Differentially Private Synthetic Graphs Preserving Triangle-Motif Cuts

Pan Peng PPENG@USTC.EDU.CN
Hangyu Xu HANGYUXU@MAIL.USTC.EDU.CN
School of Computer Science and Technology, University of Science and Technology of China, Hefei, China

Editors: Nika Haghtalab and Ankur Moitra

Abstract

We study the problem of releasing a differentially private (DP) synthetic graph G' that well approximates the triangle-motif sizes of all cuts of any given graph G, where a motif in general refers to a frequently occurring subgraph within complex networks. Non-private versions of such graphs have found applications in diverse fields such as graph clustering, graph sparsification, and social network analysis. Specifically, we present the first (ε, δ)-DP mechanism that, given an input graph G with n vertices, m edges and local sensitivity of triangles $\ell_3(G)$, generates a synthetic graph G' in polynomial time, approximating the triangle-motif sizes of all cuts $(S, V \setminus S)$ of the input graph G up to an additive error of $\tilde{O}(\sqrt{m\ell_3(G)}n/\varepsilon^{3/2})$. Additionally, we provide a lower bound of $\Omega(\sqrt{mn}\ell_3(G)/\varepsilon)$ on the additive error for any DP algorithm that answers the triangle-motif size queries of all (S,T)-cut of G. Finally, our algorithm generalizes to weighted graphs, and our lower bound extends to any K_h-motif cut for any constant $h \geq 2$.

Keywords: Differential privacy, motif cut, synthetic graph, mirror descent

1. Introduction

A graph can take on various forms: a social network, where each edge represents a friendship relation; a financial graph, where each edge represents a transaction relation; or a healthcare network containing patient disease information. In numerous applications, there is a pressing need to disseminate valuable information derived from these graphs while ensuring the protection of individual privacy. Differentially private (DP) algorithms, as pioneered by Dwork et al. (2006), have demonstrated powerful abilities to address such tasks and have garnered significant attention in both academia and industry. Essentially, these DP algorithms ensure that, regardless of the adversary's knowledge about the graph, the privacy of individual users remains intact in the algorithm's output. Recently, there have been numerous studies on DP algorithms for graphs. Typical examples include privately releasing subgraph counting (Chen and Zhou, 2013; Karwa et al., 2011; Zhang et al., 2015; Blocki et al., 2022; Nguyen et al., 2024), degree sequence/distribution (Hay et al., 2009; Karwa and Slavković, 2012; Proserpio et al., 2012), and cut queries (Gupta et al., 2012; Blocki et al., 2012; Upadhyay, 2013; Arora and Upadhyay, 2019; Eliáš et al., 2020; Liu et al., 2024). We refer to the survey (Li et al., 2023) for more examples and applications.

Of particular interest is the exploration of releasing a *synthetic graph* that preserves the *size* of all cuts of the original graph in a differentially private manner (Gupta et al., 2012; Blocki et al., 2012; Upadhyay, 2013; Arora and Upadhyay, 2019; Eliáš et al., 2020; Liu et al., 2024). With such a synthetic graph, analysts can compute answers to queries concerning the cut size information of the graph without acquiring any private information from the input.

The synthetic graphs described above effectively and privately preserve valuable lower-order structures such as the edge connectivity. However, they fail to provide **higher-order structural insights** of complex networks, particularly small network subgraphs (e.g. triangles), which are considered as fundamental building blocks for complex networks (Milo et al., 2002). Specifically, *motifs* – frequently occurring subgraphs – play a crucial role across various domains: (1) *Frequent patterns:* Networks exhibit rich higher-order organizational structures. For example, triangle-motifs play a vital role in social networks as friends of friends tend to become friends themselves (Wasserman, 1994), two-hop paths are crucial for deciphering air traffic patterns (Rosvall et al., 2014) and feed-forward loops and bi-fans are recognized as significant interconnection patterns in various networks (Milo et al., 2002). (2) *Graph clustering:* Works by Benson et al. (2016); Tsourakakis et al. (2017) proposed graph clustering methods based on the concept of motif conductance, which measures the quality of a cluster by leveraging the number of instances of a given motif crossing a cut, rather than merely counting edges. These algorithms typically begin by constructing a motif weighted graph, where each edge is weighted by the number of copies of a given motif it contains, before applying spectral clustering. This approach effectively identifies clusters with high internal motif connectivity and low connectivity between clusters. Furthermore, motifs based embeddings provide a stronger inductive bias by more accurately capture the rich underlying community or cluster structures (Zhang et al., 2018; Nassar et al., 2020). (3) *Motif cut sparsifier:* Kapralov et al. (2022) introduced the notion of a *motif cut sparsifier*, which serves as a sparse weighted graph well approximating the count of motifs crossing each cut within the original graph. Such a motif cut sparsifier stands as a natural extension of the concept of a cut sparsifier (Benczúr and Karger, 1996) and hypergraph cut sparsifier (Chen et al., 2020).

Motivated by the aforementioned considerations on differential privacy in graphs, the widespread application of motifs and the utility of motif cut sparsifiers, we ask how one can efficiently release a synthetic graph that well preserves the motif size of all cuts:

Given a weighted graph $G = (V, E)$ and a motif M, how can we efficiently find another graph $G' = (V, E')$ in a differentially private manner such that for every $S \subset V$, the weight of the M-motif cut $(S, V \setminus S)$ in G is approximated in G' with a small error?

By releasing a synthetic graph that maintains the size of the motif cut, data analysts can query the motif cut each time it is required; existing graph algorithms related to motif cuts can be directly applied to the synthetic graph without compromising privacy. For example, once the synthetic graph G' is generated, motif cut sparsifiers (Kapralov et al., 2022) and higher-order clustering methods (Benson et al., 2016) can be utilized on G' without disclosing sensitive information.

In this paper, we address the above question for the most fundamental motif, i.e., the triangle-motif. We give the first efficient DP algorithms for releasing synthetic graphs that well preserve the triangle-motif weights of all cuts. In addition, we provide lower bounds on the incurred additive error for any such DP algorithms.

1.1. Basic Definitions and Our Contributions

Before we formally state our main result, we first introduce some definitions. Given a graph $G = (V, E, \mathbf{w})$ with weight vector $\mathbf{w} \in \mathbb{R}_+^{\binom{V}{2}}$, and a graph $M = (V_M, E_M)$ which we assume to be a frequently occurring subgraph of G and which is referred to as a *motif*. An instance of motif M is a subgraph of G, which is isomorphic to M, and the *weight of an instance* $I = (V_I, E_I)$ is defined[1] as the product of its edge weights, i.e., $w(I) = \prod_{e \in E_I} w(e)$.

Let S, T be two disjoint subsets of V. The cut (S, T) refers to the set of edges with one endpoint in S and the other endpoint in T. If one of the edges of an instance $I = (V_I, E_I)$ crosses the cut (S, T), we say that I crosses this cut. It is also equivalent to $V_I \cap S \neq \emptyset$ and $V_I \cap T \neq \emptyset$ when the motifs are connected. Then the motif size of the cut (S, T) is defined as the sum of weights of motif instances cross the cut. Formally, we have the definition below.

Definition 1 (Motif size of a cut (S, T)) *Given $G = (V, E, \mathbf{w})$ and a connected motif M, let $\mathcal{M}(G, M)$ be the set of all instances or copies of M in G. Then the M-*motif size *of a cut (S, T) is defined as $\mathrm{Cut}_M^{(G)}(S, T) = \sum_{I \in \mathcal{M}(G,M): I \text{ crosses } (S,T)} w(I)$.*

As mentioned before, we will mainly focus on the triangle-motif, which is one of the most fundamental and well studied motifs (Tsourakakis et al., 2011; Satuluri et al., 2011; Benson et al., 2016; Seshadhri et al., 2020). We will use '$\triangle$' to denote 'M' when referring specifically to triangle-motifs. Now we give the formal definition of differential privacy. Let $\mathcal{D}$ be some domain of datasets.

Definition 2 ((ε, δ)-differential privacy; (Dwork et al., 2006)) *Let $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ be a randomized algorithm or mechanism, where $\mathcal{R}$ is the output domain. For fixed $\varepsilon > 0$ and $\delta \in [0, 1)$, we say that $\mathcal{M}$ preserves (ε, δ)-differential privacy if for any measurable set $S \subset \mathcal{R}$ and any pair of neighboring datasets $x, y \in \mathcal{D}$, it holds that $\Pr[\mathcal{M}(x) \in S] \leq \Pr[\mathcal{M}(y) \in S] \cdot e^{\varepsilon} + \delta$. If $\delta = 0$, we also say $\mathcal{M}$ preserves pure differential privacy (denoted by ε-DP).*

We consider the standard notion of edge privacy and two graphs are called *neighboring* if the two vectors corresponding to their edge weights differ by at most 1 in the ℓ_1 norm (see Definition 9 for the formal definition).

Let $\ell_3(G)$ denotes the local sensitivity of triangle-motif cuts of G. Note that, its local sensitivity is defined as the maximum triangle-motif cut difference between G and its neighboring graphs, equivalently, $\ell_3(G) = \max_{(i,j) \in \binom{V}{2}} \sum_{s \in V \setminus \{i,j\}} \mathbf{w}_{(i,s)} \mathbf{w}_{(j,s)}$.

Our main algorithmic contribution is the following DP algorithm for releasing a synthetic graph for approximating the triangle weights of all cuts of an input graph. We assume that the sum of edge weights is polynomially bounded by n.

Theorem 3 *There exists a polynomial time (ε, δ)-DP algorithm that given an n-vertex weighted graph G with total edge weight W and maximum edge weight $w_{\max}$, outputs a weighted graph G' such that with probability at least $3/4$, for any cut $(S, V \setminus S)$,*

$$\left| \mathrm{Cut}_{\triangle}^{(G)}(S, V \setminus S) - \mathrm{Cut}_{\triangle}^{(G')}(S, V \setminus S) \right| = O(\sqrt{W \cdot \ell_3(G)} \cdot n w_{\max} / \varepsilon^{\frac{3}{2}} \cdot \log^2(n/\delta))).$$

Note that for unweighted graphs with m edges, it holds that $W = m$ and $w_{\max} = 1$. Thus, our algorithm achieves $\tilde{O}(\sqrt{m \cdot \ell_3(G)} \cdot n / \varepsilon^{3/2})$ additive error. Note that for certain classes of graphs G, $\ell_3(G)$ can be relatively small (in comparison to n). For example, if G has maximum degree at

1. As justified in (Kapralov et al., 2022), in integer-weighted graphs, motifs are often viewed as unweighted multigraphs, where each edge is replaced by multiple copies based on its weight. Extending this idea, the motif weight is naturally defined as the product of its edge weights.

most d, then $\ell_3(G) \le d$. (See Section 4 for more discussions of local sensitivity.) Our bound can be compared to the trivial upper bounds $O(m^{3/2}), O(n^3)$ or $O(mn)$ on the triangle-motif size of any cut in a graph. We remark that before our work, there was no known polynomial-time DP algorithm for this problem. We also give a $(\varepsilon, 0)$-DP algorithm that outputs a synthetic graph (in Appendix F) with error $\tilde{O}(n^{\frac{5}{2}})$ based on randomized response, using an analysis similar to (Gupta et al., 2012) for the edge case. Note that the bound $O(\sqrt{m \cdot \ell_3(G)} \cdot n)$ from Theorem 3 is never worse than, and can often be much better than, the $\tilde{O}(n^{\frac{5}{2}})$ bound from randomized response.

Our algorithms are built upon the framework of solving a related convex problem using the mirror descent approach and adding noise appropriately, similar to the approach in (Eliáš et al., 2020). However, unlike the edge case, answering motif cut queries on a graph does *not* align with the extensively studied problem of query release for exponentially sized families of linear queries on a dataset, due to the inherently non-linear nature of motif cut queries. Consequently, we had to resolve several technical challenges (see Section 1.2).

Furthermore, since the convex problem we are using can also be adjusted for other 3-vertex motifs (e.g., wedges), our algorithmic results can be readily extended to these settings as well.

Next, we present a lower bound on the additive error for any DP algorithm that answers triangle-motif size queries for all (S, T)-cuts of G.

Theorem 4 *Let $\mathcal{M}$ be an (ε, δ)-differentially private mechanism, and let G be a graph generated from $G(n, p)$ with $((\log n)/n)^{1/2} \ll p \le \frac{1}{2}$. In this case, G has $m = \Theta(n^2 p)$ edges with high probability. If $\mathcal{M}$ answers the triangle-motif size queries of all (S, T)-cut on G, or on a scaled version of version of G with total edge weight W, up to an additive error α with probability at least $3/4$, then $\alpha \ge \Omega\left(\max\left(\frac{\sqrt{mn}\cdot\ell_3(G)}{\varepsilon}(1-c), \frac{\sqrt{Wn}\cdot\ell_3(G)}{\varepsilon^{\frac{1}{2}}}(1-c)\right)\right)$, where[2] $c = \frac{12(e-1)\delta}{e^{\varepsilon}-1}$.*

We remark that our lower bound also generalizes to all K_h motifs, i.e., the complete graph on h vertices, for any constant $h \ge 2$. For the K_h motifs, our lower bound on the additive error becomes $\Omega\left(\max\left(\frac{\sqrt{mn}\cdot\ell_h(G)}{\varepsilon}(1-c), \frac{\sqrt{Wn}\cdot\ell_h(G)}{\varepsilon^{\frac{1}{2}}}(1-c)\right)\right)$ (while the corresponding requirement for p is $((\log n)/n)^{1/(h-1)} \ll p \le \frac{1}{2}$), where $\ell_h(G)$ is the local sensitivity of K_h-motif cuts of G (see Theorem 6). Note that it also recovers the $\Omega(\sqrt{mn})$ lower bound for the edge motif case (as here $\ell_2(G) = 1$). Our lower bound is based on the connections to discrepancy theory shown by (Muthukrishnan and Nikolov, 2012) and a lower bound on the discrepancy of some coloring function on hypergraphs.

1.2. Our Techniques

Our algorithm is inspired by the work of Eliáš, Kapralov, Kulkarni, and Lee (Eliáš et al., 2020) for privately releasing synthetic graph for edge cut, which we abbreviated as the EKKL approach. We will start by recalling their techniques and then show why the immediate extension of their approach still fails. Finally, we show how we overcome the difficulties and present our algorithm.

The EKKL approach for the edge-motif cut structure The main idea underlying the EKKL approach is as follows. Let $\overline{G}$ be a graph with sum of edge weights at most W. For any fixed

2. Note that c is a very small number as δ is usually set to be smaller than inverse of any polynomial in n.

graph G, the cut difference (or distance), i.e., the maximum difference in weight of some (S,T)-cut in $\overline{G}$ and G, can be bounded by the SDP: $\max_{\mathbf{X}\in\mathcal{D}}\{(\begin{matrix}\mathbf{0} & \mathbf{A}-\overline{\mathbf{A}} \\ \mathbf{A}-\overline{\mathbf{A}} & \mathbf{0}\end{matrix}) \bullet \mathbf{X}\}$, where $\mathbf{A}$ and $\overline{\mathbf{A}}$ denote the weighted adjacency matrices of G and $\overline{G}$, respectively, and the domain $\mathcal{D} = \left\{\mathbf{X}\in\mathbb{R}^{2n} : \mathbf{X} \text{ is symmetric}, \mathbf{X}\succeq \frac{1}{n}\mathbf{I}_{2n}, \text{and } \mathbf{X}_{ii}=1 \text{ for } \forall i\right\}$. Then to find a synthetic graph G that well approximates $\overline{G}$ privately, the algorithm in (Eliáš et al., 2020) tries to solve the following optimization problem: $\min_{\mathbf{w}\in\mathcal{X}'}\max_{\mathbf{X}\in\mathcal{D}}\{(\begin{matrix}\mathbf{0} & \mathbf{A}-\overline{\mathbf{A}} \\ \mathbf{A}-\overline{\mathbf{A}} & \mathbf{0}\end{matrix}) \bullet \mathbf{X} + \lambda\log\det(\mathbf{X})\}$, where $\mathcal{X}' = \{\mathbf{w}\in\mathbb{R}_+^{\binom{V}{2}} : \sum_{e\in\binom{V}{2}}\mathbf{w}_e = W\}$ and W is the total edge weight. Note that in the above, the regularizer $\lambda\log\det\mathbf{X}$ term is added to the original SDP to control the privacy.

Since the objective function is convex with respect to $\mathbf{w}$, the EKKL approach finds a nearly-optimal weight vector $\mathbf{w}$ by stochastic mirror descent (see e.g. (Bubeck, 2015)), which can approximately solve a convex optimization using an iterative process. In each iteration, the mirror descent algorithm requires an approximation to the gradient of the objective function. It further add noise in each iteration by using Johnson-Lindenstrauss transform for privacy guarantee. Then one can derive the tradeoffs between privacy, utility, the parameter λ and the number of iterations, from which we can obtain a (ε,δ)-DP algorithm for the edge cut synthetic graph with small additive error.

Directly applying the EKKL approach to the associated hypergraph or the triangle-motif weighted graph do not seem work due to the *non-linearity* property of triangle cut. That is, the sum of the triangle-motif sizes for cut $(S, V\setminus S)$ of two graphs G_1 and G_2, is not equal to the triangle-motif size for cut $(S, V\setminus S)$ of the sum of the two graphs G_1+G_2, or formally, $\mathrm{Cut}_{\triangle}^{(G_1)}(S,V\setminus S) + \mathrm{Cut}_{\triangle}^{(G_2)}(S,V\setminus S) \neq \mathrm{Cut}_{\triangle}^{(G_1+G_2)}(S,V\setminus S)$. We compare in more details about the difference of the EKKL approach and ours in Appendix C.2 and also discuss some tempting but unsuccessful approaches in Appendix A.3.

Our approach for the triangle-motif cut structure Now we discuss our approach. For a graph $G=(V,E,\mathbf{w})$, we let $\mathbf{A}_{\triangle}$ be the triangle adjacency matrix of G such that $(\mathbf{A}_{\triangle})_{i,j}$ is the sum of weights of triangle instances in which both i and j are involved. Note that $\mathbf{A}_{\triangle}$ is a matrix whose entries depend non-linearly on $\mathbf{w}$. We consider the following optimization problem extended from the one for the edge case:

$$\min_{\mathbf{w}\in\mathcal{X}'}\max_{\mathbf{X}\in\mathcal{D}}\{(\begin{matrix}\mathbf{0} & \mathbf{A}_{\triangle}-\overline{\mathbf{A}}_{\triangle} \\ \mathbf{A}_{\triangle}-\overline{\mathbf{A}}_{\triangle} & \mathbf{0}\end{matrix}) \bullet \mathbf{X} + \lambda\log\det(\mathbf{X})\}, \tag{1}$$

where $\overline{\mathbf{A}}_{\triangle}$ is the triangle adjacency matrix of the input graph $\bar{G}=(V,E,\overline{\mathbf{w}})$ with sum of edge weights at most W. Intuitively, finding a weight vector $\mathbf{w}$ that minimizes the inner SDP of Equation (1) will result in a synthetic graph with a low triangle-motif cut difference between $\mathbf{w}$ and $\overline{\mathbf{w}}$. It is crucial that the objective function is defined in terms of the weight function $\mathbf{w}$ of the target graph G instead of $\mathbf{w}_{\triangle}$, which is the edge weight vector of the triangle-motif weighted graph $G_{\triangle}$.

However, due to the non-linearity property of triangle-motif cut, Equation (1) is no longer convex with respect to $\mathbf{w}$, thus it cannot be solved by convex optimization techniques. Our approach is to add a convexity regularizer[3] $6nw_{\max}\sum_{e\in\binom{V}{2}}(\mathbf{w}_e-\overline{\mathbf{w}}_e)^2$ to the objective function to control the

3. Here, we use the term $6nw_{\max}\sum_{e\in\binom{V}{2}}(\mathbf{w}_e-\overline{\mathbf{w}}_e)^2$ to provide a basic understanding of the main idea; in our actual proof, we incorporate a slightly more complicated term to achieve a smaller error.

convexity, while ensuring that we do not add too much error. Define

$$F_{\triangle}(\mathbf{w},\mathbf{X}) := \begin{pmatrix} \mathbf{0} & \mathbf{A}_{\triangle} - \overline{\mathbf{A}}_{\triangle} \\ \mathbf{A}_{\triangle} - \overline{\mathbf{A}}_{\triangle} & \mathbf{0} \end{pmatrix} \bullet \mathbf{X} + \lambda \log\det(\mathbf{X}) + 6nw_{\max} \sum_{e\in\binom{V}{2}} (\mathbf{w}_e - \overline{\mathbf{w}}_e)^2$$

Then we aim to solve: $\min_{\mathbf{w}\in\mathcal{X}} \max_{\mathbf{X}\in\mathcal{D}} F_{\triangle}(\mathbf{w},\mathbf{X})$, for some appropriately defined domain $\mathcal{X}$.

With the convexity regularizer, we can apply Danskin's theorem (Danskin, 2012) to show that our objective function $f_{\triangle}(\mathbf{w}) := \max_{\mathbf{X}\in\mathcal{D}} F_{\triangle}(\mathbf{w},\mathbf{X})$ is convex. However, solving the corresponding optimization using the stochastic mirror descent becomes more challenging. Specifically, we need to utilize the gradient of $f_{\triangle}(\mathbf{w})$ for updating the mirror descent, which corresponds to finding the minimum of the Bregman divergence associated to some mirror map function (see Appendix B.4), and bounding the final additive error. Unlike the edge case, where the gradient is a constant, the gradient of $f_{\triangle}(\mathbf{w})$ is a relatively complex function that depends on $\mathbf{w}$ (see Lemma 30). This dependency arises from the higher-order nature of the triangle-motif and introduces additional technical difficulties.

Our solution is to impose additional constraints $\{\mathbf{w}_e \leq w_{\max}, e \in \binom{V}{2}\}$ on $\mathcal{X}'$ to obtain a more restricted domain $\mathcal{X}$, and then introduce a new mirror descent update rule based on a greedy method. Specifically, we transform the problem of updating $\mathbf{w}$, i.e., finding the minimizer of the Bregman divergence, into another convex optimization problem (see Equation (6)). Utilizing the Karush-Kuhn-Tucker (KKT) conditions (Ghojogh et al., 2021), we derive the necessary and sufficient conditions that the optimal solution of this optimization problem must satisfy. Based on these conditions, we develop an efficient algorithm (Algorithm 2) to update the weight vector $\mathbf{w}$ using a greedy method.

Once the mirror descent step is defined and an updated solution $\mathbf{w}$ is obtained, we apply the Johnson-Lindenstrauss transform to privately release $\mathbf{X}$. This allows us to analyze the trade-offs between privacy, utility, the parameter λ, and the number of iterations, as in the edge case. Based on these trade-offs, we derive a (ε, δ)-DP algorithm for generating a triangle-motif cut synthetic graph with a small additive error.

Overview of lower bound Our lower bound is established within the discrepancy framework introduced by Muthukrishnan and Nikolov (2012). This framework was also utilized in proving the lower bound for DP algorithms concerning edge motif cut in (Eliáš et al., 2020). Essentially, if there exists a DP algorithm $\mathcal{M}$ for the motif cut problem with additive error smaller than the discrepancy of the motif cut size function over certain classes of graphs, then algorithm $\mathcal{M}$ can be exploited to approximately recover the input, thereby compromising privacy and resulting in a contradiction. To leverage this framework effectively, we need to show that the discrepancy of this function over some class of graphs is relatively large to exclude DP algorithms with small additive error. Specifically, to derive the lower bound for the K_h-motif cut for any complete graph K_h with h vertices, we employ the discrepancy of 3-colorings (i.e. each motif is colored with $+1, -1$ or 0) of h-uniform hypergraphs, which is a generalization of the corresponding discrepancy of graphs. In particular, we identify a set of properties (e.g., each vertex has roughly the same degree and each edge belongs to roughly the same number of K_h instances) sufficient to ensure that: 1) a graph from $G(n, p)$ will satisfy with high probability, and 2) the corresponding h-uniform hypergraph (defined by treating any subset of h vertices as a hyperedge) exhibits large discrepancy.

2. The Algorithm

For the sake of notation convenience, we let $\hat{G} = (V, \hat{E}, \hat{\mathbf{w}}_e)$ denote the input graph, and let G with edge vector $\mathbf{w}$ denote the output graph of our algorithm. Our algorithm outputs G (ε, δ)-differentially privately, and guarantees that each triangle-motif cut of $\hat{G}$ and G will be close.

Preprocessing We use W and $\mathbf{u}$ to denote the differentially privately released approximations of the sum of edge weights and the upper bound of each edge weight of $\hat{G}$ respectively. Then we reweigh the graph $\hat{G}$ to obtain $\overline{G}$ with adjacency matrix $\overline{\mathbf{A}}$, where the sum of edge weights is W. Specifically, we require that $W \geq \hat{W}$ and $\mathbf{u} \geq \overline{\mathbf{w}}$ with high probability. In the non-degenerate case when $\hat{W}$ and $\ell_3(\overline{G})$ are moderately large, we can guarantee that $W = \Theta(\hat{W})$ and $\tilde{\ell}_3(\hat{G}) = \Theta(\ell_3(\hat{G})) = \Theta(\ell_3(\overline{G}))$. In the following, we treat $\overline{G}$ as the input graph with public W and $\mathbf{u}$. More details are deferred to Appendix C.1.

Recall that $\mathbf{A}_\triangle$ is the adjacency matrix of the triangle-motif weighted graph of G. We let $\mathbf{D}_\triangle^{(e)(t)}$ denote the derivative of $\mathbf{A}_\triangle^{(t)}$ at $\mathbf{w}_e^{(t)}$, which is defined in Fact 19. For more details, refer to Appendix B.2.

For more details, refer to Appendix C.1 and Appendix B.2.

An optimization problem Our algorithm will be based on the following optimization problem, which in turns is based on Appendix D.1 that relates the triangle-motif cut difference and the following SDP form approximation.

$$F_\triangle(\mathbf{w}, \mathbf{X}) = \begin{pmatrix} \mathbf{0} & \mathbf{A}_\triangle - \overline{\mathbf{A}}_\triangle \\ \mathbf{A}_\triangle - \overline{\mathbf{A}}_\triangle & \mathbf{0} \end{pmatrix} \bullet \mathbf{X} + \lambda \log\det \mathbf{X}$$
$$+ \sum_{(i,j)\in\binom{V}{2}} 3(\mathbf{w}_{(i,j)} - \overline{\mathbf{w}}_{(i,j)})^2 \cdot \sum_{s\in V\setminus\{i,j\}} (\mathbf{u}_{(i,s)} + \mathbf{u}_{(j,s)}) \tag{2}$$

$$f_\triangle(\mathbf{w}) = \max_{\mathbf{X}\in\mathcal{D}} F_\triangle(\mathbf{w}, \mathbf{X}), \tag{3}$$

where the domains of $\mathbf{w}$ and $\mathbf{X}$ are defined as follows:

$$\mathcal{X} = \{\mathbf{w} \in \mathbb{R}_+^{\binom{V}{2}} : \sum_{e\in\binom{V}{2}} \mathbf{w}_e = W, \mathbf{w}_e \leq \mathbf{u}_e \text{ for } \forall e\} \tag{4}$$

$$\mathcal{D} = \left\{\mathbf{X} \in \mathbb{R}^{2n} : \mathbf{X} \text{ is symmetric}, \mathbf{X} \succeq \frac{1}{n}\mathbf{I}_{2n}, \text{and } \mathbf{X}_{ii} = 1 \text{ for } \forall i\right\} \tag{5}$$

Our algorithm Now, we state our algorithm (Algorithm 1), which iteratively and privately solves the above optimization problem and outputs a graph G that approximates each triangle-motif cut of $\hat{G}$. Specifically, our algorithm is an instantiation of the stochastic mirror descent algorithm (Algorithm 3), that provides an iterative approach to solve an optimization problem defined by a convex function f. In each iteration, the algorithm computes an unbiased estimator g of the gradient ∇f, and then updates the solution based on g and the mirror function Φ.

We invoke Algorithm 3 with $f(x) = f_\triangle(\mathbf{w})$, $\Phi(x) = \sum_{e\in\binom{V}{2}} \mathbf{w}_e \log(\mathbf{w}_e)$, and mirror update step from Algorithm 2. Additionally, we set $\|\cdot\|$ as l_1 norm, hence $\|\cdot\|_*$ is l_∞ norm.

Denote $N(\mathbf{x}, \mathbf{\Sigma})$ as the multivariate normal distribution with mean $\mathbf{x}$ and covariance matrix Σ. Additionally, $\tilde{\ell}_3(\hat{G})$, $U_\triangle$, U_Λ are some quantities defined in Appendix C.1.

Algorithm 1 Private triangle-motif cut Approximation

Input: Graph $\overline{G}$ with adjacency matrix $\overline{\mathbf{A}}$, where $\overline{G}$ is the graph after preprocessing $\hat{G}$, step length η.

Output: Privately release graph G, s.t. G approximates each triangle-motif cut of $\hat{G}$.

