# Pointwise Lipschitz Continuous Graph Algorithms

**Quanquan C. Liu**
Yale University
quanquan.liu@yale.edu

**Grigoris Velegkas**
Google Research
gvelegkas@google.com

**Yuichi Yoshida**
National Institute of Informatics
yyoshida@nii.ac.jp

**Felix Zhou**
Yale University
felix.zhou@yale.edu

## Abstract

In many real-world applications, it is undesirable to drastically change the problem solution after a small perturbation in the input as unstable outputs can lead to costly transaction fees, privacy and security concerns, reduced user trust, and lack of replicability. Despite the widespread application of graph algorithms, many classical algorithms are not robust to small input disturbances. Towards addressing this issue, we study the *pointwise Lipschitz continuity* of graph algorithms, a notion of stability introduced by Kumabe and Yoshida [2023, FOCS'23] and further studied in related settings [Kumabe and Yoshida, 2024, ICALP'24], [Kumabe and Yoshida, 2025, SODA'25], [Gima et al., 2025, ESA'25].

Our main result is a linear programming (LP) based minimum $S$-$T$ cut algorithm with a provably optimal Lipschitz constant, as witnessed by an accompanying lower bound. As a direct corollary, we give the *first* dynamic minimum $S$-$T$ cut algorithm with non-trivial recourse bound. At the core of our techniques is a novel framework for analyzing the Lipschitz constant of regularized LP relaxations. Our framework crucially unlocks the use of weighted regularizers, which could not be analyzed through previous methods and leads to polynomial improvements in the Lipschitz constant compared to what is achievable through previous techniques. To demonstrate the flexibility of our methods, we also design an LP-based $\boldsymbol{b}$-matching algorithm that improves on the state-of-the-art Lipschitz constant in certain input regimes when $\boldsymbol{b} \equiv 1$. Moreover, our algorithm cleanly extends to the general case when $\boldsymbol{b} \geq 1$, whereas prior works are specialized to the case of $\boldsymbol{b} \equiv 1$.

## 1 Introduction

The stability of algorithmic solutions is an important property for many real-world applications, where stability is generally defined as how much the solution to a problem changes when small perturbations to the input occur. This topic has long been studied by the algorithmic community in various domains including privacy Watson et al. [2020], Wang et al. [2016], replicability Impagliazzo et al. [2022], perturbation resilience Awasthi et al. [2012], Angelidakis et al. [2017], Balcan and Liang [2016], Bilu and Linial [2012], bounded recourse algorithms Davis et al. [2006], Gupta et al. [2017], Cohen-Addad et al. [2019], Bernstein et al. [2025], average sensitivity Yoshida and Zhou [2021], Kumabe and Yoshida [2022a], Varma and Yoshida [2021], Kumabe and Yoshida [2022b], Peng and Yoshida [2020], and Lipschitz continuous graph algorithms Kumabe and Yoshida [2023, 2025, 2024], Gima et al. [2025].

As a concrete example, consider the computer vision task of segmenting an object from the background [Greig et al., 1989, Boykov and Funka-Lea, 2006]. The user marks some pixels as "object" (source $S$) and some as "background" (sink $T$) and edges between adjacent pixels are weighted

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

inversely proportional to their similarity. We would like to find the boundary that separates the object seeds from the background seeds while minimizing the cost of the boundary, *i.e.*, compute the minimum $S-T$ cut. For critical downstream applications, stability of the solution is crucial because minor perturbations of the pixels should not drastically change the segmentation boundary.

In this work, our goal is to design graph algorithms that satisfy two desiderata: accuracy and stability. Notice that, without taking into account the quality of the solution, one can provide trivial algorithms that are perfectly stable under any perturbation of the input. However, algorithms that achieve perfect stability through overly simplistic methods (*e.g.*, always outputting the same value) often suffer from poor accuracy.

Towards this end, we study *pointwise Lipschitz continuity*, a notion of algorithmic stability introduced by Kumabe and Yoshida [2023].[1] Informally speaking, the Lipschitz constant of an algorithm is defined by considering the expected symmetric difference ($\ell_1$-distance) of the outputs of the algorithm under sufficiently small perturbations of edge-weights, where the expectation is taken with respect to the internal randomness of the algorithm. Specifically, we design pointwise Lipschitz algorithms for the minimum $S$-$T$ cut and $\boldsymbol{b}$-matching problems.

Our main algorithmic tool is regularized linear programming relaxations with natural choices of *weighted* regularizers to ensure the stability of the optimal fractional solution. Paired with an appropriate stable rounding scheme, this approach can yield *optimal* stability-approximation tradeoffs, as illustrated by a tight lower bound we provide for the minimum $S$-$T$ cut problem. As a direct corollary of our minimum $S$-$T$ cut algorithm, we achieve the first dynamic minimum $S$-$T$ cut algorithm with non-trivial recourse.

Similar to Kumabe and Yoshida [2023], we also add a smooth, strongly convex regularizer to the natural LP relaxation of the underlying graph problem. However, we depart from the direct application of first-order optimality conditions when analyzing the stability of optimal solutions and instead analyze the stability of iterates generated by the proximal gradient method, a first-order optimization method that is known to converge. This leads to a general framework for analyzing the Lipschitz constant of regularized LP relaxations we call the *proximal gradient trajectory analysis* (PGTA). Importantly, our bounds hold **regardless of the specific optimization algorithm employed to find the solution**, but we choose proximal gradient iterates as it is suitable for composite functions (*e.g.*, an objective function plus a regularizer). Crucially for the learning theory community, this framework bridges the gap between continuous optimization and discrete algorithmic stability.

Beyond leading to improved bounds for the minimum $S$-$T$ cut problem, PGTA unifies the analysis of the Lipschitz constant of convex relaxations for many problems under one clean and easy-to-use framework. For example, we design a bipartite matching algorithm that improves the stability-approximation tradeoffs compared to the Kumabe and Yoshida [2023] bipartite matching algorithm for certain input regimes. More importantly, our algorithm and its analysis cleanly generalizes to the case of bipartite $\boldsymbol{b}$-matchings when $\boldsymbol{b} \geq 1$, whereas the Kumabe and Yoshida [2023] algorithm is hardwired for 1-matchings. To further demonstrate the flexibility of our approach, we also design a pointwise Lipschitz algorithm for packing integer programs.

### 1.1 Summary of Results

Before we discuss our results, we give an informal description of the pointwise Lipschitz constant of a graph algorithm Kumabe and Yoshida [2023]. For the formal definition, we refer the reader to Section 3.1. We say that a randomized algorithm $\mathcal{A}$ that operates on an $n$-vertex $m$-edge graph $G = (V, E)$ with weights $\boldsymbol{w} \in \mathbb{R}^E_{\geq 0}$ is pointwise Lipschitz continuous with pointwise Lipschitz constant $P_{G,\boldsymbol{w}}$ if the earth mover's distance (EMD) between the output distributions of $\mathcal{A}$ on inputs $(G, \boldsymbol{w})$ and $(G, \tilde{\boldsymbol{w}})$ is at most $P_{G,\boldsymbol{w}} \cdot \|\boldsymbol{w} - \tilde{\boldsymbol{w}}\|_1$. Here, $\tilde{\boldsymbol{w}}$ is assumed to be in a sufficiently small neighborhood of $\boldsymbol{w}$. Crucially, the pointwise Lipschitz constant is allowed to depend on the structure of the graph $G$ and the particular weight vector $\boldsymbol{w}$. The reason for this becomes more apparent in our discussion of the formal definition in Section 3.1.1.

Our algorithms produce $(\zeta, \alpha)$-bicriteria approximations where $\zeta$ is the multiplicative factor in the approximation and $\alpha$ is the additive error. We write $\zeta$-approximation to indicate a $(\zeta, 0)$-approximation

[1] To be more precise, Kumabe and Yoshida [2023] introduced *two* related notions of Lipschitz continuity, the *weighted* and *unweighted* one. They focused mostly on designing algorithms that satisfy the weighted version, whereas we extensively study the unweighted version; See Section 2 for a detailed comparison.

algorithm as a shorthand.

**Minimum Vertex $S$-$T$ Cut.**

1. Let $S, T \subseteq V$ be disjoint vertex subsets. We write $\partial A \coloneqq \{uv \in E : u \in A, v \notin A\}$. The *minimum vertex $S$-$T$ cut* problem asks for a vertex subset $A \supseteq S$ that minimizes the cut weight $\boldsymbol{w}(\partial A) = \sum_{e \in \partial A} w_e$ subject to the constraint $A \cap T = \varnothing$. Note that this is a generalization of the minimum vertex $s$-$t$ cut problem. We design a Lipschitz continuous algorithm for this problem, accompanied by a tight lower bound on the Lipschitz constant.
2. We also design a Lipschitz continuous algorithm for the minimum $\beta$-balanced vertex $S$-$T$ cut problem, which is a variant of the minimum vertex $S$-$T$ cut problem where we additionally restrict the feasible cut sets $A$ to have at least $\beta n$ vertices and at most $(1-\beta)n$ vertices.

