New Additive Spanners

Shiri Chechik *

Abstract

This paper considers additive and purely additive spanners. We present a new purely additive spanner of size $\tilde{O}(n^{7/5})$ with additive stretch 4. This construction fills in the gap between the two existing constructions for purely additive spanners, one for 2-additive spanner of size $O(n^{3/2})$ and the other for 6-additive spanner of size $O(n^{4/3})$, and thus answers a main open question in this area. In addition, we present a construction for additive spanners with $\tilde{O}(n^{1+\varepsilon})$ edges and additive stretch of $\tilde{O}(n^{1/2-3\delta/2})$ for any $3/17 \leq \delta < 1/3$, improving the stretch of the existing constructions from $O(n^{1-3\delta})$ to $\tilde{O}(\sqrt{n^{1-3\delta}})$. Finally, we show that our $(1, n^{1/2-3\delta/2})$-spanner construction can be tweaked to give a sublinear additive spanner of size $\tilde{O}(n^{1+3\delta/17})$ with additive stretch $O(\sqrt{\text{distance}})$.

1 Introduction

Graph spanners are sparse subgraphs that faithfully preserve the pairwise distances of a given graph. Formally, an $(\alpha, \beta)$-spanner of a graph $G = (V, E)$ is a subgraph $H$ such that for any pair of nodes $s, t$, $\text{dist}(s, t, H) \leq \alpha \cdot \text{dist}(s, t, G) + \beta$, where $\text{dist}(s, t, H')$ for a subgraph $H'$ is the distance from $s$ to $t$ in $H'$. If $\alpha = 1$ we say that the spanner is additive and if in addition $\beta = O(1)$, we say that the spanner is purely additive. If $\beta = 0$ we say that the spanner is multiplicative, otherwise we say that the spanner is mixed.

Graph spanners were extensively studied since they were first introduced in [19, 20] in the late 80’s. Many distributed applications use spanners as a key ingredient, e.g., synchronizers [20], compact routing schemes [21, 9, 26, 10, 25], distance oracles [3, 27], broadcasting [18], near-shortest path algorithms [12, 13, 16], etc.

Much of the work on spanners considers multiplicative spanners. It is well-known that one can efficiently construct a $(2k-1, 0)$-spanner with $O(n^{1+k})$ edges [2]. This size-stretch ratio is conjectured to be tight based on the girth conjecture of Erdős [17]. The girth conjecture has been proved for the specific cases of $k = 1, 2, 3,$ and 5 [29].

Although many papers considered additive spanners or mixed spanners, several key questions in this area remain open. The girth conjecture applies only to short distances. In particular, it does not contradict the existence of $(1, 2k-2)$-spanners of size $O(n^{1+1/k})$, or any $(\alpha, \beta)$-spanners of size $O(n^{1+1/k})$ such that $\alpha + \beta = 2k - 1$ with $\alpha \geq 1$ and $\beta > 0$. The first construction for purely additive spanners was presented by Aingworth et al. [1]. They show how to efficiently construct a $(1, 2)$ spanner, or a 2-additive spanner for short, with $O(n^{3/2})$ edges (see [11, 15, 28, 24] for further follow-up). Later, an efficient construction for 6-additive spanners with $O(n^{1/3})$ edges was presented by Baswana et al. [4, 5]. Woodruff [31] later presented a different construction for 6-additive spanners with $\tilde{O}(n^{4/3})$ edges with better construction time. These are the only two purely additive spanners known so far. A major open problem in this field concerns the existence of purely additive spanners with $O(n^{1+\varepsilon})$ edges for any fixed $\varepsilon > 0$. Woodruff [30] showed a lower bound for additive spanners matching the girth conjecture bounds but independent of the correctness of the conjecture. More precisely, he showed the existence of graphs for which any spanner of size $O(k^{-1}n^{1+k})$ has an additive stretch of at least $2k - 1$.

In the absence of additional purely additive spanners or impossibility results, attempts were made to seek spanners with either non-constant additive stretch or a mix of both multiplicative and additive stretch (see, e.g., [15, 28, 23, 5]).

Bollobás et al. [6] presented efficient constructions for a spectrum of additive spanners with additive stretch that depends on $n$. More precisely, they show how to efficiently construct a $(1, n^{1-2\delta})$-spanner with $O(2^{1/\delta}n^{1+\varepsilon})$ edges for any $\delta > 0$. This additive stretch was later improved to $(1, n^{1-3\delta})$ by Baswana et al. in [4, 5] and to $(1, n^{9/16-7\delta/8})$ by Pettie [22, 23] (the latter is smaller than the former for every $\delta < 7/34$). In addition, sublinear additive spanners, namely, additive spanners with stretch that is sublinear in the distances, were also considered. Thorup and Zwick [28] showed how to construct a spanner of size $O(kn^{1+1/k})$ such that for every pair of nodes $s$ and $t$, the additive stretch is $O(d^{1/k} + 2k)$, where $d = \text{dist}(s, t)$. Pettie [22, 23] later improved that result presenting an efficient span-
ner construction of size $O(kn^{1+\frac{1+\frac{1-(3+3\sqrt{3})}{2}}{2+3\sqrt{3}+k^2}})$ with additive stretch of $O(kd^{1-\frac{1}{k}} + k^3)$, where $d = \text{dist}(s,t)$. Specifically, for $k = 2$, the size of the spanner is $O(n^{5/3})$ and the additive stretch is $O(\sqrt{n})$. For further results on mixed spanners see [14, 15, 4, 28, 22, 23, 5].

This paper considers additive and purely additive spanners. We make an additional step towards better understanding the picture of purely additive spanners, by presenting a new simple algorithm for $(1,4)$-additive spanners with $O(n^{7/5})$ edges. We thus answer one of the main open questions in this area of purely additive spanners, by filling in the gap between the two existing constructions.

In addition, we present a construction for additive spanners with $O(n^{1+\delta})$ edges and additive stretch of $O(n^{1/2-\frac{3\delta}{5}})$ for any $3/17 \leq \delta < 1/3$. We thus decrease the stretch for this range to the root of the best known additive stretch so far. We note that it is possible to extend this range a little bit (to $\delta$ values smaller than $3/17$) but the construction and analysis become much more complex. It would be interesting to see if this range can be extended all the way, to any $0 \leq \delta < 1/3$. Our construction for $(1, n^{1/2-\frac{3\delta}{5}})$-spanners with $O(n^{1+\delta})$ edges is quite involved and requires a number of new ideas. The construction consists of several procedures, where each procedure provides certain desired properties and may be of independent interest. Finally, we show that our $(1, n^{1/2-\frac{3\delta}{5}})$-spanner construction can be tweaked to slightly improve the size of the sublinear additive spanner of Pettie [22, 23] with additive stretch $O(\sqrt{n})$ from $O(n^{1+\delta/5})$ to $O(n^{1+\delta/17})$.

$2$ $\tilde{O}(n^{7/5})$ edge spanners with additive stretch $4$

In this section we present a new construction for a $(1, 4)$-spanner with $O(n^{7/5}\log^{1/5} n)$ edges. Here and throughout, $n = |V|$ and $m = |E|$. Let us introduce some preliminaries. Denote the vertex set and edge set of a subgraph $H$ by $V(H)$ and $E(H)$, respectively. For nodes $x, y \in V$ and subgraph $H$, $\text{dist}(x, y, H)$ is the distance between $x$ and $y$ in $H$. For a node $x \in V$, a set of nodes $S \subseteq V$ and subgraph $H$, $\text{dist}(x, S, H)$ is the distance between $x$ and the node $y \in S$ closest to $x$ in $H$.

For a node $x \in V$, we denote $\Gamma(x)$ the subgraph of $H$ induced by $\Gamma(x)$, namely, $\Gamma(x, r, H) = \{v \in V : \text{dist}(x, v, H) \leq r\}$, and let $\Gamma^\ast(x, r, H) = \{v \in V : \text{dist}(x, v, H) = r\}$. Similarly, for a path $P$, an integer $r$ and a subgraph $H$, denote the set of neighbors of $P$ by $\Gamma(P, r, H) = \{v \in V : \text{dist}(v, V(H), H) \leq r\}$. To simplify notation, when $H = G$ and/or when $r = 1$ we omit them. Let $|P|$ denote the number of edges in $P$.

Let $\text{deg}(v)$ for a node $v$ be its degree. We say that a node is heavy if its degree is at least $\mu = \lfloor n^{2/5}/\log^{1/5} n \rfloor$, and light otherwise. For every pair of nodes $s$ and $t$, select a shortest path $P(s, t)$ from $s$ to $t$ in $G$ and let $P = \{P(s, t) : s, t \in V\}$. Let $P(V_1, V_2) = \{P(s, t) : s \in V_1, t \in V_2\}$ for subsets of nodes $V_1$ and $V_2$. The heavy distance between $s$ and $t$, denoted $\text{heavy}_\text{dist}(s, t, G)$, is defined to be the number of heavy nodes on the path $P(s, t)$. Similarly, for a path $P$, denote by $\text{heavy}_\text{dist}(P, G)$ the number of heavy nodes on the path $P$.

We now turn to describe our $(1, 4)$-spanner construction. Initially set $H$ to be $(V, \emptyset)$. The construction consists of three stages. In the first stage, add to $H$ all edges incident to light nodes. In the second stage, randomly select a set of nodes $S_1$ of expected size $9\mu$, by choosing every node from $V$ independently at random with probability $9\mu/n$. For every node $x \in S_1$, construct a BFS tree $T(x)$ rooted at $x$ spanning all vertices $V_x$, and add the edges of $T(x)$ to $H$.

In the third stage, and final stage, choose a set $S_2$ of $n/\mu$ nodes in expectation, called hereafter center clusters. This can be done by choosing each node independently at random with probability $1/\mu$. Next, for each heavy node $x$ such that none of the nodes in $\{x\} \cup \Gamma(x)$ were chosen to $S_2$, add all incident edges of $x$ to $H$. For each node $x \in S_2$, create a cluster $C(x)$, initially set to $\{x\}$. For every heavy node $v$ such that $v \notin S_2$ and $\Gamma(v) \cap S_2 \neq \emptyset$, arbitrarily choose one node $x$ in $\Gamma(v) \cap S_2$, add $v$ to $x$’s cluster $C(x)$ and add the edge $(v, x)$ to $H$. Finally, for each pair of nodes $x_1$ and $x_2$ in $S_2$ do the following. Consider all shortest paths $P(y_1, y_2) \in P(C(x_1), C(x_2))$ such that $\text{heavy}_\text{dist}(y_1, y_2, G) \leq \mu^3/n$, namely, all shortest paths $P(y_1, y_2) \in P$ such that $y_1 \in C(x_1)$, $y_2 \in C(x_2)$ and $\text{heavy}_\text{dist}(y_1, y_2, G) \leq \mu^3/n$. Choose the path $P(y_1, y_2)$ with minimal length $|P(y_1, y_2)|$, and add it to $H$.

This completes the description of our spanner construction. See Procedure 4-Additive-Spanner for the formal code.