1: Set $T = \Theta(\frac{W\cdot(\varepsilon U_\triangle + U_\Lambda)}{n\log(n/\delta)\tilde{\ell}_3(\hat{G})})$,$L = \log_3(\frac{3}{\beta})$, $\lambda = \Theta(\varepsilon^{-1})\tilde{\ell}_3(\hat{G})\sqrt{T}\log^{\frac{3}{2}}(\frac{T}{\delta})\log(\frac{3}{\beta})$, $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = \frac{\varepsilon}{6}$, $\varepsilon_4 = \frac{\varepsilon}{6L}$

2: **for** $\ell = 1, \ldots, L$ **do**

3: Choose $\mathbf{w}^{(1)}$ such that $\mathbf{w}_e^{(1)} = \frac{W}{\binom{n}{2}}$, for all $e \in \binom{V}{2}$

4: Choose random variables $\nu_e \sim \mathrm{Lap}(\frac{1}{\varepsilon_4})$ and release $\tilde{\mathbf{w}}_e = \overline{\mathbf{w}}_e + \nu_e$

5: **for** $t = 1, \ldots, T$ **do**

6: Find the maximizer $\mathbf{X}^{(t)} = \arg\max_{\mathbf{X}\in\mathcal{D}} F_\triangle(\mathbf{w}^{(t)}, \mathbf{X})$, where $F_\triangle$ is defined in Equation (2)

7: Choose a random vector $\zeta \sim N(\mathbf{0}, \mathbf{I}_{2n})$ and release $(\mathbf{X}^{(t)})^{\frac{1}{2}}\zeta$

8: Compute the approximate gradient for all $e = (i,j) \in \binom{V}{2}$:

$$\mathbf{g}_e^{(t)} = \left(\mathbf{X}^{(t)})^{\frac{1}{2}}\zeta\zeta^\top(\mathbf{X}^{(t)})^{\frac{1}{2}}\right)\bullet\begin{pmatrix} \mathbf{0} & \mathbf{D}_\triangle^{(e)(t)} \\ \mathbf{D}_\triangle^{(e)(t)} & \mathbf{0} \end{pmatrix} + 6\sum_{s\in V\setminus\{i,j\}}(\mathbf{u}_{(i,s)} + \mathbf{u}_{(j,s)})(\mathbf{w}_e^{(t)} - \tilde{\mathbf{w}}_e)$$

9: Mirror Descent Step: $\mathbf{w}^{(t+1)} = \mathrm{MD\_Update}(\mathbf{w}^{(t)}, \mathbf{g}^{(t)}, W, \mathbf{u}, \eta)$

10: **end for**

11: Let $\mathbf{w}_\ell = \frac{1}{T}\sum_{i=1}^{T}\mathbf{w}^{(t)}$

12: **end for**

13: **return** $\mathbf{w} = \arg\min(f_\triangle(\mathbf{w}_1), \ldots, f_\triangle(\mathbf{w}_L))$

Mirror descent update In Line 9 of Algorithm 1, we use the algorithm $\mathrm{MD\_Update}$ to update our solution. We now describe this mirror descent update step. Given a weight vector $\mathbf{w}^{(t)}$ for some $t \geq 1$, the estimated gradient $\mathbf{g}^{(t)}$, and $W, \mathbf{u}, \eta$, we update $\mathbf{w}^{(t)}$ to $\mathbf{w}^{(t+1)}$ using Algorithm 2.

The above algorithm corresponds to the stochastic mirror descent framework from Algorithm 3 by setting $f(x) = f_\triangle(\mathbf{w})$ which is defined in Equation (3), and $\Phi(x) = \Phi(\mathbf{w}) = \sum_{e\in\binom{V}{2}}\mathbf{w}_e\log(\mathbf{w}_e)$.

3. Analysis of the Algorithm: Proof Sketch of Theorem 3

In this section, we provide a proof sketch for Theorem 3. The proof proceeds as follows.

(1) SDP Approximation We first verify the fact that $\mathrm{Cut}_\triangle^{(G)}(S, V\setminus S) = \frac{1}{2}\mathbf{1_S}^\top\mathbf{A}_\triangle\mathbf{1}_{\overline{S}}$. That is, the triangle-motif cut of G is exactly half of the cut of the triangle-motif weighted graph of G. Then by a similar approach in (Eliáš et al., 2020), we provide a SDP approximation of the cut difference of the triangle-motif weighted graph. It follows that

$$\max_{\mathbf{X}}\left\{\begin{pmatrix} \mathbf{0} & (\mathbf{A}_2)_\triangle - (\mathbf{A}_1)_\triangle \\ (\mathbf{A}_2)_\triangle - (\mathbf{A}_1)_\triangle & \mathbf{0} \end{pmatrix}\bullet\mathbf{X}\right\}$$

Algorithm 2 MD_Update

Input: $\mathbf{w}^{(t)}, \mathbf{g}^{(t)}, W, \mathbf{u}, \eta$.
Output: $\mathbf{w}^{(t+1)}$.

1: Set $\mathbf{y}_e^{(t+1)} = \mathbf{w}_e^{(t)} \exp(-\eta \mathbf{g}_e^{(t)})$ for $\forall e \in \binom{V}{2}$ and let $N = \binom{n}{2}$
2: Sort edges in non-increasing order so that $\frac{\mathbf{y}_{e_1}^{(t+1)}}{\mathbf{u}_{e_1}} \geq \frac{\mathbf{y}_{e_2}^{(t+1)}}{\mathbf{u}_{e_2}} \geq \cdots \geq \frac{\mathbf{y}_{e_N}^{(t+1)}}{\mathbf{u}_{e_N}}$
3: Compute $S_i = \sum_{j=i}^{N} \mathbf{y}_{e_j}^{(t+1)}$ for $i = 1, \ldots, N$
4: Let $W_1 = W$
5: **for** $i = 1, \ldots, N$ **do**
6: $\quad \mathbf{w}_{e_i}^{(t+1)} = \min(\frac{W_i \cdot \mathbf{y}_{e_i}^{(t+1)}}{S_i}, \mathbf{u}_{e_i})$
7: $\quad$ Set $W_{i+1} = W_i - \mathbf{w}_{e_i}^{(t+1)}$
8: **end for**
9: **return** $\mathbf{w}^{(t+1)}$

is a SDP approximation of triangle-motif cut difference within constant factor, where $\mathbf{X}$ is symmetric, semi-definite and $\mathbf{X}_{i,i} = 1$ for $\forall i$. More details are deferred to Appendix D.1.

(2) Gradient and Convexity We first compute $\nabla_{\mathbf{w}} F_{\triangle}(\mathbf{w}, \mathbf{X})$ and $\nabla^2_{\mathbf{w}} F_{\triangle}(\mathbf{w}, \mathbf{X})$. Then we show $\nabla^2_{\mathbf{w}} F_{\triangle}(\mathbf{w}, \mathbf{X})$ is a diagonally dominant matrix, from which we can prove that $\nabla^2_{\mathbf{w}} F_{\triangle}(\mathbf{w}, \mathbf{X})$ is a semi-definite matrix. It follows that $F_{\triangle}(\mathbf{w}, \mathbf{X})$ is a convex function. By applying Danskin's theorem (Theorem 31), our objective function $f_{\triangle}(\mathbf{w})$ is a convex function. Additionally, we compute $\nabla f_{\triangle}(\mathbf{w}^{(t)})$ by Danskin's theorem, and verify that $\mathbf{g}_e^{(t)}$ is its unbiased approximation. More details are deferred to Appendix D.2.

(3) Mirror Descent Update To prove the correctness of our algorithm, we need to ensure that the update method Algorithm 2 correctly implements the mirror descent framework in our setting. Note that our domain of $\mathbf{w}$ is more restricted than the one in (Eliáš et al., 2020), with additional constraints

$$\{\mathbf{w}_e \leq w_{\max}, \forall e \in \binom{V}{2}\}.$$

This implies that the update method from (Eliáš et al., 2020) does not work here, as the updated solution may lie outside the original domain $\mathcal{X}'$. As a result, we introduce a new mirror descent update algorithm (Algorithm 2) to update our solution. We provide more details in Section 3.1.

(4) Privacy Guarantee First, we use the Laplace Mechanism (Lemma 14) to show that our preprocessing steps and the release of ν in Algorithm 1 are $(\varepsilon_1', 0)$-differentially private. Then, since we use the Johnson-Lindenstrauss transform to privately release $\mathbf{X}$ (as in the edge case), we can demonstrate that the privacy loss in each iteration is bounded by $O(\frac{\ell_3(G)}{\lambda})$ by a similar analysis to (Eliáš et al., 2020). As a result, after approximately $T \approx \frac{\varepsilon^2 \lambda^2}{\ell_3(G)^2}$ iterations, we achieve (ε_2', δ)-differential privacy if we choose

$$\lambda \approx \Theta(\varepsilon^{-1}) \cdot \ell_3(G)\sqrt{T}.$$

To sum up, by Composition Theorem (Lemma 12) and choosing proper values of ε_1', ε_2', our algorithm is proved (ε, δ)-differentially private. For more details, we refer to Appendix D.4.

(5) Utility Guarantee We ensure that the introduction of the convexity term does not significantly increase the error. Assuming the output of the algorithm is $\mathbf{w}$. We can first upper bound

$f_{\triangle}(\mathbf{w})$ by $O(\frac{Wnw_{\max}^2}{\varepsilon^2\sqrt{T}})$. As the convexity regularizer is semi-positive and the privacy regularizer satisfies $|\lambda \log \det \mathbf{X}| \leq O(\lambda n \log n)$, we can lower bound $f_{\triangle}(\mathbf{w})$ as

$$\Theta(\max |\text{Cut}_{\triangle}(\mathbf{w}) - \text{Cut}_{\triangle}(\overline{\mathbf{w}})|) - \Omega(\lambda n \log n).$$

Since we can show that $\max |\text{Cut}_{\triangle}(\mathbf{w}) - \text{Cut}_{\triangle}(\overline{\mathbf{w}})| \leq O(\ell_3(G)\frac{\log(1/\beta)}{\varepsilon})$, the resulting additive error of the algorithm is approximately

$$O(\frac{Wnw_{\max}^2}{\varepsilon^2\sqrt{T}} + \lambda n \log n + \ell_3(G)).$$

Note that we choose $\lambda \approx \Theta(\varepsilon^{-1}) \cdot \ell_3(G)\sqrt{T}$. This implies if we choose $T \approx \frac{Ww_{\max}^2}{\varepsilon \ell_3(G)}$, the algorithm achieves differential privacy with a total additive error of approximately $\tilde{O}(\sqrt{W \cdot \ell_3(G)} \cdot nw_{\max}/\varepsilon^{3/2})$. More details is given in Appendix D.5.

(6) Running time The analysis of running time is mainly based on the algorithm of Lee et al. (2015), which can find an approximate solution of the SDP in Line 6 of Algorithm 1 in time $\tilde{O}(n^6)$. More details are deferred to Appendix D.6.

3.1. Mirror Descent Update

We now show that the update method Algorithm 2 correctly implements the mirror descent framework in our setting. To do so, we note that the mirror function in our algorithm is $\Phi(\mathbf{w}) = \sum_{e\in\binom{V}{2}} \mathbf{w}_e \log(\mathbf{w}_e)$, and thus it suffices to prove that the output of Algorithm 2 is indeed the minimizer of $D_\Phi(\mathbf{w}, \mathbf{y}^{(t+1)})$, where $\mathbf{y}^{(t+1)}$ is the vector satisfying the Line 1 in Algorithm 2, i.e., $\mathbf{y}_e^{(t+1)} = \mathbf{w}_e^{(t)} \exp(-\eta \mathbf{g}_e^{(t)})$ for $\forall e \in \binom{V}{2}$.

Theorem 5 *Denote the output of Algorithm 2 as $\mathbf{w}^{(t+1)}$. Then it holds that $D_\Phi(\mathbf{w}^{(t+1)}, \mathbf{y}^{(t+1)}) = \min_{\mathbf{w}\in\mathcal{X}} D_\Phi(\mathbf{w}, \mathbf{y}^{(t+1)})$, where*

$$D_\Phi(\mathbf{w}, \mathbf{y}^{(t+1)}) = \Phi(\mathbf{w}) - \Phi(\mathbf{y}^{(t+1)}) - (\nabla\Phi(\mathbf{y}^{(t+1)}))^\top(\mathbf{w} - \mathbf{y}^{(t+1)}),$$

and $\mathbf{y}^{(t+1)} \in \mathbb{R}^{\binom{V}{2}}$ s.t. $\nabla\Phi(\mathbf{y}^{(t+1)}) = \nabla\Phi(\mathbf{w}^{(t)}) - \eta \mathbf{g}^{(t)}$.

Note that, $\mathbf{y}^{(t+1)}$ intuitively is the update towards the direction of gradient $\mathbf{g}_e^{(t)}$ without any domain constraints, and $D_\Phi(\mathbf{w}, \mathbf{y}^{(t+1)})$ represents the difference between $\mathbf{w}$ and $\mathbf{y}^{(t+1)}$ to some extent. Namely, we are trying to find the most similar solution $\mathbf{w}^{(t+1)}$ in $\mathcal{X}$ to $\mathbf{y}$.

Proof [Partial Proof of Theorem 5] We first show that $\min_{\mathbf{w}\in\mathcal{X}} D_\Phi(\mathbf{w}, \mathbf{y}^{(t+1)})$ is a convex optimization, which satisfies the Slater's condition (Lemma 25). This will imply the KKT conditions are the necessary and sufficient condition for its optimal solution (by Lemma 25).

Since $\Phi(\mathbf{w}) = \sum_{e\in\binom{V}{2}} \mathbf{w}_e \log(\mathbf{w}_e)$, we have $\nabla\Phi(\mathbf{y}^{(t+1)}) = \nabla\Phi(\mathbf{w}^{(t)}) - \eta \mathbf{g}^{(t)}$, that is $1 + \log(\mathbf{y}_e^{(t+1)}) = 1 + \log(\mathbf{w}_e^{(t)}) - \eta \mathbf{g}_e^{(t)}$. Therefore, we have $\mathbf{y}_e^{(t+1)} = \mathbf{w}_e^{(t)} \exp(-\eta \mathbf{g}_e^{(t)})$ for any $e \in \binom{V}{2}$. Then, note that

$$\begin{aligned} D_\Phi(\mathbf{w}, \mathbf{y}^{(t+1)}) &= \Phi(\mathbf{w}) - \Phi(\mathbf{y}^{(t+1)}) - \nabla\Phi(\mathbf{y}^{(t+1)})^\top(\mathbf{w} - \mathbf{y}^{(t+1)}) \\ &= \sum_{e\in\binom{V}{2}} (\mathbf{w}_e \log(\mathbf{w}_e) - \mathbf{w}_e(1 + \log(\mathbf{y}_e^{(t+1)}))) + R(\mathbf{y}^{(t+1)}). \end{aligned}$$

where $R(\mathbf{y}^{(t+1)})$ is some remaining term only depends on $\mathbf{y}^{(t+1)}$. Thus, we only need to consider the following optimization problem.

$$\min_{\mathbf{w}} D_\Phi(\mathbf{w}, \mathbf{y}_e^{(t+1)}) = \sum_{e \in \binom{V}{2}} \left(\mathbf{w}_e \log(\mathbf{w}_e) - \mathbf{w}_e \left(1 + \log \left(\mathbf{y}_e^{(t+1)} \right) \right) \right) + R(\mathbf{y}^{(t+1)}) \tag{6}$$

$$\text{s.t.} \sum_{e \in \binom{V}{2}} \mathbf{w}_e = W \quad \text{and} \quad \mathbf{w}_e \leq \mathbf{u}_e, \forall e \tag{7}$$

Recall $\mathcal{X} = \{\mathbf{w} \in \mathbb{R}_+^{\binom{V}{2}} : \sum_{e \in \binom{V}{2}} \mathbf{w}_e = W, \mathbf{w}_e \leq \mathbf{u}_e \text{ for } \forall e\}$. Because $\mathbf{u}_e = \overline{\mathbf{w}}_e + \mathrm{Lap}(3/\varepsilon) + 3\log(3n^2/\beta)/\varepsilon + \frac{W}{\binom{n}{2}} > \frac{W}{\binom{n}{2}}$ for any $e \in \binom{V}{2}$ with high probability, $\mathbf{w}$ is an inner point of $\mathcal{X}$ when $\mathbf{w}_e = \frac{W}{\binom{n}{2}}, \forall e \in \binom{V}{2}$, i.e., making the inequality constraints in $\mathcal{X}$ strictly feasible. It implies that the optimization satisfies Slater's condition. Besides, it can be verified that $D_\Phi(\mathbf{w}, \mathbf{y}^{(t+1)})$ is a convex function with respect to w. Thus by Lemma 25, the following conditions are the necessary and sufficient condition for the optimal solution:

$$\begin{cases} \nabla_{\mathbf{w}} D_\Phi(\mathbf{w}, \mathbf{y}^{(t+1)}) + \sum_{e \in \binom{V}{2}} \lambda_e \nabla_{\mathbf{w}}(\mathbf{w}_e - \mathbf{u}_e) + \mu \nabla_{\mathbf{w}}(\sum_{e \in \binom{V}{2}} \mathbf{w}_e - W) = 0 & (8) \\ \lambda_e(\mathbf{w}_e - \mathbf{u}_e) = 0, \forall e & (9) \\ \sum_{e \in \binom{V}{2}} \mathbf{w}_e = W & (10) \\ \mathbf{w}_e \leq \mathbf{u}_e, \forall e & (11) \\ \lambda_e \geq 0, \forall e & (12) \end{cases}$$

where Equation (8) is equivalent to $1 + \log(\mathbf{w}_e) - 1 - \log(\mathbf{y}_e^{(t+1)}) + \lambda_e + \mu = 0$. Therefore we have,

$$\mathbf{w}_e = \mathbf{y}_e^{(t+1)} \exp(-\lambda_e - \mu) \tag{13}$$

Denote $S = \sum_{e \in \binom{V}{2}} \mathbf{y}_e^{(t+1)}$. Assume $\mathbf{w}^* \in \arg\min_{\mathbf{w} \in \mathcal{X}} \min_{\mathbf{w} \in \mathcal{X}} D_\Phi(\mathbf{w}, \mathbf{y}^{(t+1)})$ is a solution with k coordinates $\mathcal{I} = \{i_1, \ldots, i_k\}$ satisfying the equality condition of Equation (11), i.e., $\mathbf{w}_{e_i} = \mathbf{u}_{e_i}$, for any $i \in \mathcal{I}$. We will consider two cases: $k = 0$ and $k \neq 0$. We defer the case that $k = 0$ to Appendix D.3.

In the following we consider the case that $k \neq 0$. In this case, let $\mathcal{I} = \{i_1, \ldots, i_k\}$ denote the index set of the k coordinates such that $\mathbf{w}_{e_i}^* = \mathbf{u}_{e_i}$, for any $i \in \mathcal{I}$, and let $\mathcal{J} = \{j_1, \ldots, j_{\binom{n}{2}-k}\}$ deonte the set of remaining indices.

Then by $\mathbf{w}_{e_j}^* \neq \mathbf{u}_{e_j}$ for any $j \in \mathcal{J}$ and Equation (9), $\lambda_j = 0$ for any $j \in \mathcal{J}$. Thus, $\mathbf{w}_{e_j}^* = \mathbf{y}_{e_j}^{(t+1)} \exp(-\mu)$, for any $j \in \mathcal{J}$. Note that $W = \sum_{\ell=1}^{N} \mathbf{w}_{e_\ell}^* = \sum_{i \in \mathcal{I}} \mathbf{u}_{e_i} + \sum_{j \in \mathcal{J}} \mathbf{y}_{e_j}^{(t+1)} \exp(-\mu)$.

Then by Equation (10), Equation (11) and Equation (13), we have $\mathbf{y}_{e_j}^{(t+1)} e^{-\mu} < \mathbf{u}_{e_j}$, for any $j \in \mathcal{J}$, and $e^{-\mu} = \frac{W - \sum_{i \in \mathcal{I}} \mathbf{u}_{e_i}}{S - \sum_{i \in \mathcal{I}} \mathbf{y}_{e_i}^{(t+1)}}$. Then by Equation (12) and Equation (13), we have $\mathbf{u}_{e_i} = \mathbf{w}_{e_i}^* \leq \mathbf{y}_{e_i}^{(t+1)} e^{-\mu}$ for any $i \in \mathcal{I}$. That is, $\frac{\mathbf{y}_{e_j}^{(t+1)}}{\mathbf{u}_{e_j}} < e^{\mu} \leq \frac{\mathbf{y}_{e_i}^{(t+1)}}{\mathbf{u}_{e_i}}$, for any $i \in \mathcal{I}$ and $j \in \mathcal{J}$. Thus, all the edges with indices in $\mathcal{I}$ appear before those with indices in in $\mathcal{J}$ according to the non-decreasing order in Algorithm 2. That is,

$$\mathcal{I} = \{1,\dots,k\}, \mathcal{J} = \{k{+}1,\dots,N\}, \text{ and } \frac{\mathbf{y}_{e_1}^{(t+1)}}{\mathbf{u}_{e_1}} \geq \dots \geq \frac{\mathbf{y}_{e_k}^{(t+1)}}{\mathbf{u}_{e_k}} \geq \exp(\mu) > \frac{\mathbf{y}_{e_{k+1}}^{(t+1)}}{\mathbf{u}_{e_{k+1}}} \dots \geq \frac{\mathbf{y}_{e_N}^{(t+1)}}{\mathbf{u}_{e_N}}.$$

Then by Algorithm 2, $W_1 = W$, and $S_1 = S$. Furthermore, for each $1 \leq \ell \leq i \leq k$, $\exp(-\mu) \geq \frac{\mathbf{u}_{e_i}}{\mathbf{y}_{e_i}^{(t+1)}} \geq \frac{\mathbf{u}_{e_\ell}}{\mathbf{y}_{e_\ell}^{(t+1)}}$. Thus, by the fact that $\frac{a_1+b_1}{a_2+b_2} \geq \min\{\frac{a_1}{a_2}, \frac{b_1}{b_2}\}$ for any $a_1, b_1 \geq 0, a_2, b_2 > 0$, we have that

$$\frac{W_1}{S_1} = \frac{W}{S} = \frac{\sum_{\ell=1}^k \mathbf{u}_{e_\ell} + \sum_{j=k+1}^N \mathbf{y}_{e_j}^{(t+1)} \exp(-\mu)}{\sum_{\ell=1}^k \mathbf{y}_{e_\ell}^{(t+1)} + \sum_{j=k+1}^N \mathbf{y}_{e_j}^{(t+1)}} \geq \min\{\frac{\mathbf{u}_{e_1}}{\mathbf{y}_{e_1}^{(t+1)}}, \exp(-\mu)\} = \frac{\mathbf{u}_{e_1}}{\mathbf{y}_{e_1}^{(t+1)}}.$$

Thus, Algorithm 2 outputs $\mathbf{w}_{e_1}^{(t+1)} = \min(\frac{W_1 \cdot \mathbf{y}_{e_1}^{(t+1)}}{S_1}, \mathbf{u}_{e_1}) = \mathbf{u}_{e_1}$.

Now by induction, we have that for $i \leq k$, $W_i = \sum_{\ell=i}^k \mathbf{u}_{e_\ell} + \sum_{j=k+1}^N \mathbf{y}_{e_j}^{(t+1)} \exp(-\mu)$, $S_i = \sum_{\ell=i}^k \mathbf{y}_{e_\ell}^{(t+1)} + \sum_{j=k+1}^N \mathbf{y}_{e_j}^{(t+1)}$, and thus

$$\frac{W_i}{S_i} = \frac{\sum_{\ell=i}^k \mathbf{u}_{e_\ell} + \sum_{j=k+1}^N \mathbf{y}_{e_j}^{(t+1)} \exp(-\mu)}{\sum_{\ell=i}^k \mathbf{y}_{e_\ell}^{(t+1)} + \sum_{j=k+1}^N \mathbf{y}_{e_j}^{(t+1)}} \geq \min\{\frac{\mathbf{u}_{e_i}}{\mathbf{y}_{e_i}^{(t+1)}}, \exp(-\mu)\} = \frac{\mathbf{u}_{e_i}}{\mathbf{y}_{e_i}^{(t+1)}}.$$

Thus, Algorithm 2 outputs $\mathbf{w}_{e_i}^{(t+1)} = \min(\frac{W_i \cdot \mathbf{y}_{e_i}^{(t+1)}}{S_i}, \mathbf{u}_{e_i}) = \mathbf{u}_{e_i}$, for each $i \leq k$.

Now let us consider $j = k+1$. It holds that $W_{k+1} = \sum_{\ell=k+1}^N \mathbf{y}_{e_\ell}^{(t+1)} \exp(-\mu)$, $S_j = \sum_{\ell=k+1}^N \mathbf{y}_{e_\ell}^{(t+1)}$. Thus,

$$\frac{W_{k+1}}{S_{k+1}} = \frac{\sum_{\ell=k+1}^N \mathbf{y}_{e_\ell}^{(t+1)} \exp(-\mu)}{\sum_{\ell=k+1}^N \mathbf{y}_{e_\ell}^{(t+1)}} = \exp(-\mu).$$

Thus, Algorithm 2 outputs $\mathbf{w}_{e_{k+1}}^{(t+1)} = \min(\frac{W_{k+1} \cdot \mathbf{y}_{e_{k+1}}^{(t+1)}}{S_{k+1}}, \mathbf{u}_{e_{k+1}}) = \exp(-\mu) \cdot \mathbf{y}_{e_{k+1}}^{(t+1)}$, as the last quantity is less than $\mathbf{u}_{e_{k+1}}$.

Now consider any $j \geq k+1$. By induction, it holds that $W_j = \sum_{\ell=j}^N \mathbf{y}_{e_\ell}^{(t+1)} \exp(-\mu)$, $S_j = \sum_{\ell=j}^N \mathbf{y}_{e_\ell}^{(t+1)}$, and thus $W_j = \exp(-\mu) S_j$. This further implies that Algorithm 2 outputs $\mathbf{w}_{e_j}^{(t+1)} = \min(\frac{W_j \cdot \mathbf{y}_{e_j}^{(t+1)}}{S_j}, \mathbf{u}_{e_j}) = \exp(-\mu) \cdot \mathbf{y}_{e_j}^{(t+1)}$, as the last quantity is less than $\mathbf{u}_{e_j}$.

Therefore, Algorithm 2 always output $\mathbf{w}^{(t+1)}$ such that $\mathbf{w}^*$. That is, $D_\Phi(\mathbf{w}^{(t+1)}, \mathbf{y}^{(t+1)}) = \min_{\mathbf{w} \in \mathcal{X}} D_\Phi(\mathbf{w}, \mathbf{y}^{(t+1)})$. This finishes the proof of the theorem. ■

4. The Lower Bound

Given graph G, let $\ell_h(G) = \max_{(i,j) \in \binom{V}{2}} \sum_{V' = (i,j,v_1,v_2,\dots,v_{h-2}) \in \binom{V}{h}} \prod_{(k,\ell) \in \binom{V'}{2} \setminus \{(i,j)\}} \mathbf{w}_{(k,\ell)}$ denotes the local sensitivity of K_h-motif cuts of G. Note that, its local sensitivity is defined as the maximum K_h-motif cut difference between G and its neighboring graphs. For unweighted graph with maximum degree at most d, we have $\ell_h(G) = O(d^{h-2})$. For unweighted dense graph, we have $\ell_h(G) = \Theta(n^{h-2})$. For unweighted graph generated from $G(n,p)$, we have $\ell_h(G) = \Theta(n^{h-2}p^{\frac{h^2-h-2}{2}})$.

We show the following lower bound. Note that Theorem 4 follows from the following theorem by setting $h = 3$ and $\beta = \frac{3}{4}$.

Theorem 6 *Let $\mathcal{M}$ be an (ε,δ)-differentially private mechanism, and let G be a graph generated from $G(n,p)$ with $(\frac{\log n}{n})^{1/(h-1)} \ll p \leq \frac{1}{2}$. If $\mathcal{M}$ answers the K_h-motif size queries of all (S,T)-cut on G, or on a scaled version of G with total edge weight W, up to an additive error α with probability at least β, then:*

$$\alpha \geq \Omega\left(\max\left(\frac{\sqrt{mn}\cdot \ell_h(G)}{\varepsilon}(1-c), \frac{\sqrt{Wn}\cdot \ell_h(G)}{\varepsilon^{\frac{1}{2}}}(1-c)\right)\right)$$

where $c = \frac{e-1}{e^{\varepsilon}-1}\cdot\frac{9\delta}{\beta}$, and $m = \Theta(pn^2)$ is the number of edges of G.

The proof is based on generalizing the lower bound in (Eliáš et al., 2020) for the edge case using the discrepancy of 3-coloring of h-uniform hypergraphs. Specifically, for an unweighted graph $G = (V,E)$, we consider the matrix $\mathbf{A}$ with $\binom{n}{h}$ columns corresponding to h-tuple of vertices (or a copy of K_h) and rows corresponding to the pairs of sets $S,T \subset V$ such that

$$\mathbf{A}_{(S,T),I} = \begin{cases} 1 & \text{if } I \in (S\times T) \\ 0 & \text{otherwise} \end{cases}$$

Note that $\mathbf{A}$ is fixed and does not depend on G. Let $\mathbf{x} \in \{0,1\}^{\binom{n}{2}}$ be the indicator vector of E. Let $\mathbf{x}_{K_h} = f_{K_h}(x) \in \{0,1\}^{\binom{n}{h}}$ be the indicator vector of K_h in G, i.e. for each tuple $i_1, i_2, \ldots, i_h$ of h different indices, $(\mathbf{x}_{K_h})_{i_1,i_2,\ldots,i_h} = 1$ if the subgraph induced by $i_1, i_2, \ldots, i_h$ is K_h and 0 otherwise. Then $\mathbf{A}\mathbf{x}_{K_h}$ specifies the K_h-motif size of all (S,T)-cuts in G, i.e., we have

$$(\mathbf{A}\mathbf{x}_{K_h})_{S,T} = \sum_{I\in\mathcal{M}(G,K_h)} \mathbf{1}_{I \text{ crosses } (S,T)}.$$

Then we define the discrepancy of A in terms of the set of 3-colorings over the set of all K_h motifs, and give bounds on its discrepancy by using random graphs, which will then imply our lower bound by a reduction from (Muthukrishnan and Nikolov, 2012). Specifically, we define the discrepancy of a matrix as follows.