We summarize our results for the minimum vertex $S$-$T$ cut in Table 1.

| Problem | Approximation | Pointwise Lipschitz Constant | Reference |
|---|---|---|---|
| Minimum Vertex $S$-$T$ Cut | $(1+o(1)) \cdot \mathrm{OPT} + O(\lambda_2)$ | $O\left(\frac{n}{\lambda_2}\right)$ | Theorem 13 |
| | $O(1) \cdot \mathrm{OPT} + O(\lambda_2)$ | $\Omega\left(\frac{n}{\lambda_2}\right)$ | Theorem 16 |
| Minimum $\beta$-Balanced Vertex $S$-$T$ Cut | $O(\beta^{-1}) \cdot \mathrm{OPT}_\beta$ | $O\left(\frac{\sqrt{n}}{\beta^2 \lambda_2}\right)$ | Theorem 15 |

Table 1: Our results. For graph problems, $n$ and $m$ denote the number of vertices and edges, respectively. For the minimum balanced $S$-$T$ cut problem, $\mathrm{OPT}_\beta$ is the minimum cut weight over vertex subsets of size at least $\beta n$ and at most $(1-\beta)n$. $\lambda_2$ is the second smallest eigenvalue of the input graph.

**Bipartite $\boldsymbol{b}$-Matching.**

1. Recall the maximum weight (bipartite) matching problem where we are given a (bipartite) graph and want to output an edge set $M \subseteq E$ such that every vertex is adjacent to at most one edge of $M$ while maximizing the matching weight $\boldsymbol{w}(M) = \sum_{e \in M} w_e$. We design a Lipschitz continuous algorithm for this problem.
2. We also design a Lipschitz continuous algorithm for the maximum weight (bipartite) $\boldsymbol{b}$-matching problem, a variation of the maximum weight (bipartite) matching problem, where each vertex $v$ can be adjacent to at most $b_v$ edges of the matching. Note the case of $\boldsymbol{b} \equiv 1$ recovers the maximum weight (bipartite) matching problem.

We summarize our results for maximum bipartite matching and its variants in Table 2. Kumabe and

| Problem | Approximation | Pointwise Lipschitz Constant | Reference |
|---|---|---|---|
| Bipartite Matching | $\frac{1}{2+\varepsilon} \cdot \mathrm{OPT}$ | $O\left(\frac{\sqrt{m}}{\varepsilon w_{\min}}\right)$ | Theorem 29 |
| Bipartite $\boldsymbol{b}$-Matching | $\frac{1}{3.164+\varepsilon} \cdot \mathrm{OPT}$ | $O\left(\frac{\sqrt{m}}{\varepsilon w_{\min}}\right)$ | Theorem 30 |

Table 2: Our results for weighted ($\boldsymbol{b}$-)matching. $\varepsilon > 0$ denotes the approximation parameter. $n$ and $m$ denote the number of vertices and edges, respectively. $w_{\min}$ denotes the minimum weight of an edge.

Yoshida [2023] obtain a $(2+\varepsilon)$-approximate maximum weight bipartite matching with pointwise Lipschitz constant $O\left(\varepsilon^{-1} n^{3/2} \log(m)/\mathrm{OPT}\right)$. In cases where the graph is sparse, *i.e.*, $m = \tilde{O}(n)$, the number of edges in the optimum matching is small, *e.g.*, $o(n)$, and the edge weights are similar in magnitude, our algorithm improves on their pointwise Lipschitz bounds.

## 2 Related Works

Our work falls under the broad literature of algorithmic stability, and there are many related works. We discuss the most relevant here and defer additional discussion to Appendix B

**Weighted Lipschitz Continuity vs. Pointwise Lipschitz Continuity.** The definition of Lipschitz

continuous graph algorithms has been given in two settings. In the first setting, Lipschitz constants are measured by considering the *weighted* $\ell_1$ distance of the characteristic vectors of the outputs under perturbations of the input. In the second setting, pointwise Lipschitz constants are measured by considering the *unweighted* $\ell_1$ distance of the characteristic vectors of the outputs. To make the distinction between the two concepts clear, we will generally refer to the former as the *weighted* Lipschitz constant and the latter as the *unweighted* or *pointwise* Lipschitz constant.

While weighted Lipschitz continuous graph algorithms with bounded weighted Lipschitz constants have been studied to a greater extent Kumabe and Yoshida [2023, 2025], Gima et al. [2025], pointwise Lipschitz continuous graph algorithms have largely been unexplored.[2] Previous work [Kumabe and Yoshida, 2023] studied the pointwise Lipschitz constant for only two problems:[3] the minimum spanning tree problem and the maximum weight bipartite matching problem, for which they developed a combinatorial and a regularized LP-based method, respectively. See Section 4 for more detailed comparison with previous regularized LP-based techniques.

**Online/Dynamic Algorithms & Recourse.** Low recourse online/dynamic algorithms (see *e.g.*, Arar et al. [2018], Henzinger et al. [2025], Megow and Nölke [2025], Gupta and Levin [2020], Bhattacharya et al. [2022], Davis et al. [2006], Gupta et al. [2017], Lattanzi and Vassilvitskii [2017], Jambulapati et al. [2022], Chen et al. [2025] and references therein) are algorithms where updates to the graph are given in an online manner, and the goal is to minimize the *recourse* or the change in the output between consecutive days over $T$ total days. For min $s$-$t$ cut, there are dynamic cut sparsifiers with bounded recourse Goranci et al. [2021], Chen et al. [2020], which may lead to $s$-$t$ cut algorithms with small recourse. However, we are not aware of any such explicit recourse bounds.

It is known that re-running any $\alpha$-approximate (pointwise) Lipschitz algorithm with (pointwise) Lipschitz constant $K$ each day from scratch immediately translates to an online $\alpha$-approximation algorithm where the total expected recourse is bounded by $KT$ (over all days) [Kumabe and Yoshida, 2023, Theorem 1.6].

**Average Sensitivity.** Varma and Yoshida [2021] introduced the notion of *average* sensitivity, which can be seen as an analogue of pointwise Lipschitz continuity for the unweighted setting. Namely, the *sensitivity* of a randomized algorithm $\mathcal{A}$ that outputs a set $S$ given the input graph $G = (V, E)$ is defined to be the maximum earth mover's distance between $\mathcal{A}(G)$ and $\mathcal{A}(G - e)$ over $e \in E$, where $G - e$ is the graph obtained from $G$ by deleting $e$, and $\mathcal{A}(G)$ and $\mathcal{A}(G - e)$ represent the output distributions of $\mathcal{A}$ on $G$ and $G - e$, respectively. In that definition, the Hamming distance is used as the underlying metric for calculating the earth mover's distance. Since its introduction, the average sensitivity of algorithms for various problems have been studied Li et al. [2025], Peng and Yoshida [2020], Kumabe and Yoshida [2022b,a], Yoshida and Zhou [2021].

All of our algorithms translate to the average sensitivity setting. Indeed, the topology of an unweighted graph can be described by a $\{0, 1\}$-valued vector on a complete graph, so we can bound the average sensitivity of an algorithm by taking the supremum of the pointwise Lipschitz constant over weight vectors between the two $\{0, 1\}$-valued vectors corresponding to $G$ and $G - e$ (see Appendix G.3).

## 3 Preliminaries

In this section, we define the notations and definitions we use in our paper.

**Notation.** We first define the notation we use throughout our work. $\boldsymbol{w}^{(j)}, \boldsymbol{w}^j$ denotes a vector indexed by $j$. $w_i$ is the $i$-th coordinate of $\boldsymbol{w}$. $\tilde{\boldsymbol{w}}$ denotes a vector obtained by perturbing $\boldsymbol{w}$ by some perturbation vector $\boldsymbol{\delta}$ with $\ell_1$-norm $\sum_i |\delta_i| > 0$. $w_{\min}, w_{\max}$ is the min/max element of vector $\boldsymbol{w}$. $\mathcal{A}$ denotes a (randomized) algorithm and $\mathcal{A}_\pi$ the deterministic algorithm obtained from $\mathcal{A}$ by fixing its internal randomness $\pi$. $A \Delta B \coloneqq (A \cup B) \setminus (A \cap B)$ is the symmetric difference. $\mathbb{1}_S$ denotes the indicator variable of set $S$.

Moreover, we say that $\mathcal{A}$ gives an $(\zeta, \alpha)$-approximation guarantee to a minimization problem if its output has value at most $\zeta \cdot \mathrm{OPT} + \alpha$, or for a maximization problem, at least $1/\zeta \cdot \mathrm{OPT} - \alpha$. When there is no additive error $\alpha$, we simply say that $\mathcal{A}$ gives an $\zeta$-approximation guarantee.

[2] In general, there is no translation between the weighted and pointwise Lipschitz constant of an algorithm due to their qualitative differences. See Section 3.1.1 for more details.

[3] Although another work Kumabe and Yoshida [2024] also studied the unweighted Lipschitz definition, they do so in the context of fair allocations in cooperative game theory, a different setting from what we consider.