We now bound the number of edges in the resulting spanner $H$.

**Lemma 2.1.** The expected number of edges in $H$ is $O(n\mu) = \tilde{O}(n^{7/5})$.

**Proof:** Let us bound the number of edges added to $H$ in the three different stages. In the first stage, only edges adjacent to light nodes were added. Each such light node contributes at most $\mu$ edges, so at most $n\mu$ edges were added to $H$ in this stage.

In the second stage, each node is added to $S_1$ with probability $9\mu/n$. Therefore, the expected number of nodes in $S_1$ is $9\mu$. For each node in $S_1$, a BFS tree of $n-1$ edges is added to $H$. Hence the expected number
of edges added in the second stage is $O(n\mu)$.

We now turn to analyze the expected number of edges added in the third stage. In the first part of the third stage, for every heavy node $v \notin S_2$ we either add $v$ to one of the clusters $C(x)$ and then add the edge $(v, x)$ to $H$, or (in case $v$ remains unclustered, as $(\{v\} \cup \Gamma(v)) \cap S_2 = \emptyset$) we add to $H$ all $\deg(v)$ edges adjacent to $v$. The probability that a node $v$ will be unclustered, and thus all of its edges will be added, is $(1 - 1/\mu)^{\deg(v)}$. We get that the expected number of edges added for a node $v$ is at most $1 + \deg(v)(1 - 1/\mu)^{\deg(v)} < \mu$. Finally, the expected number of clusters is $n/\mu$, therefore the number of cluster pairs is $n^2/\mu^2$. For each such pair, we add a path $P = P(y_1, y_2) \in \mathcal{P}$ of heavy distance $\text{heavy dist}(y_1, y_2, G) \leqslant \mu^3/n$. Note that all edges of the path that are adjacent to light nodes were already added to $H$ on the first stage, and as there are at most $\mu^3/n$ heavy nodes on $P$, at most $\mu^3/n$ edges are added for the path $P$ on the third stage. We conclude that the number of edges added for all cluster pairs is $O(n^2/\mu^2) \cdot (\mu^3/n)) = O(n\mu) = O(n^{7/5})$.

Next, we show that the additive stretch of the result spanner is indeed at most 4.

**Lemma 2.2.** For every two nodes $s$ and $t$, $\text{dist}(s, t, H) \leqslant \text{dist}(s, t, G) + 4$ with probability at least $1 - 1/n^3$.

**Proof:** Consider two nodes $s$ and $t$. A node is said to be covered by the spanner $H$ if all its adjacent edges are in $H$. Notice that it is enough to prove the lemma for pairs of nodes $s$ and $t$ that are both uncovered. To see this, let $s'$ be the first uncovered node on the path $P(s, t)$ and let $t'$ be the last such node. Note that all edges from $s$ to $s'$ on the path $P(s, t)$ and all edges from $t'$ to $t$ on $P(s, t)$ are in $H$. Therefore, if the lemma holds for $s'$ and $t'$, namely, with probability at least $1 - 1/n^3$, $\text{dist}(s, t, H) \leqslant \text{dist}(s, t, G) + 4$, then it holds for $s, t$ with probability at least $1 - 1/n^3$.

So we assume now that $s$ and $t$ are uncovered.

We consider two cases and prove the claim separately for each case. The first case is when $\text{heavy dist}(s, t, G) > \mu^3/n$. Note that since $P(s, t)$ is a shortest path in $G$, necessarily every node $v \in V$ can have at most three neighbors in $P(s, t)$. Combining this with the fact that the number of heavy nodes in $P(s, t)$ is more than $\mu^3/n$, and hence the sum of their degrees is more than $\mu^4/n$, we get that $|\Gamma(P(s, t))| > \mu^3/(3n)$. We claim that the probability that $\Gamma(P(s, t)) \cap S_1 \neq \emptyset$
is at least $1-1/n^3$, as

$$\mathbb{P}(\Gamma(P(s,t)) \cap S_1 = \emptyset) \leq (1 - 9\mu/n)^{4/(3n)} \leq (1 - 9\log^{1/3}/n^{3/5}((n^{3/5}/(9\log 1/\delta)n))logs \approx 1/n^3.$$ 

We now claim that if $\Gamma(P(s,t)) \cap S_1 \neq \emptyset$ then $\text{dist}(s,t,H) \leq \text{dist}(s,t,G) + 2$. To see this, let $x \in \Gamma(P(s,t)) \cap S_1$ and let $z$ be $x$’s neighbor in $P(s,t)$ (or $x$ itself in case $x$ is on $P(s,t)$). Recall that a BFS tree rooted at $x$ is added to $H$ in the second stage. Therefore, $\text{dist}(x,y,G)$ for every $y \in V$. We get that

$$\text{dist}(s,t,H) \leq \text{dist}(s,x,H) + \text{dist}(x,t,H)$$
$$= \text{dist}(s,x,G) + \text{dist}(x,t,G)$$
$$\leq \text{dist}(s,z,G) + 1 + \text{dist}(z,t,G) + 1$$
$$= \text{dist}(s,t,G) + 2.$$ 

We are left with the second case, where $\text{heavy_dist}(s,t,G) \leq \mu^3/n$. In this case, the claim holds deterministically. Notice that there exists center clusters $x_1, x_2 \in S_2$ such that $s \in C(x_1)$ and $t \in C(x_2)$, as otherwise we would have added all their adjacent edges to $H$, making them covered. Let $C_1 = C(x_1)$ and $C_2 = C(x_2)$. In the third stage of the algorithm, the shortest path $P = P(y_1, y_2)$, among all paths $P(y_1, y_2)$ such that $y_1 \in C_1$ and $y_2 \in C_2$ and $\text{heavy_dist}(y_1, y_2, G) \leq \mu^3/n$, is added to $H$. Note that $|P| \leq |P(s,t)|$. Note also that as $y_1 \in C_1$ and $y_2 \in C_2$, we have that $\text{dist}(s, y_1, H) \leq 2$ and $\text{dist}(y_2, t, H) \leq 2$. We get that

$$\text{dist}(s,t,H)$$
$$\leq \text{dist}(s,y_1,H) + \text{dist}(y_1,y_2,H) + \text{dist}(y_2,t,H)$$
$$\leq 2 + |P| + 2 \leq 4 + |P(s,t)|$$
$$= \text{dist}(s,t,G) + 4.$$ 

The lemma follows. 

We note that the technique for handling pairs of nodes $s$ and $t$ such that $|\Gamma(P(s,t))| > \mu^4/(3n)$ (by selecting independently at random a set of nodes that with high probability contains a node in $\Gamma(P(s,t))$) is already used by Woodruff in [31].

By applying the union bound on all pairs of nodes, we get the following corollary.

**Corollary 2.1.** With probability at least $1-1/n$, the constructed subgraph $H$ is a 4-additive spanner for $G$.

### 3. $\tilde{O}(n^{1+\delta})$ edge spanners with additive stretch $O(n^{1/2-3\delta/2})$

In this section we present a construction for a $(1,\tilde{O}(n^{1/2-3\delta/2}))$-spanner with $\tilde{O}(n^{1+\delta})$ edges for any $3/17 \leq \delta \leq 1/3$.

Throughout, let $\tilde{B}(v) = \Gamma(v, \mu)$ and $\mu = n^{1/2-3\delta/2}$. Let us partition the nodes into three sets. The set $S_1$ contains all nodes $v$ such that $|\tilde{B}(v)| \leq \mu^3$. The set $S_2$ contains all nodes $v$ such that $\mu^2 < |\tilde{B}(v)| \leq n^{3\delta}$ (note that in the relevant range of $3/17 \leq \delta \leq 1/3$, $\mu^2 < n^{3\delta}$, and long otherwise). Procedure **Short-distances** consists of three sub procedures: **Very-sparse**, **Sparse** and **Dense**.

Procedure **Short-distances** adds a set of edges $E_{\text{short}}$ to the constructed spanner $H$ such that the distance for every two close nodes in $H$ is within $O(\mu \log n)$ additive stretch from their distance in $G$. As mentioned above, Procedure **Short-distances** consists of three sub procedures: **Very-sparse**, **Sparse** and **Dense**. Procedure **Very-sparse** handles very sparse areas, namely; nodes $v \in S_1$. The high level idea is that in very sparse areas, the algorithm adds a small set of edges $E_{\text{vs}}$ such that for every node $v$ in $S_1$, prefixes to all its shortest paths are contained in $E_{\text{vs}}$. Therefore, in some sense (and as will become clearer later on) these nodes are already “taken care of”. Procedure **Sparse** handles sparse areas, namely, nodes $v \in S_2$. In this case by adding a set of edges $E_{\text{sparse}}$ the algorithm ensures an additive stretch of at most $3\log n$ for node pairs in sparse areas at distance up to $\mu$.

Loosely speaking, Procedure **Sparse** partitions all nodes of degree $n^\delta$ or higher into disjoint clusters. Each such cluster $C$ is centered at some node $v$, and all nodes that belong to the cluster $C$ are at distance 1 from $v$. For every cluster, the procedure adds edges between the center cluster to the other nodes in that cluster. The procedure then looks on balls $\tilde{B}(v)$ for every such center cluster $v$, and by a sophisticated BFS algorithm it ensures an additive stretch of at most $3\log n$ between $v$ and every node in $\tilde{B}(v)$ (the main difference between the outcome of this algorithm and the standard BFS algorithm is that this algorithm adds a smaller number of edges at the price of approximated distances that are within $O(\log n)$ additive stretch from the exact distances, by exploiting the fact that the algorithm already added some edges inside the clusters).
Finally, Procedure Dense handles dense areas, namely, nodes \( v \in S_3 \). More precisely, it picks a set \( C_{rep} \) and adds a set of edges \( E_{dense} \) to \( H \) with the following properties. Every pair of nodes in \( C_{rep} \) has “small” additive stretch, in addition, all nodes in \( S_3 \) have a node in \( C_{rep} \) close to them.

The rough idea of the analysis of Procedure Short-distances is as follows. To handle close pair of nodes \( s \) and \( t \), the general idea is as follows. We show that \( E_s \cup E_{sparse} \) contains a path \( P_1 \) between \( s \) and some node \( c_1 \in C_{rep} \) with the following properties. First, \( c_1 \) is “close” to some node on the path \( P(s,t) \). Second, the path \( P_1 \) is within \( O(\mu \log n) \) additive stretch from the distance between \( s \) and \( c_1 \). Similarly, we show that \( E_s \cup E_{sparse} \) contains a path \( P_2 \) between some node \( c_2 \in C_{rep} \) and \( t \) with the following properties. First, \( c_2 \) is “close” to some node on the path \( P(s,t) \). Second, the path \( P_2 \) is within \( O(\mu \log n) \) additive stretch from the distance between \( c_2 \) and \( t \) in \( G \). In addition, we show that the set of edges \( E_{dense} \) contains a path \( P_3 \) between \( c_1 \) and \( c_2 \) with a small additive stretch. Concatenating all these paths together, we get a path from \( s \) to \( t \) with a small additive stretch. This handles close pair of nodes.

