in the ACT model, which ultimately leads to a runtime of $O(n \frac{\log^2 n}{\log \log n})$. Our approach is inherently different, and in particular we do not need to run Dijkstra's algorithm or any other single-source shortest (or approximately shortest) path algorithm. The key to our efficiency is in a careful usage of the new notion of augmented diameter, as well as its interplay with the potential function argument and the hierarchical partition that we use. We stress again that our approach is non-geometric, and the only potential usage of geometry is in the sparse spanner construction that we apply. (Indeed, the sparse spanner construction that we chose to apply is geometric, but this is not a must.)

2 Preliminaries

Let G be an arbitrary weighted graph. We denote by $V(G)$ and $E(G)$ the vertex set and edge set of G , respectively. We denote by $w : E(G) \rightarrow \mathbb{R}^+$ the weight function on the edge set. Sometimes we write $G = (V, E)$ to clearly explicate the vertex set and edge set of G , and $G = (V, E, w)$ to further indicate the weight function w associated with G . We use $\text{MST}(G)$ to denote a minimum spanning tree of G ; when the graph is clear from context, we simply use MST as a shorthand for $\text{MST}(G)$.

For a subgraph H of G , we use $w(H) \stackrel{\text{def.}}{=} \sum_{e \in E(H)} w(e)$ to denote the total edge weight of H . The *distance* between two vertices p, q in G , denoted by $d_G(p, q)$, is the minimum weight of a path between them in G . The diameter of G , denoted by $\text{Dm}(G)$, is the maximum pairwise distance in G . A *diameter path* of G is a shortest (i.e., of minimum weight) path in G realizing the diameter of G , that is, it is a shortest path between some pair u, v of vertices in G such that $\text{Dm}(G) = d_G(u, v)$.

Sometimes we shall consider graphs with weights on both *edges and vertices*. We define the *augmented weight* of a path to be the total weight of all edges and vertices along the path. The *augmented distance* between two vertices in G is defined as the minimum augmented weight of a path between them in G . Likewise, the *augmented diameter* of G , denoted by $\text{Adm}(G)$, is the maximum pairwise augmented distance in G ; since we will focus on non-negative weights, the augmented distance and augmented diameter are no smaller than the (ordinary notions of) distance and diameter. An *augmented diameter path* of G is a path of minimum augmented weight realizing the augmented diameter of G .

Given a subset of vertices $X \subseteq V(G)$, we denote by $G[X]$ the subgraph of G *induced by* X : $G[X]$ has $V(G[X]) = X$ and $E(G[X]) = \{(u, v) \in E(G) \mid u, v \in X\}$. Let $F \subseteq E(G)$ be a subset of edges of G ; we denote by $G[F]$ the subgraph of G with $V(G[F]) = V(G)$ and $E(G[F]) = F$.

Let S be a *spanning* subgraph of G ; weights of edges in S are inherited from G . The *stretch* of S is given by $\max_{x, y \in V(G)} \frac{d_S(x, y)}{d_G(x, y)}$, and it is realized by some edge e of G . We say that S is a *t-spanner* of G if the stretch of S is at most t . There is a simple greedy algorithm, called **path greedy** (or shortly **greedy**), to find a *t-spanner* of a graph G : Examine the edges $e = (x, y)$ in G in nondecreasing order of weights, and add to the spanner edge (x, y) iff the distance between x and y in the *current* spanner is larger than $t \cdot w(x, y)$.

We say that a subgraph H of G is a *t-spanner* for a *subset of edges* $X \subseteq E$ if $\max_{(u, v) \in X} \frac{d_H(u, v)}{d_G(u, v)} \leq t$.

In the context of minor-free graphs, we denote by G/e the graph obtained from G by contracting e , where e is an edge in G . If G has weights on edges, then every edge in G/e inherits its weight from G .

In addition to general and minor-free graphs, this paper studies *geometric graphs*. Let P be a set of n points in $\mathbb{R}^d$. We denote by $\|p, q\|$ the Euclidean distance between two points $p, q \in \mathbb{R}^d$. A *geometric graph* G for P is a graph where the vertex set corresponds to the point set, i.e., $V(G) = P$, and the edge weights are the Euclidean distances, i.e., $w(u, v) = \|u, v\|$ for every edge (u, v) in G . Note that G need not be a complete graph. If G is a complete graph, i.e., $G = (P, \binom{P}{2}, \|\cdot\|)$, then G is equivalent to the *Euclidean space* induced by the point set P . For geometric graphs, we use the term *vertex* and *point* interchangeably.

We use $[n]$ and $[0, n]$ to denote the sets $\{1, 2, \dots, n\}$ and $\{0, 1, \dots, n\}$, respectively.

3 Lightness Lower Bounds

In this section, we provide lower bounds on light $(1+\epsilon)$ spanners to prove the lower bound in Theorem 1.5. Interestingly, our lower bound construction draws a connection between geometry and graph spanners: we construct a fractal-like geometric graph of weight $\Omega(\frac{\text{MST}}{\epsilon^2})$ such that it has treewidth at most 4 and any $(1+\epsilon)$ -spanner of the graph must take all the edges.

Theorem 3.1. *For any $n = \Omega(\epsilon^{\Theta(1/\epsilon)})$ and $\epsilon < 1$, there is an n -vertex graph G of treewidth at most 4 such that any light $(1+\epsilon)$ -spanner of G must have lightness $\Omega(\frac{1}{\epsilon^2})$.*

Before proving Theorem 3.1, we show its implications to the lower bound in Theorem 1.5.

Proof: [Proof of the lower bound in Theorem 1.5] First, construct a complete graph H_1 on $r-1$ vertices for which any $(1+\epsilon)$ -spanner has lightness $\Omega(\frac{r}{\epsilon})$ as follows: Let $X_1 \subseteq V(H_1)$ be a subset of $r/2$ vertices and $X_2 = V(H_1) \setminus X_1$. We assign weight 2ϵ to every edge with both endpoints in X_1 or X_2 , and weight 1 to every edge between X_1 and X_2 . Clearly $\text{MST}(H_1) = 1 + (r-2)2\epsilon$. We claim that any $(1+\epsilon)$ -spanner S_1 of H_1 must take every edge between X_1 and X_2 ; otherwise, if $e = (u, v)$ is not taken where $u \in X_1, v \in X_2$, then $d_{S_1}(u, v) \geq d_{H_1 \setminus e}(u, v) = 1 + 2\epsilon > (1+\epsilon)d_G(u, v)$. Thus, $w(S_1) \geq |X_1||X_2| = \Omega(r^2)$. This implies $w(S_1) = \Omega(\frac{r}{\epsilon})w(\text{MST}(H_1))$.

Let H_2 be an $(n-r+1)$ vertex graph of treewidth 4 guaranteed by Theorem 3.1; H_2 excludes K_r as a minor for any $r \geq 6$. We scale edge weights of H_1 appropriately so that $w(\text{MST}(H_2)) = w(\text{MST}(H_1))$. Connect H_1 and H_2 by a single edge of weight $2w(\text{MST}(H_1))$ to form a graph G . Then G excludes K_r as minor (for $r \geq 5$) since H_1 and H_2 both exclude K_r as a minor. Furthermore, any $(1+\epsilon)$ -spanner of G must have lightness at least $\Omega(\frac{r}{\epsilon} + \frac{1}{\epsilon^2})$ as $w(\text{MST}(G)) = 4w(\text{MST}(H_1))$. $\square$

We now focus on proving Theorem 3.1. The core gadget in our construction is depicted in Figure 2. Let C_r be a circle on the plane centered at a point o of radius r . We use $\widehat{ab}$ to denote an arc of C_r with two endpoints a and b . We say $\widehat{ab}$ has angle θ if $\angle aob = \theta$. We use $|\widehat{ab}|$ to denote the (arc) length of $\widehat{ab}$, and $\|a, b\|$ to denote the Euclidean length between a and b .

By elementary geometry and Taylor's expansion, one can verify that if $\widehat{ab}$ has angle θ , then:

$$\begin{aligned} |\widehat{ab}| &= \theta r \\ \|a, b\| &= 2r \sin(\theta/2) = r\theta(1 - \theta^2/24 + o(\theta^3)) \\ \|a, b\| &= \frac{2 \sin(\theta/2)}{\theta} |\widehat{ab}| = (1 - \theta^2/24 + o(\theta^3)) |\widehat{ab}| \end{aligned} \tag{2}$$

Core Gadget. The construction starts with an arc ab of angle $\sqrt{\epsilon}$ of a circle C_r . W.l.o.g., we assume that $\frac{1}{\epsilon}$ is an odd integer. Let $k = \frac{1}{2}(\frac{1}{\epsilon} + 1)$. Let $\{a \equiv x_1, x_2, \dots, x_{2k} \equiv b\}$ be the set of points, called *break points*, on the arc ab such that $\angle x_i o x_{i+1} = \epsilon^{3/2}$ for any $1 \leq i \leq 2k-1$.

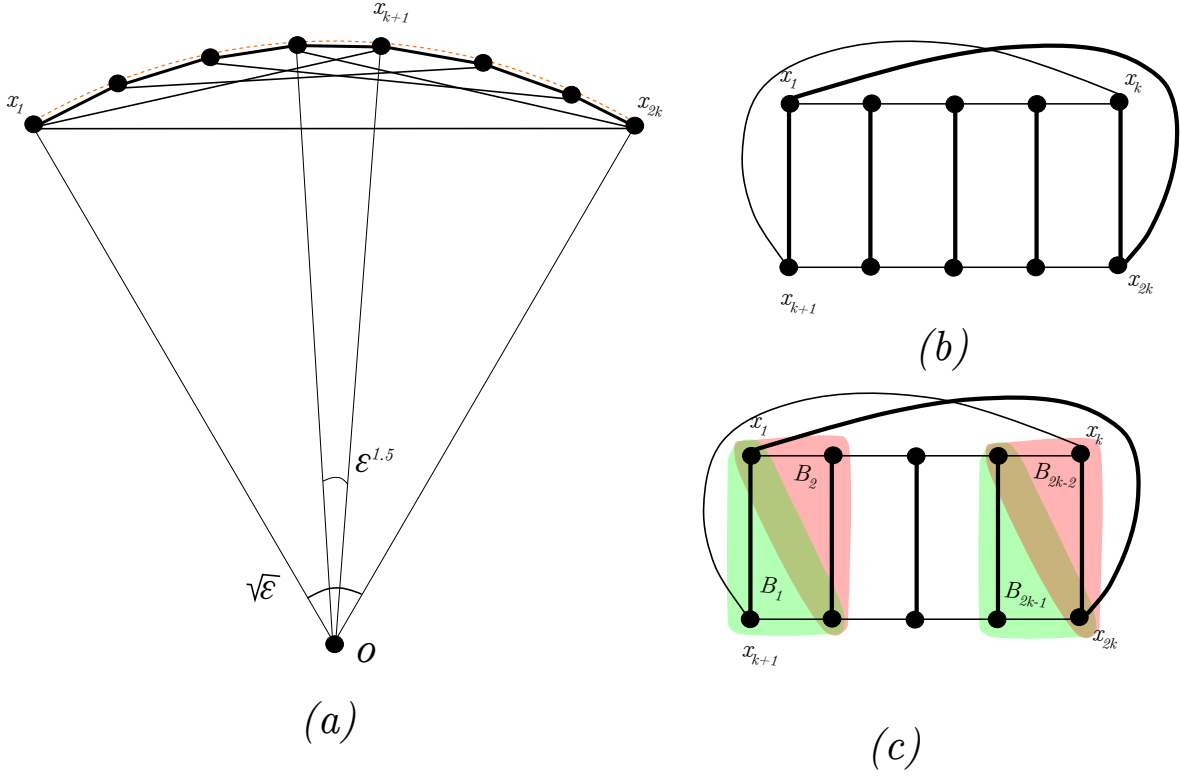


Figure 2: (a) The core gadget. (b) A different view of the core gadget. (c) A tree decomposition of the core gadget.

Let H_r be a graph with vertex set $V(H_r) = \{x_1, \dots, x_{2k}\}$. We call x_1 and x_{2k} two *terminals* of H_r . For each $i \in [2k - 1]$, we add an edge $x_i x_{i+1}$ of weight $w(x_i x_{i+1}) = \|x_i, x_{i+1}\|$ to $E(H_r)$. We refer to edges between $x_i x_{i+1}$ for $i \in [2k - 1]$ as *short edges*. For each $i \in [k]$, we add an edge $x_i x_{i+k}$ of weight $\|x_i, x_{i+k}\|$. We refer to these edges as *long edges*. Finally, we add edge $\|x_1, x_{2k}\|$ of $E(H_r)$, that we refer to as the *terminal edge* of H_r . We call H_r a *core gadget* of scale r . See Figure 2(a) for a geometric visualization of H_r and Figure 2(b) for an alternative view of H_r .

We observe that:

Observation 3.2. H_r has the following properties:

1. For any edge $e \in E(H_r)$, we have:

$$w(e) = \begin{cases} 2r \sin(\epsilon^{3/2}/2) & \text{if } e \text{ is a short edge} \\ 2r \sin(k\epsilon^{3/2}/2) & \text{if } e \text{ is a long edge} \\ 2r \sin(\sqrt{\epsilon}/2) & \text{if } e \text{ is the terminal edge} \end{cases} \quad (3)$$

2. $w(\text{MST}(H_r)) \leq r\sqrt{\epsilon}$.

3. $w(H_r) \geq \frac{r}{6\sqrt{\epsilon}}$ when $\epsilon \ll 1$.

Proof: We only verify (3); other properties can be seen by direct calculation. By Taylor's expansion, each long edge of H_r has weight $w(e) = 2 \sin(\frac{1}{4}(\sqrt{\epsilon} + \epsilon^{3/2})) = \frac{r}{2}(\sqrt{\epsilon} + o(\epsilon)) \geq r\sqrt{\epsilon}/3$ when $\epsilon \ll 1$. Since H_r has k long edges, $w(H_r) \geq kr\sqrt{\epsilon}/3 \geq \frac{r}{6\sqrt{\epsilon}}$. $\square$

Next, we claim that H_r has small treewidth.

Claim 3.3. H_r has treewidth at most 4.

Proof: We construct a tree decomposition of width 4 of H_r . In fact, we can construct a path decomposition of width 4 for H_r . Let $B_1, \dots, B_{2k-2}$ be set of vertices where $B_{2i-1} = \{x_{2i-1}, x_{2i+k-1}, x_{2i+k}\}$ and $B_{2i} = \{x_{2i-1}, x_{2i+k}, x_{2i}\}$ for each $i \in [k-1]$ (see Figure 2(c)). We then add x_1 and x_k to every B_i . Then, $\mathcal{P} = \{B_1, \dots, B_{2k-2}\}$ is a path decomposition of H_r of width 4. $\square$

Remark: It can be seen that H_r has K_4 as a minor, thus has treewidth at least 3. Showing that H_r has treewidth at least 4 needs more work.

Lemma 3.4. *There is a constant c such that any $(1 + \epsilon/c)$ -spanner of H_r must have weight at least*

$$\frac{w(\text{MST}(H_r))}{6\epsilon}.$$

Proof: Let e be a long edge of H_r and $G_e = H_r \setminus \{e\}$.

We claim that the shortest path between e 's endpoints in

G_e must have length at least $(1 + \epsilon/c)w(e)$ for some constant c . That implies any $(1 + \epsilon/c)$ -spanner of H_r must include all long edges. The lemma then follows from Observation 3.2 since H_r has at least $1/2\epsilon$ long edges, and each has length at least $w(\text{MST}(H_r))/3$ for $\epsilon \ll 1$.

Suppose that $e = x_i x_{i+k}$. Let P_e is a shortest path between x_i and x_{i+k} in G_e . Suppose that $w(P_e) \leq (1 + \epsilon/c)w(e)$. Since the terminal edge has length at least $3/2w(e)$, P_e cannot contain the terminal edge. For the same reason, P_e cannot contain two long edges. It remains to consider two cases:

1. P_e contains exactly one long edge. Then, it must be that $P_e = \{x_i, x_{i+1}, x_{i+k+1}, x_{i+k}\}^2$ (Figure 3(a)) or $P_e = \{x_i, x_{i-1}, x_{i+k-1}, x_{i+k}\}$ (Figure 3(b)). In both case, $w(P_e) = w(e) + 4r \sin(\epsilon^{3/2}/2) \geq w(e)(1 + 2 \frac{\sin(\epsilon^{3/2}/2)}{\sin(k\epsilon^{3/2}/2)}) \geq (1 + 2\epsilon)w(e)$.
2. P_e contains no long edge. Then, $P_e = \{x_i, x_{i+1}, \dots, x_{i+k}\}$. Thus we have:

$$\frac{w(P_e)}{w(e)} = \frac{2kr \sin(\epsilon^{3/2}/2)}{2r \sin(k\epsilon^{3/2}/2)} = 1 + \epsilon/96 + o(\epsilon) \geq 1 + \epsilon/100$$

Thus, by choosing $c = 100$, we derive a contradiction. $\square$

Proof of Theorem 3.1. The construction is recursive. Let H_1 the core gadget of scale 1. Let s_1 (ℓ_1) be the length of short edges (long edges) of H_1 . Let $x_1^1, \dots, x_k^1$ be break points of H_1 . Let δ be the ratio of the length of a short edge to the length of the terminal edge. That is:

$$\delta = \frac{\|x_1^1, x_2^1\|}{\|x_1^1, x_{2k}^1\|} = \frac{\sin(\epsilon^{3/2}/2)}{\sin(\sqrt{\epsilon}/2)} = \epsilon + o(\epsilon) \quad (4)$$

Let $L = \frac{1}{\epsilon}$. We construct a set of graphs $G_1, \dots, G_L$ recursively; the output graph is G_L . We refer to G_i is the level- i graph.

Level-1 graph $G_1 = H_1$. We refer to breakpoints of H_1 as breakpoints of G .

²indices are mod $2k$.

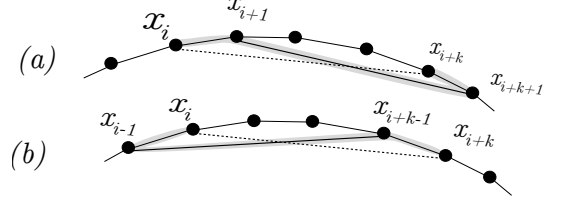


Figure 3: Paths P_e between x_i and x_{i+k} are highlighted.

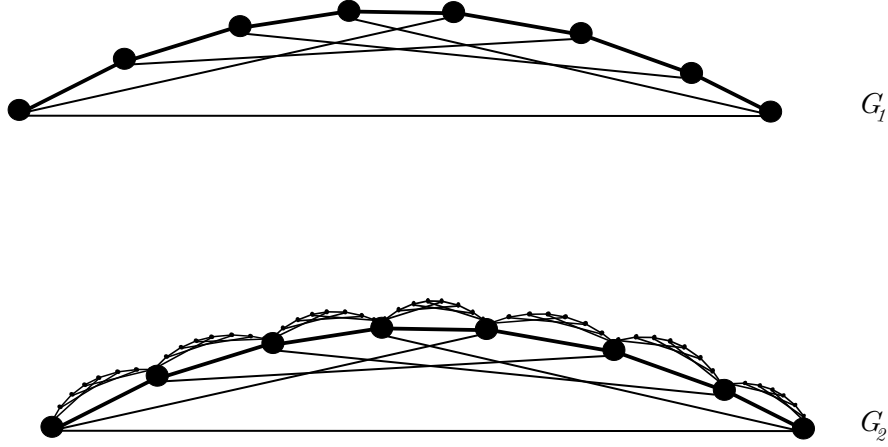


Figure 4: An illustration of the recursive construction of G_L with two levels.

Level-2 graph G_2 obtained from G_1 by: (1) making $2k - 1$ copies of the core gadget H_δ at scale δ (each H_δ is obtained by scaling every edge the core gadget by δ), (2) for each $i \in [2k - 1]$, attach each copy of H_δ to G_1 by identifying the terminal edge of H_δ and the edge between two consecutive breakpoints $x_i^1 x_{i+1}^1$ of G_1 . We then refer to breakpoints of all H_δ as breakpoints of G_2 . (See Figure 4.) Note that by definition of δ , the length of the terminal edge of H_δ is equal to $\|x_i^1, x_{i+1}^1\|$. We say two adjacent breakpoints of G_2 *consecutive* if they belong to the same copy of H_δ in G_2 and are connected by one short edge of H_δ .

Level- j graph G_j obtained from G_{j-1} by: (1) making $(2k - 1)^j$ copies of the core gadget $H_{\delta^{j-1}}$ at scale δ^{j-1} , (2) for every two consecutive breakpoints of G_{j-1} , attach each copy of $H_{\delta^{j-1}}$ to G_{j-1} by identifying the terminal edge of $H_{\delta^{j-1}}$ and the edge between the two consecutive breakpoints. This completes the construction.

We now show some properties of G_L . We first claim that:

Claim 3.5. G_L has treewidth at most 4.

Proof: Let T_1 be the tree decomposition of G_1 of width 5, as guaranteed by Claim 3.3. Note that for every pair of consecutive breakpoints x_i^1, x_{i+1}^1 of G_1 , there is a bag, say X_i , of T_1 contains both x_i^1 and x_{i+1}^1 . Also, there is a bag of T_1 containing both terminals of T_1 .

We extend the tree decomposition T_1 to a tree decomposition T_2 of G_2 as follows. For each gadget H_δ attached to G_1 via consecutive breakpoints x_i^1, x_{i+1}^1 , we add a bag $B = \{x_i^1, x_{i+1}^1\}$, connect B to X_i of T_1 and to the bag containing terminals of the tree decomposition of H_δ . Observe that the resulting tree decomposition T_2 has treewidth at most 4. The same construction can be applied recursively to construct a tree decomposition of G_L of width at most 4. $\square$

Claim 3.6. $w(\text{MST}(G_L)) = O(1)w(\text{MST}(H_1))$.

Proof: Let $r(\epsilon)$ be the ratio between $\text{MST}(H_1)$ and the length of the terminal edge of H_1 . Note that $\text{MST}(H_1)$ is a path of short edges between x_1^1 and x_{2k}^1 . By Observation 3.2, we have:

$$r(\epsilon) \leq \frac{r\sqrt{\epsilon}}{2r \sin(\sqrt{\epsilon}/2)} = 1 + \epsilon/24 + o(\epsilon) \leq 1 + \epsilon \quad (5)$$

when $\epsilon \ll 1$. When we attach copies of H_δ to edges between two consecutive breakpoints of G_1 , by re-routing each edge of $\text{MST}(H_1)$ through the path $\text{MST}(H_\delta)$ between H_δ 's terminals, we obtain a spanning

tree of G_2 of weight at most $r(\epsilon)w(\text{MST}(H_1)) \leq (1 + \epsilon)w(\text{MST}(H_1))$. By induction, we have:

$$w(\text{MST}(G_j)) \leq (1 + \epsilon)w(\text{MST}(G_{j-1})) \leq (1 + \epsilon)^{j-1}w(\text{MST}(H_1))$$

This implies that $w(\text{MST}(G_L)) \leq (1 + \epsilon)^{L-1}w(\text{MST}(H_1)) = O(1)w(\text{MST}(H_1))$. $\square$

Let S be an $(1 + \epsilon/100)$ -spanner of G_L ($c = 100$ in Lemma 3.4). By Lemma 3.4, S includes every long edge of all copies of H_r at every scale r in the construction. Recall that $\|x_1^1, x_{2k}^1\|$ is the terminal edge of G_1 . Let L_j be the set of long edges of all copies of $H_{\delta^{j-1}}$ added at level j . Since $\frac{\text{MST}(G_1)}{\|x_1^1, x_{2k}^1\|} = r(\epsilon)$, we have:

$$w(\text{MST}(G_1)) = \frac{r(\epsilon)}{r(\epsilon) - 1} (w(\text{MST}(G_1)) - \|x_1^1, x_{2k}^1\|) \geq \frac{24}{\epsilon} (w(\text{MST}(G_1)) - \|x_1^1, x_{2k}^1\|) \quad (6)$$

By Lemma 3.4, we have:

$$\begin{aligned} w(L_1) &\geq \frac{1}{6\epsilon} w(\text{MST}(G_1)) \geq \frac{4}{\epsilon^2} (w(\text{MST}(G_1)) - \|x_1^1, x_{2k}^1\|) \\ w(L_2) &\geq \frac{4}{\epsilon^2} (w(\text{MST}(G_2)) - \text{MST}(G_1)) \\ &\dots \\ w(L_j) &\geq \frac{4}{\epsilon^2} (w(\text{MST}(G_j)) - w(\text{MST}(G_{j-1}))) \end{aligned} \quad (7)$$

Thus, we have:

$$w(S) \geq \sum_{j=1}^L w(L_j) \geq \frac{1}{4\epsilon^2} (w(\text{MST}(G_L)) - \|x_1^1, x_{2k}^1\|) = \Omega\left(\frac{1}{\epsilon^2}\right) w(\text{MST}(G_L))$$

By setting $\epsilon \leftarrow \epsilon/100$, we complete the proof of Theorem 3.1. The condition on n follows from the fact that G_L has $|V(G_L)| = O((2k - 1)^L) = O((\frac{1}{\epsilon})^{\frac{1}{\epsilon}})$ vertices. $\square$

Part I

Our Unified Framework: Applications (Section 4 and Section 5)

In this part, we show applications of our unified framework described in Theorem 1.10 in obtaining results in Section 1.

4 Applications of the Unified Framework: Fast Constructions

In this section we implement the SSA for each of the graph classes. By plugging the SSA on top of the general transformation, as provided by Theorem 1.10, we shall prove all theorems stated in Section 1. We assume that $\epsilon \ll 1$, and this is without loss of generality since we can remove this assumption by scaling $\epsilon \leftarrow \epsilon'/c$ for any $\epsilon' \in (0, 1)$ and c is sufficiently large constant. The scaling will incur a constant loss on lightness and runtime, as the dependency on $1/\epsilon$ is polynomial in all constructions below.

4.1 Euclidean Spanners and UDG Spanners

In this section we prove the following theorem.

Theorem 4.1. *Let $G = (V, E, w)$ be a $(1 + \epsilon)$ -spanner either for a set of n points P or for the unit ball graph U of P in $\mathbb{R}^d$. There is an algorithm that can compute a $(1 + O(\epsilon))$ -spanner H of G in the ACT model with lightness $O((\epsilon^{-(d+2)} + \epsilon^{-4}) \log(1/\epsilon))$ in time $O(m\epsilon^{-1}(\alpha(m, n) + \epsilon^{1-d}) \log(1/\epsilon))$.*

We now show that Theorem 4.1 implies Theorem 1.1 and Theorem 1.3. Our construction for UDGs relies on the following result by Fürer and Kasiviswanathan [35].

Lemma 4.2 (Corollary 1 in [36]). *Given a set of n points P in $\mathbb{R}^d$, there is an algorithm that constructs a $(1 + \epsilon)$ -spanner of the unit ball graph for P with $O(n\epsilon^{1-d})$ edges. For $d = 2$, the running time is $O(n(\epsilon^{-2} \log n))$; for $d = 3$, the running time is $\tilde{O}(n^{4/3}\epsilon^{-3})$; and for $d \geq 4$, the running time is $O(n^{2 - \frac{2}{\lceil d/2 \rceil + 1} + \delta} \epsilon^{-d+1} + n\epsilon^{-d})$ for any constant $\delta > 0$.*

Proof: [Proofs of Theorem 1.1 and Theorem 1.3]

It is known that a Euclidean $(1 + \epsilon)$ -spanner for a set of n points P in $\mathbb{R}^d$ with degree $O(\epsilon^{1-d})$ can be constructed in $O(n \log n)$ time in the ACT model (cf. Theorems 10.1.3 and 10.1.10 in [53]). Furthermore, when $m = O(n\epsilon^{1-d})$, we have that:

$$\alpha(m, n) = \alpha(nO(\epsilon^{-d}), n) = O(\alpha(n) + \log(\epsilon^{-d})) = O(\alpha(n) + d \log(1/\epsilon)).$$

Thus, Theorem 1.1 follows from Theorem 4.1.

By Lemma 4.2, we can construct sparse $(1 + \epsilon)$ -spanners for unit ball graphs with $m = O(n\epsilon^{1-d})$ edges in $O(n(\epsilon^{-2} \log n))$ time when $d = 2$, $\tilde{O}(n^{4/3}\epsilon^{-3})$ time when $d = 3$, and $O(n^{2 - \frac{2}{\lceil d/2 \rceil + 1} + \delta} \epsilon^{-d+1} + n\epsilon^{-d})$ time for any constant $\delta > 0$ when $d \geq 4$. Thus, Theorem 1.3 follows from Theorem 4.1. $\square$

By Theorem 1.10, in order to prove Theorem 4.1, it suffices to implement the SSA for Euclidean and UDG spanners. Next, we give a detailed geometric implementation of the SSA, hereafter SSA_{Geom} ; note that the stretch parameter t in the geometric setting is $1 + \epsilon$. The idea is to use a Yao-graph like construction: For each node $\varphi_C \in \mathcal{V}$, we construct a collection of cones of angle ϵ around the representative $r(C) = \text{source}(\varphi_C)$ of the cluster C corresponding to φ_C . Recall that we have access to a **source** function

that returns the representative of each cluster in $O(1)$ time. Then for each cone, we look at all the representatives of the neighbors (in $\mathcal{G}$) of C that fall into that cone, and pick to $\mathcal{E}^{\text{pruned}}$ the edge that connects $r(C)$ to the representative that is closest to it.

SSA_{Geom} (Euclidean and UDG): The input is a (L, ϵ, β) -cluster graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \omega)$ that corresponds to a Euclidean or UDG spanner. The output is $\mathcal{E}^{\text{pruned}}$; initially, $\mathcal{E}^{\text{pruned}} = \emptyset$.

For each node $\varphi_{C_u} \in \mathcal{V}$, do the following:

- Let $\mathcal{N}(\varphi_{C_u})$ be the set of neighbors of φ_{C_u} in $\mathcal{G}$. We construct a collection of $\tau = O(\epsilon^{1-d})$ cones $\text{Cone}(C_u) = \{Q_1, Q_2, \dots, Q_\tau\}$ that partition $\mathbb{R}^d$, each of angle ϵ and with apex at $r(C_u)$, the representative of C_u . It is known (see, e.g. Lemma 5.2.8 in [53]) that we can construct $\text{Cone}(C_u)$ in time $O(\epsilon^{1-d})$ in the ACT model.
 - For each $j \in [\tau]$:
 - Let $R_j = \{r(C') : \varphi_{C'} \in \mathcal{N}(\varphi_{C_u}) \wedge (r(C') \in Q_j)\}$ be the set of representatives that belong to the cone $Q_j \in \text{Cone}(C_u)$. Let $r_j^* = \arg \min_{r \in R_j} \|r(C_u), r\|$ be the representative in R_j that is closest to $r(C_u)$.
 - Let φ_{C_v} be the node of $\mathcal{G}$ whose cluster C_v has r_j^* as the representative. By the definition of R_j , $(\varphi_{C_u}, \varphi_{C_v})$ is an edge in $\mathcal{E}$. Add $(\varphi_{C_u}, \varphi_{C_v})$ to $\mathcal{E}^{\text{pruned}}$.
- /* We add at most one edge to $\mathcal{E}^{\text{pruned}}$ incident on φ_{C_u} for each of the τ cones. */

We next analyze the running time of SSA_{Geom}, and also show that it satisfies the two properties of (Sparsity) and (Stretch) required by the abstract SSA; these properties are described in Section 1.3. Recall that $H_{<(1+\epsilon)L}$ is the graph obtained by adding the source edges of $\mathcal{E}^{\text{pruned}}$ to $H_{<L}$, which is the spanner for all edges in G of weight $< L$. Note that the stretch of $H_{<L}$ is $t(1 + \gamma\epsilon)$ for $t = 1 + \epsilon$, where γ is a constant. Furthermore, as mentioned, we assume w.l.o.g. that ϵ is sufficiently smaller than 1.

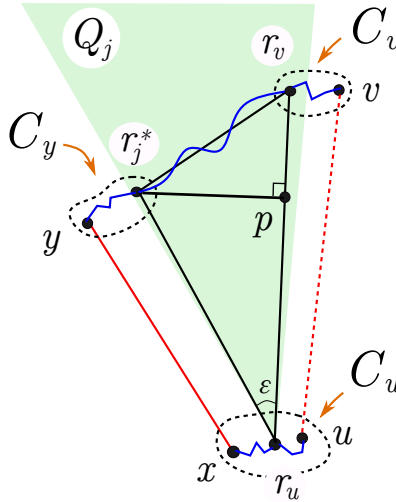


Figure 5: Illustration for the stretch bound proof of Lemma 4.3. Black dashed curves represent three clusters C_u, C_v, C_y . The solid red edge (x, y) corresponds to an edge added to $\mathcal{E}^{\text{pruned}}$, while the dashed red edge (u, v) is not added. The green shaded region represents cone Q_j of angle ϵ with apex at r_u .

Lemma 4.3. SSA_{Geom} can be implemented in $O((|\mathcal{V}| + |\mathcal{E}|)\epsilon^{1-d})$ time in the ACT model. Furthermore, 1. (Sparsity) $|\mathcal{E}^{\text{pruned}}| = O(\epsilon^{1-d}|\mathcal{V}|)$, and 2. (Stretch) For each edge $(\varphi_{C_u}, \varphi_{C_v}) \in \mathcal{E}$, $d_{H_{<(1+\epsilon)L}}(u, v) \leq t(1 + s_{\text{SSA}_{\text{Geom}}}(\beta)\epsilon)w(u, v)$, where $(u, v) = \text{source}(\varphi_{C_u}, \varphi_{C_v})$, $s_{\text{SSA}_{\text{Geom}}}(\beta) = 2(19\beta + 14)$ and $\epsilon \leq \min\{\frac{1}{\gamma}, \frac{1}{8\beta+6}\}$.

Proof: We first analyze the running time. We observe that, since we can construct $\text{Cone}(C_u)$ for a single node φ_{C_u} in $O(\epsilon^{1-d})$ time in the ACT model, the running time to construct all sets of cones $\{\text{Cone}(C_u)\}_{\varphi_{C_u} \in \mathcal{V}}$ is $O(|\mathcal{V}|\epsilon^{1-d})$. Now consider a specific node φ_{C_u} . For each neighbor $\varphi_{C'} \in \mathcal{N}(\varphi_{C_u})$ of φ_{C_u} , finding the cone $Q_j \in \text{Cone}(C_u)$ such that $r(C') \in Q_j$ takes $O(\tau) = O(\epsilon^{1-d})$ time. Thus, $\{R_j\}_{j=1}^\tau$ can be constructed in $O(|\mathcal{N}(\varphi_{C_u})|\epsilon^{1-d})$ time. Finding the set of representatives $\{r_j^*\}_{j=1}^\tau$ takes $O(|\mathcal{N}(\varphi_{C_u})|)$ time by calling function $\text{source}(\cdot)$. Thus, the total running time to implement SSA_{Geom} is:

$$O(|\mathcal{V}|\epsilon^{1-d}) + \sum_{\varphi_{C_u} \in \mathcal{V}} O(|\mathcal{N}(\varphi_{C_u})|\epsilon^{1-d}) = O((|\mathcal{V}| + |\mathcal{E}|)\epsilon^{1-d}),$$

as claimed.

By the construction of the algorithm, for each node $\varphi_C \in \mathcal{V}$, we add at most $\tau = O(\epsilon^{1-d})$ incident edges in $\mathcal{E}$ to $\mathcal{E}^{\text{pruned}}$; this implies Item 1.

It remains to prove Item 2: For each edge $(\varphi_{C_u}, \varphi_{C_v}) \in \mathcal{E}$, the stretch in $H_{<(1+\epsilon)L}$ of the corresponding edge $(u, v) = \text{source}(\varphi_{C_u}, \varphi_{C_v})$ is at most $(1 + s_{\text{SSA}_{\text{Geom}}}(\beta)\epsilon)$ with $s_{\text{SSA}_{\text{Geom}}}(\beta) = 2(19\beta + 14)$. Let $r_u \stackrel{\text{def.}}{=} r(C_u)$ and $r_v \stackrel{\text{def.}}{=} r(C_v)$ be the representatives of C_u and C_v , respectively. Let Q_j be the cone in $\text{Cone}(C_u)$ such that $r_v \in Q_j$ for some $j \in [\tau]$ (we are using the notation in SSA_{Geom}). If $r_v = r_j^*$, then $(u, v) \in H_{<(1+\epsilon)L}$ by the construction in SSA_{Geom} , and so the stretch is 1. Otherwise, let C_y be the level- i cluster that contains the representative r_j^* . By the construction in SSA_{Geom} , there is an edge $(x, y) \in H_{<(1+\epsilon)L}$ where $x \in C_u$ and $y \in C_y$. (See Figure 5.) By property 4 of $\mathcal{G}$ in Definition 1.9, $\max\{\text{Dm}(H_{<(1+\epsilon)L}[C_u]), \text{Dm}(H_{<(1+\epsilon)L}[C_v]), \text{Dm}(H_{<(1+\epsilon)L}[C_y])\} \leq \beta\epsilon L$. Note that edges in $\mathcal{E}$ have weights in $[L, (1 + \epsilon)L]$ by property 3 in Definition 1.9. By the triangle inequality:

$$\begin{aligned} \|r_u, r_v\| &\leq \|u, v\| + 2\beta\epsilon L \leq (1 + (1 + 2\beta)\epsilon)L \\ \|r_u, r_j^*\| &\leq \|x, y\| + 2\beta L \leq (1 + (1 + 2\beta)\epsilon)L \\ \|u, v\| &\leq \|r_u, r_v\| + 2\beta\epsilon L \quad \text{and} \quad \|x, y\| \leq \|r_u, r_j^*\| + 2\beta\epsilon L \end{aligned} \tag{8}$$

Furthermore, since $L \leq \|u, v\|, \|x, y\| \leq (1 + \epsilon)L$, it follows that:

$$\|u, v\| \leq (1 + \epsilon)\|x, y\| \quad \text{and} \quad \|x, y\| \leq (1 + \epsilon)\|u, v\| \tag{9}$$

Claim 4.4. $\|r_v, r_j^*\| \leq (8\beta + 6)\epsilon L$.

Proof: Recall that $\|r_u, r_j^*\| \leq \|r_u, r_v\|$. Let p be the projection of r_j^* onto the segment $r_u r_v$ (see Figure 5). Since $\angle r_v r_u r_j^* \leq \epsilon$, $\|r_j^*, p\| \leq \sin(\epsilon)\|r_u, r_j^*\| \leq \sin(\epsilon)\|r_u, r_v\| \leq \epsilon(1 + (1 + 2\beta)\epsilon)L$. We have:

$$\begin{aligned} \|r_v, r_j^*\| &\leq \|p, r_j^*\| + \|r_v, p\| \leq \|p, r_j^*\| + \|r_u, r_v\| - (\|r_u, r_j^*\| - \|r_j^*, p\|) \\ &\leq (\|r_u, r_v\| - \|r_u, r_j^*\|) + 2\epsilon(1 + (1 + 2\beta)\epsilon)L \end{aligned} \tag{10}$$

We now bound $(\|r_u, r_v\| - \|r_u, r_j^*\|)$. By Equation (8) and Equation (9), it holds that:

$$\|r_u, r_v\| - \|r_u, r_j^*\| \leq \|u, v\| + 2\beta\epsilon L - (\|x, y\| - 2\beta\epsilon L) \leq (4\beta + 1 + \epsilon)\epsilon L \tag{11}$$

Plugging Equation (11) into Equation (10), we get:

$$\|r_v, r_j^*\| \leq (4\beta + 1 + \epsilon)\epsilon L + 2\epsilon(1 + (1 + 2\beta)\epsilon)L \leq (8\beta + 6)\epsilon L \quad (\text{since } \epsilon \leq 1),$$

as claimed. This completes the proof of Claim 4.4. $\square$

Next we continue with the proof of Lemma 4.3. By Claim 4.4, $\|r_v, r_j^*\| < L$ when $\epsilon < 1/(8\beta + 6)$. If the input graph is a UDG, then $\mathcal{E} \neq \emptyset$ only if $L \leq 1$. Thus, $\|r_v, r_j^*\| \leq 1$ and hence, there is an edge (r_v, r_j^*) of length $\|r_v, r_j^*\|$ in the input UDG. (This is the only place, other than starting our construction with a $(1 + \epsilon)$ -spanner for the input UDG, where we exploit the fact that the input graph is a UDG.)

Since $\|r_v, r_j^*\| < L$, the distance between r_v and r_j^* is preserved up to a factor of $(1 + \gamma\epsilon)$ in $H_{<L}$. That is, $d_{H_{<(1+\epsilon)L}}(r_v, r_j^*) \leq (1 + \gamma\epsilon)\|r_v, r_j^*\|$.

Note that r_u, r_v, r_j^* are all in the input point set P by the definition of representatives. By the triangle inequality, it follows that:

$$\begin{aligned} d_{H_{<(1+\epsilon)L}}(u, v) &\leq d_{H_{<(1+\epsilon)L}}(u, x) + \|x, y\| + d_{H_{<(1+\epsilon)L}}(y, r_j^*) + d_{H_{<(1+\epsilon)L}}(r_j^*, r_v) + d_{H_{<(1+\epsilon)L}}(r_v, v) \\ &\leq \beta\epsilon L + \|x, y\| + \beta\epsilon L + (1 + \gamma\epsilon)\|r_v, r_j^*\| + \beta\epsilon L \\ &\leq \|x, y\| + 3\beta\epsilon L + \underbrace{(1 + \gamma\epsilon)}_{\leq 2 \text{ since } \epsilon \leq 1/\gamma} (8\beta + 6)\epsilon L \quad (\text{by Claim 4.4}) \\ &\leq \|x, y\| + (19\beta + 12)\epsilon L \end{aligned} \tag{12}$$

By Equation (9), $\|x, y\| \leq (1 + \epsilon)\|u, v\| \leq \|u, v\| + (1 + \epsilon)\epsilon L \leq \|u, v\| + 2\epsilon L$. Thus, by Equation (12):

$$d_{H_{<(1+\epsilon)L}}(u, v) \leq \|u, v\| + (19\beta + 14)\epsilon L \stackrel{\|u, v\| \geq L/2}{\leq} (1 + 2(19\beta + 14)\epsilon)\|u, v\|.$$

That is, the stretch of (u, v) in $H_{<(1+\epsilon)L}$ is at most $1 + s_{\text{SSA}_{\text{Geom}}}(\beta)\epsilon$ with $s_{\text{SSA}_{\text{Geom}}}(\beta) = 2(19\beta + 14)$. $\square$

Remark 4.5. SSA_{Geom} can be implemented slightly faster, within time $O(|\mathcal{V}|\epsilon^{1-d} + |\mathcal{E}|\log(1/\epsilon))$, by using a data structure that allows us to search for the cone that a representative belongs to in $O(\log(1/\epsilon))$ time. Such a data structure is described in Theorem 5.3.2 in the book by Narasimhan and Smid [53].

Proof: [Proof of Theorem 4.1] We use SSA_{Geom} in place of the abstract SSA in Theorem 1.10 to construct the light spanner. By Lemma 4.3, we have $s_{\text{SSA}}(\beta) = 2(19\beta + 14)$, $\chi = O(\epsilon^{1-d})$ and $\tau(m', n') = O(\epsilon^{1-d})$. Thus, by plugging in the values of χ and τ , we obtain the lightness and the running time as required by Theorem 4.1. The stretch of the spanner is $(1 + \epsilon)(1 + (s_{\text{SSA}}(O(1)) + O(1))\epsilon) = (1 + O(\epsilon))$ when $\epsilon \leq 1$. $\square$

4.2 General Graphs

In this section, we prove Theorem 1.2 by giving a detailed implementation of SSA for general graphs, hereafter SSA_{Gen} . Here we have $t = 2k - 1$ for an integer parameter $k \geq 2$. We will use as a black-box the linear-time construction of sparse spanners in general unweighted graphs by Halperin and Zwick [41].

Theorem 4.6 (Halperin-Zwick [41]). *Given an unweighted n -vertex graph G with m edges, a $(2k - 1)$ -spanner of G with $O(n^{1+\frac{1}{k}})$ edges can be constructed deterministically in $O(m + n)$ time, for any $k \geq 2$.*

SSA_{Gen} (General Graphs): The input is a (L, ϵ, β) -cluster graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \omega)$. The output is $\mathcal{E}^{\text{pruned}}$, initially, $\mathcal{E}^{\text{pruned}} = \emptyset$.

We construct a new *unweighted* graph $J = (V_J, E_J)$ as follows. For each node in $\varphi \in \mathcal{V}$, we add a vertex v_φ to V_J . For each edge $(\varphi_1, \varphi_2) \in \mathcal{V}$, we add an edge $(v_{\varphi_1}, v_{\varphi_2})$ to E_J .