**Graph-Theoretic Notions.** Let $G = (V, E)$ be a graph and $\boldsymbol{w} \in \mathbb{R}^E_{\geq 0}$ be a vector that represents the edge weights. For a vertex $v \in V$, let $d_{G,\boldsymbol{w}}(v) \coloneqq \sum_{e \in E : v \in e} w_e$ be the *weighted degree* of $v$ and $d^u_G(v) \coloneqq |\{w \in V : uw \in E\}|$ be the *unweighted degree* of $v$. For an edge set $F \subseteq E$, let $\boldsymbol{w}(F) \coloneqq \sum_{e \in F} w_e$. The cut weight of $S \subseteq V$ is defined to be $\mathrm{cut}_{G,\boldsymbol{w}}(S) \coloneqq \boldsymbol{w}(\partial_G S)$, where $\partial_G S \coloneqq E(S, V \setminus S)$ (*i.e.*, the edges going between $S$ and $V \setminus S$). We may drop subscripts if they are clear from the context.

### 3.1 Pointwise Lipschitz Continuity

Let $d : 2^V \times 2^V \to \mathbb{R}_+$ denote a metric on the output space of a randomized graph algorithm $\mathcal{A}(G, \boldsymbol{w})$. For example, if $\mathcal{A}$ outputs subsets of vertices, then one choice is to take $d$ to be the $\ell_1$-metric $d(V_1, V_2) = |V_1 \Delta V_2| = \|\mathbb{1}_{V_1} - \mathbb{1}_{V_2}\|_1$. Unless otherwise stated, we take $d$ to be the $\ell_1$ metric. We write $(\pi, \tilde{\pi}) \sim \mathcal{D}$ to denote a coupling[4] between the random variables $\pi, \tilde{\pi}$. Recall the *earth mover's distance*[5] [Rubner et al., 1998] on the output distributions of $\mathcal{A}(G, \boldsymbol{w})$ is defined to be $\mathrm{EMD}_d(\mathcal{A}(G, \boldsymbol{w}), \mathcal{A}(G, \tilde{\boldsymbol{w}})) \coloneqq \inf_{\mathcal{D}} \mathbb{E}_{(\pi,\tilde{\pi}) \sim \mathcal{D}} d\left(\mathcal{A}_\pi(G, \boldsymbol{w}), \mathcal{A}_{\tilde{\pi}}(G, \tilde{\boldsymbol{w}})\right)$. Kumabe and Yoshida [2023] introduced the following notion of (pointwise) Lipschitz continuity.

**Definition 1** (Kumabe and Yoshida [2023])**.** Suppose the (randomized) algorithm $\mathcal{A}(G, \boldsymbol{w})$ outputs subsets of some universe $\mathfrak{S}$. Let $d : 2^{\mathfrak{S}} \times 2^{\mathfrak{S}} \to \mathbb{R}_+$ be a metric on $2^{\mathfrak{S}}$. We say that $\mathcal{A}$ has *pointwise Lipschitz constant $c_{\boldsymbol{w}}$ with respect to $d$* if

$$\limsup_{\tilde{\boldsymbol{w}} \to \boldsymbol{w}} \frac{\mathrm{EMD}_d(\mathcal{A}(G, \boldsymbol{w}), \mathcal{A}(G, \tilde{\boldsymbol{w}}))}{\|\boldsymbol{w} - \tilde{\boldsymbol{w}}\|_1} \leq c_{\boldsymbol{w}}.$$

Note here that we identify subsets $S \subseteq \mathfrak{S}$ with their indicator variables $\mathbb{1}_S \in \{0, 1\}^{\mathfrak{S}}$.

#### 3.1.1 Remarks on Definition 1

As mentioned in Section 2, Kumabe and Yoshida [2023] defined two notions of Lipschitz continuity: weighted Lipschitz continuity and unweighted pointwise Lipschitz continuity. From hereonforth, **we write Lipschitz continuity/constant to mean pointwise Lipschitz continuity/constant** for simplicity.

At first glance, it may seem difficult to compute the Lipschitz constant for perturbations of more than one entry, as necessitated by Definition 1. Kumabe and Yoshida [2023] showed the following lemma, which essentially says that it suffices to analyze the Lipschitz constant under the perturbation of a single entry of the weight vector.

**Lemma 2** (See also Lemma 7.1 of Kumabe and Yoshida [2023])**.** *Let $\delta > 0$ and $\tilde{\boldsymbol{w}}$ be obtained from $\boldsymbol{w}$ by choosing $\delta_{e_i} \in \mathbb{R}$ for each $i \in [m]$ such that $\sum_{i \in [m]} |\delta_{e_i}| = \delta$ and setting $\tilde{w}_{e_i} \coloneqq w_{e_i} + \delta_{e_i}$. Let $\boldsymbol{w}^{(i)} \coloneqq \boldsymbol{w} + \sum_{j=1}^{i} \delta_{e_j} \mathbb{1}_{e_j}$ so that $\boldsymbol{w}^{(0)} = \boldsymbol{w}$ and $\boldsymbol{w}^{(m)} = \tilde{\boldsymbol{w}}$. Suppose for every $i \in [m]$, $\mathrm{EMD}_d\left(\mathcal{A}(G, \boldsymbol{w}^{(i-1)}), \mathcal{A}(G, \boldsymbol{w}^{(i)})\right) \leq c_{\boldsymbol{w}} |\delta_{e_i}|$, where $c_{\boldsymbol{w}}$ is a function of only $\boldsymbol{w}$. Then*

$$\frac{\mathrm{EMD}_d(\mathcal{A}(G, \boldsymbol{w}), \mathcal{A}(G, \tilde{\boldsymbol{w}}))}{\|\boldsymbol{w} - \tilde{\boldsymbol{w}}\|_1} \leq c_{\boldsymbol{w}}.$$

As the intermediary step of our algorithms, we consider algorithms with *fractional* pointwise Lipschitz constant. That is, $\mathcal{A}(G, \boldsymbol{w})$ outputs a (deterministic) vector in $\mathbb{R}^{\mathfrak{S}}$ such that the following quotient is bounded in the limit:

$$\frac{d(\mathcal{A}(G, \boldsymbol{w}), \mathcal{A}(G, \tilde{\boldsymbol{w}}))}{\|\boldsymbol{w} - \tilde{\boldsymbol{w}}\|_1}.$$

Here, $d : \mathbb{R}^{\mathfrak{S}} \times \mathbb{R}^{\mathfrak{S}} \to \mathbb{R}_+$ is a metric over the vector space $\mathbb{R}^{\mathfrak{S}}$. We can interpret such an output vector as a point in the convex hull of indicator vectors of $2^{\mathfrak{S}}$. Note that in the case of fractional Lipschitz algorithms, it suffices to consider deterministic algorithms.

Finally, we remark that all our algorithms can be implemented with shared randomness. See Appendix F for details.

[4] Recall $\mathcal{D}$ is a joint distribution whose marginals on the first and second coordinates are $\pi, \tilde{\pi}$, respectively.

[5] This metric is also known as the *Wasserstein 1-distance* [Villani, 2009], *Kantorovich-Rubinstein metric* [Kantorovich, 1960], or *Mallows' 1-distance* [Mallows, 1972, Levina and Bickel, 2001].

# 4 PGTA Framework

Our first main contribution is a general framework bounding the Lipschitz constant of regularized convex programs. Our approach builds on the LP-based technique of Kumabe and Yoshida [2023], who used regularized LP relaxations to obtain stable fractional solutions as an important intermediate step. However, our analysis departs from their direct analysis using first-order optimality conditions and more subtly controls the Lipschitz constant by analyzing the iterates of a well-behaved process that converges to the optimal solution.

**Definition 3** (Proximal Gradient Method)**.** The *proximal operator* $\mathrm{prox}_{L,f} : \mathbb{R}^n \to K$ for a composite function $h(\boldsymbol{x}) \coloneqq f(\boldsymbol{x}) + g(\boldsymbol{x})$, is given by $\mathrm{prox}_{L,f}(\boldsymbol{x}) \coloneqq \mathrm{argmin}_{\boldsymbol{z}\in K} \frac{L}{2}\|\boldsymbol{z}-\boldsymbol{x}\|_2^2 + f(\boldsymbol{z})$. Then, given an initial point $\boldsymbol{x}^0 \in \mathbb{R}^n$, PGM with step size $1/L$ iteratively applies the update rule: $\boldsymbol{x}^{t+1} = \mathrm{prox}_{L,f}\left(\boldsymbol{x}^t - \frac{1}{L}\boldsymbol{\nabla} g(\boldsymbol{x}^t)\right)$.