To handle long pair of nodes Procedure Long-distances uses a similar technique. The procedure picks a set \( R_{long} \) and a set of edges \( E_{long} \) with the following properties. First, every pair of nodes in \( R_{long} \) is within additive stretch 2 from the distance in \( G \). Second, for every pair of nodes \( s, t \) that is far away (i.e. not close), we show that there exist nodes \( r_1, r_2 \in R_{long} \) such that \( r_1 \) and \( r_2 \) are “close” to some nodes on the path \( P(s,t) \) and in addition \( r_1 \) is closed to \( s \) and \( r_2 \) is closed to \( t \). As \( s \) and \( r_1 \) are closed, as explained above, procedure Short-distances guarantees that the constructed spanner \( H \) contains a path between \( r_1 \) and \( r_2 \) that is within additive stretch 2 from their distance in \( G \). Concatenating all these path together we get a path from \( s \) to \( t \) that is within additive stretch \( O(\mu \log n) \) from their distance in \( G \).

Let us introduce some definitions. For a node \( v \), the sparse threshold of \( v \), denoted by \( st(v) \), is the smallest integer \( r \) such that \( |\Gamma(v,r,G)| \leq r \cdot n^d \). For a subgraph \( P \) and set of edges \( E' \), let \( cost(P, E') = |E(P) \setminus E'| \).

For simplicity of presentation, assume the shortest path between any two nodes is unique and every subpath of a shortest path is also a shortest path. (This is without loss of generality since one can enforce it by a perturbation of the edge weights.)

Very sparse areas. Procedure Very-sparse handles very sparse areas (i.e., nodes \( v \in S_1 \), by constructing an edge subset \( E_v \) for these areas and adding it to the constructed spanner \( H \). In this case the algorithm tries to add prefixes of exact shortest paths for node pairs of distance \( 2\mu \) or higher. More precisely, the algorithm adds a set of edges \( E_v \), to the constructed spanner \( H \) such that for every node \( v \in S_1 \) and for every node \( z \) such that \( dist(v,z,G) > 2\mu \), a nonempty prefix of the path \( P(v,z) \) is contained in \( E_v \).

Roughly speaking, if a node \( v \) satisfies \( |\tilde{B}(v)| = |\Gamma(v,\mu)| \leq \mu \cdot n^d \), then there must be a radius \( r \leq \mu \) such that \( |\Gamma^*(v,r)| \leq n^5 \). We add a BFS tree from every node in \( \Gamma^*(v,r) \) spanning the nodes in \( \tilde{B}(v) \), and since \( \Gamma^*(v,r) \) contains a “small” number of nodes this process requires adding a “small” number of edges. Moreover, for every node \( z \in \tilde{B}(v) \) and for every node \( y \) at distance greater than \( 2\mu \) from it, the path \( P(z,y) \) must intersect with \( \Gamma^*(v,r) \). We thus can show that a prefix of this path is added to the constructed spanner. In other words, for every node in a very sparse area, we add prefixes to all its shortest paths (that are of length greater than \( 2\mu \)).

Formally, Procedure Very-sparse operates as follows. Initially all nodes are unmarked. While there is an unmarked node \( v \) with \( 1 < st(v) \leq \mu \), choose \( v \) to be the unmarked node with maximal \( st(v) \). For every node \( x \) in \( \Gamma^*(v, st(v)) \) construct a BFS tree \( T(x) \) rooted at \( x \) in the induced graph \( \Gamma(v, st(v) - 1) \cup \{x\} \), add the edges of \( T(x) \) to \( E_v \) and mark all nodes in \( \Gamma(v, st(v) - 1, G) \). See Procedure Very-sparse for pseudocode.

Lemma 3.1. Procedure Very-sparse satisfies the following two properties.

(a) For every node \( z_1 \) such that \( |\Gamma(z_1, \mu)| \leq \mu \cdot n^d \), and for every node \( z_2 \) such that \( dist(z_1, z_2) > 2\mu \), there exists a node \( v \neq z_1 \) on \( P(z_1, z_2) \) such that \( E_v \) contains the path \( P(z_1, z) \).

(b) \( |E_v| = O(n^{1+\epsilon}) \).

Proof: Let \( S_\alpha \) be the set of nodes \( v \) that were chosen in the while loop of Procedure Very-sparse and for each \( v \in S_\alpha \), let \( i(v) \) be the iteration of Procedure Very-sparse in which \( v \) was chosen. To prove (a), consider two nodes \( z_1 \) and \( z_2 \) as in the lemma. Since \( |\Gamma(z_1, \mu)| \leq \mu \cdot n^d \), \( z_1 \) must be marked at the end of Procedure Very-sparse. Let \( v \in S_\alpha \), be the node such that \( z_1 \in \Gamma(v, st(v) - 1, G) \) and \( z_1 \) is marked in iteration \( i(v) \). Note that \( z_2 \notin \Gamma(v, st(v) - 1, G) \) as \( dist(z_1, z_2) > 2\mu \). Clearly, \( \Gamma(v, st(v)) \cap P(z_1, z_2) \neq \emptyset \). Let \( z \) be the first node of \( \Gamma^*(v, st(v)) \) on the path \( P(z_1, z_2) \). Note that \( P(z_1, z) \) is contained in the induced graph.
Procedure **Very-sparse**($G(V,E)$)

$E_{vs} \leftarrow \emptyset$

Unmark all nodes $v \in V$

While there is an unmarked node $v$ with $1 < \text{st}(v) \leq \mu$ do:

Choose $v$ to be the unmarked node with maximal $\text{st}(v)$

For every node $x$ in $\Gamma^*(v,\text{st}(v))$ do:

Construct a BFS tree $T(x)$ rooted at $x$ in the induced graph

$\Gamma(v,\text{st}(v) - 1) \cup \{x\}$

$E_{vs} \leftarrow E_{vs} \cup E(T(x))$

Mark all nodes of $T(x)$, i.e., $\Gamma(v,\text{st}(v) - 1, G)$

Return $E_{vs}$

of $(\Gamma(v, \text{st}(v) - 1, G) \cup \{z\})$ and since a BFS tree $T(z)$ rooted at $z$ in the induced graph $\Gamma(v, \text{st}(v) - 1, G) \cup \{z\}$ is added to $E_{vs}$, we get that $P(z_1, z) \subseteq E_{vs}$, as required.

To prove (b), we consider two types of nodes, the first type is nodes $v$ such that $\text{st}(v) \leq 2$, and the second type is nodes $v$ such that $\text{st}(v) > 2$. We show that the number of edges added to $E_{vs}$ for each type separately is $O(n^{1+\delta})$.

Consider the first type and let $v \in S_z$ such that $\text{st}(v) \leq 2$. Note that in this case $O(n^\delta)$ edges are added to $E_{vs}$ for $T(v)$. Hence, for all nodes $v'$ such that $\text{st}(v') \leq 2$, $O(n^{1+\delta})$ are added to $E_{vs}$.

Consider now the second type and let $v \in S_z$ such that $\text{st}(v) > 2$. Let $E_{vs}^v$ be the set of edges added to $E_{vs}$ in iteration $i(v)$ of Procedure **Very-sparse**. We claim that $|\Gamma^*(v, \text{st}(v))| \leq n^\delta$. To see this, note that by the definition of $\text{st}(v)$, $|\Gamma(v, \text{st}(v) - 1, G)| \geq (\text{st}(v) - 1)n^\delta$ and that $|\Gamma(v, \text{st}(v), G)| \leq \text{st}(v) \cdot n^\delta$. It follows that $|E_{vs}^v| \leq n^\delta \cdot |\Gamma(v, \text{st}(v) - 1, G)|$. By definition of $\text{st}(v)$ we also have $|\Gamma(v, \lfloor \text{st}(v) - 1/2 \rfloor, G)| \geq \lfloor \text{st}(v) - 1/2 \rfloor \cdot n^\delta$. It is not hard to verify that for every $\text{st}(v) > 2$, $|\Gamma(v, \lfloor \text{st}(v) - 1/2 \rfloor, G)| = O(|\Gamma(v, \text{st}(v) - 1, G)|)$, and hence

$$|E_{vs}| \leq \sum_{v \in S_z} |E_{vs}^v| \leq n^\delta \sum_{v \in S_z} |\Gamma(v, \text{st}(v) - 1, G)| \leq n^\delta \sum_{v \in S_z} O(|\Gamma(v, \lfloor \text{st}(v) - 1/2 \rfloor, G)|).$$

Next, we show that the sets $\Gamma(v, \lfloor \text{st}(v) - 1/2 \rfloor, G)$ are disjoint for $v \in S_z$. Assume towards contradiction that $z \in \Gamma(v_1, \lfloor \text{st}(v_1) - 1/2 \rfloor, G)$ and $z \in \Gamma(v_2, \lfloor \text{st}(v_2) - 1/2 \rfloor, G)$ for some $v_1, v_2 \in S_z$. Assume w.l.o.g. that $v_1$ is chosen first in Procedure **Very-sparse**. Then $\text{dist}(z, v_1) \leq \lfloor \text{st}(v_1) - 1/2 \rfloor$ and $\text{dist}(z, v_2) \leq \lfloor \text{st}(v_2) - 1/2 \rfloor$, so $\text{dist}(v_1, v_2) \leq \lfloor \text{st}(v_1) - 1/2 \rfloor + \lfloor \text{st}(v_2) - 1/2 \rfloor \leq \text{st}(v_1) - 1$. It follows that $v_2$ was marked at the end of iteration $i(v_1)$ of Procedure **Very-sparse**, contradiction. It follows that $\sum_{v \in S_z} |\Gamma(v, \lfloor \text{st}(v) - 1/2 \rfloor, G)| \leq n$, hence $|E_{vs}| = O(n^{1+\delta})$.

Sparse areas. We now turn to describing Procedure **Sparse**, for handling sparse areas, namely, nodes $v$ such that $\mu \cdot n^\delta \leq |B(v)| \leq n^{3\delta}$.

In this case the algorithm attempts to ensure an additive stretch of $3 \log n$ for node pairs at distance up to $\mu$. Specifically, the algorithm adds a set of edges $E_{\text{sparse}}$ such that for every node $v \in S_2$, the distance from $v$ to all nodes in $B(v)$ is within additive stretch $3 \log n$ from the distance in $G$.

Procedure **Sparse** starts with sampling a set of center nodes $C_{\text{sparse}}$, of expected size $n^{1-\delta}$, by selecting every node at random with probability $1/n^\delta$. For every node $v$ none of whose neighbors was chosen to $C_{\text{sparse}}$, add all its incident edges to $E_{\text{sparse}}^0$ (initially set to be empty). Otherwise, pick a neighbor $\text{center}(v, C_{\text{sparse}}) \in C_{\text{sparse}}$ of $v$ and add the edge $(v, \text{center}(v, C_{\text{sparse}}))$ to the constructed spanner. This essentially attempts to partition all nodes of degree $n^\delta$ or higher into disjoint clusters.