Next, we run Halperin-Zwick's algorithm (Theorem 4.6) on J to construct a $(2k - 1)$ -spanner S_J for J . Then for each edge $(v_{\varphi_1}, v_{\varphi_2})$ in $E(S_J)$, we add the corresponding edge (φ_1, φ_2) to $\mathcal{E}^{\text{pruned}}$.

We next analyze the running time of SSA_{Gen} , and also show that it satisfies the two properties of (Sparsity) and (Stretch) required by the abstract SSA ; these properties are described in Section 1.3.

Lemma 4.7. SSA_{Gen} can be implemented in $O(|\mathcal{V}| + |\mathcal{E}|)$ time. Furthermore, 1. (Sparsity) $\mathcal{E}^{\text{pruned}} = O(n^{1/k})|\mathcal{V}|$, and 2. (Stretch) For each edge $(\varphi_{C_u}, \varphi_{C_v}) \in \mathcal{E}$, $d_{H_{<(1+\epsilon)L}}(u, v) \leq (2k-1)(1+s_{\text{SSA}_{\text{Gen}}}(\beta)\epsilon)w(u, v)$, where $(u, v) = \text{source}(\varphi_{C_u}, \varphi_{C_v})$, $s_{\text{SSA}_{\text{Gen}}}(\beta) = (2\beta + 1)$ and $\epsilon \leq 1$.

Proof: The running time of SSA_{Gen} follows directly from Theorem 4.6. Also, by Theorem 4.6, $|\mathcal{E}^{\text{pruned}}| = O(|\mathcal{V}|^{1+1/k}) = O(n^{1/k}|\mathcal{V}|)$; this implies Item 1.

It remains to prove Item 2: For each edge $(\varphi_{C_u}, \varphi_{C_v}) \in \mathcal{E}$, the stretch in $H_{<(1+\epsilon)L}$ (constructed as described in SSA) of the corresponding edge $(u, v) = \text{source}(\varphi_{C_u}, \varphi_{C_v})$ is at most $(2k-1)(1+(2\beta+1)\epsilon)w(u, v)$. Recall that $H_{<(1+\epsilon)L}$ is the graph obtained by adding the source edges of $\mathcal{E}^{\text{pruned}}$ to $H_{<L}$.

Let (u_1, v_1) be the edge in E_J that corresponds to the edge $(\varphi_{C_u}, \varphi_{C_v})$. By Theorem 4.6, there is a path P between u_1 and v_1 in J such that P contains at most $2k-1$ edges. We write $P = (u_1 = x_0, (x_0, x_1), x_1, (x_1, x_2), \dots, x_p = v_1)$ as an alternating sequence of vertices and edges. Let $\mathcal{P} = (\varphi_0, (\varphi_0, \varphi_1), \varphi_1, (\varphi_1, \varphi_2), \dots, \varphi_p)$ be a path of $\mathcal{G}$, written as an alternating sequence of vertices and edges, that is obtained from P where φ_j corresponds to x_j , $1 \leq j \leq p$. Note that $\varphi_1 = \varphi_{C_u}$ and $\varphi_p = \varphi_{C_v}$.

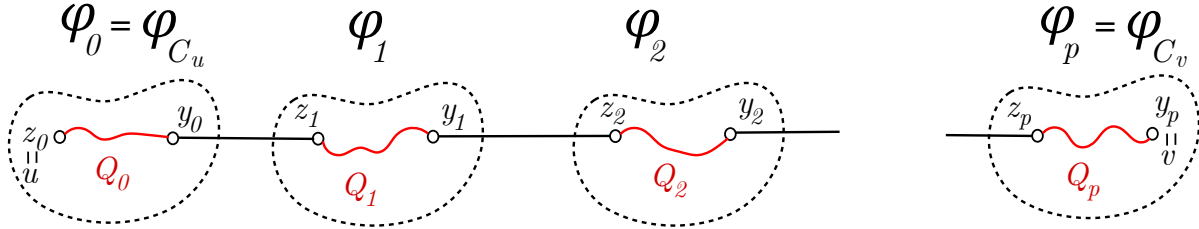


Figure 6: A path from u to v .

Let $\{y_i\}_{i=0}^p$ and $\{z_i\}_{i=0}^p$ be two sequences of vertices of G such that (a) $z_0 = u$ and $y_p = v$, and (b) (y_{i-1}, z_i) is the edge in G corresponding to edge $(\varphi_{i-1}, \varphi_i)$ in $\mathcal{P}$, for $1 \leq i \leq p$. Let Q_i , $0 \leq i \leq p$, be a shortest path in $H_{<L}[C_i]$ between z_i and y_i , where C_i is the cluster corresponding to φ_i . See Figure 6 for an illustration. Observe that $w(Q_i) \leq \beta\epsilon L$ by property 4 in Definition 1.9. Let $P' = Q_0 \circ (y_0, z_1) \circ \dots \circ Q_p$ be a (possibly non-simple) path from u to v in $H_{<(1+\epsilon)L}$; here $\circ$ is the path concatenation operator.

$$\begin{aligned} w(P') &\leq (2k-1)(1+\epsilon)L + (2k)\beta\epsilon L \leq (2k-1)(1+\epsilon+2\beta\epsilon)L \\ &\leq (2k-1)(1+(2\beta+1)\epsilon)w(u, v) \quad (\text{since } w(u, v) \geq L) \end{aligned} \tag{13}$$

Thus, the stretch of edge (u, v) is at most $(2k-1)(1+(2\beta+1)\epsilon)$, as required. $\square$

Proof: [Proof of Theorem 1.2] We use algorithm SSA_{Gen} in place of the abstract SSA in Theorem 1.10 to construct the light spanner. By Lemma 4.3, we have $s_{\text{SSA}}(\beta) = (2\beta + 14)$, $\chi = O(n^{1/k})$ and $\tau(m', n') = O(1)$. Thus, by plugging in the values of χ and τ , we obtain the lightness and the running time as required by Theorem 1.2. The stretch of the spanner is $(2k-1)(1+(s_{\text{SSA}}(O(1)) + O(1))\epsilon) = (2k-1)(1+O(\epsilon))$. By scaling, we get the required stretch of $(2k-1)(1+\epsilon)$. $\square$

4.3 Minor-free Graphs

In this section, we prove a weaker version of Theorem 1.4, where the running time is $O(nr\sqrt{r}\alpha(nr\sqrt{r}, n))$. In Section 9 we show how to achieve a linear running time, via an adaptation of our framework (described

in detail in Section 6) to minor-free graphs.

The implementation of the abstract algorithm SSA for minor-free graphs, hereafter $\text{SSA}_{\text{Minor}}$, simply outputs the edge set $\mathcal{E}$. Note that the stretch in this case is $t = 1 + \epsilon$.

$\text{SSA}_{\text{Minor}}$ (Minor-free Graphs): The input is a (L, ϵ, β) -cluster graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \omega)$. The output is $\mathcal{E}^{\text{pruned}}$.

The algorithm returns $\mathcal{E}^{\text{pruned}} = \mathcal{E}$.

We next analyze the running time of $\text{SSA}_{\text{Minor}}$, and also show that it satisfies the two properties of (Sparsity) and (Stretch) required by the abstract SSA .

Lemma 4.8. *$\text{SSA}_{\text{Minor}}$ can be implemented in $O((|\mathcal{V}| + |\mathcal{E}|))$ time. Furthermore, 1. (Sparsity) $\mathcal{E}^{\text{pruned}} = O(r\sqrt{\log r})|\mathcal{V}|$, and 2. (Stretch) For each edge $(\varphi_{C_u}, \varphi_{C_v}) \in \mathcal{E}$, $d_{H_{<(1+\epsilon)L}}(u, v) \leq (1+\epsilon)(1+s_{\text{SSA}_{\text{Minor}}}(\beta)\epsilon)w(u, v)$, where $(u, v) = \text{source}(\varphi_{C_u}, \varphi_{C_v})$, $s_{\text{SSA}_{\text{Minor}}}(\beta) = 0$ and $\epsilon \leq 1$.*

Proof: The running time of $\text{SSA}_{\text{Minor}}$ follows trivially from the construction. Noting that $\mathcal{G}$ is a minor of the input graph G , $\mathcal{G}$ is K_r -minor-free. Thus, $|\mathcal{E}| = O(r\sqrt{\log r})|\mathcal{V}|$ by the sparsity of minor-free graphs [45, 59]; this implies Item 1. Since we take every edge of $\mathcal{E}$ to $\mathcal{E}^{\text{pruned}}$, the stretch is 1 and hence $s_{\text{SSA}_{\text{Minor}}}(\beta) = 0$, yielding Item 2. $\square$

We are now ready to prove a weaker version of Theorem 1.4 for minor-free graphs, where the running time is $O(nr\sqrt{r}\alpha(nr\sqrt{r}, n))$.

Proof: [Proof of Theorem 1.4] We use algorithm $\text{SSA}_{\text{Minor}}$ in place of the abstract SSA in Theorem 1.10 to construct the light spanner. By Lemma 4.3, we have $s_{\text{SSA}}(\beta) = 0$, $\chi = O(r\sqrt{\log r})$ and $\tau(m', n') = O(1)$. Thus, by plugging in the values of χ and τ , we obtain the lightness claimed in Theorem 1.4 and a running time of $O(nr\sqrt{r}\alpha(nr\sqrt{r}, n))$, for a constant ϵ . The stretch of the spanner is:

$$(1 + \epsilon)(1 + (s_{\text{SSA}}(O(1)) + O(1))\epsilon) = (1 + O(\epsilon))$$

By scaling, we get a stretch of $(1 + \epsilon)$. $\square$

5 Applications of the Unified Framework: Fine-Grained Optimality

In this section, we use the framework outlined in Section 1.3 to obtain all results regarding fine-grained lightness bounds stated in Section 1.2: Theorem 1.5, Theorem 1.6, Theorem 1.7, and Theorem 1.8. We do so by introduce another layer of abstraction via an object that we call *general sparse spanner oracle* (GSSO) in Section 5.1: we show that the existence of GSSO implies the existence of light spanners. In Section 5.2, we construct GSSOs for different class of graphs: general graphs, high dimensional Euclidean spanners, and Steiner Euclidean spanners. Finally, in Section 5.3, we construct a light spanner for minor-free graphs by directly implementing SSO. See Figure 1 for relationships between theorems/lemmas.

5.1 General Sparse Spanner Oracles

We introduce the notion of a *general sparse spanner oracle* (GSSO). Our GSSO for stretch $t = 1 + \epsilon$ coincides with a notion called *spanner oracle*, introduced by Le [47]; nonetheless, our goal is much more ambitious: First we wish to optimize the fine-grained dependencies and second we wish to do so while considering a much wider regime of the stretch parameter t , which may even depend on n .

Definition 5.1 (General Sparse Spanner Oracle). *Let G be an edge-weighted graph and let $t > 1$ be a stretch parameter. A general sparse spanner oracle (GSSO) of G for a given stretch t is an algorithm that, given a subset of vertices $T \subseteq V(G)$ and a distance parameter $L > 0$, outputs in polynomial time a subgraph S of G such that for every pair of vertices $x, y \in T, x \neq y$ with $L \leq d_G(x, y) < 2L$:*

$$d_S(x, y) \leq t \cdot d_G(x, y). \quad (14)$$

We denote a GSSO of G with stretch t by $\mathcal{O}_{G,t}$, and its output subgraph is denoted by $\mathcal{O}_{G,t}(T, L)$, given two parameters $T \subseteq V(G)$ and $L > 0$.

Definition 5.2 (Sparsity). *Given a GSSO $\mathcal{O}_{G,t}$ of a graph G , we define weak sparsity and strong sparsity of $\mathcal{O}_{G,t}$, denoted by $\text{Ws}_{\mathcal{O}_{G,t}}$ and $\text{Ss}_{\mathcal{O}_{G,t}}$ respectively, as follows:*

$$\begin{aligned} \text{Ws}_{\mathcal{O}_{G,t}} &= \sup_{T \subseteq V, L \in \mathbb{R}^+} \frac{w(\mathcal{O}_{G,t}(T, L))}{|T|L} \\ \text{Ss}_{\mathcal{O}_{G,t}} &= \sup_{T \subseteq V, L \in \mathbb{R}^+} \frac{|E(\mathcal{O}_{G,t}(T, L))|}{|T|} \end{aligned} \quad (15)$$

We observe that:

$$\text{Ws}_{\mathcal{O}_{G,t}} \leq t \cdot \text{Ss}_{\mathcal{O}_{G,t}}, \quad (16)$$

since every edge $E(\mathcal{O}_{G,t}(T, L))$ must have weight at most $t \cdot L$; indeed, otherwise we can remove it from $\mathcal{O}_{G,t}(T, L)$ without affecting the stretch. Thus, when t is constant, strong sparsity implies weak sparsity; note, however, that this is not necessarily the case when t is super-constant.

Our main result in this section is to show that for stretch $t \geq 2$, we can construct a light spanner with lightness bound roughly $O(\frac{1}{\epsilon})$ times the sparsity of the spanner oracle (Theorem 5.3). For stretch $t = 1 + \epsilon$, we can construct a light spanner with lightness bound roughly $O(\frac{1}{\epsilon})$ times the sparsity of the spanner oracle plus an additive factor $1/\epsilon^2$.

Theorem 5.3. *Let G be an arbitrary edge-weighted graph that admits a GSSO $\mathcal{O}_{G,t}$ of weak sparsity $\text{Ws}_{\mathcal{O}_{G,t}}$ for $t \geq 2$. Then for any $\epsilon > 0$, we can construct in polynomial time a $t(1 + \epsilon)$ -spanner for G with lightness $\tilde{O}_\epsilon\left(\frac{\text{Ws}_{\mathcal{O}_{G,t}}}{\epsilon}\right)$*

Theorem 5.4. *Let G be an arbitrary edge-weighted graph that admits a GSSO $\mathcal{O}_{G,1+\epsilon}$ of weak sparsity $\text{Ws}_{\mathcal{O}_{G,1+\epsilon}}$ for any $\epsilon > 0$. Then there exists an $(1 + O(\epsilon))$ -spanner for G with lightness $\tilde{O}_\epsilon\left(\frac{\text{Ws}_{\mathcal{O}_{G,t}}}{\epsilon} + \frac{1}{\epsilon^2}\right)$.*

In both Theorem 5.3 and Theorem 5.4, $\tilde{O}_\epsilon(\cdot)$ hides a factor of $\log \frac{1}{\epsilon}$. The proofs of these theorem are presented in Section 5.1.

The bound in Theorem 5.4 improves over the lightness bound due to Le [49] by a factor of $\frac{1}{\epsilon^2}$. The stretch of S in Theorem 5.4 is $1 + O(\epsilon)$, but we can scale it down to $(1 + \epsilon)$ while increasing the lightness by a constant factor. Moreover, this bound is optimal, as we shall assert next. First, the additive factor $\frac{\text{Ws}_{\mathcal{O}_{G,t}}}{\epsilon}$ is unavoidable: the authors showed in [50] that there exists a set of n points in $\mathbb{R}^d$ such that any $(1 + \epsilon)$ -spanner for it must have lightness $\Omega(\epsilon^{-d})$, while the result of Le [49] implies that point sets in $\mathbb{R}^d$ have GSSOs with weak sparsity $O(\epsilon^{1-d})$. Second, the additive factor $\frac{1}{\epsilon^2}$ is tight by the following theorem.

Theorem 5.5. *For any $\epsilon < 1$ and $n \geq (\frac{1}{\epsilon})^{\Theta(\frac{1}{\epsilon})}$, there is an n -vertex graph G admitting a GSSO of stretch $(1 + \epsilon)$ with weak sparsity $O(1)$ such that any $(1 + \epsilon)$ -spanner of G must have lightness $\Omega(\frac{1}{\epsilon^2})$.*

Proof: Le (Theorem 1.3 in [49]), building upon the work of Krauthgamer, Nguyễn and Zondiner [46], showed that graphs with treewidth tw have a 1-spanner oracle with weak sparsity $O(\text{tw}^4)$. Since the treewidth of G in Theorem 3.1 is 4, it has a 1-spanner oracle with weak sparsity $O(1)$; this implies Theorem 5.5. $\square$

Light spanners from GSSO. We now turn to proving Theorem 5.3 and Theorem 5.4. We do so by providing an implementation of SSO using a GSSO. We assume that we are given a GSSO $\mathcal{O}_{G,t}$ with weak sparsity $\text{Ws}_{\mathcal{O}_{G,t}}$. We denote the algorithm by $\text{SSO}_{\text{Oracle}}$. We assume that every edge in G is a shortest path between its endpoints; otherwise, we can safely remove them from the graph.

SSO_{Oracle}: The input is an (L, ϵ, β) -cluster graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \omega)$. The output is a set of edges F .

For each node $\varphi_C \in \mathcal{V}(\mathcal{G})$ corresponding to a cluster C , we choose a $v \in C$. Let S be the set of chosen vertices. Let

$$F = E(\mathcal{O}_{G,t}(S, L/2)) \cup E(\mathcal{O}_{G,t}(S, L)) \cup E(\mathcal{O}_{G,t}(S, 2L)) \quad (17)$$

be the edge set of the spanner returned by the oracle. We then return F .

We now show that $\text{SSO}_{\text{Oracle}}$ has all the properties as described in the abstract SSO.

Lemma 5.6. *Let F be the output of $\text{SSO}_{\text{Oracle}}$. Then $w(F) = O(\text{Ws}_{\mathcal{O}_{G,t}})L \cdot |\mathcal{V}|$. Furthermore, $d_{H_{<2L}}(u, v) \leq t(1 + s_{\text{SSO}_{\text{Oracle}}}(\beta)\epsilon)w(u, v)$ for every edge (u, v) corresponding to an edge in $\mathcal{E}$, where $s_{\text{SSO}_{\text{Oracle}}}(\beta) = 4\beta$ and ϵ is sufficiently smaller than 1, in particular $\epsilon \leq 1/(4\beta)$.*

Proof: Since we only choose exactly one vertex in S per node in $\mathcal{G}$, $|S| = |\mathcal{V}|$. By the definition of the sparsity of an oracle (Definition 5.2), $w(F) \leq \text{Ws}_{\mathcal{O}_{G,t}}(L/2) \cdot |S| + \text{Ws}_{\mathcal{O}_{G,t}}L \cdot |S| + \text{Ws}_{\mathcal{O}_{G,t}}2L \cdot |S| = O(\text{Ws}_{\mathcal{O}_{G,t}})L \cdot |\mathcal{V}|$; this implies the first claim.

Let (u, v) be an edge in G corresponding to an edge $(\varphi_{C_u}, \varphi_{C_v}) \in \mathcal{E}$. We have that $L \leq w(u, v) < 2L$ by property 3 in Definition 1.9. By the construction of S in $\text{SSO}_{\text{Oracle}}$, there are two vertices $u_1 \in C_u$ and $v_1 \in C_v$ that are in S . Let $P_{u_1, u}$ ($P_{v_1, v}$) be the shortest path in $H_{<L}[C_u]$ ($H_{<L}[C_v]$) between u and u_1 (v and v_1). By property 4 in Definition 1.9, we have that $\max\{w(P_{u_1, u}), w(P_{v_1, v})\} \leq \beta\epsilon L$. By the triangle inequality, we have:

$$d_G(u_1, v_1) \leq w(u, v) + 2\beta\epsilon L < (2 + 2\beta\epsilon)L \leq 4L, \quad (18)$$

since $\epsilon \leq 1/\beta$. Also by the triangle equality, it follows that:

$$d_G(u_1, v_1) \geq w(u, v) - 2\beta\epsilon L \geq (1 - 2\beta\epsilon)L \geq L/2, \quad (19)$$

since $\epsilon \leq \frac{1}{4\beta}$. Thus, $d_G(u_1, v_1) \in [L/2, 2L]$. It follows by the definition of GSSO (Definition 5.1) that there is a path, say P_{u_1, v_1} , of weight at most $t \cdot d_G(u_1, v_1)$ between u_1 and v_1 in the graph induced by F . Let $P_{u, v} = P_{u_1, u} \circ P_{u_1, v_1} \circ P_{v_1, v}$ be the path between u and v obtained by concatenating $P_{u_1, u}$, P_{u_1, v_1} , $P_{v_1, v}$. By the triangle inequality, it follows that:

$$\begin{aligned} w(P_{u, v}) &\leq w(P_{u_1, v_1}) + w(P_{u_1, u}) + w(P_{v_1, v}) \leq t \cdot d_G(u_1, v_1) + 2\epsilon\beta L \\ &\stackrel{\text{Eq. (18)}}{=} t \cdot (w(u, v) + 2\epsilon\beta L) + 2\epsilon\beta L \\ &\leq t \cdot (w(u, v) + 4\epsilon\beta L) \leq t \cdot (1 + 4\epsilon\beta)w(u, v) \quad (\text{since } w(u, v) \geq L \text{ and } t \geq 1), \end{aligned} \quad (20)$$

as desired. $\square$

Proof: [Proof of Theorem 5.3] By Theorem 1.10 and Lemma 5.6, we can construct in polynomial time a spanner H with stretch $t(1 + (2s_{\text{SSO}_{\text{Oracle}}}(O(1)) + O(1))\epsilon)$ where $s_{\text{SSO}_{\text{Oracle}}}(\beta) = 8\beta$. Thus, the stretch of H is $t(1 + O(\epsilon))$; we then can recover stretch $t(1 + \epsilon)$ by scaling. The lightness of H is $\tilde{O}_\epsilon((\chi\epsilon^{-1}))$ with $\chi = O(\text{Ws}_{\mathcal{O}_{G,t}})$. That implies a lightness of $\tilde{O}_\epsilon((\text{Ws}_{\mathcal{O}_{G,t}}\epsilon^{-1}))$ as claimed. $\square$

Proof: [Proof of Theorem 5.4] The proof follows the same line of the proof of Theorem 5.3. The difference is that we apply Lemma 5.6 and Theorem 1.10 with $t = 1 + \epsilon$ to construct H . Thus, the stretch of H is $t(1 + O(\epsilon)) = 1 + O(\epsilon)$. Since $\chi = \text{Ws}_{\mathcal{O}_{G,1+\epsilon}}$, the lightness is $\tilde{O}_\epsilon\left(\frac{\text{Ws}_{\mathcal{O}_{G,t}}}{\epsilon} + \frac{1}{\epsilon^2}\right)$ as claimed. $\square$

5.2 Constructing General Sparse Spanner Oracles

We construct GSSOs for different class of graphs: general graphs, high dimensional metric spanners, and Steiner Euclidean spanners. This together with Theorem 5.3 and Theorem 5.4 give Theorem 1.6, Theorem 1.7, and Theorem 1.8.

5.2.1 General graphs and high dimensional metric spaces: Proof of Theorem 1.6 and Theorem 1.8

Theorem 5.7. *The following GSSOs exist.*

1. For any weighted graph G and any $k \geq 2$, $\text{Ws}_{\mathcal{O}_{G,2k-1}} = O(g(n, k))$.
2. For the complete weighted graph G corresponding to any Euclidean space (in any dimension) and for any $t \geq 1$, $\text{Ws}_{\mathcal{O}_{G, \mathcal{O}(t)}} = O(tn^{\frac{1}{t^2}} \log n)$.
3. For the complete weighted graph G corresponding to any finite ℓ_p normed space for $p \in (1, 2]$ and for any $t \geq 1$, $\text{Ws}_{\mathcal{O}_{G, \mathcal{O}(t)}} = O(tn^{\frac{\log t}{tp}} \log n)$.

Theorem 1.6 follows directly from Theorem 5.3 and Item (1) of Theorem 5.7; Theorem 1.8 follows directly from Theorem 5.3 and Item (2) and Item (3) of Theorem 5.7 with $\epsilon = 1/2$; any constant $\epsilon < 1$ works. See Figure 1 for a graphical illustration of the relationships between these theorems. We now focus on proving Theorem 5.7.

General graphs. For a given graph $G(V, E)$ and $T \subseteq V$, we construct another weighted graph $G_T(T, E_T, w_T)$ with vertex set T such that for every two vertices u, v that form a critical pair, we add an edge (u, v) with weight $w_T(u, v) = d_G(u, v)$.

We apply the greedy algorithm [2] to G_T with $t = 2k - 1$ and return the output of the greedy spanner, say S_T , (after replacing each artificial edge by the shortest path between its endpoints) as the output of the oracle $\mathcal{O}_{G,2k-1}$. We now bound the weak sparsity of $\mathcal{O}_{G,2k-1}$.

It was shown (Lemma 2 in [2]) that S_T has girth $2k + 1$ and hence has at most $g(|T|, k)|T| \leq g(n, k)|T|$ edges. It follows that $w(S_T) \leq |g(n, k)|T|2L = O(g(n, k))|T|L$. That implies:

$$\text{Ws}_{\mathcal{O}_{G,2k-1}} = \sup_{T \subseteq V, L \in \mathcal{R}^+} \frac{O(g(n, k))|T|L}{|T|L} = O(n^{1/k}).$$

This implies Item (1) of Theorem 5.7.

High dimensional metric spaces. Let (X, d_X) be a metric space and $\mathcal{P}$ be a partition of (X, d_X) into clusters. We say that $\mathcal{P}$ is Δ -bounded if $\text{Dm}(P) \leq \Delta$ for every $P \in \mathcal{P}$. For each $x \in X$, we denote the cluster containing x in $\mathcal{P}$ by $\mathcal{P}(x)$. The following notion of (t, Δ, δ) -decomposition was introduced by Filtser and Neiman [33].

Definition 5.8 ((t, Δ, η) -decomposition). *Given parameters $t \geq 1, \Delta > 0, \eta \in [0, 1]$, a distribution $\mathcal{D}$ over partitions of (X, d_X) is a (t, Δ, η) -decomposition if:*

- (a) *Every partition $\mathcal{P}$ drawn from $\mathcal{D}$ is $t \cdot \Delta$ -bounded.*
 - (b) *For every $x \neq y \in X$ such that $d_X(x, y) \leq \Delta$, $\Pr_{\mathcal{P} \sim \mathcal{D}}[\mathcal{P}(x) = \mathcal{P}(y)] \geq \eta$*
- (X, d) is (t, η) -decomposable if it has a (t, Δ, η) -decomposition for any $\Delta > 0$.

Claim 5.9. *If (X, d_X) is (t, η) -decomposable, it has a GSSO $\mathcal{O}_{X, O(t)}$ with sparsity $\text{Ws}_{\mathcal{O}_{X, O(t)}} = O(\frac{t \log |X|}{\eta})$. Furthermore, there is a polynomial time Monte Carlo algorithm constructing $\mathcal{O}_{X, O(t)}$ with constant success probability.*

Proof: Let T be a set of terminals given to the oracle $\mathcal{O}_{X, O(t)}$. Let $\mathcal{D}$ be a $(t, 2L, \eta)$ -decomposition of (X, d_X) .

Initially the spanner S has $V(S) = T$ and $E(S) = \emptyset$. We sample $\rho = \frac{2 \ln |T|}{\eta}$ partitions from $\mathcal{D}$, denoted by $\mathcal{P}_1, \dots, \mathcal{P}_\rho$. For each $i \in [\rho]$ and each cluster $C \in \mathcal{P}_i$, if $|T \cap C| \geq 2$, we pick a terminal $t \in C$ and add to S edges from t to all other terminals in C . We then return S as the output of the oracle.

For each partition $\mathcal{P}_i$, the set of edges added to S forms a forest. That implies we add to S at most $|T| - 1$ edges per partition. Thus, $|E(S)| \leq (|T| - 1)\rho = O(\frac{|T| \log |T|}{\eta})$. Observe that $w(S) \leq |E(S)| \cdot t2L = (\frac{2|T|tL \log |T|}{\eta})$ since each edge has weight at most $t \cdot (2L)$. Thus, $\text{Ws}_{\mathcal{O}} = O(\frac{t \log |T|}{\eta}) = O(\frac{t \log |X|}{\eta})$.

It remains to show that with constant probability, $d_S(x, y) \leq O(t)d_X(x, y)$ for every $x \neq y \in T$ such that $L \leq d_X(x, y) < 2L$. Observe by construction that if x and y fall into the same cluster in any partition, there is a 2-hop path of length at most $4tL = O(t)d_X(x, y)$. Thus, we only need to bound the probability that x and y are clustered together in some partition. Observe that the probability that there is no cluster containing both x and y in ρ partitions is at most:

$$(1 - \eta)^\rho = (1 - \eta)^{\frac{2 \ln |T|}{\eta}} \leq \frac{1}{|T|^2}$$

Since there are at most $\frac{|T|^2}{2}$ distinct pairs, by union bound, the desired probability is at least $\frac{1}{2}$. $\square$

Filtser and Neiman [33] showed that any n -point Euclidean metric is $(t, n^{-O(\frac{1}{t^2})})$ -decomposable for any given $t > 1$; this implies Item (2) in Theorem 5.7. If (X, d_X) is an ℓ_p metric with $p \in (1, 2)$, Filtser and Neiman [33] showed that it is $(t, n^{-O(\frac{\log t}{t^2})})$ -decomposable for any given $t > 1$; this implies Item (3) in Theorem 5.7.

5.2.2 Steiner Euclidean Spanners

To prove Theorem 1.7, we allow the oracle to include Steiner points, i.e., points in $\mathbb{R}^d \setminus P$ in the construction of GSSO (Theorem 5.4 remains true for GSSO with Steiner points). Formally, a GSSO with Steiner points, given a subset of points $T \subseteq P$ and a distance parameter $L > 0$, outputs a Euclidean graph $S(V_S, E_S)$ with $T \subseteq V_S$ such that $d_S(x, y) \leq (1 + \epsilon)||x, y||$ for any $x \neq y$ in T ,³ where $||x, y|| \in [L, 2L]$. We denote the oracle by $\mathcal{O}_{P, 1+\epsilon}$. Our construction of the GSSO with Steiner points uses the sparse Steiner $(1 + \epsilon)$ -spanner from our previous work [50] (in the full version) as a black-box.

Theorem 5.10 (Theorem 1.3 [50]). *Given an n -point set $P \in \mathbb{R}^d$, there is a Steiner $(1 + \epsilon)$ -spanner for P with $\tilde{O}_\epsilon(\epsilon^{-(d-1)/2}|P|)$ edges.*

³ $||x, y||$ is the Euclidean distance between two points $x, y \in \mathbb{R}^d$.

Theorem 5.11. Any point set P in $\mathbb{R}^d$ admits a GSSO with Steiner points that has weak sparsity $\mathbf{ws}_{\mathcal{O}_{P,t+\epsilon}} = \tilde{O}_\epsilon(\epsilon^{-(d-1)/2})$.

We note that Theorem 1.7 follows directly from Theorem 5.11 and Theorem 5.4.

Proof: Let $T \subseteq P$ be a subset of points given to the oracle and L be the distance parameter. By Theorem 5.10, we can construct a Steiner $(1 + \epsilon)$ -spanner S for T with $|E(S)| = \tilde{O}_\epsilon(\epsilon^{-(d-1)/2}|T|)$. We observe that:

Observation 5.12. Let $x \neq y$ be two points in T such that $\|x, y\| \leq 2L$, and Q be a shortest path between x and y in S . Then, for any edge e such that $w(e) \geq 4L$, $e \notin Q$ when $\epsilon < 1$.

Proof: Since S is a $(1 + \epsilon)$ -spanner, $w(P) \leq (1 + \epsilon)\|x, y\| \leq (1 + \epsilon)2L < 4L$. $\square$

Let $\mathcal{O}_{P,(1+\epsilon)}(T, L)$ be the graph obtained from S by removing every edge $e \in E(S)$ such that $w(e) \geq 4L$. By Observation 5.12, $\mathcal{O}_{P,(1+\epsilon)}(T, L)$ is a $(1 + \epsilon)$ -spanner for T . Observe that

$$w(\mathcal{O}_{P,(1+\epsilon)}(T, L)) \leq 4L|E(\mathcal{O}_{P,(1+\epsilon)}(T, L))| \leq 4L|E(S)| = \tilde{O}_\epsilon(\epsilon^{-(d-1)/2}|T|L).$$

It follows that $\mathbf{ws}_{\mathcal{O}_{P,1+\epsilon}} = \tilde{O}_\epsilon(\epsilon^{-(d-1)/2})$. This completes the proof of Theorem 5.11.

5.3 Light Spanners for Minor-Free Graphs

In this section, we provide an implementation of SSO for minor-free graphs, which we denote by $\text{SSO}_{\text{Minor}}$. The algorithm simply outputs the edge set $\mathcal{E}$. Note that in this case, we set $t = 1 + \epsilon$.

SSO_{Minor}: The input is an (L, ϵ, β) -cluster graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \omega)$. The output is a set of edges F .

Let F be the subset of edges of G that correspond to edges in $\mathcal{E}$. We then return F .

We now show that $\text{SSO}_{\text{Minor}}$ has all the properties as described in the abstract SSO, which implies Theorem 1.5.

Theorem 1.5. Any K_r -minor-free graph admits a $(1 + \epsilon)$ -spanner with lightness $\tilde{O}_{r,\epsilon}(\frac{r}{\epsilon} + \frac{1}{\epsilon^2})$ for any $\epsilon < 1$ and $r \geq 3$.

Furthermore, for any fixed $r \geq 6$, any $\epsilon < 1$ and $n \geq r + (\frac{1}{\epsilon})^{\Theta(1/\epsilon)}$, there is an n -vertex graph G excluding K_r as a minor for which any $(1 + \epsilon)$ -spanner must have lightness $\Omega(\frac{r}{\epsilon} + \frac{1}{\epsilon^2})$.

Proof: Since we add every edge corresponds to an edge in $\mathcal{E}$ in $\text{SSO}_{\text{Minor}}$, $s_{\text{SSO}_{\text{Minor}}}(\beta) = 0$. By Theorem 1.10 and Lemma 5.6, we can construct in polynomial time a spanner H with stretch $t(1 + (2s_{\text{SSO}_{\text{Minor}}}(O(1)) + O(1))\epsilon) = (1 + O(\epsilon))$; note that $t = (1 + \epsilon)$ in this case. We then can recover stretch $(1 + \epsilon)$ by scaling.

We observe that $\mathcal{G}$ is a minor of G and hence is K_r -minor-free. Thus, by the sparsity of minor-free graphs, $|\mathcal{E}| = O(r\sqrt{\log r})|\mathcal{V}|$. It follows that $w(F) = O(r\sqrt{\log r})L \cdot |\mathcal{V}|$ since every edge in $\mathcal{G}$ has weight at most $2L$. This gives $\chi = O(r\sqrt{\log r})$. By Theorem 1.10 for the case $t = 1 + \epsilon$, The lightness of H is $\tilde{O}_\epsilon((\chi\epsilon^{-1}) + \epsilon^{-2}) = \tilde{O}_{\epsilon,r}(r\epsilon^{-1} + \epsilon^{-2})$ as claimed. $\square$

Part II

Our Unified Framework: The Proof (Section 6 — Section 12)

In this part, we present the proof of Theorem 1.10 in detail. We start by setting in a technical framework on which the proof rests.

6 Unified Framework: Technical Setup

In Section 6.1, we outline a technical framework that we use to prove Theorem 1.10. The proof of Theorem 1.10 boils down to constructions of clusters and associated subgraphs. In Section 7, we show how to design a fast algorithm to find the clusters and the subgraphs. In Section 10, we construct the clusters and the subgraphs that have a small dependency on $1/\epsilon$.

6.1 The Framework

Our starting point is a basic hierarchical partition, which dates back to the early 90s [4, 15], and was used by most if not all of the works on light spanners (see, e.g., [29, 30, 17, 10, 11, 50]). The current paper takes this hierarchical partition approach to the next level by proposing a unified framework.

Let MST be a minimum spanning tree of the input n -vertex m -edge graph $G = (V, E, w)$. Let T_{MST} be the running time needed to construct MST . By scaling, we shall assume w.l.o.g. that the minimum edge weight is 1. Let $\bar{w} = \frac{w(\text{MST})}{m}$. We remove from G all edges of weight larger than $w(\text{MST})$; such edges do not belong to any shortest path, hence removing them does not affect the distances between vertices in G . We define two sets of edges, E_{light} and E_{heavy} , as follows:

$$E_{\text{light}} = \{e \in E : w(e) \leq \frac{\bar{w}}{\epsilon}\} \quad \& \quad E_{\text{heavy}} = E \setminus E_{\text{light}} \quad (21)$$

It could be that $\frac{\bar{w}}{\epsilon} < 1$; in this case, $E_{\text{light}} = \emptyset$. The next observation follows from the definition of $\bar{w}$.

Observation 6.1. $w(E_{\text{light}}) \leq \frac{w(\text{MST})}{\epsilon}$.

Recall that the parameter ϵ is in the stretch $t(1 + \epsilon)$ in Theorem 1.10. It controls the stretch blow-up in Theorem 1.10, and ultimately, the stretch of the final spanner. There is an inherent trade-off between the stretch blow-up (a factor of $1 + \epsilon$) and the blow-up of the other parameters, including runtime and lightness, by at least a factor of $1/\epsilon$.

By Observation 6.1, we can safely add E_{light} to our final spanner, while paying only an additive $+\frac{1}{\epsilon}$ factor to the lightness bound. Hence, as the stretch of a spanner is realized by some edge of the graph, in the spanner construction that follows, it suffices to focus on the stretch for edges in E_{heavy} . Next, we partition the edge set E_{heavy} into subsets of edges, such that for any two edges e, e' in the same subset, their weights are either *almost the same* (up to a factor of $1 + \psi$) or they are *far apart* (by at least a factor of $\frac{1}{\epsilon(1+\psi)}$), where ψ is a parameter to be optimized later. In fast constructions (Section 7), we choose $\psi = \epsilon$ and in optimal lightness constructions (Section 10), we choose $\psi = 1/250$.

Definition 6.2 (Partitioning E_{heavy}). *Let ψ be any parameter in the range $(0, 1]$. Let $\mu_\psi = \lceil \log_{1+\psi} \frac{1}{\epsilon} \rceil$. We partition E_{heavy} into subsets $\{E^\sigma\}_{\sigma \in [\mu_\psi]}$ such that $E^\sigma = \cup_{i \in \mathbb{N}^+} E_i^\sigma$ where:*

$$E_i^\sigma = \left\{ e : \frac{L_i}{1 + \psi} \leq w(e) < L_i \right\} \text{ with } L_i = L_0 / \epsilon^i, L_0 = (1 + \psi)^\sigma \bar{w} . \quad (22)$$

By definition, we have $L_i = L_{i-1}/\epsilon$ for each $i \geq 1$. Readers may notice that if $\log_{1+\psi} \frac{1}{\epsilon}$ is not an integer, by the definition of E^σ , it could be that $E^{\mu_\psi} \cap E^1 \neq \emptyset$, in which case $\{E^\sigma\}_{\sigma \in [\mu_\psi]}$ is not really a partition of E_{heavy} . This can be fixed by taking to E^{μ_ψ} edges that are not in $\cup_{1 \leq \sigma \leq \mu_\psi - 1} E^\sigma$. We henceforth assume that $\{E^\sigma\}_{\sigma \in [\mu_\psi]}$ is a partition of E_{heavy} . The following lemma shows that it suffices to focus on the stretch of edges in E^σ , for an arbitrary $\sigma \in [\mu_\psi]$.

Lemma 6.3. *If for every $\sigma \in [\mu_\psi]$, we can construct a k -spanner $H^\sigma \subseteq G$ for E^σ with lightness at most Light_{H^σ} in time $\text{Time}_{H^\sigma}(m, n)$ (where Light_{H^σ} and $\text{Time}_{H^\sigma}(m, n)$ do not depend on σ), then we can construct a k -spanner for G with lightness $O\left(\frac{\text{Light}_{H^\sigma} \log(1/\epsilon)}{\psi} + \frac{1}{\epsilon}\right)$ in time $O\left(\frac{\text{Time}_{H^\sigma}(m, n) \log(1/\epsilon)}{\psi} + T_{\text{MST}}\right)$.*

Proof: Let H be a graph with $V(H) = V(G)$ and $E(H) = E_{light} \cup \left(\cup_{\sigma \in [\mu_\psi]} H^\sigma\right)$. The fact that H is a k -spanner of G follows directly from the fact that the stretch of a spanner is realized by some edge of the graph. The lightness bound follows from the fact that $\mu_\psi = O(\frac{\log(1/\epsilon)}{\log(1+\psi)}) = O(\log(1/\epsilon)/\psi)$ and Observation 6.1.

To bound the running time, note that the time needed to construct E_{light} is $T_{\text{MST}} + O(m) = O(T_{\text{MST}})$. Since we remove edges of weight at least MST from G and every edge in E_{heavy} has a weight at least $\frac{\bar{w}}{\epsilon} = \frac{w(\text{MST})}{\epsilon m}$, the number of sets that each E^σ is partitioned to is $O(\log_{1/((1+\psi)\epsilon)}(\epsilon m)) = O(\log(m))$ for any $\epsilon \leq 1/2$. Thus, the partition of E_{heavy} can be trivially constructed in $O(m)$ time. The running time bound now follows. $\square$

We shall henceforth focus on constructing a spanner for E^σ , for an arbitrarily fixed $\sigma \in [\mu_\psi]$. In what follows we present a clustering framework for constructing a spanner H^σ for E^σ with *stretch* $t(1+\epsilon)$. We will assume that ϵ is sufficiently smaller than 1.

Subdividing MST. We subdivide each edge $e \in \text{MST}$ of weight more than $\bar{w}$ into $\lceil \frac{w(e)}{\bar{w}} \rceil$ edges of weight (of at most $\bar{w}$ and at least $\bar{w}/2$ each) that sums to $w(e)$. (New edges do not have to have equal weights.) Let $\widetilde{\text{MST}}$ be the resulting subdivided MST. We refer to vertices that are subdividing the MST edges as *virtual vertices*. Let $\tilde{V}$ be the set of vertices in V and virtual vertices; we call $\tilde{V}$ the *extended set* of vertices. Let $\tilde{G} = (\tilde{V}, \tilde{E})$ be the graph that consists of the edges in $\widetilde{\text{MST}}$ and E^σ .

Observation 6.4. $|\tilde{E}| = O(m)$.

Proof: It suffices to show that $|E(\widetilde{\text{MST}})| = O(m)$. Indeed, since $w(\widetilde{\text{MST}}) = w(\text{MST})$ and each edge of $\widetilde{\text{MST}}$ has weight at least $\bar{w}/2$, we have $|E(\widetilde{\text{MST}})| \leq 2m$. $\square$

The $t(1+\epsilon)$ -spanner that we construct for E^σ is a subgraph of $\tilde{G}$ containing all edges of $\widetilde{\text{MST}}$; we can enforce this assumption by adding the edges of $\widetilde{\text{MST}}$ to the spanner. By replacing the edges of $\widetilde{\text{MST}}$ by those of MST , we can transform any subgraph of $\tilde{G}$ that contains the entire tree $\widetilde{\text{MST}}$ to a subgraph of G that contains the entire tree MST . We denote by $\tilde{H}^\sigma$ the $t(1+\epsilon)$ -spanner of E^σ in $\tilde{G}$; by abusing the notation, we will write H^σ rather than $\tilde{H}^\sigma$ in the sequel, under the understanding that in the end we transform H^σ to a subgraph of G .

Recall that $E^\sigma = \cup_{i \in \mathbb{N}^+} E_i^\sigma$ where E_i^σ is the set of edges defined in Equation (22). We refer to edges in E_i^σ as *level- i edges*. We say that a level i is empty if the set E_i^σ of level- i edges is empty; in the sequel, we shall only consider the nonempty levels.

Claim 6.5. *The number of (nonempty) levels is $O(\log m)$.*

Proof: The claim follows from the fact that every edge of E^σ has weight at least $\frac{\bar{w}}{\epsilon}$ and at most $w(\text{MST}) = m\bar{w}$, and the weight of edges in E_{i+1}^σ is at least $\frac{1}{(1+\psi)\epsilon}$ times the weight of edges E_i^σ . $\square$

Our construction crucially relies on a *hierarchy of clusters*. A *cluster* in a graph is simply a subset of vertices in the graph. Nonetheless, as will become clear soon, we care also about *edges* connecting vertices in the cluster, and of the properties that these edges possess. Our hierarchy of clusters, denoted by $\mathcal{H} = \{\mathcal{C}_1, \mathcal{C}_2, \dots\}$ satisfies the following properties:

- **(P1)** For any $i \geq 1$, each $\mathcal{C}_i$ is a partition of $\tilde{V}$. When i is large enough, $\mathcal{C}_i$ contains a single set $\tilde{V}$ and $\mathcal{C}_{i+1} = \emptyset$.
- **(P2)** $\mathcal{C}_i$ is an $\Omega(\frac{1}{\epsilon})$ -refinement of $\mathcal{C}_{i+1}$, i.e., every cluster $C \in \mathcal{C}_{i+1}$ is obtained as the union of $\Omega(\frac{1}{\epsilon})$ clusters in $\mathcal{C}_i$ for $i \geq 1$.
- **(P3)** For each cluster $C \in \mathcal{C}_i$, we have $\text{Dm}(H^\sigma[C]) \leq gL_{i-1}$, for a sufficiently large constant g to be determined later. (Recall that L_i is defined in Equation (22).)

