Directed Low Diameter Decomposition for Structured Digraphs

Shinwoo An
Bar-Ilan University, Ramat Gan, Israel

Arnold Filtser
Bar-Ilan University, Ramat Gan, Israel

Abstract

Low diameter decompositions, or LDDs for short, are a fundamental primitive in the design of efficient graph algorithms. Roughly speaking, an LDD is a distribution over partitions of the vertices into bounded-diameter clusters such that nearby vertices are likely to be clustered together. Recently, there has been growing interest in lifting the notion of LDDs into *directed graphs*. In particular, there are two natural directed analogues. The first is a directed LDD, where after removing a random subset of edges, every strongly connected component has a small diameter. The second is a quasipartition, which imposes the stronger requirement that whenever one vertex can still reach another after the edge removal, the two vertices must be close in the original directed metric. Every quasipartition yields an LDD, but the converse does not necessarily hold.

In this work, we initiate the systematic study of LDDs in structured directed graphs. As our first main result, we show that any directed graph with pathwidth pw admits an $(O(\mathsf{pw}), \Delta)$-LDD. This improves upon the previous best-known $(2^{O(\mathsf{pw}^2)}, \Delta)$-LDD construction, which was implicitly derived from the quasipartition result of Salmasi, Sidiropoulos, and Sridhar [SODA'19].

As our second result, we show that the integrality gap of the Directed Non-Bipartite Sparsest-Cut LP relaxation on an n-vertex graph with treewidth tw is $O(\mathsf{tw} \log n)$. This improves upon the $O(\mathsf{tw} \log^2 n)$ bound of Mémoli, Sidiropoulos, and Sridhar [ICALP'16, Algorithmica'18]. We obtain this result through the refined analysis of the quasipartition construction of Mémoli et al. for bounded treewidth graphs.

2012 ACM Subject Classification Theory of computation → Random projections and metric embeddings

Keywords and phrases Low diameter decomposition, quasipartition, metric embedding, pathwidth

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2026.16

Category APPROX

Related Version *Full Version*: `https://arxiv.org/abs/2606.31560`

Funding The work by An was supported by the Isreal Science Foundation grant #1042/22, Israel Science Foundation grant #800/22, and US-Israel Binational Science Foundation Grant #2022/131. The work by Filtser was supported by the Isreal Science Foundation grant #1042/22.

1 Introduction

Decomposing graphs into well-structured pieces is a central paradigm in the design of graph algorithms. In particular, low diameter decompositions (LDDs), which are a distribution over partitions of the vertices into bounded-diameter clusters such that nearby vertices are likely to be clustered together, have played a central role in undirected graphs. They serve as key primitive underlying a wide range of problems, a partial list of examples include: metric embedding [6, 18, 34, 41, 21], multi-commodity flow [32, 38], Steiner point removal [17], near-linear SDD solvers [10], Lipschitz extension [36], spanners [24, 28], and more. In

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2026).
Editors: Mohit Singh and Tom Gur; Article No. 16; pp. 16:1–16:24
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

undirected graphs, low diameter decompositions have been extensively studied not only in general graphs, but also in structured graph classes [32, 19, 1, 3, 20, 23, 15], where additional structure often leads to improved guarantees.

Recently, there has been a growing interest in extending low diameter decompositions to *directed graphs* [8, 9, 12, 39], despite the inherent technical challenges compared to the undirected setting. A notable application of this line of work is a recent near-linear time algorithm for the negative-weight single-source shortest path (SSSP) problem [9]. A *strongly connected component* of a directed graph G, an SCC in short, is a maximal subgraph H of G such that for every pair u, v of vertices in H, both u is reachable from v and v is reachable from u. For vertices u, v of G, let $d_G(u, v)$ denote the length of a shortest directed path from u to v in G, with $d_G(u, v) = \infty$ if no such path exists. For a subgraph $H \subseteq G$, the strong diameter of H is $\max_{u,v \in V(H)} d_H(u, v)$, whereas the weak diameter of H is $\max_{u,v \in V(H)} d_G(u, v)$. Here, $V(H)$ denotes the vertex set of H.

▶ **Definition 1** (Directed Low-Diameter Decomposition, [8, 9])**.** *Given an edge-weighted directed graph $G = (V, E, w)$ and a parameter $\Delta > 0$, a (β, Δ)-LDD is a distribution over subsets $S \subset E$ of edges, called a cutset, such that*
1. *every strongly connected component in $G \setminus S$ has a weak diameter of at most Δ, and*
2. *for every edge $e \in E$, $\Pr[e \in S] \leq \beta \cdot \frac{w(e)}{\Delta}$.*

We call β the loss parameter.

On the other hand, a parallel line of work has focused on *quasipartitions* for directed graphs [40, 31, 43], motivated by their connection to the integrality gap of the directed sparsest cut problem and the directed multicut problem [13].

▶ **Definition 2** ((Random) Quasipartition [40])**.** *Given an edge-weighted directed graph $G = (V, E, w)$ and a parameter $\Delta > 0$, a (β, Δ)-quasipartition is a distribution over a cutset $S \subseteq E$ such that*
- *for every u-v path in $G \setminus S$, $d_G(u, v) \leq \Delta$, and*
- *for every edge $e \in E$, $\Pr[e \in S] \leq \beta \cdot \frac{w(e)}{\Delta}$.*

We call β the loss parameter.

Notably, quasipartitions impose strictly stronger constraints than LDDs, and thus any (β, Δ)-quasipartition immediately yields a (β, Δ)-LDD, but not vice versa. Intuitively, in the quasipartition construction, one has to enforce that the distance between endpoints of every valid path is at most Δ. In contrast, the LDD construction only restricts the distance between endpoints of the paths within a *strongly connected component.* One specific example is directed acyclic graphs (DAGs), which are directed graphs avoiding directed cycles. It is easy to observe that every DAG itself is a $(0, \Delta)$-LDD for every parameter $\Delta > 0$ since its SCCs are isolated singletons. On the other hand, constructing a quasipartition with a small loss parameter on a DAG remains highly non-trivial, as a DAG still might contain arbitrarily long paths that must be cut. This separation is also reflected in a polynomial gap over the achievable loss parameters. While every directed graph on n vertices admits an $(O(\log n \cdot \log \log n), \Delta)$-LDD [12], there exists an n-vertex directed graph that does not admit any $(o(n^{1/7} \log^{4/7} n), \Delta)$-quasipartition [40].

Quasipartitions and LDDs are not merely theoretical concepts. Rather, they play a crucial role in driving distance-preserving graph sparsification scheme. Here, the goal is to obtain a simplified representation of the given graph while approximately preserving its distances. A number of such structures for directed graphs have been studied: including spanners [7], distance preservers [11], hopsets [33], and more. From the aforementioned lower bound for the quasipartition, [40] showed that one cannot efficiently embed a directed graph into *quasiultrametric space*, which is equivalent to the shortest path metric in directed trees. In contrast, by changing the viewpoint to LDDs, Assadi, Hoppenworth, and Wein [5] showed that one can (stochastically) embed any directed graph into two DAGs with distortion $O(\log^3 n)$. The distortion bound has been improved to $O(\log n \cdot \log\log n)$ by Filtser [22].

In structured directed graph classes, the distinction between the two decompositions remains poorly explored. First, all known LDD results for structured directed graphs have an indirect nature: Instead of being tailored directly to LDDs, they are either obtained via quasipartition constructions (bounded pathwidth and bounded treewidth [40, 43]) or simply bypassed by applying general graph bounds [12] (planar graphs [31]). In addition, there are no techniques that are specifically tailored to constructing LDDs in this setting.

Second, among the prior works that are indirectly derived from the quasipartition constructions, the resulting bounds fail to yield a clear improvement over the general graph bounds. Specifically, the loss parameter either suffers from a polynomial dependence on treewidth [40] or even worse, an exponential dependency on pathwidth [43]. Consequently, these results surpass the general bounds of [12] only when the parameters are strictly sublogarithmic in the graph size.

To overcome these barriers, a natural starting point is to focus on pathwidth [42]- one of the most standard and well-studied graph parameters which intuitively measures the degree to which a given graph resembles a path. Indeed, based on their structural advantage, bounded pathwidth graphs admit a wide range of efficient algorithms for fundamental graph problems [2, 25, 26, 30, 35]. From the perspective of LDDs, such graphs also enjoy improved guarantees on the loss parameter compared to general graphs in the undirected setting [3]. Hence, it is natural to ask whether the same phenomenon holds in the directed setting.

Motivated by these insights and the gap between the undirected and directed settings, we initiate the study of LDDs in structured directed graphs, especially bounded pathwidth graphs, moving beyond those obtained indirectly via quasipartition approaches. Our main result is that every directed graph of pathwidth pw admits an $(O(\mathsf{pw}), \Delta)$-LDD. This significantly improves the previous best-known result of $(2^{O(\mathsf{pw}^2)}, \Delta)$-LDD [40] in this setting.

▶ **Theorem 3.** *Let G be an edge-weighted directed graph and $\mathsf{pw}(G) = k$. Then, for any $\Delta > 0$, there is a $(O(k), \Delta)$-LDD of G.*

Classical constructions of (undirected/directed) LDDs rely on the ball carving technique [6], where we recursively compute clusters by removing regions defined by distance thresholds around a center. In graphs of bounded pathwidth, prior work for undirected graphs [3] bypasses the general bound of [6] via a path-carving approach: repeatedly choosing a shortest path as a backbone and removing a distance-threshold neighborhood around it, exploiting the path-like structure of the graph. Provided that one can efficiently remove this neighborhood [3, 29], this approach yields an improved loss bound compared to [6].

On the other hand, in general directed graphs, one natural difficulty is that, once we carve a nearby region around a center as a cluster, we cannot guarantee the diameter of the cluster. To overcome this difficulty, all prior works that directly compute LDDs [8, 9, 12, 39] recursively apply the ball carving approach for each carved cluster. It seems difficult to eliminate the $\log n$ dependency on the loss parameter under this approach.

Our main technical contribution is a novel carving procedure with respect to a shortest path, which we call *scissors carving*, that avoids this difficulty. In essence, our approach carefully decomposes a shortest path into shorter subpaths, and applies the ball carving to each segment in a "local" sense. The key insight is that, we can efficiently carve away a path

Table 1 Previous and new results on graph partition in directed graphs.

Graph Family	Partition type	Loss parameter β	Reference
General graphs	Undirected LDD	$\Theta(\log n)$ (UB and LB)	[6]
Genus g		$\Theta(\log g)$	[37]
Pathwidth pw		$\Theta(\log \mathsf{pw})$	[3]
Treewidth tw		$\Theta(\log \mathsf{tw})$	[23]
K_r-minor free		$\Theta(\log r)$	[15]
General digraphs	Directed LDD	$O(\log n \cdot \log\log n)$	[12]
General digraphs	Quasipartition	$\Omega(n^{1/7}\log^{4/7} n)$ (LB)	[14, 40]
Planar	Quasipartition	$O(\log^2 n)$	[31]
Treewidth ≤ 2	Quasipartition	$O(1)$	[43]
Treewidth tw	Quasipartition	$O(\mathsf{tw}\log n)$	[40]
Pathwidth pw	Quasipartition	$2^{O(\mathsf{pw}^2)}$	[43]
Pathwidth pw	Directed LDD	$O(\mathsf{pw})$	Theorem 3

whose length is sufficiently short[1]. In addition, we ensure that any carved cluster from such short paths has a small diameter. We believe that our technique provides a first step toward developing LDD techniques tailored to structured directed graphs, and that the technique itself is of independent interest.