Definition 7 *Let $\mathbf{B}$ be a $0/1$ matrix with $\binom{n}{h}$ columns and $\mathcal{C} \subseteq \{-1,0,1\}^{\binom{n}{h}}$ be the set of allowed K_h colorings. We define*

$$\mathrm{disc}_{\mathcal{C}}(\mathbf{B}) = \min\{\|\mathbf{B}\chi\|_\infty : \chi \in \mathcal{C}\}$$

We will prove the following lemma, which is a generalization of a result of (Eliáš et al., 2020) (see also (Bollobás and Scott, 2006)).

Lemma 8 *Let $\gamma,\sigma \in (0,1/2)$. Let $\mathcal{C}_{\sigma,\gamma}$ be the set of all vectors $\chi = \mathbf{x}_{K_h} - \mathbf{x}'_{K_h}$ where $\mathbf{x},\mathbf{x}'$ are the indicator vector of edges of graphs, denoted by $G = (V,E)$ and $G' = (V,E')$, respectively, such that*

1. $\|\mathbf{x} - \mathbf{x}'\|_1 \geq \sigma\gamma n^2$;

2. *for each vertex $v \in V$, its degree belongs to the interval $[\gamma n/2, 2\gamma n]$;*

3. *for each edge $e \in E \cup E'$, the number of K_h-instances containing e belongs to the interval $[\frac{\ell_h(G)}{2}, 2\ell_h(G)]$;*

4. *for $1 \leq i \leq h$ and any subset B with i distinct vertices in G and G', the number of vertices t such that t is connected to all vertices in B belongs to the interval $\left[\frac{n\gamma^i}{2}, 2n\gamma^i\right]$.*

Then for the matrix $\mathbf{A}$ defined above, we have

$$\mathrm{disc}_{\mathcal{C}_{\sigma,\gamma}}(\mathbf{A}) \geq 2^{-h-5}\sigma \cdot \gamma^{1/2} n^{3/2} \ell_h(G).$$

Once we have the above lemma, we can finish the proof of Theorem 6 using standard arguments. We defer the details to Appendix E.

5. Conclusion

In this paper, we present the first polynomial-time algorithm for releasing a synthetic graph that effectively preserves the triangle-motif cut structure of an input graph in a differentially private manner. This algorithm extends previous studies (Gupta et al., 2012; Blocki et al., 2012; Upadhyay, 2013; Arora and Upadhyay, 2019; Eliáš et al., 2020; Liu et al., 2024) on differentially private algorithms that maintain edge-motif cut structures to higher-order organizations. This higher-order property has wide applications in analyzing complex networks (Milo et al., 2002; Benson et al., 2016) and has garnered increasing attention in the theoretical computer science community (Kapralov et al., 2022). We also establish a lower bound of the additive error for DP algorithms that answers the K_h-motif cut queries.

Our work leaves several interesting open questions. One immediate question is to develop nearly matching upper and lower bounds for the entire class of graphs, which would likely require new ideas. Another interesting direction is to give a differentially private algorithm for weighted graphs whose additive error depends solely on the number of edges rather than the maximum or total edge weights. This has recently been achieved for edge-motif cut structures using a topology sampler and leveraging the linearity property of these structures (Liu et al., 2024). However, the inherent non-linearity of triangle-motif cut structures presents a significant challenge in extending this approach. Finally, it would be interesting to investigate differentially private algorithms for preserving motif cut structures beyond triangles (or 3-vertex motifs). We believe that our method, combined with optimization techniques for tensors and hypergraphs, could be useful in generating synthetic graphs that preserve K_h-motif cut structures for any constant h.

Acknowledgments

This work is supported in part by NSFC Grant 62272431 and Innovation Program for Quantum Science and Technology (Grant No. 2021ZD0302901).

References

Noga Alon and Assaf Naor. Approximating the cut-norm via grothendieck's inequality. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 72–80, 2004.

Raman Arora and Jalaj Upadhyay. On differentially private graph sparsification and applications. *Advances in neural information processing systems*, 32, 2019.

Kazuoki Azuma. Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal, Second Series*, 19(3):357–367, 1967.

Aharon Ben-Tal and Arkadi Nemirovski. *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*. SIAM, 2001.

András A Benczúr and David R Karger. Approximating st minimum cuts in õ (n 2) time. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 47–55, 1996.

András A Benczúr and David R Karger. Randomized approximation schemes for cuts and flows in capacitated graphs. *SIAM Journal on Computing*, 44(2):290–319, 2015.

Austin R Benson, David F Gleich, and Jure Leskovec. Higher-order organization of complex networks. *Science*, 353(6295):163–166, 2016.

Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. The johnson-lindenstrauss transform itself preserves differential privacy. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 410–419. IEEE, 2012.

Jeremiah Blocki, Elena Grigorescu, Tamalika Mukherjee, and Samson Zhou. How to make your approximation algorithm private: A black-box differentially-private transformation for tunable approximation algorithms of functions with low sensitivity. *arXiv preprint arXiv:2210.03831*, 2022.

Béla Bollobás and Alexander D Scott. Discrepancy in graphs and hypergraphs. *More Sets, Graphs and Numbers: A Salute to Vera Sós and András Hajnal*, pages 33–56, 2006.

Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

Mark Bun, Marek Elias, and Janardhan Kulkarni. Differentially private correlation clustering. In *International Conference on Machine Learning*, pages 1136–1146. PMLR, 2021.

Mark Mar Bun. *New Separations in the Complexity of Differential Privacy*. PhD thesis, 2016.

Shixi Chen and Shuigeng Zhou. Recursive mechanism: towards node differential privacy and unrestricted joins. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 653–664, 2013.

Yu Chen, Sanjeev Khanna, and Ansh Nagda. Near-linear size hypergraph cut sparsifiers. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 61–72. IEEE, 2020.

Vincent Cohen-Addad, Chenglin Fan, Silvio Lattanzi, Slobodan Mitrovic, Ashkan Norouzi-Fard, Nikos Parotsidis, and Jakub M Tarnawski. Near-optimal correlation clustering with privacy. *Advances in Neural Information Processing Systems*, 35:33702–33715, 2022.

John M Danskin. *The theory of max-min and its application to weapons allocation problems*, volume 5. Springer Science & Business Media, 2012.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006.

Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60. IEEE, 2010.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Talya Eden, Quanquan C Liu, Sofya Raskhodnikova, and Adam Smith. Triangle counting with local edge differential privacy. *arXiv preprint arXiv:2305.02263*, 2023.

Marek Eliáš, Michael Kapralov, Janardhan Kulkarni, and Yin Tat Lee. Differentially private release of synthetic graphs. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 560–578. SIAM, 2020.

Alan Frieze and Ravi Kannan. Quick approximation to matrices and applications. *Combinatorica*, 19(2):175–220, 1999.

Benyamin Ghojogh, Ali Ghodsi, Fakhri Karray, and Mark Crowley. Kkt conditions, first-order and second-order optimization, and distributed optimization: tutorial and survey. *arXiv preprint arXiv:2110.01858*, 2021.

Anupam Gupta, Aaron Roth, and Jonathan Ullman. Iterative constructions and private data release. In *Theory of Cryptography: 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings 9*, pages 339–356. Springer, 2012.

Moritz Hardt and Guy N Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *2010 IEEE 51st annual symposium on foundations of computer science*, pages 61–70. IEEE, 2010.

Michael Hay, Chao Li, Gerome Miklau, and David Jensen. Accurate estimation of the degree distribution of private networks. In *2009 Ninth IEEE International Conference on Data Mining*, pages 169–178. IEEE, 2009.

Jacob Imola, Takao Murakami, and Kamalika Chaudhuri. {Communication-Efficient} triangle counting under local differential privacy. In *31st USENIX security symposium (USENIX Security 22)*, pages 537–554, 2022.

Jacob Imola, Alessandro Epasto, Mohammad Mahdian, Vincent Cohen-Addad, and Vahab Mirrokni. Differentially private hierarchical clustering with provable approximation guarantees. In *International Conference on Machine Learning*, pages 14353–14375. PMLR, 2023.

Michael Kapralov, Mikhail Makarov, Sandeep Silwal, Christian Sohler, and Jakab Tardos. Motif cut sparsifiers. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 389–398. IEEE, 2022.

David R Karger. Random sampling in cut, flow, and network design problems. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 648–657, 1994.

Vishesh Karwa and Aleksandra B Slavković. Differentially private graphical degree sequences and synthetic graphs. In *Privacy in Statistical Databases: UNESCO Chair in Data Privacy, International Conference, PSD 2012, Palermo, Italy, September 26-28, 2012. Proceedings*, pages 273–285. Springer, 2012.

Vishesh Karwa, Sofya Raskhodnikova, Adam Smith, and Grigory Yaroslavtsev. Private analysis of graph structure. *Proceedings of the VLDB Endowment*, 4(11):1146–1157, 2011.

Solomon Kullback. A lower bound for discrimination information in terms of variation (corresp.). *IEEE transactions on Information Theory*, 13(1):126–127, 1967.

Yin Tat Lee, Aaron Sidford, and Sam Chiu-wai Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 1049–1065. IEEE, 2015.

Yang Li, Michael Purcell, Thierry Rakotoarivelo, David Smith, Thilina Ranbaduge, and Kee Siong Ng. Private graph data release: A survey. *ACM Computing Surveys*, 55(11):1–39, 2023.

Jingcheng Liu, Jalaj Upadhyay, and Zongrui Zou. Optimal bounds on private graph approximation. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1019–1049. SIAM, 2024.

Sebastian Meiser. Approximate and probabilistic differential privacy definitions. *Cryptology ePrint Archive*, 2018.

Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

Shanmugavelayutham Muthukrishnan and Aleksandar Nikolov. Optimal private halfspace counting via discrepancy. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 1285–1292, 2012.

Huda Nassar, Caitlin Kennedy, Shweta Jain, Austin R Benson, and David Gleich. Using cliques with higher-order spectral embeddings improves graph visualizations. In *Proceedings of The Web Conference 2020*, pages 2927–2933, 2020.

Dung Nguyen, Mahantesh Halappanavar, Venkatesh Srinivasan, and Anil Vullikanti. Faster approximate subgraph counts with privacy. *Advances in Neural Information Processing Systems*, 36, 2024.

Davide Proserpio, Sharon Goldberg, and Frank McSherry. A workflow for differentially-private graph synthesis. In *Proceedings of the 2012 ACM workshop on Workshop on online social networks*, pages 13–18, 2012.

Martin Rosvall, Alcides V Esquivel, Andrea Lancichinetti, Jevin D West, and Renaud Lambiotte. Memory in network flows and its effects on spreading dynamics and community detection. *Nature communications*, 5(1):4630, 2014.

Mark Rudelson and Roman Vershynin. Hanson-wright inequality and sub-gaussian concentration. 2013.

Venu Satuluri, Srinivasan Parthasarathy, and Yiye Ruan. Local graph sparsification for scalable clustering. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 721–732, 2011.

C Seshadhri, Aneesh Sharma, Andrew Stolman, and Ashish Goel. The impossibility of low-rank representations for triangle-rich complex networks. *Proceedings of the National Academy of Sciences*, 117(11):5631–5637, 2020.

Morton Slater. Lagrange multipliers revisited. In *Traces and emergence of nonlinear programming*, pages 293–306. Springer, 2013.

Konstantinos Sotiropoulos and Charalampos E Tsourakakis. Triangle-aware spectral sparsifiers and community detection. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1501–1509, 2021.

Charalampos E Tsourakakis, Mihail N Kolountzakis, and Gary L Miller. Triangle sparsifiers. *J. Graph Algorithms Appl.*, 15(6):703–726, 2011.

Charalampos E Tsourakakis, Jakub Pachocki, and Michael Mitzenmacher. Scalable motif-aware graph clustering. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1451–1460, 2017.

Jalaj Upadhyay. Random projections, graph sparsification, and differential privacy. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 276–295. Springer, 2013.

Stanley Wasserman. Social network analysis: Methods and applications. *The Press Syndicate of the University of Cambridge*, 1994.

Hao Yin, Austin R Benson, Jure Leskovec, and David F Gleich. Local higher-order graph clustering. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 555–564, 2017.

Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Private release of graph statistics using ladder functions. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, pages 731–745, 2015.

Ziwei Zhang, Peng Cui, Xiao Wang, Jian Pei, Xuanrong Yao, and Wenwu Zhu. Arbitrary-order proximity preserved network embedding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2778–2786, 2018.

Appendix A. More Discussions

A.1. Other Related Work

Gupta et al. (2012) introduced a framework called iterative database construction algorithms (IDC) and utilized this framework to design various mechanisms that can provide private responses to edge cut queries. These mechanisms include the Multiplicative Weight Update IDC, which is based on the private Multiplicative Weight Update algorithm (Hardt and Rothblum, 2010), with an additive error of approximately $\tilde{O}(m^{\frac{1}{2}}n^{\frac{1}{2}}/\varepsilon^{\frac{1}{2}})$, as well as the Frieze and Kannan IDC, which is based on the Frieze and Kannan low-rank decomposition algorithm (Frieze and Kannan, 1999) and has an additive error of approximately $O(m^{\frac{1}{4}}n/\varepsilon^{\frac{1}{2}})$.

However, the above algorithms can only release the value of edge cuts. In order to release a synthetic graph that approximates every edge cut, Gupta et al. (2012) gave an $(\varepsilon, 0)$-differentially private algorithm with additive error $O(n\sqrt{n}/\varepsilon)$, which is based on randomized response. After that, Blocki et al. (2012) showed an algorithm that can release a synthetic graph answering k predetermined queries with additive error at most $O(n\sqrt{\log k}/\varepsilon)$, hence $O(n\sqrt{n}/\varepsilon)$ when considering all edge cuts, by a nice utilization of Johnson-Lindenstrauss transform. Subsequent advancements by Upadhyay (2013), Arora and Upadhyay (2019), Eliáš et al. (2020), and Liu et al. (2024) introduced refined algorithms. In particular, Eliáš et al. (2020) came up with an mirror descent based algorithm which can achieve $\tilde{O}(\sqrt{\|\mathbf{w}\|_1 n/\varepsilon})$ error guarantee, and Liu et al. (2024) designed an algorithm that can achieve $\tilde{O}(\sqrt{mn}/\varepsilon)$ error guarantee, which is based on a topology sampler and the EKKL approach.

There have been several theoretical work on counting motifs (e.g. triangles) differentially privately (Chen and Zhou, 2013; Karwa et al., 2011; Zhang et al., 2015; Blocki et al., 2022; Nguyen et al., 2024). Chen and Zhou (2013) provided a solution of subgraph counting to achieve node DP, for any kind of subgraphs. There also exist efficient polynomial time private algorithms for subgraph counting focused on k-triangles, k-stars and k-cliques (Karwa et al., 2011; Zhang et al., 2015; Nguyen et al., 2024). Several works studied triangle counting problem under the setting of local differential privacy (Eden et al., 2023; Imola et al., 2022). Note that our goal problem is to privately release a synthetic graph that can answer all the triangle-motif cut queries including triangle counting (by querying the sparsifier on the n singleton cuts), hence it is a harder problem than differentially private triangle counting.

Motif analysis has had a profound impact on graph clustering (Benson et al., 2016; Yin et al., 2017; Tsourakakis et al., 2017). Differentially private graph clustering also received increasing attention recently in the community (Bun et al., 2021; Cohen-Addad et al., 2022; Imola et al., 2023).

Graph sparsification is a well-known technique to speed up the algorithms based on edge cut (Karger, 1994; Benczúr and Karger, 2015), which is to construct a sparse graph that approximates all the cuts within a $1+\varepsilon$ factor. Benczúr and Karger (1996) achieved a cut sparsifier with $O(n/\varepsilon^2)$ size in nearly linear time, and Chen et al. (2020) constructed a hypergraph sparsifier with near-linear size.

For the study of motif sparsification, Tsourakakis et al. (2011) came up with an algorithm that can obtain a sparse subgraph that preserves the global triangle count. After that, Kapralov et al. (2022) obtained a stronger algorithm, which can release a sparse subgraph that preserve count of motifs crossing each cut. Moreover, Sotiropoulos and Tsourakakis (2021) introduced a triangle-aware spectral sparsifier, which is sparsifier with respect to edges that has better space bounds for graphs containing many triangles.

A.2. Synthetic Graphs with Multiplicative Errors and Interactive Solutions

We note that if one allows for exponential time and multiplicative error, then one can achieve a better additive error. This is similar to the edge motif cut case. The reason is as follows: It is known that there exists a polynomial-time algorithm that constructs a motif cut sparsifier with only $\tilde{O}(n/\eta^2)$ edges for any $\eta > 0$ (see (Kapralov et al., 2022)). This sparsifier ensures that for every cut $(S, V \setminus S)$, the weighted count of copies of motif M crossing the cut in G' is within a $1 + \eta$ factor of the number of copies of M crossing the same cut in G.

Given the existence of the above sparsifier, we can apply the exponential mechanism and restrict its range to every potential output graph with $\tilde{O}(n)$ edges (for any constant η). Moreover, we can use the maximum motif weight cut error as the scoring function. One can then show that the exponential mechanism enables the release of a synthetic graph G' where each motif weight cut of G is approximated within an expected additive error of $\tilde{O}(n^2)$ and a multiplicative error of $(1+\eta)$ in expectation. However, the main drawback of this approach lies in its exponential time complexity.

Similar to the synthetic graph for edge counts of cuts (Eliáš et al., 2020), the synthetic graph released by our algorithm is not necessarily sparse, i.e., it may not have $\tilde{O}(n)$ edges. If necessary, one can indeed sparsify the output of our algorithm using the motif cut sparsification algorithm given by Kapralov et al. (2022) to obtain a DP sparsifier, leveraging the post-processing property of differential privacy. However, this will introduce multiplicative errors.

Finally, we note that an interactive solution for preserving privacy in the motif cut structure could also be considered (see Appendix A.1 for considerations related to edge cuts). In this scenario, data analysts could specify any cut $(S, V \setminus S)$ with the aim of determining (with some acceptable error) the number of triangles or motifs connecting the two groups while maintaining privacy. However, our primary focus is on a stronger, non-interactive solution, i.e. to release a private synthetic dataset: a new, private graph that approximately preserves the motif cut function of the original graph.

A.3. Some Tempting Approaches That Do Not Work

There are several natural approaches that may seem promising for privately releasing synthetic graphs for triangle-motif cuts, which we outline below and briefly explain why they do not work.

One initial approach could involve utilizing the motif cut sparsifier algorithm proposed by Kapralov et al. (2022), followed by incorporating a noise addition mechanism into the sparsification process. However, similar to the challenges faced in the edge cut case, making such sparsification algorithms differentially private is highly challenging. This difficulty arises primarily because the algorithm relies on importance sampling of existing edges and never outputs non-edges, which poses a fundamental obstacle for DP algorithms. We refer to the discussion in (Eliáš et al., 2020) for a more detailed explanation of the limitations associated with this approach.

Another natural approach is as follows: One can first convert the original graph to a (triangle) hypergraph by creating a hyperedge for each triangle, and then attempt to apply DP hyperedge cut release algorithms. However, to the best of our knowledge, there is no DP algorithm for releasing a synthetic hypergraph while preserving the hyperedge cuts. Furthermore, even if such an algorithm existed, it would not fully address our problem because we cannot convert a hypergraph back into a graph. In fact, it is possible for two graphs G and G' to be very different from each other, while their corresponding hypergraphs are identical. For example, if both G and G' are triangle-free and far from each other, their hypergraphs will be the same, i.e., the hypergraph with no hyperedge at all.

The third approach is to make use of the triangle-motif-weighted graph associated with the input graph (Benson et al., 2016). That is, one can first convert the original graph G into a triangle-motif weighted graph $G_\triangle$ with a weight vector $\mathbf{w}_\triangle$, where $\mathbf{w}_\triangle(e)$ denotes the sum of the weights of triangles containing the endpoints of edge e simultaneously. For triangle-motif weighted graphs, there is a useful property: the size of the edge cut $(S, V \setminus S)$ in $G_\triangle$ is exactly twice the sum of the weights of triangles crossing the cut in G. Therefore, a naive approach would be to use a private edge cut release algorithm, such as the one proposed in (Liu et al., 2024), on the triangle-motif weighted graph. However, after applying the DP algorithm on $G_\triangle$, the resulting graph H may not correspond to a triangle-motif weighted graph, i.e. there does not exist a graph whose triangle-motif weighted graph is H. As a result, through this approach, we can only privately release the values of triangle-motif cuts but cannot release a synthetic graph.

Appendix B. Supplementary Preliminaries

Here, we give the definitions of differential privacy, motif adjacency matrix, and introduce the convex optimization that will be utilized by the algorithms and their analysis.

B.1. Differential Privacy

The definition of differential privacy (Definition 2) relies on the definition of neighboring datasets. For weighted graphs, we have the following definition of neighboring graphs:

Definition 9 (Neighboring graphs) *Given weighted graphs G with edge weight vector $\mathbf{w}$ and G' with edge weight vector $\mathbf{w}'$, G and G' are called neighboring graphs if $\mathbf{w}$ and $\mathbf{w}'$ differ by at most 1 in the ℓ_1 norm, i.e., $\|\mathbf{w} - \mathbf{w}'\|_1 \leq 1$.*

For an unweighted graph G, we let $\mathbf{w}_e = 1$ if G has the edge e; otherwise, $\mathbf{w}_e = 0$. Therefore, for unweighted graphs, G and G' are neighboring graphs if they differs by exactly 1 edge.

A key feature of differential privacy algorithms is that they preserve privacy under post-processing. That is, without any auxiliary information about the dataset, any analyst cannot compute a function that makes the output less private.

Lemma 10 (Post processing (Dwork et al., 2014)) *Let $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ be a (ε, δ)-differentially private algorithm. Let $f : \mathcal{R} \to \mathcal{R}'$ be any function, then $f \circ \mathcal{M}$ is also (ε, δ)-differentially private.*

Sometimes we need to repeatedly use differentially private mechanisms on the same dataset, and obtain a series of outputs. The privacy guarantee of the outputs can be derived by following theorems.

Lemma 11 (Adaptive composition (Dwork et al., 2006)) *Suppose $\mathcal{M}_1(x) : \mathcal{D} \to \mathcal{R}_1$ is $(\varepsilon_1, \delta_1)$-differentially private and $\mathcal{M}_2(x, y) : \mathcal{D} \times \mathcal{R}_1 \to \mathcal{R}_2$ is $(\varepsilon_2, \delta_2)$-differentially private with respect to x for any fixed $y \in \mathcal{R}_1$, then the composition $(\mathcal{M}_1(x), \mathcal{M}_2(x, \mathcal{M}_1(x)))$ is $(\varepsilon_1 + \varepsilon_2, \delta_1 + \delta_2)$-differentially private.*

Lemma 12 (Advanced composition lemma (Dwork et al., 2010)) *For parameters $\varepsilon > 0$ and $\delta, \delta' \in [0,1]$, the composition of k (ε, δ)-differentially private algorithms is a $(\varepsilon', k\delta+\delta')$-differentially private algorithm, where $\varepsilon' = \sqrt{8k\log(1/\delta')}$.*

Now, we introduce basic mechanisms that preserve differential privacy, which are ingredients that build our algorithm. First, we define the sensitivity of query functions.

Definition 13 (ℓ_p-sensitivity) *Let $f : \mathcal{D} \to \mathbb{R}^k$ be a query function on datasets. The sensitivity of f (with respect to $\mathcal{X}$) is $\Delta_p(f) = \max_{\substack{x,y\in\mathcal{D}\\ x\sim y}} \|f(x) - f(y)\|_p$.*

Based on the definition of sensitivity and Laplace distribution, we can get a mechanism that preserve differential privacy as follows.

Lemma 14 (Laplace mechanism) *Suppose $f : \mathcal{D} \to \mathbb{R}^k$ is a query function with ℓ_1 sensitivity $\Delta_1(f) \le \Delta$. Then the mechanism $\mathcal{M}(D) = f(D) + (Z_1, \cdots, Z_k)^\top$ where $Z_1, \cdots, Z_k$ are i.i.d random variables drawn from $\mathrm{Lap}\left(\frac{\Delta}{\varepsilon}\right)$. Given $b > 0$, $\mathrm{Lap}\,(b)$ is the Laplace distribution with density*

$$\mathrm{Lap}(x; b) := \frac{1}{2b}\exp\left(-\frac{|x|}{b}\right).$$

The Laplace distribution has the following concentration bound,

Lemma 15 (Laplace concentration bound) *If $Y \sim \mathrm{Lap}(b)$, then for any $t > 0$, we have*

$$\Pr[|Y| \ge tb] = \exp(-t)$$

Another important concept is probabilistic differential privacy. It is defined as follows.

Definition 16 ((ε, δ)-probabilistic differential privacy) *For fixed $\varepsilon > 0$ and $\delta \in [0, 1)$, we say that $\mathcal{M}$ preserves (ε, δ)-probabilistic differential privacy if $\mathcal{M}$ is ε-differentially private with probability at least $(1 - \delta)$, i.e., for any pair of neighboring datasets $x, y \in \mathcal{D}$, there is a set $S^\delta \subset \mathcal{R}$ with $\Pr[\mathcal{M}(x) \in S^\delta] \le \delta$, s.t. for any measurable set $S \subset \mathcal{R}$, it holds that*

$$\Pr[\mathcal{M}(x) \in S] \le \Pr[\mathcal{M}(y) \in S] \cdot e^{\varepsilon}.$$

If a mechanism preserves (ε, δ)-probabilistic differential privacy, then it also preserves (ε, δ)-differential privacy, while the opposite direction does not hold (Meiser, 2018).

B.2. Motif Cut and Motif Adjacency Matrix

We will make use of the motif adjacency matrix to deal with the motif cut, which is defined as follows.

Definition 17 (Motif adjacency matrix (Benson et al., 2016)) *Given $G = (V, E, \mathbf{w})$ and a motif M, a motif adjacency matrix of G with respect to M is defined by,*

$$(\mathbf{A}_M)_{i,j} = \sum_{I\in\mathcal{M}(G,M):i,j\in V_I} w(I).$$

That is, $(\mathbf{A}_M)_{i,j}$ is the sum of weights of motif instances in which both i and j are involved.

Notably, the computational time to form a motif adjacency matrix $\mathbf{A}_M$ is bounded by the time to find all instances of the motif in the graph. And obviously, for a motif on k nodes, we can compute $\mathbf{A}_M$ in $\Theta(n^k)$ time by checking all the k-tuples of nodes in a graph. When the motif is triangle, there exist more efficient algorithms to list all the motifs (Benson et al., 2016).

Fact 18 *Let $\mathbf{D}_M^{(e)}$ denote the derivative of $\mathbf{A}_M$ at $\mathbf{w}_e$ for some $e = (k, \ell) \in \binom{V}{2}$.*

$$\left(\mathbf{D}_M^{(k,\ell)}\right)_{i,j} = \sum_{I \in \mathcal{M}(K^n, M): i,j,k,\ell \in V_I} w^{(e)}(I),$$

where $w^{(e)}(I) = \prod_{e' \in E_I : e' \neq e} w(e')$.

It is worth noting that $\mathbf{D}_M^{(e)}$ is actually the divergence of motif adjacency matrix between graph G and its neighboring graph G', which differs from G in only one edge e by 1 weight.

Furthermore, the form of the second-order or higher order derivative of $\mathbf{A}_M$ are similar to the above.

In this paper, we focus on the special case when the motif is triangle. The "M" in the relevant notation will be replaced by "$\triangle$".

In addition, when we consider triangles, the derivatives of $\mathbf{A}_\triangle$ have explicit form as follows. Recall that $\mathbf{D}_\triangle^{(k,\ell)}$ denotes the derivative of $\mathbf{A}_\triangle$ at $\mathbf{w}_e$, where $e = (k, \ell)$.

Fact 19 *Let $\mathbf{E}_\triangle^{((i,j),(k,\ell))}$ denote the second-order derivative of $\mathbf{A}_\triangle$ at $\mathbf{w}_{(i,j)}$ and $\mathbf{w}_{(k,\ell)}$, and let $\mathbf{B}_{(e)}$ denote a matrix which has value 1 at the entry corresponding to e and value 0 otherwise. Then it holds that*

$$\left(\mathbf{D}_\triangle^{(k,\ell)}\right)_{i,j} = \begin{cases} \sum_{s \neq k,\ell} \mathbf{w}_{(k,s)} \mathbf{w}_{(s,\ell)}, & (k,\ell) = (i,j) \\ \mathbf{w}_{(k,j)} \mathbf{w}_{(j,\ell)}, & k = i \text{ and } j \neq \ell \\ 0, & o.w. \end{cases}$$

$$\mathbf{E}_\triangle^{((i,j),(k,\ell))} = \begin{cases} \mathbf{w}_{(j,\ell)} \cdot (\mathbf{B}_{(i,j)} + \mathbf{B}_{(i,\ell)} + \mathbf{B}_{(j,\ell)}), & i = k \text{ and } j \neq \ell \\ 0, & o.w. \end{cases}$$

B.3. Cut Norm and Its Approximation

The following method to bound the additive error for edge cut is introduced by Alon and Naor (2004).

Consider graphs $\overline{G}$ and G, where $\overline{G}$ can be considered as the original graph, and G can be considered as an approximation to $\overline{G}$. Let $\overline{\mathbf{A}}$ and $\mathbf{A}$ be their adjacency matrices. Then for a fixed cut $(S, V \setminus S)$, the edge-cut size of $\overline{G}$ is $\sum_{v \in S, u \in V \setminus S} \left(\overline{\mathbf{A}}\right)_{u,v}$ and the edge-cut size of G is $\sum_{v \in S, u \in V \setminus S} \left(\mathbf{A}\right)_{u,v}$. So we can see that the edge-cut difference between G and $\overline{G}$ is the following expression:

$$\max_{S \subset V} \left\{ \left| \sum_{v \in S, u \in V \setminus S} \left(\overline{\mathbf{A}}\right)_{u,v} - \sum_{v \in S, u \in V \setminus S} \left(\mathbf{A}\right)_{u,v} \right| \right\} = \max_{\mathbf{x},\mathbf{y} \in \{0,1\}^n, \mathbf{x}+\mathbf{y}=\mathbf{1}} \left\{ \left| \mathbf{x}^\top \left(\overline{\mathbf{A}} - \mathbf{A}\right) \mathbf{y} \right| \right\}$$

This expression can be bounded by the cut norm which is defined as follows.