Let us introduce some notation. For a path $P$, let $\text{centers}(P, C_{\text{sparse}}) = \{c \in C_{\text{sparse}} \mid \exists z \in P, c = \text{center}(z, C_{\text{sparse}})\}$. Consider a path $P$, a node $v \in V$, and a subgraph $H$. Let $\text{BFS-Val}(P, v, C_{\text{sparse}}, H)$ be the number of center nodes $c \in \text{centers}(P, C_{\text{sparse}})$ such that adding $P$ to $H$ will improve their distance to $v$, namely, such that $\text{dist}(c, v, H \cup P) < \text{dist}(c, v, H)$.

For a subgraph $H$, a node $v$, a center node $c \in C_{\text{sparse}}$, and a path $P = P(c, z)$ from $c$ to some node $z$ in $P(c, v)$, let $\text{First-not-Help}(P, v, c, C_{\text{sparse}}, H)$ be the node $v'$ on $P$ closest to $c$ such that adding $P$ to $H$ does not help the center $c'$ of $v'$ (in terms of its distance from $v$), or formally, such that $\text{dist}(c', v, H \cup P) = \text{dist}(c', v, H)$ where $c' = \text{center}(v', C_{\text{sparse}})$.

Procedure **Sparse** employs a procedure **Approximate-BFS** that given a node $v$ where
[\tilde{B}(v)] \leq n^{38}$, returns a set of edges $E_{\text{BFS}}$ of size $O(n^{25})$ such that the distance from $v$ to all nodes in $\tilde{B}(v)$ in $E_{\text{BFS}} \cup E_{\text{sparse}}^0$ is within additive stretch $3 \log n$ from the distance in $G$.

Procedure Approximate-BFS is invoked on every center $v \in C_{\text{sparse}}$ and we show that by adding $O(n^{25} \log n)$ additional edges, the distance between $v$ and every node in $\tilde{B}(v)$ is within $O(\log n)$ additive stretch from the distance in $G$. In particular, the procedure examines every center $c \in C_{\text{sparse}}$ and adds some prefix of the path $P(c, v)$. It first tries to add the entire path $P(c, v)$, but would take the entire path only if sufficient many other centers benefit from it. Otherwise, the procedure will try to add a subpath of $P(c, v)$ of at most half its length, again, only provided there are many centers who may benefit. This testing process continues until the procedure finds a prefix whose "benefit" is sufficiently large with respect to its length.

Formally, Procedure Approximate-BFS operates as follows. Let $(v, C(v), E_{\text{sparse}}^0, G)$ be its input, where $v \in C_{\text{sparse}}$ and $C(v) = \tilde{B}(v) \cap C_{\text{sparse}}$. Initially, set $E_{\text{BFS}} = \emptyset$. For every node $c \in C(v)$, $v'$ is set to be $v$ and the path $P(c, v')$ is examined and we add this path to the constructed spanner if $6 \cdot \text{BFS-Val}(P(c, v'), v, C(v), E_{\text{sparse}}^0 \cup E_{\text{BFS}}) \geq \text{cost}(P(c, v'), E_{\text{sparse}}^0 \cup E_{\text{BFS}})$. If not, we set $v' = \text{First-not-Help}(P(c, v'), v, c, C(v), E_{\text{sparse}}^0 \cup E_{\text{BFS}})$. This process continues until $6 \cdot \text{BFS-Val}(P(c, v'), v, C(v), E_{\text{sparse}}^0 \cup E_{\text{BFS}}) \geq \text{cost}(P(c, v'), E_{\text{sparse}}^0 \cup E_{\text{BFS}})$. Notice that the process ends as the inequality holds for $v' = c$.

**Lemma 3.2.** For every node $v \in C_{\text{sparse}}$, the set $E_{\text{BFS}}$ returned by Procedure Approximate-BFS satisfies that $dist(c, v, E_{\text{sparse}}^0 \cup E_{\text{BFS}}) \leq \text{dist}(c, v, G) + 2 \log n$ for every node $c \in C_{\text{sparse}} \cap \tilde{B}(v)$.

**Proof.** Consider a node $c \in C(v)$. Let $j$ be the number of iterations in the while loop of Procedure Approximate-BFS for the node $c \in C(v)$. Let $v(i)$ be the node $v'$ in the end of iteration $i$ for $i < j$ and let $c(i) = \text{center}(v(i), C_{\text{sparse}})$. See Figure 1 for illustration.

We claim that $\text{dist}(v(i), v, E_{\text{sparse}}^0 \cup E_{\text{BFS}}) \leq \text{dist}(v(i), v, G) + 2i$. The proof of this claim is by induction on $i$. For $i = 0$, namely $v' = v$, the claim is trivial. Assume correctness for $i < k$ and consider $i = k$. Recall that the node $v(k)$ is a node on the path $P = P(c, v(k - 1))$ such that $\text{dist}(c(k), v, E_{\text{sparse}}^0 \cup E_{\text{BFS}}) = \text{dist}(c(k), v, E_{\text{sparse}}^0 \cup E_{\text{BFS}})$. We get that

$$\text{dist}(v(k), v, E_{\text{sparse}}^0 \cup E_{\text{BFS}}) \leq 1 + \text{dist}(c(k), v, E_{\text{sparse}}^0 \cup E_{\text{BFS}})$$

$$= 1 + \text{dist}(c(k), v, E_{\text{sparse}}^0 \cup E_{\text{BFS}} \cup P)$$

$$\leq 2 + \text{dist}(v(k), v, E_{\text{sparse}}^0 \cup E_{\text{BFS}} \cup P)$$

$$\leq 2 + \text{dist}(v(k), v, E_{\text{sparse}}^0 \cup E_{\text{BFS}} \cup P)$$

$$\leq 2 + \text{dist}(v(k), v, E_{\text{sparse}}^0 \cup E_{\text{BFS}})$$

$$\leq 2 + \text{dist}(v(k), v, E_{\text{sparse}}^0 \cup E_{\text{BFS}})$$

$$\leq 2 + \text{dist}(v(k), v, E_{\text{sparse}}^0 \cup E_{\text{BFS}})$$

$$= 2k + \text{dist}(v(k), v, G),$$

where the last inequality follows from the inductive hypothesis. We next show that the number of centers adjacent to the considered path is at least halved in each iteration of the procedure, and hence $j \leq \log n$, which yields the lemma. Formally, we show that $|\text{centers}(P(c, v(i)), C_{\text{sparse}})| \leq |\text{centers}(P(c, v(i-1)), C_{\text{sparse}})|/2$. By definition of $v(i)$, for every node $x \in \text{centers}(P(c, v(i)), C_{\text{sparse}})$, $\text{dist}(x, v, E_{\text{sparse}}^0 \cup E_{\text{BFS}}) > \text{dist}(x, v, E_{\text{sparse}}^0 \cup E_{\text{BFS}} \cup P(c, v(i-1)))$. We get that, $\text{BFS-Val}(P(c, v(i-1)), v, C(v), E_{\text{sparse}}^0 \cup E_{\text{BFS}}) \geq |\text{centers}(P(c, v(i-1)), C_{\text{sparse}})|$. Recall that if the path was not chosen, then by the condition in the procedure, $6 \cdot \text{BFS-Val}(P(c, v(i-1)), v, C(v), E_{\text{sparse}}^0 \cup E_{\text{BFS}}) < \text{cost}(P(c, v(i-1)), E_{\text{sparse}}^0 \cup E_{\text{BFS}}) \leq |\text{centers}(P(c, v(i-1)), C_{\text{sparse}})|$, where the last inequality follows from the fact that every $c \in \text{centers}(P(c, v(i-1)), C_{\text{sparse}})$ can have at most three neighbors in $P(c, v(i-1))$ since $P(c, v(i-1))$ is a shortest path. We get that $|\text{centers}(P(c, v(i)), C_{\text{sparse}})| \leq |\text{centers}(P(c, v(i-1)), C_{\text{sparse}})|/2$, as required.

**Lemma 3.3.** For every $v \in C_{\text{sparse}}$, the set $E_{\text{BFS}}$ returned by Procedure Approximate-BFS satisfies $|E_{\text{BFS}}| = O(|C(v)| \log n) = O(n^{25} \log n)$.

**Proof.** Let $E_{\text{BFS}}^0$ be the set $E_{\text{BFS}}$ at the beginning of $c$’s iteration of Procedure Approximate-BFS. Let $P(c) = P(c, v')$ be the path that was added to $E_{\text{BFS}}$ in $c$’s iteration of Procedure Approximate-BFS, where $P(c, v')$ can also be empty if $v' = c$. We argue that the cost of adding $P(c)$ is roughly proportional to its benefit. Consider the set $X(c) = \{y \in \text{centers}(P(c, E_{\text{sparse}}^0) \cup E_{\text{BFS}}) \mid \text{dist}(y, v, E_{\text{sparse}}^0 \cup E_{\text{BFS}} \cup P(c) < \text{dist}(y, v, E_{\text{sparse}}^0 \cup E_{\text{BFS}})\}$. Note that $\text{BFS-Val}(P(c), v, C(v), E_{\text{sparse}}^0 \cup E_{\text{BFS}}) = |X(c)|$. We claim that each node $z \in C(v)$
Procedure Approximate-BFS$(v, \tilde{C}(v), E_\text{sparse}^0)$

$E_\text{BFS} \leftarrow \emptyset$

For every node $c \in \tilde{C}(v)$ do:

Set $v' \leftarrow v$ and $\text{ind} = \text{false}$.

While ($\text{ind} = \text{false}$) do:

If $6 \cdot \text{BFS-Val}(P(c, v'), v, \tilde{C}(v), E_\text{sparse}^0, E_\text{BFS}) \geq \text{cost}(P(c, v'), E_\text{sparse}^0, E_\text{BFS})$ then:

Set $E_\text{BFS} \leftarrow E_\text{BFS} \cup P(c, v')$

Set $\text{ind} = \text{true}$

Else set $v' \leftarrow \text{First-not-Help}(P(c, v'), v, c, \tilde{C}(v), E_\text{sparse}^0, E_\text{BFS})$

Return $E_\text{BFS}$

Figure 1: Illustration for Lemma 3.2.

may belong to at most $O(\log n)$ sets $X(c)$. To see this let $c \in \tilde{C}(v)$ be the first node that was considered in Procedure Approximate-BFS and that $z \in X(c)$. By the analysis of Lemma 3.2, after adding the path $P(c)$ to $E_\text{BFS}$, $\text{dist}(c, v, E_\text{sparse}^0, E_\text{BFS}) \leq \text{dist}(c, v, G) + 2\log n$. Thus the distance between $c$ and $v$ can improve at most $2\log n$ times. Since $|\tilde{B}(v)| = n^3$ and $C_\text{sparse}$ contains each node with probability $1/n^3$, we get that in expectation $|\tilde{C}(v)| \leq n^{3}$.

By Lemma 3.2 we have the following.