As a secondary contribution, we revisit quasipartitions in directed graphs of bounded treewidth [40], and improve the $O(\mathsf{tw}\log^2 n)$ bound on the integrality gap of the directed non-bipartite sparsest cut [40] by a factor of $O(\log n)$, through a better analysis. Previous and new results about the integrality gap are summarized in Table 2. See Section 5 for the problem definitions and details of prior works on it.

▶ **Theorem 4.** *Let G be an n-vertex directed graph of treewidth tw. The integrality gap (also called the flow-cut gap) of the Directed Non-Bipartite Sparsest-Cut LP relaxation on G is $O(\mathsf{tw}\log n)$.*

2 Preliminaries

Throughout this paper, all missing proofs can be found in the full version. For a positive integer n, we denote by $[n]$ the set of all positive integers at most n.

Graph notations. Let $G = (V, E, w)$ be an edge-weighted directed graph, with n vertices. We often use $V(G)$ and $E(G)$ to denote the vertex set and the edge set in G, respectively. For a set A of vertices in G, we use $G \setminus A$ to denote the graph obtained from G by removing all vertices in A and their adjacent edges. We use $G[A]$ to denote a *induced subgraph* of G with respect to A. That is, $G[A] = G \setminus (V \setminus A)$. Similarly, for a set B of edges in G, we use $G \setminus B$ to denote the graph obtained from G by removing all edges in B. Two vertices u, v in

[1] See Section 3.2 for more details.

Table 2 Previous and new results on the integrality gap of the directed non-bipartite sparsest cut LP. Here, $\tilde{O}(\cdot)$ hides polylog n factor.

Graph Family	Integrality Gap	Ref.	Notes
General digraphs	$O(\sqrt{n})$	[27]	UB
	$\tilde{O}(n^{11/23})$	[4]	UB
	$\tilde{\Omega}(n^{1/7})$	[14]	LB
Planar	$O(\log^3 n)$	[31]	UB
Pathwidth pw	$2^{O(\mathsf{pw}^2)} \cdot \log n$	[40]+[43]	UB
Treewidth tw	$O(\mathsf{tw} \log^2 n)$	[40]	UB
Treewidth tw	$O(\mathsf{tw} \log n)$	Theorem 4	UB

G are *strongly connected* if there exist both a u-v path and a v-u path in G. A graph G is *strongly connected* if there is a u-v path for every $u, v \in V$. A *strongly connected component* of G, an SCC in short, is a maximal strongly connected subgraph of G. For a path P in G, the *length* of P is defined as the total edge weight in P. We use $d_G(u, v)$ to denote the *distance* between u and v in G. The *diameter* of G is defined as the maximum value of $d_G(u, v)$ among every pair $(u, v) \in V \times V$.

Treewidth and Pathwidth. A *tree-decomposition* (resp. *path-decomposition*) of an *undirected* graph G is a pair $(T, \{B_t\}_{t \in V(T)})$ consisting of a tree (resp. path) T and a family of sets $\{B_t\}_{t \in V(T)}$ of vertices in G such that

1. $V(G) = \bigcup_{t \in V(T)} B_t$,
2. for every edge $uv \in E(G)$, there exists a node $t \in V(T)$ such that $\{u, v\} \subseteq B_t$, and
3. for every vertex $v \in V(G)$, the set $\{t \in V(T) : v \in B_t\}$ induces a connected subtree (resp. subpath) of T.

The *width* of a tree-decomposition (resp. path-decomposition) $(T, \{B_t\}_{t \in V(T)})$ is defined as $\max_{t \in V(T)} |B_t| - 1$. The *treewidth* $\mathsf{tw}(G)$ (resp. *pathwidth* $\mathsf{pw}(G)$) of a graph G is the minimum width among all tree decompositions (resp. path decompositions) of G.

For a *directed* graph $G = (V, E)$, we consider the undirected graph $\bar{G}$ obtained by ignoring the edge directions and removing any parallel edges. Formally speaking, $\bar{G}$ has a vertex set $V(\bar{G}) = V$ and an edge set $E(\bar{G}) = \{\{u, v\} \mid (u, v) \in E \text{ or } (v, u) \in E\}$. We say that G has treewidth (resp. pathwidth) k if its underlying undirected graph $\bar{G}$ has treewidth (resp. pathwidth) k.

3 Technical Overview

Low diameter decomposition (LDD) for undirected graphs. Let G be an undirected graph. A (β, Δ)-LDD of G is a distribution over vertex partitions such that in every partition, each cluster has a diameter of at most Δ, and for every vertex pair u, v, the probability that u, v belong to different clusters is at most $\beta \cdot \frac{d_G(u,v)}{\Delta}$. We call β the *loss parameter*. Every n-vertex graph admits an $(O(\log n), \Delta)$-LDD for any $\Delta > 0$, which is tight [6].

Ball carving is a classical technique for computing an LDD. For a vertex $u \in V$ and parameter $R > 0$, a *ball* $B_G(u, R)$ is defined as the set of vertices $v \in V$ such that $d_G(u, v) \leq R$. We call R the *radius* of $B_G(u, R)$. The algorithm of Bartal [6] repeatedly takes a vertex w from the remaining graph, samples a radius R from an exponential distribution with the parameter $\Theta(\log n/\Delta)$, and removes the ball $B_G(w, R)$ from the graph as a single cluster. Using the memoryless property of the exponential distribution, Bartal [6] showed that the algorithm yields an $(O(\log n), \Delta)$-LDD.

Low diameter decomposition for bounded pathwidth graphs. Abraham et al. [3] showed that if G has pathwidth pw, it admits an $(O(\log \mathsf{pw}), \Delta)$-LDD. While this is a special case of a more general framework for minor-free graphs, we sketch their approach for the bounded pathwidth graphs. The following lemma was implicitly proved by several works (e.g., [3, 29]).

▶ **Lemma 5.** *Let G be an undirected graph and P be a shortest path in G. For any $\Delta > 0$, there is a set $N \subset P$ of vertices, called a net, such that*

1. *(Covering) $P \subset \bigcup_{v \in N} B_G(v, \Delta)$, and*
2. *(Sparsity) for every $u \in G$, $|B_G(u, 2\Delta) \cap N| = O(1)$.*

Their algorithm works in several rounds. In each round, for each connected component in the current graph, we pick a shortest path P connecting a vertex in the first bag and a vertex in the last bag (in a path decomposition), and compute a net N of that path using Lemma 5. Next, for each $v \in N$, we sample a radius R_v from a *truncated exponential distribution* with a parameter λ over the range $[\frac{\Delta}{4}, \frac{\Delta}{2}]$ so that each cluster has a diameter at most Δ. That is, the exponential distribution with the parameter λ conditioned on the event that the output belongs to $[\frac{\Delta}{4}, \frac{\Delta}{2}]$. We will define λ later. Finally, we carve $B_G(v, R_v)$ from the current graph as a single cluster.

Crucially, due to the covering property of Lemma 5, all vertices of P are carved away after the carving step. Moreover, since P is a path connecting a vertex in the first bag and a vertex in the last bag, P intersects every bag of the path decomposition. Therefore, the pathwidth of the graph decreases by at least one. Thus, the algorithm terminates after pw rounds. Due to the sparsity property of Lemma 5, we can show that for every pair $u, v \in V$, at most $\nu = O(\mathsf{pw})$ balls, called threateners, affect the probability that u, v belong to different clusters.

Roughly speaking, the truncated exponential distribution behaves similarly to the exponential distribution, i.e., retaining a memoryless property, up to a small error. These errors from the sampled radii of different threateners are accumulated. In order to keep the total error negligible, the parameter should be at least logarithmic in the number of threateners. In light of this, Abraham et al. [3] set the parameter $\lambda = \Theta(\frac{\log \nu}{\Delta}) = \Theta(\frac{\log \mathsf{pw}}{\Delta})$, yielding an $(O(\log \mathsf{pw}), \Delta)$-LDD of G. This path-carving framework is the key ingredient that we will adapt to the directed setting later.

Low diameter decomposition for general directed graphs. A directed analogue of the low diameter decomposition (Definition 1) was very recently introduced by Bernstein, Gutenberg, and Wulff-Nilsen [8] (though not stated explicitly), who showed that every n-vertex directed graph G admits an $(O(\log^2 n), \Delta)$-LDD for every parameter $\Delta > 0$. Later, Bernstein, Nanongkai, and Wulff-Nilsen [9] explicitly defined the notion of a low diameter decomposition and presented a simpler algorithm yielding an $(O(\log^2 n), \Delta)$-LDD. This bound has been improved to $O(\log n \cdot \log\log n)$ by Bringmann et al. [12].

The algorithm of Bernstein et al. [9] follows the ball carving framework: it samples a radius and carves either an in-ball or an out-ball[2] around a chosen vertex. A key difficulty in the directed setting is that, unlike the undirected case, one cannot guarantee the weak

[2] We defer the definitions of an in-ball and an out-ball to Section 3.2

diameter bound of an SCC contained in a single ball. To resolve this, the algorithm of [9] recursively applies the same carving procedure to every SCC until all SCCs have a small weak diameter. By carefully selecting centers, Bernstein et al. [9] managed to bound the recursion depth by $O(\log n)$. This yields an $O(\log^2 n)$ bound on the loss parameter.

3.1 Low diameter decomposition for special classes of directed graphs

The aforementioned issue remains even for special classes of directed graphs, such as those with bounded pathwidth, bounded treewidth, or planar graphs. We briefly review some prior works that circumvented this issue. We remark that while all prior work primarily addressed random quasipartitions for those graph classes, their quasipartition results directly translate to low diameter decompositions with the same loss parameter[3]. Consequently, we present the prior work from the viewpoint of low diameter decompositions.

Separator-based approach. Mémoli, Sidiropoulos, and Sridhar [40] showed that every n-vertex directed graph with treewidth tw admits an $(O(\mathsf{tw}\log n), \Delta)$-LDD for every parameter $\Delta > 0$. Their approach is based on a balanced separator. We call $U \subset V(G)$ a *balanced separator* if the number of vertices in each connected component of $G \setminus U$ is at most $\frac{2}{3}|V(G)|$. It is well known that every graph G with treewidth tw admits a *balanced separator* $U \subset V(G)$ of size at most $\mathsf{tw} + 1$.

The algorithm of [40] computes a balanced separator U, samples a radius, and constructs both in-balls and out-balls around each vertex of U. Next, it adds the boundary edges of these balls to the cutset S. Then, it recursively proceeds on each SCC of $G \setminus U$. Critically, any SCC of $G \setminus S$ is either disjoint from U or has a weak diameter of at most Δ. The loss parameter at a single recursion level is $O(\mathsf{tw})$ since the algorithm computes a cutset through $2|U| = O(\mathsf{tw})$ balls. In addition, the recursion depth is $O(\log n)$. Overall, the algorithm of [40] yields an $(O(\mathsf{tw}\log n), \Delta)$-LDD.

Kawarabayashi and Sidiropoulos [31] showed that every n-vertex planar directed graph admits an $(O(\log^2 n), \Delta)$-LDD. Instead of a (balanced) vertex separator, they utilized the shortest path separator of [44], which consists of a constant number of shortest paths. Again, the $O(\log n)$ factor arises from the fact that the recursion depth is $O(\log n)$. We remark that it seems difficult to avoid the $O(\log n)$ factor in any separator-based approach since the recursion depth is $O(\log n)$ and the edge-cut probability incurred in each round must be handled independently.