Remark 6.6. (1) We construct H^σ along with the cluster hierarchy. Suppose that at some step s of the algorithm, we construct a level- i cluster C . Let H_s^σ be H^σ at step s . We shall maintain (P3) by maintaining the invariant that $\text{Dm}(H_s^\sigma[C]) \leq gL_{i-1}$; indeed, adding more edges in later steps of the algorithm does not increase the diameter of the subgraph induced by C .

(2) It is time-expensive to compute the diameter of a cluster exactly. Thus, we explicitly associate with each cluster $C \in \mathcal{C}_i$ a proxy parameter of the diameter during the course of the construction. This proxy parameter has two properties: (a) it is at least the diameter of the cluster, and (b) it is lower-bounded by $\Omega(L_{i-1})$. Property (a) is crucial in arguing for the stretch of the spanner. Property (b) is crucial to have an upper bound on the number of level- i clusters contained in a level- $(i+1)$ cluster, which speeds up its (the level- $(i+1)$ cluster's) construction.

When ϵ is sufficiently small, specifically smaller than the constant hiding in the Ω -notation in property (P2) by at least a factor of 2, it holds that $|\mathcal{C}_{i+1}| \leq |\mathcal{C}_i|/2$, yielding a geometric decay in the number of clusters at each level of the hierarchy. This geometric decay is crucial to our fast constructions.

Our construction of the cluster hierarchy $\mathcal{H}$ will be carried out level by level, starting from level 1. After we construct the set of level- $(i+1)$ clusters, we compute a subgraph $H_i^\sigma \subseteq G$ as stated in Theorem 1.10. The final spanner H^σ is obtained as the union of all subgraphs $\{H_i^\sigma\}_{i \in \mathbb{N}^+}$. To bound the weight of H^σ , we rely on a potential function Φ that is formally defined as follows:

Definition 6.7 (Potential Function Φ). We use a potential function $\Phi : 2^{\tilde{V}} \rightarrow \mathbb{R}^+$ that maps each cluster C in the hierarchy $\mathcal{H}$ to a potential value $\Phi(C)$, such that the total potential of clusters at level 1 satisfies:

$$\sum_{C \in \mathcal{C}_1} \Phi(C) \leq w(\text{MST}) . \quad (23)$$

Level- i potential is defined as $\Phi_i = \sum_{C \in \mathcal{C}_i} \Phi(C)$ for any $i \geq 1$. The potential change at level i , denoted by Δ_i for every $i \geq 1$, is defined as:

$$\Delta_i = \Phi_{i-1} - \Phi_i . \quad (24)$$

The key to our framework is the following lemma.

Lemma 6.8. Let $\psi \in (0, 1]$, $t \geq 1$, $\epsilon \in (0, 1)$ be parameters, and $E^\sigma = \cup_{i \in \mathbb{N}^+} E_i^\sigma$ be the set of edges defined in Equation (22). Let $\{a_i\}_{i \in \mathbb{N}^+}$ be a sequence of positive real numbers such that $\sum_{i \in \mathbb{N}^+} a_i \leq A \cdot w(\text{MST})$ for some $A \in \mathbb{R}^+$. Let $H_0 = \text{MST}$. For any level $i \geq 1$, if we can compute all subgraphs $H_1, \dots, H_i \subseteq G$ as well as the cluster sets $\{\mathcal{C}_1, \dots, \mathcal{C}_i, \mathcal{C}_{i+1}\}$ in total runtime $O(\sum_{j=1}^i (|\mathcal{C}_j| + |E_j^\sigma|)f(n, m) + m)$ for some function $f(\cdot, \cdot)$ such that:

- (1) $w(H_i) \leq \lambda \Delta_{i+1} + a_i$ for some $\lambda \geq 0$,

(2) for every $(u, v) \in E_i^\sigma$, $d_{H_{<L_i}}(u, v) \leq t(1 + \rho \cdot \epsilon)w(u, v)$ when $\epsilon \in (0, \epsilon_0)$ for some constants ρ and ϵ_0 , where $H_{<L_i}$ is the spanner constructed for edges of G of weight less than L_i .

Then we can construct a $t(1 + \rho\epsilon)$ -spanner for $G(V, E)$ with lightness $O(\frac{\lambda + A + 1}{\psi} \log \frac{1}{\epsilon} + \frac{1}{\epsilon})$ in time $O(\frac{mf(n, m)}{\psi} \log \frac{1}{\epsilon} + T_{\text{MST}})$ when $\epsilon \in (1, \epsilon_0)$.

Proof: Let $H^\sigma = \cup_{i \in \mathbb{N}} H_i$. The stretch bound $t(1 + \rho\epsilon)$ follows directly from the fact that $E^\sigma = \cup_{i \in \mathbb{N}^+} E_i^\sigma$, Item (2), and Lemma 6.3. By condition (1) of Lemma 1.10 and Equation (23), we have:

$$w(H^\sigma) \leq \lambda \sum_{i \in \mathbb{N}^+} \Delta_i + \sum_{i \in \mathbb{N}^+} a_i + w(\text{MST}) \leq \lambda \cdot \Phi_1 + A \cdot w(\text{MST}) + w(\text{MST}) \leq (\lambda + A + 1)w(\text{MST}).$$

This and Lemma 6.3 implies the lightness upper bound; here $\text{Light}_{H^\sigma} = (O(\lambda) + A + 1)$. To bound the running time, we note that $\sum_{i \in \mathbb{N}^+} |E_i^\sigma| \leq m$ and by property (P2), we have $\sum_{i \in \mathbb{N}^+} |\mathcal{C}_i| = |\mathcal{C}_1| \sum_{i \in \mathbb{N}^+} \frac{O(1)}{\epsilon^{i+1}} = O(|\mathcal{C}_1|) = O(m)$. Thus, by the assumption of Lemma 1.10, the total running time to construct H^σ is $\text{Time}_{H^\sigma}(m, n) = O((\sum_{i \in \mathbb{N}^+} (|\mathcal{C}_i|) + |E_i|)f(m, n) + m) = O(mf(m, n))$. Plugging this runtime bound on top of Lemma 6.3 yields the required runtime bound in Lemma 1.10. $\square$

Remark 6.9. In Lemma 6.8, we construct spanners for edges of G level by level, starting from level 1. By Item (2), when constructing spanners for edges in E_i^σ , we could assume by induction that all edges of weight less than $L_i/(1 + \psi)$ already have stretch $t(1 + \rho\epsilon)$ in the spanner constructed so far, denoted by $H_{<L_i/(1+\psi)}$. By defining $H_{<L_i} = H_{<L_i/(1+\psi)} \cup H_i$, we get a spanner for edges of length less than L_i .

In summary, two important components in our spanner construction is a hierarchy of clusters and a potential function as defined in Definition 6.7. In Section 6.2, we present a construction of level-1 clusters and a general principle for assigning potential values to clusters. The construction of clusters at any level $i + 1$ for $i \geq 1$, which basically gives the proof of Theorem 1.10, is presented in Section 7 and Section 10.

6.2 Designing A Potential Function

In this section, we present in detail the underlying principle used to design the potential function Φ in Definition 6.7. We start by constructing and assigning potential values for level-1 clusters.

Lemma 6.10. In time $O(m)$, we can construct a set of level-1 clusters $\mathcal{C}_1$ such that, for each cluster $C \in \mathcal{C}_1$, the subtree $\widetilde{\text{MST}}[C]$ of $\widetilde{\text{MST}}$ induced by C satisfies $L_0 \leq \text{Dm}(\widetilde{\text{MST}}[C]) \leq 14L_0$.

Proof: We apply a simple greedy construction to break $\widetilde{\text{MST}}$ into a set $\mathcal{S}$ of subtrees of diameter at least L_0 and at most $5L_0$ as follows. (1) Repeatedly pick a vertex v in a component T of diameter at least $4L_0$, break a minimal subtree of radius at least L_0 with center v from T , and add the minimal subtree to $\mathcal{S}$. (2) For each remaining component T' after step (1), there must be an $\widetilde{\text{MST}}$ edge e connecting T' and a subtree $T \in \mathcal{S}$ formed in step (1); we add T' and e to T . Finally, we form $\mathcal{C}_1$ by taking the vertex set of each subtree in $\mathcal{S}$ to be a level-1 cluster. The running time bound follows directly from the construction.

We now bound the diameter of each subtree in $\mathcal{S}$. In step (1), the diameter is at most $2(L_0 + \bar{w})$. In step (2), each subtree T is augmented by subtrees of diameter at most $4L_0$ via $\widetilde{\text{MST}}$ edges in a star-like way. Thus, the diameter of the resulting subtrees is at most $2(L_0 + \bar{w}) + 2(4L_0 + \bar{w}) \leq 14L_0$, as required. $\square$

By choosing $g \geq 14$, clusters in $\mathcal{C}_1$ satisfy properties (P1) and (P3). Note that (P2) is not applicable to level-1 clusters by definition. As for (P3), $\text{Dm}(H^\sigma[C]) \leq 14L_0$, for each $C \in \mathcal{C}_1$.

Next, we assign a potential value for each level-1 cluster as follows:

$$\Phi(C) = \text{Dm}(\widetilde{\text{MST}}[C]) \quad \forall C \in \mathcal{C}_1 \quad (25)$$

We now claim that the total potential of all clusters at level 1 is at most $w(\text{MST})$ as stated in Definition 6.7.

Lemma 6.11. $\Phi_1 \leq w(\text{MST})$.

Proof: By definition of Φ_1 , we have:

$$\Phi_1 = \sum_{C \in \mathcal{C}_1} \Phi(C) = \sum_{C \in \mathcal{C}_1} \text{Dm}(\widetilde{\text{MST}}[C]) \leq \sum_{C \in \mathcal{C}_1} w(\widetilde{\text{MST}}[C]) \leq w(\widetilde{\text{MST}}) = w(\text{MST}).$$

The penultimate inequality holds since level-1 clusters induce vertex-disjoint subtrees of $\widetilde{\text{MST}}$. $\square$

While the potential of a level-1 cluster is the diameter of the subtree induced by the cluster, the potential assigned to a cluster at level at least 2 need not be the diameter of the cluster. Instead, it is an *overestimate* of the cluster's diameter, as imposed by the following *potential-diameter (PD) invariant*.

PD Invariant: For every cluster $C \in \mathcal{C}_i$ and any $i \geq 1$, $\text{Dm}(H_{<L_{i-1}}[C]) \leq \Phi(C)$. (Recall that $H_{<L_{i-1}}$ is the spanner constructed for edges of G of weight less than L_{i-1} , as defined in Lemma 6.8.)

Remark 6.12. As discussed in Remark 6.6, it is time-expensive to compute the diameter of each cluster. By the PD Invariant, we can use the potential $\Phi(C)$ of a cluster $C \in \mathcal{C}_i$ as an upper bound on the diameter of $H_{<L_{i-1}}[C]$. As we will demonstrate in Section 7, $\Phi(C)$ can be computed efficiently.

To define potential values for clusters at levels at least 2, we introduce a *cluster graph*, in which the nodes correspond to clusters. We shall derive the potential values of clusters via their *structure* in the cluster graph, as described next.

Definition 6.13 (Cluster Graph). A cluster graph at level $i \geq 1$, denoted by $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}'_i, \omega)$, is a simple graph where each node corresponds to a cluster in $\mathcal{C}_i$ and each inter-cluster edge corresponds to an edge between vertices that belong to the corresponding clusters. We assign weights to both nodes and edges as follows: for each node $\varphi_C \in \mathcal{V}_i$ corresponding to a cluster $C \in \mathcal{C}_i$, $\omega(\varphi_C) = \Phi(C)$, and for each edge $\mathbf{e} = (\varphi_{C_u}, \varphi_{C_v}) \in \mathcal{E}'_i$ corresponding to an edge (u, v) of $\tilde{G}$, $\omega(\mathbf{e}) = w(u, v)$.

Remark 6.14. The notion of cluster graphs in Definition 6.13 is slightly different from that of (L, ϵ, β) -cluster graphs defined in Definition 1.9. In particular, cluster graphs in Definition 6.13 have weights on both edges and nodes, while (L, ϵ, β) -cluster graphs in Definition 1.9 have weights on edges only.

In our framework, we want the cluster graph $\mathcal{G}_i$ to have the following basic properties.

Definition 6.15 (Properties of $\mathcal{G}_i$). (1) The edge set $\mathcal{E}'_i$ of $\mathcal{G}_i$ is the union $\widetilde{\text{MST}}_i \cup \mathcal{E}_i$, where $\widetilde{\text{MST}}_i$ is the set of edges corresponding to edges in $\widetilde{\text{MST}}$ and $\mathcal{E}_i$ is the set of edges corresponding to a subset of edges in E_i^σ .

(2) $\widetilde{\text{MST}}_i$ induces a spanning tree of $\mathcal{G}_i$. We abuse notation by using $\widetilde{\text{MST}}_i$ to denote the induced spanning tree.

At the outset of the construction of level- $(i+1)$ clusters, we construct a cluster graph $\mathcal{G}_i$. We assume that the spanning tree $\widetilde{\text{MST}}_i$ of $\mathcal{G}_i$ is given, as we construct the tree by the end of the construction of level- i clusters. After we complete the construction of level- $(i+1)$ clusters, we construct $\widetilde{\text{MST}}_{i+1}$ for the next level.

Observation 6.16. *At level 1, both $\mathcal{V}_1$ and $\widetilde{\text{MST}}_1$ can be constructed in $O(m)$ time.*

Proof: Edges of $\widetilde{\text{MST}}_1$ correspond to the edges of $\widetilde{\text{MST}}$ that do not belong to any level-1 cluster, i.e., to any $\widetilde{\text{MST}}[C]$, where $C \in \mathcal{C}_1$. Thus, the observation follows from Observation 6.4 and Lemma 6.10. $\square$

The structure of level- $(i+1)$ clusters. Next, we describe how to construct the level- $(i+1)$ clusters via the cluster graph $\mathcal{G}_i$. We shall construct a collection of subgraphs $\mathbb{X}$ of $\mathcal{G}_i$, and then map each subgraph $\mathcal{X} \in \mathbb{X}$ to a cluster $C_{\mathcal{X}} \in \mathcal{C}_{i+1}$ as follows:

$$C_{\mathcal{X}} = \cup_{\varphi_C \in \mathcal{V}(\mathcal{X})} C. \quad (26)$$

That is, $C_{\mathcal{X}}$ is the union of all level- i clusters that correspond to nodes in $\mathcal{X}$.

For any subgraph $\mathcal{X}$ in a cluster graph, we denote by $\mathcal{V}(\mathcal{X})$ and $\mathcal{E}(\mathcal{X})$ the vertex and edge sets of $\mathcal{X}$, respectively. To guarantee properties (P1)-(P3) defined before Remark 6.6 for clusters in $\mathcal{C}_{i+1}$, we will make sure that subgraphs in $\mathbb{X}$ satisfy the following properties:

- **(P1’).** $\{\mathcal{V}(\mathcal{X})\}_{\mathcal{X} \in \mathbb{X}}$ is a partition of $\mathcal{V}_i$.
- **(P2’).** $|\mathcal{V}(\mathcal{X})| = \Omega(\frac{1}{\epsilon})$.
- **(P3’).** $L_i \leq \text{Adm}(\mathcal{X}) \leq gL_i$.

Recall that $\text{Adm}(\mathcal{X})$ is the augmented diameter of $\mathcal{X}$, a variant of diameter defined for graphs with weights on both nodes and edges, see Section 2. Recall that the augmented diameter of $\mathcal{X}$ is at least the diameter of the corresponding cluster $C_{\mathcal{X}}$.

We then set the potential of cluster $C_{\mathcal{X}}$ corresponding to subgraph $\mathcal{X}$ as:

$$\Phi(C_{\mathcal{X}}) = \text{Adm}(\mathcal{X}). \quad (27)$$

Thus, the augmented diameter of any such subgraph $\mathcal{X}$ will be the weight of the corresponding node in the level- $(i+1)$ cluster graph $\mathcal{G}_{i+1}$. Our goal is to construct H_i along with $\mathcal{C}_{i+1}$ as guaranteed by Theorem 1.10. H_i consists of a subset of the edges in E_i^σ (and in the case of optimal lightness constructions, some edges of G as well). We can assume that the vertex set of H_i is just the entire set V . Up to this point, we have not explained yet how H_i is constructed, since the exact construction of H_i depends on specific incarnations of our framework, which may change from one graph class to another.

While properties (P1’) and (P2’) directly imply properties (P1) and (P2) of $C_{\mathcal{X}}$, property (P3’) does not directly imply property (P3); although the diameter of any weighted subgraph (with edge and vertex weights) is upper bounded by its augmented diameter, we need to guarantee that the (corresponding) edges of $\mathcal{X}$ belong to $H_{<L_i}$. Indeed, without this condition, the diameter of $H_{<L_i}$ could be much larger than the augmented diameter of $\mathcal{X}$.

Lemma 6.17. *Let $\mathcal{X} \in \mathbb{X}$ be a subgraph of $\mathcal{G}_i$ satisfying properties (P1’)-(P3’). Suppose that for every edge $(\varphi_{C_u}, \varphi_{C_v}) \in \mathcal{E}(\mathcal{X})$, $(u, v) \in H_{<L_i}$. By setting the potential value of $C_{\mathcal{X}}$ to be $\Phi(C_{\mathcal{X}}) = \text{Adm}(\mathcal{X})$ for every $\mathcal{X} \in \mathbb{X}$, the PD Invariant is satisfied, and that $C_{\mathcal{X}}$ satisfies all properties (P1)-(P3).*

Proof: It can be seen directly that properties (P1’) and (P2’) of $\mathcal{X}$ directly imply properties (P1) and (P2) of $C_{\mathcal{X}}$, respectively. We prove, by induction on i , that property (P3) holds and that the PD Invariant is satisfied. The basis $i = 1$ is trivial. For the induction step, we assume inductively that for each cluster $C \in \mathcal{C}_i$, $\text{Dm}(H_{<L_{i-1}})[C] \leq gL_{i-1}$ and that the PD Invariant is satisfied: $\Phi(C) \geq \text{Dm}(H_{<L_{i-1}})[C]$. Consider any level- $(i+1)$ cluster $C_{\mathcal{X}}$ corresponding to a subgraph $\mathcal{X} \in \mathbb{X}$. Let $H_{C_{\mathcal{X}}}$ be the graph obtained by first taking the union $\cup_{\varphi_C \in \mathcal{V}(\mathcal{X})} H_{<L_{i-1}}[C]$ and then adding in the edge set

$\{(u, v)\}_{(\varphi_{C_u}, \varphi_{C_v}) \in \mathcal{E}(\mathcal{X})}$. Observe that $H_{C_{\mathcal{X}}}$ is a subgraph of $H_{<L_i}$ by the assumption that $(u, v) \in H_{<L_i}$ for every edge $(\varphi_{C_u}, \varphi_{C_v}) \in \mathcal{E}(\mathcal{X})$. We now show that $\text{Dm}(H_{C_{\mathcal{X}}}) \leq \text{Adm}(\mathcal{X})$, which is at most gL_i by property (P3'). This would imply both property (P3) and the PD Invariant for $C_{\mathcal{X}}$ since $\Phi(C_{\mathcal{X}}) = \text{Adm}(\mathcal{X})$, which would complete the proof of the induction step.

Let u, v be any two vertices in $H_{C_{\mathcal{X}}}$ whose shortest distance in $H_{C_{\mathcal{X}}}$ realizes $\text{Dm}(H_{C_{\mathcal{X}}})$. Let $\varphi_{C_u}, \varphi_{C_v}$ be the two nodes in $\mathcal{X}$ that correspond to two clusters C_u, C_v containing u and v , respectively. Let $\mathcal{P}_{u,v}$ a path in $\mathcal{G}_i$ of minimum augmented weight between φ_{C_u} and φ_{C_v} . Observe that $\omega(\mathcal{P}_{u,v}) \leq \text{Adm}(\mathcal{X})$. We now construct a path $P_{u,v}$ between u and v in $H_{C_{\mathcal{X}}}$ as follows. We write $\mathcal{P}_{u,v} \equiv (\varphi_{C_u} = \varphi_{C_1}, \mathbf{e}_1, \varphi_{C_2}, \mathbf{e}_2, \dots, \varphi_{C_\ell} = \varphi_{C_v})$ as an alternating sequence of nodes and edges. For every $1 \leq p \leq \ell - 1$, let (u_p, v_p) be the edge in E_i^σ that corresponds to $\mathbf{e}_p$. We then define $v_0 = u, u_\ell = v$ and

$$P_{u,v} = Q_{H_{<L_{i-1}}[C_1]}(v_0, u_1) \circ (u_1, v_1) \circ Q_{H_{<L_{i-1}}[C_2]}(v_1, u_2) \circ (u_2, v_2) \circ \dots \circ Q_{H_{<L_{i-1}}[C_\ell]}(v_{\ell-1}, u_\ell),$$

where $Q_{H_{<L_{i-1}}[C_p]}(v_{p-1}, u_p)$ for $1 \leq p \leq \ell$ denotes the shortest path in the corresponding subgraph (between the endpoints of the respective edge, as specified in all the subscripts), and $\circ$ is the path concatenation operator. By the induction hypothesis for the PD Invariant and i , $w(Q_{H_{<L_{i-1}}[C_p]}(v_{p-1}, u_p)) \leq \omega(\varphi_{C_p})$ for each $1 \leq p \leq \ell$. Thus, $w(P_{u,v}) \leq \omega(\mathcal{P}_{u,v}) \leq \text{Adm}(\mathcal{X})$. It follows that $\text{Dm}(H_{C_{\mathcal{X}}}) \leq w(P_{u,v}) \leq \text{Adm}(\mathcal{X})$ as desired. $\square$

Local potential change. For each subgraph $\mathcal{X} \in \mathbb{X}$, we define the *local potential change* of $\mathcal{X}$, denoted by $\Delta_{i+1}(\mathcal{X})$ as follows:

$$\Delta_{i+1}(\mathcal{X}) \stackrel{\text{def.}}{=} \left(\sum_{\varphi_C \in \mathcal{V}(\mathcal{X})} \Phi(C) \right) - \Phi(C_{\mathcal{X}}) = \left(\sum_{\varphi_C \in \mathcal{V}(\mathcal{X})} \omega(\varphi_C) \right) - \text{Adm}(\mathcal{X}). \quad (28)$$

Claim 6.18. $\Delta_{i+1} = \sum_{\mathcal{X} \in \mathbb{X}} \Delta_{i+1}(\mathcal{X})$.

Proof: By property (P1), subgraphs in $\mathbb{X}$ are vertex-disjoint and cover the vertex set $\mathcal{V}_i$, hence $\sum_{\mathcal{X} \in \mathbb{X}} (\sum_{\varphi_C \in \mathcal{V}(\mathcal{X})} \Phi(C)) = \sum_{C \in \mathcal{C}_i} \Phi(C) = \Phi_i$. Additionally, by the construction of level- $(i+1)$ clusters, $\sum_{\mathcal{X} \in \mathbb{X}} \Phi(C_{\mathcal{X}}) = \sum_{C' \in \mathcal{C}_{i+1}} \Phi(C') = \Phi_{i+1}$. Thus, we have:

$$\sum_{\mathcal{X} \in \mathbb{X}} \Delta_{i+1}(\mathcal{X}) = \sum_{\mathcal{X} \in \mathbb{X}} \left(\left(\sum_{\varphi_C \in \mathcal{V}(\mathcal{X})} \Phi(C) \right) - \Phi(C_{\mathcal{X}}) \right) = \Phi_i - \Phi_{i+1} = \Delta_{i+1},$$

as claimed. $\square$

The decomposition of the (global) potential change into local potential changes makes the task of analyzing the spanner weight (Item (1) in Theorem 1.10) easier as we can do so locally. Specifically, we often construct H_i by considering each node in $\mathcal{V}_i$ and taking a subset of (the corresponding edges of) the edges incident to the node to H_i . We then calculate the number of edges taken to H_i incident to all nodes in $\mathcal{X}$, and bound their total weight by the local potential change of $\mathcal{X}$. By summing up over all $\mathcal{X}$, we obtain a bound on $w(H_i)$ in terms of the (global) potential change Δ_{i+1} .

6.3 Summary

We have introduced the technical framework (Lemma 6.8) for constructing light spanners that we will use to both design fast construction of light spanners (Section 7) and spanners with optimal lightness

(Section 10). The construction boils down to constructing two objects: (a) clusters for level i satisfying all properties (P1)-(P3) and (b) a spanner H_i for E_i^σ whose weight is bounded by potential change at level i (Item (1) in Lemma 6.8). The cluster construction is based on a cluster graph Definition 6.13: each level $i + 1$ cluster $\mathcal{X}$ corresponds to a subgraph of the cluster graph $\mathcal{G}_i$ satisfying properties (P1')-(P3'). The detailed construction of level $i + 1$ clusters for fast algorithms is different from the construction for optimal lightness, and is deferred to the corresponding sections (Section 7 and Section 10). Table 1 below summarizes the notation introduced in this section.

Notation	Meaning
E^{light}	$\{e \in E(G) : w(e) \leq w/\epsilon\}$
E^{heavy}	$E \setminus E^{light}$
E^σ	$\bigcup_{i \in \mathbb{N}^+} E_i^\sigma$
E_i^σ	$\{e \in E(G) : \frac{L_i}{1+\psi} \leq w(e) < L_i\}$
g	constant in property (P3).
$\mathcal{G}_i = (V_i, \widetilde{\text{MST}}_i \cup \mathcal{E}_i, \omega)$	cluster graph; see Definition 6.13.
$\mathcal{E}_i$	corresponds to a subset of edges of E_i^σ
$\mathbb{X}$	a collection of subgraphs of $\mathcal{G}_i$
$\mathcal{X}, \mathcal{V}(\mathcal{X}), \mathcal{E}(\mathcal{X})$	a subgraph in $\mathbb{X}$, its vertex set, and its edge set
Φ_i	$\sum_{c \in C_i} \Phi(c)$
Δ_{i+1}	$\Phi_i - \Phi_{i+1}$
$\Delta_{i+1}(\mathcal{X})$	$(\sum_{\phi_C \in \mathcal{X}} \Phi(C)) - \Phi(C_{\mathcal{X}})$
$C_{\mathcal{X}}$	$\bigcup_{\phi_C \in \mathcal{X}} C$

Table 1: Notation introduced in Section 6.

7 Fast Construction: Proof of Theorem 1.10(1)

In this section, we give the detailed construction of level $i + 1$ clusters and graph H_i , thereby proving Item (1) in Theorem 1.10. We set $\psi = \epsilon$ where ψ is the parameter in Equation (22).

We guarantee that the cluster graph $\mathcal{G}_i$ introduced in Section 6.2 satisfies an additional property, which we will exploit for efficient construction.

Definition 7.1 (Additional Properties of $\mathcal{G}_i$). *$\mathcal{G}_i$ satisfies properties (1) and (2) in Definition 6.15, and the following property:*

- (3) $\mathcal{G}_i$ has no removable edge: an edge $(\varphi_{C_u}, \varphi_{C_v}) \in \mathcal{E}_i$ is removable if (3a) the path $\widetilde{\text{MST}}_i[\varphi_{C_u}, \varphi_{C_v}]$ between φ_{C_u} and φ_{C_v} only contains nodes in $\widetilde{\text{MST}}_i$ of degree at most 2 and (3b) $\omega(\widetilde{\text{MST}}_i[\varphi_{C_u}, \varphi_{C_v}]) \leq t(1 + 6\epsilon)\omega(\varphi_{C_u}, \varphi_{C_v})$.

As we will show in the sequel, if an edge $(\varphi_{C_u}, \varphi_{C_v})$ satisfies property (3b), there is a path of stretch at most $t(1 + 6\epsilon)$ in $H_{<L_{i-1}}$ between u and v and hence, we do not need to consider edge (u, v) in the construction of H_i . To meet the required lightness bound, it turns out that it suffices to remove edges satisfying both properties (3a) and (3b), rather than removing all edges satisfying property (3b).

7.1 Constructing Level- $(i + 1)$ Clusters

To obtain a fast spanner construction, we will maintain for each cluster $C \in \mathcal{C}_i$ a *representative* vertex $r(C) \in C$. If C contains at least one original vertex, then $r(C)$ is one original vertex in C ; otherwise, $r(C)$ is a virtual vertex. (Recall that virtual vertices are those subdividing MST edges.) For each vertex $v \in C$, we designate $r(C)$ as the *representative* of v , i.e., we set $r(v) = r(C)$ for each $v \in C$. We use the UNION-FIND data structure to maintain these representatives. Specifically, the representative of v will be given as $\text{FIND}(v)$. Whenever a level- $(i + 1)$ cluster is formed from level- i clusters, we call UNION (sequentially on the level- i clusters) to construct a new representative for the new cluster.

A careful usage of the Union-Find data structure. We will use the UNION-FIND data structure [58] for grouping subsets of clusters to larger clusters (via the UNION operation) and checking whether two given vertices belong to the same cluster (via the FIND operation). To reduce the amortized time to $O(\alpha(m, n))$, we only store original vertices in the UNION-FIND data structure. To this end, for each virtual vertex, say x , which subdivides an edge $(u, v) \in \text{MST}$, we store a pointer, denoted by $p(x)$, which points to one of the endpoints, say u , in the same cluster with x , *if there is at least one endpoint in the same cluster with x* . In particular, any virtual vertex has at most two optional clusters that it can belong to at each level of the hierarchy. Hence, we can apply every UNION-FIND operation to $p(x)$ instead of x . For example, to check whether two virtual vertices x and y are in the same cluster, we compare $r(p(x)) \stackrel{?}{=} r(p(y))$ via two FIND operations. The total number of UNION and FIND operations in our construction remains $O(m)$ while the number of vertices that we store in the data structure is reduced to n . Thus, the amortized time of each operation reduces to $O(\alpha(m, n))$ and the total runtime due to all these operations is $O(m\alpha(m, n))$.

Following the approach in Section 6.2, we construct a graph $\mathcal{G}_i$ satisfying all properties in Definition 6.15 and Definition 7.1. Then we construct a set $\mathbb{X}$ of subgraphs of $\mathcal{G}_i$ satisfying the three properties (P1')-(P3') and a subgraph H_i of G (and of $\tilde{G}$ as well). Each subgraph $\mathcal{X} \in \mathbb{X}$ is then converted to a level- $(i + 1)$ cluster by Equation (26).

Constructing $\mathcal{G}_i$. We shall assume inductively on $i, i \geq 1$ that:

- The set of edges $\widetilde{\text{MST}}_i$ is given by the construction of the previous level i in the hierarchy; for the base case $i = 1$ (see Section 6.2), $\widetilde{\text{MST}}_1$ is simply a set of edges of $\widetilde{\text{MST}}$ that are not in any level-1 cluster.
- The weight $\omega(\varphi_C)$ on each node $\varphi_C \in \mathcal{V}_i$ is the potential value of cluster $C \in \mathcal{C}_i$; for the base case $i = 1$, the potential values of level-1 clusters were computed in $O(m)$ time by Section 6.2.

By the end of this section, we will have constructed the edge set $\widetilde{\text{MST}}_{i+1}$ and the weight function on nodes of $\mathcal{G}_{i+1}$, in time $O(|\mathcal{V}_i|\alpha(m, n))$. Computing the weight function on nodes of $\mathcal{G}_{i+1}$ is equivalent to computing the augmented diameter of $\mathcal{X}$, which in turn, is related to the potential function. The fact that we can compute all the weights efficiently in almost linear time is the crux of our framework.

Note that we make no inductive assumption regarding the set of edges E_i^σ , which can be computed once in $O(m)$ overall time at the outset for all levels $i \geq 1$, since the edge sets $E_1^\sigma, E_2^\sigma, \dots$ are pairwise disjoint and the number of levels is $O(m)$ by Claim 6.5.

Lemma 7.2. $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i \cup \widetilde{\text{MST}}_i, \omega)$ can be constructed in $O(\alpha(m, n)(|\mathcal{V}_i| + |E_i^\sigma|))$ time, where $\alpha(\cdot, \cdot)$ is the inverse-Ackermann function.

Proof: Note that $\widetilde{\text{MST}}_i$ and E_i^σ are given at the outset of the construction of $\mathcal{G}_i$. To construct the edge set $\mathcal{E}_i$, we do the following. For each edge $e = (u, v) \in E_i^\sigma$, we compute the representatives $r(u), r(v)$;

this can be done in $O(\alpha(m, n))$ amortized time over all the levels up to i using the UNION-FIND data structure. Equipped with the representatives, it takes $O(1)$ time to check whether e 's endpoints lie in the same level- i cluster and check in $O(1)$ time whether edges $e = (u, v)$ and $e' = (u', v')$ are parallel in the cluster graph. Next, we remove all removable edges from $\mathcal{G}_i$ as specified by property (3b) in Definition 7.1. First we find in $O(|\mathcal{V}_i|)$ time a collection $\mathbb{P}$ of *maximal paths* in $\widetilde{\text{MST}}_i$ that only contain degree-2 vertices. We then find for each path $\mathcal{P} \in \mathbb{P}$ a subset of edges $\mathcal{E}_{\mathcal{P}} \subseteq \mathcal{E}_i$ whose both endpoints belong to $\mathcal{P}$. Finally, for each path $\mathcal{P} \in \mathbb{P}$ and each edge $(\varphi_{C_u}, \varphi_{C_v}) \in \mathcal{E}_{\mathcal{P}}$, we can compute $\omega(\mathcal{P}[\varphi_{C_u}, \varphi_{C_v}])$ in $O(1)$ time, after an $O(|\mathcal{V}(\mathcal{P})|)$ preprocessing time by fixing an endpoint $\varphi_C \in \mathcal{P}$ and for every node $\varphi_{C'} \in \mathcal{P}$, we compute $\omega(\mathcal{P}[\varphi_C, \varphi_{C'}])$ in total $O(|\mathcal{V}(\mathcal{P})|)$ time. Given $\omega(\mathcal{P}[\varphi_{C_u}, \varphi_{C_v}])$, we can check in $O(1)$ time whether $(\varphi_{C_u}, \varphi_{C_v})$ is removable and if so, we remove it from $\mathcal{E}_i$. The total running time is $O(|\mathcal{V}_i| + |E_i^{\sigma}|)$. $\square$

The following key lemma states all the properties of clusters constructed in our framework; the details of the construction are deferred to Section 8.

Lemma 7.3. *Given $\mathcal{G}_i$, we can construct in time $O((|\mathcal{V}_i| + |\mathcal{E}_i|)\epsilon^{-1})$ (i) a partition of $\mathcal{V}_i$ into three sets $\{\mathcal{V}_i^{\text{high}}, \mathcal{V}_i^{\text{low}^+}, \mathcal{V}_i^{\text{low}^-}\}$ and (ii) a collection $\mathbb{X}$ of subgraphs of $\mathcal{G}_i$ and their augmented diameters, such that:*

- (1) *For every node $\varphi_C \in \mathcal{V}_i$: If $\varphi_C \in \mathcal{V}_i^{\text{high}}$, then φ_C is incident to $\Omega(1/\epsilon)$ edges in $\mathcal{E}_i$; otherwise ($\varphi_C \in \mathcal{V}_i^{\text{low}^+} \cup \mathcal{V}_i^{\text{low}^-}$), the number of edges in $\mathcal{E}_i$ incident to φ_C is $O(1/\epsilon)$.*
- (2) *If a subgraph $\mathcal{X}$ contains at least one node in $\mathcal{V}_i^{\text{low}^-}$, then every node of $\mathcal{X}$ is in $\mathcal{V}_i^{\text{low}^-}$. Let $\mathbb{X}^{\text{low}^-} \subseteq \mathbb{X}$ be a set of subgraphs whose nodes are in $\mathcal{V}_i^{\text{low}^-}$ only.*
- (3) *Let $\Delta_{i+1}^+(\mathcal{X}) = \Delta(\mathcal{X}) + \sum_{\mathbf{e} \in \widetilde{\text{MST}}_i \cap \mathcal{E}(\mathcal{X})} w(\mathbf{e})$. Then, $\Delta_{i+1}^+(\mathcal{X}) \geq 0$ for every $\mathcal{X} \in \mathbb{X}$, and*

$$\sum_{\mathcal{X} \in \mathbb{X} \setminus \mathbb{X}^{\text{low}^-}} \Delta_{i+1}^+(\mathcal{X}) = \sum_{\mathcal{X} \in \mathbb{X} \setminus \mathbb{X}^{\text{low}^-}} \Omega(|\mathcal{V}(\mathcal{X})|\epsilon^2 L_i). \quad (29)$$

- (4) *There is no edge in $\mathcal{E}_i$ between a node in $\mathcal{V}_i^{\text{high}}$ and a node in $\mathcal{V}_i^{\text{low}^-}$. Furthermore, if there exists an edge $(\varphi_{C_u}, \varphi_{C_v}) \in \mathcal{E}_i$ such that both φ_{C_u} and φ_{C_v} are in $\mathcal{V}_i^{\text{low}^-}$, then $\mathcal{V}_i^{\text{low}^-} = \mathcal{V}_i$ and $|\mathcal{E}_i| = O(\frac{1}{\epsilon^2})$; we call this case the degenerate case.*
- (5) *For every subgraph $\mathcal{X} \in \mathbb{X}$, $\mathcal{X}$ satisfies the three properties (P1')-(P3') with constant $g = 31$ and $\epsilon \leq \frac{1}{8(g+1)}$, and $|\mathcal{E}(\mathcal{X}) \cap \mathcal{E}_i| = O(|\mathcal{V}(\mathcal{X})|)$.*

Furthermore, the construction of $\mathbb{X}$ can be constructed in the pointer-machine model with the same running time.

We observe the following observations about subgraphs of $\mathbb{X}$ in Lemma 7.3.

Observation 7.4. *If a subgraph $\mathcal{X} \in \mathbb{X}$ has $\mathcal{V}(\mathcal{X}) \cap (\mathcal{V}_i^{\text{high}} \cup \mathcal{V}_i^{\text{low}^+}) \neq \emptyset$, then $\mathcal{V}(\mathcal{X}) \subseteq (\mathcal{V}_i^{\text{high}} \cup \mathcal{V}_i^{\text{low}^+})$.*

Proof: Follows from Item (2) in Lemma 7.3 and the fact that $\{\mathcal{V}_i^{\text{high}}, \mathcal{V}_i^{\text{low}^+}, \mathcal{V}_i^{\text{low}^-}\}$ is a partition of $\mathcal{V}_i$. $\square$

Observation 7.5. *Unless the degenerate case happens, for every edge $(\varphi_{C_u}, \varphi_{C_v})$ with one endpoint in $\mathcal{V}_i^{\text{low}^-}$, w.l.o.g. φ_{C_v} , the other endpoint φ_{C_u} must be in $\mathcal{V}_i^{\text{low}^+}$. As a result, $\mathcal{E}(\mathcal{X}) \cap \mathcal{E}_i = \emptyset$ if $\mathcal{X} \in \mathbb{X}^{\text{low}^-}$.*

Proof: If the degenerate case does not happen, by Item (4) in Lemma 7.3, any edge incident to a node in $\mathcal{V}_i^{\text{low}^-}$ must be incident to a node in $\mathcal{V}_i^{\text{low}^+}$. By Item (2), if $\mathcal{X} \in \mathbb{X}^{\text{low}^-}$, then $\mathcal{V}(\mathcal{X}) \subseteq \mathcal{V}_i^{\text{low}^-}$ and hence, there is no edge between two nodes in $\mathcal{X}$. Thus, $\mathcal{E}(\mathcal{X}) \cap \mathcal{E}_i = \emptyset$. $\square$

Next, we show how to construct $\widetilde{\text{MST}}_{i+1}$ for the construction of the next level.

Lemma 7.6. *Given the collection of subgraphs $\mathbb{X}$ of $\mathcal{G}_i$ and their augmented diameters, we can construct the set of nodes $\mathcal{V}_{i+1}$, and their weights, and the cluster tree $\widetilde{\text{MST}}_{i+1}$ of $\mathcal{G}_{i+1}$ in $O(|\mathcal{V}_i|\alpha(m, n))$ time.*

Proof: For each subgraph $\mathcal{X} \in \mathbb{X}$, we call UNION operations sequentially on the set of clusters corresponding to the nodes of $\mathcal{X}$ to create a level- $(i+1)$ cluster $C_{\mathcal{X}} \in \mathcal{C}_{i+1}$. Then we create a set of nodes $\mathcal{V}_{i+1}$ for $\mathcal{G}_{i+1}$: each node $\varphi_{C_{\mathcal{X}}}$ corresponds to a cluster $C_{\mathcal{X}} \in \mathcal{C}_{i+1}$ (and also subgraph $\mathcal{X} \in \mathbb{X}$). Next, we set the weight $\omega(\varphi_{C_{\mathcal{X}}}) = \text{Adm}(\mathcal{X})$. The total running time of this step is $O(|\mathcal{V}_i|\alpha(m, n))$.

We now construct $\widetilde{\text{MST}}_{i+1}$. Let $\widetilde{\text{MST}}_i^{\text{out}} = \widetilde{\text{MST}}_i \setminus (\cup_{\mathcal{X} \in \mathbb{X}} (\mathcal{E}(\mathcal{X}) \cap \widetilde{\text{MST}}_i))$ be the set of $\widetilde{\text{MST}}_i$ edges that are not contained in any subgraph $\mathcal{X} \in \mathbb{X}$. Let $\widetilde{\text{MST}}_{i+1}'$ be the graph with vertex set $\mathcal{V}_{i+1}$ and there is an edge between two nodes $(\mathcal{X}, \mathcal{Y})$ in $\mathcal{V}_{i+1}$ if there is at least one edge in $\widetilde{\text{MST}}_i^{\text{out}}$ between two nodes in the two corresponding subgraphs $\mathcal{X}$ and $\mathcal{Y}$. Since $\widetilde{\text{MST}}_i$ is a spanning tree of $\mathcal{G}_i$, $\widetilde{\text{MST}}_{i+1}'$ must be connected. $\widetilde{\text{MST}}_{i+1}$ is then a spanning tree of $\widetilde{\text{MST}}_{i+1}'$. $\square$

7.2 Constructing H_i : Proof of Theorem 1.10(1)

Recall that to obtain a fast algorithm for constructing a light spanner, Lemma 6.8 requires a fast construction of clusters at every level and a fast construction of H_i , the spanner for level- i edges E_i^σ . In Section 7.1, we have designed an efficient construction of level- i clusters (Lemma 7.6). In this section, we show to construct H_i efficiently with stretch $t(1 + \max\{s_{\text{SSA}}(2g) + 4g, 10g\}\epsilon)$; that is parameter ρ in Lemma 6.8 is $\rho = \max\{s_{\text{SSA}}(2g) + 4g, 10g\}$. By induction, we assume that the stretch of every edge of weight less than $L_i/(1 + \psi)$ in $H_{<L_i/(1+\psi)}$ is $t(1 + \max\{s_{\text{SSA}}(2g) + 4g, 10g\}\epsilon)$. Note that $H_{<L_i} = H_{<L_i/(1+\psi)} \cup H_i$; see Remark 6.9.

Our construction of H_i assumes the existence of SSA. Since edges of the input graph to SSA must have weights in $[L, (1 + \epsilon)L)$ for some parameter L , we set parameter ψ in Lemma 6.8 to be ϵ . Thus, level- i edges E_i^σ (and hence edges in $\mathcal{E}_i$ of $\mathcal{G}_i$) have weights in $[L_i/(1 + \epsilon), L_i)$.

We now go into the details of the construction of H_i . We assume that we are given the collection $\mathbb{X}$ of subgraphs as described in Lemma 7.3. Define:

$$\begin{aligned} \mathbb{X}^{\text{high}} &= \{\mathcal{X} \in \mathbb{X} : \mathcal{V}(\mathcal{X}) \cap \mathcal{V}_i^{\text{high}} \neq \emptyset\} \\ \mathbb{X}^{\text{low}^+} &= \{\mathcal{X} \in \mathbb{X} : \mathcal{V}(\mathcal{X}) \cap \mathcal{V}_i^{\text{low}^+} \neq \emptyset\} \end{aligned} \tag{30}$$

It could be that $\mathbb{X}^{\text{high}} \cap \mathbb{X}^{\text{low}^+} \neq \emptyset$. By Observation 7.4, $\{\mathbb{X}^{\text{high}} \cup \mathbb{X}^{\text{low}^+}, \mathbb{X}^{\text{low}^-}\}$ is a partition of $\mathbb{X}$.