Corollary 3.1. Let $H'$ be a $\log n/3$-multiplicative spanner. The set $E_\text{sparse}$ returned by Procedure Sparse satisfies the following. For every $v \in V$ such that $|\tilde{B}(v)| \leq n^3$ and every $x \in \tilde{B}(v)$, $\text{dist}(v, x, E_\text{sparse} \cup H') \leq \text{dist}(v, x, G) + 3\log n$.

Proof: Consider a node $v \in V$ such that $|\tilde{B}(v)| \leq n^3$ and a node $x \in \tilde{B}(v)$. Let $c_v = \text{center}(v, C_\text{sparse})$ and $c_x = \text{center}(x, C_\text{sparse})$ (as explained above it is enough to consider the case where both $v$ and $x$ are uncovered and thus $c_v$ and $c_x$ are well defined). If $\text{dist}(v, x) \leq \mu - 2$ then it is not hard to verify that $c_x \in \tilde{B}(c_v)$. We thus have by Lemma 3.2 that $\text{dist}(c_v, c_x, E_\text{sparse}^0, E_\text{BFS}) \leq \text{dist}(c_v, c_x, E_\text{sparse}^0) + \text{dist}(c_v, c_x, G) + 2\log n$. Hence, $\text{dist}(v, x, E_\text{sparse}^0) \leq \text{dist}(v, x, E_\text{sparse}^0) + \text{dist}(c_v, c_x, E_\text{sparse}^0) + \text{dist}(c_v, c_x, E_\text{sparse}^0) \leq 2 + \text{dist}(c_v, c_x, G) + 2\log n \leq 4 + \text{dist}(v, x, G) + 2\log n \leq \text{dist}(x, v, G) + 3\log n$. Let us consider now the end case where $\text{dist}(v, x) \geq \mu - 1$. In this case it might be that $c_x \notin \tilde{B}(c_v)$. Let $x'$ be the node on the path $P(x, v)$ at distance 2 from $x$. It is not hard to verify that $\text{center}(x', C_\text{sparse}) \in \tilde{B}(c_v)$ and thus, using similar analysis as above, $\text{dist}(x', v, E_\text{sparse}) \leq \text{dist}(x', v, G) + 2\log n + 4$. Since $H'$ contains $\log n/3$-multiplicative spanner, we get $\text{dist}(x, v, E_\text{sparse} \cup H') \leq \text{dist}(x, v, G) + 2\log n + 4 \leq \text{dist}(x, v, G) + 3\log n$.

Lemma 3.4. The expected number of edges in $E_\text{sparse}$ is $O(n^{3} n + \log n)$.

Proof: Consider a node $c \in C_\text{sparse} \cap (S_1 \cup S_2)$, i.e., such that $|\tilde{B}(c)| \leq n^3$. By Lemma 3.3, the number of edges added to $E_\text{sparse}$ in Procedure Sparse in $c$’s iteration is $O(n^{3} n + \log n)$. The expected number of nodes in $C_\text{sparse}$ is $O(n^{3} n + \log n)$. Thus the expected number of edges added to $E_\text{sparse}$ is $O(n^{3} n + \log n)$.

Dense areas. An $r$-separated $r$-dominating set (or an $r - SD$ for short) for a set of nodes $C'$ is a subset $C''$ of $C'$ such that all nodes in $C''$ are at distance at least $r$ from one another and every node in $C'$ has a node at distance at most $r$ from it in $C''$. Note that such a set always exists and can be constructed greedily, as one can simply consider the nodes in $C'$ one by one and add each node $c$ to the set $C''$ if none of the nodes already in $C''$ is at distance $r$ from $c$. It is not hard to verify that $C''$ is an $r - SD$ set.

Procedure Dense handles dense areas. Procedure Dense picks a set of edges $E_\text{dense}$ such that the set of edges $E' = E_\text{dense} \cup E_\text{sparse}$ satisfies that for every two close nodes $x_1$ and $x_2$, their distance in $E'$ is within additive stretch $O(\log n \cdot \mu)$ from the distance in $G$. 
Procedure \texttt{Sparse}(G)

\[ E_{\text{sparse}}, E_{\text{sparse}}^0 \leftarrow \emptyset \]

Choose a set of nodes \( C_{\text{sparse}} \) by independently sampling at random every node with probability \( 1/n^3 \).

For every node \( v \):

- If \( \Gamma(v) \cap C_{\text{sparse}} = \emptyset \) add all incident edges of \( v \) to \( E_{\text{sparse}} \).
- Else do:
  - Select \( \text{center}(v, C_{\text{sparse}}) \) to be some neighbor of \( v \) in \( C_{\text{sparse}} \).
  - Add the edge \( (v, \text{center}(v, C_{\text{sparse}})) \) to \( E_{\text{sparse}} \).

For every node \( v \in C_{\text{sparse}} \) such that \( |\tilde{B}(v)| \leq n^{3\delta} \) do:

\[ E_{\text{new}} \leftarrow \text{Approximate-BFS}(v, \tilde{B}(v) \cap C_{\text{sparse}}, E_{\text{sparse}}, G) \]

\[ E_{\text{sparse}} \leftarrow E_{\text{sparse}} \cup E_{\text{new}} \]

Return \( E_{\text{sparse}} \).

More precisely, it picks a maximal \((3\mu) - SD \) set \( C_{\text{rep}} \) for \( S_3 \) and a set of edges \( E_{\text{dense}} \) with the following properties. Every pair of nodes in \( C_{\text{rep}} \) has "small" additive stretch, in addition, all nodes in \( S_3 \) have a node in \( C_{\text{rep}} \) close to them.

For a center \( c \in C_{\text{rep}} \), let \( Cluster(c) \) be the set of all nodes \( v \in V \) such that \( \text{dist}(c, v) \leq 3\mu \) and \( c \) is closer to \( v \) than all nodes in \( C_{\text{rep}} \) (recall that we assume uniqueness of the shortest path). Note that all nodes \( x \) on the shortest path from \( v \in Cluster(c) \) to \( c \) satisfy \( x \in Cluster(c) \). Note that every node \( x \in S_3 \) satisfies \( x \in Cluster(c) \) for some \( c \in C_{\text{rep}} \).

For a path \( P \), let \( E_{\text{important}}(P) \) be the set of edges of \( P \) at distance at most \( 2\mu \) from some node \( v \in V(P) \cap S_3 \).

For a path \( P \), let \( C_P = \{ e \in C_{\text{rep}} \mid \exists v \in V(P), v \in Cluster(c) \} \).

Formally, Procedure \texttt{Dense} operates as follows. First pick a maximal \((3\mu) - SD \) set \( C_{\text{rep}} \) for \( S_3 \). For every node \( c \in C_{\text{rep}} \) construct a BFS tree on \( Cluster(c) \) and add the edges of the tree to the constructed spanner.

Next, the procedure goes over pairs of centers \((c_1, c_2) \in C_{\text{rep}} \), and considers adding their path \( P = P(c_1, c_2) \) to the spanner. Adding this path will benefit certain sufficiently close pairs of centers from \( C_P \), by reducing their distance. The procedure will refrain from adding the path \( P(c_1, c_2) \) to the output spanner if there exists a center \( c \in C_P \) such that both \((c_1, c)\) and \((c, c_2)\) have already benefited from paths that were added to the spanner earlier on. Formally, unmark all pairs \((c_1, c_2)\) such that \( c_1, c_2 \in C_{\text{rep}} \). For every two centers \( c_1, c_2 \in C_{\text{rep}} \) do the following. If there is no node \( c \in C_P \) such that both \((c_1, c)\) and \((c, c_2)\) are marked, then add \( E_{\text{important}}(P(c_1, c_2)) \) to the constructed spanner, and mark all pairs \((c_1, c)\) and \((c, c_2)\) such that \( c \in C_P \).

Consider two nodes \( c_1, c_2 \in C_{\text{rep}} \). We say that the path \( P(c_1, c_2) \) is purchased by the algorithm if the set of edges \( E_{\text{important}}(P(c_1, c_2)) \) were added to the spanner.

**Lemma 3.5.** \( |E_{\text{dense}}| \leq O(n^{1+\delta}) \) for every \( 3/17 \leq \delta \).

**Proof:** The sets \( Cluster(c) \) for \( c \in C_{\text{rep}} \) are disjoint. Moreover, since the nodes in \( C_{\text{rep}} \) are at distance at least \( 3\mu \) from one another, we get that \( \tilde{B}(c) \subseteq Cluster(c) \) for every \( c \in C_{\text{rep}} \). Recall that \( |\tilde{B}(c)| \geq n^{3\delta} \) for every \( c \in C_{\text{rep}} \). We thus get that \( |C_{\text{rep}}| \leq n^{1-3\delta} \). Therefore, there are at most \( n^{2-6\delta} \) pairs of nodes in \( C_{\text{rep}} \). Consider a path \( P = P(c_1, c_2) \) that was purchased by Procedure \texttt{Dense}. Consider a center \( c \in C_P \), let \( v_1 \) be the first node (closest to \( c_1 \) in \( P \) such that \( v_1 \in Cluster(c) \cap S_3 \) and let \( v_2 \) be the last node (closest to \( c_2 \)) in \( P \) such that \( v_2 \in Cluster(c) \cap S_3 \). We claim that \( |P(v_1, v_2)| \leq 6\mu \). To see this, note that \( \text{dist}(v_1, c) \leq 3\mu \) and \( \text{dist}(v_2, c) \leq 3\mu \). Therefore, the number of edges in \( P \) that are added to \( E_{\text{important}}(P(c_1, c_2)) \) for all nodes \( v \in S_3 \cap Cluster(c) \) is \( O(\mu) \) (the edges that are at distance \( 2\mu \) from \( v_1 \) or \( v_2 \) plus the path \( P(v_1, v_2) \)). We get that the number of edges in \( E_{\text{important}}(P) \) is at most \( O(\mu |C_P|) \). Notice that the number of pairs in \( C_{\text{rep}} \) that are marked in Procedure \texttt{Dense} after purchasing the path \( P \) is at least \( |C_P| \) and that every pair is marked once. Let \( P_{\text{dense}} \) be the set of paths that were purchased by Procedure \texttt{Dense}. We have

\[
|E_{\text{dense}}| \leq \sum_{P \in P_{\text{dense}}} |E_{\text{important}}(P)| \\
\leq \sum_{P \in P_{\text{dense}}} O(\mu |C_P|) \leq O(\mu n^{2-6\delta}) \\
\leq O(n^{1+\delta}),
\]

where the last inequality holds for every \( 3/17 \leq \delta \). In addition, a BFS tree on \( Cluster(c) \) is constructed and added to \( E_{\text{dense}} \) for every \( c \in C_{\text{rep}} \). Note that each
Procedure Dense($G$)

$E_{dense} \leftarrow \emptyset$
Let $S_3$ be the set of dense nodes, namely, $v \in V$ such that $|\tilde{B}(v)| > n^{35}$
Let $C_{rep}$ be a maximal $(3\mu) - SD$ set for $S_3$
For every node $v$, let $ctr(v)$ be the center $c \in C_{rep}$ closest to $v$
For every $c \in C_{rep}$, let $Cluster(c) = \{ v \mid ctr(v) = c \}$
For every node $c \in C_{rep}$ construct a BFS tree on $Cluster(c)$ and add the edges of the tree to $E_{dense}$
Unmark all pairs $(c_1, c_2)$ such that $c_1, c_2 \in C_{rep}$
For every two centers $c_1, c_2 \in C_{rep}$ do:
  Let $C_P = \{ v \in C_{rep} \mid \exists w \in P(c_1, c_2), v \in Cluster(c) \}$
  If $\not\exists$ a center $c \in C_P$ such that both $(c_1, c)$ and $(c, c_2)$ are marked then:
    $E_{important}(P(c_1, c_2)) \leftarrow \{ e \in P(c_1, c_2) \mid \text{dist}(e, S_3 \cap P(c_1, c_2)) \leq 2\mu \}$
    $E_{dense} \leftarrow E_{dense} \cup E_{important}(P(c_1, c_2))$
  Mark all pairs $(c_1, c)$ and $(c, c_2)$ such that $c \in C_P$
Return $E_{dense}$

We consider two cases, the first case is where $x_1 \in S_1 \cup S_3$ and the second case is where $x_1 \in S_2$.
First note that every node $x \in P(x_1, x_2) \cap (S_1 \cup S_3)$ such that $dist(x, x_2) \geq 2\mu$ satisfies the following. There exists a node $y \in P(x, x_2)$ such that $y \neq x$ and $P(x, y) \subseteq E_{short}$. In case $x \in S_1$, the claim follows by Lemma 3.1. In case $x \in S_3$, the claim follows by the fact that $P(x_1, x_2)$ is $S_3$-tolerant.