LDD of constant loss parameter for bounded pathwidth directed graphs. Salmasi, Sidiropoulos, and Sridhar [43] showed that every n-vertex directed graph G with pathwidth pw admits a $(2^{O(\mathsf{pw}^2)}, \Delta)$-LDD for every parameter $\Delta > 0$. This is the only known LDD result for structured directed graphs where the loss parameter is independent of n.

We briefly sketch their approach at a high level. We start by setting the current graph to $H = G$. In each round, [43] computes a u-v shortest path P that connects the first and last bags of the path decomposition. Instead of carving the balls around the net vertices along P, the algorithm computes a random shift $z \in [0, \Delta]$ and adds all edges crossing distances of $i\Delta + z$ from u for every integer i, to the cutset. Notably, after this, they remove the edge set of P from H (recall that in the ball carving framework, the cutset is removed) and recurse. Finally, the algorithm repeats the entire procedure in the reversed direction. The recursion depth is $O(\mathsf{pw}^2)$.

[3] From Definitions 1 and 2, it is easy to check that every (β, Δ)-quasipartition is a (β, Δ)-LDD.

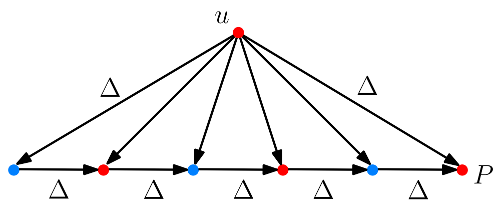

Figure 1 Counter-example of Lemma 5 for directed graphs. Consider a directed graph G consisting of a shortest path P of n edges of weight Δ, an apex vertex u, and an edge (u, p) of weight Δ for every vertex $p \in P$. Then, for any subset $N \subseteq P$ that satisfies the covering property, i.e., $P \subset \bigcup_{v \in N} B^+_G(v, \Delta)$, notice that each ball covers at most $O(1)$ vertices of P. Therefore, $|N| = \Theta(n)$. The blue points indicate the points in N. Since $d_G(u, v) = \Delta$ for every $v \in P$, $N \subseteq B^+_G(u, 2\Delta)$.

Salmasi, Sidiropoulos, and Sridhar [43] managed to eliminate the dependency on n by carving away the boundary edges of concentric layers with a random shift. On the other hand, they remove the edge set of P instead of the cutset edges. This causes a more involved analysis of bounding the weak diameter, and yields a $(2^{O(\mathsf{pw}^2)}, \Delta)$-LDD. We also remark that their analysis crucially utilizes the structure of the bounded pathwidth graphs, making it seemingly difficult to lift their result to bounded treewidth graphs or planar graphs.

3.2 Our solution: Scissors Carving

Our algorithm departs from the previous approaches. We develop a new carving procedure, which we call *scissors carving*. This is our main technical contribution that enables us to achieve an $(O(\mathsf{pw}), \Delta)$-LDD. Let $G = (V, E, w)$ be an edge-weighted directed graph. For a vertex $u \in V$ and a parameter $R > 0$, we define the *out-ball* $B^+_G(u, R)$ and the *in-ball* $B^-_G(u, R)$ as

$$B^+_G(u, R) := \{x \in V \mid d_G(u, x) \leq R\}, \text{ and } B^-_G(u, R) := \{x \in V \mid d_G(x, u) \leq R\},$$

respectively. We call R the *radius*.

At a high level, suppose we are given a target x-y path P that we want to carve away. We compute an out-ball around x and an in-ball around y with radii proportional to $d_G(x, y)$, and then remove their boundary edges. Then, we recurse on each SCC intersecting P.

The key insight is that this approach always makes progress for each recursive SCC: either the length of the portion of P contained in each SCC, called the *forward distance*, is reduced, or the pathwidth of the SCC is decreased. Once the length of the current path becomes sufficiently small, we apply the same procedure in the reverse direction, i.e., computing an in-ball around x and an out-ball around y, to reduce the *backward distance* $d_G(y, x)$. The separation between the forward-reduction stage and the backward-reduction stage is essential, as reducing the backward distance using two balls is only possible once the forward distance is sufficiently small.

We demonstrate the two advantages of the scissors carving framework as follows. First, we recursively compute two balls of geometrically decreasing radii. Even though we may compute many balls across the recursion, most of them have radii much larger than Δ. This allows us to bound the edge cut probability by $\Pr[e \in S] \leq O(1) \cdot \frac{w(e)}{\Delta}$.

Second, after both the forward-reduction stage and the backward-reduction stage, we ensure that both the forward distance and the backward distance of each remaining SCC H are sufficiently small. Using this structural advantage, we show that any SCC of H fully contained in a single ball of radius Δ around $u \in H$ has a weak diameter of at most $O(\Delta)$.

This overcomes the limitation that, in general directed graphs, an SCC contained in a single ball of radius Δ can have an arbitrarily large diameter. This allows us to ensure the desired diameter bound.

Based on the scissors carving procedure, we compute an $(O(\mathsf{pw}), \Delta)$-LDD following the framework of [3]. Specifically, we start by setting the current graph to $H = G$. At each stage, we take a shortest path P in H that we want to carve away. In particular, we choose P such that $\mathsf{pw}(H \setminus P) < \mathsf{pw}(H)$. Then, we apply scissors carving to P and update the cutset S. This ensures that every SCC of $H \setminus S$ either has a bounded weak diameter or a reduced pathwidth (See Lemma 6). Furthermore, we shall guarantee that every SCC disjoint from P has a reduced pathwidth. Then, we recurse on every SCC of $H \setminus S$ whose weak diameter is greater than Δ. By taking the union S of the cutsets computed from all recursive calls, we ensure that every SCC of $G \setminus S$ has a weak diameter of at most Δ. The recursion depth is at most $\mathsf{pw}(G)$. Due to the union bound of probability, the probability bound of Lemma 6 yields $\Pr[e \in S] \le O(\mathsf{pw}(G)) \cdot \frac{w(e)}{\Delta}$.

▶ **Lemma 6** (informal).[4] *Let H be a strongly connected subgraph of G. There is a randomized algorithm that computes a cutset $S \subset E(H)$ such that for every strongly connected component C of $H \setminus S$, either $\mathsf{pw}(C) < \mathsf{pw}(H)$ or the weak diameter of C is $O(\Delta)$. Moreover, for every $e \in E(H)$, $\Pr[e \in S] \le O(1) \cdot \frac{w(e)}{\Delta}$.*

In the rest of this section, we explain the core idea of the scissors carving technique and give a proof sketch of Lemma 13, especially for the first statement. We begin by defining some notation used in this paper. For a vertex set $C \subseteq V$, we denote the set of edges whose source endpoint is in C and target endpoint is outside C by $\delta^+(C)$. Similarly, we denote the set of edges whose source endpoint is outside C and target endpoint is inside C by $\delta^-(C)$.

Let H be a subgraph of G and P be a shortest path in G whose two endpoints are contained in H. For a strongly connected *subgraph* C of H, let u and v be the first and the last vertices of $P \cap C$ (along the order of P), respectively. We define the *surviving subpath* of P with respect to C, denoted as $P[C]$, as the u-v subpath of P. Note that $P[C]$ may not be fully contained in C. We define the *forward distance* and the *backward distance* of C (with respect to P) as $d_G(u, v)$ and $d_G(v, u)$, respectively, and denote them by $f_P(C)$ and $b_P(C)$, respectively. Note that we analyze these distances with respect to the shortest paths in G, rather than the current subgraph C. This is sufficient for our purpose since our goal is to bound the *weak diameter* of each SCC.

We use the following lemma. See Figure 2 for an illustration.

▶ **Lemma 7.** *Let $B_1 = B^*_G(v, R)$ and $B_2 = B^{*'}_G(v, R)$ with $*, *' \in \{+, -\}$, and let C be a strongly connected subgraph of G. Then,*

- *every SCC of $C \setminus (\delta^*(B_1) \cup \delta^{*'}(B_2))$ is either fully contained in $B_1 \cap B_2$ or disjoint from $B_1 \cap B_2$, and*
- *every SCC of $C \setminus (\delta^*(B_1) \cup \delta^{*'}(B_2))$ is either fully contained in B_1, or fully contained in $B_2 \setminus B_1$, or disjoint from $B_1 \cup B_2$.*

Next, we explain the scissors carving procedure. We are given a strongly connected subgraph H of G and a shortest path P as input, such that the two endpoints of P are contained in H, and $\mathsf{pw}(H \setminus P) < \mathsf{pw}(H)$. The algorithm consists of three stages, and the first two stages operate recursively. As we will see below, these stages aim to reduce the forward distance, reduce the backward distance, and bound the weak diameter of each SCC, respectively.

[4] Here, we give an informal statement in order to hide several technical terms behind it. See Lemma 13 for details.

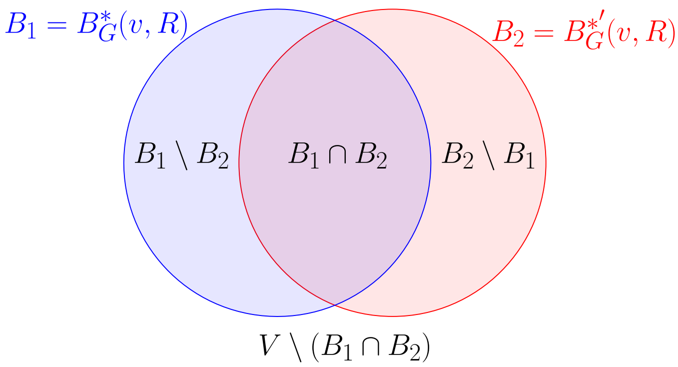

Figure 2 Due to the definition of $\delta^*(\cdot)$, every SCC of $H \setminus \delta^*(B_1)$ is either contained in B_1 or disjoint from B_1. The same holds symmetrically for B_2. Therefore, every SCC of $H \setminus (\delta^*(B_1) \cup \delta^{*'}(B_2))$ is fully contained in one of the regions of the Venn diagram. This proves both items of Lemma 7 for the special case of $C = G$. See Section 4 for the general case.

First stage: reducing forward distance. In the first stage we are given a strongly connected subgraph C of H as input. Let $P[C] = \langle u, \ldots, v \rangle$ and $\tau = f_P(C)$. We sample the radius R uniformly at random from $[\frac{1}{2}\tau, \frac{2}{3}\tau]$, and compute two balls $B_1 = B_G^+(u, R)$, $B_2 = B_G^-(v, R)$ and add their boundary edges $\delta^+(B_1) \cup \delta^-(B_2)$ into the cutset S. We then recursively apply this approach to each SCC of $C \setminus S$ contained in $B_1 \cup B_2$. We halt the first stage when $f_P(C) \leq \Delta$, and proceed to the second stage. This completes the first stage.