Recall that each edge $(\varphi_{C_u}, \varphi_{C_v}) \in \mathcal{E}_i$ has a corresponding edge $(u, v) \in E_i^\sigma$ where u and v are in two level- i clusters C_u and C_v , respectively. Our goal in this section is to prove the following lemma.

Lemma 7.7. *Given SSA, we can construct H_i in total time $O((|\mathcal{V}_i| + |\mathcal{E}_i|)\tau(m, n))$ satisfying Lemma 6.8 with $\lambda = O(\chi\epsilon^{-2} + \epsilon^{-3})$, and $A = O(\chi\epsilon^{-2} + \epsilon^{-3})$, when $\epsilon \leq 1/(2g)$. Furthermore, the stretch of every edge in E_i^σ in $H_{<L_i}$ is $t(1 + \max\{s_{\text{SSA}}(2g) + 4g, 10g\}\epsilon)$.*

We apply SSA to $\mathcal{V}_i^{\text{high}}$ that has size at most n since every level- i cluster corresponding to a node in $\mathcal{V}_i^{\text{high}}$ contains at least one original vertex in G . Furthermore, $|\mathcal{E}_i^{\text{high}}|$ is bounded by m and hence, $\tau(|\mathcal{E}_i^{\text{high}}|, |\mathcal{V}_i^{\text{high}}|) \leq \tau(m, n)$.

Remark 7.8. *If SSA can be implemented in the ACT model in time $O((|\mathcal{V}_i^{\text{high}}| + |\mathcal{E}_i^{\text{high}}|)\tau(m, n))$, then the construction of H_i can be implemented in the ACT model in time $O((|\mathcal{V}_i| + |\mathcal{E}_i|)\tau(m, n))$.*

Constructing H_i . We construct H_i in three steps, as briefly described in the construction overview above. Initially H_i contains no edges.

- **(Step 1).** For every sugraph $\mathcal{X} \in \mathbb{X}$ and every edge $\mathbf{e} = (\varphi_{C_u}, \varphi_{C_v}) \in \mathcal{E}(\mathcal{X})$ such that $\mathbf{e} \in \mathcal{E}_i$, we add the corresponding edge (u, v) to H_i . (Note that if $\mathbf{e} \notin \mathcal{E}_i$, it is in $\widehat{\text{MST}}_i$ and hence (u, v) belongs to H_0).
- **(Step 2).** For each node $\varphi_{C_u} \in \mathcal{V}_i^{\text{low}^+} \cup \mathcal{V}_i^{\text{low}^-}$, and for each edge $(\varphi_{C_u}, \varphi_{C_v})$ in $\mathcal{E}_i$ incident to φ_{C_u} , we add the corresponding edge (u, v) to H_i ,
- **(Step 3).** Let $\mathcal{E}_i^{\text{high}} \subseteq \mathcal{E}_i$ be the set of edges whose both endpoints are in $\mathcal{V}_i^{\text{high}}$, and $\mathcal{K}_i = (\mathcal{V}_i^{\text{high}}, \mathcal{E}_i^{\text{high}}, \omega)$ be a subgraph of $\mathcal{G}_i$. We run SSA on $\mathcal{K}_i$ to obtain $\mathcal{E}_i^{\text{pruned}}$. For every edge $(\varphi_{C_u}, \varphi_{C_v}) \in \mathcal{E}_i^{\text{pruned}}$, we add the corresponding edge (u, v) to H_i .

Analysis. In Claim 7.9, Claim 7.11, and Claim 7.12 below, we bound the running time to construct H_i , the stretch of edges in E_i^σ , and the weight of H_i , respectively. The following claims follows directly from the construction.

Claim 7.9. H_i can be constructed in time $O((|\mathcal{V}_i| + |\mathcal{E}_i|)\tau(m, n))$.

We bound the stretch of edges in E_i^σ . We first show that the input to SSA satisfies its requirement.

Claim 7.10. $\mathcal{K}_i = (\mathcal{V}_i^{\text{high}}, \mathcal{E}_i^{\text{high}}, \omega)$ is a $(L, \epsilon, \beta, \Upsilon = 1 + \epsilon)$ -cluster graph with $L = L_i/(1 + \epsilon)$, $\beta = 2g$, and $H_{<L} = H_{<L_i/(1+\epsilon)}$, where $H_{<L_i/(1+\epsilon)}$ is the spanner constructed for edges of weight less than $L_i/(1 + \epsilon)$ (see Remark 6.9 with $\psi = \epsilon$). Furthermore, the stretch of $H_{<L}$ for edges of weight less than L is $t(1 + \max\{s_{\text{SSA}}(2g) + 4g, 10g\}\epsilon)$.

Proof: We verify all properties in Definition 1.9. Properties (1) and (2) follow directly from the definition of $\mathcal{K}_i$. Since we set $\psi = \epsilon$, every edge $(u, v) \in E_i^\sigma$ has $L_i/(1 + \epsilon) \leq w(u, v) \leq L_i$. As $L = L_i/(1 + \epsilon)$, property (3) follows. By property (P3), $\text{Dm}(H_{<L_i/(1+\epsilon)}[C]) \leq gL_{i-1} = g(1 + \epsilon)\epsilon L \leq 2g\epsilon L = \beta\epsilon L$ when $\epsilon < 1$. Thus, $\mathcal{K}_i$ is a (L, ϵ, β) -cluster graph. By induction, the stretch of $H_{<L}$ is $t(1 + \max\{s_{\text{SSA}}(2g) + 4g, 10g\}\epsilon)$. $\square$

Claim 7.11. $\forall (u, v) \in E_i^\sigma$, $d_{H_{<L_i}}(u, v) \leq t(1 + \max\{s_{\text{SSA}}(2g) + 4g, 10g\}\epsilon)w(u, v)$ when $\epsilon \leq 1/(2g)$.

Proof: Let $F_i^\sigma = \{(u, v) \in E_i^\sigma : \exists (\varphi_{C_u}, \varphi_{C_v}) \in \mathcal{E}_i\}$ be the set of edges in E_i^σ that correspond to the edges in $\mathcal{E}_i$. We first show that:

$$d_{H_{<L_i}}(u, v) \leq t(1 + s_{\text{SSA}}(2g)\epsilon)w(u, v) \quad \forall (u, v) \in F_i^\sigma. \quad (31)$$

To that end, let $(\varphi_{C_u}, \varphi_{C_v}) \in \mathcal{E}_i$ be the edge corresponding to (u, v) where $(u, v) \in F_i^\sigma$. If at least one of the endpoints of $(\varphi_{C_u}, \varphi_{C_v})$ is in $\mathcal{V}_i^{\text{low}^+} \cup \mathcal{V}_i^{\text{low}^-}$, then $(u, v) \in H_i$ by the construction in Step 2, hence Equation (31) holds. Otherwise, $\{\varphi_{C_u}, \varphi_{C_v}\} \subseteq \mathcal{V}_i^{\text{high}}$, which implies that $(\varphi_{C_u}, \varphi_{C_v}) \in \mathcal{E}_i^{\text{high}}$. Since we add all edges of $\mathcal{E}_i^{\text{pruned}}$ to H_i , by property (2) of SSA and Claim 7.10, the stretch of (u, v) is $t(1 + s_{\text{SSA}}(2g)\epsilon)$.

It remains to bound the stretch of any edge $(u', v') \in E_i^\sigma \setminus F_i^\sigma$. Recall that (u', v') is not added to $\mathcal{E}_i$ because (a) both u' and v' are in the same level- i cluster in the construction of the cluster graph in Lemma 7.2, or (b) (u', v') is parallel with another edge (u, v) also in Lemma 7.2, or (c) the edge $(\varphi_{C_{u'}}, \varphi_{C_{v'}})$ corresponding to (u', v') is a removable edge (see Definition 6.15).

In case (a), since the level- i cluster containing both u' and v' has diameter at most gL_{i-1} by property (P3), we have a path from u' to v' in $H_{<L_{i-1}}$ of diameter at most $gL_{i-1} = g\epsilon L_i \leq \frac{L_i}{1+\psi} \leq w(u', v')$

when $\epsilon \leq 1/(2g)$. Thus, the stretch of edge (u', v') is 1. For case (c), the stretch of (u', v') in $H_{<L_{i-1}}$ is $t(1 + 6g\epsilon)$ since $\epsilon \leq 1$. Thus, in both cases, we have:

$$d_{H_{<L_i}}(u', v') \leq t(1 + 6g\epsilon)w(u', v') \quad (32)$$

We now consider case (b). Let C_u and C_v be two level- i clusters containing u and v , respectively. W.l.o.g, we assume that $u' \in C_u$ and $v' \in C_v$. Since we only keep the edge of minimum weight among all parallel edges, $w(u, v) \leq w(u', v')$. Since the level- i clusters that contain u and v have diameters at most $gL_{i-1} = g\epsilon L_i$ by property (P3), it follows that $\text{Dm}(H_{<L_i}[C_u]), \text{Dm}(H_{<L_i}[C_v]) \leq g\epsilon L_i$. We have:

$$\begin{aligned} d_{H_{<L_i}}(u', v') &\leq d_{H_{<L_i}}(u, v) + \text{Dm}(H_{<L_i}[C_u]) + \text{Dm}(H_{<L_i}[C_v]) \\ &\leq t(1 + \max\{s_{\text{SSA}}(2g), 6g\}\epsilon)w(u, v) + 2g\epsilon L_i \quad (\text{by Equation (31) and Equation (32)}) \\ &\leq t(1 + \max\{s_{\text{SSA}}(2g), 6g\}\epsilon)w(u', v') + 2g\epsilon L_i \leq t(1 + \max\{s_{\text{SSA}}(2g) + 4g, 10g\}\epsilon)w(u', v') \end{aligned}$$

Since $w(u', v') \geq L_i/(1 + \epsilon) \geq L_i/2$ and $t \geq 1$. The lemma now follows. $\square$

Claim 7.12. Let $\widetilde{\text{MST}}_i^{\text{in}} = \cup_{\mathcal{X} \in \mathbb{X}} (\mathcal{E}(\mathcal{X}) \cap \widetilde{\text{MST}}_i)$ be the set of $\widetilde{\text{MST}}_i$ edges that are contained in subgraphs in $\mathbb{X}$. Then, $w(H_i) \leq \lambda \Delta_{i+1} + a_i$ for $\lambda = O(\chi\epsilon^{-2} + \epsilon^{-3})$ and $a_i = (\chi\epsilon^{-2}) \cdot w(\widetilde{\text{MST}}_i^{\text{in}}) + O(L_i/\epsilon^2)$.

Proof: Let $\widetilde{\text{MST}}_i^{\text{in}}(\mathcal{X}) = \mathcal{E}(\mathcal{X}) \cap \widetilde{\text{MST}}_i$ for each subgraph $\mathcal{X} \in \mathbb{X}$. By the definition of $\mathbb{X}^{\text{low}^+}$ and $\mathbb{X}^{\text{high}}$ (see Equation (30)), it holds that:

$$|\mathcal{V}_i^{\text{high}}| \leq \sum_{\mathcal{X} \in \mathbb{X}^{\text{high}}} |\mathcal{V}(\mathcal{X})| \quad \text{and} \quad |\mathcal{V}_i^{\text{low}^+}| \leq \sum_{\mathcal{X} \in \mathbb{X}^{\text{low}^+}} |\mathcal{V}(\mathcal{X})| \quad (33)$$

First, we consider the non-degenerate case where $\mathcal{V}_i^{\text{low}^-} \neq \mathcal{V}_i$. By Observation 7.5, any edge in $\mathcal{E}_i$ incident to a node in $\mathcal{V}_i^{\text{low}^-}$ is also incident to a node in $\mathcal{V}_i^{\text{low}^+}$. We bound the total weight of the edges added to H_i by considering each step in the construction of H_i separately. Let $F_i^{(a)} \subseteq E_i^\sigma$ be the set of edges added to H_i in the construction in Step a , $a \in \{1, 2, 3\}$.

By Observation 7.5, $\mathcal{E}(\mathcal{X}) \cap \mathcal{E}_i = \emptyset$ if $\mathcal{X} \in \mathbb{X}^{\text{low}^-}$. Recall that $\mathbb{X}^{\text{high}} \cup \mathbb{X}^{\text{low}^+} = \mathbb{X} \setminus \mathbb{X}^{\text{low}^-}$. By Item (5) in Lemma 7.3, the total weight of the edges added to H_i in Step 1 is:

$$\begin{aligned} w(F_i^{(1)}) &= \sum_{\mathcal{X} \in \mathbb{X}^{\text{high}} \cup \mathbb{X}^{\text{low}^+}} O(|\mathcal{V}(\mathcal{X})|)L_i \stackrel{\text{Eq. (29)}}{=} O\left(\frac{1}{\epsilon^2}\right) \sum_{\mathcal{X} \in \mathbb{X}^{\text{high}} \cup \mathbb{X}^{\text{low}^+}} \Delta_{i+1}^+(\mathcal{X}) = O\left(\frac{1}{\epsilon^2}\right) \sum_{\mathcal{X} \in \mathbb{X}} \Delta_{i+1}^+(\mathcal{X}) \\ &= O\left(\frac{1}{\epsilon^2}\right) \sum_{\mathcal{X} \in \mathbb{X}} \left(\Delta_{i+1}(\mathcal{X}) + w(\widetilde{\text{MST}}_i^{\text{in}}(\mathcal{X})) \right) = O\left(\frac{1}{\epsilon^2}\right) (\Delta_{i+1} + w(\widetilde{\text{MST}}_i^{\text{in}})) \quad (\text{by Claim 6.18}) . \end{aligned} \quad (34)$$

Next, we bound $w(F_i^{(2)})$. Let (u, v) be an edge added to H_i in Step 2 and let $(\varphi_{C_u}, \varphi_{C_v})$ be the corresponding edge of (u, v) . Since $\mathcal{V}_i^{\text{low}^-} \neq \mathcal{V}_i$, at least one of the endpoints of $(\varphi_{C_u}, \varphi_{C_v})$, w.l.o.g. φ_{C_u} , is in $\mathcal{V}_i^{\text{low}^+}$ by Observation 7.5. Recall by Item (1) of Lemma 7.3 that all nodes in $\mathcal{V}_i^{\text{low}^+}$ have low degree, i.e., incident to $O(1/\epsilon)$ edges in $\mathcal{E}_i$. Thus, $|F_i^{(2)}| = O(\frac{1}{\epsilon})|\mathcal{V}_i^{\text{low}^+}|$. We have:

$$\begin{aligned} w(F_i^{(2)}) &= O\left(\frac{1}{\epsilon}\right)|\mathcal{V}_i^{\text{low}^+}|L_i \stackrel{\text{Eq. (33)}}{=} O\left(\frac{1}{\epsilon}\right) \sum_{\mathcal{X} \in \mathbb{X}^{\text{low}^+}} |\mathcal{V}(\mathcal{X})|L_i = O\left(\frac{1}{\epsilon}\right) \sum_{\mathcal{X} \in \mathbb{X}^{\text{high}} \cup \mathbb{X}^{\text{low}^+}} |\mathcal{V}(\mathcal{X})|L_i \\ &\stackrel{\text{Eq. (29)}}{=} O\left(\frac{1}{\epsilon^3}\right) \sum_{\mathcal{X} \in \mathbb{X}^{\text{high}} \cup \mathbb{X}^{\text{low}^+}} \Delta_{i+1}^+(\mathcal{X}) = O\left(\frac{1}{\epsilon^3}\right) (\Delta_{i+1} + w(\widetilde{\text{MST}}_i^{\text{in}})) . \end{aligned} \quad (35)$$

By property (1) of SSA, the number of edges added to H_i in Step 3 is at most $\chi|\mathcal{V}_i^{\text{high}}|$. Thus:

$$\begin{aligned} w(F_i^{(3)}) &\leq \chi|\mathcal{V}_i^{\text{high}}|L_i \stackrel{\text{Eq. (33)}}{\leq} \chi \sum_{\mathcal{X} \in \mathbb{X}^{\text{high}}} |\mathcal{V}(\mathcal{X})|L_i \leq \chi \sum_{\mathcal{X} \in \mathbb{X}^{\text{high}} \cup \mathbb{X}^{\text{low}^+}} |\mathcal{V}(\mathcal{X})|L_i \\ &\stackrel{\text{Eq. (29)}}{=} O(\chi\epsilon^{-2}) \sum_{\mathcal{X} \in \mathbb{X}^{\text{high}} \cup \mathbb{X}^{\text{low}^+}} \Delta_{i+1}^+(\mathcal{X}) = O(\chi\epsilon^{-2})(\Delta_{i+1} + w(\widetilde{\text{MST}}_i^{\text{in}})) . \end{aligned} \quad (36)$$

By Equations (34) to (36), we conclude that:

$$w(H_i) = O(\chi\epsilon^{-2} + \epsilon^{-3})(\Delta_{i+1} + w(\widetilde{\text{MST}}_i^{\text{in}})) \leq \lambda(\Delta_{i+1} + w(\widetilde{\text{MST}}_i^{\text{in}})) \quad (37)$$

for some $\lambda = O(\chi\epsilon^{-2} + \epsilon^{-3})$.

It remains to consider the degenerate case where $\mathcal{V}_i^{\text{low}^-} = \mathcal{V}_i$. Even if we add every single edge that corresponds to an edge in $\mathcal{E}_i$ to H_i , Item (3) in Lemma 7.3 implies that the number of such edges is at most $O(\frac{1}{\epsilon^2})$. Thus, we have:

$$w(H_i) = O(\frac{L_i}{\epsilon^2}) \leq \lambda \cdot (\Delta_{i+1} + w(\widetilde{\text{MST}}_i^{\text{in}})) + O(\frac{L_i}{\epsilon^2}) \quad (38)$$

where in the last equation, we use the fact that:

$$\Delta_{i+1} + w(\widetilde{\text{MST}}_i^{\text{in}}) \stackrel{\text{Claim 6.18}}{=} \sum_{\mathcal{X} \in \mathbb{X}} (\Delta_{i+1}(\mathcal{X}) + \widetilde{\text{MST}}_i^{\text{in}}(\mathcal{X})) = \sum_{\mathcal{X} \in \mathbb{X}} \Delta_{i+1}^+(\mathcal{X}) \geq 0$$

by Item (3) of Lemma 7.3. Thus, the claim follows from Equations (37) and (38). $\square$

Proof: [Proof of Lemma 7.7] The running time follows from Claim 7.9. By Claim 7.11, the stretch is $t(1 + \max\{s_{\text{SSA}}(2g) + 4g, 10g\}\epsilon)$. By Claim 7.12, we have $\sum_{i \in \mathbb{N}^+} a_i = \sum_{i \in \mathbb{N}^+} (\lambda \widetilde{\text{MST}}_i^{\text{in}} + O(L_i/\epsilon^2))$. Observe by the definition that the sets of corresponding edges of $\widetilde{\text{MST}}_i^{\text{in}}$ and $\widetilde{\text{MST}}_j^{\text{in}}$ are disjoint for any $i \neq j \geq 1$. Thus, $\sum_{i \in \mathbb{N}^+} \widetilde{\text{MST}}_i^{\text{in}} \leq w(\text{MST})$. Observe that:

$$\sum_{i \in \mathbb{N}^+} O(\frac{L_i}{\epsilon^2}) = O(\frac{1}{\epsilon^2}) \sum_{i=1}^{i_{\max}} \frac{L_{i_{\max}}}{\epsilon^{i_{\max}-i}} = O(\frac{L_{i_{\max}}}{\epsilon^2(1-\epsilon)}) = O(\frac{1}{\epsilon^2})w(\text{MST}) ;$$

here $i_{\max}$ is the maximum level. The last equation is due to that $\epsilon \leq 1/2$ and every edge has weight at most $w(\text{MST})$ (by the removal step in the construction of $\tilde{G}$). Thus, $A = \lambda + O(\epsilon^{-2}) = O(\chi\epsilon^{-2} + \epsilon^{-3}) + O(\epsilon^{-2}) = O(\chi\epsilon^{-2} + \epsilon^{-3})$ as claimed. $\square$

We are now ready to prove Item (1) of Theorem 1.10.

Proof of Item (1) of Theorem 1.10. By Lemma 7.3 and Lemma 7.6, level- $(i+1)$ clusters can be constructed in time $O((|\mathcal{V}_i| + |\mathcal{E}_i|)\epsilon^{-1} + |\mathcal{V}_i|\alpha(m, n)) = O((|\mathcal{C}_i| + |E_i^\sigma|)(\alpha(m, n) + \epsilon^{-1}))$ when $\epsilon \ll 1$. By Lemma 7.7, H_i can be constructed in time $O((|\mathcal{V}_i| + |\mathcal{E}_i|)\tau(m, n)) = O((|\mathcal{C}_i| + |E_i^\sigma|)\tau(m, n))$.

We can construct a minimum spanning tree in time $T_{\text{MST}} = O((n+m)\alpha(m, n))$ by using Chazelle's algorithm [16]. Thus, by Lemma 6.8, the construction time of the light spanner is

$$O(m\epsilon^{-1}(\tau(m, n) + \alpha(m, n) + \epsilon^{-1})\log(1/\epsilon) + T_{\text{MST}}) = O(m\epsilon^{-1}(\tau(m, n) + \alpha(m, n) + \epsilon^{-1})\log(1/\epsilon)) .$$

By Lemma 7.7 and Lemma 6.8, the lightness of the spanner is

$$O\left(\frac{\lambda + A + 1}{\epsilon} \log \frac{1}{\epsilon} + \frac{1}{\epsilon}\right) = O((\chi\epsilon^{-3} + \epsilon^{-4}) \log(1/\epsilon)).$$

Note that we set $\psi = \epsilon$ in this case. Since $g = 31$, by Lemma 7.7 and Lemma 6.8, the stretch of the spanner is

$$t(1 + \max\{s_{\text{SSA}}(2g) + 4g, 10g\}\epsilon) \leq t(1 + (s_{\text{SSA}}O(1)) + O(1))\epsilon).$$

This completes the proof of the theorem. $\square$

8 Clustering: Proof of Lemma 7.3

In this section, we construct the set of subgraph $\mathbb{X}$ of the cluster graph $\mathcal{G}_i = (\mathcal{V}_i, \widetilde{\text{MST}}_i \cup \mathcal{E}_i, \omega)$ as claimed in Lemma 7.3. See Table 1 for a summary of notation we introduced in Section 6. Our construction builds upon the construction of Borradaile, Le and Wulff-Nilsen (BLW) [10]. However, unlike their construction, which is inefficient, our main focus here is on having a *linear-time* construction. Using the augmented diameter, we could bound the size of subgraphs (specifically in the construction of Step 4) arising during the course of our algorithm, and compute the augmented diameters of clusters efficiently. We note that in Borradaile, Le and Wulff-Nilsen [10], the efficiency of the construction is not relevant since they use the cluster hierarchy to *analyze the greedy algorithm*, not in the construction of the spanner.

Our construction has five main steps (Steps 1-5). In Step 1, we group all vertices of $\mathcal{V}_i^{\text{high}}$ and their neighbors into subgraphs of $\mathbb{X}$; see Lemma 8.1. In Step 2, we deal with *branching nodes* of $\widetilde{\text{MST}}_i$; see Lemma 8.2. In Step 3, we augment existing subgraphs formed in Steps 1 and 2, to guarantee a special structure of the ungrouped nodes. In Step 4, we group subpaths of $\widetilde{\text{MST}}_i$ connected by an edge $\mathbf{e}$ in $\widetilde{\text{MST}}_i$ into clusters; see Lemma 8.4. Finally, in Step 5, we deal with the remaining nodes of $\mathcal{V}_i$.

Recall that g is a constant defined in property (P3) (by Lemma 7.3, $g = 31$), and that MST_i is a spanning tree of $\mathcal{G}_i$. We refer readers to Table 1 for a summary of the notation.

Lemma 8.1 (Step 1). *Let $\mathcal{V}_i^{\text{high}} = \{\varphi_C \in \mathcal{V} : \varphi_C \text{ is incident to at least } \frac{2g}{\epsilon} \text{ edges in } \mathcal{E}_i\}$. Let $\mathcal{V}_i^{\text{high}+}$ be obtained from $\mathcal{V}_i^{\text{high}}$ by adding all neighbors that are connected to nodes in $\mathcal{V}_i^{\text{high}}$ via edges in $\mathcal{E}_i$. We can construct in $O(|\mathcal{V}_i| + |\mathcal{E}_i|)$ time a collection of node-disjoint subgraphs $\mathbb{X}_1$ of $\mathcal{G}_i$ such that:*

- (1) *Each subgraph $\mathcal{X} \in \mathbb{X}_1$ is a tree.*
- (2) *$\cup_{\mathcal{X} \in \mathbb{X}_1} \mathcal{V}(\mathcal{X}) = \mathcal{V}_i^{\text{high}+}$.*
- (3) *$L_i \leq \text{Adm}(\mathcal{X}) \leq 13L_i$, assuming that $\epsilon \leq 1/g$.*
- (4) *$|\mathcal{V}(\mathcal{X})| \geq \frac{2g}{\epsilon}$.*

Proof: Let $\mathcal{J} = (\mathcal{V}_i, \mathcal{E}_i)$ be the subgraph of $\mathcal{G}_i$ with the same vertex set and with edge set $\mathcal{E}_i$. Let $\mathcal{N}_{\mathcal{J}}(\varphi)$ be the set of neighbors of a node φ in $\mathcal{J}$, and $\mathcal{N}_{\mathcal{J}}[\varphi] = \mathcal{N}_{\mathcal{J}}(\varphi) \cup \{\varphi\}$. We construct $\mathbb{X}_1$ in three steps; initially, $\mathbb{X}_1 = \emptyset$.

- (1) Let $\mathcal{I}$ be a *maximal* set of nodes in $\mathcal{V}_i^{\text{high}}$ such that for any two nodes $\varphi_1, \varphi_2 \in \mathcal{I}$, $\mathcal{N}_{\mathcal{J}}[\varphi_1] \cap \mathcal{N}_{\mathcal{J}}[\varphi_2] = \emptyset$. For each node $\varphi \in \mathcal{I}$, we form a subgraph $\mathcal{X}$ that consists of φ , its neighbors $\mathcal{N}_{\mathcal{J}}[\varphi]$, and all incident edges in $\mathcal{E}_i$ of φ . We then add $\mathcal{X}$ to $\mathbb{X}_1$.
- (2) We iterate over all nodes of $\mathcal{V}_i^{\text{high}} \setminus \mathcal{I}$ that are not grouped yet to any subgraph. For each such node $\varphi \in \mathcal{V}_i^{\text{high}} \setminus \mathcal{I}$, there must be a neighbor φ' that is already grouped to a subgraph, say $\mathcal{X} \in \mathbb{X}_1$; if there are multiple such neighbors, we pick one of them arbitrarily. We add φ and the edge (φ, φ') to $\mathcal{X}$. Observe that every node in $\mathcal{V}_i^{\text{high}}$ is grouped to some subgraph at the end of this step.

- (3) For each node φ in $\mathcal{V}_i^{\text{high}+}$ that has not grouped to a subgraph in steps (1) and (2), there must be at least one neighbor, say φ' , of φ that is grouped in step (1) or step (2) to a subgraph $\mathcal{X} \in \mathbb{X}_1$; if there are multiple such nodes, we pick one of them arbitrarily. We then add φ and the edge (φ, φ') to $\mathcal{X}$.

This completes the construction of $\mathbb{X}_1$. We now show that subgraphs in $\mathbb{X}_1$ have all desired properties.

Observe that Items (1) and (2) follow directly from the construction. For Item (4), we observe that every subgraph $\mathcal{X} \in \mathbb{X}_1$ is created in step (1) and hence, contains a node $\varphi \in \mathcal{V}_i^{\text{high}}$ and all of its neighbors (in $\mathcal{J}$) by the definition of $\mathcal{I}$. Thus, $|\mathcal{V}(\mathcal{X})| \geq 2g/\epsilon$ since φ has at least $2g/\epsilon$ neighbors. For Item (3), we observe that each subgraph $\mathcal{X} \in \mathbb{X}_1$ after step (3) has hop-diameter at least 2 and at most 6. Thus, $\text{Adm}(\mathcal{X}) \leq 7g\epsilon L_i + 6L_i \leq 13L_i$. Furthermore, since every edge $e \in \mathcal{E}_i$ has a weight of at least $L_i/(1+\psi) \geq L_i/2$ and $\mathcal{X}$ has at least two edges in $\mathcal{E}_i$, $\text{Adm}(\mathcal{X}) \geq 2(L_i/2) = L_i$. The construction time follows straightforwardly from the algorithm. $\square$

Given a tree T , we say that a node $x \in T$ is *T-branching* if it has degree at least 3 in T . For brevity, we shall omit the prefix T in “*T-branching*” whenever this does not lead to confusion. Given a forest F , we say that x is *F-branching* if it is *T-branching* for some tree $T \subseteq F$. The construction of Step 2 is described in the following lemma.

Lemma 8.2 (Step 2). *Let $\tilde{F}_i^{(2)}$ be the forest obtained from $\widetilde{\text{MST}}_i$ by removing every node in $\mathcal{V}_i^{\text{high}+}$ (defined in Lemma 8.1). We can construct in $O(|\mathcal{V}_i|)$ time a collection $\mathbb{X}_2$ of subtrees of $\tilde{F}_i^{(2)}$ such that for every $\mathcal{X} \in \mathbb{X}_2$:*

- (1) $\mathcal{X}$ is a tree and has an $\mathcal{X}$ -branching node.
- (2) $L_i \leq \text{Adm}(\mathcal{X}) \leq 2L_i$.
- (3) $|\mathcal{V}(\mathcal{X})| = \Omega(\frac{1}{\epsilon})$ when $\epsilon \leq 1/g$.
- (4) Let $\tilde{F}_i^{(3)}$ be obtained from $\tilde{F}_i^{(2)}$ by removing every node contained in subgraphs of $\mathbb{X}_2$. Then, for every tree $\tilde{T} \subseteq \tilde{F}_i^{(3)}$, either (4a) $\text{Adm}(\tilde{T}) \leq 6L_i$ or (4b) $\tilde{T}$ is a path.

Proof: We say that a tree $\tilde{T} \in \tilde{F}_i^{(2)}$ is *long* if $\text{Adm}(\tilde{T}) \geq 6L_i$ and *short* otherwise. We construct $\mathbb{X}_2$, initially empty, as follows:

- Pick a long tree $\tilde{T}$ of $\tilde{F}_i^{(2)}$ that has at least one $\tilde{T}$ -branching node, say φ . We traverse $\tilde{T}$ starting from φ and *truncate* the traversal at nodes whose augmented distance from φ is at least L_i , which will be the leaves of the subtree. (The exact implementation details are delayed until the end of this proof.) As a result, the augmented radius (with respect to the center φ) of the subtree induced by the visited (non-truncated) nodes is at least L_i and at most $L_i + \bar{w} + g\epsilon L_i$. We then form a subgraph, say $\mathcal{X}$, from the subtree induced by the visited nodes, add $\mathcal{X}$ to $\mathbb{X}_2$, remove every node of $\mathcal{X}$ from $\tilde{T}$, and repeat this step until it no longer applies.

We observe that Item (1) follows directly from the construction. Since the algorithm only stops when every long tree has no branching node, meaning that it is a path, Item (4) is satisfied. By construction, $\mathcal{X}$ is a tree of augmented radius at least L_i and at most $L_i + g\epsilon L_i + \bar{w}$, hence $L_i \leq \text{Adm}(\mathcal{X}) \leq 2(L_i + g\epsilon L_i + \bar{w}) \leq 6L_i$ since $\bar{w} < L_i$ and $\epsilon \leq 1/g$; this implies Item (2). Let $\mathcal{D}$ be an augmented diameter path of $\mathcal{X}$; $\text{Adm}(\mathcal{D}) \geq L_i$ by construction. Note that every edge has a weight of at most $\bar{w} \leq L_{i-1}$ and every node has a weight of in $[L_{i-1}, gL_{i-1}]$ by property (P3'). Thus, $\mathcal{D}$ has at least $\frac{\text{Adm}(\mathcal{D})}{2gL_{i-1}} \geq \frac{L_i}{2g\epsilon L_i} = \Omega(\frac{1}{\epsilon})$ nodes; this implies Item (3). The construction of $\mathbb{X}_2$ can be implemented efficiently in $O(|\mathcal{V}_i|)$ by simply maintaining a list $\mathcal{B}$ of branching nodes of $\tilde{F}_i^{(2)}$. $\square$

The goal of constructing a subgraph from a branching node φ is to guarantee that there must be at least one neighbor, say φ' , of φ that does not belong to the augmented diameter path of $\mathcal{X}$. Thus, we could show that the amount of corrected potential change $\Delta_{i+1}^+(\mathcal{X})$ is at least $\omega(\varphi') \geq L_{i-1} = \epsilon L_i$. This will ultimately help us show that the corrected potential change $\Delta_{i+1}^+(\mathcal{X})$ is $\Omega(\epsilon^2 |\mathcal{V}(\mathcal{X})| L_i)$.

Step 3: Augmenting $\mathbb{X}_1 \cup \mathbb{X}_2$. Let $\tilde{F}_i^{(3)}$ be the forest obtained in Item (4b) in Lemma 8.2. Let $\mathcal{A}$ be the set of all nodes φ in $\tilde{F}_i^{(3)}$ such that φ is in a tree $\tilde{T} \in \tilde{F}_i^{(3)}$ of augmented diameter at least $6L_i$ and φ is a branching node in $\widetilde{\text{MST}}_i$. For each node $\varphi \in \mathcal{A}$ such that φ is connected to a node, say φ' , in a subgraph $\mathcal{X} \in \mathbb{X}_1 \cup \mathbb{X}_2$ via an $\widetilde{\text{MST}}_i$ edge $\mathbf{e}$, we add φ and $\mathbf{e}$ to $\mathcal{X}$. We note that φ' exists since φ has degree at least 3 in MST_i . (If there are many such nodes φ' , we choose an arbitrary one.)

The following lemma follows directly from the construction.

Lemma 8.3. *The augmentation in Step 3 can be implemented in $O(|\mathcal{V}_i|)$ time, and increases the augmented diameter of each subgraph in $\mathbb{X}_1 \cup \mathbb{X}_2$ by at most $4L_i$ when $\epsilon \leq 1/g$.*

Furthermore, let $\tilde{F}_i^{(4)}$ be the forest obtained from $\tilde{F}_i^{(3)}$ by removing every node in $\mathcal{A}$. Then, for every tree $\tilde{T} \subseteq \tilde{F}_i^{(4)}$, either:

- (1) $\text{Adm}(\tilde{T}) \leq 6L_i$ or
- (2) $\tilde{T}$ is a path such that (2a) every node in $\tilde{T}$ has degree at most 2 in $\widetilde{\text{MST}}_i$ and (2b) at least one endpoint φ of $\tilde{T}$ is connected via an $\widetilde{\text{MST}}_i$ edge to a node φ' in a subgraph of $\mathbb{X}_1 \cup \mathbb{X}_2$, unless $\mathbb{X}_1 \cup \mathbb{X}_2 = \emptyset$.

The main intuition behind Step 3 is to guarantee properties (2a) and (2b) for every long path $\tilde{T} \in \tilde{F}_i^{(4)}$. Recall that in Item (3) of Definition 6.15, we guarantee that $\mathcal{G}_i$ has no removable edge. Thus, any edge between two nodes in $\tilde{T}$ is not removable. Later, we use this property to argue that the corrected potential change $\Delta_{i+1}^+(\mathcal{X})$ is non-trivial for every subgraph $\mathcal{X}$ formed in the construction of Step 4 below.

Required definitions/preparations for Step 4. Let $\tilde{F}_i^{(4)}$ be the forest obtained from $\tilde{F}_i^{(3)}$ as described in Lemma 8.3. By Item (2b) in Lemma 8.3, every tree of augmented diameter at least $6L_i$ of $\tilde{F}_i^{(4)}$ is a simple path, which we call a *long path*.

Red/Blue Coloring. Given a path $\tilde{P} \subseteq \tilde{F}_i^{(4)}$, we color their nodes red or blue. If a node has augmented distance at most L_i from at least one of the path's endpoints, we color it red; otherwise, we color it blue. Observe that each red node belongs to the suffix or prefix of $\tilde{P}$; the other nodes are colored blue.

Lemma 8.4 (Step 4). *Let $\tilde{F}_i^{(4)}$ be the forest obtained from $\tilde{F}_i^{(3)}$ as described in Lemma 8.3. We can construct in $O((|\mathcal{V}_i| + |\mathcal{E}_i|)\epsilon^{-1})$ time a collection $\mathbb{X}_4$ of subgraphs of $\mathcal{G}_i$ such that every $\mathcal{X} \in \mathbb{X}_4$:*

- (1) $\mathcal{X}$ contains a single edge in $\mathcal{E}_i$.
- (2) $L_i \leq \text{Adm}(\mathcal{X}) \leq 5L_i$.
- (3) $|\mathcal{V}(\mathcal{X})| = \Theta(\frac{1}{\epsilon})$ when $\epsilon \leq 1/(8(g+1))$.
- (4) $\Delta_{i+1}^+(\mathcal{X}) = \Omega(\epsilon^2 |\mathcal{V}(\mathcal{X})| L_i)$.
- (5) Let $\tilde{F}_i^{(5)}$ be obtained from $\tilde{F}_i^{(4)}$ by removing every node contained in subgraphs of $\mathbb{X}_4$. If we apply Red/Blue Coloring to each path of augmented diameter at least $6L_i$ in $\tilde{F}_i^{(5)}$, then there is no edge in $\mathcal{E}_i$ that connects two blue nodes in $\tilde{F}_i^{(5)}$.

Proof: We only apply the construction to paths of augmented diameter at least $6L_i$ in $\tilde{F}_i^{(4)}$, called *long paths*. Let $\tilde{P}$ be a long path. For each blue node $\varphi \in \tilde{P}$, we assign a subpath $\mathcal{I}(\varphi)$ of $\tilde{P}$, called the *interval of φ* , which contains every node within an augmented distance (in $\tilde{P}$) at most L_i from φ . By definition, we have:

Claim 8.5. *For any blue node ν , it holds that*

- (a) $(2 - (3g + 2)\epsilon)L_i \leq \text{Adm}(\mathcal{I}(\nu)) \leq 2L_i$.
- (b) Denote by $\mathcal{I}_1$ and $\mathcal{I}_2$ the two subpaths obtained by removing ν from the path $\mathcal{I}(\nu)$. Each of these subpaths has $\Theta(\frac{1}{\epsilon})$ nodes and augmented diameter at least $(1 - 2(g + 1)\epsilon)L_i$.

We keep track of a list $\mathcal{B}$ of edges in $\mathcal{E}_i$ with both *blue endpoints*. We then construct $\mathbb{X}_4$, initially empty, as follows:

- Pick an edge (ν, μ) with both blue endpoints, form a subgraph $\mathcal{X} = \{(\nu, \mu) \cup \mathcal{I}(\nu) \cup \mathcal{I}(\mu)\}$, and add $\mathcal{X}$ to $\mathbb{X}_4$. We then remove all nodes in $\mathcal{I}_\nu \cup \mathcal{I}_\mu$ from the path or two paths containing ν and μ , update the color of nodes in the new paths to satisfy Red/Blue Coloring and the edge set $\mathcal{B}$, and repeat this step until it no longer applies.

We observe that Items (1) and (5) follow directly from the construction. For Item (2), we observe by Claim 8.5 that $\mathcal{I}(\nu)$ has augmented diameter at most $2L_i$ and at least L_i when $\epsilon \leq \frac{1}{8(g+1)}$, and the weight of the edge (μ, ν) is at most L_i . Thus, $L_i \leq \text{Adm}(\mathcal{X}) \leq L_i + 2 \cdot 2L_i = 5L_i$, as claimed. Item (3) follows directly from Claim 8.5 since $|\mathcal{I}(\nu)| = \Theta(\frac{1}{\epsilon})$ and $|\mathcal{I}(\mu)| = \Theta(\frac{1}{\epsilon})$.

Next, we show that the construction of $\mathbb{X}_4$ can be implemented efficiently. Since the interval $\mathcal{I}(\nu)$ assigned to each blue node ν consists of $O(\frac{1}{\epsilon})$ nodes by Claim 8.5(b), it takes $O(|\mathcal{E}_i|\epsilon^{-1})$ time to construct $\mathcal{B}$. For each edge $(\nu, \mu) \in \mathcal{B}$ picked in the construction of $\mathbb{X}_4$, forming $\mathcal{X} = \{(\nu, \mu) \cup \mathcal{I}(\nu) \cup \mathcal{I}(\mu)\}$ takes $O(1)$ time. When removing any such interval $\mathcal{I}(\nu)$ from a path $\tilde{P}$, we may create two new sub-paths $\tilde{P}_1, \tilde{P}_2$, and then need to recolor the nodes following Red/Blue Coloring. Specifically, some blue nodes in the prefix and/or suffix of $\tilde{P}_1, \tilde{P}_2$ are colored red; importantly, a node's color may only change from blue to red, but it may not change in the other direction. Since the total number of nodes to be recolored as a result of removing such an interval $\mathcal{I}(\nu)$ is $O(\frac{1}{\epsilon})$, the total recoloring running time is $O(|\mathcal{V}(\tilde{F}_i^{(4)})|\epsilon^{-1}) = O(|\mathcal{V}_i|\epsilon^{-1})$. To bound the time required for updating the edge set $\mathcal{B}$ throughout this process, we note that edges are never added to $\mathcal{B}$ after its initiation. Specifically, when a blue node ν is recolored as red, we remove all incident edges of ν from $\mathcal{B}$, and none of these edges will be considered again; this can be done in $O(\frac{1}{\epsilon})$ time per node ν , since ν is incident to at most $\frac{2g}{\epsilon} = O(\frac{1}{\epsilon})$ edges in $\mathcal{E}_i$ due to the construction of Step 1 (Lemma 8.1). Once a node is added to $\mathcal{X}$, it will never be considered again. It follows that the total running time required for implementing Step 3 is $O(|\mathcal{V}_i|\epsilon^{-1})$, as claimed.

We now prove Item (4). We consider two cases.

Case 1: $\mathcal{I}(\nu) \cap \mathcal{I}(\mu) = \emptyset$. Let $\mathcal{X} = (\nu, \mu) \cup \mathcal{I}(\nu) \cup \mathcal{I}(\mu)$ where $\mathbf{e} = (\nu, \mu)$ is the only edge in $\mathcal{E}_i$ contained in $\mathcal{X}$. For any subgraph $\mathcal{Z}$ of $\mathcal{X}$, we define:

$$\Phi^+(\mathcal{Z}) = \sum_{\alpha \in \mathcal{Z}} \omega(\alpha) + \sum_{\mathbf{e}' \in \widetilde{\text{MST}}_i \cap \mathcal{E}(\mathcal{Z})} \omega(\mathbf{e}) \quad (39)$$

to be the total weight of nodes and $\widetilde{\text{MST}}_i$ edges in $\mathcal{Z}$. Let $\mathcal{D}$ be an augmented diameter path of $\mathcal{X}$, and $\mathcal{Y} = \mathcal{X} \setminus \mathcal{V}(\mathcal{D})$ be the subgraph obtained from $\mathcal{X}$ by removing nodes on $\mathcal{D}$. Let $\mathcal{I}(\nu)$ and $\mathcal{I}(\mu)$ be two intervals in the construction on Step 4 that are connected by an edge $\mathbf{e} = (\nu, \mu)$.

Claim 8.6. $\Phi^+(\mathcal{Y}) = \frac{5L_i}{4} + \Omega(|\mathcal{V}(\mathcal{Y})|\epsilon L_i)$.