Definition 20 (Cut norm (Frieze and Kannan, 1999)) *For a matrix* $\mathbf{M} \in \mathbb{R}^{n\times n}$ *its* cut norm *is defined as*

$$\|\mathbf{M}\|_{cut} = \max\left\{\left|\mathbf{x}^\top \mathbf{M}\mathbf{y}\right| : \mathbf{x}, \mathbf{y} \in \{0,1\}^n\right\}$$

The following lemma was due to (Alon and Naor, 2004).

Lemma 21 ((Alon and Naor, 2004)) *The cut norm of the matrix* $\mathbf{M}$ *can be approximated up to a constant factor using the following SDP.*

$$\max\left\{\sum_{i,j=1}^{n} \mathbf{M}_{i,j}\mathbf{u}_i^\top \mathbf{v}_j : \mathbf{u}_i, \mathbf{v}_i \in \mathbb{R}^n, \|\mathbf{u}_i\| = \|\mathbf{v}_i\| = 1, \forall i\right\}$$

For any two $n \times n$ matrices $\mathbf{B}, \mathbf{C}$, we let $\mathbf{B} \bullet \mathbf{C} = \operatorname{tr}\left(\mathbf{B}^\top \mathbf{C}\right) = \sum_{i,j=1}^{n} \mathbf{B}_{i,j}\mathbf{C}_{i,j}$. We have the following lemma.

Lemma 22 *The edge-cut difference between* G *and* $\overline{G}$ *can be bounded by the following SDP up to a constant factor:*

$$\max\left\{\begin{pmatrix} \mathbf{0} & \mathbf{A} - \overline{\mathbf{A}} \\ \mathbf{A} - \overline{\mathbf{A}} & \mathbf{0} \end{pmatrix} \bullet \mathbf{X} : \mathbf{X} \text{ is symmetric}, \mathbf{X} \succeq \mathbf{0}, \text{and } \mathbf{X}_{i,i} = 1 \text{ for } \forall i\right\} \tag{14}$$

Proof

Let $\mathbf{z}_i = (\mathbf{0}, \mathbf{u}_i) \in \mathbb{R}^{2n}$ for $i = 1, \ldots, n$ and $\mathbf{z}_i = (\mathbf{0}, \mathbf{v}_i) \in \mathbb{R}^{2n}$ for $i = n+1, \ldots, 2n$. Define $\mathbf{X} \in \mathbb{R}^{2n\times 2n}$ s.t. $\mathbf{X}_{i,j} = \mathbf{z}_i^\top \mathbf{z}_j$ for $\forall i, j$. Note that $\mathbf{X}$ is symmetric, $\mathbf{X} \succeq \mathbf{0}$, and $\mathbf{X}_{i,i} = 1$ for $\forall i$. Since $\mathbf{M} := \mathbf{A} - \overline{\mathbf{A}}$ is symmetric, then,

$$\sum_{i,j}^{n} \mathbf{M}_{i,j}\mathbf{u}_i^\top \mathbf{v}_j = \frac{1}{2}\sum_{i,j}^{n}\left(\mathbf{M}_{i,j}\mathbf{u}_i^\top \mathbf{v}_j + \mathbf{M}_{j,i}\mathbf{u}_j^\top \mathbf{v}_i\right) = \frac{1}{2}\sum_{i,j}^{n}\mathbf{M}_{i,j}\left(\mathbf{z}_i^\top \mathbf{z}_{n+j} + \mathbf{z}_j^\top \mathbf{z}_{n+i}\right) = \frac{1}{2}\begin{pmatrix} \mathbf{0} & \mathbf{M} \\ \mathbf{M} & \mathbf{0} \end{pmatrix} \bullet \mathbf{X}.$$

■

B.4. Convex Optimization

Our algorithm needs to solve a minimization problem for a convex function while preserving differential privacy. The method we use for convex optimization is stochastic mirror descent, which is fully analyzed by Bubeck (2015). Given a convex function $f(x)$ defined over a convex set $\mathcal{X}$, a mirror map $\Phi(x)$ (a strongly convex function), and step length η, we define here the Bregman divergence associated to Φ as

$$D_\Phi(x, y) = \Phi(x) - \Phi(y) - \nabla\Phi(y)^\top (x - y).$$

Then the algorithm is described as follows.

Denote $R_{\ell,t}$ as the randomness while computing $g^{(t)}$ in the outer iteration ℓ. For any fixed outer iteration ℓ, the difference between $\mathrm{E}_{\{\gamma_\ell, R_{\ell,t}\}}[f(\hat{x}_\ell)]$ and $\min_{x\in\mathcal{X}} f(x)$ is bounded by the following theorem. Here, $\mathrm{E}_{\{\gamma_\ell, R_{\ell,t}\}}$ represents the expectation taken over the randomness corresponding to random variables $\gamma_\ell, R_{\ell,t}$, where $\ell = 1, \ldots, L$ and $t = 1, \ldots, T$.

Algorithm 3 Stochastic Mirror Descent

Input: A convex function $f(x)$ defined over a convex set $\mathcal{X}$, a mirror function $\Phi(x)$, noise distributions Γ_ℓ, step length η, success probability β.

Output: $\hat{x} \in \mathcal{X}$ s.t. $f(\hat{x})$ approximates $\min_{x\in\mathcal{X}} f(x)$.

1: Set $L = \log_3(\frac{1}{\beta})$
2: **for** $\ell = 1, \ldots, L$ **do**
3:   Choose $x^{(1)} \in \arg\min_{x\in\mathcal{X}} \Phi(x)$
4:   Choose γ_ℓ from distribution Γ_ℓ
5:   **for** $t = 1, \ldots, T$ **do**
6:     Compute an unbiased estimator $g^{(t)}$ of $\nabla f(x^{(t)})$
7:     Update $g^{(t)} \leftarrow g^{(t)} + \gamma_\ell$
8:     Choose $y^{(t+1)}$ s.t. $\nabla\Phi(y^{(t+1)}) = \nabla\Phi(x^{(t)}) - \eta g^{(t)}$
9:     Let $x^{(t+1)} = \arg\min_{x\in\mathcal{X}} D_\Phi(x, y^{(t+1)})$
10:   **end for**
11:   Let $\hat{x}_\ell = \frac{1}{T}\sum_{t=1}^{T} x^{(t)}$
12: **end for**
13: **return** $\hat{x} = \arg\min(f(\hat{x}_1), \ldots, f(\hat{x}_L))$;

Theorem 23 (Stochastic mirror descent (Bubeck, 2015)) *Let Φ be a ρ-strongly convex map with respect to $\|\cdot\|$. Given a convex function f defined over convex set $\mathcal{X}$ with $x^* = \arg\min_{x\in\mathcal{X}} f(x)$. Additionally, assume the noises satisfy $\mathbb{E}[\gamma_\ell] = 0$. Consider Algorithm 3 for some fixed outer iteration $\ell \leq L$.*

Assume that $\Phi(x^*) - \min_{x\in\mathcal{X}} \Phi(x) = R^2$, $\mathbb{E}_{\{\gamma_\ell, R_{\ell,t}\}}[g^{(t)}] = \nabla f(x^{(t)})$ and $\mathbb{E}_{\{\gamma_\ell, R_{\ell,t}\}}[\|g^{(t)}\|_*^2] \leq B^2$ for all $t \leq T$, where $\|\cdot\|_*$ denotes the norm dual to $\|\cdot\|$. After T iterations with step length $\eta = \frac{R}{B}\sqrt{\frac{2}{T}}$, denote the output of the Line 3 to Line 11 in Algorithm 3 as $\hat{x}_\ell \in \mathcal{X}$, then $\mathbb{E}_{\{\gamma_\ell, R_{\ell,t}\}}[f(\hat{x}_\ell)] \leq f(x^*) + RB\sqrt{\frac{2}{\rho T}}$.

The above theorem is a generalization of the stochastic mirror descent algorithm (Theorem 6.1 in (Bubeck, 2015)). Following the proof of Theorem 4.2 in (Bubeck, 2015), we can have $\sum_{t=1}^{T} {g^{(t)}}^\top (x^{(t)} - x) \leq \frac{R^2}{\eta} + \frac{\eta}{2\rho}\sum_{t=1}^{T}\left\|g^{(t)}\right\|_*^2$. Then since f is a convex function, we have

$$\begin{aligned}
&\mathbb{E}_{\{\gamma_\ell, R_{\ell,t}\}}\left[f(\frac{1}{T}\sum_{t=1}^{T} x^{(t)}) - f(x)\right] \leq \frac{1}{T}\mathbb{E}_{\{\gamma_\ell, R_{\ell,t}\}}\left[\sum_{t=1}^{T}(f(x^{(t)}) - f(x))\right] \\
\leq &\frac{1}{T}\mathbb{E}_{\{\gamma_\ell, R_{\ell,t}\}}\left[\sum_{t=1}^{T}\nabla f(x^{(t)})^\top (x^{(t)} - x)\right] = \frac{1}{T}\mathbb{E}_{\{\gamma_\ell, R_{\ell,t}\}}\left[\sum_{t=1}^{T}\mathbb{E}_{\{\gamma_\ell, R_{\ell,t}\}}\left[g^{(t)}\right]^\top (x^{(t)} - x)\right] \\
= &\frac{1}{T}\mathbb{E}_{\{\gamma_\ell, R_{\ell,t}\}}\left[\sum_{t=1}^{T} {g^{(t)}}^\top (x^{(t)} - x)\right] \leq \frac{1}{T}\mathbb{E}_{\{\gamma_\ell, R_{\ell,t}\}}\left[\frac{R^2}{\eta} + \frac{\eta}{2\rho}\sum_{t=1}^{T}\left\|g^{(t)}\right\|_*^2\right] \leq \frac{R^2}{\eta} + \frac{\eta B^2}{2\rho}.
\end{aligned}$$

The theorem is proved when we choose $\eta = \frac{R}{B}\sqrt{\frac{2}{T}}$.

Note that by Markov inequality, for each $\ell \leq L$, it holds that with probability at least $2/3$, $f(\hat{x}_\ell) \leq 3f(x^*) + 3RB\sqrt{\frac{2}{\rho T}}$. Therefore, with probability at least $1 - (1/3)^L \geq 1 - \beta$, at least one of $\hat{x}_1, \ldots, \hat{x}_L$, say $\hat{x}_{i_0}$, satisfies that $f(\hat{x}_{i_0}) \leq 3f(x^*) + 3RB\sqrt{\frac{2}{\rho T}}$. Thus, we have the following corollary.

Corollary 24 *Let Φ be a ρ-strongly convex map with respect to $\|\cdot\|$. Given a convex function f defined over convex set $\mathcal{X}$ with $x^* = \arg\min_{x\in\mathcal{X}} f(x)$. Assume that Γ_ℓ has zero expectation for each ℓ. Assume that $\Phi(x^*) - \min_{x\in\mathcal{X}} \Phi(x) = R^2$, $\mathbb{E}_{\{\gamma_\ell, R_{\ell,t}\}}[g^{(t)}] = \nabla f(x^{(t)})$ and $\mathbb{E}_{\{\gamma_\ell, R_{\ell,t}\}}[\|g^{(t)}\|_*^2] \leq B^2$ for all $t \leq T$ and $\ell \leq L$, where $\|\cdot\|_*$ denotes the norm dual to $\|\cdot\|$. Algorithm 3 with parameter $\eta = \frac{R}{B}\sqrt{\frac{2}{T}}$ will output $\hat{x} \in \mathcal{X}$ s.t. $f(\hat{x}) \leq 3f(x^*) + 3RB\sqrt{\frac{2}{\rho T}}$ with probability at least $1-\beta$.*

However, when we need to compute the optimal solution of some convex optimization precisely, we can use KKT conditions (Ben-Tal and Nemirovski, 2001; Ghojogh et al., 2021) instead.

Lemma 25 (KKT conditions, Slater's condition (Slater, 2013)) *For a constrained optimization problem:*

$$\begin{aligned} &\text{minimize}_{x\in\mathcal{X}} f(x) \\ \text{s.t.}\ & g_i(x) \leq 0, \forall i \in \{1, \ldots, m_1\} \\ & h_i(x) = 0, \forall i \in \{1, \ldots, m_2\} \end{aligned}$$

Then we have:

1. *The optimal solution $\mathbf{x}$ must satisfy KKT conditions as follows:*

$$\begin{cases} \nabla_x f(x) + \sum_i^n \lambda_i \nabla_x g_i(x) + \sum_i^n \mu_i \nabla_x h_i(x) = 0 \\ \lambda_i g_i(x) = 0, \forall i = 1, \ldots, m_1 \\ g_i(x) \leq 0, \forall i = 1, \ldots, m_1 \\ h_i(x) = 0, \forall i = 1, \ldots, m_2 \\ \lambda_i \geq 0, \forall i = 1, \ldots, m_2 \end{cases}$$

2. *We say the optimization satisfies Slater's condition, if there exists an inner point x of $\mathcal{X}$ satisfying that:*

$$\begin{aligned} g_i(x) < 0, \forall i \in \{1, \ldots, m_1\} \\ h_i(x) = 0, \forall i \in \{1, \ldots, m_2\} \end{aligned}$$

   Namely, there is a inner point making the inequality constraints strictly feasible.

3. *If $f(x)$ is a convex function over a convex set $\mathcal{X}$, and the optimization satisfies Slater's condition, then KKT conditions are the necessary and sufficient condition for the optimal solution.*

Appendix C. Technical Details from Section 2

C.1. Preprocessing

We use W and $\mathbf{u}$ to denote the differentially privately released approximations of the sum of edge weights and the upper bound of each edge weight of $\hat{G}$ respectively. Specifically, we do the following. Recall ε is the parameter for (ε, δ)-DP, and β is some parameter for the success probability.

1. Set $W = \sum_{e\in\binom{V}{2}} \hat{\mathbf{w}}_e + \mathrm{Lap}(1/\varepsilon_1) + \log(3/\beta)/\varepsilon_1$. The term $\log(3/\beta)/\varepsilon_1$ is to guarantee $W \geq \sum_{e\in\binom{V}{2}} \hat{\mathbf{w}}_e$ with high probability;

2. Normalize the weights of $\hat{G}$ to obtain a graph $\overline{G} = (V, \overline{E}, \overline{\mathbf{w}})$ with the same vertex and edge sets as $\hat{G}$ (i.e., $\overline{E} = \hat{E}$), while the edge weights $\overline{G}$ of sum up to W. That is, $\overline{\mathbf{A}} = (W/\hat{W})\hat{\mathbf{A}}$, where $\overline{\mathbf{A}}$ and $\hat{\mathbf{A}}$ denote adjacency matrices of $\overline{G}$ and $\hat{G}$, respectively;

3. Set $\mathbf{u}_e = \overline{\mathbf{w}}_e + \mathrm{Lap}(1/\varepsilon_2) + \log(6n^2/\beta)/\varepsilon_2 + \frac{W}{\binom{n}{2}}$. The term $\log(6n^2/\beta)/\varepsilon_2$ is to guarantee $\mathbf{u} \geq \overline{\mathbf{w}}_e + \frac{W}{\binom{n}{2}}$ with high probability, since we need to guarantee that $\overline{\mathbf{w}}$ and $(\frac{W}{\binom{n}{2}})_{e\in\binom{V}{2}}$ fall in the domain $\mathcal{X}$ in update step.

4. Set $u_{\max} = \max_{e\in\binom{V}{2}} \mathbf{u}_e$. Recall that $\ell_3(\hat{G}) = \max_{(i,j)\in\binom{V}{2}} \sum_{s\in V\setminus\{i,j\}} \hat{\mathbf{w}}_{(i,s)}\hat{\mathbf{w}}_{(j,s)}$ denotes the local sensitivity of triangle-motif cuts of $\hat{G}$. We further define $\tilde{\ell}_3(\hat{G}) = \ell_3(\hat{G}) + u_{\max}(\mathrm{Lap}(1/\varepsilon_3) + \log(6n^2/\beta)/\varepsilon_3)$.

By Lemma 14 and Lemma 10, the released W, $\mathbf{u}_e$ and $\tilde{\ell}_3(\hat{G})$ are $\varepsilon_1, \varepsilon_2, \varepsilon_3$-differentially private respectively. By Lemma 15 and union bound, with probability at least $1 - \frac{\beta}{3} - \binom{n}{2}\frac{\beta}{3n^2} \geq 1 - 2\beta/3$, it holds that

$$\overline{\mathbf{w}}_e + \frac{W}{\binom{n}{2}} \leq \mathbf{u}_e \leq \overline{\mathbf{w}}_e + \frac{W}{\binom{n}{2}} + 2\log(6n^2/\beta)/\varepsilon_1$$

for any $e \in \binom{V}{2}$ and that

$$\sum_{e\in\binom{V}{2}} \hat{\mathbf{w}}_e \leq W \leq \sum_{e\in\binom{V}{2}} \hat{\mathbf{w}}_e + 2\log(3/\beta)/\varepsilon_2.$$

and that

$$\ell_3(\hat{G}) \leq \tilde{\ell}_3(\hat{G}) \leq \ell_3(\hat{G}) + 2u_{\max}\log(6n^2/\beta)/\varepsilon_3$$

For the sake of convenience, we further introduce some quantities used in our algorithm and analysis:

1. $U_\triangle = \max_{(i,j)\in\binom{V}{2}} \sum_{s\in V\setminus\{i,j\}} (\mathbf{u}_{(i,j)}\mathbf{u}_{(i,s)} + \mathbf{u}_{(i,s)}\mathbf{u}_{(j,s)} + \mathbf{u}_{(j,s)}\mathbf{u}_{(i,j)})$,

2. $U_\Lambda = \max_{(i,j)\in\binom{V}{2}} \sum_{s\in V\setminus\{i,j\}} (\mathbf{u}_{(i,s)} + \mathbf{u}_{(j,s)})$.

Note that $U_\triangle, U_\Lambda$ can be viewed as the maximum pairwise triangle importance and wedge importance, respectively, in the graph with edge weights given by $\mathbf{u}$.

Assume that $\hat{W} = \Omega(\frac{1/\beta}{\varepsilon})$ and $\ell_3(\hat{G}) = \Omega(\frac{w_{\max}\log^2(n/\beta)}{\varepsilon^2})$. We refer to this as the *non-degenerate* case. (When this assumption does not hold, we call it the *degenerate* case. We will provide a case analysis in Corollary 45.) Recall that $\ell_3(\overline{G}) = \max_{(i,j)\in\binom{V}{2}} \sum_{s\in V\setminus\{i,j\}} \overline{\mathbf{w}}_{(i,s)}\overline{\mathbf{w}}_{(j,s)}$ is the local sensitivity of triangle cuts in $\overline{G}$. Since $\hat{W} = \Omega(\frac{1/\beta}{\varepsilon})$, we have $\ell_3(\overline{G}) = (\frac{\overline{W}}{\hat{W}})^2\ell_3(\hat{G}) = \Theta(\ell_3(\hat{G}))$. Additionally, by the assumption $\ell_3(\hat{G}) = \Omega(\frac{w_{\max}\log^2(n/\beta)}{\varepsilon^2})$ and the fact $\tilde{\ell}_3(\hat{G}) \leq \ell_3(\hat{G}) + 2u_{\max}\log(6n^2/\beta)/\varepsilon_3 \leq \ell_3(\hat{G}) + O(w_{\max}\log^2(n/\beta)/\varepsilon_2\varepsilon_3)$, $\varepsilon_2 = \varepsilon_3 = \varepsilon/6$, it holds that $\tilde{\ell}_3(\hat{G}) = \Theta(\ell_3(\hat{G}))$.

C.2. Key Technical Differences between Our Approach and the EKKL Approach

Here we highlight and summarize the key technical differences between our approach and the EKKL approach (Eliáš et al., 2020):

- We introduce a convexity regularizer to ensure the optimization problem for privately releasing a graph preserving the triangle-motif cut structure is convex.
- After adding the convexity regularizer, the mirror descent step used by the EKKL approach becomes invalid as the gradient of our objective function depends on $\mathbf{w}$, due to the higher-order structure of the triangle-motif. Thus, we reformulate the problem of updating the descent, i.e., for minimizing the Bregman divergence, as a new convex optimization problem with appropriate constraints.
- We introduce a new greedy algorithm for the mirror descent update step. This greedy algorithm is guaranteed to output a solution that satisfies the KKT conditions and thus ensures valid updates. In contrast, the EKKL method uses a straightforward update rule for the descent step.
- Our lower bound requires proving the discrepancy of 3-colorings of h-uniform hypergraphs for any constant $h \geq 2$, while the work in (Eliáš et al., 2020) only proves the discrepancy of graphs which corresponds to $h = 2$.

Specifically, we have the following two main technical difference in our algorithm:

- For the *optimization problem*, the independent variable of our function is $\mathbf{w}$, while $\mathbf{A}_\triangle$ is used in Equation (2). However, $\begin{pmatrix} \mathbf{0} & \mathbf{A}_\triangle - \overline{\mathbf{A}}_\triangle \\ \mathbf{A}_\triangle - \overline{\mathbf{A}}_\triangle & \mathbf{0} \end{pmatrix} \bullet \mathbf{X}$ is neither private nor convex. Therefore, we use the regularizer $\lambda \log\det \mathbf{X}$ to control the stability and $\sum_{(i,j)\in\binom{V}{2}} 3(\mathbf{w}_{(i,j)} - \overline{\mathbf{w}}_{(i,j)})^2 \cdot \sum_{s\in V\setminus\{i,j\}}(\mathbf{u}_{(i,s)} + \mathbf{u}_{(j,s)})$ to control the convexity. Besides, we use a more restricted domain $\mathcal{X}$ compared to (Eliáš et al., 2020), which leads to a deliberately designed mirror update step described at Section 3.1.
- For the *mirror descent update step*, $\sum_e \mathbf{w}_e = W$ is the only constraint $\mathbf{w}$ the method proposed by Eliáš et al. (2020) needs to satisfy, hence the update step is simply $\mathbf{w}_e^{(t+1)} = \frac{W\mathbf{y}_e^{(t)}}{\sum_e \mathbf{y}_e^{(t)}}$, for $\forall e \in \binom{V}{2}$. However, in our setting, since we add constraints $\mathbf{w}_e \leq \mathbf{u}_e$ for $\forall e \in \binom{V}{2}$, we have to use a more complicated method (Algorithm 2) to update weights.

Appendix D. Deferred Proofs and Technical Details from Section 3

In the following, we prove the correctness of Algorithm 1. In Appendix D.1, we derive the SDP approximation of the triangle-motif cut difference. In Appendix D.2, we derive the gradient of $f_\triangle$ and prove its convexity. Note that in Section 3.1, we have proved the correctness of Algorithm 2 for mirror descent update step, where the deferred proofs is in Appendix D.3. Then we give privacy analysis in Appendix D.4 and determines the value of λ to guarantee the (ε, δ)-DP property. Appendix D.5 contains the utility analysis, in which we show the difference between $f_\triangle$ and cut norm, and use Corollary 24 to determine the value of T and bound the additive error. Finally, in Appendix D.6, we analyze the running time of the algorithm.

D.1. The SDP Approximation

Focusing on the special case when the motif is triangle, the motif adjacency matrix has a very direct relationship with motif size of cut in this case. Denote $\mathbf{1_S}$ as the indicative vector of the vertex set S, which has value one only in the coordinates corresponding to points in S. The following fact is a straightforward generalization of a result (for unweighted graphs) from (Benson et al., 2016).

Fact 26 *It holds that*

$$\mathrm{Cut}_\triangle^{(G)}(S, V \setminus S) = \frac{1}{2}\mathbf{1_S}^\top \mathbf{A}_\triangle \mathbf{1}_{V\setminus S}$$

Proof We note that for any cut of G, a triangle is either crossed zero times or twice. Thus

$$\begin{aligned}
\mathrm{Cut}_\triangle^{(G)}(S, V \setminus S) &= \sum_{I \in \mathcal{M}(G,\triangle): I \text{ crosses } (S, V\setminus S)} w(I) \\
&= \frac{1}{2} \sum_{(i,j)\in\binom{V}{2}:(i,j) \text{ crosses } (S,V\setminus S)} \sum_{k:(i,j,k)\in\mathcal{M}(G,\triangle)} w((i,j,k)) \\
&= \frac{1}{2} \sum_{(i,j)\in\binom{V}{2}:(i,j) \text{ crosses } (S,V\setminus S)} (\mathbf{A}_\triangle)_{i,j} \\
&= \frac{1}{2}\mathbf{1_S}^\top \mathbf{A}_\triangle \mathbf{1}_{V\setminus S}
\end{aligned}$$

■

This also implies that the triangle-motif cut of G is exactly half of the cut of the triangle-motif weighted graph of G. We note that this property also applies to any other 3-vertex motif (such as the length-2 path), allowing for easy generalization of our algorithm. However, for motifs with more than 3 vertices, there lacks a similar characterization (see e.g. (Benson et al., 2016)).

Inspired by the setting of edge cut (Eliáš et al., 2020) described in Appendix B.3, we have the following method to bound the additive error for triangle-motif cut. Consider graphs G_1 and G_2. Let $(\mathbf{A}_1)_\triangle$ and $(\mathbf{A}_2)_\triangle$ be their triangle adjacency matrices. Then for a fixed cut $(S, V \setminus S)$, the $\triangle$-motif size of G_1 is $\sum_{v\in S, u\in V\setminus S} \left((\mathbf{A}_1)_\triangle\right)_{u,v}$ and the $\triangle$-motif size of G_2 is $\sum_{v\in S, u\in V\setminus S} \left((\mathbf{A}_2)_\triangle\right)_{u,v}$. So we can see that the triangle-motif cut difference between G_1 and G_2 is twice of the following expression:

$$\max_{S\subset V}\left\{\left|\sum_{v\in S,u\in V\setminus S}\left((\mathbf{A}_1)_\triangle\right)_{u,v} - \sum_{v\in S,u\in V\setminus S}\left((\mathbf{A}_2)_\triangle\right)_{u,v}\right|\right\} = \max_{\mathbf{x},\mathbf{y}\in\{0,1\}^n, \mathbf{x}+\mathbf{y}=\mathbf{1}}\left\{\left|\mathbf{x}^\top\left((\mathbf{A}_1)_\triangle - (\mathbf{A}_2)_\triangle\right)\mathbf{y}\right|\right\}$$

According to (Frieze and Kannan, 1999), the cut on a graph can be bounded by the cut norm which is defined in Appendix B.3. Thus by an approximation in (Alon and Naor, 2004) and Lemma 22, we have the following lemma:

Lemma 27 *The triangle-motif cut difference between G_1 and G_2 can be bounded by the following SDP up to a constant factor:*

$$\max\left\{\begin{pmatrix} \mathbf{0} & (\mathbf{A}_2)_\triangle - (\mathbf{A}_1)_\triangle \\ (\mathbf{A}_2)_\triangle - (\mathbf{A}_1)_\triangle & \mathbf{0} \end{pmatrix} \bullet \mathbf{X} : \mathbf{X} \text{ is symmetric}, \mathbf{X} \succeq \mathbf{0}, \text{and } \mathbf{X}_{i,i} = 1 \text{ for } \forall i\right\} \tag{15}$$

To sum up, the triangle-motif cut difference between G_1 and G_2 can be bounded by Equation (15) up to a constant factor.

D.2. Gradient and Convexity

Recall $\mathcal{D}$ is the domain of $\mathbf{X}$, which is defined in Equation (5). We first state some of its useful properties.

Lemma 28 *It holds that*

1. *For any $\mathbf{X} \in \mathcal{D}$, we have $\mathbf{X}_{ij} \in [-1, 1]$ for any i and j.*
2. *For any $\mathbf{X} \in \mathcal{D}$ with eigenvalues $\lambda_1, \ldots, \lambda_{2n}$, we have $\lambda_i \in [\frac{1}{n}, 2n]$ for any i.*

Fact 29 *If $\mathbf{A} \in \mathbb{R}^{n\times n}$ is a semi-definite matrix, then there exists $\mathbf{x}_i \in \mathbb{R}^n, i = 1, \ldots, n$ s.t. $\mathbf{A} = \sum_{i=1}^n \mathbf{x}_i^\top \mathbf{x}$.*

Proof [Proof of Lemma 28]Since $\mathbf{X} \in \mathcal{D}$ is a semi-definite matrix, by Fact 29, there exist vectors $\mathbf{x}_1, \ldots, \mathbf{x}_{2n} \in \mathbb{R}^{2n}$ such that $\mathbf{X}_{i,j} = \mathbf{x}_i^\top \mathbf{x}_j$. Then by $\mathbf{X}_{ii} = 1$ and $\mathbf{X}_{i,i} = \mathbf{x}_i^\top \mathbf{x}_i$, we have $|\mathbf{x}_i| = 1$ for any $i \le n$. Therefore, $|\mathbf{X}_{i,j}| \le \sqrt{|\mathbf{x}_i||\mathbf{x}_j|} \le 1$.

Then by $\mathbf{X}_{i,i} = 1$ for $i = 1, \ldots, n$, we have $\sum_{i=1}^{2n} \lambda_i = \mathrm{tr}(\mathbf{X}) = \sum_{i=1}^{2n} \mathbf{X}_{i,i} = 2n$. Moreover, since $\mathbf{X} \succeq \frac{1}{n}\mathbf{I}_{2n}$, it holds that $(\mathbf{X} - \frac{1}{n}\mathbf{I}_{2n}) \succeq \mathbf{0}$. Thus we have $\lambda_i \ge \frac{1}{n}$. Therefore, we can conclude that $\lambda_i \in [\frac{1}{n}, 2n]$ for any i. ■

We next prove the convexity of $f_\triangle(\mathbf{w})$ and compute its gradient.