We show that the above procedure ensures that each resulting SCC for the recursive calls has a reduced forward distance, while the remaining SCCs have a reduced pathwidth. Hence, it suffices to proceed with the SCCs of the former type, i.e., those with a reduced forward distance. To see this, since we take R from the range $[\frac{1}{2}\tau, \frac{2}{3}\tau]$ and $P[C]$ is a shortest path in G, for every $w \in P[C]$, either $d_G(u, w) \leq R$ or $d_G(w, v) \leq R$. Hence, $P \subseteq B_1 \cup B_2$. In addition, due to Lemma 7, every SCC of $C \setminus S$ is either fully contained in B_1, or fully contained in $B_2 \setminus B_1$, or disjoint from $B_1 \cup B_2$. Let C' be an SCC of $C \setminus S$ and $P[C'] = \langle x, \ldots, y \rangle$. Since $P[C']$ is a subpath of P and $C' \subseteq C$, $P[C']$ is a subpath of $P[C]$. If C' is contained in B_1,

$$d_G(x, y) \leq d_G(u, y) \overset{B_1}{\leq} R \leq \frac{2}{3}\tau.$$

Otherwise, if C' is contained in $B_2 \setminus B_1$, we have that

$$d_G(x, y) \leq d_G(x, v) \overset{B_2}{\leq} R \leq \frac{2}{3}\tau.$$

Finally, if C' is disjoint from $B_1 \cup B_2$, $P \subseteq B_1 \cup B_2$ implies that C' is a subset of $C \setminus P$. Then, we have that $\mathsf{pw}(C') \leq \mathsf{pw}(C \setminus P) \leq \mathsf{pw}(H \setminus P) < \mathsf{pw}(H)$. Hence, we conclude that every SCC of $G \setminus S$ either has a reduced forward distance or a reduced pathwidth, as desired.

Second stage: reducing backward distance. In the second stage, we are given a strongly connected subgraph C of H as input with the additional property that $f_P(C) \leq \Delta$. Let $P[C] = \langle u, \ldots, v \rangle$ and $b_P(C) = \tau'$. Importantly, throughout this stage, we ensure that the condition on $f_P(C)$ always holds. Our algorithm is almost identical to that of the first stage, except that we compute two balls in the reversed direction and sample the radius R from a narrower interval. The reason we choose a narrower interval is to reduce the backward distance by a constant fraction.

More precisely, we sample R uniformly at random from $[\frac{1}{2}\tau', \frac{13}{24}\tau']$, compute two balls $B_1 = B^-_G(u, R), B_2 = B^+_G(v, R)$ and add $\delta^-(B_1) \cup \delta^+(B_2)$ to the cutset S. We recursively apply this approach to each SCC of $C \setminus S$ contained in $B_1 \cup B_2$. We halt the second stage when $b_P(C) \le 8\Delta$. Here, 8 is also a carefully chosen constant in order to bound the backward distance.

We show that a symmetrical property holds for this stage: Every SCC contained in $B_1 \cup B_2$ has a reduced backward distance, and every SCC disjoint from $B_1 \cup B_2$ has a reduced pathwidth. Let C' be an SCC of $C \setminus S$ and $P[C'] = \langle x, \ldots, y\rangle$. First, we bound $b_P(C')$ in the case where $C' \subseteq B_1 \cup B_2$. From Lemma 7, without loss of generality, let us assume $C' \subseteq B_1$. The other case where $C' \subseteq B_2 \setminus B_1$ is analogous. From the argument of the first stage, $P[C']$ is a subpath of $P[C]$ and $d_G(u, v) = f_P(C) \le \Delta$. In addition, due to the triangle inequality, we have that:

$$d_G(y, x) \overset{\triangle}{\le} d_G(y, u) + d_G(u, x) \le d_G(y, u) + d_G(u, v) \overset{B_1}{\le} R + d_G(u, v) \le R + \Delta \le \frac{13}{24}\tau' + \Delta.$$

Since the halting condition is $\tau' \le 8\Delta$, we have that $\frac{13}{24}\tau' + \Delta \le \frac{2}{3}\tau'$ throughout the second stage. Hence, the backward distance is reduced by $2/3$.

Next, we bound the pathwidth of C' in the case where C' is disjoint from $B_1 \cup B_2$. The difference is that $P[C]$ may not be fully contained in $B_1 \cup B_2$. For simplicity, let us assume that u and v are the leftmost and rightmost vertices of P in the path decomposition, respectively. That is, for two bags X_u and X_v containing u and v respectively, every bag X that intersects P, lies between X_u and X_v in the path decomposition[5]. We consider a shortest v-u path P' of length τ' in G. Since $R \ge \frac{1}{2}\tau'$, it is easy to check that $P' \subseteq B_1 \cup B_2$. Moreover, since P' intersects every bag X between X_u and X_v, P' intersects every bag X that intersects P. Since C' is disjoint from $B_1 \cup B_2$, it is disjoint from P' as well. Hence, for each bag X intersecting $P \cap C'$, we have that:

$$|C' \cap X| < |C' \cap X| + |P' \cap X| = |(C' \cup P') \cap X| \le |C \cap X|.$$

This establishes that $\mathsf{pw}(C') < \mathsf{pw}(C)$, as desired. See Figure 3(b) for an illustration.

Third stage: reducing the weak diameter. Finally, the third stage serves as the base case. We are given a strongly connected subgraph C of H as input, where the two previous stages guarantee that $f_P(C) \le \Delta$ and $b_P(C) \le 8\Delta$. One might ask whether $f_P(C), b_P(C) = O(\Delta)$ directly implies that the weak diameter of C is $O(\Delta)$. This is not true in general. What we can guarantee is only the (bi-)distance between two endpoints of $P[C]$, which does not necessarily bound the distance between every pair of vertices in C.

Let $P[C] = \langle u, \ldots, v\rangle$. We compute two balls $B^+_G(u, R_1)$ and $B^-_G(u, R_2)$ around the same vertex u, where we sample R_1, R_2 uniformly at random from $[\Delta, 2\Delta]$ and $[9\Delta, 10\Delta]$, respectively. Then, we add $\delta^+(B_1) \cup \delta^-(B_2)$ into the cutset S. Every SCC contained in $B_1 \cap B_2$ has a small weak diameter, since for every pair of vertices in $B_1 \cap B_2$, we can draw a detour of length $O(\Delta)$ through u. In addition, it is straightforward to verify that $P[C] \subset B_1 \cap B_2$. Subsequently, every SCC disjoint from $B_1 \cap B_2$ has a reduced pathwidth. Due to Lemma 7 we conclude that every SCC of $G \setminus S$ after the third stage either has a weak diameter of at most Δ or a reduced pathwidth. This completes the proof sketch of Lemma 6.

[5] Indeed, we can guarantee this assumption by carefully taking certain subpaths in each recursion step. See Section 4 for details.

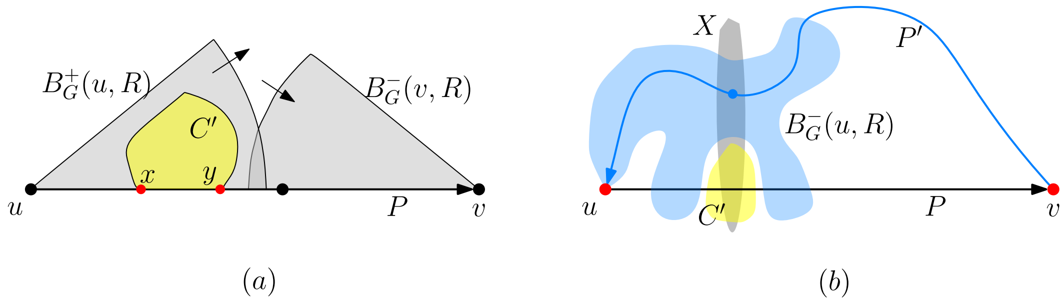

Figure 3 (a) Illustration of the first stage. The gray region denotes two balls $B_G^+(u,R)$ and $B_G^-(v,R)$ where R is sampled from $[\frac{1}{2}\tau, \frac{2}{3}\tau]$ with $\tau = d_G(u,v)$. For an SCC C' (yellow region) contained in $B_1 = B_G^+(u,R)$ and $x,y \in P \cap C'$, we have that $d_G(x,y) \le d_G(u,y) \le \frac{2}{3}\tau$, since P is a shortest path in G. (b) Illustration of the second stage. The blue region denotes $B_G^-(u,R)$, and P' represents the shortest v-u path in G. In addition, the yellow region denotes an SCC C' which is disjoint from $B_G^-(u,R)$. Let us assume that u and v are the leftmost and rightmost vertices of P with respect to the path decomposition. Then, for every bag X that intersects $P \cap C'$, X also intersects P' (blue vertex). Since C' is disjoint from P', we can derive $\mathsf{pw}(C') < \mathsf{pw}(C)$.

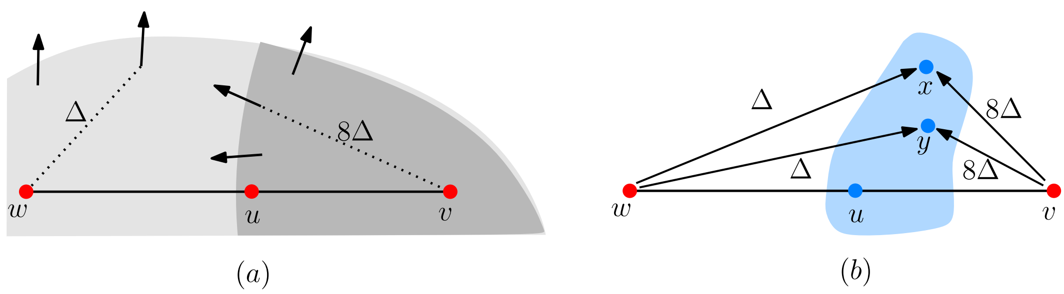

Figure 4 (a) Illustration of the graph partition from the first two stages. In the first stage, an out-ball $B_G^+(w,\Delta)$ (light gray region) containing u,v is computed. In the second stage, an out-ball $B_G^+(v,8\Delta)$ (dark gray region) containing u is computed. (b) At the start of the third stage, an SCC (blue region) might have a large diameter, as no upper bounds for $d_G(x,u)$ and $d_G(y,u)$ are assumed. In the third stage, we reduce the weak diameter of an SCC intersecting P by computing two balls of radii $O(\Delta)$ around u.

▶ Remark. One may expect a logarithmic loss parameter of $O(\log(\mathsf{ppw}(G)))$ as in [3]. However, our scissors carving framework differs from previous ball carving in the following sense. In previous work, once the interior of a ball was carved away, the resulting cluster became "safe", meaning that it was no longer affected by further ball carving steps. This allowed [3] to sample the radius from a (truncated) exponential distribution and utilize its (almost) memoryless property, ensuring them to bound the loss parameter by the logarithmic number of threateners.

In contrast, in our scissors carving, once we compute two balls that cover a shortest path, the interiors of these balls are conveyed to the next round. That is, each edge can be threatened multiple times regardless of whether it is inside a ball or not. Thus, the cut probability becomes linear in the number of threateners. We also remark that the same phenomenon is observed in the directed treewidth case [40].

4 Directed LDDs for Bounded-Pathwidth Directed Graphs

▶ **Theorem 3.** *Let G be an edge-weighted directed graph and $\mathsf{pw}(G) = k$. Then, for any $\Delta > 0$, there is a $(O(k), \Delta)$-LDD of G.*

In this section, we prove Theorem 3. While we state our result in terms of pathwidth, we find it more convenient to use the notion of *path partition-width* (see Definition 8 below). This is a directed analogue of the (undirected) tree partition-width [16] for the directed graphs with small pathwidth.

▶ **Definition 8** (Path Partition)**.** *A* path partition *of a directed graph $G = (V, E)$ is a path $\mathcal{P}$ whose vertices are bijectively associated with the sets of a partition $\mathcal{X} = \{X_1, X_2, \dots, X_m\}$ of V, called* bags*, such that for every $(u, v) \in E$, there exist $X_i, X_{i+1} \in \mathcal{X}$ for some $1 \le i \le m-1$ such that $\{u, v\} \subset X_i \cup X_{i+1}$. The* width *of $\mathcal{X}$ is $\max_{i\in[m]}\{|X_i|\}$. The path partition-width $\mathsf{ppw}(G)$ of G is the minimum width among all path partitions of G.*

To begin with, we apply an edge-preserving isometric embedding from a directed graph with bounded pathwidth into another directed graph with bounded path partition-width. Note that our embedding is not new. Filtser et al. [23] defined a similar embedding from (undirected) bounded treewidth graphs into graphs of bounded tree partition-width[6] graphs. Although not stated explicitly, their embedding is also an edge-preserving isometric embedding.