Proof: Let $\mathcal{A} = \mathcal{Y} \setminus (\mathcal{I}(\nu) \cup \mathcal{I}(\mu))$ be the subgraph of $\mathcal{Y}$ obtained by removing every node in $\mathcal{I}(\nu) \cup \mathcal{I}(\mu)$ from $\mathcal{Y}$, and $\mathcal{B} = \mathcal{Y} \cap (\mathcal{I}(\nu) \cup \mathcal{I}(\mu))$ be the subgraph of $\mathcal{Y}$ induced by nodes of $\mathcal{Y}$ in $(\mathcal{I}(\nu) \cup \mathcal{I}(\mu))$. Observe that $\Phi^+(\mathcal{A}) \geq |\mathcal{V}(\mathcal{A})|L_{i-1} = |\mathcal{V}(\mathcal{A})|\epsilon L_i$. If $\mathcal{D}$ does not contain the edge (ν, μ) (see Figure 7(a)), then $\mathcal{I}(\nu) \cap \mathcal{D} = \emptyset$, say, which implies $\Phi^+(\mathcal{B}) \geq \text{Adm}(\mathcal{I}(\nu)) \geq (2 - (3g+2)\epsilon)L_i$ by Claim 8.5. If $\mathcal{D}$ contains the edge (ν, μ) (see Figure 7(b)), then at least two sub-intervals, say $\mathcal{I}_1, \mathcal{I}_2$, are disjoint from $\mathcal{D}$. By Claim 8.5, $\Phi^+(\mathcal{B}) \geq \text{Adm}(\mathcal{I}_1) + \text{Adm}(\mathcal{I}_2) \geq (2 - 4(g+1)\epsilon)L_i$. In both cases, $\Phi^+(\mathcal{B}) \geq (2 - 4(g+1)\epsilon)L_i \geq \frac{3L_i}{2}$ when $\epsilon \leq \frac{1}{8(g+1)}$. Thus:

$$\Phi^+(\mathcal{Y}) \geq \Phi^+(\mathcal{A}) + \frac{3L_i}{2} = \frac{5L_i}{4} + \Omega((|\mathcal{V}(\mathcal{A})| + |\mathcal{V}(\mathcal{B})|)\epsilon L_i) = \frac{5L_i}{4} + \Omega(|\mathcal{V}(\mathcal{Y})|\epsilon L_i),$$

which concludes the proof of Claim 8.6. $\square$

Note that $|\mathcal{V}(\mathcal{D})| \leq \frac{gL_i}{L_{i-1}} = O(\frac{1}{\epsilon})$ since every node has weight at least L_{i-1} by property (P3'). Thus, we have:

$$\begin{aligned} \Delta_i^+(\mathcal{X}) &= \Phi^+(\mathcal{D}) + \Phi^+(\mathcal{Y}) - \text{Adm}(\mathcal{X}) = \Phi(\mathcal{Y}) - \omega(\mathbf{e}) \\ &\geq L_i/4 + \Omega(|\mathcal{V}(\mathcal{Y})|\epsilon L_i) \quad (\text{by Claim 8.6}) \\ &= \Omega(|\mathcal{V}(\mathcal{D})|\epsilon L_i) + \Omega(|\mathcal{V}(\mathcal{Y})|\epsilon L_i) = \Omega(|\mathcal{V}(\mathcal{X})|\epsilon L_i). \end{aligned}$$

Thus, Item (4) of Lemma 8.4 follows.

Case 2: $\mathcal{I}(\nu) \cap \mathcal{I}(\mu) \neq \emptyset$. Let $\mathcal{D}$ be a diameter path of $\mathcal{X}$, and $\mathcal{Y} = \mathcal{X} \setminus \mathcal{V}(\mathcal{D})$. Recall that $\mathcal{X}$ contains only one edge $\mathbf{e} = (\nu, \mu) \in \mathcal{E}_i$. Let $\mathcal{P}_{\mathbf{e}} = (\nu, \mathbf{e}, \mu)$ be the path that consists of only edge $\mathbf{e}$ and its endpoints. Let $\widetilde{\text{MST}}_i$ between ν and μ .

We observe that $\mathbf{e}$ is not removable by Item (3) of Definition 6.15. Then it follows that:

$$\begin{aligned} \omega(\mathcal{P}[\nu, \mu]) - \omega(\mathcal{P}_{\mathbf{e}}) &> 6g\epsilon \cdot \omega(\mathbf{e}) - w(\nu) - w(\mu) \\ &> 6g\epsilon L_i/2 - 2g\epsilon L_i = g\epsilon L_i \end{aligned} \quad (40)$$

In particular, this means that $\omega(\mathcal{P}(\nu, \mu)) \geq \omega(\mathbf{e})$.

Thus, if $\mathcal{D}$ contains both ν and μ , then it must contain $\mathbf{e}$, since otherwise, $\mathcal{D}$ must contain $\mathcal{P}[\nu, \mu]$ and by replacing $\mathcal{P}[\nu, \mu]$ by $\mathcal{P}_{\mathbf{e}}$ we obtain a shorter path by Equation (40) (see Figure 8). Observe that

Observation 8.7. $|\mathcal{V}(\mathcal{P}[\nu, \mu])| \leq \frac{4}{\epsilon}$ and $|\mathcal{V}(\mathcal{D})| \leq \frac{g}{\epsilon}$.

We consider two cases:

- *Case 1* If $\mathcal{D}$ does not contain edge $\mathbf{e}$, then (a) $\mathcal{D} \subseteq \widetilde{\text{MST}}_i$ and (b) $|\{\nu, \mu\} \cap \mathcal{D}| \leq 1$. From (a) and Observation 8.7, we have:

$$\begin{aligned} \Delta_{i+1}^+(\mathcal{X}) &\geq \text{Adm}(\mathcal{D}) + \Phi^+(\mathcal{Y}) - \text{Adm}(\mathcal{X}) = \Phi^+(\mathcal{Y}) \\ &\geq \text{Adm}(\mathcal{P}[\mu, \nu]) + \Phi^+(\mathcal{Y} \setminus \mathcal{P}[\mu, \nu]) \\ &\geq w(\mathbf{e}) + |\mathcal{V}(\mathcal{Y} \setminus \mathcal{P}[\mu, \nu])|L_{i-1} \geq L_i/2 + |\mathcal{V}(\mathcal{Y} \setminus \mathcal{P}[\mu, \nu])|\epsilon L_i \\ &= \Omega(\epsilon(|\mathcal{V}(\mathcal{P}[\mu, \nu])| + |\mathcal{V}(\mathcal{D})|)L_i) + |\mathcal{V}(\mathcal{Y} \setminus \mathcal{P}[\mu, \nu])|\epsilon L_i = \Omega(|\mathcal{V}(\mathcal{X})|\epsilon L_i) \end{aligned} \quad (41)$$

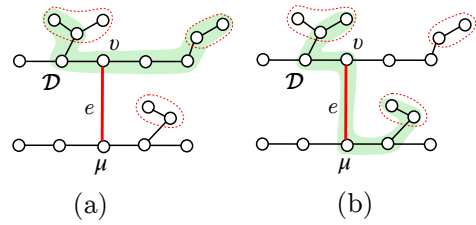


Figure 7: $\mathcal{D}$ is the diameter path and enclosed trees are augmented to a Step-4 subgraph in Step 5A. The green shaded regions contain nodes in $\mathcal{D}$. (a) $\mathcal{D}$ does not contain $\mathbf{e}$. (b) $\mathcal{D}$ contains $\mathbf{e}$.

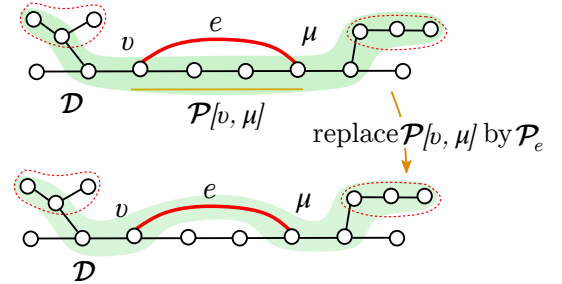


Figure 8: Nodes enclosed in dashed red curves are augmented to $\mathcal{X}$ in Step 4.

- *Case 2* If $\mathcal{D}$ contains $\mathbf{e}$, then $\mathcal{D} \cap \mathcal{P}(\nu, \mu) = \emptyset$; here $\mathcal{P}(\nu, \mu)$ is the path obtained from $\mathcal{P}[\nu, \mu]$ by removing its endpoints. It follows that

$$\begin{aligned}
\Delta_{i+1}^+(\mathcal{X}) &\geq \text{Adm}(\mathcal{D}) + \Phi^+(\mathcal{Y}) - \text{Adm}(\mathcal{X}) = \Phi(\mathcal{Y}) - w(\mathbf{e}) \\
&\geq \text{Adm}(\mathcal{P}[\mu, \nu]) + \Phi^+(\mathcal{Y} \setminus \mathcal{P}[\mu, \nu]) - w(\mathbf{e}) \\
&\geq g\epsilon L_i + |\mathcal{V}(\mathcal{Y} \setminus \mathcal{P}[\mu, \nu])|L_{i-1} \quad (\text{by Equation (40)}) \\
&= \Omega(|\mathcal{V}(\mathcal{P}[\mu, \nu])| + |\mathcal{V}(\mathcal{D})|)\epsilon^2 L_i + |\mathcal{V}(\mathcal{Y} \setminus \mathcal{P}[\mu, \nu])|\epsilon L_i = \Omega(|\mathcal{V}(\mathcal{X})|)\epsilon^2 L_i
\end{aligned} \tag{42}$$

In both cases, we have $\Delta_{i+1}^+(\mathcal{X}) = \Omega(|\mathcal{V}(\mathcal{X})|)\epsilon^2 L_i$ as claimed in Item (4) of Lemma 8.4. $\square$

Observation 8.8. *For every tree $\tilde{T} \subseteq \tilde{F}_i^{(5)}$ such that $\text{Adm}(\tilde{T}) \leq 6L_i$, then $\tilde{T}$ is connected via $\widetilde{\text{MST}}_i$ edge to a node in some subgraph $\mathcal{X} \in \mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4$, unless there is no subgraph formed in Steps 1-4, i.e., $\mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4 = \emptyset$.*

We call the case where $\mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4 = \emptyset$ the *degenerate case*. When the degenerate case happens, $\mathcal{G}_i$ has a very special structure, which will be described later (in Lemma 8.11); for now, we focus on the construction of the last step.

Step 5. Let $\tilde{T}$ be a path in $\tilde{F}_i^{(5)}$ obtained by Item (5) of Lemma 8.4. We construct two sets of subgraphs, denoted by $\mathbb{X}_5^{\text{intrnl}}$ and $\mathbb{X}_5^{\text{pref}}$, of $\mathcal{G}_i$. The construction is broken into two steps. Step 5A is only applicable when the degenerate case does not happen; Step 5B is applicable regardless of the degenerate case.

- (Step 5A) If $\tilde{T}$ has augmented diameter at most $6L_i$, let $\mathbf{e}$ be an $\widetilde{\text{MST}}_i$ edge connecting $\tilde{T}$ and a node in some subgraph $\mathcal{X} \in \mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4$; $\mathbf{e}$ exists by Observation 8.8. We add both $\mathbf{e}$ and $\tilde{T}$ to $\mathcal{X}$.
- (Step 5B) Otherwise, the augmented diameter of $\tilde{T}$ is at least $6L_i$ and hence, it must be a path by Item (4) in Lemma 8.2. In this case, we greedily break $\tilde{T}$ into subpaths of augmented diameter at least L_i and at most $2L_i$. Let $\tilde{P}$ be a subpath broken from $\tilde{T}$. If $\tilde{P}$ is connected to a node in a subgraph $\mathcal{X}$ via an edge $\mathbf{e} \in \widetilde{\text{MST}}_i$, we add $\tilde{P}$ and $\mathbf{e}$ to $\mathcal{X}$. If $\tilde{P}$ contains an endpoint of $\tilde{T}$, we add $\tilde{P}$ to $\mathbb{X}_5^{\text{pref}}$; otherwise, we add $\tilde{P}$ to $\mathbb{X}_5^{\text{intrnl}}$.

Lemma 8.9. *We can implement the construction of $\mathbb{X}_5^{\text{intrnl}}$ and $\mathbb{X}_5^{\text{pref}}$ in $O(|\mathcal{V}_i|)$ time. Furthermore, every subgraph $\mathcal{X} \in \mathbb{X}_5^{\text{intrnl}} \cup \mathbb{X}_5^{\text{pref}}$ satisfies:*

- (1) $\mathcal{X}$ is a subpath of $\widetilde{\text{MST}}_i$.
- (2) $L_i \leq \text{Adm}(\mathcal{X}) \leq 2L_i$ when $\epsilon \leq 1/g$.
- (3) $|\mathcal{V}(\mathcal{X})| = \Theta(\frac{1}{\epsilon})$.

Proof: Items (1) and (2) follow directly from the construction. For Item (3), we observe the following facts: $\text{Adm}(\mathcal{X}) \geq L_i$, each edge has a weight of at most L_{i-1} , and each node has a weight of at most gL_{i-1} . Thus, $|\mathcal{V}(\mathcal{X})| \geq \frac{L_i}{(1+g)L_{i-1}} = \Omega(\frac{1}{\epsilon})$. By the same argument, since each node has a weight at least L_{i-1} by property (P3'), $|\mathcal{V}(\mathcal{X})| \leq \frac{2L_i}{L_{i-1}} = O(1/\epsilon)$. The construction time follows by implementing the algorithm greedily.

Finally, we construct the collection $\mathbb{X}$ of subgraphs of $\mathcal{G}_i$ as follows:

$$\mathbb{X} = \mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4 \cup \mathbb{X}_5^{\text{intrnl}} \cup \mathbb{X}_5^{\text{pref}}. \tag{43}$$

To complete the proof of Lemma 7.3, we need to:

1. show that subgraphs in $\mathbb{X}$ satisfies three properties: (P1'), (P2'), and (P3'), and that $|\mathcal{E}_i \cap \mathcal{E}(\mathcal{X})| = O(|\mathcal{V}(\mathcal{X})|)$. This implies Item (5) of Lemma 7.3. We present the proof in Section 8.1.
2. construct a partition $\{\mathcal{V}_i^{\text{high}}, \mathcal{V}_i^{\text{low}^+}, \mathcal{V}_i^{\text{low}^-}\}$ of $\mathcal{V}_i$, show Items (1)-(4) and the running time bound as claimed by Lemma 7.3. We present the proof in Section 8.2

8.1 Properties of $\mathbb{X}$

In this section, we prove the following lemma.

Lemma 8.10. *Let $\mathbb{X}$ be the set of subgraph as defined in Equation (43). For every subgraph $\mathcal{X} \in \mathbb{X}$, $\mathcal{X}$ satisfies the three properties (P1')-(P3') with $g = 31$ and $\epsilon \leq \frac{1}{8(g+1)}$, and $|\mathcal{E}(\mathcal{X}) \cap \mathcal{E}_i| = O(|\mathcal{V}(\mathcal{X})|)$. Furthermore, $\mathbb{X}$ can be constructed in $O((|\mathcal{V}_i| + |\mathcal{E}_i|)\epsilon^{-1})$ time.*

Proof: We observe that property (P1') follows directly from the construction. Additionally, property (P2') follows from Item (4) of Lemma 8.1, Items (3) of Lemma 8.2, Lemma 8.4, and Lemma 8.9. The lower bound L_i on the augmented diameter of a subgraph $\mathcal{X} \in \mathbb{X}$ follows from Item (3) of Lemma 8.1, Items (2) of Lemma 8.2, Lemma 8.4, and Lemma 8.9. Thus, to complete the proof of property (P3'), it remains to show that $\text{Adm}(\mathcal{X}) \leq gL_i$ with $g = 31$ and $\epsilon \leq \frac{1}{8(g+1)}$. Observe that the condition that $\epsilon \leq \frac{1}{8(g+1)}$ follows by considering all constraints on ϵ in Lemmas 8.1 to 8.4 and 8.9.

If $\mathcal{X}$ is formed in Step 5B, that is $\mathcal{X} \in \mathbb{X}_5^{\text{intrnl}} \cup \mathbb{X}_5^{\text{pref}}$, then $\text{Adm}(\mathcal{X}) \leq 2L_i$ by Lemma 8.9. Otherwise, excluding any augmentation to $\mathcal{X}$ due to Step 5, Lemma 8.1, Lemma 8.2 and Lemma 8.3 yield $\text{Adm}(\mathcal{X}) \leq 13L_i + 4L_i \leq 17L_i$ where $+4L_i$ is due to the augmentation in Step 3 (see Lemma 8.3). By Lemma 8.4, $\text{Adm}(\mathcal{X}) \leq \max(17L_i, 5L_i) = 17L_i$.

We then may augment $\mathcal{X}$ with trees of diameter at most $6L_i$ (Step 5A) and/or with subpaths of diameter at most $2L_i$ (Step 5B). As the augmentation is star-like and via $\widetilde{\text{MST}}_i$ edges, if we denote the resulting subgraph by $\mathcal{X}^+$, then

$$\text{Adm}(\mathcal{X}^+) \leq \text{Adm}(\mathcal{X}) + 2\bar{w} + 12L_i \leq \text{Adm}(\mathcal{X}) + 14L_i \leq 31L_i.$$

Property (P3') now follows.

The fact that $|\mathcal{E}(\mathcal{X}) \cap \mathcal{E}_i| = O(|\mathcal{V}(\mathcal{X})|)$ and the running time bound follow directly from Lemma 8.1, Lemma 8.2, Lemma 8.3, Lemma 8.4 and Lemma 8.9. Recall that the augmentation in Step 3 is in a star-like way and hence, no cycle is formed in subgraphs of $\mathbb{X}_1 \cup \mathbb{X}_2$ after the augmentation. $\square$

8.2 Constructing a Partition of $\mathcal{V}_i$

We first consider the degenerate case where $\mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4 = \emptyset$.

Lemma 8.11 (Structure of Degenerate Case). *If $\mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4 = \emptyset$, then $\widetilde{F}_i^{(5)} = \widetilde{\text{MST}}_i$, and $\widetilde{\text{MST}}_i$ is a single (long) path. Moreover, every edge $\mathbf{e} \in \mathcal{E}_i$ must be incident to a node in $\widetilde{P}_1 \cup \widetilde{P}_2$, where $\widetilde{P}_1$ and $\widetilde{P}_2$ are the prefix and suffix subpaths of $\widetilde{\text{MST}}_i$ of augmented diameter at most L_i . Consequently, we have that $|\mathcal{E}_i| = O(1/\epsilon^2)$.*

Proof: By the assumption of the lemma, no subgraph is formed in Steps 1-4.

Since no subgraph is formed in Step 1, $\tilde{F}_i^{(2)} = \widetilde{\text{MST}}_i$. Since no subgraph is formed in Step 2, there is no branching node in $\tilde{F}_i^{(2)}$; thus $\tilde{F}_i^{(3)} = \tilde{F}_i^{(2)}$ and it is a single (long) path. Since $\mathbb{X}_1 \cup \mathbb{X}_2 = \emptyset$, there is no augmentation in Step 3. Since no subgraph is formed in Step 4, $\tilde{F}_i^{(5)} = \tilde{F}_i^{(4)}$ and both are equal to $\widetilde{\text{MST}}_i$, which is a long path (see Figure 9).

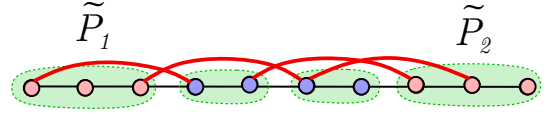


Figure 9: Red edges are edges in $\mathcal{E}_i$; every edge is incident to at least one red node.

By Item (5) in Lemma 8.4, any edge $e \in \mathcal{E}_i$ must be incident to a red node. The augmented distance from any red node to at least one endpoint of $\widetilde{\text{MST}}_i$ is at most L_i by the definition of Red/Blue Coloring, and hence every red node belongs to $\tilde{P}_1 \cup \tilde{P}_2$. Since each node has a weight of at least L_{i-1} by property (P3'), we have:

$$|\mathcal{V}(\tilde{P}_1 \cup \tilde{P}_2)| \leq \frac{2L_i}{L_{i-1}} = \frac{2}{\epsilon}$$

Since each node of $\tilde{P}_1 \cup \tilde{P}_2$ is incident to at most $\frac{2g}{\epsilon}$ edges in $\mathcal{E}_i$ (as there is no subgraph formed in Step 1; $\mathcal{V}_i^{\text{high}} = \emptyset$), it holds that $|\mathcal{E}_i| = O(1/\epsilon^2)$, as desired. $\square$

We are now ready to describe the construction of the partition $\{\mathcal{V}_i^{\text{high}}, \mathcal{V}_i^{\text{low}^+}, \mathcal{V}_i^{\text{low}^-}\}$ of $\mathcal{V}_i$

Construct Partition $\{\mathcal{V}_i^{\text{high}}, \mathcal{V}_i^{\text{low}^+}, \mathcal{V}_i^{\text{low}^-}\}$: If the degenerate case happens, we define $\mathcal{V}_i^{\text{low}^-} = \mathcal{V}_i$ and $\mathcal{V}_i^{\text{high}} = \mathcal{V}_i^{\text{low}^+} = \emptyset$. Otherwise, we define $\mathcal{V}_i^{\text{high}}$ to be the set of all nodes that are incident to at least $2g/\epsilon$ edges in $\mathcal{E}_i$, $\mathcal{V}_i^{\text{low}^-} = \cup_{\mathcal{X} \in \mathbb{X}_5^{\text{intrnl}}} \mathcal{V}(\mathcal{X})$ and $\mathcal{V}_i^{\text{low}^+} = \mathcal{V}_i \setminus (\mathcal{V}_i^{\text{high}} \cup \mathcal{V}_i^{\text{low}^-})$.

We show the following property of $\{\mathcal{V}_i^{\text{high}}, \mathcal{V}_i^{\text{low}^+}, \mathcal{V}_i^{\text{low}^-}\}$, which is equivalent to Item (4) in Lemma 7.3.

Lemma 8.12. (1) If $\mathcal{X}$ contains a node in $\mathcal{V}_i^{\text{low}^-}$, then $\mathcal{V}(\mathcal{X}) \subseteq \mathcal{V}_i^{\text{low}^-}$.

(2) There is no edge in $\mathcal{E}_i$ between a node in $\mathcal{V}_i^{\text{high}}$ and a node in $\mathcal{V}_i^{\text{low}^-}$.

(3) If there exists an edge $(\varphi_{C_u}, \varphi_{C_v}) \in \mathcal{E}_i$ such that both φ_{C_u} and φ_{C_v} are in $\mathcal{V}_i^{\text{low}^-}$, then the degenerate case happens.

Proof: Item (1) follows directly from the construction. By the construction of Step 1 (Lemma 8.1), any neighbor, say φ , of a node in $\mathcal{V}_i^{\text{high}}$ is in $\mathcal{V}_i^{\text{high}^+}$. Thus, φ will not be considered after Step 1. It follows that there is no edge between a node in $\mathcal{V}_i^{\text{high}}$ and a node in $\mathcal{V}_i^{\text{low}^-}$ since nodes in $\mathcal{V}_i^{\text{low}^-}$ are in Step 5; Item (2) follows. To show Item (3), we observe that every node, say φ_{C_u} , in $\mathcal{V}_i^{\text{low}^-}$ is a blue node of some long path $\tilde{P}$ in $\tilde{F}_i^{(5)}$. If the degenerate case does not happen, then by Item (5) of Lemma 8.4, every edge $(\varphi_{C_u}, \varphi_{C_v})$ must have the node φ_{C_v} being a red node of $\tilde{P}$. But then by the construction of Step 5B, φ_{C_v} belongs to some subgraph of $\mathbb{X}_5^{\text{pref}}$ and hence is not in $\mathcal{V}_i^{\text{low}^-}$. $\square$

Next, we focus on bounding the corrected potential change $\Delta_i^+(\mathcal{X})$ of every cluster $\mathcal{X} \in \mathbb{X}$. Specifically, we show that:

- if $\mathcal{X} \in \mathbb{X}_1$, then $\Delta_{i+1}^+(\mathcal{X}) = \Omega(|\mathcal{V}(\mathcal{X})|L_i\epsilon)$; the proof is in Lemma 8.13.
- if $\mathcal{X} \in \mathbb{X}_2$, then $\Delta_{i+1}^+(\mathcal{X}) = \Omega(|\mathcal{V}(\mathcal{X})|L_i\epsilon^2)$; the proof is in Lemma 8.14.
- if $\mathcal{X} \in \mathbb{X}_4$, then $\Delta_{i+1}^+(\mathcal{X}) = \Omega(|\mathcal{V}(\mathcal{X})|L_i\epsilon^2)$; the proof is in Lemma 8.15.

- the corrected potential change is non-negative and we provide a lower bound of the average corrected potential change for subgraphs in $\mathbb{X} \setminus \mathbb{X}^{\text{low-}}$ in Lemma 8.16.

Lemma 8.13. *For every subgraph $\mathcal{X} \in \mathbb{X}_1$, it holds that $\Delta_{i+1}^+(\mathcal{X}) \geq \frac{|\mathcal{V}(\mathcal{X})|L_i\epsilon}{2}$.*

Proof: Let $\mathcal{X} \in \mathbb{X}_1$ be a subgraph formed in Step 1. By Item (4) of Lemma 8.1, $|\mathcal{V}(\mathcal{X})| \geq \frac{2g}{\epsilon}$. By definition of $\Delta_L^i(\mathcal{X})$ (Lemma 7.3), we have:

$$\begin{aligned} \Delta_{i+1}^+(\mathcal{X}) &\geq \sum_{\varphi \in \mathcal{V}(\mathcal{X})} \omega(\varphi) - \text{Adm}(\mathcal{X}) \stackrel{(\text{P3}')} {\geq} \sum_{\varphi \in \mathcal{X}} L_{i-1} - gL_i = \frac{|\mathcal{V}(\mathcal{X})|L_{i-1}}{2} + \underbrace{\left(\frac{|\mathcal{V}(\mathcal{X})|L_{i-1}}{2} - gL_i \right)}_{\geq 0 \text{ since } |\mathcal{V}(\mathcal{X})| \geq (2g)/\epsilon} \\ &\geq \frac{|\mathcal{V}(\mathcal{X})|L_{i-1}}{2} = \frac{|\mathcal{V}(\mathcal{X})|\epsilon L_i}{2}, \end{aligned} \quad (44)$$

as claimed. $\square$

Lemma 8.14. *For every subgraph $\mathcal{X} \in \mathbb{X}_2$, it holds that $\Delta_{i+1}^+(\mathcal{X}) = \Omega(|\mathcal{V}(\mathcal{X})|L_i\epsilon^2)$.*

Proof: Let $\mathcal{X}$ be a subgraph that is initially formed in Step 2 and could possibly be augmented in Steps 3 and 5. Recall that in the augmentation done in Step 3, we add to $\mathcal{X}$ nodes of $\mathcal{V}_i$ via $\widetilde{\text{MST}}_i$ edges, and in the augmentation done in Step 5, we add to $\mathcal{X}$ subtrees of $\widetilde{\text{MST}}_i$ via $\widetilde{\text{MST}}_i$ edges. Thus, the resulting subgraph after the augmentation remains, as prior to the augmentation, a subtree of $\widetilde{\text{MST}}_i$. That is, $\mathcal{E}(\mathcal{X}) \subseteq \widetilde{\text{MST}}_i$. Letting $\mathcal{D}$ be an augmented diameter path of $\mathcal{X}$, we have by definition of augmented diameter that

$$\text{Adm}(\mathcal{X}) = \sum_{\varphi \in \mathcal{D}} \omega(\varphi) + \sum_{e \in \mathcal{E}(\mathcal{D})} \omega(e)$$

Let $\mathcal{Y} = \mathcal{V}(\mathcal{X}) \setminus \mathcal{V}(\mathcal{D})$. Then $|\mathcal{Y}| > 0$ since $\mathcal{X}$ has a $\mathcal{X}$ -branching node by Item (1) of Lemma 8.2 and that

$$\Delta_{i+1}^+(\mathcal{X}) = \left(\sum_{\varphi \in \mathcal{X}} \omega(\varphi) + \sum_{e \in \mathcal{E}(\mathcal{X})} \omega(e) \right) - \text{Adm}(\mathcal{X}) \geq \sum_{\varphi \in \mathcal{Y}} \omega(\varphi) \stackrel{(\text{P3}')} {\geq} |\mathcal{Y}|L_{i-1} \quad (45)$$

As $\text{Adm}(\mathcal{D}) \leq gL_i$, it holds that $|\mathcal{V}(\mathcal{D})| = O(1/\epsilon) = O(\frac{|\mathcal{Y}|}{\epsilon})$. Thus,

$$\Delta_{i+1}^+(\mathcal{X}) \geq \frac{|\mathcal{Y}|L_{i-1}}{2} + \Omega(\epsilon|\mathcal{V}(\mathcal{D})|L_{i-1}) = \Omega((|\mathcal{Y}| + \mathcal{V}(\mathcal{D}))\epsilon L_{i-1}) = \Omega(|\mathcal{V}(\mathcal{X})|\epsilon^2 L_i),$$

as claimed. $\square$

Lemma 8.15. *For every subgraph $\mathcal{X} \in \mathbb{X}_4$, it holds that $\Delta_{i+1}^+(\mathcal{X}) = \Omega(|\mathcal{V}(\mathcal{X})|L_i\epsilon^2)$.*

Proof: Let $\mathcal{X} \in \mathbb{X}_4$ be a subgraph initially formed in Step 4; $\mathcal{X}$ is possibly augmented in Step 5. Let $\mathcal{X}^+$ be $\mathcal{X}$ after the augmentation (if any). Let $\mathcal{D}^+$ be the augmented diameter path of $\mathcal{X}^+$ and $\mathcal{D} = \mathcal{D}^+ \cap \mathcal{X}$. By the same argument in Lemma 8.14,

$$\Delta_{i+1}^+(\mathcal{X}) = \Omega(|\mathcal{V}(\mathcal{X})|\epsilon^2 L_i) = \Omega(|\mathcal{V}(\mathcal{X}) \cup \mathcal{V}(\mathcal{D}^+)|\epsilon^2 L_i). \quad (46)$$

Furthermore,

$$\begin{aligned}
\Delta_{i+1}^+(\mathcal{X}^+) &= \sum_{\varphi \in \mathcal{X}^+} \omega(\varphi) + \sum_{\mathbf{e} \in \mathcal{E}(\mathcal{X}^+) \cap \widetilde{\text{MST}}_i} \omega(\mathbf{e}) - \omega(\mathcal{D}^+) \\
&\geq \sum_{\varphi \in \mathcal{Y}} \omega(\varphi) + \sum_{\varphi \in \mathcal{X}} \omega(\varphi) + \sum_{\mathbf{e} \in \mathcal{E}(\mathcal{X}) \cap \widetilde{\text{MST}}_i} \omega(\mathbf{e}) - \omega(\mathcal{D}) \\
&\geq \Omega(L_i \epsilon |\mathcal{Y}|) + \Delta_{i+1}^+(\mathcal{X}) \stackrel{\text{Eq. (46)}}{=} \Omega(|\mathcal{Y}| \epsilon L_i) + \Omega(|\mathcal{V}(\mathcal{X}) \cup \mathcal{V}(\mathcal{D}^+)| \epsilon^2 L_i) \\
&= \Omega(|\mathcal{V}(\mathcal{X}) \cup \mathcal{V}(\mathcal{D}^+) \cup \mathcal{Y}| \epsilon^2 L_i) = \Omega(|\mathcal{V}(\mathcal{X}^+)| \epsilon^2 L_i),
\end{aligned}$$

as claimed. $\square$

Next, we show Item (3) of Lemma 7.3 regarding the corrected potential changes of subgraphs in $\mathbb{X}$.

Lemma 8.16. $\Delta_{i+1}^+(\mathcal{X}) \geq 0$ for every $\mathcal{X} \in \mathbb{X}$, and

$$\sum_{\mathcal{X} \in \mathbb{X} \setminus \mathbb{X}^{\text{low}-}} \Delta_{i+1}^+(\mathcal{X}) = \sum_{\mathcal{X} \in \mathbb{X} \setminus \mathbb{X}^{\text{low}-}} \Omega(|\mathcal{V}(\mathcal{X})| \epsilon^2 L_i).$$

Proof: If $\mathcal{X} \in \mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4$, then $\Delta_{i+1}^+(\mathcal{X}) \geq 0$ by Lemmas 8.13 to 8.15. Otherwise, $\mathcal{X} \in \mathbb{X}_5^{\text{pref}} \cup \mathbb{X}_5^{\text{intrnl}}$, and hence is a subpath of $\widetilde{\text{MST}}_i$. Thus, by definition, $\Delta_{i+1}^+(\mathcal{X}) = \sum_{\varphi \in \mathcal{X}} \omega(\varphi) + \sum_{\mathbf{e} \in \mathcal{E}(\mathcal{X}) \cap \widetilde{\text{MST}}_i} \omega(\mathbf{e}) - \text{Adm}(\mathcal{X}) = 0$. That is, $\Delta_{i+1}^+(\mathcal{X}) \geq 0$ in every case.

We now show a lower bound on the average potential change of subgraphs in $\mathbb{X} \setminus \mathbb{X}^{\text{low}-}$. We assume that the degenerate case does not happen; otherwise, $\mathbb{X} \setminus \mathbb{X}^{\text{low}-} = \emptyset$ and there is nothing to prove. By Item (1) of Lemma 8.12, $\mathbb{X}^{\text{low}-} = \mathbb{X}_5^{\text{intrnl}}$ and only subgraphs in $\mathbb{X}_5^{\text{pref}}$ may not have positive potential change. By Lemmas 8.13 to 8.15, on average, each node φ in any subgraph $\mathcal{X} \in \mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4$ has $\Omega(\epsilon^2 L_i)$ corrected potential change, denoted by $\overline{\Delta}(\varphi)$.

By construction, a subgraph in $\mathbb{X}_5^{\text{pref}}$ is a prefix (or suffix), say $\tilde{P}_1$, of a long path $\tilde{P}$. The other suffix, say $\tilde{P}_2$, of $\tilde{P}$ is augmented to a subgraph, say $\mathcal{X} \in \mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4$ by the construction of Step 5BFast and Item (2) Lemma 8.3. Since $|\mathcal{V}(\tilde{P}_2)| = \Omega(1/\epsilon)$ by Item (3) of Lemma 8.9, $\sum_{\varphi \in \tilde{P}_2} \overline{\Delta}(\varphi) = \Omega(1/\epsilon)(\epsilon^2 L_i) = \Omega(\epsilon L_i)$. We distribute half this corrected potential change to all the nodes in $\tilde{P}_1$, by Item (3) of Lemma 8.9, each gets $\Omega(\frac{\epsilon L_i}{1/\epsilon}) = \Omega(\epsilon^2 L_i)$. This implies:

$$\sum_{\mathcal{X} \in \mathbb{X} \setminus \mathbb{X}^{\text{low}-}} \Delta_{i+1}^+(\mathcal{X}) = \sum_{\varphi \in \mathcal{V}_i \setminus \mathcal{V}_i^{\text{low}-}} \Omega(\epsilon^2 L_i) = \sum_{\mathcal{X} \in \mathbb{X} \setminus \mathbb{X}^{\text{low}-}} \Omega(|\mathcal{V}(\mathcal{X})| \epsilon^2 L_i),$$

as desired. $\square$

We are now ready to prove Lemma 7.3.

Proof: [Proof of Lemma 7.3] We observe that Items (1), (2) and (4) follow directly Lemma 8.11 and Lemma 8.12. Item (5) follows from Lemma 8.10. Item (3) follows from Lemma 8.16. The construction time is asymptotically the same as the construction time of $\mathbb{X}$, which is $O((|\mathcal{V}_i| + |\mathcal{E}_i|) \epsilon^{-1})$ by Lemma 8.10.

Finally, we compute the augmented diameter of each subgraph $\mathcal{X} \in \mathbb{X}$. We observe that the augmentations in Step 3 and Step 5 do not create any cycle. Thus, if $\mathcal{X}$ is initially formed in Steps 1, 2 or 5B, then finally $\mathcal{X}$ is a tree. It follows that the augmented diameter of $\mathcal{X}$ can be computed in $O(|\mathcal{V}(\mathcal{X})|)$ time by a simple tree traversal. If $\mathcal{X}$ is formed in Step 4, then it has exactly one edge $\mathbf{e}$ not in $\widetilde{\text{MST}}_i$ by Item (1) in Lemma 8.4 and that $\mathcal{X}$ contains at most one cycle. Let $\mathcal{Z}$ be such a cycle (if any); $\mathcal{Z}$ has

$O(1/\epsilon)$ edges by Item (3) in Lemma 8.4. Thus, we can reduce computing the diameter of $\mathcal{X}$ to computing the diameter of trees by guessing an edge of $\mathcal{Z}$ that does not belong to the diameter path of $\mathcal{X}$ and remove this edge from $\mathcal{X}$; the resulting graph is a tree. There are $O(\frac{1}{\epsilon})$ guesses and each for each guess, computing the diameter takes $O(|\mathcal{V}(\mathcal{X})|)$ time, which implies $O(|\mathcal{V}(\mathcal{X})|\epsilon^{-1})$ time to compute $\text{Adm}(\mathcal{X})^4$. Thus, the total running time is $\sum_{\mathcal{X} \in \mathbb{X}} O(|\mathcal{V}(\mathcal{X})|\epsilon^{-1}) = O(|\mathcal{V}_i|\epsilon^{-1})$. $\square$

9 Light Spanners for Minor-free Graphs in Linear Time

In Section 4, we show a construction of a light spanner for K_r -minor-free graphs with running time $O(nr\sqrt{r}\alpha(nr\sqrt{r}, n))$. The extra factor $\alpha(nr\sqrt{r}, n)$ is due to UNION-FIND data structure in the proof of Theorem 1.10. To remove this factor, we do not use UNION-FIND. Instead, we follow the idea of Mareš [52] that was applied to construct a minimum spanning tree for K_r -minor-free graphs. Specifically, after the construction of level- $(i+1)$ clusters, we prune the set of edges that are involved in the construction of levels at least $i+1$, which is $\cup_{j \geq i+1} E_j^\sigma$, as follows.

Let $E_{\geq i}^\sigma = \cup_{j \geq i} E_j^\sigma$. We inductively maintain a set of edges $\mathcal{E}_{\geq i}$, where each edge in $\mathcal{E}_{\geq i}$ corresponds to an edge in $E_{\geq i}^\sigma$. (Note that only those in $\mathcal{E}_i$ are involved in the construction of spanner at level i .) Furthermore, we inductively guarantee that $|\mathcal{E}_{\geq i}| = O(r\sqrt{\log r}|\mathcal{V}_i|)$; we call this the *size invariant*. Upon completing the construction of level- $(i+1)$ clusters, we construct the set of nodes $\mathcal{V}_{i+1}$. We now consider the set of edges $\mathcal{E}'_{\geq i+1} = \mathcal{E}_{\geq i} \setminus \mathcal{E}$. Let $\tilde{\mathcal{E}}_{\geq i+1}$ be obtained from $\mathcal{E}'_{\geq i+1}$ by removing parallel edges: two edges (φ_1, φ_2) and (φ'_1, φ'_2) are *parallel* if there exist two subgraphs $\mathcal{X}, \mathcal{Y} \in \mathbb{X}$ such that, w.l.o.g, $\varphi_1, \varphi'_1 \in \mathcal{V}(\mathcal{X})$ and $\varphi_2, \varphi'_2 \in \mathcal{V}(\mathcal{Y})$. (Among all parallel edges, we keep the edge with minimum weight in $\tilde{\mathcal{E}}_{\geq i+1}$.) We construct the edge set $\mathcal{E}_{\geq i+1}$ (between vertices in $\mathcal{V}_{i+1}$) at level $(i+1)$ from $\tilde{\mathcal{E}}_{\geq i+1}$ by creating one edge $(\mathcal{X}, \mathcal{Y}) \in \mathcal{E}_{\geq i+1}$ for each corresponding edge $(\varphi_x, \varphi_y) \in \tilde{\mathcal{E}}_{\geq i+1}$ where $\varphi_x \in \mathcal{V}(\mathcal{X})$ and $\varphi_y \in \mathcal{V}(\mathcal{Y})$; $\omega(\mathcal{X}, \mathcal{Y}) = \omega(\varphi_x, \varphi_y)$.

Observe that $\mathcal{E}_{i+1}$ corresponds to a subset of edges of $E_{\geq i+1}^\sigma$ since $\mathcal{E}'_{\geq i+1}$, by definition, corresponds to a subset of edges of $E_{\geq i+1}^\sigma$. The stretch is in check (at most $(1 + O(\epsilon))$), since we only remove parallel edges and that level- $(i+1)$ clusters have diameter $O(\epsilon)$ times the weight of level- $(i+1)$ edges by property (P3). Furthermore, since $\mathcal{E}_{\geq i} = O(r\sqrt{\log r}|\mathcal{V}_i|)$ by the size invariant, the construction of $\mathcal{E}_{i+1}$ can be done in $O(|\mathcal{V}_i|)$ time. Since the graph $(\mathcal{V}_{i+1}, \mathcal{E}_{\geq i+1})$ is a minor of G and hence, is K_r -minor-free, we conclude that $|\mathcal{E}_{\geq i+1}| = O(r\sqrt{\log r}|\mathcal{V}_{i+1}|)$ by the sparsity of minor-free graphs, which implies the size invariant for level $i+1$.

By the size invariant, we do not need UNION-FIND data structure, as $\mathcal{E}_{\geq i}$ now has $O(r\sqrt{\log r}|\mathcal{V}_i|) = O(r\sqrt{\log r}|\mathcal{C}_i|)$ edges. Thus, the running time to construct $\mathcal{G}_i$ in Lemma 7.2 becomes $O_\epsilon(|\mathcal{C}_i| + |\mathcal{E}_i|) = O_\epsilon(r\sqrt{\log r}|\mathcal{C}_i|)$, and the running time to construct $\widetilde{\text{MST}}_{i+1}$ in Lemma 7.6 also becomes $O((r\sqrt{\log r}|\mathcal{C}_i|))$. The rest of the proof is the same as the proof in Section 4.3.

10 Fine-Grained Optimal Lightness: Proof of Theorem 1.10(2)

Our goal is to construct a cluster graph $\mathcal{G}_i$ and a collection $\mathbb{X}$ of subgraphs of $\mathcal{G}_i$ satisfying properties (P1')-(P3'). We set $\psi = 1/250$ where ψ is the parameter in Equation (22).

By Lemma 6.17, the set of level- $(i+1)$ obtained from subgraphs in $\mathbb{X}$ obtained by applying the transformation in Equation (26) will satisfy properties (P1)-(P3). To be able to bound the set of edges in H_i (constructed in Sections 11 and 12), we need to guarantee that subgraphs in $\mathbb{X}$ have sufficiently large

⁴It is possible to compute the augmented diameter of $\mathcal{X}$ in $O(|\mathcal{V}(\mathcal{X})|)$ time using a more involved approach.

potential changes. This indeed is the crux of our construction. We assume that $\epsilon > 0$ is a sufficiently small constant, i.e., $\epsilon \ll 1, \epsilon = \Omega(1)$.

Constructing $\mathcal{G}_i$. We shall assume inductively on $i, i \geq 1$ that:

- The set of edges $\widetilde{\text{MST}}_i$ is given by the construction of the previous level i in the hierarchy; for the base case $i = 1$ (see Section 6.2), $\widetilde{\text{MST}}_1$ is simply a set of edges of $\widetilde{\text{MST}}$ that are not in any level-1 cluster.
- The weight $\omega(\varphi_C)$ on each node $\varphi_C \in \mathcal{V}_i$ is the potential value of cluster $C \in \mathcal{C}_i$; for the base case $i = 1$, the potential values of level-1 clusters were set in Equation (25).

After completing the construction of $\mathbb{X}$, we can compute the weight of each node of $\mathcal{G}_{i+1}$ by computing the augmented diameter of each subgraph in $\mathcal{X}$; the running time is clearly polynomial. By the end of this section, we show to compute the spanning tree $\widetilde{\text{MST}}_{i+1}$ for $\mathcal{G}_{i+1}$ for the construction of the next level.

Realization of a path. Let $\mathcal{P} = (\varphi_0, (\varphi_0, \varphi_1), \varphi_1, (\varphi_1, \varphi_2), \dots, \varphi_p)$ be a path of $\mathcal{G}_i$, written as an alternating sequence of vertices and edges. Let C_i be the cluster corresponding to φ_i , $0 \leq i \leq p$. Let u and v be two vertices such that u is in the cluster corresponding to φ_0 and v is in the cluster corresponding to φ_p . See Figure 6 for an illustration.

Let $\{y_i\}_{i=0}^p$ and $\{z_i\}_{i=0}^p$ be sequences of vertices of G such that (a) $z_0 = u$ and $y_p = v$ and (b) (y_{i-1}, z_i) is the edge on G corresponds to edge $(\varphi_{i-1}, \varphi_i)$ in $\mathcal{P}$ for $1 \leq i \leq p$. Let Q_i , $0 \leq i \leq p$, be a shortest path in $H_{<L_{i-1}}[C_i]$ between z_i and y_i where C_i is the cluster corresponding to φ_i . Let $P = Q_0 \circ (y_0, z_1) \circ \dots \circ Q_p$ be a (possibly non-simple) path from u to v . We call P a *realization of $\mathcal{P}$ with respect to u and v* . The following observation follows directly from the definition of the weight function of $\mathcal{G}_i$.

Observation 10.1. *Let P be a realization of $\mathcal{P}$ w.r.t two vertices u and v . Then $w(P) \leq \omega(\mathcal{P})$.*

Next, we show that to construct H_i , it suffices to focus on the edges of E_i^σ that correspond to edges in $\mathcal{E}_i$ of $\mathcal{G}_i$.