Lemma 30 *The function $f_\triangle(\mathbf{w})$ is convex and differentiable with respect to $\mathbf{w}$. Furthermore, for any fixed pair $e \in \binom{V}{2}$, it holds that*

$$\nabla f_\triangle(\mathbf{w})_e = \nabla_{\mathbf{w}} F_\triangle(\mathbf{w}, \mathbf{X}^*)_e = \begin{pmatrix} \mathbf{0} & \mathbf{D}_\triangle^{(e)} \\ \mathbf{D}_\triangle^{(e)} & \mathbf{0} \end{pmatrix} \bullet \mathbf{X}^* + 6 \sum_{s \in V\setminus\{i,j\}: e=(i,j)} (\mathbf{u}_{(i,s)} + \mathbf{u}_{(j,s)})(\mathbf{w}_e - \overline{\mathbf{w}}_e),$$

where $\mathbf{X}^*$ denotes the maximizer such that $F_\triangle(\mathbf{w}, \mathbf{X}^*) = \max_{\mathbf{X}\in\mathcal{D}} F_\triangle(\mathbf{w}, \mathbf{X})$, for some fixed $\mathbf{w}$.

To prove the above lemma, we need the following theorem.

Theorem 31 (Danskin's theorem (Danskin, 2012)) *Let $\mathcal{D} \in \mathbb{R}^m$ be a compact subset and $\phi : \mathbb{R}^n \times \mathcal{D} \to \mathbb{R}$ be a continuous function such that $\phi(\cdot, x)$ is convex for fixed $x \in \mathcal{D}$. Then the function $f : \mathbb{R}^n \to \mathbb{R}$ defined as $f(z) = \max_{x\in\mathcal{D}} \phi(z, x)$ is convex. If there is a unique maximizer x^* such that $\phi(z, x^*) = \max_{x\in\mathcal{D}} \phi(z, x)$ and that $\phi(z, x^*)$ is differentiable at z, then f is differentiable at z and $\nabla f(z) = \nabla_z \phi(z, x^*) = \left(\frac{\partial \phi(z,x^*)}{\partial z_i}\right)_{i=1}^n$.*

We will also need the following useful property of a semi-definite matrix.

Fact 32 *A symmetric matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ with non-negative diagonal entries is semi-definite if $\mathbf{A}$ is diagonally dominant, i.e.,*

$$\mathbf{A}_{i,i} \geq \sum_{j\neq i} |\mathbf{A}_{i,j}|, \quad \textit{for any } 1 \leq i \leq n.$$

Now we prove Lemma 30.

Proof [Proof of Lemma 30] Recall in Fact 19, we denote $\mathbf{D}_{\triangle}^{(k,\ell)}$ as the derivative of $\mathbf{A}_{\triangle}$ at $\mathbf{w}_{(k,\ell)}$ and $\mathbf{E}_{\triangle}^{((i,j),(k,\ell))}$ as the second-order derivative of $\mathbf{A}_{\triangle}$ at $\mathbf{w}_{(i,j)}$ and $\mathbf{w}_{(k,\ell)}$. Therefore, for any fixed $\mathbf{X} \in \mathcal{D}$, we have that

$$\nabla_{\mathbf{w}} F_{\triangle}(\mathbf{w}, \mathbf{X})_e = \begin{pmatrix} \mathbf{0} & \mathbf{D}_{\triangle}^{(e)} \\ \mathbf{D}_{\triangle}^{(e)} & \mathbf{0} \end{pmatrix} \bullet \mathbf{X} + 6 \sum_{s\in V\setminus\{i,j\}:e=(i,j)} (\mathbf{u}_{(i,s)} + \mathbf{u}_{(j,s)})(\mathbf{w}_e - \overline{\mathbf{w}}_e),$$

and the second-order partial derivatives are

$$\begin{aligned} \nabla^2_{\mathbf{w}} F_{\triangle}(\mathbf{w}, \mathbf{X})_{(i,j),(k,\ell)} &= \begin{pmatrix} \mathbf{0} & \mathbf{E}_{\triangle}^{((i,j),(k,\ell))} \\ \mathbf{E}_{\triangle}^{((i,j),(k,\ell))} & \mathbf{0} \end{pmatrix} \bullet \mathbf{X} + 6 \sum_{s\in V\setminus\{i,j\}} (\mathbf{u}_{(i,s)} + \mathbf{u}_{(j,s)}) \cdot \mathbf{1}_{(i,j)=(k,\ell)} \\ &= \begin{cases} 6\sum_{s\in V\setminus\{i,j\}} (\mathbf{u}_{(i,s)} + \mathbf{u}_{(j,s)}), & (i,j) = (k,\ell) \\ \mathbf{w}_{(j,\ell)}(\sum_{i',j'\in\{i,j,\ell\}} \mathbf{X}_{i',n+j'} + \sum_{i',j'\in\{i,j,\ell\}} \mathbf{X}_{n+i',j'}), & i = k \text{ and } j \neq \ell \\ 0, & o.w. \end{cases} \end{aligned}$$

Then for any $(i,j) \in \binom{V}{2}$, it holds that,

$$\begin{aligned} \sum_{(k,\ell)\neq(i,j)} |\nabla^2_{\mathbf{w}} F_{\triangle}(\mathbf{w}, \mathbf{X})_{(i,j),(k,\ell)}| &= \sum_{s\in V\setminus\{i,j\}} \left(|\nabla^2_{\mathbf{w}} F_{\triangle}(\mathbf{w}, \mathbf{X})_{(i,j),(i,s)}| + |\nabla^2_{\mathbf{w}} F_{\triangle}(\mathbf{w}, \mathbf{X})_{(i,j),(j,s)}|\right) \\ &= \sum_{s\in V\setminus\{i,j\}} \left(\mathbf{w}_{(i,s)} + \mathbf{w}_{(j,s)}\right) | \sum_{i',j'\in\{i,j,s\}} \mathbf{X}_{i',n+j'} + \sum_{i',j'\in\{i,j,s\}} \mathbf{X}_{n+i',j'}| \\ &\leq 6 \sum_{s\in V\setminus\{i,j\}} \left(\mathbf{u}_{(i,s)} + \mathbf{u}_{(j,s)}\right) = \nabla^2_{\mathbf{w}} F_{\triangle}(\mathbf{w}, \mathbf{X})_{(i,j),(i,j)}, \end{aligned}$$

where the second to last inequality follows from the fact that $\mathbf{w}_e \leq \mathbf{u}_e$ and that $|\mathbf{X}_{i,j}| \leq 1$.

Thus, by Fact 32, for fixed $\mathbf{X} \in \mathcal{D}$, $\nabla^2_{\mathbf{w}} F_{\triangle}(\mathbf{w}, \mathbf{X})$ is semi-definite, so $F_{\triangle}(\mathbf{w}, \mathbf{X})$ is convex. By Theorem 31, $f_{\triangle}(\mathbf{w}) = \max_{\mathbf{X}\in\mathcal{D}} F_{\triangle}(\mathbf{w}, \mathbf{X})$ is convex with respect to $\mathbf{w}$.

Moreover, for any fixed $\mathbf{X} \in \mathcal{D}$, it can be verified that $F_\triangle(\mathbf{w}, \mathbf{X})$ is differentiable at $\mathbf{w}$. Therefore, by Theorem 31, $f_\triangle(\mathbf{w})$ is differentiable. Furthermore,

$$\nabla f_\triangle(\mathbf{w})_e = \nabla_\mathbf{w} F_\triangle(\mathbf{w}, \mathbf{X}^*)_e = \begin{pmatrix} \mathbf{0} & \mathbf{D}_\triangle^{(e)(t)} \\ \mathbf{D}_\triangle^{(e)(t)} & \mathbf{0} \end{pmatrix} \bullet \mathbf{X}^* + 6 \sum_{s \in V \setminus \{i,j\} : e=(i,j)} (\mathbf{u}_{(i,s)} + \mathbf{u}_{(j,s)})(\mathbf{w}_e^{(t)} - \overline{\mathbf{w}}_e),$$

where $\mathbf{X}^*$ denotes the maximizer such that $F_\triangle(\mathbf{w}, \mathbf{X}^*) = \max_{\mathbf{X} \in \mathcal{D}} F_\triangle(\mathbf{w}, \mathbf{X})$. ■

D.3. Deferred Proofs of Theorem 5

Now we prove the remaining case that $k = 0$ for the proof of Theorem 5.

When $k = 0$, we have $\lambda_e = 0$ for $\forall e \in \binom{V}{2}$. Thus, by Equation (13),

$$\begin{aligned} \mathbf{w}_e^* &= \mathbf{y}_e^{(t+1)} \exp(-\mu), \text{ for } \forall e \in \binom{V}{2}, \\ W &= \sum_{e \in \binom{V}{2}} \mathbf{w}_e^* = \sum_{e \in \binom{V}{2}} \mathbf{y}_e^{(t+1)} \exp(-\mu) = S \cdot \exp(-\mu), \end{aligned}$$

which further imply that $\exp(-\mu) = \frac{W}{S}$. Thus, the optimal solution is

$$\mathbf{w}_e^* = \frac{W \cdot \mathbf{y}_e^{(t+1)}}{S} = \exp(-\mu) \cdot \mathbf{y}_e^{(t+1)}, \text{ for } \forall e \in \binom{V}{2}.$$

Furthermore, by the above, for any subset $F \subseteq \binom{V}{2}$ of edges, it holds that

$$\frac{\sum_{e \in F} \mathbf{w}_e^*}{\sum_{e \in F} \mathbf{y}_e^{(t+1)}} = \exp(-\mu).$$

Since there is no coordinate satisfying the equality condition of Equation (11), we have $\mathbf{w}_e^* < \mathbf{u}_e$ for any e. Let $e_1, e_2, \ldots, e_N$ be the ordered sequence of edges given in Algorithm 2. Note that by the above discussion, $W_1 = W$, $S_1 = S$, and

$$\mathbf{w}_{e_1}^{(t+1)} = \frac{W_1 \cdot \mathbf{y}_{e_1}^{(t+1)}}{S_1} = \frac{W \cdot \mathbf{y}_{e_1}^{(t+1)}}{S} = \exp(-\mu) \cdot \mathbf{y}_{e_1}^{(t+1)} = \mathbf{w}_{e_1}^*.$$

Then by induction we have that $W_{i+1} = W_1 - \sum_{\ell=1}^{i} \mathbf{w}_{e_\ell}^{(t+1)}$ and $S_{i+1} = S_1 - \sum_{\ell=1}^{i} \mathbf{y}_{e_\ell}^{(t+1)}$, which further implies that $W_{i+1} = S_{i+1} = \exp(-\mu)$. Thus, $\mathbf{w}_{e_{i+1}}^{(t+1)} = \frac{W_{i+1} \cdot \mathbf{y}_{e_i}^{(t+1)}}{S_{i+1}} = \exp(-\mu) \cdot \mathbf{y}_{e_i}^{(t+1)} = \mathbf{w}_{e_{i+1}}^*$. Therefore, the output solution $\mathbf{w}^{(t+1)}$ of Algorithm 2 is indeed the optimum solution $\mathbf{w}^*$.

D.4. Privacy Analysis

The privacy analysis follows from a similar approach to (Eliáš et al., 2020). We will first bound the privacy loss in each inner iteration when computing the noisy gradient $\mathbf{g}^{(t)}$, and apply the advanced composition (Lemma 11) over all steps from Line 3 to Line 11 in Algorithm 1. Then, we consider the L composition in the outer iteration, hence giving the total privacy guarantee for Algorithm 1.

For any fixed outer iteration, recall that we treat $\overline{G}$ as the input graph with public W and $\mathbf{u}$ first. Then, we denote $\overline{G}'$ as some neighboring graph of $\overline{G}$, which differs from $\overline{G}$ by one edge. Note that, for the sake of convenience, we first consider two graphs to be neighboring if they differ by exactly one edge. The more standard situation based on the ℓ_1 norm will be discussed later. And denote $G^{(t)}$ as our solution at step t. Let $(\overline{\mathbf{X}}^{(t)})$ and $\overline{\mathbf{X}}'^{(t)}$ be the maximizer of $F_\triangle(\mathbf{w}^{(t)}, \mathbf{X})$ corresponding to $\overline{G}$ and $\overline{G}'$ respectively.

The proofs of the following two lemmas are almost identical to those in (Eliáš et al., 2020), so we skip them.

Lemma 33 *Let $H(\mathbf{M}) = \max_{\mathbf{X}\in\mathcal{D}} \mathbf{M} \bullet \mathbf{X} + \lambda \log\det(\mathbf{X}) + S(\mathbf{M})$, where $S(\mathbf{M})$ is a function dependent only on $\mathbf{M}$. For two matrices $\mathbf{M}$ and $\mathbf{M}'$, we denote $\mathbf{X}^*$ as the maximizer of $H(\mathbf{M})$ and $\mathbf{X}'^*$ as the maximizer of $H(\mathbf{M}')$. Then we have*

$$\|(\mathbf{X}^*)^{-\frac{1}{2}}(\mathbf{X}'^* - \mathbf{X}^*)(\mathbf{X}^*)^{-\frac{1}{2}}\|_F \leq \frac{32}{\lambda}\|(\mathbf{X}^*)^{\frac{1}{2}}(\mathbf{M}' - \mathbf{M})(\mathbf{X}^*)^{\frac{1}{2}}\|_F$$

Intuitively speaking, the above lemma measures the difference between $\mathbf{X}'^*$ and $\mathbf{X}^*$.

Lemma 34 ((Eliáš et al., 2020)) *Let $\overline{\delta}$ a fixed parameter and $\mathbf{X}, \mathbf{X}' \in \mathbb{R}^{2n\times 2n}$ be symmetric positive definite matrices s.t. $\|\mathbf{X}^{-\frac{1}{2}}(\mathbf{X}' - \mathbf{X})\mathbf{X}^{-\frac{1}{2}}\|_F < \frac{1}{2}$. Denote $\mathrm{pdf}_{\mathbf{X}}(\mathbf{x})$ and $\mathrm{pdf}_{\mathbf{X}'}(\mathbf{x})$ the probability density functions of $N(\mathbf{0}, \mathbf{X})$ and $N(\mathbf{0}, \mathbf{X}')$ respectively. Let $\overline{\varepsilon} = O(\log\frac{1}{\overline{\delta}} \cdot \|\mathbf{X}^{-\frac{1}{2}}(\mathbf{X}' - \mathbf{X})\mathbf{X}^{-\frac{1}{2}}\|_F)$. Then we have*

$$\mathrm{pdf}_{\mathbf{X}}(\mathbf{x}) \leq e^{\overline{\varepsilon}}\mathrm{pdf}_{\mathbf{X}'}(\mathbf{x})$$

with probability at least $(1 - \overline{\delta})$ over $\mathbf{x} \in N(\mathbf{0}, \mathbf{X})$.

Recall that $\zeta \sim N(\mathbf{0}, \mathbf{I}_{2n})$ is defined in Algorithm 1. Therefore, $(\mathbf{X}^{(t)})^{\frac{1}{2}}\zeta \sim N(\mathbf{0}, \mathbf{X}^{(t)})$, $(\overline{\mathbf{X}}^{(t)})^{\frac{1}{2}}\zeta \sim N(\mathbf{0}, \overline{\mathbf{X}}^{(t)})$. We will instantiate Lemma 34 with $\mathbf{X}^* = \overline{\mathbf{X}}^{(t)}$, $\mathbf{X}'^* = \overline{\mathbf{X}}'^{(t)}$. Now, let us analyze the privacy guarantee of the algorithm.

Theorem 35 (Privacy Guarantee) *Algorithm 1 with parameter $\lambda = \Theta(\varepsilon^{-1})\tilde{\ell}_3(\hat{G})\sqrt{T}\log^{\frac{3}{2}}(\frac{T}{\delta})$, $\varepsilon_0 = O(\frac{\varepsilon}{\sqrt{T\log\frac{T}{\delta}}})$ is (ε, δ)-differentially private.*

Proof For some fixed outer iteration and some fixed inner iteration t, assume $\overline{\mathbf{G}}'$ differs from $\overline{\mathbf{G}}$ in edge $e = (i, j)$ (has one more weight in e). As mentioned above, $\overline{\mathbf{D}}_\triangle^{(e)}$ is actually the divergence of motif adjacency matrix between neighboring graphs $\overline{\mathbf{G}}$ and $\overline{\mathbf{G}}'$, i.e., $\overline{\mathbf{A}}'_\triangle - \overline{\mathbf{A}}_\triangle = \overline{\mathbf{D}}_\triangle^{(e)}$.

Now let $\mathbf{M} = \begin{pmatrix} \mathbf{0} & \mathbf{A}_\triangle^{(t)} - \overline{\mathbf{A}}_\triangle \\ \mathbf{A}_\triangle^{(t)} - \overline{\mathbf{A}}_\triangle & \mathbf{0} \end{pmatrix}$, and $\mathbf{M}' = \begin{pmatrix} \mathbf{0} & \mathbf{A}_\triangle^{(t)} - \overline{\mathbf{A}}'_\triangle \\ \mathbf{A}_\triangle^{(t)} - \overline{\mathbf{A}}'_\triangle & \mathbf{0} \end{pmatrix}$.

Then it holds that that

$$\mathbf{M}' - \mathbf{M} = \begin{pmatrix} \mathbf{0} & \overline{\mathbf{A}}_\triangle - \overline{\mathbf{A}}'_\triangle \\ \overline{\mathbf{A}}_\triangle - \overline{\mathbf{A}}'_\triangle & \mathbf{0} \end{pmatrix}.$$

We define $\ell_3(\overline{G}) = \max_{(i,j)\in\binom{V}{2}} \sum_{s\in V\setminus\{i,j\}}(\overline{\mathbf{w}}_{(i,j)}\overline{\mathbf{w}}_{(i,s)} + \overline{\mathbf{w}}_{(i,s)}\overline{\mathbf{w}}_{(j,s)} + \overline{\mathbf{w}}_{(j,s)}\overline{\mathbf{w}}_{(i,j)})$ as the maximum triangle-motif cut difference of $\overline{G}$. Therefore,

$$\left\|\mathbf{M}' - \mathbf{M}\right\|_1 = 2\left\|\overline{\mathbf{D}}_{\triangle}^{(e)}\right\|_1 = 4(\sum_{\ell\neq i,j}\overline{\mathbf{w}}_{(i,\ell)}\overline{\mathbf{w}}_{(\ell,j)}) \leq 4\ell_3(\overline{G}).$$

Denote $\mathbf{M}' - \mathbf{M} = \sum_{i=1}^{(2n)^2} c_i\mathbf{E}_i$, where E_i has only one single non-zero-entry equal to 1. Then we have

$$\sum_{i,j} c_i c_j = (\sum_i c_i)^2 \leq (\sum_i |c_i|)^2 \leq (\left\|\mathbf{M}' - \mathbf{M}\right\|_1)^2 = 16\ell_3(\overline{G})^2.$$

We now instantiate Lemma 33 with $\mathbf{X}^* = \overline{\mathbf{X}}^{(t)}$, $\mathbf{X}'^* = \overline{\mathbf{X}}'^{(t)}$, $\mathbf{M}, \mathbf{M}'$ defined as above, and $H(\mathbf{M}) = f_{\triangle}(\mathbf{w})$, $S(\mathbf{M}) = 3\sum_{s\in V\setminus\{i,j\}}(\mathbf{u}_{(i,s)} + \mathbf{u}_{(j,s)})(\mathbf{w}_e^{(t)} - \mathbf{w}_e)^2$.

Then it holds that,

$$\begin{aligned}
\left\|(\overline{\mathbf{X}}^{(t)})^{\frac{1}{2}}(\mathbf{M}' - \mathbf{M})(\overline{\mathbf{X}}^{(t)})^{\frac{1}{2}}\right\|_F^2 &= \mathrm{tr}\left((\overline{\mathbf{X}}^{(t)})^{\frac{1}{2}}(\mathbf{M}' - \mathbf{M})(\overline{\mathbf{X}}^{(t)})^{\frac{1}{2}}(\overline{\mathbf{X}}^{(t)})^{\frac{1}{2}}(\mathbf{M}' - \mathbf{M})(\overline{\mathbf{X}}^{(t)})^{\frac{1}{2}}\right) \\
&= \mathrm{tr}\left(\overline{\mathbf{X}}^{(t)}(\sum_{i=1}^{(2n)^2} c_i\mathbf{E}_i)\overline{\mathbf{X}}^{(t)}(\sum_{i=1}^{(2n)^2} c_i\mathbf{E}_i)\right) \\
&= \sum_{i,j} c_i c_j \cdot \mathrm{tr}(\overline{\mathbf{X}}^{(\mathrm{t})}\mathbf{E}_{\mathrm{i}}\overline{\mathbf{X}}^{(\mathrm{t})}\mathbf{E}_{\mathrm{j}}) \\
&= \sum_{i,j} c_i c_j (\overline{\mathbf{X}}^{(t)}\mathbf{E}_i)^{\top} \bullet (\overline{\mathbf{X}}^{(t)}\mathbf{E}_j) \\
&\leq 16\ell_3(\overline{G})^2
\end{aligned}$$

Therefore, by Lemma 33, we have

$$\|(\overline{\mathbf{X}}^{(t)})^{-\frac{1}{2}}(\overline{\mathbf{X}}'^{(t)} - \overline{\mathbf{X}}^{(t)})(\overline{\mathbf{X}}^{(t)})^{-\frac{1}{2}}\|_F \leq \frac{32}{\lambda}\left\|(\overline{\mathbf{X}}^{(t)})^{\frac{1}{2}}(\mathbf{M}' - \mathbf{M})(\overline{\mathbf{X}}^{(t)})^{\frac{1}{2}}\right\|_F \leq \frac{32}{\lambda}\cdot 4\ell_3(\overline{G}) = \frac{128}{\lambda}\ell_3(\overline{G}).$$

Recall ζ is distributed as $N(0, \mathbf{I}_{d_{\mathbf{M}}})$. By Lemma 34 and Definition 16, we can say that, the release of $(\mathbf{X}^{(t)})^{\frac{1}{2}}\zeta$ in each mirror descent step preserves $(\overline{\varepsilon}, \overline{\delta})$-differential privacy, where $\overline{\varepsilon} = O(\frac{1}{\lambda}\ell_3(\overline{G})\log\frac{1}{\delta})$.

We choose $\overline{\delta} = \frac{\delta}{2T}$. Then by advanced composition (Lemma 12), the privacy guarantee of T mirror descent steps from Line 3 to Line 11 in Algorithm 1 is $\left(\overline{\varepsilon}\cdot 4\sqrt{T\log\frac{4}{\delta}}, \delta\right)$. Additionally, by Line 6 in Algorithm 1, we $(\varepsilon_4, 0)$-differentially privately release ν each outer iteration ℓ. To sum up, by composition (Lemma 11), the total privacy guarantee in the update process is $\left(L\overline{\varepsilon}\cdot 4\sqrt{T\log\frac{4}{\delta}} + L\varepsilon_4, \delta\right)$.

Recall that the release of W, $\mathbf{u}$, and $\tilde{\ell}_3(\hat{G})$ is $(\varepsilon_1 + \varepsilon_2 + \varepsilon_3)$-differentially private according to Appendix C.1. Since $L = \log_3(\frac{3}{\beta})$, $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = \frac{\varepsilon}{6}$ and $\varepsilon_4 = \frac{\varepsilon}{6L}$, by adaptive composition (Lemma 11), Algorithm 1 is $\left(2\varepsilon/3 + L\overline{\varepsilon}\cdot 4\sqrt{T\log\frac{4}{\delta}}, \delta\right)$-differentially private, where $\overline{\varepsilon} = O(\frac{1}{\lambda}\ell_3(\overline{G})\log\frac{2T}{\delta})$.

Additionally, recall that $\ell_3(\overline{G}) = \Theta(\ell_3(\hat{G}))$ and $\tilde{\ell}_3(\hat{G})$ is differentially privately released from $\ell_3(\hat{G})$. Therefore, when we set $\lambda = \Theta(\varepsilon^{-1})\tilde{\ell}_3(\hat{G})\sqrt{T}\log^{\frac{3}{2}}(\frac{T}{\delta})\log(\frac{3}{\beta})$, we can guarantee that the algorithm is (ε,δ)-DP. ■

Now we consider the standard notion of edge privacy (Definition 9), that two graphs are called neighboring if the two vectors corresponding to their edge weights differ by at most 1 in the ℓ_1 norm. We first consider the graph $\overline{G}$ and its neighboring graph. Let $e_1, e_2, \ldots, e_{\binom{n}{2}}$ denote an ordering of all possible edges. Denote $\overline{G}'$ (with edge weight function $\overline{\mathbf{w}}'$) as a neighboring graph of $\overline{G}$ (with edge weight function $\overline{\mathbf{w}}$). Then we denote $G_0 = \overline{G}$, and for $i = 1, \ldots, \binom{n}{2}$, we let G_i denote the graph generated by replacing the weight of edge e_i in G_{i-1} with $\overline{\mathbf{w}}'_{e_i}$. It follows that $G_{\binom{n}{2}} = \overline{G}'$. Note that G_{i-1} and G_i differ only by $\left|\overline{\mathbf{w}}'_{e_i} - \overline{\mathbf{w}}_{e_i}\right|$ at e_i entry, and $\sum_{i=1}^{\binom{n}{2}}\left|\overline{\mathbf{w}}'_{e_i} - \overline{\mathbf{w}}_{e_i}\right| = 1$. We let $\mathbf{A}_{\triangle,i}$ denote the adjacency matrix of triangle-motif graph of G_i. We have that $\|\mathbf{A}_{\triangle,i} - \mathbf{A}_{\triangle,i-1}\|_1 = \left|\overline{\mathbf{w}}'_{e_i} - \overline{\mathbf{w}}_{e_i}\right| \cdot \left\|\mathbf{D}_{\triangle,i}^{(e_i)}\right\|_1 \leq \left|\overline{\mathbf{w}}'_{e_i} - \overline{\mathbf{w}}_{e_i}\right| \cdot \ell_3(G_i)$.In the non-degenerate case (see Corollary 45), we have that $\ell_3(\overline{G}) = \Theta(\ell_3(\hat{G})) = \Omega(\frac{w_{\max}\log^2(n/\beta)}{\varepsilon^2})$; furthermore, we have $\left|\ell_3(G_i) - \ell_3(\overline{G})\right| \leq w_{\max}$, which implies that $\ell_3(G_i) = \Theta(\ell_3(\overline{G}))$. Thus $\left\|\overline{\mathbf{A}}_\triangle - \overline{\mathbf{A}}'_\triangle\right\|_1 \leq \sum_{i=1}^{\binom{n}{2}}\|\mathbf{A}_{\triangle,i} - \mathbf{A}_{\triangle,i-1}\|_1 \leq O(\sum_{i=1}^{\binom{n}{2}}\left|\overline{\mathbf{w}}'_{e_i} - \overline{\mathbf{w}}_{e_i}\right| \cdot \ell_3(\hat{G})) \leq O(\ell_3(\hat{G}))$. The remaining proof follows similarly to the analysis in Theorem 35.

Now let us recall that the real input graph is $\hat{G}$ instead of $\overline{G}$. While the distance between the actual input graph $\hat{G}$ and its neighboring graph is at most 1 (in terms of edge differences), the distance between the corresponding rescaled neighboring graph and $\overline{G}$ is not necessarily 1. Since in the non-degenerate case it holds that $W = \Omega(\frac{\log(1/\beta)}{\varepsilon})$ and $\overline{G} = \frac{W}{\hat{W}}\hat{G}$, it follows that $\overline{G}$ is a scaled version of $\hat{G}$ by a constant factor. Thus the privacy loss in each inner iteration of Algorithm 1 is still $O(\frac{\ell_3(\overline{G})}{\lambda})$.

Thus, our algorithm is (ε,δ)-DP under the standard notion of edge DP (Definition 9).

D.5. Utility Analysis

To analyze the utility, we first compute the triangle-motif cut difference between $\overline{G}$ and the original input graph $\hat{G}$.

Lemma 36 *Given a graph $\hat{G}$ with edge weights $\hat{W}$s.t. $\hat{W} = \sum_{e\in\binom{V}{2}}\hat{\mathbf{w}}_e$. Let $W = \hat{W} + \mathrm{Lap}(3/\varepsilon) + 3\log(3/\beta)/\varepsilon$ and $\overline{\mathbf{w}} = \frac{W}{\hat{W}}\hat{\mathbf{w}}$. Assume that $\hat{W} = \Omega(\frac{\log(1/\beta)}{\varepsilon})$. Then we have*

$$|\mathrm{Cut}_\triangle^{(\overline{G})}(S,T) - \mathrm{Cut}_\triangle^{(\hat{G})}(S,T)| \leq O\left(\ell_3(\hat{G})\frac{\log(1/\beta)}{\varepsilon}\right) \text{ for all } S,T \subseteq V$$

with probability at least $(1-\beta/3)$.