▶ **Definition 9.** (Edge-preserving isometric embedding)[23] *Let $G = (V, E, w)$, $H = (U, E_H, w_H)$ be directed graphs. We say that a map $\phi : V \to U$ is an* edge preserving isometric embedding *if the following conditions hold:*

- *(isometry) $d_G(u, v) = d_H(\phi(u), \phi(v))$ for every $u, v \in V$,*
- *(edge preservation) there is an injective mapping $\bar{\phi} : E \to E_H$ such that for every $e \in E$, $w(e) = w_H(\bar{\phi}(e))$ and $\bar{\phi}(e)$ belongs to a shortest $\phi(u)$-$\phi(v)$ path in H.*

While the ultimate goal is to design an LDD for directed graphs of bounded pathwidth, our construction algorithm is formally defined on graphs of bounded path partition-width. We remark that a similar approach was used by Filtser et al. [23] to compute an LDD for (undirected) graphs of bounded treewidth. There is an edge-preserving isometric embedding from graphs with bounded pathwidth into graphs with bounded path partition-width. In addition, an LDD of the embedded graph can be transferred back to the original graph without any additional loss.

▶ **Lemma 10.** *Let $G = (V, E, w)$ be a directed graph. Then, there is an edge-preserving isometric embedding from G into a directed graph $H = (U, E_H, w_H)$ such that $\mathsf{ppw}(H) \le \mathsf{pw}(G) + 1$.*

▶ **Lemma 11.** *Let ϕ be an edge-preserving isometric embedding from G into H, and $\mathcal{D}$ be a (β, Δ)-LDD of H. Then there is a (β, Δ)-LDD of G.*

Based on Lemmas 10 and 11, it suffices to show the existence of $(O(k), \Delta)$-LDD for the directed graphs with path partition-width at most k. Throughout this section, we let G be a directed graph, and let $(\mathcal{P}, \mathcal{X} = \{X_1, X_2, \dots, X_m\})$ be the path partition of G with width $\mathsf{ppw}(G)$. If it is clear from the context, we slightly abuse the notation and simply

[6] Tree partition can be viewed as a natural generalization of path partition, in the same way that tree decomposition generalizes path decomposition.

denote the path partition $(\mathcal{P}, \mathcal{X})$ by $\mathcal{P}$. Furthermore, we may assume that G is strongly connected as otherwise, we can independently execute our algorithm on each SCC of G. For a subgraph H of G, we define the *induced path partition* of H with respect to $\mathcal{P}$ as $(\mathcal{P}, \{X_i \cap V(H) \mid X_i \in \mathcal{X}\})$, and denote it by $\mathcal{P}[H]$. We denote the width of $\mathcal{P}[H]$ by $w(\mathcal{P}[H])$. Note that for every subgraph H of G, $\mathsf{w}(\mathcal{P}[H]) \leq \mathsf{w}(\mathcal{P}[G]) = \mathsf{ppw}(G)$ trivially holds.

The remainder of this section is devoted to proving the following lemma.

▶ **Lemma 12.** *For any $\Delta > 0$, there is an $(O(\mathsf{ppw}(G)), \Delta)$-LDD of G.*

Next, we present a randomized recursive algorithm Directed-LDD, which receives as input a directed graph $G = (V, E, w)$, a path partition $\mathcal{P}$ of G, a strongly connected subgraph H of G, and a parameter $\Delta > 0$. In addition, Directed-LDD computes a (randomized) cutset of H whose distribution induces a $(O(\mathsf{w}(\mathcal{P}[H])), \Delta)$-LDD of H. See Algorithm 1 for the pseudocode. The algorithm immediately terminates if $\mathsf{w}(\mathcal{P}[H]) = 0$, which implies that H is an empty subgraph. For the other case where $\mathsf{w}(\mathcal{P}[H]) > 0$, Algorithm 1 calls Scissors-Carving algorithm[7], whose function is summarized in the following lemma. We defer the detailed description of Scissors-Carving to Sections 4.1.

▶ **Lemma 13.** *There is an algorithm Scissors-Carving$(G, \mathcal{P}, H, \bar{P}, \Delta)$ which computes a (random) cutset S of $E(H)$ such that:*

1. *For every strongly connected component C of $H \setminus S$, either $\mathsf{w}(\mathcal{P}[C]) < \mathsf{w}(\mathcal{P}[H])$ or the weak diameter of C is at most 12Δ, and*
2. *for every edge e in $E(H)$, $\Pr[e \in S] \leq O(1) \cdot \frac{w(e)}{\Delta}$.*

Then, for each strongly connected component C of $H \setminus S$ such that the weak diameter of C is greater than Δ, the algorithm executes a recursive call Directed-LDD $(G, \mathcal{P}, C, \Delta)$ to compute a random cutset S_C of C. Finally, Directed-LDD returns the union of S and all cutsets S_C returned by the recursive calls. This completes the description of Directed-LDD.

Algorithm 1 Directed-LDD$(G, \mathcal{P}, H, \Delta)$.

---

if $\mathsf{w}(\mathcal{P}[H]) = 0$ **then**
  return $\emptyset$
$S \leftarrow$ Scissors-Carving$(G, \mathcal{P}, H, \bar{P}, \Delta)$
$\mathcal{C} \leftarrow$ Collection of SCCs of $H \setminus S$ whose weak diameter is greater than 12Δ
for all $C \in \mathcal{C}$ **do**
  $S_C \leftarrow$ Directed-LDD$(G, \mathcal{P}, C, \Delta)$
  $S \leftarrow S \cup S_C$
return S

Let $S =$ Directed-LDD$(G, \mathcal{P}, H, \Delta)$.

▶ **Lemma 14.** *The weak diameter of each SCC of $H \setminus S$ is at most 12Δ.*

▶ **Lemma 15.** *For every edge e in H, $\Pr[e \in S] \leq O(\mathsf{w}(\mathcal{P}[H])) \cdot \frac{w(e)}{\Delta}$.*

Finally, Lemma 12 follows from applying Lemmas 14 and 15 by taking $H \leftarrow G$ and $\Delta \leftarrow \frac{\Delta}{12}$.

[7] The algorithm receives $\bar{P}$ as an additional input parameter, where $\bar{P}$ is a sequence of vertices in G satisfying certain conditions. We provide the details of $\bar{P}$ in Section 4.1.

4.1 Reducing path partition-width

In this section, we explain our key subroutine Scissors-Carving. To this end, we define some additional notation that will be used in this section. Recall that $\mathcal{P} = (\mathcal{P}, \mathcal{X})$ consists of m bags $X_1, X_2, \dots X_m$. Let H be a strongly connected subgraph of G. We say an index $i \in [m]$ is a *critical* index of H if $|V(H) \cap X_i| = \mathsf{w}(\mathcal{P}[H])$. Equivalently, we refer to X_i as a *critical bag* of H. Let $\mathcal{C}(H) = \{j_1, j_2, \dots, j_\ell\}$ be the set of critical indices of H such that $1 \le j_1 < j_2 < \dots < j_\ell \le m$.

We say a shortest u-v path P in G is a *critical path* with respect to H if $u \in X_{j_1} \cap H$ and $v \in X_{j_\ell} \cap H$. From the definition of a path partition, P intersects every bag X_t with $j_1 \le t \le j_\ell$. Hence, P intersects every critical bag of H.

▶ **Observation 16.** *For any critical path P with respect to H, $\mathsf{w}(\mathcal{P}[H \setminus P]) < \mathsf{w}(\mathcal{P}[H])$.*

Due to technical reasons, throughout the algorithm, we maintain a certain subsequence of the critical path P, called a *critical sequence*, which is defined as follows. For each critical index j of H, let $v_j \in X_j \cap H$ be the vertex that appears earliest in P along the direction from u to v. We may assume $v = v_{j_\ell}$ as otherwise, we can take the u-v_{j_ℓ} subpath of P. Then, $v_{j_1}, v_{j_2}, \dots, v_{j_\ell}$ are naturally ordered along the path P. We call $\langle v_{j_1}, v_{j_2}, \dots v_{j_\ell} \rangle$ a *critical sequence* of H with respect to P, and denote it by $\bar{P}[H]$. Note that this critical sequence $\bar{P}[H]$ is uniquely defined for a fixed H and P. See Figure 5(a) for an illustration. Since $\bar{P}[H]$ intersects every critical bag of H, we observe the following.

▶ **Observation 17.** *For a critical sequence $\bar{P}[H]$ of H, $\mathsf{w}(\mathcal{P}[H \setminus \bar{P}[H]]) < \mathsf{w}(\mathcal{P}[H])$.*

For a critical sequence $\bar{P} = \langle u, \dots, v \rangle$ of H with respect to P, and a strongly connected subgraph C of H, let x and y be the first and the last vertices of $\bar{P} \cap C$, respectively. We define the *surviving subsequence* of $\bar{P}$ with respect to C, denoted by $\bar{P}[C]$, as the consecutive subsequence of $\bar{P}[H]$ that starts at x and ends at y. Note that while $\bar{P}[C]$ may not be fully contained in C, the two endpoints x, y are contained in C. See Figure 5(b) for an illustration.

We define the *forward disatnce* and the *backward distnace* of C (with respect to P) as $d_G(x, y)$ and $d_G(y, x)$, respectively, and denote them by $f_{\bar{P}}(C)$ and $b_{\bar{P}}(C)$, respectively. Note that we analyze these distances with respect to the shortest paths in the original graph G, rather than the subgraph C. This is sufficient for our purpose since our goal is to bound the *weak diameter* of each SCC. We show that $\bar{P}[C]$ plays the role of a "critical" sequence for C. That is, the removal of $\bar{P}[C]$ from C reduces its path partition-width. Based on the following lemma, in the remainder of this section, we slightly abuse the notation and refer to $\bar{P}[C]$ as a *critical sequence* of C.

▶ **Lemma 18.** *Let C be a strongly connected subgraph of H and let $\bar{P}$ be a critical sequence of H with respect to a critical path P. Then, $\mathsf{w}(\mathcal{P}[C \setminus \bar{P}[C]]) < \mathsf{w}(\mathcal{P}[H])$.*

▶ **Observation 19.** *Let $\bar{P}$ be a critical sequence of H with respect to a critical path P. Then, for every strongly connected subgraph C of H, $\bar{P}[C]$ is a subsequence of a shortest path in G.*

Algorithm. We describe our algorithm Scissors-Carving $(G, \mathcal{P}, H, \bar{P}, \Delta)$. See Algorithm 2 for the pseudocode. It receives as input a directed graph G, a path partition $\mathcal{P}$ of G, a strongly connected subgraph H of G, a critical sequence $\bar{P}$ of H with respect to P, and a parameter $\Delta > 0$. Moreover, it outputs a random cutset S of $E(H)$ such that for every SCC C of $H \setminus S$, either $\mathsf{w}(\mathcal{P}[C]) < \mathsf{w}(\mathcal{P}[H])$ or the weak diameter of C is at most 12Δ.