**Theorem 4** (Informal; see Theorem 10)**.** *Let $h_{\boldsymbol{w}}(x) = f_{\boldsymbol{w}}(x) + g_{\boldsymbol{w}}(x)$ be a composite function parameterized by weight vector $\boldsymbol{w}$, where $f_{\boldsymbol{w}} : K \to \mathbb{R}$ is convex and $g_{\boldsymbol{w}} : K \to \mathbb{R}$ is $\sigma_{\boldsymbol{w}}$-strongly convex and $L_{\boldsymbol{w}}$-smooth. Moreover, let $\tilde{\boldsymbol{w}}$ be obtained from $\boldsymbol{w}$ by perturbing one coordinate by $\delta > 0$. Suppose $\|\boldsymbol{\nabla} g_{\boldsymbol{w}} - \boldsymbol{\nabla} g_{\tilde{w}}\|_2 = O(L_{\boldsymbol{w}} \cdot \delta)$ and $\|\mathrm{prox}_{L_{\boldsymbol{w}},f_{\boldsymbol{w}}} - \mathrm{prox}_{L_{\boldsymbol{w}},f_{\tilde{\boldsymbol{w}}}}\|_2 = O(\delta)$. Then the optimal solutions $\boldsymbol{x}^*, \tilde{\boldsymbol{x}}^*$ corresponding to $h_{\boldsymbol{w}}$ and $h_{\tilde{\boldsymbol{w}}}$ satisfy $\|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}^*\|_2 = O\left(\frac{L_{\boldsymbol{w}}}{\sigma_{\boldsymbol{w}}}\delta\right)$.*

The formal statement and proof of Theorem 4 is deferred to Appendix C. Before outlining our approach in Section 4.2, we sketch the techniques from prior works that bound the Lipschitz constant of regularized convex programs in Section 4.1.

## 4.1 Prior Works & Challenges

Before proceeding further, we outline the approach of Kumabe and Yoshida [2023] for controlling the Lipschitz constant. Let $h(\boldsymbol{x}) = f(\boldsymbol{x}) + \Lambda g(\boldsymbol{x})$ where, for simplicity, $g$ is a 1-strongly convex regularizer and $\Lambda > 0$ is the regularization parameter. We use $\tilde{h}, \tilde{f}, \Lambda\tilde{g}$ to denote the perturbed objective functions. We write $\boldsymbol{x}^\star, \tilde{\boldsymbol{x}}^\star$ to be the optimal solutions of the original and perturbed instances, respectively. By using first-order optimality conditions as well as the definition of strong convexity, it can be shown that (see *e.g.*, Chen et al. [2023, Section 5]) $\Lambda\|\boldsymbol{x}^\star - \tilde{\boldsymbol{x}}^\star\|^2 \le h(\tilde{\boldsymbol{x}}^\star) - h(\boldsymbol{x}^\star) + \tilde{h}(\boldsymbol{x}^\star) - \tilde{h}(\tilde{\boldsymbol{x}}^\star)$. When $g = \tilde{g}$ and $f, \tilde{f}$ are linear, Kumabe and Yoshida [2023] observed that $\Lambda\|\boldsymbol{x}^\star - \tilde{\boldsymbol{x}}^\star\|^2 \le f(\tilde{\boldsymbol{x}}^\star) - f(\boldsymbol{x}^\star) + \tilde{f}(\boldsymbol{x}^\star) - \tilde{f}(\tilde{\boldsymbol{x}}^\star) \le \delta\|\boldsymbol{x}^\star - \tilde{\boldsymbol{x}}^\star\|$ or $\|\boldsymbol{x}^\star - \tilde{\boldsymbol{x}}^\star\| \le \frac{\delta}{\Lambda}$. Here $\delta$ is the magnitude of perturbation. Crucially, in order to obtain a bound that scales *linearly* with the weight perturbation $\delta$, this analysis relies on using the same regularizer $\tilde{g} = g$ on the perturbed instance. Unfortunately, using unweighted regularizers yields sub-optimal Lipschitz constants for minimum vertex $S$-$T$ cut (see Section 5.1) and does not allow us to extend the matching algorithm of Kumabe and Yoshida [2023], which relies on the entropy regularizer, to the natural generalization of $\boldsymbol{b}$-matchings.

## 4.2 Our Approach

First, we consider an appropriate convex relaxation of the underlying graph problem on a graph with edge weights $\boldsymbol{w}$. Then, we add a *$\sigma$-strongly convex* and *$L$-smooth* regularizer to our objective. Subsequently, we show that the optimal solution corresponding to any slightly perturbed edge weights $\tilde{\boldsymbol{w}}$ will be close to the optimal solution corresponding to our original edge weights $\boldsymbol{w}$. Our proof is inspired by the work of Hardt et al. [2016] and goes through the proximal gradient trajectory analysis (PGTA), where we show that the distance between the trajectories of the proximal gradient method (Definition 3) under $\boldsymbol{w}$ and $\tilde{\boldsymbol{w}}$ scales linearly with $\|\boldsymbol{w} - \tilde{\boldsymbol{w}}\|_1$. While Hardt et al. [2016] also show that various first-order algorithms are stable, their notion of stability has a more discrete flavor and is defined on neighboring datasets that differ in one entry, whereas we require our algorithms to be continuously stable across instances parameterized by weights $\boldsymbol{w}$ and $\tilde{\boldsymbol{w}}$.

Roughly speaking, for given initial points $x^{(0)}$, PGM generates a sequence of points $x^{(t+1)} = G(x^{(t)})$ that converge to the optimal solution $x^{(t)} \to x^\star$ as $t \to \infty$, where $G$ is an update operator. As we will see, $G, \tilde{G}$ are contractions and are sufficiently well-behaved across perturbed instances so that

$$\begin{aligned}\|G(x^{(t)}) - \tilde{G}(\tilde{x}^{(t)})\| &\le (1 - \sigma/L)\|x^{(t)} - \tilde{x}^{(t)}\| + O(\|\boldsymbol{w} - \tilde{\boldsymbol{w}}\|_1) \\ &\le (1-\sigma/L)^t\|x^{(0)} - \tilde{x}^{(0)}\| + \sum_{\tau=1}^{t}(1-\sigma/L)^\tau \cdot O(\|\boldsymbol{w} - \tilde{\boldsymbol{w}}\|_1) \le O\left(\frac{\|\boldsymbol{w} - \tilde{\boldsymbol{w}}\|_1}{\sigma/L}\right).\end{aligned}$$

Taking the limit as $t \to \infty$ yields a perturbation bound on the optimal fractional solutions $x^\star, \tilde{x}^\star$. Crucially, our analysis relies only on the $\sigma$-strong convexity and $L$-smoothness of the regularizer and does *not* require the regularizer to be identical on perturbed instances. This opens the door to using weighted regularizers, which are the key to obtaining tight pointwise Lipschitz guarantees for minimum vertex $S$-$T$ cut and extending our matching algorithm to $\boldsymbol{b}$-matchings.

Finally, we need to round the continuous solution of the relaxed program to a feasible discrete one in a way that (approximately) maintains both the quality of the objective and the pointwise Lipschitz guarantee. The rounding schemes we use are problem-specific and are usually the most technically involved steps of our approach.

# 5 Minimum Vertex $S$-$T$ Cut

Let $G = (V, E)$ be an undirected graph with edge weights $\boldsymbol{w} \in \mathbb{R}^E_{\geq 0}$ and $S, T \subseteq V$ be disjoint vertex subsets. For $A \subseteq V$, we write $\partial A \coloneqq \{uv \in E : u \in A, v \notin A\}$. The *minimum vertex $S$-$T$ cut* problem asks for a vertex subset $A \supseteq S$ such that $A \cap T = \varnothing$ minimizing the cut weight $\boldsymbol{w}(\partial A) = \sum_{e \in \partial A} w_e$. Note that this is a generalization of the minimum vertex $s$-$t$ cut problem.

Our main result for $S - T$ cuts is the following.

**Theorem 5** (Informal; See Theorem 13, Theorem 16)**.** *There is an efficient $(1 + O(1/\sqrt{n}), O(\lambda_2))$-approximation algorithm for minimum $S$-$T$ cut with pointwise Lipschitz constant $O(n/\lambda_2)$, where $\lambda_2$ is the second smallest eigenvalue of the unnormalized Laplacian matrix of the input graph. Moreover, any $(O(1), O(\lambda_2))$-approximation algorithm must have pointwise Lipschitz constant $\Omega(n/\lambda_2)$.*

As a direct corollary, we obtain the first dynamic $S - T$ cut algorithm with non-trivial support. See Corollary 14. We also design an algorithm for the balanced $S - T$ cut problem.

**Theorem 6** (Informal; See Theorem 15)**.** *There is an efficient $O(1/\beta)$-approximation algorithm to the $\beta$-balanced $S$-$T$ cut problem with pointwise Lipschitz constant $O\left(\frac{\sqrt{n}}{\beta^2 \lambda_2}\right)$, where $\lambda_2$ is the second smallest eigenvalue of the unnormalized Laplacian matrix of the input graph.*

The formal statement and proof of Theorems 5 and 6 is presented in Appendix D. We briefly outline existing techniques and their limitations in Section 5.1 before overviewing our algorithmic approach in Section 5.2. Finally, we describe a matching lower bound in Section 5.3.