In particular, the above observation holds for $x = x_1$, so let $y_1$ be the node satisfying $y_1 \in P(x_1, x_2)$ and $P(x_1, y_1) \subseteq E_{short}$. Hence $dist(x_1, y_1, E_{short}) = dist(x_1, y_1, G)$. By the induction hypothesis we have, $dist(y_1, x_2, E_{short}) \leq dist(y_1, x_2, G) + \Delta(y_1, x_2)$. We thus get that, $dist(x_1, x_2, E_{short}) \leq dist(x_1, y_1, E_{short}) + dist(y_1, x_2, E_{short}) \leq dist(x_1, y_1, G) + dist(y_1, x_2, G) + \Delta(y_1, x_2) \leq dist(x_1, x_2, G) + \Delta(x_1, x_2)$

We are left with the case where $x_1 \in S_2$. Let $z_1$ be the node at distance $\mu$ from $x_1$ on $P(x_1, x_2)$. Again, we handle separately the cases $z_1 \in S_1 \cup S_3$ and $z_1 \in S_2$. If $z_1 \in S_2$ then let $z_2$ be the node at distance $\mu + 1$ from $z_1$ on $P(z_1, x_2)$. Note that the additive distortion from $x_1$ to $z_2$ is at most $7 \log n$. To see this, let $y$ be the node at distance $\mu$ from $z_1$ on $P(z_1, x_2)$. Note that $y$ and $z_2$ are neighbors. Since $x_1, z_1 \in S_2$, by Lemma 3.1 we have $dist(x_1, z_1, E_{short}) \leq dist(x_1, z_1, G) + 3\log n$ and $dist(z_1, y, E_{short}) \leq dist(z_1, y, G) + 3\log n$. In addition, since $E_{short}$ contains a $(\log n/3)$-multiplicative spanner and since $dist(y, z_2, G) = 1$, we have $dist(y, z_2, E_{short}) \leq log n/3$. We thus conclude $dist(x_1, x_2, E_{short}) \leq dist(x_1, x_2, G) + 7 \log n$.

Otherwise, if $z_1 \in (S_1 \cup S_3)$ then note that the adjacent edge to $z_1$ in $P(z_1, x_2)$ belongs to $E_{short}$.

node belongs to only one cluster and thus at most $O(n)$ edges are added by this step. We thus conclude that the number of edges in $E_{dense}$ is $O(n^{1+\epsilon})$. 

Short Distances. Procedure Short-distances handles short distances, namely, pair of nodes $s$ and $t$ such that $|\Gamma(P(s, t, \mu))| \leq n^{1-25}$. Procedure Short-distances starts by constructing a $(\log n/3)$-multiplicative spanner and adding its edges to the constructed spanner. It then invokes Procedures Very-sparse($G$), Sparse($G$) and Dense($G$) and adds the set of edges returned by these procedures to the constructed spanner. We show that the set of edges $E_{short}$ returned by Procedure Short-distances satisfies that the distance for every two close nodes is within additive stretch $O(\mu \log n)$ from the distance in $G$.

We say that a path $P$ is $S_3$-tolerant if the edges of $P$ that are incident to nodes in $S_3$ belong to $E_{short}$. Towards proving the desired additive stretch on short distances, we first prove the following auxiliary lemma. The lemma bounds the additive stretch incurred by certain pairs of nodes $x, y$ in $E_{short}$ by the term $\Delta(x, y) = \lceil |\Gamma(P(x, y, \mu))|/(\mu - n^3) \rceil \cdot 7 \log n + \mu \cdot \log n$.

Lemma 3.6. For every two close nodes $x_1$ and $x_2$ such that $P(x_1, x_2) \in S_3$-tolerant,
\[ \text{dist}(x_1, x_2, E_{short}) \leq \text{dist}(x_1, x_2, G) + \Delta(x_1, x_2). \]

Proof: The proof is by induction on $\text{dist}(x_1, x_2, G)$. If $\text{dist}(x_1, x_2, G) < 3\mu$ then the lemma follows by the fact that $E_{short}$ contains a $(\log n/3)$-multiplicative spanner. Assume the lemma holds for every two nodes $x_1$ and $x_2'$ such that $\text{dist}(x_1, x_2', G) < d$ and consider two nodes $x_1$ and $x_2$ such that $\text{dist}(x_1, x_2, G) = d$.
Let $y$ be the first node on $P(z_1, x_2)$ such that the adjacent edge to $y$ in $P(y, x_2)$ is not in $E_{short}$ and that $\text{dist}(z_1, y) \leq \mu$.

If no such node exists then set $z_2$ to be the node at distance $\mu + 1$ from $z_1$ on $P(z_1, x_2)$. Note that in this case $P(z_1, z_2) \subseteq E_{short}$. We thus get $\text{dist}(x_1, z_2, E_{short}) \leq \text{dist}(x_1, z_1, E_{short}) + \text{dist}(z_1, z_2) + 3 \log n + \text{dist}(z_1, x_2, G) + 3 \log n$. Otherwise, if there exists such node $y$, set $z_2$ to be the node at distance $\mu$ from $y$ on $P(y, x_2)$. Note that $y \in S_2$. We get that

$$\text{dist}(x_1, z_2, E_{short}) \leq \text{dist}(x_1, z_1, E_{short}) + \text{dist}(z_1, y, E_{short}) + \text{dist}(y, z_2, E_{short}) \leq \text{dist}(x_1, z_1, G) + 3 \log n + \text{dist}(z_1, y, G) + \text{dist}(y, z_2, G) + 3 \log n \leq \text{dist}(x_1, z_2, G) + 6 \cdot \log n.$$

Moreover, note that in all cases $\text{dist}(x_1, z_2, G) \geq 2\mu + 1$. Using shortest path properties, it is not hard to verify that $\Gamma(x_1, G) \cap \Gamma(P(z_2, x_2), G) = \emptyset$. Recalling that $x_1 \in S_2$, we thus have $|\Gamma(P(z_1, x_2), G) - |\Gamma(x_1, G)| - |\Gamma(P(x_1, x_2), G)| - \mu \cdot n^3|$. Hence by the induction hypothesis,

$$\text{dist}(x_1, x_2, E_{short}) \leq \text{dist}(x_1, z_2, E_{short}) + \text{dist}(z_2, x_2, E_{short}) \leq \text{dist}(x_1, z_2, G) + 7 \log n + \text{dist}(x_1, x_2, G) + \Delta(x_1, x_2).$$

Consider two close nodes $x_1$ and $x_2$ on some path $P(c_1, c_2)$ that was purchased by Procedure Dense. It is not hard to verify that since $P(c_1, c_2)$ was purchased by Procedure Dense then the path $P(c_1, c_2)$ is $S_3$-tolerant. In addition, every subpath of an $S_3$-tolerant path is also $S_3$-tolerant. Hence $P(x_1, x_2)$ is $S_3$-tolerant and we have the following corollaries.

**Corollary 3.2.** For every two close nodes $x_1$ and $x_2$ on some path $P(c_1, c_2)$ that was purchased by Procedure Dense,

$$\text{dist}(x_1, x_2, E_{short}) \leq \text{dist}(x_1, x_2, G) + \Delta(x_1, x_2).$$

**Corollary 3.3.** For every two close centers $c_1, c_2 \in C_{rep}$, if $P(c_1, c_2)$ was purchased by the algorithm, then

$$\text{dist}(c_1, c_2, E_{short}) \leq \text{dist}(c_1, c_2, G) + 8\mu \cdot \log n.$$

**Lemma 3.7.** For every two close nodes $x_1, x_2$ such that $x_1, x_2 \in S_3$, $\text{dist}(x_1, x_2, E_{short}) \leq \text{dist}(x_1, x_2, G) + 17\mu \log n$.

**Proof:** Consider two close nodes $x_1, x_2 \in S_3$ and let $c_1$ and $c_2$ be the centers in $C_{rep}$ such that $x_1 \in \text{Cluster}(c_1)$ and $x_2 \in \text{Cluster}(c_2)$. Let $d = \text{dist}(x_1, x_2, G)$. We consider two cases, the first case is when the pair $(c_1, c_2)$ is marked by Procedure Dense and the second case is when it is not marked.

Consider the first case where the pair $(c_1, c_2)$ is marked. The pair $(c_1, c_2)$ is marked since there was some path $P(c_3, c_4)$ (could be that $P(c_3, c_4) = P(c_1, c_2)$) such that $P(c_3, c_4)$ was purchased by the algorithm and there are two nodes $y_1$ and $y_2$ on $P(c_3, c_4)$ such that $y_1 \in \text{Cluster}(c_1)$ and $y_2 \in \text{Cluster}(c_2)$.

By Corollary 3.2, $\text{dist}(y_1, y_2, E_{short}) \leq \text{dist}(y_1, y_2, G) + 8\mu \cdot \log n$. Since the distance from a node to its cluster center is at most $3\mu$, we get that

$$\text{dist}(x_1, x_2, E_{short}) \leq \text{dist}(x_1, y_1, E_{short}) + \text{dist}(y_1, y_2, E_{short}) + \text{dist}(y_2, x_2, E_{short}) \leq \text{dist}(x_1, y_1, E_{short}) + \text{dist}(y_1, y_2, G) + 8\mu \cdot \log n + \text{dist}(y_2, x_2, G) + 8\mu \cdot \log n + 6\mu \cdot \log n \leq \text{dist}(x_1, x_2, G) + 12\mu + 8\mu \cdot \log n$$

where the last inequality holds for every $\log n > 12$. Consider the second case where the pair $(c_1, c_2)$ is not marked. The path $P(c_1, c_2)$ was not purchased by the algorithm since there are a node $z$ on $P(c_1, c_2)$ such that $z \in \text{Cluster}(c_3)$ and $(c_1, c_3)$ is marked and $(c_2, c_3)$ is marked.