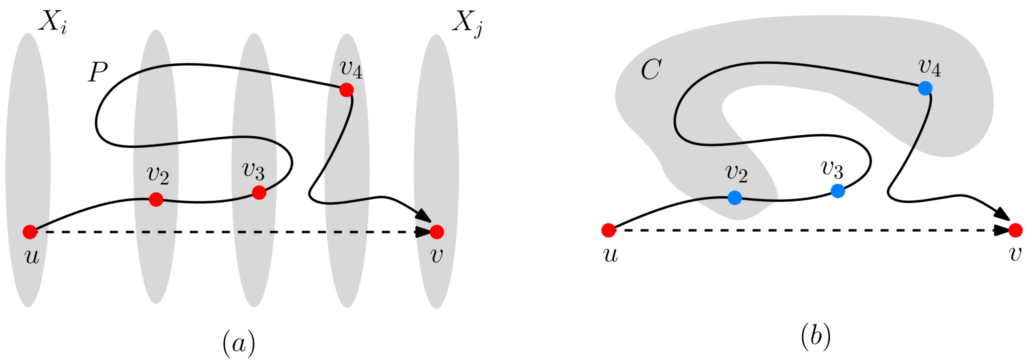

Figure 5 Illustration of the critical path and critical sequence of H. (a) The gray shaded region denotes the critical bags of H. A critical path P connects a vertex u in the first critical bag and a vertex v in the last critical bag. A critical sequence $\bar{P}$ of H with respect to P is $\langle u, v_2, v_3, v_4, v\rangle$. (b) For a subgraph C of H, the surviving subsequence $\bar{P}[C] = \langle v_2, v_3, v_4\rangle$ is depicted as blue points.

The algorithm is a recursive procedure with respect to H and $\bar{P}$. Depending on the forward distance $f_{\bar{P}}(H)$ and the backward distance $b_{\bar{P}}(H)$ of H, the procedure proceeds in one of three stages. The first stage is executed when $f_{\bar{P}}(H) = \tau > \Delta$. At this stage, we sample a random radius R uniformly at random from $[\frac{1}{2}\tau, \frac{2}{3}\tau]$. Then, we compute two balls, an out-ball $B_1 = B^+_G(u, R)$ and an in-ball $B_2 = B^-_G(v, R)$, compute a set S_1 of boundary edges of B_1 and B_2 that intersect H, i.e., $S_1 = H \cap (\delta^+(B_1) \cup \delta^-(B_2))$. Next, we recursively apply the algorithm to each SCC C of $H \setminus S_1$ which is fully contained in $B_1 \cup B_2$. In addition, we pass the surviving subsequence $\bar{P}[C]$ as the critical sequence to the next recursive call.

The second stage is executed when $f_{\bar{P}}(H) \leq \Delta$ and $b_{\bar{P}}(H) = \tau' > 8\Delta$. At this stage, we sample a random radius R uniformly at random from $[\frac{1}{2}\tau', \frac{13}{24}\tau']$. Then, we compute two balls, an in-ball $B_1 = B^-_G(u, R)$ and an out-ball $B_2 = B^+_G(v, R)$, compute a set S_1 of boundary edges of B_1 and B_2 that intersect H, i.e., $S_1 = H \cap (\delta^-(B_1) \cup \delta^+(B_2))$. Next, we recursively apply the algorithm to each SCC of $H \setminus S_1$ which is fully contained in $B_1 \cup B_2$. In addition, we pass the surviving subsequence $\bar{P}[C]$ to the next recursive call.

The third stage is executed when $f_{\bar{P}}(H) \leq \Delta$ and $b_{\bar{P}}(H) \leq 8\Delta$. We sample two random radii R_1, R_2 uniformly at random from $[\Delta, 2\Delta]$ and $[9\Delta, 10\Delta]$, respectively. Then, we compute two balls, an out-ball $B_1 = B^+_G(u, R_1)$ and an in-ball $B_2 = B^-_G(v, R_2)$, return $H \cap (\delta^+(B_1) \cup \delta^-(B_2))$.

Finally, we provide a crucial technical remark: As an extreme case, our algorithm immediately returns the empty set whenever $\bar{P}[C]$ is empty (lines 1-2). Intuitively, due to Lemma 18, $\bar{P}[C] = \emptyset$ implies $\mathsf{w}(\mathcal{P}[C]) < \mathsf{w}(\mathcal{P}[H])$, which satisfies our structural goal.

Algorithm 2 Scissors-Carving $(G, \mathcal{P}, H, \bar{P} = \langle u, \dots, v \rangle, \Delta)$.

if $\bar{P} = \emptyset$ **then**
    return $\emptyset$
$\tau, \tau', S_1, S_2 \leftarrow d_G(u, v), d_G(v, u), \emptyset, \emptyset$
/* **Stage 1. Reducing forward distance** */
if $\tau > \Delta$ **then**
    Sample R uniformly at random in the range $[\frac{1}{2}\tau, \frac{2}{3}\tau]$
    $B_1 \leftarrow B_G^+(u, R)$
    $B_2 \leftarrow B_G^-(v, R)$
    $S_1 \leftarrow H \cap (\delta^+(B_1) \cup \delta^-(B_2))$
    $\mathcal{C} \leftarrow$ SCCs of $H \setminus S_1$ that are fully contained in $B_1 \cup B_2$
    for $C \in \mathcal{C}$ **do**
        $S_2 \leftarrow S_2 \cup$ Scissors-Carving $(G, \mathcal{P}, C, \bar{P}[C], \Delta)$
    return $S_1 \cup S_2$
/* **Stage 2. Reducing backward distance** */
if $\tau \le \Delta$ and $\tau' > 8\Delta$ **then**
    Sample R uniformly at random in the range $[\frac{1}{2}\tau', \frac{13}{24}\tau']$
    $B_1 \leftarrow B_G^-(u, R)$
    $B_2 \leftarrow B_G^+(v, R)$
    $S_1 \leftarrow H \cap (\delta^-(B_1) \cup \delta^+(B_2))$
    $\mathcal{C} \leftarrow$ SCCs of $H \setminus S_1$ that are fully contained in $B_1 \cup B_2$
    for $C \in \mathcal{C}$ **do**
        $S_2 \leftarrow S_2 \cup$ Scissors-Carving $(G, \mathcal{P}, C, \bar{P}[C], \Delta)$
    return $S_1 \cup S_2$
/* **Stage 3. Bounding weak diameter** */
else
    Sample R_1 uniformly at random from $[\Delta, 2\Delta]$, and R_2 uniformly at random from $[9\Delta, 10\Delta]$
    $B_1 \leftarrow B_G^+(u, R_1)$
    $B_2 \leftarrow B_G^-(u, R_2)$
    return $H \cap (\delta^+(B_1) \cup \delta^-(B_2))$

5 Directed Sparsest Cut Problem for Bounded-Treewidth Digraphs

Directed non-bipartite sparsest cut problem. Let G be an edge-weighted directed graph with *capacities* $c(e)$ for all edges $e \in E(G)$. Let T be a set of terminal pairs $T = \{(s_1, t_1), (s_2, t_2), \dots, (s_\ell, t_\ell)\}$ of vertices. For each $i \in [\ell]$, let $\mathsf{dem}(i)$ be a non-negative real number, called the *demand*, for the terminal pair (s_i, t_i). A *cut* S of G is a subset $S \subseteq E(G)$ of edges. For a cut S, let $c(S) := \sum_{e \in S} c(e)$ be the *capacity* of S. Let $I_S \subseteq [\ell]$ be the set of integers i such that every s_i-t_i path (not necessarily a shortest path) contains at least one edge of S. We define the *demand* of S by $\mathsf{dem}(S) := \sum_{i \in I_S} \mathsf{dem}(i)$. The *sparsity* of S is defined by $c(S)/\mathsf{dem}(S)$. The goal of the directed non-bipartite sparsest cut problem is to find a cut S that minimizes the sparsity.

The integrality gap of the LP relaxation for this problem is called the directed multi-commodity flow cut-gap. For non-uniform demands, Hajiaghayi and Räcke showed an $O(\sqrt{n})$ upper bound for the flow-cut gap [27]. This has been improved to $\tilde{O}(n^{11/23})$ by Agarwal, Alon, and Charikar [4]. On the other hand, the current best-known lower bound is due to Chuzhoy and Khanna [14], who showed a $\tilde{\Omega}(n^{1/7})$ lower bound for the flow-cut gap.

Mémoli, Sidiropoulos, and Sridhar [40] showed that every n-vertex directed graph with treewidth tw admits an $(O(\mathsf{tw}\log n), \Delta)$-quasipartition for every parameter $\Delta > 0$. They used their quasipartition construction to show that the flow-cut gap is $O(\mathsf{tw}\log^2 n)$ for such graphs. We now present the main result of this section.

▶ **Theorem 4.** *Let G be an n-vertex directed graph of treewidth tw. The integrality gap (also called the flow-cut gap) of the Directed Non-Bipartite Sparsest-Cut LP relaxation on G is $O(\mathsf{tw}\log n)$.*

This improves upon the $\log n$ factor of the previously best-known bound from [40]. At a high level, we follow the procedure of [40], but provide a better analysis of their approach.

We briefly explain the approach of [40], which proved that the integrality gap is $O(\mathsf{tw}\log^2 n)$. First, they showed that for every $\Delta > 0$, G admits a (β, Δ)-quasipartition with $\beta = O(\mathsf{tw}\log n)$. Using this quasipartition, they proved that the quasimetric space with respect to the shortest path metric of G can be embedded into a convex combination of 0-1 quasimetric spaces (see Definition 21 and Lemma 22) with a distortion of $O(\beta\log n) = O(\mathsf{tw}\log^2 n)$. Finally, by using the known relationship between the convex combination of 0-1 quasimetric spaces and the integrality gap for the directed non-bipartite sparsest cut problem [13], they showed that the integrality gap is $O(\mathsf{tw}\log^2 n)$.

▶ **Definition 20.** *A quasimetric space $M = (X, d)$ is a function d over $X \times X$ that satisfies all but the symmetry axiom for a metric space. In other words, d satisfies the following properties:*

- *$d(u, u) = 0$ for all $u \in X$.*
- *(Triangle inequality) $d(u, v) \le d(u, w) + d(w, v)$ for all $u, v, w \in X$.*

▶ **Definition 21.** *A quasimetric space $M = (X, d)$ is a* 0-1 *quasimetric space if every metric distance $d(u, v)$ is either 0 or 1.*

▶ **Lemma 22** (Lemma 16 of [40])**.** *Suppose a directed graph G admits a (β, Δ)-quasipartition for every $\Delta > 0$. Then there is an embedding of G into a convex combination of* 0-1 *quasimetric spaces with a distortion of $O(\beta\log n)$.*

5.1 Deep dive into the approach of [40]

In this section, we carefully review the previous approach of [40] step-by-step, and then describe our improvement. First, we analyze the $(O(\mathsf{tw}\log n), \Delta))$-quasipartition construction of [40].

The construction is achieved by a recursive approach. Given a tree decomposition $(T, \{B_t\}_{t\in T})$, we choose a central bag B_t, meaning that each subgraph of G induced by the maximal subtree of $T \setminus \{t\}$ contains at most $\frac{2}{3}|V(G)|$ vertices. We then sample a radius R uniformly at random from $[\frac{1}{2}\Delta, \Delta]$. For each vertex $v \in B_t$, we compute two balls, an out-ball $B_1 := B^+_G(v, R)$ and an in-ball $B_2 := B^-_G(v, R)$, add $\delta^+(B_1) \cup \delta^-(B_2)$ to the cutset S, and recursively apply this procedure to each subgraph induced by the connected components of $T \setminus \{t\}$. Note that any x-y path intersecting v such that $d_G(x, y) > 2\Delta$ must intersect $\delta^+(B_1) \cup \delta^-(B_2)$ because, due to the triangle inequality, we have that:

$$\max(d_G(x, v), d_G(v, y)) \ge \frac{1}{2}(d_G(x, v) + d_G(v, y)) \ge \frac{1}{2}d_G(x, y) > \Delta.$$

This implies that for every path (not necessarily a shortest path) connecting two vertices from different trees of $T \setminus \{t\}$, either it intersects S, or it satisfies $d_G(x, y) \le 2\Delta$. This justifies the recursion: after the recursion terminates, we can conclude that for every x-y path in $G \setminus S$, we have $d_G(x, y) \le 2\Delta$.