## 5.1 Existing Techniques & Limitations

Before summarizing our algorithm, we work through the approach of prior works.

As mentioned in Section 4.1, Kumabe and Yoshida [2023] directly uses the definition of strong convexity to control the change in output in response to $\ell_1$ input perturbations. If we were to directly apply the analysis of Kumabe and Yoshida [2023] for the minimum vertex $S$-$T$ cut problem, we would need to choose an unweighted regularizer, of which the $\ell_2$-regularizer $\|\cdot\|_2^2$ is the most canonical. As shown in Appendix H, following the steps of previous work would yield a fractional Lipschitz algorithm with $O(\Lambda n)$ additive error (no multiplicative error) and a Lipschitz constant of $O(n/\Lambda)$ for some regularization parameter $\Lambda$. In order to match the additive error of $O(\lambda_2)$ in Theorem 5, we must set the regularization parameter to $\Lambda = \lambda_2/n$. However, this yields a Lipschitz constant of $n^2/\lambda_2$, which is strictly worse than the guarantees of Theorem 5.

This bottleneck seems to be inherent to their analysis rather than a specific limitation of the unweighted $\ell_2$-regularizer. Indeed, the analysis of Kumabe and Yoshida [2023] suffers an additive error proportional to the maximum range of the chosen regularizer. Meanwhile, it is known that any $\Lambda$-strongly convex function over $[-1, 1]^n$ (even with respect to the $\ell_\infty$-norm) must have range $\Omega(\Lambda n)$. This phenomenon, informally known as the "$\ell_\infty$-barrier", was responsible for the previous stalled progress on several important optimization problems [Sherman, 2017, Sidford and Tian, 2018]. PGTA sidesteps this barrier as the derived fractional Lipschitz constant does not directly depend on the range of the regularizer.

## 5.2 Our Approach

Our algorithm first solves a standard minimum $S$-$T$ cut LP relaxation regularized by the quadratic form of the unnormalized Laplacian (Appendix D.1). We can then directly bound the Lipschitz

constant of the optimal fractional solution by applying the PGTA framework. Let $\lambda_2$ be the *algebraic connectivity* of the input graph, that is, the second smallest eigenvalue of the unnormalized Laplacian of the graph. Roughly speaking, the optimal fractional solution is a $(1+\varepsilon)$-approximation with Lipschitz constant $O(1/\varepsilon\lambda_2)$, where $\varepsilon > 0$ is a tunable parameter. Given stable fractional solutions, we can then apply the classical threshold rounding to obtain integral solutions.

One technical complication we need to handle, is that in order to apply our framework, we need to ensure that the objective function is strongly convex. Thus, due to the natural choice of the Laplacian regularizer, we need to optimize over a feasible region that is orthogonal to the all-ones vector. In particular, the feasible region is a subset of $[-1,1]^n$ rather than the standard $[0,1]^n$. As a result, we must perform threshold rounding with a random threshold in $[-1,1]$, which returns a non-trivial vertex set with probability merely $1/2$ and we must take additional steps to boost the success probability. Notice that, in the absence of the pointwise Lipschitz continuity requirement, boosting the success probability of such an algorithm is straightforward; we can just run multiple independent instances of the algorithm and, with high probability, we will obtain a feasible output. The key difficulty is smoothly selecting the feasible output, as it might be completely different in two executions of the algorithm under perturbed weights.

We design two different stable boosting algorithms: the first utilizes the exponential mechanism [Varma and Yoshida, 2021] (modified from McSherry and Talwar [2007]) to smoothly select a feasible output from independent executions and the second circumvents the selection issue by taking advantage of the *$k$-way submodularity property* [Harvey et al., 2006] of the objective function to *combine* solutions. We believe that our boosting mechanisms may be of independent interest. This leads to two algorithms for the minimum vertex $S$-$T$ cut problem.

**Exponential Mechanism Boosting (Appendix D.2).** As mentioned, vanilla threshold rounding gives a feasible solution with probability merely $1/2$. To achieve a higher success probability, we first compute a fractional solution assuming that the optimal cut (as a vertex set) has size around $\gamma i \cdot n$ for each $i \in \{1, \dots, 1/\gamma\}$, where $\gamma > 0$ is a small constant, and then apply the exponential mechanism to select the index $i^*$. Then, we apply threshold rounding to the fractional solution corresponding to $i^*$, which leads to a success probability of $1-\gamma$.

The optimal choice of $\varepsilon > 0$ above is $\varepsilon = 1/\sqrt{n}$ when rounding using the exponential mechanism. This leads to a polynomial-time $(1 + O(1/\sqrt{n}), O(\lambda_2))$-approximation algorithm with pointwise Lipschitz constant $O(n/\lambda_2)$. We remark that the additive error is a product of the rounding scheme and is not present in the fractional solution. We show that this algorithm is tight in the sense that even an $(O(1), \lambda_2)$-approximation algorithm must have pointwise Lipschitz constant $\Omega(n/\lambda_2)$. Thus, our PGTA framework can achieve tight pointwise Lipschitz bounds given a sufficiently stable rounding scheme.

**$k$-Way Submodularity Boosting (Appendix D.3).** The second algorithm is a polynomial-time algorithm that, given a graph $G = (V, E)$, a weight vector $\boldsymbol{w} \in \mathbb{R}^E_{\geq 0}$, and a parameter $\beta \in (0, 1/2)$, outputs a vertex set of cut weight at most $O(\beta^{-1}) \cdot \mathrm{OPT}_\beta$ with pointwise Lipschitz constant $O(\beta^{-2}\lambda_2\sqrt{n})$, where $\mathrm{OPT}_\beta$ is the minimum weight of a cut $A$ with $\beta n \leq |A| \leq (1-\beta)n$.

In contrast to the first boosting algorithm, the second algorithm has no additive error, but the multiplicative error is with respect to the $\beta$-balanced cuts. The reason why we consider $\beta$-balanced cuts instead of arbitrary cuts is to ensure that the success probability of the thresholding-based rounding is bounded away from $1/2$ by at least $\beta$. In this algorithm, we first compute vertex sets $A_1, \dots, A_k$ by independently applying the threshold rounding multiple times to the (single) fractional solution assuming that the optimal cut (as a vertex set) has size between $\beta n$ and $(1-\beta)n$. Then, we utilize the $k$-way submodularity [Harvey et al., 2006] of the cut function to combine these vertex sets into a single vertex set with a small pointwise Lipschitz constant. Using the fact that at least half of $A_1, \dots, A_k$ are feasible (with high probability, because $0 < \beta < 1/2$), we can show that the output set is feasible (with high probability). The $k$-way submodularity property guarantees that the output set is a $O(\beta^{-1})$-approximation.

### 5.3 Lower Bound

In order to show the optimality of our framework, we show a matching lower bound for the minimum $S$-$T$ cut problem. Our approach consists of two main steps: first we consider a family of graphs, parametrized by the number of vertices $n$, and two weight vectors $\boldsymbol{w}, \boldsymbol{w}'$ that are sufficiently far

apart. Then, we show that any algorithm that achieves the desired approximation ratio must output solutions that, with high probability, differ in a large number of vertices under $\boldsymbol{w}, \boldsymbol{w}'$. Lastly, we show this construction implies a lower bound on the Lipschitz constant of this algorithm by applying a compactness argument on a path from $\boldsymbol{w}$ to $\boldsymbol{w}'$ (see Theorem 46). We refer the reader to Appendix D.4 for the formal construction.

# 6 Bipartite $\boldsymbol{b}$-Matching

Given a graph $G = (V, E)$ (which may be bipartite) and a weight vector $\boldsymbol{w} \in \mathbb{R}^E_{>0}$, the maximum weight matching problem asks for a subset of edges $M \subseteq E$ which forms a matching in $G$ and maximizes the sum of the weights of the edges in $M$. The maximum weight $\boldsymbol{b}$-matching problem is defined similarly to the maximum weight matching problem, except each vertex $v$ is associated with a vertex capacity $b_v$ indicating the maximum number of neighbors that can be matched to $v$ in the $\boldsymbol{b}$-matching. Note that a matching is a $\boldsymbol{b}$-matching for $\boldsymbol{b} \equiv 1$. Our main results are the following.

**Theorem 7** (Informal; See Theorem 29 )**.** *There is an efficient $(2+\varepsilon)$-approximation algorithm for maximum weight bipartite matching with pointwise Lipschitz constant $O\left(\frac{\sqrt{m}}{\varepsilon w_{\min}}\right)$. Here, $w_{\min}$ is the minimum input edge weight.*

**Theorem 8** (Informal; See Theorem 30 )**.** *There is an efficient $\left(\frac{2+\varepsilon}{1-1/e}\right)$-approximation algorithm for maximum weight bipartite $\boldsymbol{b}$-matching with pointwise Lipschitz constant $O\left(\frac{\sqrt{m}}{\varepsilon w_{\min}}\right)$. Here, $w_{\min}$ is the minimum input edge weight.*

The formal statement and proofs of Theorems 7 and 8 are deferred to Appendix E. We briefly discuss our approach in Section 6.1.