Using the same analysis as before, we get that

$$\text{dist}(x_1, z, E_{short}) \leq \text{dist}(x_1, z, G) + 12\mu + 8\mu \cdot \log n$$
dist(z, x_2, E_{short}) \leq dist(z, x_2, G) + 12\mu + 8\mu \cdot \log n.

Thus,
\[
\text{dist}(x_1, x_2, E_{short}) \\
\leq \text{dist}(x_1, z, E_{short}) + \text{dist}(z, x_2, E_{short}) \\
\leq \text{dist}(x_1, z, G) + 12\mu + 8\mu \cdot \log n \\
+ \text{dist}(z, x_2, G) + 12\mu + 8\mu \cdot \log n \\
\leq \text{dist}(x_1, x_2, G) + 17\mu \cdot \log n,
\]
where the last inequality holds for every log \( n > 24 \).

**Lemma 3.8.** For every two close nodes \( x_1, x_2 \),
\[
\text{dist}(x_1, x_2, E_{short}) \leq \text{dist}(x_1, x_2, G) + 33\mu \log n.
\]

**Proof:** By Lemma 3.6, we have that every shortest path \( P(y_1, y_2) \) between two close nodes \( y_1 \) and \( y_2 \) such that all nodes on \( P(y_1, y_2) \setminus \{y_2\} \) are not in \( S_3 \), satisfies
\[
\text{dist}(y_1, y_2, E_{short}) \leq \text{dist}(y_1, y_2, G) + \Delta(y_1, y_2) \leq \text{dist}(y_1, y_2, G) + 8\mu \cdot \log n.
\]

If \( P(x_1, x_2) \setminus \{x_2\} \cap S_3 = \emptyset \), we get
\[
\text{dist}(x_1, x_2, E_{short}) \leq \text{dist}(x_1, x_2, G) + \Delta(x_1, x_2) \leq \text{dist}(x_1, x_2, G) + 8\mu \log n.
\]
Otherwise, let \( z_1 \) (respectively, \( z_2 \)) be the first (respectively, last) node of \( S_3 \) on the path \( P(x_1, x_2, G) \) (it could be that \( z_1 = z_2 \)).

By Lemma 3.7, \( \text{dist}(z_1, z_2, E_{short}) \leq \text{dist}(z_1, z_2, G) + 17\mu \log n \).

We thus have,
\[
\text{dist}(x_1, x_2, E_{short}) \leq \text{dist}(x_1, z_1, E_{short}) + \text{dist}(z_1, z_2, E_{short}) + \text{dist}(z_2, x_2, E_{short}) \\
\leq \text{dist}(x_1, z_1, G) + 8\mu \log n + \text{dist}(z_1, z_2, G) + 17\mu \log n + \text{dist}(z_2, x_2, G) + 8\mu \log n = \text{dist}(x_1, x_2, G) + 33\mu \log n.
\]

**Long Distances.** Procedure Long-distances handles long distances. More specifically, we show the following. Consider a randomly selected set of vertices \( R_{long} \) obtained by taking each node with probability \( 9 \log n / n^{1-\frac{25}{6}} \). Procedure Long-distances finds a set of edges \( E_{long} \) with the following properties. First, the number of edges in \( E_{long} \) is \( O(n^{1+\epsilon}) \). Second, for every pair of nodes \( u, v \in R_{long} \),
\[
\text{dist}(u, v, E_{long}) \leq \text{dist}(u, v, G) + 2.
\]

Let \( \text{cater}(P, R) \) for a path \( P \) and a set of nodes \( R \) be the caterpillar that is obtained by taking the path \( P \) and connecting all nodes in \( \Gamma(P) \cap R \setminus P \) to the path \( P \) by a single edge. Then \( \text{Gain}(P, R, E') \) denote the set of pairs \( \{r_1, r_2\} \) such that \( r_1, r_2 \in \Gamma(P) \cap R \) and adding the caterpillar \( \text{cater}(P, r_1, r_2, E') \) improves their distance, i.e.,
\[
\text{dist}(r_1, r_2, E' \cup \text{cater}(P, R)) < \text{dist}(r_1, r_2, E') \text{, and let}
\text{value}(P, R, E') = \text{Gain}(P, R, E').
\]

Formally, Procedure Long-distances operates as follows. For every node \( v \) with degree at most \( n^3 \), i.e., \( |\Gamma(v)| \leq n^3 \), add to \( E_{long} \) all edges incident to \( v \). Next, choose a set \( R_{long} \) by independently sampling at random every node with probability \( 9 \log n / n^{1-\frac{25}{6}} \). For every pair of nodes \( \{r_1, r_2\} \) such that \( r_1, r_2 \in R_{long} \) do the following. Add \( \text{cater}(P, R_{long}) \) to \( E_{long} \) if \( 4 \cdot \text{value}(P, R_{long}, E_{long}) \cdot n^{1-\frac{25}{6}} > \text{cost}(\text{cater}(P, R_{long}), E_{long}) \), where \( P = P(r_1, r_2) \).

Here, we say that a node \( v \) is heavy if its degree is at least \( n^3 \), namely, \( \Gamma(v) \geq n^3 \) and light otherwise. Let \( \text{heavy-dist}(P) \) be the number of nodes in \( P \) with degree at least \( n^3 \).

Denote by \( E_{long} \) the subgraph under construction after the \( i \)-th iteration of Procedure Long-distances, namely, the subgraph \( E_{long} \) after considering the first \( i \) paths \( P(r_1, r_2) \). Note that \( E_{long} \) contains all incident edges to light nodes. Let \( P_{i} \) be the path considered during the \( i \)-th iteration.

Denote by \( \mathcal{T} \) the set of iterations in which Procedure Long-distances added the caterpillars considered to the constructed spanner \( E_{long} \) and by \( \mathcal{T}' \) the set of paths that their caterpillars were added to \( E_{long} \) using this process.

**Lemma 3.9.** The expected number of edges added by Procedure Long-distances is \( O(n^{1+\epsilon}) \).

**Proof:** In the first step of Procedure Long-distances all edges incident to light nodes are added to \( E_{long} \). It is not hard to verify that \( O(n^{1+\epsilon}) \) edges are added for this step.

Note that each pair of nodes \( u \) and \( v \) can belong to some set \( \text{Gain}(P_{i}, R_{long}, E_{long}) \) that is added to \( E_{long} \) in at most 5 iterations. To see this, let \( \hat{P} \) be the first path added to \( E_{long} \) such that the pair \( \{u, v\} \) belongs to \( \text{Gain}(\hat{P}, R_{long}, E_{long}) \). Assume \( u \) and \( v \) are not in \( \hat{P} \), and let \( u' \) and \( v' \) be the nodes in \( \hat{P} \) connected to \( u \) and \( v \) respectively. Let the distance between \( u \) and \( v \) be \( d \) and the distance between \( u' \) and \( v' \) be \( d' \). Note that \( d' \leq d + 2 \) and \( d \leq d' + 2 \), we get that the distance between \( u \) and \( v \) in \( \text{cater}(P, R_{long}) \) is at most \( d + 4 \). We get that the distance between \( u \) and \( v \) can be improved at most 5 times. If both \( u \) and \( v \) are in \( P \) then the shortest path between \( u \) and \( v \) in \( \text{cater}(P, R_{long}) \) is also the shortest path between them in \( G \), hence the distance between \( u \) and \( v \) can not improve anymore and the pair \( \{u, v\} \) belongs only to \( \text{Gain}(P, R_{long}, E_{long}) \). We are left with the case where exactly one of \( u \) and \( v \) is in \( \hat{P} \). In this case, \( d' \leq d + 1 \) and \( d \leq d' + 1 \), we get that the distance between \( u \) and \( v \) in \( \text{cater}(P, R_{long}) \).

**Procedure Main(G)**

\[
H \leftarrow \text{Short-distances}(G) \cup \text{Long-distances}(G)
\]

Return \( H \).
Procedure **Long-distances**($G$)

\[ E_{\text{long}} \leftarrow \emptyset \]

For every node $v$ such that $|\Gamma(v)| \leq n^6$, add to $E_{\text{long}}$ all edges incident to $v$.

Choose a set $R_{\text{long}}$ by independently sampling at random every node with probability $9\log n/n^{1-2\delta}$.

For every pair of nodes $r_1, r_2 \in R_{\text{long}}$ do:

Let $P = P(r_1, r_2)$

Add $\text{cater}(P, R_{\text{long}})$ to $E_{\text{long}}$ if $(4 \cdot \text{value}(P, R_{\text{long}}, E_{\text{long}}) \cdot n^{1-\delta}) \geq \text{cost}(\text{cater}(P, R_{\text{long}}), E_{\text{long}})$

Return $E_{\text{long}}$

is at most $d + 2$, therefore the distance between $u$ and $v$ can improve at most 3 times. This implies that the sum of values in $P$ is $O(n^{2\delta})$ as the expected number of nodes in $R_{\text{long}}$ is $O(n^{2\delta})$. By the rule used by Procedure **Long-distances** to add $\text{cater}(P, R_{\text{long}})$ to $E_{\text{long}}$, we thus have, $\sum_{i \in P} \text{cost}(\text{cater}(P_i, R_{\text{long}}), E_{\text{long}}^{-1}) \leq 4 \cdot n^{1-\delta}$. The lemma follows.

**Proof:** Consider two nodes $u$ and $v$ such that $|\Gamma(P(u, v))| \geq n^{1-2\delta}/3$. The probability that none of the nodes in $\Gamma(P(u, v))$ were chosen to $R_{\text{long}}$ is

\[(1 - \frac{1}{3 \cdot n^{2\delta}}) \approx \frac{1}{n^{\delta}}.\]

By the union bound we get that the probability that there is a pair of nodes $u, v$ such that $|\Gamma(P(u, v))| \geq n^{1-2\delta}/3$ and $\Gamma(P(u, v)) \cap R_{\text{long}} = \emptyset$ is at most $1/n$.

**Lemma 3.11.** Assume that $\Gamma(P(u, v)) \cap R_{\text{long}} \neq \emptyset$ for every two nodes $u, v$ such that $|\Gamma(P(u, v))| \geq n^{1-2\delta}/3$. Then for every pair of nodes $x$ and $y$, $\text{cost}(\text{cater}(P(x, y), R_{\text{long}}), E_{\text{long}}) \leq |R_P(x, y)| \cdot (n^{1-\delta} + 3) + n^{1-\delta}$.

**Proof:** First note that $\text{cost}(\text{cater}(P(x, y), R_{\text{long}}), E_{\text{long}}) \leq \text{heavy}_0(x, y) + |R_P(x, y)|$. To see this, recall that $E_{\text{long}}^0$ contains all incident edges to light nodes.