We briefly analyze the probability that a single edge belongs to the cutset, i.e., $\Pr[e \in S]$. Since B_t plays the role of a separator, each edge e is affected by at most one instance at each level of the recursion. Moreover, since B_t is a central bag, the recursion depth is $O(\log n)$. At a single level, e is affected by at most $2|B_t| = 2(\mathsf{tw}+1)$ balls. Hence, due to the union bound on the probability,

$$\Pr[e \in S] \le O(\log n) \cdot O(\mathsf{tw}) \cdot \frac{w(e)}{\Delta} = O(\mathsf{tw} \log n) \cdot \frac{w(e)}{\Delta}.$$

Next, we analyze the second part which gives an embedding of a quasimetric space into a convex combination of 0-1 quasimetric spaces (Lemma 22). For simplicity, suppose every edge weight of G is a positive integer and $\max_{u,v}\{d_G(u,v)\} = 2^k$ for some integer $k = O(\log n)$. Then, for each scale $i \in \{0, 1, \ldots, k\}$, we compute a $(\beta, 2^i)$-quasipartition $\mathcal{D}_i$ of G using the above procedure. Importantly, each quasipartition $\mathcal{D}_i$ is a distribution over a cutset $S_i \subset E(G)$. Using these, we compute a random quasipartition of G according to the following procedure. Let $c = 2^{k+1} - 1$. First, we select one $\mathcal{D}_i$ among $\{\mathcal{D}_0, \mathcal{D}_1, \ldots, \mathcal{D}_k\}$ with probability $\frac{2^i}{c}$. Then, we pick a random cutset S from the chosen $\mathcal{D}_i$. This yields a distribution $\mathcal{D}$ over the cutsets.

Next, for each cutset S of $\mathcal{D}$, we define a 0-1 quasimetric space $(V(G), d_S)$ where $d_S(x,y) = 0$ if there is an x-y path in $G \setminus S$ and $d_S(x,y) = 1$ otherwise. This yields a distribution over 0-1 quasimetric spaces which can be expressed as a convex combination of 0-1 quasimetric spaces. Let $(V(G), d^*)$ be the resulting quasimetric space. Then, we have

$$\begin{aligned}
d^*(u,v) &= \Pr_{S \sim \mathcal{D}}[\text{there is no } u\text{-}v \text{ path in } G \setminus S] && (1)\\
&= \sum_{i=0}^{k} \frac{2^i}{c} \Pr_{S_i \sim \mathcal{D}_i}[\text{there is no } u\text{-}v \text{ path in } G \setminus S_i] && (2)\\
&\le \sum_{i=0}^{k} \frac{2^i}{c} \beta \cdot \frac{d_G(u,v)}{2^i} && (3)\\
&= \frac{(k+1)\cdot\beta}{c} d_G(u,v). && (4)
\end{aligned}$$

For the other direction, [40] showed that $d^*(u,v) \ge \frac{1}{c} \cdot d_G(u,v)$, yielding a distortion of $(k+1)\beta = O(\beta \log n)$.

5.2 Removing the extra $O(\log n)$ factor from Lemma 22

In this section, we provide a better analysis that improves the distortion by a factor of $O(\log n)$. We focus on the $O(\log n)$ additional factor arising from Lemma 22. This factor comes from combining the distortion contributions across $O(\log n)$ scales, where each scale contributes independently.

Our idea is to avoid this factor by considering all scales simultaneously. To formalize this idea, we revisit the construction in Section 5.1. For a fixed edge $e = (x,y)$ and a scale i, we compute a set $V(e) \subset V(G)$ of size $O(\mathsf{tw} \log n)$ as follows. For each recursive instance that contains e, we add the vertices in the central bag to $V(e)$. Since e is contained in $O(\log n)$ instances (from the construction in Section 5.1), we have $|V(e)| = O(\mathsf{tw} \log n)$. Note that $V(e)$ is independent of the scale i.

Now, we reinterpret the construction in Section 5.1 as follows. For every scale i, we sample a radius R_i uniformly at random from $[2^{i-1}, 2^i]$, and we add e to the cutset S_i if e is contained in one of the boundaries of the (in- and out-) balls of radius $R_i \in [2^{i-1}, 2^i]$ centered at some vertex in $V(e)$. This defines the quasipartition $\mathcal{D}_i$ over the cutsets. We prove the following useful lemma.

▶ **Lemma 23.** *Let $e = (x, y)$ be an edge and z be a vertex in $V(e)$. Then, there are at most $O(1)$ scales $i \in [k]$ satisfying the following conditions:*
(a) $\Pr[e \in \delta^+(B_G^+(z, R_i))] > 0$*, and*
(b) $2^i > w(e)$*.*

Proof. Let i be an index satisfying conditions $(a), (b)$. Note that e is contained in $\delta^+(B_G^+(z, R_i))$ if and only if $d_G(z, x) \le R_i$ and $d_G(z, y) > R_i$. Subsequently, from condition (a), we have that:

$$\Pr_{R_i}[(d_G(z, x) \le R_i) \cap (d_G(z, y) > R_i))] > 0.$$

Since R_i is sampled uniformly at random from $[2^{i-1}, 2^i]$, we have that:

$$d_G(z, x) < 2^i, \text{ and } d_G(z, y) \ge 2^{i-1}.$$

By the triangle inequality and condition (b), we obtain:

$$2^{i-1} \le d_G(z, y) \le d_G(z, x) + d_G(x, y) \le d_G(z, x) + w(e) < 2^i + 2^i = 2^{i+1}.$$

Let j be the fixed integer such that $d_G(z, y) \in [2^j, 2^{j+1}]$. Then, $i \in \{j-1, j, j+1, j+2\}$. We conclude that there are at most 4 scales i satisfying both conditions $(a), (b)$. This completes the proof. ◀

Using exactly the same argument, we can show the following lemma.

▶ **Lemma 24.** *Let $e = (x, y)$ be an edge and z be a vertex in $V(e)$. Then, there are at most $O(1)$ scales $i \in [k]$ satisfying the following conditions:*
(a) $\Pr[e \in \delta^-(B_G^-(z, R_i))] > 0$*, and*
(b) $2^i > w(e)$*.*

For an edge e and a vertex $z \in V(e)$, let $I(z, e)$ be the set of indices i such that $\Pr[e \in \delta^+(B_G^+(z, R_i)) \cup \delta^-(B_G^-(z, R_i))] > 0$ and $2^i > w(e)$. Due to Lemmas 23 and 24, we have that:

$$|I(z, e)| = O(1). \tag{5}$$

Let $i \in I(z, e)$. Since R_i is chosen uniformly at random from $[2^{i-1}, 2^i]$, we have that:

$$\Pr_{R_i}[(d_G(z, x) \le R_i) \cap (d_G(z, y) > R_i))] \le O(1) \cdot \frac{w(e)}{2^i}. \tag{6}$$

Next, we analyze the probability $q := \Pr_{S \sim \mathcal{D}}$[there is no u-v path in $G \backslash S$]. Our argument closely follows that of [40]. Let $\mathcal{P} = \{p_1, p_2, \ldots, p_\ell\}$ be the set of all u-v paths in G. For every $m \in [\ell]$, let X_m be the event that p_m contains at least one edge in S. Then, $q = \Pr[X_1 \wedge X_2 \wedge \ldots \wedge X_\ell] \le \Pr[X_m]$ for all $m \in [\ell]$. Now let us consider a fixed *shortest path* $p_m = \langle u = v_1, v_2, \ldots, v_t = v\rangle \in \mathcal{P}$. For every $j \in [t-1]$, let Y_j be the event that $(v_j, v_{j+1}) \in S$. Then, $\Pr[X_m] = \Pr[Y_1 \vee Y_2 \vee \ldots \vee Y_{t-1}] \le \sum_j \Pr[Y_j]$ due to the union bound of probability. In addition, let $e_j = (v_j, v_{j+1})$. Then,

$$\begin{aligned}
\Pr[Y_j] = \Pr[e_j \in S] &= \sum_{i=0}^{k} \frac{2^i}{c} \cdot \Pr[e_j \in S_i] && (7)\\
&\le \sum_{i=0}^{k} \frac{2^i}{c} \cdot \sum_{z \in V(e_j)} \Pr_{R_i}[(d_G(z, v_j) \le R_i) \cap (d_G(z, v_{j+1}) > R_i))] && (8)\\
&= \sum_{z \in V(e_j)} \sum_{i=0}^{k} \frac{2^i}{c} \cdot \Pr_{R_i}[(d_G(z, v_j) \le R_i) \cap (d_G(z, v_{j+1}) > R_i))]. && (9)
\end{aligned}$$

We analyze the inner summation over i as follows. We split the sum over k scales at $i = \lfloor \log w(e_j) \rfloor$. Since $I(z, e_j) \subseteq \{i : 2^i > w(e_j)\}$, we can bound each part separately. Let $P_{i,z} := \Pr_{R_i}[(d_G(z, v_j) \le R_i) \cap (d_G(z, v_{j+1}) > R_i))]$. Then,

$$\begin{aligned}
&\sum_{i=0}^{k} \frac{2^i}{c} \cdot \Pr_{R_i}[(d_G(z, v_j) \le R_i) \cap (d_G(z, v_{j+1}) > R_i))] = \sum_{i=0}^{k} \frac{2^i}{c} \cdot P_{i,z} \\
&= \sum_{i=0}^{\lfloor \log w(e_j) \rfloor} \frac{2^i}{c} \cdot P_{i,z} + \sum_{i=\lfloor \log w(e_j) \rfloor + 1}^{k} \frac{2^i}{c} \cdot P_{i,z} \\
&\le \sum_{i=0}^{\lfloor \log w(e_j) \rfloor} \frac{2^i}{c} + \sum_{i \in I(z,e_j)} \frac{2^i}{c} \cdot P_{i,z} \\
&\overset{(6)}{\le} \frac{2 \cdot w(e_j)}{c} + \sum_{i \in I(z,e_j)} \frac{2^i}{c} \cdot (O(1) \cdot \frac{w(e_j)}{2^i}) \overset{(5)}{=} O(1) \cdot \frac{w(e_j)}{c}.
\end{aligned}$$

Returning to Equation (9), we obtain:

$$\begin{aligned}
\Pr[Y_j] &\le \sum_{z \in V(e_j)} \sum_{i=0}^{k} \frac{2^i}{c} \cdot \Pr_{R_i}[(d_G(z, v_j) \le R_i) \cap (d_G(z, v_{j+1}) > R_i))]. \\
&\le \sum_{z \in V(e_j)} O(1) \cdot \frac{w(e_j)}{c} = O(|V(e_j)|) \cdot \frac{w(e_j)}{c} = O(\mathsf{tw} \log n) \cdot \frac{w(e_j)}{c}.
\end{aligned}$$

Here, the last equality follows from the fact that $|V(e)| = O(\mathsf{tw} \log n)$ for every edge e. Since p_m is a shortest u-v path in G, we have that:

$$q \le \Pr[X_m] \le \sum_j \Pr[Y_j] \le O(\mathsf{tw} \log n) \cdot \sum_j \frac{w(e_j)}{c} = O(\mathsf{tw} \log n) \cdot \frac{d_G(u, v)}{c}. \tag{10}$$

Putting everything together, we show that the probability bound from Eq.(1) to Eq.(4) can be improved as follows:

$$d^*(u, v) = \Pr_{S \sim \mathcal{D}}[\text{there is no } u\text{-}v \text{ path in } G \setminus S] \overset{(10)}{\le} O(\mathsf{tw} \log n) \cdot \frac{d_G(u, v)}{c}.$$

This confirms the improved distortion guarantee for the algorithm of [40].