## 6.1 Our Approach

PGTA applies directly to the weighted $\ell_2$-regularizer and ($\boldsymbol{b}$-)matching polytope. For the simple case of 1-matchings, it suffices to apply the rounding algorithm due to Kumabe and Yoshida [2023]. However, for general $\boldsymbol{b}$-matchings, we can no longer directly use their rounding scheme to obtain an integral solution. This is due to the fact that our $b_v$ constraints must be satisfied, which the rounding scheme given in Kumabe and Yoshida [2023] does not necessarily ensure. Instead, we design a new multi-item auction-based rounding scheme where multiple vertices on the left (buyers) bid on multiple vertices on the right (sellers). This yields a $\frac{2(1+\varepsilon)}{1-1/e}$-approximate maximum weight bipartite $\boldsymbol{b}$-matching with pointwise Lipschitz constant $O\left(\frac{\sqrt{m}}{\varepsilon w_{\min}}\right)$.

Let $x^\star$ denote the optimal fractional solution. Loosely speaking, we think of each seller $v$ now as having $b_v$ items to sell and each buyer $u$ now selects $b_u$ sellers independently with probability $\frac{x^\star_{uv}}{b_u}$. For each selected seller, the buyer bids on one of the seller's items uniformly at random. Each seller $v$ accumulates the bids for each of its $b_v$ items and accepts a bid uniformly at random per item. The main challenge compared to the matching case ($\boldsymbol{b} \equiv 1$) is that an edge $uv$ can be selected at most once in the $\boldsymbol{b}$-matching, but the buyer $u$ may purchase multiple items from the seller $v$ in the described multi-item auction scheme. In other words, the probability of the edge $uv$ being selected in the rounding scheme is equal to the probability of *at least* one transaction occurring between $u, v$ but does not scale with the *total* number of transactions. This is the cause of the additional $(1 - 1/e)$ factor in the approximation ratio compared to our bipartite matching algorithm.

# 7 Conclusion

In this work, we developed a principled way to design pointwise Lipschitz continuous graph algorithms. We demonstrated that our framework, paired with stable rounding schemes, can give algorithms with tight trade-offs between their Lipschitz constant and approximation guarantees, as demonstrated by our set of results regarding minimum vertex $S$-$T$ cut. In addition to its broad applicability, our framework yields better bounds than possible compared to previous techniques.

One very interesting future direction is to obtain tight bounds for the trade-offs between Lipschitz constants and approximation guarantees for the rest of the problems we consider in our work, including matchings and $\boldsymbol{b}$-matchings. Understanding the tight Lipschitz constant for specific problems yields a better comprehension of which problems have (approximately) optimal solutions that can be tracked in a stable way.

## References


Akshay Agrawal, Brandon Amos, Shane T. Barratt, Stephen P. Boyd, Steven Diamond, and J. Zico Kolter. Differentiable convex optimization layers. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 9558–9570, 2019.

Brandon Amos. Differentiable optimization-based modeling for machine learning. *Ph. D. thesis*, 2019.

Haris Angelidakis, Konstantin Makarychev, and Yury Makarychev. Algorithms for stable and perturbation-resilient problems. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 438–451, 2017.

Moab Arar, Shiri Chechik, Sarel Cohen, Cliff Stein, and David Wajc. Dynamic matching: Reducing integral algorithms to approximately-maximal fractional algorithms. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, volume 107 of *LIPIcs*, pages 7:1–7:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi: 10.4230/LIPICS.ICALP.2018.7. URL `https://doi.org/10.4230/LIPIcs.ICALP.2018.7`.

Pranjal Awasthi, Avrim Blum, and Or Sheffet. Center-based clustering under perturbation stability. *Information Processing Letters*, 112(1-2):49–54, 2012.

Maria Florina Balcan and Yingyu Liang. Clustering under perturbation resilience. *SIAM Journal on Computing*, 45(1):102–155, 2016.

Maria-Florina Balcan, Nika Haghtalab, and Colin White. k-center clustering under perturbation resilience. *ACM Transactions on Algorithms (TALG)*, 16(2):1–39, 2020.

Aaron Bernstein, Jiale Chen, Aditi Dudeja, Zachary Langley, Aaron Sidford, and Ta-Wei Tu. Matching composition and efficient weight reduction in dynamic matching. In Yossi Azar and Debmalya Panigrahi, editors, *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2025, New Orleans, LA, USA, January 12-15, 2025*, pages 2991–3028. SIAM, 2025. doi: 10.1137/1.9781611978322.97. URL `https://doi.org/10.1137/1.9781611978322.97`.

Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis R. Bach. Learning with differentiable perturbed optimizers. *CoRR*, abs/2002.08676, 2020.

Sayan Bhattacharya, Silvio Lattanzi, and Nikos Parotsidis. Efficient and stable fully dynamic facility location. *Advances in neural information processing systems*, 35:23358–23370, 2022.

Yonatan Bilu and Nathan Linial. Are stable instances easy? *Combinatorics, Probability and Computing*, 21(5):643–660, 2012.

Yonatan Bilu, Amit Daniely, Nati Linial, and Michael Saks. On the practically interesting instances of MAXCUT. In Natacha Portier and Thomas Wilke, editors, *30th International Symposium on Theoretical Aspects of Computer Science (STACS 2013)*, volume 20 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 526–537, Dagstuhl, Germany, 2013. ISBN 978-3-939897-50-7. doi: 10.4230/LIPIcs.STACS.2013.526.

Jeremiah Blocki, Elena Grigorescu, and Tamalika Mukherjee. Privately estimating graph parameters in sublinear time. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 229 of *LIPIcs*, pages 26:1–26:19, 2022. doi: 10.4230/LIPICS.ICALP.2022.26.

Mathieu Blondel, Olivier Teboul, Quentin Berthet, and Josip Djolonga. Fast differentiable sorting and ranking. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 950–959. PMLR, 2020.

Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient nd image segmentation. *International journal of computer vision*, 70(2):109–131, 2006.

Mark Bun, Marco Gaboardi, Max Hopkins, Russell Impagliazzo, Rex Lei, Toniann Pitassi, Satchit Sivakumar, and Jessica Sorrell. Stability is stable: connections between replicability, privacy, and adaptive generalization. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC)*, pages 520–527, 2023.

Zachary Chase, Shay Moran, and Amir Yehudayoff. Stability and replicability in learning. In *Proceedings of the IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 2430–2439, 2023.

Zachary Chase, Bogdan Chornomaz, Shay Moran, and Amir Yehudayoff. Local borsuk-ulam, stability, and replicability. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 1769–1780. ACM, 2024. doi: 10.1145/3618260.3649632.

Hongjie Chen, Vincent Cohen-Addad, Tommaso d'Orsi, Alessandro Epasto, Jacob Imola, David Steurer, and Stefan Tiegel. Private estimation algorithms for stochastic block models and mixture models. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL `http://papers.nips.cc/paper_files/paper/2023/hash/d702d78b2468d2bc80b22a2fc3e59faf-Abstract-Conference.html`.

Jiale Chen, Aaron Sidford, and Ta-Wei Tu. Entropy regularization and faster decremental matching in general graphs. In Yossi Azar and Debmalya Panigrahi, editors, *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2025, New Orleans, LA, USA, January 12-15, 2025*, pages 3069–3115. SIAM, 2025. doi: 10.1137/1.9781611978322.99. URL `https://doi.org/10.1137/1.9781611978322.99`.

Li Chen, Gramoz Goranci, Monika Henzinger, Richard Peng, and Thatchaphol Saranurak. Fast dynamic cuts, distances and effective resistances via vertex sparsifiers. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1135–1146. IEEE, 2020.

Vincent Cohen-Addad, Niklas Oskar D Hjuler, Nikos Parotsidis, David Saulpic, and Chris Schwiegelshohn. Fully dynamic consistent facility location. *Advances in Neural Information Processing Systems (NeurIPS)*, 32, 2019.

Mina Dalirrooyfard, Slobodan Mitrovic, and Yuriy Nevmyvaka. Nearly tight bounds for differentially private multiway cut. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Sashka Davis, Jeff Edmonds, and Russell Impagliazzo. Online algorithms to minimize resource reallocations and network communication. In *International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX)*, pages 104–115, 2006.

Michael Dinitz, George Z Li, Quanquan C Liu, and Felix Zhou. Differentially private matchings. *arXiv preprint arXiv:2501.00926*, 2025.

Peter Dixon, Aduri Pavan, Jason Vander Woude, and N. V. Vinodchandran. List and certificate complexities in replicable learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Josip Djolonga and Andreas Krause. Differentiable learning of submodular models. *Advances in Neural Information Processing Systems*, 30, 2017.

Priya L. Donti, David Rolnick, and J. Zico Kolter. DC3: A learning method for optimization with hard constraints. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Eric Eaton, Marcel Hussing, Michael Kearns, and Jessica Sorrell. Replicable reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Hossein Esfandiari, Alkis Kalavasis, Amin Karbasi, Andreas Krause, Vahab Mirrokni, and Grigoris Velegkas. Replicable bandits. In *The 11th International Conference on Learning Representations (ICLR)*, 2023a.