Lemma 10.2. *Let $\psi = 1/250$. We can construct a cluster graph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i \cup \widetilde{\text{MST}}_i, \omega)$ in polynomial time such that $\mathcal{G}_i$ satisfies all properties in Definition 6.15. Furthermore, let F_i^σ be the set of edges in E_i^σ that correspond to $\mathcal{E}_i$. If every edge in F_i^σ has a stretch $t(1 + s \cdot \epsilon)$ in $H_{<L_i}$ for some constant $s \geq 1$, then every edge in E_i^σ has stretch $t(1 + (2s + 16g + 1)\epsilon)$ when $\epsilon < \frac{1}{2(12g+1)}$.*

Proof: Since $\widetilde{\text{MST}}_i$ is given at the outset of the construction of $\mathcal{G}_i$, we only focus on constructing $\mathcal{E}_i$. For each edge $e = (u, v) \in E_i^\sigma$, we add an edge $(\varphi_{C_u}, \varphi_{C_v})$ to $\mathcal{G}_i$. Next, we remove edges from $\mathcal{G}_i$. (Step 1) we remove self-loops and parallel edges from $\mathcal{G}_i$; we only keep the edge of minimum weight in $\mathcal{G}_i$ among parallel edges. (Step 2) If $t \geq 2$, we remove every edge $(\varphi_{C_u}, \varphi_{C_v})$ from $\mathcal{G}_i$ such that $\omega(\widetilde{\text{MST}}_i[\varphi_{C_u}, \varphi_{C_v}]) \leq t(1 + 6g\epsilon)\omega(\varphi_{C_u}, \varphi_{C_v})$; the remaining edges of $\mathcal{G}_i$ not in $\widetilde{\text{MST}}_{i+1}$ are $\mathcal{E}_i$. If $t = 1 + \epsilon$, we apply the **path greedy** algorithm to $\mathcal{G}_i$ with stretch $t(1 + 6g\epsilon)$ to obtain $\mathcal{S}_i$. (Note that we use augmented distances rather than normal distances when apply the greedy algorithm.) It was shown [2] that the **path greedy** algorithm contains the minimum spanning tree of the input graph. Thus, $\mathcal{S}_i$ contains $\widetilde{\text{MST}}_i$ as a subgraph. We then set $\mathcal{E}_i = \mathcal{E}(\mathcal{S}_i) \setminus \widetilde{\text{MST}}_i$; this completes the construction of $\mathcal{G}_i$.

We now show the second claim: the stretch of E_i^σ in $H_{<L_i}$ is $t(1 + \max\{s + 4g, 10g\}\epsilon)$. Let (u', v') be any edge in $E_i^\sigma \setminus F_i^\sigma$. Recall that (u', v') is not in F_i^σ because (a) both u' and v' are in the same level- i cluster in the construction of the cluster graph in Lemma 10.2, or (b) (u', v') is parallel with another edge (u, v) , or (c) the edge $(\varphi_{C_{u'}}, \varphi_{C_{v'}})$ corresponding to (u', v') is removed from $\mathcal{G}_i$ in Step 2.

Case (a) does not happen since otherwise, there is a path in $H_{<L_i}$ of length at most $gL_{i-1} = g\epsilon L_i \leq \frac{L_i}{1+\psi} \leq w(u', v')$ when $\epsilon < \frac{1}{(1+\psi)g}$, contradicting that every edge is a shortest path between its endpoints.

For case (c), observe that if $t \geq 2$, then by construction, $d_{H_{<L_{i-1}}}(u', v') \leq t(1+6g\epsilon)w(u', v')$. Otherwise ($t = 1 + \epsilon$), let $\mathcal{P}'$ be the shortest path between $\varphi_{C_{u'}}$ and $\varphi_{C_{v'}}$ in $\mathcal{S}_i$. Since $\mathcal{S}_i$ is a $t(1+6g\epsilon)$ -spanner of $\mathcal{G}_i$, we have:

$$\omega(\mathcal{P}') \leq (1 + \epsilon)(1 + 6g\epsilon)\omega(\varphi_{C_{u'}}, \varphi_{C_{v'}}) \leq (1 + (12g + 1)\epsilon)w(u', v') \quad (47)$$

Observe that $\mathcal{P}'$ contains at most one edge in $\mathcal{E}_i$. Let P' be a realization of $\mathcal{P}'$ w.r.t u' and v' . If $\mathcal{P}'$ contains no edge in $\mathcal{E}_i$, then P' is a path in $H_{<L_{i-1}}$. This implies that $d_{H_{\leq i}}(u', v') \leq (1 + (12g + 1)\epsilon)w(u', v') \leq t(1 + (12g + 1)\epsilon)w(u', v')$ since $t \geq 1$. Otherwise, P' contains exactly one edge $(x, y) \in F_i^\sigma$. Let Q' be obtained from P' by replacing edge (x, y) by a shortest path from x to y in $H_{<L_i}$. Since $d_{H_{<L_i}}(x, y) \leq t(1 + s \cdot \epsilon)w(x, y)$. Then by Equation (47), we have:

$$w(Q') \leq t(1 + s \cdot \epsilon)w(P') \leq t(1 + (2s + 12g + 1) \cdot \epsilon)w(u', v') \quad (\text{since } (12g + 1)\epsilon \leq 1)$$

Thus, in all cases, $d_{H_i}(u', v') \leq t(1 + (2s + 12g + 1) \cdot \epsilon)w(u', v')$.

We now consider case (b); that is, (u', v') is not in F_i^σ because it is parallel with another edge (u, v) . Let C_u and C_v be two level- i clusters containing u and v , respectively. W.l.o.g, we assume that $u' \in C_u$ and $v' \in C_v$. Since we only keep the edge of minimum weight among all parallel edges, $w(u, v) \leq w(u', v')$. By property (P3), $\text{Dm}(H_{<L_i}[C_u]), \text{Dm}(H_{<L_i}[C_v]) \leq g\epsilon L_i$.

$$\begin{aligned} d_{H_{<L_i}}(u', v') &\leq d_{H_{<L_i}}(u, v) + \text{Dm}(H_{<L_i}[C_u]) + \text{Dm}(H_{<L_i}[C_v]) \\ &\leq t(1 + (2s + 12g + 1)\epsilon)w(u, v) + 2g\epsilon L_i \\ &\leq (1 + (2s + 16g + 1)\epsilon)w(u', v') \quad (\text{since } t \geq 1), \end{aligned}$$

Since $w(u', v') \geq L_i/(1 + \psi) \geq L_i/2$. □

To construct the set of subgraphs $\mathbb{X}$ of $\mathcal{G}_i$, we distinguish between two cases: (a) $t = 1 + \epsilon$ and (b) $t \geq 2$. Subgraphs in $\mathbb{X}$ constructed for the case $t = 1 + \epsilon$ have properties similar to those of subgraphs constructed in Section 7; the key difference is that subgraphs constructed in our work have a larger *average potential change*, which ultimately leads to an optimal dependency on ϵ of the lightness. When the stretch $t \geq 2$, we show that one can construct a set of subgraphs $\mathbb{X}$ of $\mathcal{G}_i$ with much larger potential change, which reduces the dependency of the lightness on ϵ by a factor $1/\epsilon$ compared to the case $t = 1 + \epsilon$. Our construction uses SSO as a black box. The following lemma summarizes our construction.

Lemma 10.3. *Given SSO, we can construct in polynomial time a set of subgraphs $\mathbb{X}$ such that every subgraph $\mathcal{X} \in \mathbb{X}$ satisfies the three properties (P1')-(P3') with constant $g = 223$, and graph H_i such that:*

$$d_{H_{<L_i}}(u, v) \leq t(1 + \max\{\text{SSO}(2g), 6g\}\epsilon)w(u, v) \quad \forall (u, v) \in F_i^\sigma$$

where F_i^σ is the set of edges defined in Lemma 10.2. Furthermore, $w(H_i) \leq \lambda\Delta_{i+1} + a_i$ such that

1. **when** $t \geq 2$: $\lambda = O(\chi\epsilon^{-1})$, and $A = O(\chi\epsilon^{-1})$.
2. **when** $t = 1 + \epsilon$: $\lambda = O(\chi\epsilon^{-1} + \epsilon^{-2})$, and $A = O(\chi\epsilon^{-1} + \epsilon^{-2})$.

Here $A \in \mathbb{R}^+$ such that $\sum_{i \in \mathbb{N}^+} a_i \leq A \cdot w(\text{MST})$.

The proof of Lemma 10.3 is deferred to Section 11 for the case $t \geq 2$ and Section 12 for the case $t = 1 + \epsilon$.

Constructing $\widetilde{\text{MST}}_{i+1}$. Let $\widetilde{\text{MST}}_i^{\text{out}} = \widetilde{\text{MST}}_i \setminus (\cup_{\mathcal{X} \in \mathbb{X}} (\mathcal{E}(\mathcal{X}) \cap \widetilde{\text{MST}}_i))$ be the set of $\widetilde{\text{MST}}_i$ edges that are not contained in any subgraph $\mathcal{X} \in \mathbb{X}$. Let $\widetilde{\text{MST}}_{i+1}'$ be the graph with vertex set $\mathcal{V}_{i+1}$ and there is an edge between two nodes $(\mathcal{X}, \mathcal{Y})$ in $\mathcal{V}_{i+1}$ if there is at least one edge in $\widetilde{\text{MST}}_i^{\text{out}}$ between two nodes in the two corresponding subgraphs $\mathcal{X}$ and $\mathcal{Y}$. Note that $\widetilde{\text{MST}}_{i+1}'$ could have parallel edges (but no self-loop). Since $\widetilde{\text{MST}}_i$ is a spanning tree of $\mathcal{G}_i$, $\widetilde{\text{MST}}_{i+1}'$ must be connected. $\widetilde{\text{MST}}_{i+1}$ is then a spanning tree of $\widetilde{\text{MST}}_{i+1}'$.

We are now ready to prove Item (2) of Theorem 1.10.

Proof: [Proof of Item (2) of Theorem 1.10] We apply Lemma 6.8 to construct a light spanner H for G where each graph H_i , $i \in \mathbb{N}^+$, is constructed using Lemma 10.3.

When $t \geq 2$, by Item (1) of Lemma 10.3 and Lemma 6.8, the lightness of H is $O((\frac{O(\chi\epsilon^{-1})+O(\chi\epsilon^{-1})+1}{1+\psi})\log(\frac{1}{\epsilon})+\frac{1}{\epsilon}) = O_\epsilon(\chi\epsilon^{-1})$. When $t = 1 + \epsilon$, by Item (2) of Lemma 10.3 and Lemma 6.8, the lightness of H is $O((\frac{O(\chi\epsilon^{-1})+O(\chi\epsilon^{-1})+1}{1+\psi})\log(\frac{1}{\epsilon})+\frac{1}{\epsilon^2}) = O_\epsilon(\chi\epsilon^{-1} + \epsilon^{-2})$.

We now bound the stretch of H . By Lemma 10.3 and Lemma 10.2, the stretch of edges in E_i^σ in the graph $H_{<L_i}$ is $t(1 + (2s_{\text{SSO}}(2g) + 16g + 1)\epsilon)$ with $g = 223$. Thus, by Lemma 6.8, the stretch of H is $t(1 + (2s_{\text{SSO}}(2g) + 16g + 1)\epsilon) = t(1 + (2s_{\text{SSO}}(O(1)) + O(1))\epsilon)$ as claimed. $\square$

11 Clustering for Stretch $t \geq 2$: Proof of Lemma 10.3(1)

In this section, we prove Item (1) of Lemma 10.3 (when the stretch t is at least 2). The general idea is to construct a set $\mathbb{X}$ of subgraphs of $\mathcal{G}_i$ such that each subgraph in $\mathbb{X}$ has a sufficiently large local potential change, and carefully choose a subset of edges of $\mathcal{G}_i$, with the help from SSO, such that the total weight could be bounded by the potential change of subgraphs in $\mathbb{X}$ and distances between endpoints of edges in $\mathcal{E}_i$ are preserved. (By Lemma 7.2, it is sufficient to preserve distances between the endpoints of edges in $\mathcal{E}_i$.) In Lemma 11.1 below, we state desirable properties of subgraphs in $\mathbb{X}$. Recall that $H_{<L_{i-1}}$ is the spanner constructed for edges of G of weight less than L_{i-1} .

Lemma 11.1. *Let $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$ be the cluster graph. We can construct in polynomial time (i) a collection $\mathbb{X}$ of subgraphs of $\mathcal{G}_i$ and its partition into two sets $\{\mathbb{X}^+, \mathbb{X}^-\}$ and (ii) a partition of $\mathcal{E}_i$ into three sets $\{\mathcal{E}_i^{\text{take}}, \mathcal{E}_i^{\text{reduce}}, \mathcal{E}_i^{\text{redund}}\}$ such that:*

- (1) *For every subgraph $\mathcal{X} \in \mathbb{X}$, $\deg_{\mathcal{G}_i^{\text{take}}}(\mathcal{V}(\mathcal{X})) = O(|\mathcal{V}(\mathcal{X})|)$ where $\mathcal{G}_i^{\text{take}} = (\mathcal{V}_i, \mathcal{E}_i^{\text{take}})$, and $\mathcal{E}(\mathcal{X}) \cap \mathcal{E}_i \subseteq \mathcal{E}_i^{\text{take}}$. Furthermore, if $\mathcal{X} \in \mathbb{X}^-$, there is no edge in $\mathcal{E}_i^{\text{reduce}}$ incident to a node in $\mathcal{X}$.*
- (2) *Let $H_{<L_i}^-$ be a subgraph obtained by adding corresponding edges of $\mathcal{E}_i^{\text{take}}$ to $H_{<L_{i-1}}$. Then for every edge (u, v) that corresponds to an edge in $\mathcal{E}_i^{\text{redund}}$, $d_{H_{<L_i}^-}(u, v) \leq 2d_G(u, v)$.*
- (3) *Let $\Delta_{i+1}^+(\mathcal{X}) = \Delta(\mathcal{X}) + \sum_{\mathbf{e} \in \widetilde{\text{MST}}_i \cap \mathcal{E}(\mathcal{X})} w(\mathbf{e})$ be the corrected potential change of $\mathcal{X}$. Then, $\Delta_{i+1}^+(\mathcal{X}) \geq 0$ for every $\mathcal{X} \in \mathbb{X}$ and*

$$\sum_{\mathcal{X} \in \mathbb{X}^+} \Delta_{i+1}^+(\mathcal{X}) = \sum_{\mathcal{X} \in \mathbb{X}^+} \Omega(|\mathcal{V}(\mathcal{X})|\epsilon L_i). \quad (48)$$
- (4) *For every edge $(\varphi_1, \varphi_2) \in \mathcal{E}_i$ such that $\varphi_1 \in \mathcal{X}, \varphi_2 \in \mathcal{Y}$ for some subgraphs $\mathcal{X}, \mathcal{Y} \in \mathbb{X}^-$, then $(\varphi_1, \varphi_2) \in \mathcal{E}_i^{\text{redund}}$, unless a degenerate case happens, in which $\mathcal{E}_i^{\text{reduce}} = \emptyset$ and $\mathcal{E}_i^{\text{take}} = O(\frac{1}{\epsilon})$.*
- (5) *For every subgraph $\mathcal{X} \in \mathbb{X}$, $\mathcal{X}$ satisfies the three properties (P1')-(P3') with constant $g = 223$. Furthermore, if $\mathcal{X} \in \mathbb{X}^-$, then $|\mathcal{E}(\mathcal{X}) \cap \mathcal{E}_i| = 0$.*

Lemma 11.1 is analogous to Lemma 7.3. Here we point out two major differences, which ultimately lead to the optimal dependency on ϵ of the lightness. In Lemma 7.3, roughly $O(1/\epsilon)$ edges are added to H_i per node of $\mathcal{V}_i$. Furthermore, each node has $\Omega(L_i\epsilon^2)$ average potential change. These two facts together incur a factor of $\Omega(1/\epsilon^3)$ in the lightness. Another factor of $1/\epsilon$ is due to $\psi = \epsilon$ for the purpose of obtaining a fast construction. The overall lightness has a factor of $1/\epsilon^4$ dependency on ϵ . Our goal is to reduce this dependency all the way down to $1/\epsilon$. By choosing $\psi = 1/250$, we already eliminate one factor of $1/\epsilon$. By carefully partitioning $\mathcal{E}_i$ into three set of edges $\{\mathcal{E}_i^{\text{take}}, \mathcal{E}_i^{\text{reduce}}, \mathcal{E}_i^{\text{redunt}}\}$, and only taking edges of $\mathcal{E}_i^{\text{take}}$ to H_i , we essentially reduce the number of edges we take per node in every subgraph $\mathcal{X}$ from $O(1/\epsilon)$ to $O(1)$ (by Item (1) in Lemma 11.1), thereby saving another factor of $1/\epsilon$. Finally, we show that (by Item (3) in Lemma 11.1), each node in $\mathbb{X}^+$ has $\Omega(L_i\epsilon)$ average potential change, which is larger than the average potential change of nodes in Lemma 7.3 by a factor of $1/\epsilon$. We crucially use the fact that $t \geq 2$ in bounding the average potential change here. All of these ideas together reduce the dependency on ϵ from $1/\epsilon^4$ to $1/\epsilon$ as desired.

Next we show to construct H_i given that we can construct a set of subgraphs $\mathbb{X}$ as claimed in Lemma 11.1. The proof of Lemma 11.1 is deferred to Section 11.2.

11.1 Constructing H_i : Proof of Lemma 10.3 for $t \geq 2$.

In this section, we construct graph H_i as described in Lemma 10.3 in two steps. In Step 1, we take every edge in $\mathcal{E}_i^{\text{take}}$ to H_i . In Step 2, we use SSO to construct a subset of edges F to provide a good stretch for edges in $\mathcal{E}_i^{\text{reduce}}$. Note that edges in F may not correspond to edges in $\mathcal{E}_i^{\text{reduce}}$. As the implementation of SSO depends on the input graph, this is the only place in our framework where the structure of the input graph plays an important role in the construction of the light spanner.

Constructing H_i : We construct H_i in two steps; initially H_i contains no edges.

- **(Step 1).** We add to H_i every edge of E_i^σ corresponding to an edge in $\mathcal{E}_i^{\text{take}}$.
- **(Step 2).** Let $\mathcal{J}_i$ be a subgraph of $\mathcal{G}_i$ induced by $\mathcal{E}_i^{\text{reduce}}$. Observe that $\mathcal{J}_i$ is a $(L_i/(1 + \psi), \epsilon, \beta, \Upsilon = 2)$ -cluster graph w.r.t $H_{<L_{i-1}}$. We run SSO on $\mathcal{J}_i$ to obtain a set of edges F . We then add every edge in F to H_i .

Analysis. Recall that F_i^σ is the set of edges in E_i^σ that correspond to $\mathcal{E}_i$.

Lemma 11.2. For every edge $(u, v) \in F_i^\sigma$, $d_{H_{<L_i}}(u, v) \leq t(1 + s_{\text{SSO}}(2g)\epsilon)w(u, v)$.

Proof: By construction, edges in F_i^σ that correspond to $\mathcal{E}_i^{\text{take}}$ are added to H_i and hence have stretch 1. By Item (2) of Lemma 11.1, edges in F_i^σ that correspond to $\mathcal{E}_i^{\text{redunt}}$ have stretch $2 \leq t$ in $H_{<L_i}$. Thus, it remains to focus on edges corresponding to $\mathcal{E}_i^{\text{reduce}}$. Let $(\varphi_{C_u}, \varphi_{C_v}) \in \mathcal{E}_i^{\text{reduce}}$ be the edge corresponding to an edge $(u, v \in F_i^\sigma)$. Since we add all edges of F to H_i , by property (2) of SSO, the stretch of edge (u, v) in $H_{<L_i}$ is at most $t(1 + s_{\text{SSO}}(\beta)\epsilon) = t(1 + s_{\text{SSO}}(2g)\epsilon)$ since $\beta = 2g$. $\square$

Let $\widetilde{\text{MST}}_i^{\text{in}}(\mathcal{X}) = \mathcal{E}(\mathcal{X}) \cap \widetilde{\text{MST}}_i$ for each $\mathcal{X} \in \mathbb{X}$. Let $\widetilde{\text{MST}}_i^{\text{in}} = \cup_{\mathcal{X} \in \mathbb{X}} (\mathcal{E}(\mathcal{X}) \cap \widetilde{\text{MST}}_i)$ be the set of $\widetilde{\text{MST}}_i$ edges that are contained in subgraphs in $\mathbb{X}$. We have the following observations.

Observation 11.3. (1) $\sum_{\mathcal{X} \in \mathbb{X}} \Delta_{i+1}^+(\mathcal{X}) = (\Delta_{i+1} + w(\widetilde{\text{MST}}_i^{\text{in}}))$. Furthermore, $(\Delta_{i+1} + w(\widetilde{\text{MST}}_i^{\text{in}})) \geq 0$.
(2) $\sum_{i \in \mathbb{N}^+} \widetilde{\text{MST}}_i^{\text{in}} \leq w(\text{MST})$.

Lemma 11.4. $w(H_i) \leq \lambda \Delta_{i+1} + a_i$ for $\lambda = O(\chi\epsilon^{-1})$ and $a_i = O(\chi\epsilon^{-1})w(\widetilde{\text{MST}}_i^{\text{in}}) + O(L_i/\epsilon)$.

Proof: First, we consider the non-degenerate case. Note by the construction of H_i that we do not add any edge corresponding to an edge in $\mathcal{E}_i^{\text{redund}}$ to H_i . Thus, we only need to consider edges in $\mathcal{E}_i^{\text{take}} \cup \mathcal{E}_i^{\text{reduce}}$. Let $\mathcal{V}_i^+ = \cup_{\mathcal{X} \in \mathbb{X}^+} \mathcal{V}(\mathcal{X})$ and $\mathcal{V}_i^- = \cup_{\mathcal{X} \in \mathbb{X}^-} \mathcal{V}(\mathcal{X})$. By Observation 11.3, any edge in $\mathcal{E}_i^{\text{take}}$ incident to a node in $\mathcal{V}_i^-$ is also incident to a node in $\mathcal{V}_i^+$. Let $F_i^{(a)}$ be the set of edges added to H_i in the construction in Step a , $a \in \{1, 2\}$.

By Item (3) of Observation 11.3, $\mathcal{E}(\mathcal{X}) \cap \mathcal{E}_i = \emptyset$ if $\mathcal{X} \in \mathbb{X}^-$. By the construction in Step 1, $F_i^{(1)}$ includes edges in E_i^σ corresponding to $\mathcal{E}_i^{\text{take}}$. By Item (1) in Lemma 11.1, the total weight of the edges added to H_i in Step 1 is:

$$w(F_i^{(1)}) = \sum_{\mathcal{X} \in \mathbb{X}^+} O(|\mathcal{V}(\mathcal{X})|) L_i \stackrel{\text{Eq. (48)}}{=} O\left(\frac{1}{\epsilon}\right) \sum_{\mathcal{X} \in \mathbb{X}^+} \Delta_{i+1}^+(\mathcal{X}) = O\left(\frac{1}{\epsilon}\right) (\Delta_{i+1} + w(\widetilde{\text{MST}}_i^{\text{in}})). \quad (49)$$

Next, we bound $w(F_i^{(2)})$. By Item (1) of Lemma 11.1, there is no edge in $\mathcal{E}_i^{\text{reduce}}$ incident to a node in $\mathcal{V}_i^-$. Thus, $\mathcal{V}(\mathcal{J}_i) \subseteq \mathcal{V}_i^+$. By property (1) of SSO, it follows that

$$w(F_i^{(2)}) \leq \chi |\mathcal{V}(\mathcal{J}_i)| L_i \leq \chi |\mathcal{V}_i^+| L_i = \chi \sum_{\mathcal{X} \in \mathbb{X}^+} |\mathcal{V}(\mathcal{X})| L_i = O(\chi/\epsilon) (\Delta_{i+1} + w(\widetilde{\text{MST}}_i^{\text{in}})). \quad (50)$$

By Equations (49) and (50), we conclude that:

$$w(H_i) = O(\chi/\epsilon) (\Delta_{i+1} + w(\widetilde{\text{MST}}_i^{\text{in}})) \leq \lambda (\Delta_{i+1} + w(\widetilde{\text{MST}}_i^{\text{in}})) \quad (51)$$

for some $\lambda = O(\chi/\epsilon)$.

It remains to consider the degenerate case. By Item (4) of Lemma 11.1, we only add to H_i edges corresponding to $\mathcal{E}_i^{\text{take}}$, and there are $O(1/\epsilon)$ such edges. Thus, we have:

$$w(H_i) = O\left(\frac{L_i}{\epsilon}\right) \leq \lambda \cdot (\Delta_{i+1} + w(\widetilde{\text{MST}}_i^{\text{in}})) + O\left(\frac{L_i}{\epsilon}\right), \quad (52)$$

since $\Delta_{i+1} + w(\widetilde{\text{MST}}_i^{\text{in}}) \geq 0$ by Item (1) in Observation 11.3. Thus, the lemma follows from Equations (51) and (52). $\square$

We are now ready to prove Lemma 10.3 for the case $t \geq 2$, which we restate below.

Lemma 10.3. *Given SSO, we can construct in polynomial time a set of subgraphs $\mathbb{X}$ such that every subgraph $\mathcal{X} \in \mathbb{X}$ satisfies the three properties (P1')-(P3') with constant $g = 223$, and graph H_i such that:*

$$d_{H_{<L_i}}(u, v) \leq t(1 + \max\{s_{\text{SSO}}(2g), 6g\}\epsilon) w(u, v) \quad \forall (u, v) \in F_i^\sigma$$

where F_i^σ is the set of edges defined in Lemma 10.2. Furthermore, $w(H_i) \leq \lambda \Delta_{i+1} + a_i$ such that

1. **when** $t \geq 2$: $\lambda = O(\chi\epsilon^{-1})$, and $A = O(\chi\epsilon^{-1})$.
2. **when** $t = 1 + \epsilon$: $\lambda = O(\chi\epsilon^{-1} + \epsilon^{-2})$, and $A = O(\chi\epsilon^{-1} + \epsilon^{-2})$.

Here $A \in \mathbb{R}^+$ such that $\sum_{i \in \mathbb{N}^+} a_i \leq A \cdot w(\text{MST})$.

Proof: [Proof of Item 1.] The fact that subgraphs in $\mathbb{X}$ satisfy the three properties (P1')-(P3') with constant $g = 223$ follows from Item (5) of Lemma 11.1. The stretch in $H_{<L_i}$ of edges in F_i^σ follows from Lemma 11.2.

By Lemma 11.4, $w(H_i) \leq \lambda \Delta_{i+1} + a_i$ where $\lambda = O(\chi \epsilon^{-1})$ and $a_i = O(\chi \epsilon^{-1})w(\widetilde{\text{MST}}_i^{\text{in}}) + O(L_i/\epsilon)$. It remains to show that $A = \sum_{i \in \mathbb{N}^+} a_i = O(\chi \epsilon^{-1})$. Observe that

$$\sum_{i \in \mathbb{N}^+} O\left(\frac{L_i}{\epsilon}\right) = O\left(\frac{1}{\epsilon}\right) \sum_{i=1}^{i_{\max}} \frac{L_{i_{\max}}}{\epsilon^{i_{\max}-i}} = O\left(\frac{L_{i_{\max}}}{\epsilon(1-\epsilon)}\right) = O\left(\frac{1}{\epsilon}\right)w(\text{MST}) ;$$

here $i_{\max}$ is the maximum level. The last equation is due to that $\epsilon \leq 1/2$ and every edge has weight at most $w(\text{MST})$ since the weight of every is the shortest distance between its endpoints. By Item (2) of Observation 11.3, $\sum_{i \in \mathbb{N}^+} \widetilde{\text{MST}}_i^{\text{in}} \leq w(\text{MST})$. Thus, $A = O(\chi/\epsilon) + O(1/\epsilon) = O(\chi/\epsilon)$ as desired. $\square$

11.2 Clustering

In this section, we give a construction of the set of subgraphs $\mathbb{X}$ of the cluster graph $\mathcal{G}_i$ as claimed in Lemma 11.1. Our construction builds on the construction in Section 8. However, there are two specific goals we would like to achieve: the total degree of nodes in each subgraph $\mathcal{X}$ in $\mathcal{G}_i^{\text{take}}$ is $O(|\mathcal{V}(\mathcal{X})|)$, and the average potential change of each node (up to some edge cases) is $\Omega(\epsilon L_i)$ (instead of $\Omega(\epsilon^2 L_i)$ as achieved in Section 8),

Our construction has 6 main steps (Steps 1-6). The first five steps are similar to the first five steps in the construction in Section 8. The major differences are in Step 2 and Step 4. In particular, in Step 2, we need to apply a clustering procedure of [50] to guarantee that the formed clusters have large average potential change. In Step 4, by using the fact that the stretch is at least 2, we form subgraphs in such a way that the potential change of the formed subgraphs is large. Step 6 is new in this paper. The idea is to post-process clusters formed in Steps 1-5 to form larger subgraphs that are trees, and hence, the average degree of nodes is $O(1)$. For those that are not grouped in the larger subgraphs, the total degree of the nodes in each subgraph is $O(1/\epsilon)$, which is at most the number of nodes. In this step, we also rely on the fact that the stretch $t \geq 2$.

Now we give the details of the construction. Recall that g is a constant defined in property (P3) ($g = 223$ in Lemma 11.1), and that $\widetilde{\text{MST}}_i$ is a spanning tree of $\mathcal{G}_i$ by Item (2) in Definition 6.15. We reuse the construction in Lemma 8.1 for Step 1 which applies to the subgraph $\mathcal{K}_i$ of $\mathcal{G}_i$ with edges in $\mathcal{E}_i$, as described by the following lemma.

Lemma 11.5. *Let $\mathcal{V}_i^{\text{high}} = \{\varphi_C \in \mathcal{V} : \varphi_C \text{ is incident to at least } \frac{2g}{\zeta_\epsilon} \text{ edges in } \mathcal{E}_i\}$. Let $\mathcal{V}_i^{\text{high}+}$ be obtained from $\mathcal{V}_i^{\text{high}}$ by adding all neighbors that are connected to nodes in $\mathcal{V}_i^{\text{high}}$ via edges in $\mathcal{E}_i$. We can construct in polynomial time a collection of node-disjoint subgraphs $\mathbb{X}_1$ of $\mathcal{K}_i = (\mathcal{V}_i, \mathcal{E}_i)$ such that:*

- (1) *Each subgraph $\mathcal{X} \in \mathbb{X}_1$ is a tree.*
- (2) *$\cup_{\mathcal{X} \in \mathbb{X}_1} \mathcal{V}(\mathcal{X}) = \mathcal{V}_i^{\text{high}+}$.*
- (3) *$L_i \leq \text{Adm}(\mathcal{X}) \leq (6 + 7\eta)L_i$, assuming that every node of $\mathcal{V}_i$ has weight at most ηL_i .*
- (4) *$\mathcal{X}$ contains a node in $\mathcal{V}_i^{\text{high}}$ and all of its neighbors in $\mathcal{K}_i$. In particular, this implies $|\mathcal{V}(\mathcal{X})| \geq \frac{2g}{\zeta_\epsilon}$.*

We note Lemma 11.5 is slightly more general than Lemma 8.1 in that we parameterize the weights of nodes in $\mathcal{V}_i$ by ηL_i . Clearly, we can choose $\eta = g\epsilon \leq 1$ when $\epsilon \leq 1/g$ since every node in $\mathcal{V}_i$ has a weight at most $g\epsilon L_i$ by property (P3') for level $i-1$. By parameterizing the weights, it will be more convenient for us to use the same construction again in Step 6 below.

Given a tree T , we say that a node $x \in T$ is *T-branching* if it has degree at least 3 in T . For brevity, we shall omit the prefix T in “ T -branching” whenever this does not lead to confusion. Given a forest F , we say that x is *F-branching* if it is T -branching for some tree $T \subseteq F$. Our construction of Step 2 uses the following lemma by [50].

Lemma 11.6 (Lemma 6.12, full version [50]). *Let $\mathcal{T}$ be a tree with vertex weights and edge weights. Let L, η, γ, β be parameters where $\eta \ll \gamma \ll 1$ and $\beta \geq 1$. Suppose that for any vertex $v \in \mathcal{T}$ and any edge $e \in \mathcal{T}$, $w(e) \leq w(v) \leq \eta L$ and $w(v) \geq \eta L/\beta$. There is a polynomial-time algorithm that finds a collection of vertex-disjoint subtrees $\mathbb{U} = \{\mathcal{T}_1, \dots, \mathcal{T}_k\}$ of $\mathcal{T}$ such that:*

- (1) $\text{Adm}(\mathcal{T}_i) \leq 190\gamma L$ for any $1 \leq i \leq k$.
- (2) Every branching node is contained in some tree in $\mathbb{U}$.
- (3) Each tree $\mathcal{T}_i$ contains a $\mathcal{T}_i$ -branching node b_i and three internally node-disjoint paths $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3$ sharing b_i as the same endpoint, such that $\text{Adm}(\mathcal{P}_1 \cup \mathcal{P}_2) = \text{Adm}(\mathcal{T}_i)$ and $\text{Adm}(\mathcal{P}_3 \setminus \{b_i\}) = \Omega(\text{Adm}(\mathcal{T}_i)/\beta)$. We call b_i the center of $\mathcal{T}_i$.
- (4) Let $\bar{\mathcal{T}}$ be obtained by contracting each subtree of $\mathbb{U}$ into a single node. Then each $\bar{\mathcal{T}}$ -branching node corresponds to a sub-tree of augmented diameter at least γL .

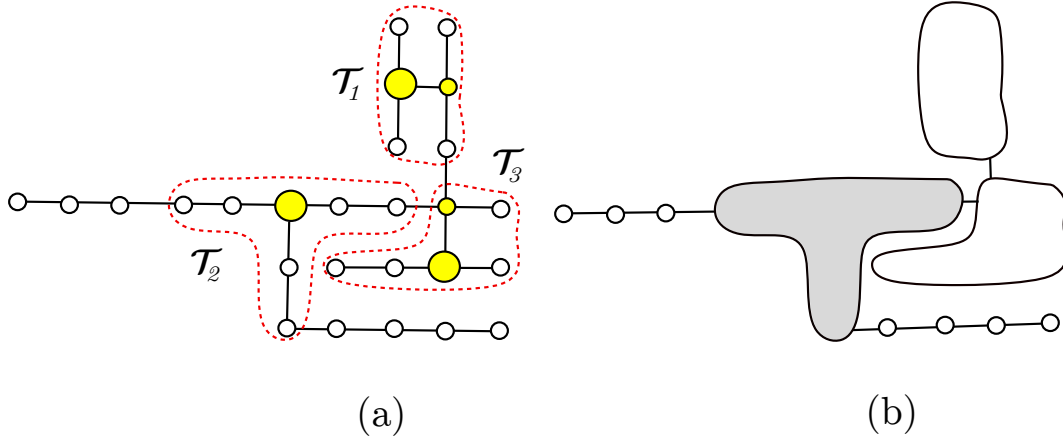


Figure 10: (a) A collection $\mathbb{U} = \{\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3\}$ of a tree $\mathcal{T}$ as in Lemma 11.6. Yellow nodes are $\mathcal{T}$ -branching nodes. Big yellow nodes are the centers of their corresponding subtrees in $\mathbb{U}$. (b) The shaded node in $\bar{\mathcal{T}}$ is a $\bar{\mathcal{T}}$ -branching node and has an augmented diameter of at least γL .

Let $\text{TREECLUSTERING}(\mathcal{T}, L, \eta, \gamma, \beta)$ be the output of Lemma 11.6 for input $\mathcal{T}$ and parameters L, η, γ, β . See an illustration of Lemma 11.6 in Figure 10. We are now ready to describe Step 2. Recall that $\zeta = 1/250$ is the constant in property (P3')

Lemma 11.7 (Step 2). *Let $\tilde{F}_i^{(2)}$ be the forest obtained from $\widetilde{\text{MST}}_i$ by removing every node in $\mathcal{V}_i^{\text{high}+}$ (defined in Lemma 11.5). Let $\mathcal{U} = \cup_{\tilde{T} \in \tilde{F}_i^{(2)}} \text{TREECLUSTERING}(\tilde{T}, L_i, g\epsilon, \zeta, g/\zeta)$ and $\mathbb{X}_2 = \{\tilde{T} \in \mathcal{U} : \text{Adm}(\tilde{T}) \geq \zeta L_i\}$. Then, for every $\mathcal{X} \in \mathbb{X}_2$,*

- (1) $\mathcal{X}$ is a subtree of $\widetilde{\text{MST}}_i$.
- (2) $\zeta L_i \leq \text{Adm}(\mathcal{X}) \leq L_i$.
- (3) $|\mathcal{V}(\mathcal{X})| = \Omega(\frac{1}{\epsilon})$ when $\epsilon \leq 2/g$.
- (4) $\Delta_{i+1}^+(\mathcal{X}) = \Omega(L_i)$.

Furthermore, let $\bar{F}_i^{(3)}$ be obtained from $\tilde{F}_i^{(2)}$ by removing every tree in $\mathbb{U}$ that is added to $\mathbb{X}_2$, and contracting each remaining tree in $\mathbb{U}$ into a single node. Then every tree $\bar{T} \subseteq \bar{F}_i^{(3)}$ is a path.

Proof: We observe that Items (1), (2), and (3) follows directly from the construction. We focus on showing Item (4). Let φ_b be the center node of $\mathcal{X}$. By Item (3) in Lemma 11.6, there are three internally node-disjoint paths $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3$ sharing φ_b as the same endpoint. There must be an least one

path, say $\mathcal{P}_1$, such that $\mathcal{P}_1 \cap \mathcal{D} \subseteq \{\varphi_b\}$. That is, $\mathcal{P}_1$ is internally disjoint from the diameter path $\mathcal{D}$. Also by Item (3) in Lemma 11.6, $\text{Adm}(\mathcal{P}_1 \setminus \{\varphi_b\}) = \Omega(\text{Adm}(\mathcal{X})/\beta) = \Omega(\zeta L_i/(g/\zeta)) = \Omega(L_i)$. Thus, $\Delta_{i+1}^+(\mathcal{X}) \geq \text{Adm}(\mathcal{P}_1 \setminus \{\varphi_b\}) = \Omega(L_i)$, as claimed. $\square$

By Item (4) of Lemma 11.7, the amount of potential change of subgraphs in $\mathbb{X}_2$ is $\Omega(L_i)$, while in subgraphs in $\mathbb{X}_2$ in the construction in Section 8 only have $\Omega(\epsilon L_i)$ potential change.

We note that there might be isolated nodes in $\overline{F}_i^{(3)}$, which we still consider as paths. We refer to nodes in $\overline{F}_i^{(3)}$ that are contracted from $\mathcal{U}$ as *contracted nodes*, and nodes that correspond to original nodes of $\tilde{F}_i^{(2)}$ as *uncontracted nodes*. For each node $\bar{\varphi} \in \overline{F}_i^{(3)}$, we abuse notation by denoting $\bar{\varphi}$ the subtree of $\tilde{F}_i^{(2)}$ corresponding to the node $\bar{\varphi}$; $\bar{\varphi}$ could be a single node in $\tilde{F}_i^{(2)}$ for the uncontracted case. We then define the weight function of $\bar{\varphi}$ as follows:

$$\omega(\bar{\varphi}) = \text{Adm}(\bar{\varphi}) \quad (53)$$

In the RHS of Equation (53), we interpret $\bar{\varphi}$ as a subtree of $\tilde{F}_i^{(2)}$ with weights on nodes and edges.

Observation 11.8. $\omega(\bar{\varphi}) \leq \zeta L_i$ for every node $\bar{\varphi} \in \overline{F}_i^{(3)}$.

For each subpath $\overline{P} \subseteq \overline{F}_i^{(3)}$, let $\tilde{P}^{\text{uctrt}}$ be the subtree of $\widetilde{\text{MST}}_i$ obtained by uncontracting the contracted nodes in $\overline{P}$. We say that a node $\bar{\varphi} \in \overline{F}_i^{(3)}$ is *incident to an edge* $\mathbf{e} \in \widetilde{\text{MST}}_i \cup \mathcal{E}_i$ if one endpoint of $\mathbf{e}$ belongs to $\bar{\varphi}$.

Step 3: Augmenting $\mathbb{X}_1 \cup \mathbb{X}_2$. Let $\overline{F}_i^{(3)}$ be the forest obtained in Item (4b) in Lemma 11.7. Let $\bar{A}$ be the set of all nodes $\bar{\varphi}$ in $\overline{F}_i^{(3)}$ such that there is (at least one) $\widetilde{\text{MST}}_i$ edge $\mathbf{e} = (\varphi_1, \varphi_2)$ between a node $\varphi_1 \in \bar{\varphi}$, and a node $\varphi_2 \in \mathcal{X}$ for some subgraph $\mathcal{X} \in \mathbb{X}_1 \cup \mathbb{X}_2$. Then, for each node $\bar{\varphi} \in \bar{A}$, we augment $\mathcal{X}$ by adding $\bar{\varphi}$ and $\mathbf{e}$ to $\mathcal{X}$.

The following lemma follows directly from the construction.

Lemma 11.9. *The augmentation in Step 3 increases the augmented diameter of each subgraph in $\mathbb{X}_1 \cup \mathbb{X}_2$ by at most $4L_i$ when $\epsilon \leq 1/g$.*

Furthermore, let $\overline{F}_i^{(4)}$ be the forest obtained from $\overline{F}_i^{(3)}$ by removing every node in $\bar{A}$. Then, for every path $\overline{P} \subseteq \overline{F}_i^{(4)}$, at least one endpoint $\bar{\varphi} \in \overline{P}$ has an $\widetilde{\text{MST}}_i$ edge to a subgraph of $\mathbb{X}_1 \cup \mathbb{X}_2$, unless $\mathbb{X}_1 \cup \mathbb{X}_2 = \emptyset$.

Required definitions/preparations for Step 4. Let $\overline{F}_i^{(4)}$ be the forest obtained from $\overline{F}_i^{(3)}$ as described in Lemma 11.9. We call every path of augmented diameter at least $6L_i$ of $\overline{F}_i^{(4)}$ a *long path*. We use red/blue coloring, which is analogous to red/blue coloring in Section 8.

Red/Blue Coloring. Given a path $\overline{P} \subseteq \overline{F}_i^{(4)}$, we color their nodes red or blue. If a node has augmented distance at most L_i from at least one of the path's endpoints, we color it red; otherwise, we color it blue. Observe that each red node belongs to the suffix or prefix of $\overline{P}$; the other nodes are colored blue.

For each blue node $\bar{\nu}$ in a long path $\overline{P}$, we denote by $\bar{I}(\bar{\nu})$ the subpath of $\overline{P}$ containing every node within an augmented distance (in $\overline{P}$) at most $(1 - \psi)L_i$ from $\bar{\nu}$. We call $\bar{I}(\bar{\nu})$ the *interval* of $\bar{\nu}$. Recall that $\psi = 1/250$ is the constant defined in Equation (22).

We define the following set of edges between nodes of $\overline{F}_i^{(4)}$.

$$\bar{\mathcal{E}}_i = \{(\bar{\mu}, \bar{\nu}) \mid \exists \mu \in \bar{\mu}, \nu \in \bar{\nu} \text{ and } (\mu, \nu) \in \mathcal{E}_i\}. \quad (54)$$

We note that there is no edge in $\mathcal{E}_i$ whose nodes belong to the same tree, say $\bar{\mu}$, that corresponds to a node in $\bar{F}_i^{(4)}$, because such an edge, say $\mathbf{e}$, will have weight at most $\omega(\bar{\mu}) \leq \zeta L_i < L_i/2 < \omega(\mathbf{e})$, a contradiction.

Next, we define the weight:

$$\omega(\bar{\mu}, \bar{\nu}) = \min_{\substack{\mu \in \bar{\mu}, \nu \in \bar{\nu} \\ (\mu, \nu) \in \mathcal{E}_i}} \omega(\mu, \nu) \quad (55)$$

That is, the weight of edges $(\bar{\mu}, \bar{\nu})$ is the minimum weight over all edges between two trees $\bar{\mu}$ and $\bar{\nu}$. We then denote (μ, ν) the edge in $\mathcal{E}_i$ corresponding to an edge $(\bar{\mu}, \bar{\nu}) \in \bar{\mathcal{E}}_i$. Next, we define:

$$\begin{aligned} \bar{\mathcal{E}}_i^{far}(\bar{F}^{(4)}) &= \{(\bar{\nu}, \bar{\mu}) \in \bar{\mathcal{E}}_i \mid \text{color}(\bar{\nu}) = \text{color}(\bar{\mu}) = \text{blue and } \bar{I}(\bar{\nu}) \cap \bar{I}(\bar{\mu}) = \emptyset\} \\ \bar{\mathcal{E}}_i^{close}(\bar{F}^{(4)}) &= \{(\bar{\nu}, \bar{\mu}) \in \bar{\mathcal{E}}_i \mid \text{color}(\bar{\nu}) = \text{color}(\bar{\mu}) = \text{blue and } \bar{I}(\bar{\nu}) \cap \bar{I}(\bar{\mu}) \neq \emptyset\} \end{aligned} \quad (56)$$

We note that the definition of $\bar{\mathcal{E}}_i^{far}(\bar{F}^{(4)})$ and $\bar{\mathcal{E}}_i^{close}(\bar{F}^{(4)})$ depends on the underlying forest $\bar{F}^{(4)}$.