We thus need to show that $\text{heavy}_0(x, y) \leq |R_P(x, y)| \cdot (n^{1-\delta} + 2) + n^{1-\delta}$. We prove by induction on the heavy distance $\text{heavy}_0(x', y')$ for every pair of nodes $x'$ and $y'$, $\text{heavy}_0(x', y') \leq |R_P(x', y')| \cdot (n^{1-\delta} + 2) + n^{1-\delta}$.

If $\text{heavy}_0(x', y') \leq n^{1-\delta}$, the claim holds trivially.

Assume the claim holds for every pair of nodes $\{x'', y''\}$ such that $\text{heavy}_0(x'', y'') < d$ and consider pair of nodes $\{x', y'\}$ such that $\text{heavy}_0(x', y') = d$ for some $d > n^{1-\delta}$. Let $x_1$ be the node in $P(x', y')$ such that $\text{heavy}_0(x', x_1) = n^{1-\delta}$. Note that $|\Gamma(P(x', x_1))| \geq n^{1-2\delta}/3$ and thus $\Gamma(P(x', x_1)) \cap R_{\text{long}} \neq \emptyset$. Let $x_2$ be the node at distance 2 from $x_1$ on $P(x_1, y')$. By shortest path properties we get that $R_P(x_2, y') \cap \Gamma(P(x_2, y')) = \emptyset$. By the induction hypothesis, we have $\text{heavy}_0(x_2, y') \leq |R_P(x_2, y')| \cdot (n^{1-\delta} + 2) + n^{1-\delta}$. We thus have, $\text{heavy}_0(x', y') \leq n^{1-\delta} + 2 + |R_P(x_2, y')| \cdot (n^{1-\delta} + 2) + n^{1-\delta} \leq n^{1-\delta} + 2 + (|R_P(x', y')| - |R_P(x_2, y')|) \cdot (n^{1-\delta} + 2) + n^{1-\delta} \leq |R_P(x, y')| \cdot (n^{1-\delta} + 2) + n^{1-\delta}$.}

**Corollary 3.4.** With high probability, for every pair of nodes $x$ and $y$, $\text{cost}(\text{cater}(P, R_{\text{long}}), E_{\text{long}}) \leq |R_P(x, y)| \cdot (n^{1-\delta} + 3) + n^{1-\delta}$.

**Lemma 3.12.** With high probability, for every pair of nodes $u, v \in R_{\text{long}}$, $\text{dist}(u, v, E_{\text{long}}) \leq \text{dist}(u, v, G) + 2$.

**Proof:** To show the lemma, we need to consider a pair of nodes $u$ and $v$ in $R_{\text{long}}$ such that $\text{dist}(u, v, E_{\text{long}}) \leq \text{dist}(u, v, G)$, namely, that the shortest path $P_i = P(u, v)$ was not added to $E_{\text{long}}$. The path $P_i$ was not added to $E_{\text{long}}$ as $4 \cdot \text{value}(P_i, E_{\text{long}}^{-1}) \cdot n^{1-\delta} \leq \text{cost}(\text{cater}(P_i, R_{\text{long}}), E_{\text{long}}^{-1})$. By Corollary 3.4, $\text{cost}(\text{cater}(P_i, R_{\text{long}}), E_{\text{long}}) \leq |R_P| \cdot (n^{1-\delta} + 3) + n^{1-\delta} \leq 2|R_P| \cdot n^{1-\delta}$, where the last inequality follows...
from the fact that $|R_{P_i}| > 1$ (as $u, v \in R_{P_i}$) and straightforward calculations. We thus get $\text{value}(P_i) < |R_{P_i}|/2$.

Consider all pairs: $A = \{(s, t) \mid s \in \{u, v\}, t \in R_{P_i} \text{ and } \text{dist}(s, t, P_i) < \text{dist}(s, t, E_{\text{long}}^{-1})\}$. By definition $|A| \leq \text{value}(P_i)$, thus $|A| < \text{value}(P_i) < |R_{P_i}|/2$.

This implies that there is a node $w \in R_{P_i}$ such that $\text{dist}(u, w, E_{\text{long}}^{-1}) \leq \text{dist}(u, w, \text{cater}(P_i, R_{\text{long}}))$

and $\text{dist}(v, w, E_{\text{long}}^{-1}) \leq \text{dist}(w, v, \text{cater}(P_i, R_{\text{long}}))$.

Let $w'$ be the node on the path $P_i$ that has an edge to $w$ in $\text{cater}(P_i, R_{\text{long}})$. Note that $\text{dist}(u, w, G) = \text{dist}(u, w', G) + \text{dist}(w', v, G)$ and that

$\text{dist}(u, w, \text{cater}(P_i, R_{\text{long}})) = \text{dist}(u, w', G) + 1$ and

$\text{dist}(w, v, \text{cater}(P_i, R_{\text{long}})) = \text{dist}(w', v, G) + 1$, therefore

$\text{dist}(u, v, \text{cater}(P_i, R_{\text{long}})) \leq \text{dist}(u, v, G) + 1 + \text{dist}(w', v, G) + 1 = \text{dist}(u, v, G) + 2$.

Putting it all together. Finally, we prove the bound on the additive stretch of the spanner in the following lemma.

**Lemma 3.13.** The stretch of the spanner is $O(\mu \log n)$.

**Proof:** Consider two nodes $s$ and $t$ that are not close, namely, $|\Gamma(P(s, t), \mu)| > n^{1-23}$. Consider the first (respectively, last) node $y_1$ (respectively, $y_2$) on $P(s, t)$ such that $|\Gamma(P(y_1), \mu)| \geq n^{1-25}$ (respectively, $|\Gamma(P(y_2), \mu)| \geq n^{1-25}$). By Lemma 3.10 we have, $\Gamma(P(s, y_1), \mu) \cap R_{\text{long}} \neq \emptyset$ and $\Gamma(P(y_2), \mu) \cap R_{\text{long}} \neq \emptyset$. Let $z_1$ be a node on the path $P(s, y_1)$ such that there exists a node $r_1 \in R_{\text{long}}$ and $\text{dist}(z_1, r_1, G) \leq \mu$ and let $z_2$ be the last node on the path $P(y_2, t)$ such that there exists a node $r_2 \in R_{\text{long}}$ and $\text{dist}(z_2, r_2, G) \leq \mu$. Let $z'_1$ be the neighbor of $z_1$ on $P(s, z_1)$ and let $z'_2$ be the neighbor of $z_2$ on $P(z_2, t)$. We claim that $|\Gamma(P(s, z'_1), \mu)| \leq n^{1-25}$ and $|\Gamma(P(z'_2, t), \mu)| \leq n^{1-25}$. To see this, note that by definition of $y_1$, every node $x$ in $P(s, y_1) \setminus \{y_1\}$ satisfies $|\Gamma(P(s, x), \mu)| \leq n^{1-25}$. Since $z'_1 \in P(s, y_1) \setminus \{y_1\}$ we have $|\Gamma(P(z'_1, \mu)| \leq n^{1-25}$. Similarly, we can show that $|\Gamma(P(z'_2, \mu)| \leq n^{1-25}$.

By Lemma 3.8, $\text{dist}(z'_1, r_1, H) \leq \text{dist}(z'_1, r_1, G) + 33\mu \log n$, $\text{dist}(z'_2, t, H) \leq \text{dist}(z'_2, t, G) + 33\mu \log n$. By Lemma 3.12, $\text{dist}(r_1, r_2, H) \leq \text{dist}(r_1, r_2, G) + 2$. Note also that $\text{dist}(z_1, z_2, G) + \text{dist}(z_2, r_2, G) \leq 2\mu + \text{dist}(z_1, z_2, G)$. We thus have $\text{dist}(r_1, r_2, H) \leq \text{dist}(z_1, z_2, G) + 2\mu + 2$. In addition, since $H$ contains a log $n/3$ multiplicative spanner, we get $\text{dist}(z'_1, r_1, H) \leq \log n/3\text{dist}(z'_1, r_1, G) \leq \log n/3(\text{dist}(z'_1, r_1, G) + \text{dist}(z_1, r_1, G)) \leq \log n(1 + \mu)/3$.

Similarly, $\text{dist}(z'_2, t, H) \leq \log n(1 + \mu)/3$. Hence $\text{dist}(s, t, H) \leq \text{dist}(s, z'_1, H) + \text{dist}(z'_1, r_1, H) + \text{dist}(r_1, r_2, H) + \text{dist}(r_2, z'_2, H) + \text{dist}(z'_2, t, H) \leq \text{dist}(s, t, G) + O(\mu \log n)$.

### 3.1 New sublinear distance stretch spanners

We note that it is possible to tweak our construction from Section 3 to give a sublinear distance stretch spanner. More precisely, we have the following.

**Lemma 3.14.** One can efficiently construct a spanner $H$ with $O(n^{1 + 3/17})$ edges such that for every pair of nodes $s, t$, $\text{dist}(s, t, H) \leq O(\sqrt{\text{dist}(s, t, G)})$.

We now sketch the construction (we omit the complete details from this version). The construction involves $\log n$ iterations, where each iteration $i$ handles distances between $2^{i-1}$ to $2^i$ (we stop once $2^{2i-1} = n^{1-9/17}$). In each iteration $i$ invoke Procedure **Short-distances** from Section 3, but use $\sqrt{2^{-i}}$ instead of $\mu$ (in every place that uses $\mu$). Let $H_i$ be the constructed spanner for iteration $i$. Add the edges of $H_i$ to the constructed spanner $H$. To handle distances greater than $n^{1-9/17}$, we simply add the spanner $H$ from Section 3 to the constructed spanner $H$.

Following the analysis of Section 3, one can show that for every pair of nodes $s, t$ such that $\text{dist}(s, t, G) = O(2^i)$, the additive stretch for the pair $s, t$ in $H_i$ is within additive stretch $O(\sqrt{2^{-i}})$. This holds pairs of nodes of distance at most $n^{1-9/17}$. It is not hard to see that pairs of nodes $s, t$ of distance greater than $n^{1-9/17}$ are satisfied by the spanner $H$ from Section 3.

### 4 Conclusions

In this paper we make an additional step towards better understanding the picture of purely additive spanners. We present a new simple algorithm for $(1, 4)$-additive spanner with $O(n^{7/5})$ edges. In addition, we present a construction for additive spanners with $O(n^{4+\epsilon})$ edges and additive stretch of $O(n^{1/2 - 3\epsilon/2})$ for any $3/17 \leq \epsilon < 1/3$. It would be interesting to extend this result to any $0 < \epsilon < 1/3$. Our result for spanners of size $o(n^{4/3})$ gives the best additive stretch known so far (for the mentioned range). However, it is unclear that indeed a polynomial stretch is needed. Specifically, a major open problem in this area is the existence of a spanner of size $O(n^{4/3-\epsilon})$ for some fixed $\epsilon$ with constant or even polylog additive stretch.

**Acknowledgement** I am very grateful to my advisor, David Peleg, for many helpful discussions and for reviewing this paper.

**References**