Hossein Esfandiari, Amin Karbasi, Vahab Mirrokni, Grigoris Velegkas, and Felix Zhou. Replicable clustering. In *37th Conference on Neural Information Processing Systems (NeurIPS)*, 2023b.

Tatsuya Gima, Soh Kumabe, and Yuichi Yoshida. Courcelle's theorem for lipschitz continuity. In *33rd Annual European Symposium on Algorithms, ESA 2025, September 15-17, 2025, Warsaw, Poland*, 2025.

Gramoz Goranci, Harald Räcke, Thatchaphol Saranurak, and Zihan Tan. The expander hierarchy and its applications to dynamic graph algorithms. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 2212–2228. SIAM, 2021. doi: 10.1137/1.9781611976465.132. URL https://doi.org/10.1137/1.9781611976465.132.

Dorothy M Greig, Bruce T Porteous, and Allan H Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 51 (2):271–279, 1989.

Anupam Gupta and Roie Levin. Fully-dynamic submodular cover with bounded recourse. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 1147–1157. IEEE, 2020. doi: 10.1109/FOCS46700.2020.00110. URL https://doi.org/10.1109/FOCS46700.2020.00110.

Anupam Gupta, Ravishankar Krishnaswamy, Amit Kumar, and Debmalya Panigrahi. Online and dynamic algorithms for set cover. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 537–550, 2017.

Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pages 1225–1234. PMLR, 2016.

Nicholas J. A. Harvey, Robert Kleinberg, and April Rasala Lehman. On the capacity of information networks. *IEEE Transactions on Information Theory*, 52(6):2345–2364, 2006.

Monika Henzinger, Evangelos Kosinas, Robin Münk, and Harald Räcke. Efficient contractions of dynamic graphs - with applications. In Anne Benoit, Haim Kaplan, Sebastian Wild, and Grzegorz Herman, editors, *33rd Annual European Symposium on Algorithms, ESA 2025, September 15-17, 2025, Warsaw, Poland*, volume 351 of *LIPIcs*, pages 36:1–36:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025. doi: 10.4230/LIPICS.ESA.2025.36. URL https://doi.org/10.4230/LIPIcs.ESA.2025.36.

Max Hopkins, Russell Impagliazzo, Daniel Kane, Sihan Liu, and Christopher Ye. Replicability in high dimensional statistics. *arXiv preprint arXiv:2406.02628*, 2024.

Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

Russell Impagliazzo, Rex Lei, Toniann Pitassi, and Jessica Sorrell. Reproducibility in learning. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 818–831, 2022.

Arun Jambulapati, Yujia Jin, Aaron Sidford, and Kevin Tian. Regularized box-simplex games and dynamic decremental bipartite matching. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPIcs*, pages 77:1–77:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi: 10.4230/LIPICS.ICALP.2022.77. URL https://doi.org/10.4230/LIPIcs.ICALP.2022.77.

Alkis Kalavasis, Amin Karbasi, Shay Moran, and Grigoris Velegkas. Statistical indistinguishability of learning algorithms. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pages 15586–15622, 2023.

Alkis Kalavasis, Amin Karbasi, Kasper Green Larsen, Grigoris Velegkas, and Felix Zhou. Replicable learning of large-margin halfspaces, 2024a.

Alkis Kalavasis, Amin Karbasi, Grigoris Velegkas, and Felix Zhou. On the computational landscape of replicable learning. *CoRR*, abs/2405.15599, 2024b. doi: 10.48550/ARXIV.2405.15599.

Leonid V Kantorovich. Mathematical methods of organizing and planning production. *Management Science*, 6(4):366–422, 1960.

Amin Karbasi, Grigoris Velegkas, Lin F Yang, and Felix Zhou. Replicability in reinforcement learning. In *37th Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Shiva Prasad Kasiviswanathan, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Analyzing graphs with node differential privacy. In *Proceedings of the 10th Theory of Cryptography Conference (TCC)*, pages 457–476, 2013.

Junpei Komiyama, Shinji Ito, Yuichi Yoshida, and Souta Koshino. Replicability is asymptotically free in multi-armed bandits. *arXiv preprint arXiv:2402.07391*, 2024.

Soh Kumabe and Yuichi Yoshida. Average sensitivity of dynamic programming. In *Proceedings of the 33th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1925–1961, 2022a.

Soh Kumabe and Yuichi Yoshida. Average sensitivity of the knapsack problem. In *30th Annual European Symposium on Algorithms (ESA)*, volume 244, pages 75:1–75:14, 2022b.

Soh Kumabe and Yuichi Yoshida. Lipschitz continuous algorithms for graph problems. In *Proceedings of the 64th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 762–797, 2023.

Soh Kumabe and Yuichi Yoshida. Lipschitz continuous allocations for optimization games. In Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson, editors, *51st International Colloquium on Automata, Languages, and Programming, ICALP 2024, July 8-12, 2024, Tallinn, Estonia*, volume 297 of *LIPIcs*, pages 102:1–102:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. URL `https://doi.org/10.4230/LIPIcs.ICALP.2024.102`.

Soh Kumabe and Yuichi Yoshida. Lipschitz continuous algorithms for covering problems. In *Proceedings of the 2025 ACM-SIAM Symposium on Discrete Algorithms, SODA 2025, New Orleans, LA, USA, January 12–15, 2025*. SIAM, 2025.

Silvio Lattanzi and Sergei Vassilvitskii. Consistent k-clustering. In *International Conference on Machine Learning*, pages 1975–1984. PMLR, 2017.

Elizaveta Levina and Peter J. Bickel. The earth mover's distance is the mallows distance: some insights from statistics. In *Proceedings of the 8th International Conference On Computer Vision (ICCV)*, pages 251–256, 2001.

Shijie Li, Weiqiang He, Ruobing Bai, and Pan Peng. Average sensitivity of hierarchical $k$-median clustering. *arXiv preprint arXiv:2507.10296*, 2025.

Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Bilu–linial stable instances of max cut and minimum multiway cut. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 890–906, 2014.

Colin L Mallows. A note on asymptotic joint normality. *The Annals of Mathematical Statistics*, pages 508–515, 1972.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 94–103, 2007.

Nicole Megow and Lukas Nölke. Online metric matching on the line with recourse. *Algorithmica*, 87(6):813–841, 2025. doi: 10.1007/S00453-025-01299-8. URL https://doi.org/10.1007/s00453-025-01299-8.

Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pages 263–275. IEEE, 2017.

Pan Peng and Yuichi Yoshida. Average sensitivity of spectral clustering. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 1132–1140, 2020.

Sofya Raskhodnikova and Adam Smith. Lipschitz extensions for node-private graph statistics and the generalized exponential mechanism. In *Proceedings of the IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 495–504, 2016.

R Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. A metric for distributions with applications to image databases. In *Proceedings of the 6th International Conference on Computer Vision (ICCV)*, pages 59–66, 1998.

Walter Rudin. *Principles of mathematical analysis*, volume 3. McGraw-hill New York, 1964.

Mark Schmidt, Nicolas Roux, and Francis Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. *Advances in Neural Information Processing Systems (NeurIPS)*, 24, 2011.

Jonah Sherman. Area-convexity, $l_\infty$ regularization, and undirected multicommodity flow. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 452–460. ACM, 2017. doi: 10.1145/3055399.3055501. URL https://doi.org/10.1145/3055399.3055501.

Aaron Sidford and Kevin Tian. Coordinate methods for accelerating $\infty$ regression and faster approximate maximum flow. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 922–933. IEEE Computer Society, 2018. doi: 10.1109/FOCS.2018.00091. URL https://doi.org/10.1109/FOCS.2018.00091.

Prabhav Singh and Jessica Sorrell. Sensitivity of stability: Theoretical & empirical analysis of replicability for adaptive data selection in transfer learning. *arXiv preprint arXiv:2508.04901*, 2025.

Nithin Varma and Yuichi Yoshida. Average sensitivity of graph algorithms. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 684–703, 2021.

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009.

Jan Vondrák, Chandra Chekuri, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 783–792, 2011.

Yu-Xiang Wang, Jing Lei, and Stephen E Fienberg. Learning with differential privacy: stability, learnability and the sufficiency and necessity of ERM principle. *Journal of Machine Learning Research*, 17(183):1–40, 2016.

Lauren Watson, Benedek Rozemberczki, and Rik Sarkar. Stability enhanced privacy and applications in private stochastic gradient descent. *arXiv preprint arXiv:2006.14360*, 2020.

Yuichi Yoshida and Samson Zhou. Sensitivity analysis of the maximum matching problem. In *Innovations in Theoretical Computer Science (ITCS)*, pages 58:1–58:20, 2021.