▶ **Lemma 25.** *Given a directed graph G having treewidth tw, there is an embedding of G into the convex combination of 0-1 quasimetric space with distortion $O(\mathsf{tw} \log n)$.*

▶ **Theorem 4.** *Let G be an n-vertex directed graph of treewidth tw. The integrality gap (also called the flow-cut gap) of the Directed Non-Bipartite Sparsest-Cut LP relaxation on G is $O(\mathsf{tw} \log n)$.*

References

1. I. Abraham, C. Gavoille, D. Malkhi, and U. Wieder. Strong-diameter decompositions of minor free graphs. *Theory of Computing Systems*, 47(4):837–855, 2010. `doi:10.1007/s00224-010-9283-6`.
2. Ittai Abraham, Arnold Filtser, Anupam Gupta, and Ofer Neiman. Metric embedding via shortest path decompositions. *SIAM J. Comput.*, 51(2):290–314, 2022. a priliminary version apperared in the proceedings of STOC 18. `doi:10.1137/19m1296021`.

3 Ittai Abraham, Cyril Gavoille, Anupam Gupta, Ofer Neiman, and Kunal Talwar. Cops, robbers, and threatening skeletons: Padded decomposition for minor-free graphs. *SIAM J. Comput.*, 48(3):1120–1145, 2019. preliminary version published in STOC 2014. doi:10.1137/17M1112406.

4 Amit Agarwal, Noga Alon, and Moses S Charikar. Improved approximation for directed cut problems. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 671–680, 2007. doi:10.1145/1250790.1250888.

5 Sepehr Assadi, Gary Hoppenworth, and Nicole Wein. Covering approximate shortest paths with dags. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC '25, pages 2269–2280, New York, NY, USA, 2025. Association for Computing Machinery. doi:10.1145/3717823.3718298.

6 Yair Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996*, pages 184–193, 1996. doi:10.1109/SFCS.1996.548477.

7 Piotr Berman, Arnab Bhattacharyya, Konstantin Makarychev, Sofya Raskhodnikova, and Grigory Yaroslavtsev. Improved approximation for the directed spanner problem. In *International Colloquium on Automata, Languages, and Programming*, pages 1–12. Springer, 2011. doi:10.1007/978-3-642-22006-7_1.

8 Aaron Bernstein, Maximilian Probst Gutenberg, and Christian Wulff-Nilsen. Near-optimal decremental SSSP in dense weighted digraphs. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 1112–1122. IEEE, 2020. doi:10.1109/FOCS46700.2020.00107.

9 Aaron Bernstein, Danupon Nanongkai, and Christian Wulff-Nilsen. Negative-weight single-source shortest paths in near-linear time. *Commun. ACM*, 68(2):87–94, 2025. preliminary version published in FOCS 2022. doi:10.1145/3631536.

10 Guy E Blelloch, Anupam Gupta, Ioannis Koutis, Gary L Miller, Richard Peng, and Kanat Tangwongsan. Nearly-linear work parallel sdd solvers, low-diameter decomposition, and low-stretch subgraphs. *Theory of Computing Systems*, 55(3):521, 2014.

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

12 Karl Bringmann, Nick Fischer, Bernhard Haeupler, and Rustam Latypov. Near-optimal directed low-diameter decompositions. In Keren Censor-Hillel, Fabrizio Grandoni, Joël Ouaknine, and Gabriele Puppis, editors, *52nd International Colloquium on Automata, Languages, and Programming, ICALP 2025, July 8-11, 2025, Aarhus, Denmark*, volume 334 of *LIPIcs*, pages 35:1–35:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.ICALP.2025.35.

13 Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Directed metrics and directed graph partitioning problems. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 51–60, 2006. URL: http://dl.acm.org/citation.cfm?id=1109557.1109564.

14 Julia Chuzhoy and Sanjeev Khanna. Polynomial flow-cut gaps and hardness of directed cut problems. *Journal of the ACM (JACM)*, 56(2):1–28, 2009. doi:10.1145/1502793.1502795.

15 Jonathan Conroy and Arnold Filtser. How to protect yourself from threatening skeletons: Optimal padded decompositions for minor-free graphs. In Michal Koucký and Nikhil Bansal, editors, *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025, Prague, Czechia, June 23-27, 2025*, pages 2281–2292. ACM, 2025. doi:10.1145/3717823.3718252.

16 Guoli Ding and Bogdan Oporowski. On tree-partitions of graphs. *Discrete Mathematics*, 149(1-3):45–58, 1996. doi:10.1016/0012-365X(94)00337-I.

17 Matthias Englert, Anupam Gupta, Robert Krauthgamer, Harald Räcke, Inbal Talgam-Cohen, and Kunal Talwar. Vertex sparsifiers: New results from old techniques. *SIAM J. Comput.*, 43(4):1239–1262, 2014. doi:10.1137/130908440.

18 Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. Syst. Sci.*, 69(3):485–497, November 2004. preliminary version published in STOC 2003. `doi:10.1016/j.jcss.2004.04.011`.

19 Jittat Fakcharoenphol and Kunal Talwar. An improved decomposition theorem for graphs excluding a fixed minor. In *RANDOM-APPROX*, pages 36–46, 2003. `doi:10.1007/978-3-540-45198-3_4`.

20 Arnold Filtser. On strong diameter padded decompositions. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2019, September 20-22, 2019, Massachusetts Institute of Technology, Cambridge, MA, USA*, pages 6:1–6:21, 2019. `doi:10.4230/LIPIcs.APPROX-RANDOM.2019.6`.

21 Arnold Filtser. On sparse covers of minor free graphs, low dimensional metric embeddings, and other applications. In Oswin Aichholzer and Haitao Wang, editors, *41st International Symposium on Computational Geometry, SoCG 2025, Kanazawa, Japan, June 23-27, 2025*, LIPIcs, pages 49:1–49:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. `doi:10.4230/LIPIcs.SOCG.2025.49`.

22 Arnold Filtser. Stochastic embedding of digraphs into dags. In Kasper Green Larsen and Barna Saha, editors, *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2026, Vancouver, BC, Canada, January 11-14, 2026*, pages 105–119. SIAM, 2026. `doi:10.1137/1.9781611978971.5`.

23 Arnold Filtser, Tobias Friedrich, Davis Issac, Nikhil Kumar, Hung Le, Nadym Mallek, and Ziena Zeif. Optimal padded decomposition for bounded treewidth graphs. *TheoretiCS*, 4, 2025. `doi:10.46298/THEORETICS.25.22`.

24 Arnold Filtser and Ofer Neiman. Light spanners for high dimensional norms via stochastic decompositions. *Algorithmica*, 84(10):2987–3007, 2022. `doi:10.1007/S00453-022-00994-0`.

25 Fedor V Fomin, Daniel Lokshtanov, Saket Saurabh, Michał Pilipczuk, and Marcin Wrochna. Fully polynomial-time parameterized computations for graphs and matrices of low treewidth. *ACM Transactions on Algorithms (TALG)*, 14(3):1–45, 2018. `doi:10.1145/3186898`.

26 Gregory Gutin, Mark Jones, and Magnus Wahlstrom. The mixed chinese postman problem parameterized by pathwidth and treedepth. *SIAM Journal on Discrete Mathematics*, 30(4):2177–2205, 2016. `doi:10.1137/15M1034337`.

27 Mohammad Taghi Hajiaghayi and Harald Räcke. An $O(n)$-approximation algorithm for directed sparsest cut. *Information Processing Letters*, 97(4):156–160, 2006. `doi:10.1016/J.IPL.2005.10.005`.

28 Sariel Har-Peled, Manor Mendel, and Dániel Oláh. Reliable spanners for metric spaces. *ACM Transactions on Algorithms*, 19(1):1–27, 2023. `doi:10.1145/3563356`.

29 Lior Kamma and Robert Krauthgamer. Metric decompositions of path-separable graphs. *Algorithmica*, 79(3):645–653, 2017. `doi:10.1007/S00453-016-0213-0`.

30 Mamadou M Kanté, Christophe Paul, and Dimitrios M Thilikos. A linear fixed parameter tractable algorithm for connected pathwidth. *SIAM Journal on Discrete Mathematics*, 36(1):411–435, 2022. `doi:10.1137/20M1369361`.

31 Ken-ichi Kawarabayashi and Anastasios Sidiropoulos. Embeddings of planar quasimetrics into directed ℓ_1 and polylogarithmic approximation for directed sparsest-cut. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 480–491. IEEE, 2022.

32 Philip Klein, Serge A. Plotkin, and Satish Rao. Excluded minors, network decomposition, and multicommodity flow. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, STOC '93, pages 682–690, New York, NY, USA, 1993. ACM. `doi:10.1145/167088.167261`.

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

34 R Krauthgamer, JR Lee, M Mendel, and A Naor. Measured descent: a new embedding method for finite metrics. *Geometric & Functional Analysis GAFA*, 15(4):839–858, 2005.

35 Michael Lampis and Manolis Vasilakis. Structural parameterizations for two bounded degree problems revisited. *ACM Transactions on Computation Theory*, 16(3):1–51, 2024. doi:10.1145/3665156.

36 J. R. Lee and A. Naor. Extending lipschitz functions via random metric partitions. *Inventiones Mathematicae*, 160(1):59–95, 2005. doi:10.1007/s00222-004-0400-5.

37 James R Lee and Anastasios Sidiropoulos. On the geometry of graphs with a forbidden minor. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 245–254, 2009. doi:10.1145/1536414.1536450.

38 Frank Thomson Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM*, 46(6):787–832, 1999. doi:10.1145/331524.331526.

39 Jason Li. Simpler and faster directed low-diameter decompositions, 2025. doi:10.48550/arXiv.2505.10244.

40 Facundo Mémoli, Anastasios Sidiropoulos, and Vijay Sridhar. Quasimetric embeddings and their applications. *Algorithmica*, 80(12):3803–3824, 2018. doi:10.1007/S00453-018-0415-8.

41 Satish Rao. Small distortion and volume preserving embeddings for planar and Euclidean metrics. In *Proceedings of the Fifteenth Annual Symposium on Computational Geometry, Miami Beach, Florida, USA, June 13-16, 1999*, pages 300–306, 1999. doi:10.1145/304893.304983.

42 Neil Robertson and Paul D Seymour. Graph minors. I. excluding a forest. *Journal of Combinatorial Theory, Series B*, 35(1):39–61, 1983. doi:10.1016/0095-8956(83)90079-5.

43 Ario Salmasi, Anastasios Sidiropoulos, and Vijay Sridhar. On constant multi-commodity flow-cut gaps for families of directed minor-free graphs. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 535–553. SIAM, 2019. doi:10.1137/1.9781611975482.34.

44 Mikkel Thorup. Compact oracles for reachability and approximate distances in planar digraphs. *Journal of the ACM (JACM)*, 51(6):993–1024, 2004. doi:10.1145/1039488.1039493.