Proof Denote $Y = \mathrm{Lap}(3/\varepsilon) + 3\log(3/\beta)/\varepsilon$. Then by Lemma 15, we have $0 < Y < 6\log(3/\beta)/\varepsilon$ with probability at least $(1-\beta/3)$. Note that $\hat{W} = \Omega(\frac{\log(1/\beta)}{\varepsilon}) = \Omega(Y)$. Thus we have $W = \Theta(\hat{W})$ and $\frac{Y}{W} = O(1)$. Therefore,

$$\mathrm{Cut}_\triangle^{(\overline{G})}(S,T) = \left(\frac{W+Y}{W}\right)^3 \mathrm{Cut}_\triangle^{(\hat{G})}(S,T) \geq \mathrm{Cut}_\triangle^{(\hat{G})}(S,T),$$

and

$$
\begin{aligned}
\mathrm{Cut}_{\triangle}^{(\overline{G})}(S,T) &= \left(\frac{W+Y}{W}\right)^3 \mathrm{Cut}_{\triangle}^{(\hat{G})}(S,T) \\
&\leq \mathrm{Cut}_{\triangle}^{(\hat{G})}(S,T) + \left(3\left(\frac{Y}{W}\right) + 3\left(\frac{Y}{W}\right)^2 + \left(\frac{Y}{W}\right)^3\right)\mathrm{Cut}_{\triangle}^{(\hat{G})}(S,T) \\
&\leq \mathrm{Cut}_{\triangle}^{(\hat{G})}(S,T) + 7\frac{Y}{W}\mathrm{Cut}_{\triangle}^{(\hat{G})}(S,T) \\
&\leq \mathrm{Cut}_{\triangle}^{(\hat{G})}(S,T) + 7Y\ell_3(\overline{G}) \\
&\leq \mathrm{Cut}_{\triangle}^{(\hat{G})}(S,T) + O\left(\ell_3(\hat{G})\frac{\log(1/\beta)}{\varepsilon}\right)
\end{aligned}
$$

The last inequality is derived from the following fact:

$$
\begin{aligned}
W\ell_3(\overline{G}) &= \sum_{e\in\binom{V}{2}} \overline{\mathbf{w}}_e \max_{(i,j)\in\binom{V}{2}} \sum_{s\in V\setminus\{i,j\}} (\overline{\mathbf{w}}_{(i,j)}\overline{\mathbf{w}}_{(i,s)} + \overline{\mathbf{w}}_{(i,s)}\overline{\mathbf{w}}_{(j,s)} + \overline{\mathbf{w}}_{(j,s)}\overline{\mathbf{w}}_{(i,j)}) \\
&\geq \sum_{(i,j)\in\binom{V}{2}} \overline{\mathbf{w}}_{(i,j)} \sum_{s\in V\setminus\{i,j\}} \overline{\mathbf{w}}_{(i,s)}\overline{\mathbf{w}}_{(j,s)} \\
&=(\frac{W+Y}{W})^3 \sum_{(i,j,k)\in\binom{V}{3}} \hat{\mathbf{w}}_{(i,j)}\hat{\mathbf{w}}_{(j,k)}\hat{\mathbf{w}}_{(k,i)} \geq \mathrm{Cut}_{\triangle}^{(\hat{G})}(S,T)
\end{aligned}
$$

■

Then to analyze how well G approximates the triangle-motif cut of $\overline{G}$, we need to check the requirements and parameters in the mirror descent (Corollary 24). First, we will prove $\mathbf{g}^{(t)}$ is an unbiased approximation of the gradient for $f_{\triangle}(\mathbf{w}^{(t)})$, and then compute the parameter B by bounding $\mathbb{E}_{\{\gamma_\ell, R_{l,t}\}}[\left\|g^{(t)}\right\|_\infty^2]$. Recall that $B^2 \geq \mathbb{E}_{\{\gamma_\ell, R_{l,t}\}}[\left\|g^{(t)}\right\|_*^2]$ and $\|\cdot\|_*$ is l_∞ norm. Note that, in our algorithm, we let γ_ℓ be $\nu_e^{(\ell)}$, and let $R_{\ell,t}$ be ζ.

Lemma 37 *Let $\mathbf{g}^{(t)}$ be the estimated gradient in Algorithm 1. Then we have, $\mathbb{E}_{\{\zeta,\nu_e^{(\ell)}\}}[\mathbf{g}^{(t)}] = \nabla f_{\triangle}(\mathbf{w}^{(t)})$ and $\mathbb{E}_{\{\zeta,\nu_e^{(\ell)}\}}[\left\|\mathbf{g}^{(t)}\right\|_\infty^2] = O\left((U_\triangle + U_\Lambda L/\varepsilon)^2 \log^2 n\right)$.*

As a directly corollary, we can have $B = \Theta\left((U_\triangle + U_\Lambda L/\varepsilon)\log n\right)$.

Denote $\|\mathbf{A}\|_{\mathrm{op}}$ as the operator norm of some matrix $\mathbf{A}$, i.e., $\|\mathbf{A}\|_{\mathrm{op}} = \sup_{\|\mathbf{x}\|_2=1}\|\mathbf{A}\mathbf{x}\|_2$. For symmetric matrix, there is a useful property: $\|\mathbf{A}\|_{\mathrm{op}} \leq \|\mathbf{A}\|_F \leq \|\mathbf{A}\|_1$. Then we have the following concentration inequality, which is needed in the proof of Lemma 37.

Theorem 38 (Hanson-Wright Theorem (Rudelson and Vershynin, 2013)) *Let $\mathbf{A}$ be an $n\times n$ matrix with entries $a_{i,j}$. If $X_1,\ldots,X_n$ are mean zero, variance one random variables with sub-Guassian tail decay, i.e., for any $t>0$ we have $\Pr[|X_i|\geq t] \leq 2\exp(-\frac{t^2}{K^2})$ for some $K>0$,* *then*

$$\Pr\left[\left|\mathrm{tr}(\mathbf{A}) - \sum_{i,j=1}^{n} a_{i,j} X_i X_j\right| \geq t\right] \leq 2\exp\left(-\min\left(\frac{t^2}{CK^4 \|\mathbf{A}\|_F^2}, \frac{t}{CK^2 \|\mathbf{A}\|_{\mathrm{op}}}\right)\right)$$

where C is some universal positive constant.

Proof [Proof of Lemma 37] Recall that,

$$\mathbf{g}_e^{(t)} = \left((\mathbf{X}^{(t)})^{\frac{1}{2}}\zeta\zeta^\top(\mathbf{X}^{(t)})^{\frac{1}{2}}\right) \bullet \begin{pmatrix} \mathbf{0} & \mathbf{D}_\triangle^{(e)(t)} \\ \mathbf{D}_\triangle^{(e)(t)} & \mathbf{0} \end{pmatrix} + 6 \sum_{s\in V\setminus\{i,j\}} (\mathbf{u}_{(i,s)} + \mathbf{u}_{(j,s)})(\mathbf{w}_e^{(t)} - \tilde{\mathbf{w}}_e^{(\ell)})$$

Since $\zeta \sim N(\mathbf{0}, \mathbf{I}_{2n})$, for any i, $\zeta_i\zeta_i$ follows the chi–squared distribution with expectation equal to 1, and for any $i \neq j$, $\zeta_i\zeta_j$ follows the product normal distribution with expectation equal to 0, i.e., $\mathbb{E}[\zeta\zeta^\top] = \mathbf{I}_{2n}$. Moreover, recall that $\tilde{\mathbf{w}}_e^{(\ell)} = \overline{\mathbf{w}}_e + \nu_e^{(\ell)}$, and $\nu_e^{(\ell)}$ is distributed from $\mathrm{Lap}(1/\varepsilon_4)$ which has expectation 0. Denote $e = (i,j)$. Therefore,

$$\begin{aligned}
\mathbb{E}_{\{\zeta,\nu_e^{(\ell)}\}}\left[\mathbf{g}_e^{(t)}\right] &= \mathbb{E}_{\{\zeta,\nu_e^{(\ell)}\}}\left[\left((\mathbf{X}^{(t)})^{\frac{1}{2}}\zeta\zeta^\top(\mathbf{X}^{(t)})^{\frac{1}{2}}\right) \bullet \begin{pmatrix} \mathbf{0} & \mathbf{D}_\triangle^{(e)(t)} \\ \mathbf{D}_\triangle^{(e)(t)} & \mathbf{0} \end{pmatrix}\right] \\
&\quad + \mathbb{E}_{\{\zeta,\nu_e^{(\ell)}\}}\left[6 \cdot \sum_{s\in V\setminus\{i,j\}} (\mathbf{u}_{(i,s)} + \mathbf{u}_{(j,s)})(\mathbf{w}_e^{(t)} - \overline{\mathbf{w}}_e + \nu_e^{(\ell)})\right] \\
&= \mathbf{X}^{(t)} \bullet \begin{pmatrix} \mathbf{0} & \mathbf{D}_\triangle^{(e)(t)} \\ \mathbf{D}_\triangle^{(e)(t)} & \mathbf{0} \end{pmatrix} + 6 \cdot \sum_{s\in V\setminus\{i,j\}} (\mathbf{u}_{(i,s)} + \mathbf{u}_{(j,s)})(\mathbf{w}_e^{(t)} - \overline{\mathbf{w}}_e) \\
&= \nabla f_\triangle(\mathbf{w})_e
\end{aligned}$$

The last equation follows from Lemma 30. Therefore, $\mathbb{E}[\mathbf{g}^{(t)}] = \nabla f_\triangle(\mathbf{w}^{(t)})$.

Next we bound $\mathbb{E}[\|\mathbf{g}\|_\infty^2]$ by bound the two terms of $\mathbf{g}_e^{(t)}$ separately.

For any e, we denote $\mathbf{M}^{(e)} = (\mathbf{X}^{(t)})^{\frac{1}{2}}(\begin{matrix} \mathbf{0} & \mathbf{D}_\triangle^{(e)(t)} \\ \mathbf{D}_\triangle^{(e)(t)} & \mathbf{0} \end{matrix})(\mathbf{X}^{(t)})^{\frac{1}{2}}$. Then we can write

$$\begin{aligned}
\left((\mathbf{X}^{(t)})^{\frac{1}{2}}\zeta\zeta^\top(\mathbf{X}^{(t)})^{\frac{1}{2}}\right) \bullet \begin{pmatrix} \mathbf{0} & \mathbf{D}_\triangle^{(e)(t)} \\ \mathbf{D}_\triangle^{(e)(t)} & \mathbf{0} \end{pmatrix} &= \mathrm{tr}\left(\begin{pmatrix} \mathbf{0} & \mathbf{D}_\triangle^{(e)(t)} \\ \mathbf{D}_\triangle^{(e)(t)} & \mathbf{0} \end{pmatrix}^\top \left((\mathbf{X}^{(t)})^{\frac{1}{2}}\zeta\zeta^\top(\mathbf{X}^{(t)})^{\frac{1}{2}}\right)\right) \\
&= \mathrm{tr}\left(\zeta^\top(\mathbf{X}^{(t)})^{\frac{1}{2}}\begin{pmatrix} \mathbf{0} & \mathbf{D}_\triangle^{(e)(t)} \\ \mathbf{D}_\triangle^{(e)(t)} & \mathbf{0} \end{pmatrix}(\mathbf{X}^{(t)})^{\frac{1}{2}}\zeta\right) \\
&= \mathrm{tr}\left(\zeta^\top\mathbf{M}^{(e)}\zeta\right) \\
&= \zeta^\top\mathbf{M}^{(e)}\zeta
\end{aligned}$$

Since $\|\mathbf{D}_{\triangle}^{(i,j)(t)}\|_1 \leq 3\sum_{\ell\neq i,j}\mathbf{w}_{(i,\ell)}^{(t)}\mathbf{w}_{(\ell,j)}^{(t)} \leq 3\sum_{\ell\neq i,j}\mathbf{u}_{(i,\ell)}\mathbf{u}_{(\ell,j)} \leq 3U_{\triangle}$. The last second inequality is derived from the assumption in Appendix C.1. Then, by denoting that $\begin{pmatrix} \mathbf{0} & \mathbf{D}_{\triangle}^{(e)(t)} \\ \mathbf{D}_{\triangle}^{(e)(t)} & \mathbf{0} \end{pmatrix} = \sum_{i=1}^{(2n)^2} c_i\mathbf{E}_i$ where $\mathbf{E}_i$ has only one single non-zero-entry equal to 1, we can have that

$$\mathbf{M}^{(e)} = \sum_{i=1}^{(2n)^2} (\mathbf{X}^{(t)})^{\frac{1}{2}} c_i\mathbf{E}_i(\mathbf{X}^{(t)})^{\frac{1}{2}}$$

Therefore, similar to the proof of Theorem 35, we have

$$\|\mathbf{M}^{(e)}\|_{\text{op}} \leq \|\mathbf{M}^{(e)}\|_F \leq 2\|\mathbf{D}_{\triangle}^{(e)(t)}\|_1 \leq 6U_{\triangle}$$

Applying Theorem 38, we have

$$\begin{aligned}\mathsf{Pr}\left[|\mathrm{tr}(\mathbf{M}^{(e)}) - \zeta^\top\mathbf{M}^{(e)}\zeta| \geq 6U_{\triangle}z\right] &\leq 2\exp\left(-\min\left(\frac{36U_{\triangle}^2z^2}{CK^4\left\|\mathbf{M}^{(e)}\right\|_F^2}, \frac{6U_{\triangle}z}{CK^2\left\|\mathbf{M}^{(e)}\right\|_{\text{op}}}\right)\right)\\ &\leq 2\exp\left(-\min\left(\frac{36U_{\triangle}^2z^2}{CK^4\cdot 36U_{\triangle}^2}, \frac{6U_{\triangle}z}{CK^2\cdot 6U_{\triangle}}\right)\right)\\ &\leq O(1)\cdot\exp^{-\Theta(z)}\end{aligned}$$

Besides, the following holds,

$$|\mathrm{tr}(\mathbf{M}^{(e)})| = \sum_{i=1}^{(2n)^2}\left|\mathrm{tr}\left(c_i\mathbf{X}^{(t)}\mathbf{E}_i\right)\right| \leq \sum_{i=1}^{(2n)^2}|c_i| = 2\|\mathbf{D}_{\triangle}^{(e)(t)}\|_1 \leq 6U_{\triangle}$$

Therefore, we have

$$\mathsf{Pr}[|\zeta^\top\mathbf{M}^{(e)}\zeta| \geq 12U_{\triangle}z] \leq O(1)\cdot\exp^{-\Theta(z)}.$$

Then, for the second term of $\mathbf{g}_e^{(t)}$, since $\mathbf{w}_e^{(t)} \leq \mathbf{u}_e$ and $\overline{\mathbf{w}}_e \leq \mathbf{u}_e$, we have

$$3\max_{e\in\binom{V}{2}}\sum_{s\in V\setminus\{i,j\}}(\mathbf{u}_{(i,s)}+\mathbf{u}_{(j,s)})(\mathbf{w}_e^{(t)}-\overline{\mathbf{w}}_e) \leq 3\max_{e\in\binom{V}{2}}\sum_{s\in V\setminus\{i,j\}}(\mathbf{u}_{(i,s)}+\mathbf{u}_{(j,s)})2\mathbf{u}_{(i,j)} \leq 6U_{\triangle}$$

Then by Lemma 15, with probability at least $(1-\beta')$, it holds that $|\nu_e^{(\ell)}| \leq \frac{\log(1/\beta')}{\varepsilon_4}$. Thus,

$$3\max_{e=(i,j)\in\binom{V}{2}}\sum_{s\in V\setminus\{i,j\}}(\mathbf{u}_{(i,s)}+\mathbf{u}_{(j,s)})\nu_e^{(\ell)} \leq 3\max_{e=(i,j)\in\binom{V}{2}}\sum_{s\in V\setminus\{i,j\}}(\mathbf{u}_{(i,s)}+\mathbf{u}_{(j,s)})\frac{\log(1/\beta')}{\varepsilon_4} \leq 3U_{\Lambda}\frac{\log(1/\beta')}{\varepsilon_4}$$

Therefore,

$$3\max_{e=(i,j)\in\binom{V}{2}}\sum_{s\in V\setminus\{i,j\}}(\mathbf{u}_{(i,s)}+\mathbf{u}_{(j,s)})(\mathbf{w}_e^{(t)}-\overline{\mathbf{w}}_e+\nu_e^{(\ell)}) \leq 6U_{\triangle}+3U_{\Lambda}\frac{\log(1/\beta')}{\varepsilon_0}$$

By applying union bound over $\binom{n}{2} < n^2$ coordinates of $\mathbf{g}^{(t)}$, we can conclude that,

$$\mathsf{Pr}\left[\left\|\mathbf{g}^{(t)}\right\|_\infty \geq (8z+6)U_\triangle + 3U_\Lambda \frac{\log(1/\beta')}{\varepsilon_4}\right] \leq O(1)n^2 e^{-\Theta(z)} + n^2\beta'$$

Recall that $\varepsilon_4 = \frac{\varepsilon}{6L}$. By choosing $\beta' = \exp^{-\Theta(z)}$ and $z = s + O(\log n^2)$, where $s > 0$, we have

$$\mathsf{Pr}\left[\left\|\mathbf{g}^{(t)}\right\|_\infty^2 \geq \left((9U_\triangle + 3U_\Lambda L/\varepsilon)\left(s + O(\log^2 n)\right)\right)^2\right] \leq O(1)n^2 e^{-s-\log n^2}$$

Denote $K = 9U_\triangle + 3U_\Lambda L/\varepsilon = O\left(U_\triangle + U_\Lambda L/\varepsilon\right)$, we have that

$$\begin{aligned}
\mathbb{E}\left[\left\|\mathbf{g}^{(t)}\right\|_\infty^2\right] &= \int_{s=0}^{\infty} s \cdot \mathsf{pdf}_{\|\mathbf{g}^{(t)}\|_\infty^2}\, ds \\
&\leq O(K^2\log^2 n) + O(1)\cdot \int_{s=O(L^2\log^2 n)}^{\infty} sd(1-\mathsf{Pr}[\left\|\mathbf{g}^{(t)}\right\|_\infty^2 \geq s]) \\
&\leq O(K^2\log^2 n) + O(1)\cdot \int_{s=1}^{\infty} K^2(s+O(\log^2 n))^2 \cdot n^2 e^{-s-\log n^2} ds \\
&\leq O(K^2\log^2 n) + O(K^2\log^2 n)\cdot \int_{s=1}^{\infty} e^{-s} ds + O(K^2)\int_{s=1}^{\infty} e^{-s}s^2 ds \\
&\leq O\left((U_\triangle + U_\Lambda L/\varepsilon)^2 \log^2 n\right)
\end{aligned}$$

The last inequality comes from that the both integrals can be bounded by a constant. ■

Then we bound the parameter R and ρ, which are also defined in Corollary 24. Recall that we choose $\|\cdot\|$ to be l_1 norm.

Lemma 39 *Let* $\Phi(\mathbf{w}) = \sum_{e\in\binom{V}{2}} \mathbf{w}_e \log \mathbf{w}_e$ *be a function defined over* $\mathcal{X} = \{\mathbf{w} \in \mathbb{R}_+^{\binom{V}{2}} : \sum_{e\in\binom{V}{2}} \mathbf{w}_e = W, \mathbf{w}_e \leq \mathbf{u}_e\}$. *Let* $R^2 = \Phi(\mathbf{w}^*) - \min_{\mathbf{w}\in\mathcal{X}} \Phi(\mathbf{w})$, *where* $\mathbf{w}^* = \arg\min_{\mathbf{w}\in\mathcal{X}} f_\triangle(\mathbf{w})$. *Then* Φ *is* $\frac{1}{W}$*-strongly convex with respect to* l_1 *norm, and* $R^2 = O(W\log n)$.

In the above, the notion ρ-strongly convex is defined as follows.

Definition 40 (ρ-strongly convex) *A function* $f(\mathbf{x})$ *is* ρ*-strongly convex with respect to* $\|\cdot\|$ *if and only if*

$$f(\mathbf{y}) \geq f(\mathbf{x}) + (\nabla f(\mathbf{x}))^\top(\mathbf{y}-\mathbf{x}) + \frac{\rho}{2}\|\mathbf{y}-\mathbf{x}\|^2$$

The Inequality described as follows is needed in the proof.

Lemma 41 (Pinsker Inequality (Kullback, 1967)) *Let* $a_i, b_i \geq 0$, $i = 1, \ldots, n$ *s.t.* $\sum_{i=1}^n a_i = \sum_{i=1}^n b_i$. *Then*

$$\sum_{i=1}^n a_i \log\frac{a_i}{b_i} \geq \frac{(\sum_{i=1}^n |a_i - b_i|)^2}{2\sum_{i=1}^n a_i}$$

Proof [Proof of Lemma 39] For any $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, we have

$$\begin{aligned}
\Phi(\mathbf{y}) - \Phi(\mathbf{x}) - (\nabla\Phi(\mathbf{x}))^\top(\mathbf{y}-\mathbf{x}) &= \sum_{e\in\binom{V}{2}} \mathbf{y}_e \log \mathbf{y}_e - \sum_{e\in\binom{V}{2}} \mathbf{x}_e \log \mathbf{x}_e - \sum_{e\in\binom{V}{2}} (1+\log \mathbf{x}_e)\cdot(\mathbf{y}_e - \mathbf{x}_e) \\
&= \sum_{e\in\binom{V}{2}} \mathbf{y}_e \log \frac{\mathbf{y}_e}{\mathbf{x}_e} \\
&\geq \frac{\left(\sum_{e\in\binom{V}{2}} |\mathbf{x}_e - \mathbf{y}_e|\right)^2}{2W} \\
&= \frac{1}{2W}\|\mathbf{x}-\mathbf{y}\|_1^2
\end{aligned}$$

The last second inequality is derived by Lemma 41. According to the definition of ρ-strongly convex, $\Phi(\mathbf{w})$ is $\frac{1}{W}$-strongly convex with respect to l_1 norm.

Then we bound R by giving the upper and lower bound for $\Phi(\mathbf{w})$. Since $\Phi(\mathbf{w})$ is convex, for $\mathbf{w} \in \mathcal{X}$, by Jensen Inequality, we have

$$\Phi(\mathbf{w}) \geq \sum_{e\in\binom{V}{2}} \mathbf{w}_e \log \frac{\sum_{e\in\binom{V}{2}} \mathbf{w}_e}{\binom{n}{2}} \geq -\Omega(W \log \frac{n^2}{W}) = -\Omega(W \log n),$$

Moreover,

$$\Phi(\mathbf{w}) \leq \sum_{e\in\binom{V}{2}} \mathbf{w}_e \log W = W \log W.$$

Therefore,

$$R^2 = \Phi(\mathbf{w}^*) - \min_{\mathbf{w}\in\mathcal{X}} \Phi(\mathbf{w}) \leq W\log W + O(W\log n) = O(W \log n).$$

where the last inequality follows from the fact that W is polynomially bounded by n. ■

Now that we have bounded all the parameters, we can bound the additive error of Algorithm 1 in the following. First, we give the bound of $f_\triangle(\mathbf{w})$ by utilizing Corollary 24.

Lemma 42 *Let $\mathbf{w}$ be the output of Algorithm 1 with parameter $\beta' = \beta/3$, $\eta = \frac{R}{B}\sqrt{\frac{2}{T}}$, where $R = O(\sqrt{W \log n})$, $B = \Theta\left((U_\triangle + U_\Lambda L/\varepsilon)\log n\right)$. Then with probability at least $1-\beta/3$,*

$$f_\triangle(\mathbf{w}) \leq O\left(W \log n \sqrt{\frac{\log n}{T}}(U_\triangle + U_\Lambda L/\varepsilon) + \lambda n \log n\right)$$

Proof Through the above discussion, we have $\rho = \frac{1}{W}$. Now we set $\beta' = \beta/3$, $\eta = \frac{R}{B}\sqrt{\frac{2}{T}}$ in Algorithm 1. By Theorem 5 and Corollary 24, with probability at least $1-\beta/3$, it holds that

$$f_\triangle(\mathbf{w}) - 3\cdot\min_{\mathbf{x}\in\mathcal{X}} f_\triangle(\mathbf{x}) \leq 3RB\sqrt{\frac{2}{\rho T}} = O\left(W\log n\sqrt{\frac{\log n}{T}}(U_\triangle + U_\Lambda L/\varepsilon)\right)$$

By Lemma 28, for any $\mathbf{X} \in \mathcal{D}$, $\lambda_i \in [\frac{1}{n}, 2n], \forall i$. Then, it holds that

$$|\log \det \mathbf{X}| = \left| \sum_{i=1}^{2n} \log \lambda_i \right| = O(n \log n).$$

Moreover, $\overline{\mathbf{w}} \in \mathcal{X}$. Therefore,

$$\min_{\mathbf{x} \in \mathcal{X}} f_{\triangle}(\mathbf{x}) \leq f_{\triangle}(\overline{\mathbf{w}}) = \lambda \log \det \mathbf{X}_{\overline{\mathbf{w}}} = \lambda O(n \log n).$$

where $\mathbf{X}_{\overline{\mathbf{w}}}$ is referred to $\arg\max_{\mathbf{X} \in \mathcal{D}} F_{\triangle}(\overline{\mathbf{w}}, \mathbf{X})$. This further implies that with probability at least $1 - \beta/3$,

$$f_{\triangle}(\mathbf{w}) \leq O\left(W \log n \sqrt{\frac{\log n}{T}} (U_{\triangle} + U_{\Lambda} L/\varepsilon) + \lambda n \log n \right).$$

■

Recall that the triangle-motif cut difference between $\overline{G}$ and $\hat{G}$ can be bounded by $O\left(\ell_3(\hat{G}) \frac{\log(1/\beta)}{\varepsilon}\right)$ according to Lemma 36. Therefore, we give the total additive error in the following, by bound the difference between the input graph $\hat{G}$ and the output graph G.

Theorem 43 (Utility Guarantee) *Let G be the output graph of Algorithm 1 with $T = \Theta(\frac{W(\varepsilon U_{\triangle} + U_{\Lambda})}{n \log(n/\delta) \tilde{\ell}_3(\hat{G})})$, $\beta' = \beta/3$, $\eta = \frac{R}{B}\sqrt{\frac{2}{T}}$, where $R = O(\sqrt{W \log n})$, $B = \Theta\left((U_{\triangle} + U_{\Lambda} L/\varepsilon) \log n\right)$. Assume that $\hat{W} = \Omega(\frac{\log(1/\beta)}{\varepsilon})$ and $\ell_3(\hat{G}) = \Omega(\frac{w_{\max} \log^2(n/\beta)}{\varepsilon^2})$. Then the triangle-motif cut difference between G and the original graph $\hat{G}$ is at most*

$$O\left(\ell_3(\hat{G}) \frac{\log(1/\beta)}{\varepsilon} + \sqrt{W n (U_{\triangle} + U_{\Lambda}/\varepsilon) \ell_3(\hat{G})/\varepsilon} \log^2(\frac{n}{\delta \beta}) \right)$$

with probability at least $(1 - \beta)$.

Proof We denote that

$$h(\mathbf{w}) = \max_{\mathbf{X} \in \mathcal{D}'} H(\mathbf{w}, \mathbf{X}), \quad \text{where } H(\mathbf{w}, \mathbf{X}) = \begin{pmatrix} \mathbf{0} & \mathbf{A}_{\triangle} - \overline{\mathbf{A}}_{\triangle} \\ \mathbf{A}_{\triangle} - \overline{\mathbf{A}}_{\triangle} & \mathbf{0} \end{pmatrix} \bullet \mathbf{X}$$

and

$$\mathcal{D}' = \left\{ \mathbf{X} \in \mathbb{R}^{2n} : \mathbf{X} \text{ is symmetric}, \mathbf{X} \succeq \mathbf{0}, \text{and } \mathbf{X}_{ii} = 1 \text{ for } \forall i \right\}$$

Recall that the triangle-motif cut difference between the output graph G and $\overline{G}$ can be bounded by $h(\mathbf{w})$ up to a constant factor according to Lemma 27. Therefore, if we can bound the gap between $h(\mathbf{w})$ and $f_{\triangle}(\mathbf{w})$, then we can bound $h(\mathbf{w})$, hence giving a bound for the triangle-motif cut difference between G and $\overline{G}$.

Denote that

$$f_1(\mathbf{w}) = \max_{\mathbf{X} \in \mathcal{D}} F_1(\mathbf{w}, \mathbf{X}), \quad \text{where } F_1(\mathbf{w}, \mathbf{X}) = \begin{pmatrix} \mathbf{0} & \mathbf{A}_{\triangle} - \overline{\mathbf{A}}_{\triangle} \\ \mathbf{A}_{\triangle} - \overline{\mathbf{A}}_{\triangle} & \mathbf{0} \end{pmatrix} \bullet \mathbf{X},$$

$$f_2(\mathbf{w}) = \max_{\mathbf{X}\in\mathcal{D}} F_2(\mathbf{w},\mathbf{X}), \quad \text{where } F_2(\mathbf{w},\mathbf{X}) = \begin{pmatrix} \mathbf{0} & \mathbf{A}_\triangle - \overline{\mathbf{A}}_\triangle \\ \mathbf{A}_\triangle - \overline{\mathbf{A}}_\triangle & \mathbf{0} \end{pmatrix} \bullet \mathbf{X} + \lambda \log\det \mathbf{X},$$

and

$$\mathcal{D} = \left\{ \mathbf{X} \in \mathbb{R}^{2n} : \mathbf{X} \text{ is symmetric}, \mathbf{X} \succeq \frac{1}{n}\mathbf{I}_{2n}, \text{and } \mathbf{X}_{ii} = 1 \text{ for } \forall i \right\}$$

Note that $f_2(\mathbf{w})$ and $f_\triangle(\mathbf{w})$ have the same maximizer. By the definition of $f_\triangle(\mathbf{w})$ and $f_2(\mathbf{w})$ and that $\sum_{(i,j)\in\binom{V}{2}} 3(\mathbf{w}_{(i,j)} - \overline{\mathbf{w}}_{(i,j)})^2 \cdot \sum_{s\in V\setminus\{i,j\}}(\mathbf{u}_{(i,s)} + \mathbf{u}_{(j,s)}) \geq 0$, we have

$$f_\triangle(\mathbf{w}) \geq f_2(\mathbf{w}).$$

As proved in Lemma 42, $|\log\det \mathbf{X}| = O(\lambda n \log n)$ for any $\mathbf{X} \in \mathcal{D}$. Let the maximizer of $f_1(\mathbf{w})$ be $\mathbf{X}^{(1)}$. Then it holds that,

$$f_2(\mathbf{w}) \geq F_2(\mathbf{w}, \mathbf{X}^{(1)}) \geq F_1(\mathbf{w}, \mathbf{X}^{(1)}) - C \cdot \lambda n \log n = f_1(\mathbf{w}) - C \cdot \lambda n \log n,$$

for some constant $C > 0$.