Lemma 11.10 (Step 4). *Let $\bar{F}_i^{(4)}$ be the forest obtained from $\bar{F}_i^{(3)}$ as described in Lemma 11.9. We can construct a collection $\mathbb{X}_4$ of subgraphs of $\mathcal{G}_i$ such that every $\mathcal{X} \in \mathbb{X}_4$:*

- (1) $\mathcal{X}$ is a tree and contains a single edge in $\mathcal{E}_i$.
- (2) $L_i \leq \text{Adm}(\mathcal{X}) \leq 5L_i$.
- (3) $|\mathcal{V}(\mathcal{X})| = \Omega(1/\epsilon)$ when $\epsilon \leq 1/8$.
- (4) $\Delta_{i+1}^+(\mathcal{X}) = \Omega(L_i)$.

Let $\bar{F}_i^{(5)}$ be obtained from $\bar{F}_i^{(4)}$ by removing every node whose corresponding tree is contained in subgraphs of $\mathbb{X}_4$. If we apply Red/Blue Coloring to each path of augmented diameter at least $6L_i$ in $\bar{F}_i^{(5)}$, then $\bar{\mathcal{E}}_i^{far}(\bar{F}^{(5)}) = \emptyset$. Furthermore, for every path $\bar{P} \subseteq \bar{F}_i^{(5)}$, at least one endpoint of $\bar{P}$ has an $\widetilde{\text{MST}}_i$ edge to a subgraph of $\mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4$, unless $\mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4 = \emptyset$.

Proof: We only apply the construction to long paths of $\bar{F}_i^{(4)}$; those that have augmented diameter at least $6L_i$. We use the following claim which is analogous to Claim 8.5.

Claim 11.11. *For any blue node ν , it holds that*

- (a) $(2 - 3\zeta - 2\epsilon - 2\psi)L_i \leq \text{Adm}(\bar{\mathcal{I}}(\bar{\nu})) \leq 2(1 - \psi)L_i$.
- (b) Denote by $\bar{\mathcal{I}}_1$ and $\bar{\mathcal{I}}_2$ the two subpaths obtained by removing $\bar{\nu}$ from the path $\bar{\mathcal{I}}(\bar{\nu})$. Each of these subpaths has augmented diameter at least $(1 - 2\zeta - \epsilon - \psi)L_i$.

We now construct $\mathbb{X}_4$, which initially is empty.

- Pick an edge $(\bar{\nu}, \bar{\mu})$ with both blue endpoints and form a subgraph $\bar{\mathcal{X}} = \{(\bar{\nu}, \bar{\mu}) \cup \bar{\mathcal{I}}(\bar{\nu}) \cup \bar{\mathcal{I}}(\bar{\mu})\}$. We remove all nodes in $\bar{\mathcal{I}}(\bar{\nu}) \cup \bar{\mathcal{I}}(\bar{\mu})$ from the path or two paths containing $\bar{\nu}$ and $\bar{\mu}$, update the color of nodes in the new paths to satisfy Red/Blue Coloring. We then uncontract nodes in $\bar{\mathcal{X}}$ to obtain a subgraph $\mathcal{X}$ of $\mathcal{G}_i$, add $\mathcal{X}$ to $\mathbb{X}_4$, and repeat this step until it no longer applies.

Items (1), (2) and (3) follows from the same argument in Lemma 8.4. We only focus on Item (4). Let $\bar{\mathcal{I}}_1, \bar{\mathcal{I}}_2, \bar{\mathcal{I}}_3, \bar{\mathcal{I}}_4$ be four paths obtained from $\bar{\mathcal{I}}(\bar{\mu})$ and $\bar{\mathcal{I}}(\bar{\nu})$ by removing $\bar{\mu}$ and $\bar{\nu}$. Let $\bar{\mathcal{D}}$ be the diameter path of $\bar{\mathcal{X}}$. Then $\bar{\mathcal{D}}$ contains at most 2 paths among the four paths, and possibly contains edge $(\bar{\nu}, \bar{\mu})$ as well. Since each path has augmented diameter at most $2L_i$ and $\omega(\bar{\nu}, \bar{\mu}) \leq L_i$, we have that:

$$\text{Delta}_{i+1}^+(\mathcal{X}) \geq \left(\sum_{\bar{\varphi} \in \bar{\mathcal{X}}} \omega(\bar{\varphi}) + \sum_{\mathbf{e} \in \mathcal{E}(\bar{\mathcal{X}}) \cap \widetilde{\text{MST}}_i} \omega(\mathbf{e}) \right) - \text{Adm}(\bar{\mathcal{D}}) \geq (1 - 8\zeta - 4\epsilon - 4\psi)L_i = \Omega(L_i),$$

when $\epsilon \leq 1/8$. □

As each node has weight at most $g\epsilon L_{i-1}$, we have:

Observation 11.12. *Let $\bar{P} \subseteq \bar{F}_i^{(3)}$ be a path of augmented diameter $\Omega(L_i)$. Then $|\mathcal{V}(\tilde{P}^{\text{uctrt}})| = \Omega(1/\epsilon)$.*

Step 5. Let $\bar{P}$ be a path in $\bar{F}_i^{(5)}$ obtained by Item (5) of Lemma 11.10. We construct two sets of subgraphs, denoted by $\mathbb{X}_5^{\text{intrnl}}$ and $\mathbb{X}_5^{\text{pref}}$, of $\mathcal{G}_i$. The construction is broken into two steps. Step 5A is only applicable when $\mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4 \neq \emptyset$.

- (Step 5A) If $\bar{P}$ has augmented diameter at most $6L_i$, let $\mathbf{e}$ be an $\widetilde{\text{MST}}_i$ edge connecting $\tilde{P}^{\text{uctrt}}$ and a node in some subgraph $\mathcal{X} \in \mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4$; $\mathbf{e}$ exists by Lemma 11.10. We add both $\mathbf{e}$ and $\tilde{P}^{\text{uctrt}}$ to $\mathcal{X}$.
- (Step 5B) Otherwise, the augmented diameter of $\bar{P}$ is at least $6L_i$. In this case, we greedily break $\bar{P}$ into subpaths $\{\bar{Q}_1, \dots, \bar{Q}_k\}$ such that for each $j \in [1, k]$, $\tilde{Q}_j^{\text{uctrt}}$ has augmented diameter at least L_i and at most $2L_i$. If $\bar{Q}_j$ is connected to a node in a subgraph $\mathcal{X} \in \mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4$ via an edge $e \in \widetilde{\text{MST}}_i$, we add $\tilde{Q}_j^{\text{uctrt}}$ and e to $\mathcal{X}$. If $\bar{Q}_j$ contains an endpoint of $\bar{P}$, we add $\tilde{Q}_j^{\text{uctrt}}$ to $\mathbb{X}_5^{\text{pref}}$; otherwise, we add $\tilde{Q}_j^{\text{uctrt}}$ to $\mathbb{X}_5^{\text{intrnl}}$.

In Step 5B, we want $\tilde{Q}_j^{\text{uctrt}}$ to have augmented diameter at least L_i (to satisfy property(P3')) instead of requiring $\text{Adm}(\bar{Q}_j) \geq L_i$ because a lower bound on the augmented diameter of $\bar{Q}_j$ does not translate to a lower bound on the augmented diameter of $\tilde{Q}_j^{\text{uctrt}}$.

Lemma 11.13. *Every subgraph $\mathcal{X} \in \mathbb{X}_5^{\text{intrnl}} \cup \mathbb{X}_5^{\text{pref}}$ satisfies:*

- (1) $\mathcal{X}$ is a subtree of $\widetilde{\text{MST}}_i$.
- (2) $L_i \leq \text{Adm}(\mathcal{X}) \leq 2L_i$.
- (3) $|\mathcal{V}(\mathcal{X})| = \Omega(1/\epsilon)$.

Furthermore, if $\mathcal{X} \in \mathbb{X}_5^{\text{pref}}$, then $\mathcal{X}$ is the uncontraction of a prefix subpath $\bar{Q}$ of a long path $\bar{P}$, and additionally, the (uncontraction of) other suffix $\bar{Q}'$ of $\bar{P}$ is augmented to a subgraph in $\mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4$, unless $\mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4 = \emptyset$.

Proof: Items (1) and (2) follow directly from the construction. Item (3) follows from Observation 11.12. The last claim about subgraphs in $\mathbb{X}_5^{\text{pref}}$ follows from Lemma 11.10. □

Lemma 11.14. *Let $\mathbb{X}' = \mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4 \cup \mathbb{X}_5^{\text{intrnl}} \cup \mathbb{X}_5^{\text{pref}}$. Every node of $\mathcal{V}_i$ is grouped to some subgraph in $\mathbb{X}'$. Furthermore, for every $\mathcal{X} \in \mathbb{X}'$,*

- (1) $\mathcal{X}$ is a tree. Furthermore, if $\mathcal{X} \notin \mathbb{X}_4$, it is a subtree of $\widetilde{\text{MST}}_i$.
- (2) $\zeta L_i \leq \text{Adm}(\mathcal{X}) \leq 31L_i$ when $\epsilon \leq 1/g$.

$$(3) |\mathcal{V}(\mathcal{X})| = \Omega(1/\epsilon).$$

Proof: The fact that every node of $\mathcal{V}_i$ is grouped to some subgraph in $\mathbb{X}'$ follows directly from the construction. Observe that only subgraphs in $\mathbb{X}'$ formed in Step 4 contain edges in $\mathcal{E}_i$, and such subgraphs are trees by Item (1) of Lemma 11.10; this implies Item (1). Item 3 follows directly from Lemmas 11.5, 11.7, 11.10 and 11.13.

We now focus on bounding $\text{Adm}(\mathcal{X})$. The lower bound on $\text{Adm}(\mathcal{X})$ follows directly from Item (3) of Lemma 11.5, Items (2) of Lemmas 11.7, 11.10 and 11.13. For the upper bound, we observe that if $\mathcal{X}$ is formed in Step 1, it could be augmented further in Step 3, and hence, by Item (3) of Lemma 11.5 (here $\eta = g\epsilon$), and Lemma 11.9, $\text{Adm}(\mathcal{X}) \leq (6 + 7g\epsilon)L_i + 4L_i \leq 17L_i$ since $\epsilon \leq 1/g$. By Items (2) of Lemmas 11.7, 11.10 and 11.13, $\text{Adm}(\mathcal{X}) \leq 5L_i$ if $\mathcal{X}$ is not initially formed in Step 1. Furthermore, the augmentation in Step 5A and 5B increases $\text{Adm}(\mathcal{X})$ by at most $2(\bar{w} + 6L_i) \leq 14L_i$. This implies that, in any case, $\text{Adm}(\mathcal{X}) \leq \max\{17L_i, 5L_i\} + 14L_i = 31L_i$. $\square$

Except for subgraphs in $\mathbb{X}_5^{\text{intrnl}} \cup \mathbb{X}_5^{\text{pref}}$, we can show every subgraph $\mathcal{X} \in \mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4$ has large potential change: $\Delta_{i+1}(\mathcal{X}) = \Omega(L_i)$. The last property that we need to complete the proof of Lemma 11.1 is to guarantee that the total degree of vertices in $\mathcal{X} \in \mathbb{X}_2 \cup \mathbb{X}_4 \cup \mathbb{X}_5^{\text{intrnl}} \cup \mathbb{X}_5^{\text{pref}}$ in $\mathcal{G}^{\text{reduce}}$ is $O(1/\epsilon)$ (we have not defined $\mathcal{G}^{\text{reduce}}$ yet). To this end, we need Step 6. The basic idea is that if any subgraph has many out-going edges in $\bar{\mathcal{E}}_i$ (defined in Equation (54)), then we apply the clustering procedure in Step 1 to group it to a larger subgraph.

Required definitions/preparations for Step 6. We construct a graph $\hat{\mathcal{K}}_i(\hat{\mathcal{V}}_i, \hat{\mathcal{E}}_i, \hat{w})$ as follows. Each node $\hat{\varphi}_{\mathcal{X}} \in \hat{\mathcal{V}}_i$ corresponds to a subgraph $\mathcal{X} \in \mathbb{X}'$. We then set $\hat{w}(\hat{\varphi}_{\mathcal{X}}) = \text{Adm}(\mathcal{X})$. There is an edge $(\hat{\varphi}_{\mathcal{X}}, \hat{\varphi}_{\mathcal{Y}}) \in \hat{\mathcal{E}}_i$ between two *different nodes* $\hat{\varphi}_{\mathcal{X}}, \hat{\varphi}_{\mathcal{Y}}$ if there exists an edge $(\varphi_1, \varphi_2) \in \mathcal{E}_i$ between a node $\varphi_1 \in \mathcal{X}$ and a node $\varphi_2 \in \mathcal{Y}$. We set the weight $\hat{w}(\hat{\varphi}_{\mathcal{X}}, \hat{\varphi}_{\mathcal{Y}})$ to be the minimum weight over all edges in $\mathcal{E}_i$ between $\mathcal{X}$ and $\mathcal{Y}$. We call nodes of $\hat{\mathcal{K}}_i$ *supernodes*.

We call $\hat{\varphi}_{\mathcal{X}}$ a *heavy* supernode if $|\mathcal{V}(\mathcal{X})| \geq \frac{2g}{\zeta\epsilon}$ or $\hat{\varphi}_{\mathcal{X}}$ is incident to at least $\frac{2g}{\zeta\epsilon}$ edges in $\hat{\mathcal{K}}_i$. Otherwise, we call $\hat{\varphi}_{\mathcal{X}}$ a *light* supernode. By definition of a heavy supernode and by Item (4) in Lemma 11.5, if $\mathcal{X}$ is formed in Step 1, then $\hat{\varphi}_{\mathcal{X}}$ is a heavy supernode. We then do the following.

We apply the construction in Lemma 11.5 to graph $\hat{\mathcal{K}}_i(\hat{\mathcal{V}}_i, \hat{\mathcal{E}}_i, \hat{w})$, where $\hat{\mathcal{V}}_i^{\text{high}}$ is the set of heavy supernodes in $\hat{\mathcal{K}}$ and $\hat{\mathcal{V}}_i^{\text{high}+}$ is obtained from $\hat{\mathcal{V}}_i^{\text{high}}$ by adding neighbors in $\hat{\mathcal{K}}_i$. Let $\hat{\mathbb{X}}_6$ be the set of subgraphs of $\hat{\mathcal{K}}_i(\hat{\mathcal{V}}_i, \hat{\mathcal{E}}_i, \hat{w})$ obtained by the construction. Every subgraph $\hat{\mathcal{X}} \in \hat{\mathbb{X}}_6$ satisfies all properties in Lemma 11.5 with $\eta = 31$.

Let $\mathbb{X}_6$ be obtained from $\hat{\mathbb{X}}_6$ by uncontracting supernodes. This completes our Step 6.

By the construction and a simple calculation, we have:

Lemma 11.15. *Every subgraph $\mathcal{X} \in \mathbb{X}_6$ has $\zeta L_i \leq \text{Adm}(\mathcal{X}) \leq 223L_i$.*

In Section 11.2.1 we construct the set of subgraphs $\mathbb{X}$, and show several properties of subgraphs in $\mathbb{X}$. In Section 11.2.2, we construct a partition of $\mathcal{E}_i$ into three sets, and prove Lemma 11.1.

11.2.1 Constructing $\mathbb{X}$

For each $i \in \{2, 4, 5\}$ let $\mathbb{X}_i^-$ be obtained from $\mathbb{X}_i$ by removing subgraphs corresponding to nodes in $\hat{\mathcal{V}}_i^{\text{high}+}$ (which then form subgraphs in $\mathbb{X}_6$). We now define $\mathbb{X}$ and a partition of $\mathbb{X}$ into two sets $\mathbb{X}^+$ and $\mathbb{X}^-$ as claimed in Lemma 11.1. We distinguish two cases:

Degenerate Case. The degenerate case is the case where $\mathbb{X}_1^- \cup \mathbb{X}_2^- \cup \mathbb{X}_4^- = \mathbb{X}_5^{\text{intrnl}} = \emptyset$. In this case, we set $\mathbb{X} = \mathbb{X}^- = \mathbb{X}_5^{\text{intrnl}} \cup \mathbb{X}_5^{\text{pref}}$, and $\mathbb{X}^+ = \emptyset$.

Non-degenerate case. If $\mathbb{X}_1^- \cup \mathbb{X}_2^- \cup \mathbb{X}_4^- = \mathbb{X}_6 \neq \emptyset$, we call this the non-degenerate case. In this case, we define.

$$\begin{aligned}\mathbb{X}^+ &= \mathbb{X}_2^- \cup \mathbb{X}_4^- \cup \mathbb{X}_5^{\text{pref}-} \cup \mathbb{X}_6, & \mathbb{X}^- &= \mathbb{X}_5^{\text{intrnl}-} \\ \mathbb{X} &= \mathbb{X}^+ \cup \mathbb{X}^-\end{aligned}\tag{57}$$

We note that every subgraph in $\mathbb{X}_1$ corresponds to a heavy supernode in $\widehat{\mathcal{K}}_i$ and hence, it will be grouped in some subgraph in $\mathbb{X}_6$.

In the analysis below, we only explicitly distinguish the degenerate case from the non-degenerate case when it is necessary, i.e, in the proof Item (4) of Lemma 11.1. Otherwise, which case we are in is either implicit from the context, or does not matter.

Lemma 11.16. *Let $\mathbb{X}$ be the subgraph as defined in Equation (57). For every subgraph $\mathcal{X} \in \mathbb{X}$, $\mathcal{X}$ is a tree and satisfies the three properties (P1')-(P3') with $g = 223$. Consequently, Item (5) of Lemma 11.1 holds.*

Proof: We observe that property (P1') follows directly from the construction. Property (P2') follows from Item (3) of Lemma 11.14. Property (P3') follows from Lemma 11.15.

By Item (1) of Lemma 11.14, every subgraph $\mathcal{X} \in \mathbb{X}'$ is a tree. Since subgraphs in $\widehat{\mathcal{X}}_6$ in the construction of Step 6 are trees, subgraphs in $\mathbb{X}$ are also trees. Thus, $|\mathcal{E}(\mathcal{X}) \cap \mathcal{E}_i| = O(|\mathcal{V}(\mathcal{X})|)$. Furthermore, if $\mathcal{X} \in \mathbb{X}^-$, by the definition $\mathbb{X}^-$, $\mathcal{X} \notin \mathbb{X}_4$. Thus, $\mathcal{X}$ is a subtree of $\widehat{\text{MST}}_i$ by Item (1) of Lemma 11.14. That implies $\mathcal{E}(\mathcal{X}) \cap \mathcal{E}_i = \emptyset$, which implies Item (5) of Lemma 11.1. $\square$

Our next goal is to show Item (3) of Lemma 11.1. Lemma 11.17 below implies that if $\mathcal{X} \in \mathbb{X}$ is formed in Steps 2,4, and 6, then $\Delta_{i+1}^+(\mathcal{X}) = \Omega(\epsilon L_i |\mathcal{V}(\mathcal{X})|)$.

Lemma 11.17. *For any subgraph $\mathcal{X} \in \mathbb{X}$ such that $|\mathcal{V}(\mathcal{X})| \geq \frac{2g}{\zeta\epsilon}$ or $\Delta_{i+1}^+(\mathcal{X}) = \Omega(L_i)$, then $\Delta_{i+1}^+(\mathcal{X}) = \Omega(\epsilon L_i |\mathcal{V}(\mathcal{X})|)$.*

Proof: We first consider the case where $|\mathcal{V}(\mathcal{X})| \geq \frac{2g}{\zeta\epsilon}$. By definition of corrected potential change in Item (3) of Lemma 11.1, we have:

$$\begin{aligned}\Delta_{i+1}^+(\mathcal{X}) &\geq \sum_{\varphi \in \mathcal{V}(\mathcal{X})} \omega(\varphi) - \text{Adm}(\mathcal{X}) \geq (\zeta\epsilon L_i |\mathcal{V}(\mathcal{X})|) - \text{Adm}(\mathcal{X}) \\ &\geq (\zeta\epsilon L_i |\mathcal{V}(\mathcal{X})|)/2 - gL_i + (\zeta\epsilon L_i |\mathcal{V}(\mathcal{X})|)/2 = \Omega(\epsilon L_i |\mathcal{V}(\mathcal{X})|) .\end{aligned}$$

Next, we consider the case where $\Delta_{i+1}^+(\mathcal{X}) = \Omega(L_i)$. If $|\mathcal{V}(\mathcal{X})| \geq \frac{2g}{\zeta\epsilon}$, then $\Delta_{i+1}^+(\mathcal{X}) = \Omega(\epsilon L_i |\mathcal{V}(\mathcal{X})|)$ as we have just shown. Otherwise, we have:

$$\Delta_{i+1}^+(\mathcal{X}) = \Omega(L_i) = \Omega(\epsilon L_i \frac{2g}{\zeta\epsilon}) = \Omega(\epsilon L_i |\mathcal{V}(\mathcal{X})|),$$

as claimed. $\square$

Lemma 11.18. $\Delta_{i+1}^+(\mathcal{X}) \geq 0$ for every $\mathcal{X} \in \mathbb{X}$ and

$$\sum_{\mathcal{X} \in \mathbb{X}^+} \Delta_{i+1}^+(\mathcal{X}) = \sum_{\mathcal{X} \in \mathbb{X}^+} \Omega(|\mathcal{V}(\mathcal{X})| \epsilon L_i).$$

Consequently, Item (3) of Lemma 11.1 holds.

Proof: The fact that $\Delta_{i+1}^+(\mathcal{X}) \geq 0$ follows directly from the definition. By Lemma 11.17 for every $\mathcal{X} \in \mathbb{X}_2^- \cup \mathbb{X}_4^- \cup \mathbb{X}_6$, it holds that

$$\Delta_{i+1}^+(\mathcal{X}) = \Omega(\epsilon L_i |\mathcal{V}(\mathcal{X})|) , \quad (58)$$

By the definition of $\mathbb{X}^+$ in Equation (57), the only case where $\Delta_{i+1}^+(\mathcal{X})$ could be 0 is $\mathcal{X} \in \mathbb{X}_5^{\text{pref}-}$. Next, we use an averaging argument to assign potential change to $\mathcal{X}$. Observe that $\mathcal{X}$ is an uncontraction of some prefix $\bar{Q}$ of some path $\bar{P} \in \bar{F}^{(5)}$. By Lemma 11.13, the uncontraction of the other suffix $\bar{Q}'$ of $\bar{P}$, say $\tilde{Q}'$, is augmented to a subgraph in $\mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4$. It follows that $\tilde{Q}'$ is a subgraph of some graph $\mathcal{Y} \in \mathbb{X}_2^- \cup \mathbb{X}_4^- \cup \mathbb{X}_6$. If we distribute the corrected potential change $\Delta_{i+1}^+(\mathcal{Y})$ to nodes in $\mathcal{Y}$, each node gets $\Omega(\epsilon L_i)$ potential change. Thus, the total potential change of nodes in $\tilde{Q}'$ is $\Omega(\epsilon L_i |\mathcal{V}(\tilde{Q}')|)$. By Item (3) of Lemma 11.13, $|\mathcal{V}(\tilde{Q}')| = \Omega(1/\epsilon)$. Thus the potential change of nodes in $\tilde{Q}'$ is $\Omega(L_i)$. We distribute *half* of the potential change to $\mathcal{X}$. Thus, $\mathcal{X}$ has $\Omega(L_i)$ potential change, and by Lemma 11.17, the potential change of $\mathcal{X}$ is $\Omega(\epsilon L_i |\mathcal{V}(\mathcal{X})|)$. This, with Equation (58), implies that:

$$\sum_{\mathcal{X} \in \mathbb{X}^+} \Delta_{i+1}^+(\mathcal{X}) = \sum_{\mathcal{X} \in \mathbb{X}^+} \Omega(|\mathcal{V}(\mathcal{X})| \epsilon L_i),$$

as desired. $\square$

11.2.2 Constructing the partition of $\mathcal{E}_i$: Proof of Lemma 11.1

In this section, we construct a partition of $\mathcal{E}$ and prove Lemma 11.1. Items (3) and (5) of Lemma 11.1 were proved in Lemma 11.18 and Lemma 11.16, respectively. In the following, we prove Items (1), (2) and (4). Indeed, Item (2) follows directly from the construction (Observation 11.20). Item (1) is proved in Lemma 11.26 and Item (4) is proved in Lemma 11.27 and Lemma 11.28.

Recall that we define $\mathbb{X}' = \mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4 \cup \mathbb{X}_5^{\text{pref}} \cup \widehat{\mathbb{X}_5^{\text{intrnl}}}$ in Lemma 11.14. We say that a subgraph $\mathcal{X} \in \mathbb{X}'$ is *light* if it corresponds to a light supernode in $\widehat{\mathcal{K}}_i$ (defined in Step 6); otherwise, we say that $\mathcal{X}$ is *heavy*. We construct $\mathcal{E}_i^{\text{take}}$ and $\mathcal{E}_i^{\text{redunt}}$ in two steps below; $\mathcal{E}_i^{\text{reduce}} = \mathcal{E}_i \setminus (\mathcal{E}_i^{\text{take}} \cup \mathcal{E}_i^{\text{redunt}})$. Initially, both sets are empty.

Constructing $\mathcal{E}_i^{\text{take}}$ and $\mathcal{E}_i^{\text{redunt}}$: Let $\mathbb{X}^{\text{light}}$ be the set of light subgraphs in $\mathbb{X}'$.

- **Step 1:** For each subgraph $\mathcal{X} \in \mathbb{X}$, we add all edges of $\mathcal{E}_i$ in $\mathcal{X}$ to $\mathcal{E}_i^{\text{take}}$. That is,

$$\mathcal{E}_i^{\text{take}} \leftarrow \mathcal{E}_i^{\text{take}} \cup (\mathcal{E}_i \cap \mathcal{E}(\mathcal{X})).$$

- **Step 2:** We construct a graph $\mathcal{H}_i = (\mathcal{V}_i, \widehat{\text{MST}}_i \cup \mathcal{E}_i^{\text{take}}, \omega)$. We then consider every edge $\mathbf{e} = (\nu \cup \mu) \in \mathcal{E}_i$, where both endpoints are in subgraphs in $\mathbb{X}^{\text{light}}$, in the non-decreasing order of the weight. If:

$$d_{\mathcal{H}_i}(\nu, \mu) > 2\omega(\mathbf{e}) , \quad (59)$$

then we add $\mathbf{e}$ to $\mathcal{E}_i^{\text{take}}$ (and hence, also to $\mathcal{H}_i$). Otherwise, we add $\mathbf{e}$ to $\mathcal{E}_i^{\text{redunt}}$. Note that the distance in $\mathcal{H}_i$ in Equation (59) is the augmented distance.

The construction in Step 2 is the path greedy algorithm. We observe that:

Observation 11.19. *For every edge $\mathbf{e} \in \mathcal{E}_i^{\text{reduce}}$, at least one endpoint of $\mathbf{e}$ is in a heavy subgraph.*

Observation 11.20. Let $H_{<L_i}^-$ be a subgraph obtained by adding corresponding edges of $\mathcal{E}_i^{\text{take}}$ to $H_{<L_{i-1}}$. Then for every edge (u, v) that corresponds to an edge in $\mathcal{E}^{\text{redundant}}$, $d_{H_{<L_i}^-}(u, v) \leq 2d_G(u, v)$.

We now focus on proving Item (1) of Lemma 11.1. The key idea is the following lemma.

Lemma 11.21. Any subgraph $\mathcal{X} \in \mathbb{X}' \setminus \mathbb{X}_1$ can be partitioned into $k = O(1/\zeta)$ subgraphs $\{\mathcal{Y}_1, \dots, \mathcal{Y}_k\}$ such that $\text{Adm}(\mathcal{Y}_j) \leq 9\zeta L_i$ for any $1 \leq j \leq k$ when $\epsilon \leq \frac{\zeta}{g}$.

Proof: Let φ be a branching node in $\tilde{F}^{(2)}$, the tree in Lemma 11.7. We say that φ is *special* if there exists three internally node disjoint paths $\tilde{P}_1, \tilde{P}_2, \tilde{P}_3$ of $\tilde{F}^{(2)}$ sharing the same node φ such that $\text{Adm}(\tilde{P}_j \setminus \{\varphi\}) \geq \zeta L_i$. Observe by the construction in Lemma 11.7 that

Observation 11.22. Any special node φ of $\tilde{F}^{(2)}$ is contained in a subgraph in $\mathbb{X}_2$.

By Lemma 11.14, $\mathcal{X}$ is a tree. Let $\mathcal{X}'$ be a maximal subtree of $\mathcal{X}$ such that $\mathcal{X}'$ is a subtree of $\widetilde{\text{MST}}_i$. If $\mathcal{X}$ is in $\mathbb{X}_2 \cup \mathbb{X}_5^{\text{pref}} \cup \mathbb{X}_5^{\text{intrnl}}$ then $\mathcal{X}' = \mathcal{X}$. Otherwise, $\mathcal{X} \in \mathbb{X}_4$, and thus it has a single edge in $\mathcal{E}_i$ by Item (1) of Lemma 11.10. That is, $\mathcal{X}$ has exactly two such maximal subtrees $\mathcal{X}'$. Thus, to complete the lemma, we show that $\mathcal{X}'$ can be partitioned into $O(1/\zeta)$ subtrees as claimed in the lemma.

Let $\mathcal{D}$ be the path in $\mathcal{X}'$ of maximum augmented diameter. Let $\mathcal{J}$ be the forest obtained from $\mathcal{X}'$ by removing nodes of $\mathcal{D}$. Then,

Observation 11.23. $\text{Adm}(\mathcal{T}) \leq 2\zeta L_i \quad \forall \text{ tree } \mathcal{T} \in \mathcal{J}$

Now we greedily partition $\mathcal{D}$ into $k = O(1/\zeta)$ subpaths $\{\mathcal{P}_1, \dots, \mathcal{P}_k\}$, each of augmented diameter at least ζL_i and at most $3\zeta L_i$. This is possible because each node/edge has a weight at most $\max\{g\epsilon L_i, \bar{w}\} \leq \max\{g\epsilon L_i, \epsilon L_i\} \leq \zeta L_i$ when $\epsilon \leq \zeta/g$. Next, for every tree $\mathcal{T} \in \mathcal{J}$, if $\mathcal{T}$ is connected to a node $\varphi \in \mathcal{P}_j$ via some $\widetilde{\text{MST}}_i$ edge $\mathbf{e}$ for some $j \in [1, k]$, we augment $\mathbf{e}$ and $\mathcal{T}$ to $\mathcal{P}_j$. By Observation 11.23, the augmentation increases the diameter of $\mathcal{P}$ by at most $2(\bar{w} + 2\zeta L_i) \leq 6\zeta L_i$ additively. $\square$

Lemma 11.24. Let $\mathcal{X}, \mathcal{Y}$ be two (not necessarily distinct) subgraphs in $\mathbb{X}^{\text{light}}$. Then there are $O(1)$ edges in $\mathcal{E}_i^{\text{take}}$ between nodes in $\mathcal{X}$ and nodes in $\mathcal{Y}$.

Proof: Let $\{\mathcal{A}_1, \dots, \mathcal{A}_x\}$ ($\{\mathcal{B}_1, \dots, \mathcal{B}_y\}$) be a partition of $\mathcal{X}$ ($\mathcal{Y}$) into $x = O(1/\zeta)$ ($y = O(1/\zeta)$) subgraphs of augmented diameter at most $9\zeta L_i$ as guarantee by Lemma 11.21. Observe that by Equation (59), there is at most one edge in $\mathcal{E}^{\text{take}}$ between $\mathcal{A}_j$ and $\mathcal{B}_k$ for any $1 \leq j \leq x, 1 \leq k \leq y$. Thus, the number of edges in $\mathcal{E}^{\text{take}}$ between $\mathcal{X}$ and $\mathcal{Y}$ is at most $x \cdot y = O(1/\zeta^2) = O(1)$. $\square$

We obtain the following corollary of Lemma 11.24.

Corollary 11.25. For any $\mathcal{X} \in \mathbb{X}^{\text{light}}$, $\deg_{\mathcal{G}_i^{\text{take}}}(\mathcal{V}(\mathcal{X})) = O(1/\epsilon) = O(|\mathcal{V}(\mathcal{X})|)$ where $\mathcal{G}_i^{\text{take}} = (\mathcal{V}_i, \mathcal{E}_i^{\text{take}})$.

We now prove Item (1) of Lemma 11.1.

Lemma 11.26. For every subgraph $\mathcal{X} \in \mathbb{X}$, $\deg_{\mathcal{G}_i^{\text{take}}}(\mathcal{X}) = O(|\mathcal{V}(\mathcal{X})|)$ where $\mathcal{G}_i^{\text{take}} = (\mathcal{V}_i, \mathcal{E}_i^{\text{take}})$, and $\mathcal{E}(\mathcal{X}) \cap \mathcal{E}_i \subseteq \mathcal{E}^{\text{take}}$. Furthermore, if $\mathcal{X} \in \mathbb{X}^-$, there is no edge in $\mathcal{E}_i^{\text{reduce}}$ incident to a node in $\mathcal{X}$.

Proof: Let $\mathcal{X}$ be a subgraph in $\mathbb{X}$. Observe by the construction of $\mathcal{E}_i^{\text{take}}$ in Step 1, $\mathcal{E} \cap \mathcal{E}(\mathcal{X}) \subseteq \mathcal{E}_i^{\text{take}}$. Clearly, the number of edges incident to nodes in $\mathcal{X}$ added in Step 1 is $O(|\mathcal{V}(\mathcal{X})|)$ since every subgraph in $\mathbb{X}$ is a tree by Lemma 11.14. Thus, it remains to bound the number of edges added in Step 2.

If $\mathcal{X} \in \mathbb{X}_2 \cup \mathbb{X}_4 \cup \mathbb{X}_5^{\text{pref}} \cup \mathbb{X}_5^{\text{intrnl-}}$, then $\mathcal{X}$ corresponds to a light supernode in $\mathcal{K}_i$. Thus, $\deg_{\mathcal{G}_i^{\text{take}}}(\mathcal{V}(\mathcal{X})) = O(|\mathcal{V}(\mathcal{X})|)$ by Corollary 11.25. Otherwise, $\mathcal{X} \in \mathbb{X}_6$. By construction in Step 6, $\mathcal{X}$ is the union heavy

subgraphs and light subgraphs (and some edges in $\mathcal{E}_i$). By construction of $\mathcal{E}_i^{\text{take}}$, only light subgraphs have nodes incident to edges in $\mathcal{E}_i^{\text{take}}$. Let $\{\mathcal{Y}_1, \dots, \mathcal{Y}_p\}$ be the set of light subgraphs constituting $\mathcal{X}$. Then, by Corollary 11.25, we have that:

$$\deg_{\mathcal{G}_i^{\text{take}}}(\mathcal{X}) \leq \sum_{k=1}^p \deg_{\mathcal{G}_i^{\text{take}}}(\mathcal{Y}_k) = \sum_{k=1}^p (|\mathcal{V}(\mathcal{Y}_k)|) = O(|\mathcal{V}(\mathcal{X})|).$$

We now show that there is no edge in $\mathcal{E}_i^{\text{reduce}}$ incident to a node in $\mathcal{X} \in \mathbb{X}^-$. Suppose otherwise, let $\mathbf{e}$ be such an edge. By Observation 11.19, $\mathbf{e}$ is incident to a node in a heavy subgraph, say $\mathcal{Y}$. That is, $\widehat{\varphi}_{\mathcal{Y}} \in \widehat{\mathcal{V}}_i^{\text{high}}$. By the construction in Step 6, $\widehat{\varphi}_{\mathcal{X}} \in \widehat{\mathcal{V}}_i^{\text{high}^+}$ and hence $\mathcal{X}$ is grouped to a larger subgraph in $\mathbb{X}_6$, contradicting that $\mathcal{X} \in \mathbb{X}^-$. $\square$

We now focus on proving Item (4) of Lemma 11.1. In Lemma 11.27, we consider the non-degenerate case, and in Lemma 11.28 we consider the degenerate case.

Lemma 11.27. *Let (φ_1, φ_2) be any edge in $\mathcal{E}_i$ between nodes of two light subgraphs $\mathcal{X}, \mathcal{Y}$ in $\mathbb{X}_5^{\text{intrnl}}$. Then, $(\varphi_1, \varphi_2) \in \mathcal{E}_i^{\text{redunt}}$.*

Proof: By the construction of Step 5, $\mathcal{X}$ and $\mathcal{Y}$ correspond to two subpaths $\bar{\mathcal{X}}$ and $\bar{\mathcal{Y}}$ of two paths $\bar{P}$ and $\bar{Q}$ in $\bar{F}^{(5)}$. Note that all nodes in $\bar{\mathcal{X}}$ and $\bar{\mathcal{Y}}$ have a blue color since the suffix/prefix of $\bar{P}$ and $\bar{Q}$ are either in $\mathbb{X}_5^{\text{pref}}$ or are augmented to existing subgraphs in Step 5B.

Since there is an edge in $\mathcal{E}_i$ between $\mathcal{X}$ and $\mathcal{Y}$, there must be an edge in $\bar{\mathcal{E}}_i$, say $(\bar{\mu}, \bar{\nu})$ between a node of $\bar{\mu} \in \bar{\mathcal{X}}$ and a node of $\bar{\nu} \in \bar{\mathcal{Y}}$ by the definition of $\bar{\mathcal{E}}_i$ (in Equation (54)) such that $\varphi_1 \in \bar{\mu}, \varphi_2 \in \bar{\nu}$.

As $\bar{\mu}$ and $\bar{\nu}$ both have a blue color, either $(\bar{\mu}, \bar{\nu}) \in \mathcal{E}_i^{\text{far}}(\bar{F}^{(5)})$ or $(\bar{\mu}, \bar{\nu}) \in \mathcal{E}_i^{\text{close}}(\bar{F}^{(5)})$ by the definition in Equation (56). By Lemma 11.10, $\mathcal{E}_i^{\text{far}}(\bar{F}^{(5)}) = \emptyset$. Thus, $(\bar{\mu}, \bar{\nu}) \in \mathcal{E}_i^{\text{close}}(\bar{F}^{(5)})$. This implies $\bar{I}(\bar{\nu}) \cap \bar{I}(\bar{\mu}) \neq \emptyset$, and hence, $\bar{\mathcal{X}}$ and $\bar{\mathcal{Y}}$ are broken from the same path, say $\bar{P} \in \bar{F}^{(5)}$, in Step 5B.

Furthermore, by the definition of $\bar{I}(\bar{\nu})$, every node $\bar{\varphi} \in \bar{I}(\bar{\nu})$ is within an augmented distance (along $\bar{P}$) of at most $(1 - \psi)L_i$ from $\bar{\nu}$. This means, $\text{Adm}(\bar{P}[\bar{\nu}, \bar{\mu}]) \leq 2(1 - \psi)L_i$. Note that the uncontraction of $\bar{P}[\bar{\nu}, \bar{\mu}]$ is a subtree of $\widetilde{\text{MST}}_i$. Thus, $d_{\widetilde{\text{MST}}_i}(\varphi_1, \varphi_2) \leq \text{Adm}(\bar{P}[\bar{\nu}, \bar{\mu}]) \leq 2(1 - \psi)L_i \leq \frac{2L_i}{1+\psi} \leq 2\omega(\varphi_1, \varphi_2)$. As $\widetilde{\text{MST}}_i$ is a subgraph of $\mathcal{H}_i$, (φ_1, φ_2) will be added to $\mathcal{E}_i^{\text{redunt}}$ in Step 2, Equation (59). $\square$

Lemma 11.28 (Structure of Degenerate Case). *If the degenerate case happens, then $\bar{F}_i^{(5)} = \bar{F}_i^{(4)} = \bar{F}_i^{(3)}$, and $\bar{F}_i^{(5)}$ is a single (long) path. Moreover, $|\mathcal{E}_i^{\text{take}}| = O(1/\epsilon)$.*

Proof: Recall that the degenerate case happens when $\mathbb{X}_1^- \cup \mathbb{X}_2^- \cup \mathbb{X}_4^- = \mathbb{X}_6 = \emptyset$. This implies $\mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4 = \emptyset$. Thus, $\bar{F}_i^{(5)} = \bar{F}_i^{(4)} = \bar{F}_i^{(3)}$. Furthermore, $\bar{F}_i^{(5)}$ is a single (long) path since $\bar{F}_i^{(3)}$ is a path by Lemma 11.7. This gives $|\mathbb{X}_5^{\text{pref}}| = 2$. By Lemma 11.27, there is no edge in $\mathcal{E}_i^{\text{take}}$ between two subgraphs in $\mathbb{X}_5^{\text{intrnl}}$. Thus, any edge in $\mathcal{E}_i^{\text{take}}$ must be incident to a node in a subgraph of $\mathcal{X} \in \mathbb{X}_5^{\text{pref}}$. By Corollary 11.25, there are $O(1/\epsilon)$ such edges. $\square$

12 Clustering for Stretch $t = 1 + \epsilon$

In this section, we prove Lemma 10.3 when the stretch $t = 1 + \epsilon$. The key technical idea is the following clustering lemma, which is analogous to Lemma 11.1 in Section 11; the highlighted texts below are the major differences. Recall that $H_{<L_{i-1}}$ is the spanner constructed for edges of G of weight less than L_{i-1} .

Lemma 12.1. Let $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$ be the cluster graph. We can construct in polynomial time (i) a collection $\mathbb{X}$ of subgraphs of $\mathcal{G}_i$ and its partition into two sets $\{\mathbb{X}^+, \mathbb{X}^-\}$ and (ii) a partition of $\mathcal{E}_i$ into three sets $\{\mathcal{E}_i^{\text{take}}, \mathcal{E}_i^{\text{reduce}}, \mathcal{E}_i^{\text{redund}}\}$ such that:

- (1) For every subgraph $\mathcal{X} \in \mathbb{X}$, $\deg_{\mathcal{G}_i^{\text{take}}}(\mathcal{V}(\mathcal{X})) = O(|\mathcal{V}(\mathcal{X})|/\epsilon)$ where $\mathcal{G}_i^{\text{take}} = (\mathcal{V}_i, \mathcal{E}_i^{\text{take}})$, and $\mathcal{E}(\mathcal{X}) \cap \mathcal{E}_i \subseteq \mathcal{E}_i^{\text{take}}$. Furthermore, if $\mathcal{X} \in \mathbb{X}^-$, there is no edge in $\mathcal{E}_i^{\text{reduce}}$ incident to a node in $\mathcal{X}$.
- (2) Let $H_{<L_i}^-$ be a subgraph obtained by adding corresponding edges of $\mathcal{E}_i^{\text{take}}$ to $H_{<L_{i-1}}$. Then for every edge (u, v) that corresponds to an edge in $\mathcal{E}_i^{\text{redund}}$, $d_{H_{<L_i}^-}(u, v) \leq (1 + 6g\epsilon)2d_G(u, v)$.
- (3) Let $\Delta_{i+1}^+(\mathcal{X}) = \Delta(\mathcal{X}) + \sum_{\mathbf{e} \in \widetilde{\text{MST}}_i \cap \mathcal{E}(\mathcal{X})} w(\mathbf{e})$ be the corrected potential change of $\mathcal{X}$. Then, $\Delta_{i+1}^+(\mathcal{X}) \geq 0$ for every $\mathcal{X} \in \mathbb{X}$ and

$$\sum_{\mathcal{X} \in \mathbb{X}^+} \Delta_{i+1}^+(\mathcal{X}) = \sum_{\mathcal{X} \in \mathbb{X}^+} \Omega(|\mathcal{V}(\mathcal{X})|\epsilon L_i). \quad (60)$$

- (4) There exists an orientation of edges in $\mathcal{E}_i^{\text{take}}$ such that for every subgraph $\mathcal{X} \in \mathbb{X}^-$, if $\mathcal{X}$ has t outgoing edges for some $t \geq 0$, then $\Delta_{i+1}^+(\mathcal{X}) = \Omega(|\mathcal{V}(\mathcal{X})|t\epsilon^2 L_i)$, unless a degenerate case happens, in which $\mathcal{E}_i^{\text{reduce}} = \emptyset$ and

$$\omega(\mathcal{E}_i^{\text{take}}) = O(\frac{1}{\epsilon^2})(\sum_{\mathcal{X} \in \mathbb{X}} \Delta_{i+1}^+(\mathcal{X}) + L_i).$$

- (5) For every subgraph $\mathcal{X} \in \mathbb{X}$, $\mathcal{X}$ satisfies the three properties (P1')-(P3') with constant $g = 31$.