Recall that

$$\mathcal{D} = \left\{ \mathbf{X} \in \mathbb{R}^{2n} : \mathbf{X} \text{ is symmetric}, \mathbf{X} \succeq \frac{1}{n}\mathbf{I}_{2n}, \text{and } \mathbf{X}_{ii} = 1 \text{ for } \forall i \right\}$$

and

$$\mathcal{D}' = \left\{ \mathbf{X} \in \mathbb{R}^{2n} : \mathbf{X} \text{ is symmetric}, \mathbf{X} \succeq \mathbf{0}, \text{and } \mathbf{X}_{ii} = 1 \text{ for } \forall i \right\}$$

Let $\mathbf{X}^{(2)}$ be the maximizer of $h(\mathbf{w}) \in \mathcal{D}'$. Let $\mathbf{X}^{(3)} = (1 - \frac{1}{n}) \cdot \mathbf{X}^{(2)} + \frac{1}{n} \cdot \mathbf{I}_{2n}$. We can see that $\mathbf{X}^{(3)} \in \mathcal{D}$. Then we have,

$$\begin{aligned}
f_1(\mathbf{w}) &\geq F_1(\mathbf{w}, \mathbf{X}^{(3)}) \\
&= \begin{pmatrix} \mathbf{0} & \mathbf{A}_\triangle - \overline{\mathbf{A}}_\triangle \\ \mathbf{A}_\triangle - \overline{\mathbf{A}}_\triangle & \mathbf{0} \end{pmatrix} \bullet \left((1 - \frac{1}{n}) \cdot \mathbf{X}^{(2)} + \frac{1}{n}\mathbf{I}_{2n} \right) \\
&= (1 - \frac{1}{n}) \cdot \begin{pmatrix} \mathbf{0} & \mathbf{A}_\triangle - \overline{\mathbf{A}}_\triangle \\ \mathbf{A}_\triangle - \overline{\mathbf{A}}_\triangle & \mathbf{0} \end{pmatrix} \bullet \mathbf{X}^{(2)} \\
&\geq \frac{n-1}{n} H(\mathbf{w}, \mathbf{X}^{(2)}) \\
&= \frac{n-1}{n} h(\mathbf{w}) \geq \frac{1}{2} h(\mathbf{w})
\end{aligned}$$

In a conclusion, we have,

$$h(\mathbf{w}) < 2f_1(\mathbf{w}) < 2f_2(\mathbf{w}) + O(\lambda n \log n) < 2f_\triangle(\mathbf{w}) + O(\lambda n \log n)$$

Since we assume $\hat{W} = \Omega(\frac{\log(1/\beta)}{\varepsilon})$, according to Lemma 36, the triangle-motif cut difference between $\overline{G}$ and $\hat{G}$ is $O(\ell_3(\hat{G})\frac{\log(1/\beta)}{\varepsilon})$. Then to sum up, with probability at least $(1 - \beta/3)$, it holds that

$$\begin{aligned}
&\left| \mathrm{Cut}_\triangle^{(G)}(S,T) - \mathrm{Cut}_\triangle^{(\hat{G})}(S,T) \right| \\
\leq &\left| \mathrm{Cut}_\triangle^{(G)}(S,T) - \mathrm{Cut}_\triangle^{(\overline{G})}(S,T) \right| + \left| \mathrm{Cut}_\triangle^{(\overline{G})}(S,T) - \mathrm{Cut}_\triangle^{(\hat{G})}(S,T) \right|
\end{aligned}$$

$$\begin{aligned}
&\leq O(h(\mathbf{w})) + \left|\mathrm{Cut}_{\triangle}^{(\overline{G})}(S,T) - \mathrm{Cut}_{\triangle}^{(\hat{G})}(S,T)\right| \\
&\leq O\left(\ell_3(\hat{G})\frac{\log(1/\beta)}{\varepsilon}\right) + 2f_{\triangle}(\mathbf{w}) + O(\lambda n \log n) \\
&\leq O\left(\ell_3(\hat{G})\frac{\log(1/\beta)}{\varepsilon} + W\log n\sqrt{\frac{\log n}{T}}\left(U_{\triangle} + \frac{U_{\Lambda}}{\varepsilon}L\right) + \lambda n \log n\right).
\end{aligned}$$

Recall that, according to Theorem 35, $\lambda = \Theta(\varepsilon^{-1})\tilde{\ell}_3(\hat{G})\sqrt{T}\log^{\frac{3}{2}}(\frac{T}{\delta})\log(3/\beta)$ and $L = \log_3(\frac{3}{\beta})$. By the assumption $\ell_3(\hat{G}) = \Omega(\frac{w_{\max}\log^2(n/\beta)}{\varepsilon^2})$, we have $\tilde{\ell}_3(\hat{G}) = \Theta(\ell_3(\hat{G}))$. Moreover, note that, the algorithm fails with at most $2\beta/3$ probability referring to Section 2. Then we can conclude that, by choosing $T = \Theta(\frac{W(\varepsilon U_{\triangle}+U_{\Lambda})}{n\log(n/\delta)\tilde{\ell}_3(\hat{G})})$, the triangle-motif cut difference between the output graph G (with edge weights $\mathbf{w}$) and $\hat{G}$ is at most

$$\begin{aligned}
&O\left(\ell_3(\hat{G})\frac{\log(1/\beta)}{\varepsilon} + W\log n\sqrt{\frac{\log n}{T}}\left(U_{\triangle} + \frac{U_{\Lambda}}{\varepsilon}L\right) + \lambda n \log n\right) \\
\leq &O\left(\ell_3(\hat{G})\frac{\log(1/\beta)}{\varepsilon} + \left(\frac{(U_{\triangle} + U_{\Lambda}/\varepsilon)W}{\sqrt{T}} + \tilde{\ell}_3(\hat{G})\sqrt{T}n\log(\frac{nT}{\delta})/\varepsilon\right)\log^{\frac{3}{2}}(\frac{nT}{\delta})\log(\frac{1}{\beta})\right) \\
\leq &O\left(\ell_3(\hat{G})\frac{\log(1/\beta)}{\varepsilon} + \sqrt{Wn(U_{\triangle} + U_{\Lambda}/\varepsilon)\ell_3(\hat{G})/\varepsilon}\log^2(\frac{n}{\delta\beta})\right)
\end{aligned}$$

with probability at least $(1-\beta)$. ■

To provide a more concise guarantee, we first give some bounds to quantities $U_{\triangle}$ and U_{Λ}. Let $w_{\max} = \max_{e\in\binom{V}{2}}\mathbf{w}_e = \Theta(\max_{e\in\binom{V}{2}}\overline{\mathbf{w}}_e)$ be the maximum edge weight.

Lemma 44 *It holds that $U_{\triangle} = O(nw_{\max}^2\frac{\log^2(n/\beta)}{\varepsilon^2})$, $U_{\Lambda} = O(nw_{\max}\frac{\log(n/\beta)}{\varepsilon})$.*

Proof We have that

$$\begin{aligned}
U_{\Lambda} = &\max_{(i,j)\in\binom{V}{2}}\sum_{s\in V\setminus\{i,j\}}(\mathbf{u}_{(i,s)} + \mathbf{u}_{(j,s)}) \leq \max_{(i,j)\in\binom{V}{2}}\sum_{s\in V\setminus\{i,j\}}(\overline{\mathbf{w}}_{(i,s)} + \overline{\mathbf{w}}_{(j,s)} + 12\frac{\log(3n^2/\beta)}{\varepsilon} + 2\frac{W}{\binom{n}{2}}) \\
&\leq 2dw_{\max} + 12n\frac{\log(3n^2/\beta)}{\varepsilon} + 4\frac{W}{n} \leq O(nw_{\max}\frac{\log(n/\beta)}{\varepsilon}),
\end{aligned}$$

and that

$$\begin{aligned}
U_{\triangle} = &\max_{(i,j)\in\binom{V}{2}}\sum_{s\in V\setminus\{i,j\}}(\mathbf{u}_{(i,j)}\mathbf{u}_{(i,s)} + \mathbf{u}_{(i,s)}\mathbf{u}_{(j,s)} + \mathbf{u}_{(j,s)}\mathbf{u}_{(i,j)}) \\
&\leq \max_{(i,j)\in\binom{V}{2}}\sum_{s\in V\setminus\{i,j\}}(\overline{\mathbf{w}}_{(i,j)}\overline{\mathbf{w}}_{(i,s)} + \overline{\mathbf{w}}_{(i,s)}\overline{\mathbf{w}}_{(j,s)} + \overline{\mathbf{w}}_{(j,s)}\overline{\mathbf{w}}_{(i,j)}) + 144n\frac{\log^2(3n^2/\beta)}{\varepsilon^2} + 2n(\frac{W}{\binom{n}{2}})^2 \\
&+ (12\frac{\log(3n^2/\beta)}{\varepsilon} + 2\frac{W}{\binom{n}{2}})\max_{(i,j)\in\binom{V}{2}}\sum_{s\in V\setminus\{i,j\}}(\overline{\mathbf{w}}_{(i,j)} + \overline{\mathbf{w}}_{(i,s)} + \overline{\mathbf{w}}_{(j,s)})
\end{aligned}$$

$$\leq \ell_3(\overline{G}) + O(nw_{\max}(\log^2(n/\beta) + \frac{W}{\binom{n}{2}})) \leq O(dw_{\max}^2 + nw_{\max}\frac{\log^2(n/\beta)}{\varepsilon^2}) \leq O(nw_{\max}^2\frac{\log^2(n/\beta)}{\varepsilon^2}).$$

■

Corollary 45 *Let G be the output graph of Algorithm 1 with $T = \Theta(\frac{W(\varepsilon U_\triangle + U_\Lambda)}{n\log(n/\delta)\tilde{\ell}_3(\hat{G})})$, $\beta' = \beta/3$, $\eta = \frac{R}{B}\sqrt{\frac{2}{T}}$, where $R = O(\sqrt{W\log n})$, $B = \Theta\left((U_\triangle + U_\Lambda L/\varepsilon)\log n\right)$. Denote $\hat{W}$ and $w_{\max}$ as the sum of edge weights and the maximum edge weight of the original graph $\hat{G}$ separately. Then the triangle-motif cut difference between G and $\hat{G}$ is at most*

$$O\left(\sqrt{\hat{W}\cdot\ell_3(\hat{G})}\cdot nw_{\max}\frac{\log^2(n/\delta\beta)}{\varepsilon^{3/2}}\right)$$

Proof We first assume that $\hat{W} = \Omega(\frac{\log(1/\beta)}{\varepsilon})$ and $\ell_3(\hat{G}) = \Omega(\frac{w_{\max}\log^2(n/\beta)}{\varepsilon^2})$, which we refer to as the non-degenerate case. By Lemma 44, $U_\triangle = O(nw_{\max}^2\frac{\log^2(n/\beta)}{\varepsilon^2})$, $U_\Lambda = O(nw_{\max}\frac{\log(n/\beta)}{\varepsilon})$. Therefore, by Theorem 43, the additive error is at most

$$\begin{aligned}
&O\left(\ell_3(\hat{G})\frac{\log(1/\beta)}{\varepsilon} + \sqrt{Wn(U_\triangle + U_\Lambda/\varepsilon)\ell_3(\hat{G})/\varepsilon}\log^2(\frac{n}{\delta\beta})\right)\\
\leq &O\left(\ell_3(\hat{G})\frac{\log(1/\beta)}{\varepsilon} + \sqrt{W\ell_3(\hat{G})}nw_{\max}\frac{\log^2(n/\delta\beta)}{\varepsilon^{3/2}}\right)\\
\leq &O\left(\sqrt{W\ell_3(\hat{G})}nw_{\max}\frac{\log^2(n/\delta\beta)}{\varepsilon^{3/2}}\right).
\end{aligned}$$

The last inequality comes from the fact that $\ell_3(\hat{G}) \leq nw_{\max}^2$.

Furthermore, recall that $W = \hat{W} + \mathrm{Lap}(3/\varepsilon) + 3\log(3/\beta)/\varepsilon$. Note that we have assumed $\hat{W} = \Omega(\frac{\log(1/\beta)}{\varepsilon})$, the additive error is at most $O\left(\sqrt{\hat{W}\cdot\ell_3(\hat{G})}\cdot nw_{\max}\frac{\log^2(n/\delta\beta)}{\varepsilon^{3/2}}\right)$.

In the degenerate case when the above assumptions are not satisfied, we will show that, the weight sum of the triangles of $\hat{G}$ is at most $O\left(\sqrt{\hat{W}\cdot\ell_3(\hat{G})}\cdot nw_{\max}\frac{\log^2(n/\delta\beta)}{\varepsilon^{3/2}}\right)$. That is, we can release an empty graph and achieving the same utility guarantee.

We first show that:

$$\sum_{(i,j,k)\in\binom{V}{3}}\hat{w}_{(i,j)}\hat{w}_{(i,s)}\hat{w}_{(j,s)} \leq \sum_{(i,j)\in\binom{V}{2}}\hat{w}_{(i,j)}\cdot\sum_{s\neq i,j}\hat{w}_{(i,s)}\hat{w}_{(j,s)} = \hat{W}\cdot\ell_3(\hat{G})$$

Additionally, we have $\hat{W} \leq n^2 w_{\max}$ and $\ell_3(\hat{G}) \leq nw_{\max}^2$. Thus it follows:

If $\hat{W} = o(\frac{\log(1/\beta)}{\varepsilon})$, then the weight sum of the triangles of $\hat{G}$ is at most

$$\hat{W}\cdot\ell_3(\hat{G}) = o(\sqrt{\hat{W}\cdot\ell_3(\hat{G})}\sqrt{\frac{\log(1/\beta)}{\varepsilon}}\sqrt{nw_{\max}^2}) = o\left(\sqrt{\hat{W}\cdot\ell_3(\hat{G})}\cdot nw_{\max}\frac{\log^2(n/\delta\beta)}{\varepsilon^{3/2}}\right).$$

If $\ell_3(\hat{G}) = o(\frac{w_{\max}\log^2(n/\beta)}{\varepsilon^2})$, then the weight sum of the triangles of $\hat{G}$ is at most

$$\hat{W}\cdot\ell_3(\hat{G}) = o(\sqrt{\hat{W}\cdot\ell_3(\hat{G})}\cdot\sqrt{n^2 w_{\max}\cdot\frac{w_{\max}\log^2(n/\beta)}{\varepsilon^2}}) = o\left(\sqrt{\hat{W}\cdot\ell_3(\hat{G})}\cdot n w_{\max}\frac{\log^2(n/\delta\beta)}{\varepsilon^{3/2}}\right).$$

■

Finally, we note that the correctness of Theorem 3 follows from Theorem 35 and Corollary 45 with error probability $\beta = \frac{1}{4}$.

D.6. Complexity Analysis

Lemma 46 *Algorithm 1 can be implemented in time $\tilde{O}(n^6 W \log^{O(1)}(n))$ with the same level of guarantee for additive error and privacy.*

Recall that we assume the bound of the sum of edge weights to be polynomial. Thus Algorithm 1 runs in polynomial time. The proof of the lemma is similar to (Eliáš et al., 2020), so here we only give a proof sketch.

Proof [Proof Sketch of Lemma 46] According to Theorem 43 and Lemma 44, we have $T = \Theta(\frac{W(\varepsilon U_\triangle + U_\Lambda)}{n\log(n/\delta)\tilde{\ell}_3(\hat{G})}) = \tilde{O}(W)$. In each inner iteration of Algorithm 1, note that Algorithm 2 runs in $O(n^2)$ time. Denote $\mathbf{X}^*$ be the maximizer of the SDP $\max_{\mathbf{X}\in\mathcal{D}} F_\triangle(\mathbf{w},\mathbf{X})$. We use the algorithm of (Lee et al., 2015) to find an approximate solution of the SDP in Line 6 of Algorithm 1 in time $\tilde{O}(n^6\log^{O(1)}(n/\mu))$, such that, $\left\|(\mathbf{X}^*)^{-\frac{1}{2}}(\mathbf{X}^*-\mathbf{X})(\mathbf{X}^*)^{-\frac{1}{2}}\right\|_F \leq \mu$. Therefore, within LT iterations, Algorithm 1 can be implemented in time $\tilde{O}(n^6 W\log^{O(1)}(n))$. According to (Eliáš et al., 2020), the additive error only differs by a constant factor when we choose $\mu = 1/n^{O(1)}$. And the privacy loss in each iteration of Algorithm 1 is still $O(\frac{\ell_3(\hat{G})}{\lambda})$, hence the privacy guarantee is the same as before. ■

Appendix E. Deferred Proofs from Section 4

Now we give the proof of Theorem 6. We consider an unweighted graph $G = (V, E)$. We construct a matrix $\mathbf{A}$ with $\binom{n}{h}$ columns corresponding to the pairs of sets $S, T \subset V$ such that

$$\mathbf{A}_{(S,T),I} = \begin{cases} 1 & \text{if } I \in (S\times T) \\ 0 & \text{otherwise} \end{cases}$$

Note that $\mathbf{A}$ is fixed and does not depend on G. Let $\mathbf{x}\in\{0,1\}^{\binom{n}{2}}$ be the indicator vector of E. Let $\mathbf{x}_{K_h} = f_{K_h}(x) \in \{0,1\}^{\binom{n}{h}}$ be the indicator vector of K_h in G, i.e. for each tuple $i_1, i_2, \ldots, i_h$ of h different indices, $(\mathbf{x}_{K_h})_{i_1,i_2,\ldots,i_h} = 1$ if the subgraph induced by $i_1, i_2, \ldots, i_h$ is K_h and 0 otherwise. Then $\mathbf{A}\mathbf{x}_{K_h}$ specifies the K_h-motif size of all (S,T)-cuts in G, i.e., we have

$$(\mathbf{A}\mathbf{x}_{K_h})_{(S,T)} = \sum_{I\in\mathcal{M}(G,K_h)} \mathbf{1}_{I \text{ crosses } (S,T)}.$$

Then we define the discrepancy of A in terms of the set of 3-colorings over the set of all K_h motifs, and give bounds on its discrepancy by using random graphs, which will then imply our lower bound by a reduction from (Muthukrishnan and Nikolov, 2012). Recall that we define

$$\mathrm{disc}_{\mathcal{C}}(\mathbf{B}) = \min\{\|\mathbf{B}\chi\|_\infty : \chi \in \mathcal{C}\}$$

as the discrepancy of a matrix B (see Definition 7).

E.1. Proof of Lemma 8

We will make use of the following lemma.

Lemma 47 (Lemma 6 in (Bollobás and Scott, 2006)) *Let $\{a_i, 1 \le i \le n\}$ be a sequence of real numbers. Let $\rho_i \in \{0,1\}$ be i.i.d. Bernoulli, for $1 \le i \le n$. Then*

$$\mathbb{E}\left[\left|\sum_{i=1}^{n} \rho_i a_i\right|\right] \ge \frac{\sum_{i=1} |a_i|}{\sqrt{8n}}.$$

Now we prove our main lemma in this section, i.e., Lemma 8, which is a generalization of a result of (Eliáš et al., 2020) (see also (Bollobás and Scott, 2006)).

Proof [Proof of Lemma 8] For any $i \le h$ and any set $W \subseteq V$, we let $W^{(i)}$ denote the set of all subsets $R \subseteq W$ with exactly i distinct vertices. For any $i \le h$ and a set $R \subseteq V^{(i)}$, and $h-i$ other vertices $t_{i+1}, \ldots, t_h$, we let $\chi_{R,t_{i+1},\ldots,t_h}$ denote the coloring of the potential K_h instance induced by the vertex set $R \cup \{t_{i+1}, \ldots, t_h\}$. We define $\chi_R = \sum_{t_{i+1},\ldots,t_h \in V^{(h-i)}} \chi_{R,t_{i+1},\ldots,t_h}$ and define $\chi_{R,t} = \sum_{t_{i+2},\ldots,t_h \in V^{(h-i-1)}} \chi_{R,t,t_{i+2},\ldots,t_h}$. We extend the definition of χ to any tuple $i_1, \ldots, i_h \in V^{(h)}$ by setting $\chi_{i_1,\ldots,i_h} = 0$ whenever there exists a repeated index in the tuple.

For two disjoint sets $P, Q \subset V$, we let

$$\chi_{1,h-1}(P,Q) = \sum_{t_1 \in P, B \in Q^{(h-1)}} \chi_{t_1,B} = \sum_{t_1 \in P, t_2,\ldots,t_h \in Q} \chi_{t_1,\ldots,t_h}.$$

We need the following lemma:

Lemma 48 *If $\mathrm{disc}_{\mathcal{C}_{\sigma,\gamma}}(\mathbf{A}) \le M$, then for any $\chi \in \mathcal{C}_{\sigma,\gamma}$ and disjoint subsets $P, Q \subset V$,*

$$|\chi_{1,h-1}(P,Q)| \le 2^{2h^2} M.$$

Proof Let Z be a random subset of P, where each vertex is chosen independently with probability p. Then

$$\mathbb{E}[(A\mathbf{x}_{K_h})_{(Z\cup Q, Z\cup Q)}] = \sum_{i=0}^{h} p^i \cdot \sum_{t_1 \in P,\ldots,t_i \in P, t_{i+1} \in Q \ldots, t_h \in Q} \chi_{t_1,\ldots,t_h}$$

By Markov Inequality, we have the fact that if $P(x)$ is a h-degree polynomial with $\sup_{x\in[0,1]} |P(x)| \le 1$ then every coefficient of $P(x)$ has absolute value at most $2^h h^{2h}/h!$. Since $M \ge \mathrm{disc}_{\mathcal{C}_{\sigma,\gamma}}(\mathbf{A}) \ge \sup_{p\in[0,1]} \left|\sum_{i=0}^{h} p^i \cdot \sum_{t_1 \in P,\ldots,t_i \in P, t_{i+1} \in Q \ldots, t_h \in Q} \chi_{t_1,\ldots,t_h}\right|$, we have $\left|\sum_{t_1 \in P, t_2 \in Q,\ldots,t_h \in Q} \chi_{t_1,\ldots,t_h}\right| \le 2^h h^{2h} M/h! \le 2^{2h^2} M$. ■

We will show that for any $\chi \in \mathcal{C}_{\sigma,\gamma}$, we can find disjoint $P, Q \subset V$ such that

$$|\chi_{1,h-1}(P,Q)| \geq \Omega_h(\sigma \cdot \gamma^{1/2} n^{3/2} \ell_h(G)).$$

This will then finish the proof of the lemma.

We first prove the following useful claim.

Claim 49 *Let $H = (W, E)$ be a graph such that for any fixed B with $|B| = i-1$, the number of $t \in W$ such that t is connected to all vertices in B is at most $2n\gamma^{i-1}$. Let $S \cup T$ be a random partition of W, i.e., each vertex is independently assigned to each class with probability $1/2$. Then*

$$\mathbb{E}\left[\sum_{B \in S^{(i-1)}} \left|\sum_{t \in T} \chi_{B,t}\right|\right] \geq i \cdot 2^{-i-3} \sum_{L \in W^{(i)}} |\chi_L| / \sqrt{n\gamma^{i-1}}$$

Proof Let $\rho_v \in \{0,1\}$ be i.i.d. Bernoulli for each v. Note that for any given $B \in S^{(i-1)}$, by Lemma 47, it holds that

$$\mathbb{E}\left[\left|\sum_{t \in T \setminus B} \chi_{B,t}\right|\right] = \mathbb{E}\left[\left|\sum_{t \in W \setminus B} \rho_t \chi_{B,t}\right|\right] \geq \frac{\sum_{t \in W \setminus B} |\chi_{B,t}|}{\sqrt{8 \cdot 2n\gamma^{i-1}}} = \frac{\sum_{L \in W^{(i)}} |\chi_L|}{\sqrt{16n\gamma^{i-1}}},$$

where we make use of the fact that for any fixed $B \in S^{(i-1)}$, the number of t such that $\chi_{B,t}$ is non-zero is at most $2n\gamma^{i-1}$, as such a t should at least connect to all vertices in B. Since the event that $B \subset S$ and the random variable $\sum_{t \in W \setminus B} |\chi_{B,t}|$ are independent, and each $L \in W^{(i)}$ occurs i times as $L = B \cup \{t\}$, we have that

$$\begin{aligned}
\mathbb{E}\left[\sum_{B \in S^{(i-1)}} \left|\sum_{t \in T} \chi_{B,t}\right|\right] &= \sum_{B \in W^{(i-1)}} \mathsf{Pr}[B \subset S] \cdot \mathbb{E}\left[\left|\sum_{t \in T} \chi_{B,t}\right|\right] \\
&\geq \sum_{B \in W^{(i-1)}} 2^{-i+1} \frac{\sum_{t \in W \setminus B} |\chi_{B,t}|}{\sqrt{16n\gamma^{i-1}}} \\
&= i2^{-i-1} \sum_{L \in W^{(i)}} |\chi_L| / \sqrt{16n\gamma^{i-1}}
\end{aligned}$$

■

Consider $V = S_h \cup T_{h-1}$ as a random partition, and let $T_{h-1} = S_{h-1} \cup T_{h-2}, \ldots, T_2 = S_2 \cup T_1$ represent random bipartitions. In each bipartition, every vertex is independently assigned to either vertex class with a probability of $1/2$. For each $i \leq h-1$ and a subset $R \subseteq T_i$ with i distinct vertices (i.e., $R \in T_i^{(i)}$), we define

$$y_i(R) = \sum_{R, s_{i+1} \in S_{i+1}, \ldots, s_h \in S_h} \chi_{R, s_{i+1}, \ldots, s_h}.$$

Let $T_h = V$ and define $y_h = \chi$. Then for $1 \leq i < h$ and $R \in T_i^{(i)}$, we have

$$y_i(R) = \sum_{s_{i+1} \in S_{i+1}, \ldots, s_h \in S_h} \chi_{R, s_{i+1}, \ldots, s_h} = \sum_{s \in S_{i+1}} \sum_{s_{i+2}, \ldots, s_h} \chi_{R \cup \{s\}, s_{i+2}, \ldots, s_h} = \sum_{s \in S_{i+1}} y_{i+1}(R \cup \{s\}).$$

Now by our assumption on the properties of the graphs G and G', each edge $e = (u, v) \in E \triangle E'$, it holds that the K_h instances containing e are either in G or in G', but not both; and each edge belongs to at least $\frac{\ell_h(G)}{2}$ number of K_h instances. Therefore,

$$|y_2(\{u,v\})| = \sum_{s_3 \in S_3, \dots, s_h \in S_h} |\chi_{u,v,s_3,\dots,s_h}| \geq \frac{\ell_h(G)}{2}$$

Note that by our definition of T_2, the probability that the two endpoints of any edge e belong to T_2 is at least $\frac{1}{2^{h+1}}$. Thus,

$$\mathbb{E}[\sum_{R \in T_2^{(2)}} |y_2(R)|] \geq 2^{-h-1} \cdot \sigma\gamma n^2 \cdot \frac{\ell_h(G)}{2}.$$

By Claim 49, we have that given T_{i+1} and y_{i+1},

$$\mathbb{E}[\sum_{R \in T_i^{(i)}} |y_i(R)|] \geq (i+1)2^{-(i+4)} \sum_{L \in T_{i+1}^{(i+1)}} |y_{i+1}(L)| / \sqrt{n\gamma^i}$$

Therefore,

$$\begin{aligned}
\mathbb{E}\left[\sum_{s \in T_1} \left| \sum_{s_2 \in S_2, \dots, s_h \in S_h} \chi_{s,s_2,\dots,s_h} \right|\right] &= \mathbb{E}[\sum_{s \in T_1} |y_1(s)|] \\
&\geq 2^{-4} \cdot \mathbb{E}\left[\sum_{R \in T_2^{(2)}} |y_2(R)| / \sqrt{n\gamma}\right] \\
&\geq 2^{-h-6} \cdot \sigma \cdot \gamma n^2 \cdot \ell_h(G)/\sqrt{n\gamma} \\
&\geq 2^{-h-6} \cdot \sigma \cdot \gamma^{1/2} n^{3/2} \cdot \ell_h(G) := \Upsilon
\end{aligned}$$

Now we let $S_1^+ = \{s \in T_1 : \sum_{s_2 \in S_2, \dots, s_h \in S_h} \chi_{s,s_2,\dots,s_h} > 0\}$, and $S_1^- = T_1 \setminus S_1^+$. Then we have

$$\begin{aligned}
\Upsilon &\leq \mathbb{E}\left[\sum_{s \in T_1} \left| \sum_{s_2 \in S_2, \dots, s_h \in S_h} \chi_{s,s_2,\dots,s_h} \right|\right] \\
&= \mathbb{E}\left[\sum_{s \in S_1^+} \left(\sum_{s_2 \in S_2, \dots, s_h \in S_h} \chi_{s,s_2,\dots,s_h} \right)\right] + \mathbb{E}\left[\sum_{s \in S_1^-} \left(- \sum_{s_2 \in S_2, \dots, s_h \in S_h} \chi_{s,s_2,\dots,s_h} \right)\right] \\
&:= \mathbb{E}\left[\left|\chi_{S_1^+, S_2, \dots, S_h}\right|\right] + \cdot \mathbb{E}\left[\left|\chi_{S_1^-, S_2, \dots, S_h}\right|\right]
\end{aligned}$$

Therefore at least one of $\mathbb{E}\left[\left|\chi_{S_1^+, S_2, \dots, S_h}\right|\right], \mathbb{E}\left[\left|\chi_{S_1^-, S_2, \dots, S_h}\right|\right]$ is at least $\Upsilon/2$. Without loss of generality, we assume that $\mathbb{E}\left[\left|\chi_{S_1^+, S_2, \dots, S_h}\right|\right] \geq \Upsilon/2$.

Now for a nonempty set $\mathcal{I} \subseteq \{2, \dots, h\}$, we let $V_{\mathcal{I}} = \cup_{i \in \mathcal{I}} S_i$ and let

$$\chi_{\mathcal{I}} = \sum_{s \in S_1^+, B \in V_{\mathcal{I}}^{(h-1)}, |B \cap S_i| > 0, \forall i \in \mathcal{I}} \chi_{s,B}.$$

Note that

$$\chi_{1,h-1}(S_1^+, V_{\mathcal{I}}) = \sum_{\emptyset \neq \mathcal{J} \subseteq \mathcal{I}} \chi_{\mathcal{J}}.$$

Note that $\chi_{\{2,\ldots,h\}} = \chi_{S_1^+, S_2, \ldots, S_h}$.

Now consider the family

$$\mathcal{F} = \{\mathcal{J} \subseteq \{2, \ldots, h\} : |\chi_{\mathcal{J}}| \geq (2h)^{-h+|\mathcal{J}|} \cdot \left|\chi_{S_1^+, S_2, \ldots, S_h}\right|\}.$$

Note that $\mathcal{F}$ is not empty, as the set $\{2, \ldots, h\}$ belongs to $\mathcal{F}$. Now we let $\mathcal{J}_0$ be the set in $\mathcal{F}$ with the minimal size. Note that by definition, for all $\mathcal{I} \subsetneq \mathcal{J}_0$, it holds that $|\chi_{\mathcal{I}}| < (2h)^{-h+|\mathcal{I}|} \cdot \left|\chi_{S_1^+, S_2, \ldots, S_h}\right|$.

Therefore,

$$\begin{aligned}
\max_{\mathcal{J} \subseteq \{2,\ldots,h\}} \left|\chi_{1,h-1}(S_1^+, V_{\mathcal{J}})\right| &\geq \left|\chi_{1,h-1}(S_1^+, V_{\mathcal{J}_0})\right| \\
&\geq |\chi_{\mathcal{J}_0}| - \sum_{\emptyset \neq \mathcal{I} \subsetneq \mathcal{J}_0} |\chi_{\mathcal{I}}| \\
&\geq \left((2h)^{-h+|\mathcal{J}_0|} - \sum_{i=1}^{|\mathcal{J}_0|-1} h^{|\mathcal{J}_0|-i} (2h)^{-h+i} \right) \left|\chi_{S_1^+, S_2, \ldots, S_h}\right| \\
&\geq 2^{-h^2} \cdot \left|\chi_{S_1^+, S_2, \ldots, S_h}\right|
\end{aligned}$$

Thus, $\mathbb{E}\left[\max_{\mathcal{J} \subseteq \{2,\ldots,h\}} \left|\chi_{1,h-1}(S_1^+, V_{\mathcal{J}})\right|\right] \geq 2^{-h^2-1}\Upsilon$. Thus, there exists a subset $\mathcal{J} \subseteq \{2, \ldots, h\}$ with

$$\mathbb{E}\left[\left|\chi_{1,h-1}(S_1^+, V_{\mathcal{J}})\right|\right] \geq 2^{-h^2-1}\Upsilon.$$

Now we choose $P = S_1^+$ and $Q = V_{\mathcal{J}}$ which achieve at least the expectation of $\mathbb{E}\left[\left|\chi_{1,h-1}(S_1^+, V_{\mathcal{J}})\right|\right]$. Then it holds that

$$|\chi_{1,h-1}(P, Q)| \geq 2^{-h^2-1}\Upsilon = \Omega_h(\sigma \cdot \gamma^{1/2} n^{3/2} \ell_h(G)).$$

By Lemma 48, this finishes the proof of Lemma 8. ■

We note that by the preconditions in Lemma 8, it holds that $m = \Theta(\gamma n^2)$. Thus, $\gamma^{1/2} n^{3/2} = \Theta(\sqrt{\frac{m}{n^2}} n^{3/2}) = \Theta(\sqrt{mn})$. This quantity will be used in our lower bound for the discrepancy.

E.2. Finishing the Proof of Theorem 6

The remaining proofs are almost the same as the corresponding proofs (with small changes) in (Eliáš et al., 2020). We present the proofs here for the sake of completeness.

Lemma 50 *Let* $\mathbf{x}$ *be an indicator vector of the edge set of a graph* $G = (V, E)$ *which satisfies the following properties* $(\star)$*:*

1. *for each vertex* $v \in V$*, its degree belongs to the interval* $[\gamma n/2, 2\gamma n]$*;*
2. *for each edge* $e \in E$*, the number of* K_h*-instances containing* e *belongs to the interval* $[\frac{\ell_h(G)}{2}, 2\ell_h(G)]$*;*

3. *for $1 \leq i \leq h$ and any subset B with i distinct vertices in G and G', the number of vertices t such that t is connected to all vertices in B belongs to the interval $\left[\frac{n\gamma^i}{2}, 2n\gamma^i\right]$.*

Let $\mathcal{M}$ be a mechanism for the motif size of all cuts that outputs $\mathbf{y}$ with the input x, i.e., $\mathbf{y} = \mathcal{M}(\mathbf{x})$. Suppose that

$$\|\mathbf{y} - \mathbf{A} \cdot \mathbf{x}_{K_h}\|_\infty \leq \frac{1}{2}\mathrm{disc}_{\mathcal{C}_{\sigma,\gamma}}(\mathbf{A}).$$

Then there exists a deterministic algorithm $\mathcal{A}$ which given as input $\mathbf{y}$ and outputs a vector $\mathcal{A}(\mathbf{y})$ such that

$$\|\mathcal{A}(\mathbf{y}) - \mathbf{x}\|_1 \leq \sigma\gamma n^2.$$

Proof Let $\mathcal{A}$ simply be the algorithm that output an indicator vector x' of any graph that satisfies the properties $(\star)$ and that

$$\left\|\mathbf{y} - \mathbf{A} \cdot \mathbf{x}'_{K_h}\right\|_\infty < \frac{1}{2}\mathrm{disc}_{\mathcal{C}_{\sigma,\gamma}}(\mathbf{A}).$$

Note that such an $\mathbf{x}'$ exists, as $\mathbf{x}$ already satisfies the required properties. We consider the vector $\mathbf{x}' - \mathbf{x}$. Assume that $\|\mathbf{x} - \mathbf{x}'\|_1 > \sigma\gamma n^2$. Then $\chi = \mathbf{x}_{K_h} - \mathbf{x}'_{K_h}$ belongs to $\mathcal{C}_{\sigma,\gamma}$ and therefore

$$\left\|\mathbf{A} \cdot (\mathbf{x}_{K_h} - \mathbf{x}'_{K_h})\right\|_\infty \geq \mathrm{disc}_{\mathcal{C}_{\sigma,\gamma}}(\mathbf{A}).$$

On the other hand,

$$\left\|\mathbf{A} \cdot (\mathbf{x}_{K_h} - \mathbf{x}'_{K_h})\right\|_\infty \leq \|\mathbf{y} - \mathbf{A} \cdot \mathbf{x}_{K_h}\|_\infty + \left\|\mathbf{y} - \mathbf{A} \cdot \mathbf{x}'_{K_h}\right\|_\infty < \mathrm{disc}_{\mathcal{C}_{\sigma,\gamma}}(\mathbf{A}),$$

which is a contradiction. Thus, it holds that $\|\mathbf{x} - \mathbf{x}'\|_1 \leq \sigma\gamma n^2$. ■

The remaining proof largely follows directly from (Eliáš et al., 2020), and we only need to make slight adjustments to adapt their proofs for our case. We state the proofs here for the sake of completeness.

Let X be the distribution of vectors $\mathbf{x} \in \{0,1\}^{\binom{n}{2}}$, where each coordinate is chosen independently such that $\mathbf{x}_i = 1$ with probability p. Thus, the distribution X is the distribution of indicator vectors of graphs $G \sim G(n,p)$, where $G(n,p)$ denotes the distribution of Erdős-Rényi random graphs.

Lemma 51 ((Eliáš et al., 2020)) *Let $\mathcal{M}$ be an (ε, δ)-differentially private mechanism and let Y be the probability distribution over the transcripts of $\mathcal{M}(\mathbf{x})$, where $\mathbf{x}$ is drawn from distribution X. Then for any $\gamma > 0$ and $\mathbf{y} \sim Y$, it holds that with probability $1 - \delta'$ over $i \in [n]$ and $\mathbf{y} \leftarrow X_{|Y=y}$, we have*

$$2^{-\varepsilon-\gamma}\frac{1-p}{p} \leq \frac{\mathsf{Pr}_{\mathbf{x} \leftarrow X_{|Y=\mathbf{y}}}[\mathbf{x}_i = 0 \mid \mathbf{x}_{-i}]}{\mathsf{Pr}_{\mathbf{x} \leftarrow X_{|Y=\mathbf{y}}}[\mathbf{x}_i = 1 \mid \mathbf{x}_{-i}]} \leq 2^{\varepsilon+\gamma}\frac{1-p}{p},$$

where x_{-i} denotes the vector of all coordinates of x excluding x_i.

Now we prove the lower bound for $\varepsilon = 1$.

Lemma 52 *Let $h \geq 2$ be a constant. Let $G \sim G(n,p)$, where $(\frac{\log n}{n})^{1/(h-1)} \ll p \leq \frac{1}{2}$, be a random graph and let $\mathcal{M}$ be a $(1,\delta)$-DP mechanism which approximates the K_h-motif size of all (S,T)-cuts of G up to additive error α with probability β. Then $\alpha \geq \Omega(\mathrm{disc}_{\mathcal{C}_{\sigma,\gamma}}(\mathbf{A}))$, where $\gamma = p, \sigma = \Omega(1 - \frac{9\delta}{\beta})$.*

Proof [Proof Sketch] We choose $\varepsilon = 1$ and $\varepsilon' = \varepsilon + 10$. This implies $\delta' = 2\delta \cdot \frac{1+e^{-\varepsilon-10}}{1-e^{-10}} \leq 3\delta$, then with probability $1-\delta'$ over over $i \in [n]$ and $\mathbf{y} \leftarrow X_{|Y=\mathbf{y}}$, we have

$$2^{-\varepsilon'}\frac{1-p}{p} \leq \frac{\Pr_{\mathbf{x}\leftarrow X_{|Y=\mathbf{y}}}[\mathbf{x}_i = 0 \mid \mathbf{x}_{-i}]}{\Pr_{\mathbf{x}\leftarrow X_{|Y=\mathbf{y}}}[\mathbf{x}_i = 1 \mid \mathbf{x}_{-i}]} \leq 2^{\varepsilon'}\frac{1-p}{p}. \tag{16}$$

Then we can prove the lemma by contradiction. That is, we assume that $\mathcal{M}$ has additive error smaller than $\mathrm{disc}_{\mathcal{C}_{\sigma,\gamma}}(\mathbf{A})/2-1$ with probability at least β. Then we can show that for each possible output $\mathbf{y}$ of the mechanism $\mathcal{M}$, with probability greater than δ', the inequality (16) is violated. To do so, we only need to show that 1) with high probability, $\mathbf{x}$ is *good* in the sense that it satisfies the desired properties; and 2) conditioned on the event that $\mathbf{x}$ is good, Equation (16) is violated with probability greater than σ', if we set $\gamma = p$ and $\sigma = 2^{-13}\cdot(1-\frac{3\delta'}{\beta})$, which leads to a contradiction.

Part 2) follows from the same argument as those in the proof of Lemma 5.3 in (Eliáš et al., 2020). For part 1), we describe our changes. Formally, we say that $\mathbf{x} \sim X_{|Y=\mathbf{y}}$ is *good*, if $\|\mathbf{A}\mathbf{x}-\mathbf{y}\|_\infty \leq \mathrm{disc}_{\mathcal{C}_{\sigma,\gamma}}(\mathbf{A})/2-1$ and the properties $(\star)$ given in the statement of Lemma 50 are satisfied.

Note that the property that $\mathbf{x} \sim X_{|Y=\mathbf{y}}$ is good is at least $(1-\frac{1}{\mathrm{poly}(n)})\cdot\beta$. This is true as our assumption that $G \sim G(n,p)$ and that $(\frac{\log n}{n})^{1/(h-1)} \ll p \leq \frac{1}{2}$, it holds that G satisfies the properties $(\star)$ with probability at least $1-\frac{1}{\mathrm{poly}(n)}$. This then finishes the proof of the lemma. ■

Finally, our lower bound Theorem 6 follows from the following lemma from (Eliáš et al., 2020), which in turn is built from Lemma 2.1.2 in (Bun, 2016).

Lemma 53 ((Eliáš et al., 2020)) *If there is no $(1,\delta)$-DP mechanism whose error is below $o\left(\overline{T}_G\cdot(1-\frac{9\delta}{\beta})\right)$, where $\overline{T}_G = \sqrt{mn}\cdot\ell_h(G)$, with probability β. Then there is no (ε,δ)-DP mechanism whose error is smaller than $o\left(\frac{\overline{T}_G}{\varepsilon}\cdot(1-c)\right)$ with probability β, where $c = \frac{e-1}{e^\varepsilon-1}\cdot\frac{9\delta}{\beta}$.*

Proof From Lemma 2.1.2 in (Bun, 2016), it holds that for any (ε,δ)-DP mechanism $\mathcal{M}$, if two graphs $\mathbf{w},\mathbf{w}'$ satisfy that $\|\mathbf{w}-\mathbf{w}'\|_1 \leq \frac{1}{\varepsilon}$, then for any output set S, $\Pr[\mathcal{M}(\mathbf{w})\in S] \leq e\Pr[\mathcal{M}(\mathbf{w}')\in S] + \frac{e-1}{e^\varepsilon-1}\delta$.

Now suppose there exists an (ε,δ)-DP mechanism $\mathcal{M}$ whose error is smaller than $o\left(\frac{\overline{T}_G}{\varepsilon}(1-c)\right)$ with probability β, where $c = \frac{e-1}{e^\varepsilon-1}\cdot\frac{9\delta}{\beta}$. Since $\ell_h(\frac{1}{\varepsilon}\cdot\mathbf{w}) = \frac{1}{\varepsilon^{\binom{h}{2}-1}}\ell_h(\mathbf{w})$, then $\overline{T}_{\frac{1}{\varepsilon}\cdot\mathbf{w}} = \frac{1}{\varepsilon^{\binom{h}{2}-1}}\overline{T}_{\mathbf{w}}$. Therefore, the mechanism $\varepsilon^{\binom{h}{2}}\cdot\mathcal{M}(\frac{1}{\varepsilon}\cdot\mathbf{w})$ is $(1,\frac{e-1}{e^\varepsilon-1}\delta)$-DP with additive error

$$\varepsilon^{\binom{h}{2}}\cdot o\left(\frac{\overline{T}_{\frac{1}{\varepsilon}\cdot\mathbf{w}}}{\varepsilon}(1-c)\right) \leq o\left(\overline{T}_{\mathbf{w}}\cdot(1-\frac{e-1}{e^\varepsilon-1}\cdot\frac{9\delta}{\beta})\right)$$

with probability at least β, which is a contradiction. ■

Proof [Proof of Theorem 6] By Lemma 8, Lemma 52 and Lemma 53, there is no (ε,δ)-DP mechanism that can release synthetic graphs preserving K_h-motif cuts for graph G with an additive error smaller than $o\left(\frac{\sqrt{mn}\cdot\ell_h(G)}{\varepsilon}\cdot(1-c)\right)$ with probability β, where $c = \frac{e-1}{e^\varepsilon-1}\cdot\frac{9\delta}{\beta}$.

Now consider the scaled version of G described in Theorem 6, where the weight vector $\mathbf{w}$ of G is scaled by a factor of $\frac{1}{\varepsilon}$, resulting in a new weight vector $\frac{1}{\varepsilon}\cdot\mathbf{w}$. Note that the total weight is

$W = \frac{m}{\varepsilon}$. When we let $\overline{T}_{\mathbf{w}} = \sqrt{Wn} \cdot \ell_h(G)$, then $\overline{T}_{\frac{1}{\varepsilon} \cdot \mathbf{w}} = \frac{1}{\varepsilon^{\binom{h}{2} - \frac{1}{2}}} \overline{T}_{\mathbf{w}}$. By a similar analysis to Lemma 53, we can establish the bound

$$\Omega\left(\max\left(\frac{\sqrt{mn} \cdot \ell_h(G)}{\varepsilon} \cdot (1-c), \frac{\sqrt{Wn} \cdot \ell_h(G)}{\varepsilon^{\frac{1}{2}}} \cdot (1-c)\right)\right).$$

■

Appendix F. Upper Bound of Randomized Response Method

In this section, we show that by using the randomized response, we can obtain an $(\varepsilon, 0)$-DP algorithm releasing a synthetic graph for triangle-motif cut queries. Specifically, given a graph G with vertex set V of size n and weight vector $\mathbf{w}$, we release the noisy weight vector $\tilde{\mathbf{w}}$ where $\tilde{\mathbf{w}}_e = \mathbf{w}_e + Y_e$ and each random variable $Y_e \sim \mathrm{Lap}(1/\varepsilon)$. Using the following analysis similar to (Gupta et al., 2012), we show that this algorithm achieves $\tilde{O}(n^{5/2})$ additive error for unweighted graph. Note that our bound of $O(\sqrt{m\ell_3(G)}n)$ is often much better than $\tilde{O}(n^{\frac{5}{2}})$.

Denote the query set $\mathcal{Q}_{\text{cuts}}$ as a collection containing all the triangle-motif cut queries on G. Since $\mathcal{Q}_{\text{cuts}}$ consists of all the (S, T) pairs, it has size at most 2^{2n}. Note that the queries in Q_{cuts} are not linear, unlike those in the edge cut setting discussed in (Gupta et al., 2012). Consequently, the Chernoff bound alone cannot be used to address the entire problem. In the following, we will see how Azuma's Inequality can be applied to overcome this challenge.

Note that each query q in $\mathcal{Q}_{\text{cuts}}$ can be view as a vector in $\{0,1\}^{\binom{|V|}{3}}$, namely,

$$q(\mathbf{w}) = \sum_{(i,j,k) \in \binom{V}{3}} q_{(i,j,k)} \mathbf{w}_{(i,j)} \mathbf{w}_{(j,k)} \mathbf{w}_{(k,i)}.$$

For convenience, we denote $\pi(i,j,k) = \{(i,j,k), (j,k,i), (k,i,j)\}$. Thus, for any $q \in \mathcal{Q}_{\text{cuts}}$, we have,

$$\begin{aligned}
& q(\tilde{\mathbf{w}}) - q(\mathbf{w}) \\
= & \sum_{(i,j,k) \in \binom{V}{3}} q_{(i,j,k)} \left((\mathbf{w}_{(i,j)} + Y_{(i,j)})(\mathbf{w}_{(j,k)} + Y_{(j,k)})(\mathbf{w}_{(k,i)} + Y_{(k,i)}) - \mathbf{w}_{(i,j)} \mathbf{w}_{(j,k)} \mathbf{w}_{(k,i)} \right) \\
= & \sum_{(i,j,k) \in \binom{V}{3}} q_{(i,j,k)} \left(Y_{(i,j)} Y_{(j,k)} Y_{(k,i)} + \sum_{(i_1,i_2,i_3) \in \pi(i,j,k)} \left(Y_{(i_1,i_2)} \mathbf{w}_{(i_2,i_3)} \mathbf{w}_{(i_3,i_1)} + Y_{(i_1,i_2)} Y_{(i_2,i_3)} \mathbf{w}_{(i_3,i_1)} \right) \right) \\
\leq & w_{\max}^2 \sum_{(i,j,k) \in \binom{V}{3}} q_{(i,j,k)} (Y_{(i,j)} + Y_{(j,k)} + Y_{(k,i)}) + w_{\max} \sum_{(i,j,k) \in \binom{V}{3}} q_{(i,j,k)} \sum_{(i_1,i_2,i_3) \in \pi(i,j,k)} Y_{(i_1,i_2)} Y_{(i_2,i_3)} \\
& + \sum_{(i,j,k) \in \binom{V}{3}} q_{(i,j,k)} Y_{(i,j)} Y_{(j,k)} Y_{(k,i)} \qquad (17)
\end{aligned}$$

By Lemma 15, with probability at least $1 - \beta/4$, we have that each of the absolute values $|Y_{(i,j)}|$'s is at most $L = O(1/\varepsilon \cdot \log(n/\beta))$. We then bound the three terms above conditioning on this event happening:

The first term $w_{\max}^2 \sum_{(i,j,k)\in\binom{V}{3}} q_{(i,j,k)}(Y_{(i,j)}+Y_{(j,k)}+Y_{(k,i)})$ in Equation (17) can be bounded by using a Chernoff bound. Specifically, since $\sum_{(i,j,k)\in\binom{V}{3}} q_{(i,j,k)}(Y_{(i,j)}+Y_{(j,k)}+Y_{(k,i)}) = \sum_{(i,j)\in\binom{V}{2}} \sum_{k\in V} q_{(i,j,k)} Y_{(i,j)}$, it holds that $\sum_{(i,j)\in\binom{V}{2}} \sum_{k\in V} q_{(i,j,k)} Y_{(i,j)}$ is the sum of $\binom{n}{2}$-many independent random variables with mean 0 and bounded by $[-nL, nL]$. Thus, by Chernoff bound,

$$\mathbf{Pr}\left[\left|\sum_{(i,j)\in\binom{V}{2}} \sum_{k\in V} q_{(i,j,k)} Y_{(i,j)}\right| \geq \alpha\right] \leq 2e^{\Omega(\frac{\alpha^2}{(nL)^2\binom{n}{2}})} = 2e^{\Omega(\frac{\alpha^2}{n^4L^2})}.$$

Therefore, by choosing $\alpha = O\left(\varepsilon^{-1} n^2 \log(n/\beta)\sqrt{\log(|\mathcal{Q}_{\text{cut}}|/\beta)}\right)$, we have that, with probability at least $1-\beta/4\,|\mathcal{Q}_{\text{cut}}|$,

$$\left|\sum_{(i,j)\in\binom{V}{2}} \sum_{k\in V} q_{(i,j,k)} Y_{(i,j)}\right| \leq O\left(\varepsilon^{-1} n^2 \log(n/\beta)\sqrt{\log(|\mathcal{Q}_{\text{cut}}|/\beta)}\right).$$

By a union bound, the probability of large deviations for any triangle-motif cut query is at most $\beta/4$.

The method to handle with the other two terms is slightly different from the analysis in (Gupta et al., 2012). Since they cannot be written as a sum of independent random variables, we cannot use Chernoff bounds. To deal with the independency, we construct martingales and use Azuma's inequality instead.

Lemma 54 (Azuma's Inequality, (Azuma, 1967)) *Let $X_0, \ldots, X_t$ be a martingale satisfying $|X_i - X_{i-1}| \leq c_i$ for any $i \in [t]$. Then for any $\alpha > 0$,*

$$\mathbf{Pr}\left[|X_t - X_0| \geq \alpha\right] \leq 2\exp\left(\frac{-\alpha^2}{2(\sum_{i=1}^t c_i^2)}\right)$$

We first consider the second term in Equation (17). The martingale considered is as follows: Given an arbitrary order of the $\binom{n}{2}$ pairs, namely, $e_1, \ldots, e_{\binom{n}{2}}$. For the sake of simplicity, we denote $\mathcal{E}_i$ as $\{e_1, e_2, \ldots, e_i\}$. Then we let $X_0 = 0$, and let

$$X_i = X_{i-1} + \sum_{(i_1,i_3)\in\mathcal{E}_{i-1}} q_{(i_1,i_2,i_3)} Y_{(i_1,i_2)} Y_{(i_1,i_3)} + \sum_{(i_2,i_3)\in\mathcal{E}_{i-1}} q_{(i_1,i_2,i_3)} Y_{(i_1,i_2)} Y_{(i_2,i_3)},$$

where we assume $e_i = (i_1, i_2)$. That is, intuitively, at time i, we add the pair (i_1, i_2) into $\mathcal{E}_{i-1}$, and add all the terms related to (i_1, i_2) between the existing pairs in $\mathcal{E}_{i-1}$ into X_{i-1}, similar to edge-exposure martingale. It is clear that $X_{\binom{n}{2}} = \sum_{(i,j,k)\in\binom{V}{3}} q_{(i,j,k)} \sum_{(i_1,i_2,i_3)\in\pi(i,j,k)} Y_{(i_1,i_2)} Y_{(i_2,i_3)}$, and X_t is a martingale since

$$\begin{aligned}
&\mathbb{E}\left[X_i | X_0, X_1, \ldots, X_{i-1}\right] \\
=&\mathbb{E}\left[X_i | e_1, e_2, \ldots, e_{i-1}\right] \\
=&X_{i-1} + \mathbb{E}\left[Y_{e_i} | Y_{e_1}, Y_{e_2}, \ldots, Y_{e_{i-1}}\right] \cdot \left(\sum_{(i_1,i_3)\in\mathcal{E}_{i-1}} q_{(i_1,i_2,i_3)} Y_{(i_1,i_3)} + \sum_{(i_2,i_3)\in\mathcal{E}_{i-1}} q_{(i_1,i_2,i_3)} Y_{(i_2,i_3)}\right)
\end{aligned}$$

$=X_{i-1}$.

Then since we can bound $|X_i - X_{i-1}|$ by $2nL^2$, by Azuma's Inequality, we have,

$$\mathbf{Pr}\left[\left|X_{\binom{n}{2}} - X_0\right| \geq \alpha\right] \leq 2\exp\left(\frac{-\alpha^2}{2\binom{n}{2}(2nL^2)^2}\right) = 2\exp\left(\frac{-\alpha^2}{8\binom{n}{2}n^2L^4}\right).$$

By choosing $\alpha = O\left(\varepsilon^{-2}n^2\log^2(n/\beta)\sqrt{\log(|\mathcal{Q}_{\text{cut}}|/\beta)}\right)$, we have that, with probability at least $1-\beta/4\,|\mathcal{Q}_{\text{cut}}|$,

$$\left|\sum_{(i,j,k)\in\binom{V}{3}} q_{(i,j,k)} \sum_{(i_1,i_2,i_3)\in\pi(i,j,k)} Y_{(i_1,i_2)}Y_{(i_2,i_3)}\right| \leq O\left(\varepsilon^{-2}n^2\log^2(n/\beta)\sqrt{\log(|\mathcal{Q}_{\text{cut}}|/\beta)}\right).$$

By a union bound, the probability of large deviations for any triangle-motif cut query is at most $\beta/4$.

For the third term in Equation (17), we can define the martingale as:

$$X_i = X_{i-1} + \sum_{(i_1,i_3),(i_2,i_3)\in\mathcal{E}_{i-1}} q_{(i_1,i_2,i_3)}Y_{(i_1,i_2)}Y_{(i_2,i_3)}Y_{(i_3,i_1)}.$$

By a similar analysis, we have,

$$\left|\sum_{(i,j,k)\in\binom{V}{3}} q_{(i,j,k)}Y_{(i,j)}Y_{(j,k)}Y_{(k,i)}\right| \leq O\left(\varepsilon^{-3}n^2\log^3(n/\beta)\sqrt{\log(|\mathcal{Q}_{\text{cut}}|/\beta)}\right)$$

for any triangle-motif cut query with probability at least $1-\beta/4$.

To sum up, with probability at least $1-\beta$, for any triangle-motif cut query $q \in \mathcal{Q}_{cuts}$, we have,

$$\begin{aligned}
&q(\tilde{\mathbf{w}}) - q(\mathbf{w}) \\
\leq& w_{\max}^2 \sum_{(i,j,k)\in\binom{V}{3}} q_{(i,j,k)}(Y_{(i,j)} + Y_{(j,k)} + Y_{(k,i)}) + w_{\max}\sum_{(i,j,k)\in\binom{V}{3}} q_{(i,j,k)} \sum_{(i_1,i_2,i_3)\in\pi(i,j,k)} Y_{(i_1,i_2)}Y_{(i_2,i_3)} \\
&+ \sum_{(i,j,k)\in\binom{V}{3}} q_{(i,j,k)}Y_{(i,j)}Y_{(j,k)}Y_{(k,i)} \\
\leq& O\left(w_{\max}^2\varepsilon^{-1}n^2\log(n/\beta)\sqrt{\log(|\mathcal{Q}_{\text{cut}}|/\beta)}\right) + O\left(w_{\max}\varepsilon^{-2}n^2\log^2(n/\beta)\sqrt{\log(|\mathcal{Q}_{\text{cut}}|/\beta)}\right) \\
&+ O\left(\varepsilon^{-3}n^2\log^3(n/\beta)\sqrt{\log(|\mathcal{Q}_{\text{cut}}|/\beta)}\right) \\
=& O\left(w_{\max}^2\varepsilon^{-1}\log(n/\beta) + w_{\max}\varepsilon^{-2}\log^2(n/\beta) + \varepsilon^{-3}\log^3(n/\beta)\right)\cdot\left(n^{5/2} + n^2\log(1/\beta)\right)
\end{aligned}$$

For unweighted graph, the corresponding bound is $O\left(\varepsilon^{-3}\log^3(n/\beta)\cdot\left(n^{5/2}+n^2\log(1/\beta)\right)\right) = \tilde{O}(n^{5/2})$.

Generally, for K_h-motif cut queries, we can release a synthetic graph by randomized response with additive error $O(\varepsilon^{-h}\log^h(n/\beta)\cdot\sqrt{\binom{n}{2}}\binom{n}{h-2}\sqrt{\log(|\mathcal{Q}_{\text{cut}}|/\beta)}) = \tilde{O}(n^{h-1/2})$.