The total node degree of $\mathcal{X}$ in $\mathcal{G}_i^{\text{take}}$ in Lemma 12.1 is worst than the total node degree of $\mathcal{X}$ in Lemma 11.1 by a factor of $1/\epsilon$. Furthermore, Item (4) of Lemma 12.1 is qualitatively different from Item (4) of Lemma 11.1 and we no longer can bound the size of $\mathcal{E}_i^{\text{take}}$ in the degenerate case. All of these are due to the fact that the stretch $t = 1 + \epsilon < 2$ when $\epsilon < 1$.

Next we show to construct H_i given that we can construct a set of subgraphs $\mathbb{X}$ as claimed in Lemma 12.1. The proof of Lemma 12.1 is deferred to Section 12.2.

12.1 Constructing H_i : Proof of Lemma 10.3 for $t = 1 + \epsilon$.

Let $\widetilde{\text{MST}}_i^{\text{in}}(\mathcal{X}) = \mathcal{E}(\mathcal{X}) \cap \widetilde{\text{MST}}_i$ for each $\mathcal{X} \in \mathbb{X}$. Let $\widetilde{\text{MST}}_i^{\text{in}} = \cup_{\mathcal{X} \in \mathbb{X}} (\mathcal{E}(\mathcal{X}) \cap \widetilde{\text{MST}}_i)$ be the set of $\widetilde{\text{MST}}_i$ edges that are contained in subgraphs in $\mathbb{X}$. The construction of H_i is exactly the same as the construction of H_i in Section 11.1: first, add every edge of $\mathcal{E}_i^{\text{take}}$ to H_i , and then apply SSO on the subgraph of $\mathcal{G}_i$ induced by $\mathcal{E}_i^{\text{reduce}}$. Furthermore, Claim 7.10 and Observation 11.3 hold here.

Recall that F_i^σ is the set of edges in E_i^σ that correspond to $\mathcal{E}_i$. By the same proof in Lemma 11.2 we have:

Lemma 12.2. For every edge $(u, v) \in F_i^\sigma$, $d_{H_{<L_i}}(u, v) \leq t(1 + \max\{s_{\text{SSO}}(2g), 6g\}\epsilon)w(u, v)$.

Next, we bound the total weight of H_i .

Lemma 12.3. $w(H_i) \leq \lambda \Delta_{i+1} + a_i$ for $\lambda = O(\chi\epsilon^{-1} + \epsilon^{-2})$ and $a_i = O(\chi\epsilon^{-1} + \epsilon^{-2})w(\widetilde{\text{MST}}_i^{\text{in}}) + O(L_i/\epsilon^2)$.

Proof: First, we consider the non-degenerate case. Note that edges in $\mathcal{E}_i^{\text{redund}}$ are not added to H_i . Let $\mathcal{V}_i^+ = \cup_{\mathcal{X} \in \mathbb{X}^+} \mathcal{V}(\mathcal{X})$ and $\mathcal{V}_i^- = \cup_{\mathcal{X} \in \mathbb{X}^-} \mathcal{V}(\mathcal{X})$. Let $F_i^{(a)}$ be the set of edges added to H_i in the construction in Step a , $a \in \{1, 2\}$.

By the construction in Step 1, $F_i^{(1)}$ includes edges in $\mathcal{E}_i^{\text{take}}$. Let $A^{(1)} \subseteq F_i^{(1)}$ be the set of edges incident to at least one node in $\mathcal{V}_i^+$ and $A^{(2)} = F_i^{(1)} \setminus A^{(1)}$. By Item (1) in Lemma 12.1, the total weight of the edges added to H_i in Step 1 is:

$$w(A_i^{(1)}) = \sum_{\mathcal{X} \in \mathbb{X}^+} O(|\mathcal{V}(\mathcal{X})|/\epsilon) L_i \stackrel{\text{Eq. (60)}}{=} O\left(\frac{1}{\epsilon^2}\right) \sum_{\mathcal{X} \in \mathbb{X}^+} \Delta_{i+1}^+(\mathcal{X}) = O\left(\frac{1}{\epsilon^2}\right) (\Delta_{i+1} + w(\widetilde{\text{MST}}_i^{\text{in}})). \quad (61)$$

By definition $A^{(2)}$ is the set of edges with both endpoints in subgraphs of $\mathcal{V}_i^-$. Consider the orientation of $\mathcal{E}_i^{\text{take}}$ as in Lemma 12.1. Then, every edge of $A^{(2)}$ is an out-going edge from some node in a graph in $\mathbb{X}^-$. For each graph $\mathcal{X} \in \mathbb{X}^-$, by Item (4) of Lemma 12.1, the total weight of incoming edges of $\mathcal{X}$ is $O(tL_i) = O(1/\epsilon^2)\Delta_{i+1}^+(\mathcal{X})$. Thus, we have:

$$w(A_i^{(2)}) = O\left(\frac{1}{\epsilon^2}\right) \sum_{\mathcal{X} \in \mathbb{X}} \Delta_{i+1}^+(\mathcal{X}) = O\left(\frac{1}{\epsilon^2}\right) (\Delta_{i+1} + w(\widetilde{\text{MST}}_i^{\text{in}})). \quad (62)$$

Thus, by Equations (61) and (62), we have $w(F^{(1)}) = O\left(\frac{1}{\epsilon^2}\right) (\Delta_{i+1} + w(\widetilde{\text{MST}}_i^{\text{in}}))$. By the exactly the same argument in Lemma 12.3, we have that $w(F^{(2)}) = O(\chi/\epsilon) (\Delta_{i+1} + w(\widetilde{\text{MST}}_i^{\text{in}}))$. This gives:

$$w(H_i) = O(\chi/\epsilon + 1/\epsilon^2) (\Delta_{i+1} + w(\widetilde{\text{MST}}_i^{\text{in}})) \leq \lambda (\Delta_{i+1} + w(\widetilde{\text{MST}}_i^{\text{in}})) \quad (63)$$

for some $\lambda = O(\chi/\epsilon + 1/\epsilon^2)$.

It remains to consider the degenerate case, and in which case, we only add to H_i edges corresponding to $\mathcal{E}_i^{\text{take}}$. Thus, by Item (4) of Lemma 12.1, we have:

$$w(H_i) = O\left(\frac{L_i}{\epsilon^2}\right) \leq \lambda \cdot (\Delta_{i+1} + w(\widetilde{\text{MST}}_i^{\text{in}})) + O\left(\frac{L_i}{\epsilon^2}\right), \quad (64)$$

since $\Delta_{i+1} + w(\widetilde{\text{MST}}_i^{\text{in}}) = \sum_{\mathcal{X} \in \mathbb{X}} \Delta_{i+1}^+(\mathcal{X})$ by Item (1) in Observation 11.3. Thus, the lemma follows from Equations (63) and (64). $\square$

We are now ready to prove Lemma 10.3 for the case $t = 1 + \epsilon$.

Proof: [Proof of Item 2 of Lemma 10.3] The fact that subgraphs in $\mathbb{X}$ satisfy the three properties (P1')-(P3') with constant $g = 31$ follows from Item (5) of Lemma 12.1. The stretch in $H_{<L_i}$ of edges in F_i^σ follows from Lemma 12.2.

By Lemma 12.3, $w(H_i) \leq \lambda \Delta_{i+1} + a_i$ where $\lambda = O(\chi\epsilon^{-1} + \epsilon^{-2})$ and $a_i = O(\chi\epsilon^{-1} + \epsilon^{-2})w(\widetilde{\text{MST}}_i^{\text{in}}) + O(L_i/\epsilon^2)$. It remains to show that $A = \sum_{i \in \mathbb{N}^+} a_i = O(\chi\epsilon^{-1} + \epsilon^{-2})$. Observe that

$$\sum_{i \in \mathbb{N}^+} O\left(\frac{L_i}{\epsilon^2}\right) = O\left(\frac{1}{\epsilon^2}\right) \sum_{i=1}^{i_{\max}} \frac{L_{i_{\max}-i}}{\epsilon^{i_{\max}-i}} = O\left(\frac{L_{i_{\max}}}{\epsilon^2(1-\epsilon)}\right) = O\left(\frac{1}{\epsilon^2}\right) w(\text{MST});$$

here $i_{\max}$ is the maximum level. The last equation is due to that $\epsilon \leq 1/2$ and every edge has weight at most $w(\text{MST})$ since the weight of every is the shortest distance between its endpoints. By Item (2) of Observation 11.3, $\sum_{i \in \mathbb{N}^+} \widetilde{\text{MST}}_i^{\text{in}} \leq w(\text{MST})$. Thus, $A = O(\chi/\epsilon^2) + O(1/\epsilon^2)$ as desired. $\square$

12.2 Clustering

In this section, we prove Lemma 12.1. The construction of $\mathbb{X}$ has 5 steps. The first four steps are exactly the same as the first four steps in the construction in Section 11. In Step 5, we construct $\mathbb{X}_5^{\text{intrnl}}$ differently, taking into account of edges in $\mathcal{E}_i^{\text{close}}$ in Equation (56). Recall that when the stretch parameter $t \geq 2$, we show that edges in $\mathcal{E}_i$ corresponding to $\widetilde{\mathcal{E}}_i^{\text{close}}$ are added to $\mathcal{E}_i^{\text{redund}}$ (implicitly in Lemma 11.27). However, when $t = 1 + \epsilon$, we could not afford to do so, and the construction in Step 5 will take care of these edges.

Steps 1-4. The construction of Steps 1 to 4 are exactly the same as Steps 1-4 in Section 11.2 to obtain three sets of clusters $\mathbb{X}_1, \mathbb{X}_2$ and $\mathbb{X}_4$ whose properties are described in Lemmas 11.5, 11.7, 11.9 and 11.10. After the four steps, we obtain the forest $\overline{F}_i^{(5)}$, where every tree is a path. In particular, for every edge $(\bar{\mu}, \bar{\nu}) \in \bar{\mathcal{E}}_i$ with both endpoints in $\overline{F}_i^{(5)}$, either (i) the edge is in $\bar{\mathcal{E}}_i^{\text{close}}(\overline{F}_i^{(5)})$, or (ii) at least one of the endpoints must belong to a low-diameter tree of $\overline{F}_i^{(5)}$ or (iii) in a (red) suffix of a long path in $\overline{F}_i^{(5)}$ of augmented diameter at most L_i .

Before moving on to Step 5, we need a preprocessing step in which we find all edges in $\mathcal{E}_i^{\text{redund}}$. The construction of Step 5 relies on edges that are not in $\mathcal{E}_i^{\text{redund}}$.

Constructing $\mathcal{E}_i^{\text{redund}}$ and $\mathcal{E}_i^{\text{take-}}$. Let $\widetilde{F}_i^{(5)}$ be obtained from $\overline{F}_i^{(5)}$ by uncontracting the contracted nodes. We apply the greedy algorithm. Initially, both $\mathcal{E}_i^{\text{redund}}$ and $\mathcal{E}_i^{\text{take-}}$ are empty sets. We construct a graph $\mathcal{H}_i = (\mathcal{V}_i, \widetilde{\text{MST}}_i \cup \mathcal{E}_i^{\text{take-}}, \omega)$, which initially only include edges in $\widetilde{\text{MST}}_i$. We then consider every edge $\mathbf{e} = (\nu \cup \mu) \in \mathcal{E}_i$, where both endpoints are in $\mathcal{V}(\widetilde{F}_i^{(5)})$, in the non-decreasing order of the weight. If:

$$d_{\mathcal{H}_i}(\nu, \mu) \leq (1 + 6g\epsilon)\omega(\mathbf{e}), \quad (65)$$

then we add $\mathbf{e}$ to $\mathcal{E}_i^{\text{redund}}$. Otherwise, we add $\mathbf{e}$ to $\mathcal{E}_i^{\text{take-}}$ (and hence to $\mathcal{H}_i$). Note that the distance in $\mathcal{H}_i$ in Equation (65) is the augmented distance. We have the following observation which follows directly from the greedy algorithm.

Observation 12.4. For every edge $\mathbf{e} = (\nu, \mu) \in \mathcal{E}_i^{\text{take-}}$, $d_{\mathcal{H}_i}(\nu, \mu) \geq (1 + 6g\epsilon)\omega(\mathbf{e})$.

Step 5. Let $\overline{P}$ be a path in $\overline{F}_i^{(5)}$ obtained by Item (5) of Lemma 11.10. We construct two sets of subgraphs, denoted by $\mathbb{X}_5^{\text{intrnl}}$ and $\mathbb{X}_5^{\text{pref}}$, of $\mathcal{G}_i$. The construction is broken into two steps. Step 5A is only applicable when $\mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4 \neq \emptyset$. In Step 5B, we need a more involved construction by [50], as described in Lemma 12.5.

- (Step 5A) If $\overline{P}$ has augmented diameter at most $6L_i$, let $\mathbf{e}$ be an $\widetilde{\text{MST}}_i$ edge connecting $\tilde{P}^{\text{uctrt}}$ and a node in some subgraph $\mathcal{X} \in \mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4$; $\mathbf{e}$ exists by Lemma 11.10. We add both $\mathbf{e}$ and $\tilde{P}^{\text{uctrt}}$ to $\mathcal{X}$.
- (Step 5B) Otherwise, the augmented diameter of $\overline{P}$ is at least $6L_i$. Let $\{\overline{Q}_1, \overline{Q}_2\}$ be the suffix and prefix of $\overline{P}$ such that $\overline{Q}_1^{\text{uctrt}}$ and $\overline{Q}_2^{\text{uctrt}}$ have augmented diameter at least L_i and at most $2L_i$. If $\overline{Q}_j$, $j \in \{1, 2\}$ is connected to a node in a subgraph $\mathcal{X} \in \mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4$ via an edge $e \in \widetilde{\text{MST}}_i$, we add $\tilde{Q}_j^{\text{uctrt}}$ and e to $\mathcal{X}$. If $\overline{Q}_j$ contains an endpoint of $\overline{P}$, we add $\tilde{Q}_j^{\text{uctrt}}$ to $\mathbb{X}_5^{\text{pref}}$.

Next, denote by $\overline{P}'$ the path obtained by removing $\overline{Q}_1, \overline{Q}_2$ from $\overline{P}$. We then apply the construction in Lemma 12.5 to $\overline{P}'$ to obtain a set of subgraphs $\mathbb{C}_5(\overline{P}')$ and an orientation of edges in $\mathcal{E}_i^{\text{take-}}(\overline{P}')$, the set edges of $\mathcal{E}_i^{\text{take-}}$ with both endpoints in the uncontraction of $\overline{P}'$. We add all edges of $\mathcal{E}_i^{\text{take-}}(\overline{P}')$ to a set $\mathcal{E}_i^{(5B)}$ (which is initially empty). We then add all subgraphs in $\mathbb{C}_5(\overline{P}')$ to $\mathbb{X}_5^{\text{intrnl}}$.

The construction of Step 5B is described in the following lemma, which is a slight adaption of Lemma 6.17 in [50]. See Figure 11 for an illustration. The construction crucially exploit the fact that $d_{\mathcal{H}_i}(\nu, \mu) \leq (1 + 6g\epsilon)\omega(\mathbf{e})$.

Lemma 12.5 (Step 5B, Lemma 6.17 in [50]). Let $\overline{P}$ be a path in $\overline{F}_i^{(5)}$. Let $\mathcal{E}_i^{\text{take-}}(\overline{P})$ be the edges of $\mathcal{E}_i^{\text{take-}}$ with both endpoints in $\tilde{P}^{\text{uctrt}}$. We can construct a set of subgraphs $\mathbb{C}_5(\overline{P})$ such that:

- (1) Subgraphs in $\mathbb{C}_5(\overline{P})$ contain every node in $\tilde{P}^{\text{uctrt}}$.
- (2) For every subgraph $\mathcal{X} \in \mathbb{C}_4(\overline{P})$, $\zeta L_i \leq \text{Adm}(\mathcal{X}) \leq 5L_i$. Furthermore, $\mathcal{X}$ is a subtree of $\tilde{P}^{\text{uctrt}}$ and some edges in $\mathcal{E}_i^{\text{take-}}(\overline{P})$ whose both endpoints are in $\mathcal{X}$.
- (3) There is an orientation of edges in $\mathcal{E}_i^{\text{take-}}(\overline{P})$ such that, for any subgraph $\mathcal{X} \in \mathbb{C}_5(\overline{P})$, if the total number of out-going edges incident to nodes in $\mathcal{X}$ is t for any $t \geq 0$, then:

$$\Delta_{i+1}^+(\mathcal{X}) = \Omega(t\epsilon^2)L_i \quad (66)$$

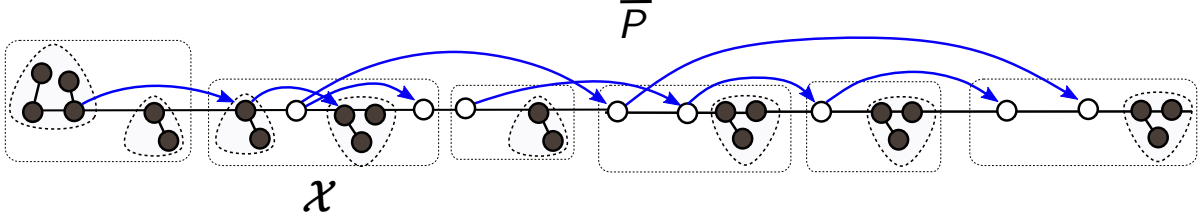


Figure 11: A path $\overline{P}$, a cluster $\mathcal{X}$, and a set of (blue) edges in $\mathcal{E}_i^{\text{take-}}(\overline{P})$. White nodes are uncontracted nodes and black nodes are those in contracted nodes (triangular shapes). $\Delta_{i+1}^+(\mathcal{X})$ is proportional to the number of out-going edges from nodes in $\mathcal{X}$, which is 3 in this case; there could be edges with both endpoints in $\mathcal{X}$.

We observe the following from the construction.

Observation 12.6. For every edge $\mathbf{e} \in \mathcal{E}_i^{\text{take-}}$, either at least one endpoint of $\mathbf{e}$ is in a subgraph in $\mathbb{X}_5^{\text{pref}}$, or both endpoints of $\mathbf{e}$ are in $\mathcal{E}_i^{(5B)}$.

The following lemma is analogous to Lemma 11.13.

Lemma 12.7. Every subgraph $\mathcal{X} \in \mathbb{X}_5^{\text{intrnl}} \cup \mathbb{X}_5^{\text{pref}}$ satisfies:

- (1) $\mathcal{X}$ is a subtree of $\widetilde{\text{MST}}_i$ if $\mathcal{X} \in \mathbb{X}_5^{\text{pref}}$.
- (2) $\zeta L_i \leq \text{Adm}(\mathcal{X}) \leq 20L_i$.
- (3) $|\mathcal{V}(\mathcal{X})| = \Omega(1/\epsilon)$.

Furthermore, if $\mathcal{X} \in \mathbb{X}_5^{\text{pref}}$, then $\mathcal{X}$ the uncontraction of a prefix/suffix subpath $\overline{Q}$ of a long path $\overline{P}$, and additionally, the (uncontraction of) other suffix $\overline{Q}'$ of $\overline{P}$ is augmented to a subgraph in $\mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4$, unless $\mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4 = \emptyset$.

In the next section, we prove Lemma 12.1.

12.2.1 Constructing $\mathbb{X}$ and the partition of $\mathcal{E}_i$: Proof of Lemma 12.1

We distinguish two cases:

Degenerate Case. The degenerate case is the case where $\mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4 = \emptyset$. In this case, we set $\mathbb{X} = \mathbb{X}^- = \mathbb{X}_5^{\text{intrnl}} \cup \mathbb{X}_5^{\text{pref}}$, and $\mathbb{X}^+ = \emptyset$.

Non-degenerate case. We define:

$$\begin{aligned}\mathbb{X}^+ &= \mathbb{X}_1 \cup \mathbb{X}_2 \cup \mathbb{X}_4 \cup \mathbb{X}_5^{\text{pref}}, \quad \mathbb{X}^- = \mathbb{X}_5^{\text{intrnl}} \\ \mathbb{X} &= \mathbb{X}^+ \cup \mathbb{X}^-\end{aligned}\tag{67}$$

Next, we construct the partition of $\{\mathcal{E}_i^{\text{take}}, \mathcal{E}_i^{\text{redund}}, \mathcal{E}_i^{\text{reduce}}\}$ of $\mathcal{E}_i$. Recall that we constructed two edge sets $\mathcal{E}_i^{\text{redund}}$ and $\mathcal{E}_i^{\text{take-}}$ above (Equation (65)). We then construct $\mathcal{E}_i^{\text{take}}$ as described below. It follows that $\mathcal{E}_i^{\text{reduce}} = \mathcal{E}_i \setminus (\mathcal{E}_i^{\text{take}} \cup \mathcal{E}_i^{\text{redund}})$.

Constructing $\mathcal{E}_i^{\text{take}}$: Let $\mathcal{V}_i^+ = \cup_{\mathcal{X} \in \mathbb{X}^+} \mathcal{V}(\mathcal{X})$ and $\mathcal{V}_i^- = \cup_{\mathcal{X} \in \mathbb{X}^-} \mathcal{V}(\mathcal{X})$. First, we add all edges in $\mathcal{E}_i^{\text{take-}}$ to $\mathcal{E}_i^{\text{take}}$. Next, we add $(\cup_{\mathcal{X} \in \mathbb{X}} \mathcal{E}(\mathcal{X}) \cap \mathcal{E}_i)$ to $\mathcal{E}_i^{\text{take}}$. Finally, for every edge $\mathbf{e} \in \mathcal{E}_i \setminus \mathcal{E}_i^{\text{redund}}$ such that $\mathbf{e}$ is incident to at least one node in $\mathcal{V}_i^-$, we add $\mathbf{e}$ to $\mathcal{E}_i^{\text{take}}$.

In the analysis below, we only explicitly distinguish the degenerate case from the non-degenerate case when it is necessary, i.e, in the proof Item (4) of Lemma 12.1. Otherwise, which case we are in is either implicit from the context, or does not matter.

We observe that Item (2) in Lemma 12.1 follows directly from the construction of $\mathcal{E}_i^{\text{redund}}$. Henceforth, we focus on proving other items of Lemma 12.1. We first show Item (5).

Lemma 12.8. *Let $\mathbb{X}$ be the subgraph as defined in Equation (67). For every subgraph $\mathcal{X} \in \mathbb{X}$, $\mathcal{X}$ satisfies the three properties (P1')-(P3') with $g = 31$. Consequently, Item (5) of Lemma 12.1 holds.*

Proof: We observe that property (P1') follows directly from the construction. Property (P2') follows directly from Lemmas 11.5, 11.7, 12.7 and 11.10. We now bound $\text{Adm}(\mathcal{X})$. The lower bound on $\text{Adm}(\mathcal{X})$ follows directly from Item (3) of Lemma 11.5, Items (2) of Lemmas 11.7, 12.7 and 11.10. For the upper bound, by the same argument in Lemma 8.10, if $\mathcal{X}$ is initially formed in Steps 1-4, then $\text{Adm}(\mathcal{X}) \leq 31L_i$. Otherwise, by Lemma 12.7, $\text{Adm}(\mathcal{X}) \leq 5L_i$, which implies property (P3') with $g = 31$. $\square$

We observe that Lemma 11.17 and Lemma 11.18 holds for $\mathbb{X}^+$, which we restate below in Lemma 12.9 and Lemma 12.10, respectively. In particular, Lemma 12.10 implies Item (3) of Lemma 12.1.

Lemma 12.9. *For any subgraph $\mathcal{X} \in \mathbb{X}$ such that $|\mathcal{V}(\mathcal{X})| \geq \frac{2g}{\zeta\epsilon}$ or $\Delta_{i+1}^+(\mathcal{X}) = \Omega(L_i)$, then $\Delta_{i+1}^+(\mathcal{X}) = \Omega(\epsilon L_i |\mathcal{V}(\mathcal{X})|)$.*

Lemma 12.10. $\Delta_{i+1}^+(\mathcal{X}) \geq 0$ for every $\mathcal{X} \in \mathbb{X}$ and

$$\sum_{\mathcal{X} \in \mathbb{X}^+} \Delta_{i+1}^+(\mathcal{X}) = \sum_{\mathcal{X} \in \mathbb{X}^+} \Omega(|\mathcal{V}(\mathcal{X})| \epsilon L_i).$$

We now prove Item (1) of Lemma 12.1, which we restate here for convenience.

Lemma 12.11. *For every subgraph $\mathcal{X} \in \mathbb{X}$, $\deg_{\mathcal{G}_i^{\text{take}}}(\mathcal{V}(\mathcal{X})) = O(|\mathcal{V}(\mathcal{X})|/\epsilon)$ where $\mathcal{G}_i^{\text{take}} = (\mathcal{V}_i, \mathcal{E}_i^{\text{take}})$, and $\mathcal{E}(\mathcal{X}) \cap \mathcal{E}_i \subseteq \mathcal{E}_i^{\text{take}}$. Furthermore, if $\mathcal{X} \in \mathbb{X}^-$, there is no edge in $\mathcal{E}_i^{\text{reduce}}$ incident to a node in $\mathcal{X}$.*

Proof: Let $\mathcal{V}_i^{\text{high}^+} = \cup_{\mathcal{X} \in \mathbb{X}_1} \mathcal{X}$. Note by the construction in Step 1 (Lemma 11.5), nodes in $\mathcal{V}_i \setminus \mathcal{V}_i^{\text{high}^+}$ have degree $O(\frac{1}{\epsilon})$. Let $\mathcal{E}_i^{(1)}$ be the set of edges in $\mathcal{E}_i^{\text{take}}$ with both endpoints in $\mathcal{V}_i^{\text{high}^+}$ and $\mathcal{E}_i^2 = \mathcal{E}_i^{\text{take}} \setminus \mathcal{E}_i^{(1)}$. Also by the construction in Step 1 (Lemma 11.5), both endpoints of every edge in $\mathcal{E}_i^2$ have degree $O(1/\epsilon)$. Thus, for any $\mathcal{X} \in \mathbb{X}$, the number of edges in $\mathcal{E}_i^{(2)}$ incident to nodes in $\mathcal{X}$ is $O(|\mathcal{V}(\mathcal{X})|/\epsilon)$.

Next, we consider $\mathcal{E}_i^{(1)}$. Observe by the construction of $\mathcal{E}_i^{\text{take}}$ that there is no edge in $\mathcal{E}_i^{(1)}$ with two endpoints in two different graphs of $\mathbb{X}_1$. Furthermore, since $\mathcal{X}$ is a tree for every subgraph $\mathcal{X} \in \mathbb{X}_1$, the

number of edges in $\mathcal{E}_i^{(1)}$ incident to nodes in $\mathcal{X}$ is $O(|\mathcal{V}(\mathcal{X})|)$. This bound also holds for every subgraph $\mathcal{X}$ not in $\mathbb{X}_1$ since the number of incident edges in $\mathcal{E}_i^{(1)}$ is 0; this implies the claimed bound on $\deg_{\mathcal{G}_i^{\text{take}}}(\mathcal{V}(\mathcal{X}))$.

For the last claim, we observe that nodes in subgraphs of $\mathbb{X}^-$ are in $\mathcal{V}_i^-$. Thus, by the construction of $\mathcal{E}_i^{\text{take}}$, every edge incident to a node of $\mathcal{X} \in \mathbb{X}^-$ is either in $\mathcal{E}_i^{\text{take}}$ or $\mathcal{E}_i^{\text{redund}}$. $\square$

We now focus on proving Item (4) of Lemma 12.1 which we restate below.

Lemma 12.12. *There exists an orientation of edges in $\mathcal{E}_i^{\text{take}}$ such that for every subgraph $\mathcal{X} \in \mathbb{X}^-$, if $\mathcal{X}$ has t out-going edges for some $t \geq 0$, then $\Delta_{i+1}^+(\mathcal{X}) = \Omega(|\mathcal{V}(\mathcal{X})|t\epsilon^2 L_i)$, unless a degenerate case happens, in which $\mathcal{E}_i^{\text{reduce}} = \emptyset$ and*

$$\omega(\mathcal{E}_i^{\text{take}}) = O\left(\frac{1}{\epsilon^2}\right)\left(\sum_{\mathcal{X} \in \mathbb{X}} \Delta_{i+1}^+(\mathcal{X}) + L_i\right).$$

Proof: First, we consider the non-degenerate case. Recall that $\{\mathcal{V}_i^+, \mathcal{V}_i^-\}$ is a partition of $\mathcal{V}_i$ in the construction of $\mathcal{E}_i^{\text{take}}$. We orient edges of $\mathcal{E}_i^{\text{take}}$ as follows.

First, for any $\mathbf{e} = (\mu, \nu) \in \mathcal{E}_i^{\text{take}}$ such that at least one endpoint, say $\mu \in \mathcal{V}_i^+$, we orient $\mathbf{e}$ as out-going from μ . (If both μ, ν are in $\mathcal{V}_i^+$, we orient $\mathbf{e}$ arbitrarily). Remaining edges are subsets of $\mathcal{E}_i^{(5B)}$ by Observation 12.6. We orient edges in $\mathcal{E}_i^{(5B)}$ as in the construction of Step 5B. For every subgraph $\mathcal{X} \in \mathbb{X}^-$, by construction, out-going edges incident to nodes in $\mathcal{X}$ are in $\mathcal{E}_i^{(5B)}$. By Item (3) of Lemma 12.5, $\Delta_{i+1}^+(\mathcal{X}) = \Omega(|\mathcal{V}(\mathcal{X})|t\epsilon^2 L_i)$.

It remains to consider the degenerate case. In this case, by the same argument in Lemma 11.28, $\overline{F}_i^{(5)} = \overline{F}_i^{(4)} = \overline{F}_i^{(3)}$, and $\overline{F}_i^{(5)}$ is a single (long) path. Furthermore, $\mathcal{E}_i^{\text{take}} = \mathcal{E}_i^{\text{take-}}$, and $|\mathbb{X}_5^{\text{pref}}| = 2$. We orient edges in $\mathcal{E}_i^{(5B)}$ as in the construction of Step 5B, and other edges of $\mathcal{E}_i^{\text{take}}$, which must be incident to nodes in subgraphs of $\mathbb{X}_5^{\text{pref}}$, are oriented as out-going from subgraphs in $\mathbb{X}_5^{\text{pref}}$. By Item (3) of Lemma 12.1, for any subgraph $\mathcal{X} \in \mathbb{X}_5^{\text{intrnl}}$ that has t out-going edges, the total weight of the out-going edges is at most $tL_i = O(1/\epsilon)\Delta_{i+1}^+(\mathcal{X})$. Thus, $\omega(\mathcal{E}_i^{(5B)}) = O(1/\epsilon)\sum_{\mathcal{X} \in \mathbb{X}_5^{\text{intrnl}}} \Delta_{i+1}^+(\mathcal{X}) = O(1/\epsilon^2)\sum_{\mathcal{X} \in \mathbb{X}} \Delta_{i+1}^+(\mathcal{X})$.

It remains to consider edges incident to at least one node in a subgraph in $\mathbb{X}_5^{\text{pref}}$ by Observation 12.6. Let $\mathcal{X} \in \mathbb{X}_5^{\text{pref}}$. Observe that if $|\mathcal{V}(\mathcal{X})| \geq \frac{2g}{\zeta\epsilon}$, then

$$\begin{aligned} \Delta_{i+1}^+(\mathcal{X}) &= O(|\mathcal{V}(\mathcal{X})|\epsilon L_i) \quad (\text{by Lemma 12.9}) \\ &= O(|\deg_{\mathcal{G}_i^{\text{take}}}(\mathcal{X})|\epsilon^2 L_i) \quad (\text{by Item (1) of Lemma 12.1}) \end{aligned}$$

Otherwise, $|\mathcal{V}(\mathcal{X})| \leq \frac{2g}{\zeta\epsilon}$, and hence $|\deg_{\mathcal{G}_i^{\text{take}}}(\mathcal{X})| = O(1/\epsilon^2)$ by Item (1) of Lemma 12.1. This implies that the total weight of edges incident to $\mathcal{X}$ is at most $|\deg_{\mathcal{G}_i^{\text{take}}}(\mathcal{X})|L_i = O(1/\epsilon^2)(\Delta_{i+1}^+(\mathcal{X}) + L_i)$. Since $|\mathbb{X}_5^{\text{pref}}| = 2$ and $\Delta_{i+1}^+(\mathcal{X}) \geq 0$ for every $\mathcal{X} \in \mathbb{X}$ by Item (3) of Lemma 12.1, we have that the total weight of edges incident to at least one node in a subgraph in $\mathbb{X}_5^{\text{pref}}$ is $O(\frac{1}{\epsilon^2})(\sum_{\mathcal{X} \in \mathbb{X}} \Delta_{i+1}^+(\mathcal{X}) + L_i)$. The lemma now follows. $\square$

Acknowledgement. Hung Le is supported by a start up funding of University of Massachusetts at Amherst and by the National Science Foundation under Grants No. CCF-2121952 and No. CCF-2237288. Shay Solomon is partially supported by the Israel Science Foundation grant No.1991/19. We thank Oded Goldreich for his suggestions concerning the presentation of this work and we thank Lazar Milenković for his support.

References

- [1] S. Alstrup, S. Dahlgaard, A. Filtser, M. Stöckel, and C. Wulff-Nilsen. Constructing light spanners deterministically in near-linear time. In *27th Annual European Symposium on Algorithms (ESA 2019)*, pages 4:1–4:15, 2019. 2, 4
- [2] I. Althöfer, G. Das, D. Dobkin, D. Joseph, and J. Soares. On sparse spanners of weighted graphs. *Discrete Computational Geometry*, 9(1):81–100, 1993. 2, 3, 4, 25, 53
- [3] S. Arya, G. Das, D. M. Mount, J. S. Salowe, and M. Smid. Euclidean spanners: Short, thin, and lanky. In *Proceedings of the Twenty-seventh Annual ACM Symposium on Theory of Computing*, STOC '95, pages 489–498, 1995. , 2
- [4] B. Awerbuch, M. Luby, A. V. Goldberg, and S. A. Plotkin. Network decomposition and locality in distributed computation. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, FOCS '89, pages 364–369, 1989. 10, 28
- [5] Surender Baswana and Sandeep Sen. A simple and linear time randomized algorithm for computing sparse spanners in weighted graphs. *Random Structures & Algorithms*, 30(4):532–563, 2007. 2
- [6] M. Ben-Or. Lower bounds for algebraic computation trees. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing*, STOC'83, pages 80–86, 1983. 2
- [7] S. Bhore and C. D. Tóth. Light Euclidean Steiner Spanners in the Plane. In *37th International Symposium on Computational Geometry*, SoCG '21, pages 15:1–15:17, 2021. 3
- [8] S. Bhore and C. D. Tóth. On Euclidean Steiner $(1 + \epsilon)$ -Spanners. In *38th International Symposium on Theoretical Aspects of Computer Science (STACS 2021)*, pages 13:1–13:16, 2021. 3
- [9] G. Bodwin. Personal communication, 2022. 6
- [10] G. Borradaile, H. Le, and C. Wulff-Nilsen. Minor-free graphs have light spanners. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science*, FOCS '17, pages 767–778, 2017. , 1, 2, 5, 10, 28, 42
- [11] G. Borradaile, H. Le, and C. Wulff-Nilsen. Greedy spanners are optimal in doubling metrics. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '19, pages 2371–2379, 2019. 10, 28
- [12] Milutin Brankovic, Joachim Gudmundsson, and André van Renssen. Local routing in a tree metric 1-spanner. In *COCOON*, volume 12273 of *Lecture Notes in Computer Science*, pages 174–185. Springer, 2020. 2
- [13] D. De Caen and L. A. Székely. *The maximum size of 4-and 6-cycle free bipartite graphs on m, n vertices*. Rheinische Friedrich-Wilhelms-Universität, 1991. 6
- [14] T.-H. Hubert Chan, Anupam Gupta, Bruce M. Maggs, and Shuheng Zhou. On hierarchical routing in doubling metrics. *ACM Trans. Algorithms*, 12(4):55:1–55:22, 2016. Preliminary version appeared in SODA 2005. 2
- [15] B. Chandra, G. Das, G. Narasimhan, and J. Soares. New sparseness results on graph spanners. In *Proceedings of the Eighth Annual Symposium on Computational Geometry*, 1992. , 2, 3, 10, 28

- [16] B. Chazelle. A minimum spanning tree algorithm with inverse-ackermann type complexity. *Journal of the ACM*, 47(6):1028–1047, 2000. 41
- [17] S. Chechik and C. Wulff-Nilsen. Near-optimal light spanners. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA’16, pages 883–892, 2016. , 1, 3, 4, 10, 28
- [18] D. Z. Chen, G. Das, and M. Smid. Lower bounds for computing geometric spanners and approximate shortest paths. *Discrete Applied Mathematics*, 110(2-3):151–167, 2001. 2
- [19] L. P. Chew. There is a planar graph almost as good as the complete graph. In *Proceedings of the Second Annual Symposium on Computational Geometry*, SCG ‘86, pages 169–177, 1986. 1
- [20] L. P. Chew. There are planar graphs almost as good as the complete graph. *Journal of Computer and System Sciences*, 39(2):205 – 219, 1989. 1
- [21] David Conlon, Jacob Fox, Benny Sudakov, and Yufei Zhao. The regularity method for graphs with few 4-cycles. *Journal of the London Mathematical Society*, 104(5):2376–2401, 2021. 6
- [22] Artur Czumaj and Andrzej Lingas. Fast approximation schemes for euclidean multi-connectivity problems. In *International Colloquium on Automata, Languages, and Programming*, pages 856–868. Springer, 2000. 2
- [23] G. Das, P. Heffernan, and G. Narasimhan. Optimally sparse spanners in 3-dimensional euclidean space. In *Proceedings of the 9th Annual Symposium on Computational Geometry*, SCG ’93, pages 53–62, 1993. 2, 11
- [24] G. Das, G. Narasimhan, and J. Salowe. A new way to weigh malnourished euclidean graphs. In *Proceedings of the 6th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA ’95, pages 215–222, 1995. 2, 11
- [25] Gautam Das and Paul J Heffernan. Constructing degree-3 spanners with other sparseness properties. *International Journal of Foundations of Computer Science*, 7(02):121–135, 1996. 2, 11
- [26] Gautam Das and Giri Narasimhan. A fast algorithm for constructing sparse euclidean spanners. In *Proceedings of 10th SoCG*, pages 132–139, 1994. , 1, 2, 11
- [27] Erik D. Demaine, MohammadTaghi Hajiaghayi, and Ken-ichi Kawarabayashi. Contraction decomposition in h-minor-free graphs and algorithmic applications. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 441–450, 2011. 5
- [28] M. Elkin and O. Neiman. Efficient algorithms for constructing very sparse spanners and emulators. *ACM Transactions on Algorithms*, 15(1), 2018. Announced at SODA ‘17. 4
- [29] M. Elkin and S. Solomon. Fast constructions of lightweight spanners for general graphs. *ACM Transactions on Algorithms*, 12(3), 2016. 10, 28
- [30] Michael Elkin, Ofer Neiman, and Shay Solomon. Light spanners. In *Proc. of 41th ICALP*, pages 442–452, 2014. 2, 3, 10, 28
- [31] Michael Elkin and Shay Solomon. Optimal euclidean spanners: Really short, thin, and lanky. *Journal of the ACM (JACM)*, 62(5):1–45, 2015. 2

- [32] M. Farshi and A. Poureidi. A lower bound for computing geometric spanners. *Computational Geometry*, 53:21–26, 2016. 2
- [33] Arnold Filtser and Ofer Neiman. Light spanners for high dimensional norms via stochastic decompositions. In *26th Annual European Symposium on Algorithms, ESA 2018, August 20-22, 2018, Helsinki, Finland*, pages 29:1–29:15, 2018. 6, 25, 26
- [34] Arnold Filtser and Shay Solomon. The greedy spanner is existentially optimal. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*, PODC '16, pages 9–17, 2016. to appear in SICOMP 2020. 3
- [35] M. Fürer and S. P. Kasiviswanathan. Approximate distance queries in disk graphs. In *International Workshop on Approximation and Online Algorithms*, WAOA '06, pages 174–187, 2006. 3, 17
- [36] Martin Fürer and Shiva Prasad Kasiviswanathan. Spanners for geometric intersection graphs with applications. *J. Comput. Geom.*, 3(1):31–64, 2012. 17
- [37] Lee-Ad Gottlieb and Liam Roditty. Improved algorithms for fully dynamic geometric spanners and geometric routing. In *SODA*, pages 591–600. SIAM, 2008. 2
- [38] M. Grigni and P. Sissokho. Light spanners and approximate TSP in weighted graphs with forbidden minors. In *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '02, pages 852–857, 2002. 2, 4, 5
- [39] Michelangelo Grigni. Approximate TSP in graphs with forbidden minors. In *Automata, Languages and Programming, 27th International Colloquium, ICALP 2000, Geneva, Switzerland, July 9-15, 2000, Proceedings*, pages 869–877, 2000. 2, 4
- [40] J. Gudmundsson, C. Levcopoulos, and G. Narasimhan. Fast greedy algorithms for constructing sparse geometric spanners. *SIAM J. Comput.*, 31(5):1479–1500, 2002. , 2, 11
- [41] S. Halperin and U. Zwick. Linear time deterministic algorithm for computing spanners for unweighted graphs, 1996. Manuscript. 2, 20
- [42] Y. Han and M. Thorup. Integer sorting in $o(n\sqrt{\log\log n})$ expected time and linear space. In *Proceedings of the 43rd Annual Symposium on Foundations of Computer Science*, FOCS '02. 5
- [43] S. Har-Peled, P. Indyk, and A. Sidiropoulos. Euclidean spanners in high dimensions. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '13, pages 804–809, 2013. 6
- [44] Shlomo Hoory and Nathan Linial. Colorings of the d -regular infinite tree. *Journal of Combinatorial Theory, Series B*, 91(2):161–167, 2004. 6
- [45] A. V. Kostochka. The minimum Hadwiger number for graphs with a given mean degree of vertices. *Metody Diskret. Analiz.*, (38):37–58, 1982. 22
- [46] Robert Krauthgamer, Huy L. Nguyen, and Tamar Zondiner. Preserving terminal distances using minors. *SIAM J. Discrete Math.*, 28(1):127–141, 2014. 24
- [47] H. Le and S. Solomon. Light euclidean spanners with steiner points. In *28th Annual European Symposium on Algorithms*, 2020. Full version at <https://arxiv.org/pdf/2007.11636.pdf>. 3, 22

- [48] H. Le and S. Solomon. Near-optimal spanners for general graphs in (nearly) linear time. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '22, pages 3332–3361. 2022. 4
- [49] Hung Le. A PTAS for subset TSP in minor-free graphs. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms*, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020, pages 2279–2298, 2020. Full version: arxiv:1804.01588. 23, 24
- [50] Hung Le and Shay Solomon. Truly optimal euclidean spanners. In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 1078–1100, 2019. Full version at <https://arxiv.org/abs/1904.12042>. 2, 3, 10, 23, 26, 28, 58, 59, 70
- [51] C. Levkopoulos and A. Lingas. There are planar graphs almost as good as the complete graphs and as short as minimum spanning trees. In *International Symposium on Optimal Algorithms*, pages 9–13, 1989. 2
- [52] M. Mareš. Two linear time algorithms for mst on minor closed graph classes. *Archivum Mathematicum*, 40(3):315–320, 2004. 52
- [53] G. Narasimhan and M. Smid. *Geometric Spanner Networks*. Cambridge University Press, 2007. , 2, 11, 17, 18, 20
- [54] D. Peleg and A. A. Schäffer. Graph spanners. *Journal of Graph Theory*, 13(1):99–116, 1989. 1
- [55] D. Peleg and J. D. Ullman. An optimal synchronizer for the hypercube. *SIAM J. Comput.*, 18(4):740–747, 1989. 1
- [56] S. B. Rao and W. D. Smith. Approximating geometrical graphs via “spanners” and “banyans”. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, STOC '98, pages 540–550, 1998. Full version at <http://graphics.stanford.edu/courses/cs468-06-winter/Papers/rs-tsp.pdf>. 2
- [57] Liam Roditty, Mikkel Thorup, and Uri Zwick. Deterministic constructions of approximate distance oracles and spanners. In *International Colloquium on Automata, Languages, and Programming*, pages 261–272. Springer, 2005. 2
- [58] Robert E. Tarjan. Efficiency of a good but not linear set union algorithm. *Journal of the ACM*, 22(2):215–225, 1975. 36
- [59] A. Thomason. An extremal function for contractions of graphs. *Mathematical Proceedings of the Cambridge Philosophical Society*, 95(2):261–265, 1984. 22