## Contents


**1 Introduction** **1**

1.1 Summary of Results . . . . . 2

**2 Related Works** **3**

**3 Preliminaries** **4**

3.1 Pointwise Lipschitz Continuity . . . . . 5

3.1.1 Remarks on Definition 1 . . . . . 5

**4 PGTA Framework** **6**

4.1 Prior Works & Challenges . . . . . 6

4.2 Our Approach . . . . . 6

**5 Minimum Vertex $S$-$T$ Cut** **7**

5.1 Existing Techniques & Limitations . . . . . 7

5.2 Our Approach . . . . . 7

5.3 Lower Bound . . . . . 8

**6 Bipartite $b$-Matching** **9**

6.1 Our Approach . . . . . 9

**7 Conclusion** **9**

**A Organization of the Appendix** **17**

**B Additional Related Works** **17**

**C Proximal Gradient Trajectory Analysis** **18**

**D Minimum Vertex $S$-$T$ Cut** **21**

D.1 Obtaining a Fractional Lipschitz Solution . . . . . 22

D.2 Rounding via Exponential Mechanism . . . . . 24

D.3 Rounding via $k$-Way Submodularity . . . . . 25

D.4 A Lower Bound . . . . . 28

**E Maximum Weight Bipartite $b$-Matching** **28**

E.1 Warmup for Bipartite 1-Matching . . . . . 29

E.2 Obtaining a Fractional Lipschitz Solution . . . . . 30

E.3 Rounding via Multi-Item Cooperative Auction . . . . . 31

**F Pointwise Lipschitz Continuity under Shared Randomness** **35**

**G Additional Tools** **36**

G.1 Nonexpansiveness of Strongly Convex Gradient . . . . . 36
G.2 Weyl's Inequality . . . . . 37
G.3 Finite Perturbation Bounds for Pointwise Lipschitz Continuity . . . . . 37

**H Naïve Algorithm for Minimum Vertex $S$-$T$ Cut** **38**
H.1 Obtaining a Fractional Lipschitz Solution . . . . . 38
H.2 Threshold Rounding . . . . . 40

# A Organization of the Appendix

We organize the appendix as follows.

- Appendix B discusses some additional related works.
- Appendix C develops the PGTA framework, our main tool for analyzing the Lipschitz constant of regularized convex programs.
- Appendix D leverages PGTA along with two novel rounding techniques to design Lipschitz algorithms for minimum $S$-$T$ cut.
- Appendix E presents our Lipschitz algorithms for bipartite matching and $\boldsymbol{b}$-matching, both of which use PGTA, whereas the latter also requires a brand new rounding scheme.
- Appendix F explains how to extend our algorithms to the more practical shared-randomness setting.
- Appendix G states some standard tools for the sake of completeness.
- Appendix H presents a natural first attempt at a Lipschitz $S$-$T$ cut algorithm that attains sub-optimal approximation-stability tradeoffs.

# B Additional Related Works

**Differential Privacy.** Although Lipschitz constants are used in differential privacy (DP), for example, Lipschitz extensions, both the definition of the constants and the analyses we perform in our paper are fundamentally distinct. Indeed, any DP algorithm for the problems we study have large known lower bounds in additive error, up to $\Omega(n)$, even in expectation. Specifically, Dalirrooyfard et al. [2023] proposed a DP algorithm for the minimum $s$-$t$ cut problem with an additive error of $O(n)$, along with an (almost) matching lower bound. For matchings, a recent lower bound shows that any *explicit*[6] DP matching algorithm requires $\Omega(n)$ additive error, even when restricting to bipartite instances and allowing constant multiplicative error Dinitz et al. [2025]. In contrast, we obtain an additive error of $O(\lambda_2)$ for min $s$-$t$ cut (Table 1) and purely multiplicative error (*no additive error*) for matchings (Table 2).

Lipschitz extensions have been used before in the context of DP graph algorithms (see *e.g.*, Kasiviswanathan et al. [2013], Raskhodnikova and Smith [2016]). However, these prior definitions rely on "worst-case" Lipschitz constants. That is, the Lipschitz constant is defined as the maximum ratio between the distance of two outputs and the distance of their corresponding inputs, considering all possible input pairs in the function's domain.[7] Such a worst-case guarantee is often necessary in order to ensure differential privacy since DP mechanisms often require worst-case sensitivity.[8] As such, the notion of the pointwise Lipschitz constant we consider, which is defined in terms of the expectation, does not immediately translate to bounds for differential privacy.

On the other hand, differential privacy for graph algorithms can be understood as bounding the Rényi divergence Mironov [2017] of the output distributions under one edge/vertex/unit weight change. Hence, it is unclear if there is a generic transformation from privacy (Rényi divergence) to Lipschitzness (earth mover's distance) that does not introduce large errors.

**Perturbation Resilience.** Perturbation resilience Awasthi et al. [2012], Angelidakis et al. [2017], Balcan and Liang [2016], Bilu and Linial [2012] is defined (informally) as the property of a problem instance where the (unique) optimal solution to the problem does not change when the specific instance is slightly perturbed. Thus, perturbation-resilient input instances are restricted classes of the input space that allow for this property. It has been shown that many natural problems, such as various clustering and cut problems Awasthi et al. [2012], Balcan et al. [2020], Balcan and Liang [2016], Bilu et al. [2013], Bilu and Linial [2012], Makarychev et al. [2014], are easier to solve when the given instances are perturbation resilient while they are NP-hard to solve in general instances.

By definition, the algorithm that computes the optimal solution on a perturbation resilient instance has a pointwise Lipschitz constant of 0. However, it is not straightforward to efficiently verify that

[6] An explicit algorithm outputs edge subsets, whereas an implicit algorithm may output a different solution concept.

[7] In differential privacy literature, the Lipschitz constant is sometimes referred to as the *global sensitivity* of the function.

[8] Sometimes *coupled sensitivity* Blocki et al. [2022] is used, where the sensitivity is the worst-case distance under a coupling.

an instance is perturbation resilient. We design pointwise Lipschitz continuous algorithms with guarantees over all input instances, where the pointwise Lipschitz constant may depend on some natural, efficiently computable properties of the input, *e.g.*, the second smallest eigenvalue of the graph Laplacian. We emphasize that perturbation resilience is a property of the input instance whereas pointwise Lipschitz continuity is a property of the algorithm itself.

**Differentiable Algorithms.** The differentiable algorithms literature seeks to incorporate structured optimization problems as a differentiable operator within deep neural networks Amos [2019], Djolonga and Krause [2017], Blondel et al. [2020], Berthet et al. [2020], Donti et al. [2021], Agrawal et al. [2019]. This usually necessitates that the output is fractional and deterministic, unlike our setting where we also randomly round fractional solutions to discrete solutions. Although we also use continuous relaxations of discrete problems as an intermediate technique, our goal is to develop algorithms with approximation and stability guarantees rather than combining them within neural networks.

**Replicability.** The notion of pointwise Lipschitz continuous algorithms is related to the notion of replicable learning algorithms that operate on data drawn from some distribution, as introduced by Impagliazzo et al. [2022]. This definition requires that when an algorithm $\mathcal{A}$ is executed on two *i.i.d.* $n$-sample datasets from some distribution $\mathcal{D}$, under *shared internal randomness*, the outputs of the algorithm are exactly the same, with high probability. We can interpret replicable learning algorithms as being stable under resampling of the dataset, where stability is measured in terms of total variation distance. Since inception, replicability has been studied in a variety of settings Singh and Sorrell [2025], Esfandiari et al. [2023b], Bun et al. [2023], Esfandiari et al. [2023a], Kalavasis et al. [2023], Chase et al. [2023], Eaton et al. [2023], Karbasi et al. [2023], Dixon et al. [2023], Chase et al. [2024], Komiyama et al. [2024], Kalavasis et al. [2024a], Hopkins et al. [2024], Kalavasis et al. [2024b].

On the other hand, the stability of pointwise Lipschitz algorithms is measured in terms of the earth mover's distance.

# C Proximal Gradient Trajectory Analysis

In this section, we consider a convex program

$$\min_{\boldsymbol{x}\in K} f_{\boldsymbol{w}}(\boldsymbol{x}),$$

where $f_{\boldsymbol{w}} : \mathbb{R}^n \to \mathbb{R}$ and $K \subseteq \mathbb{R}^n$ are both convex and $f_{\boldsymbol{w}}$ is parameterized by a vector $\boldsymbol{w} \in \mathbb{R}^m$. We make no further assumption on $f_{\boldsymbol{w}}(\boldsymbol{x})$ other than it being real-valued and convex[9].

Our goal is to establish a fractional pointwise Lipschitz-continuous algorithm for the problem. We can think of $f_{\boldsymbol{w}}$ as the objective function for some convex relaxation of a combinatorial optimization problem such as the minimum $s$-$t$ cut problem, and $\boldsymbol{w}$ represents the weights on edges.

Note that, without any additional assumptions on the objective function $f_{\boldsymbol{w}}$, even small perturbations in the weights $\boldsymbol{w}$ can lead to drastically different optimal solutions. In order to address this, we introduce a smooth, strongly convex regularizer $g_{\boldsymbol{w}}(\boldsymbol{x})$ and instead solve the program

$$\min_{\boldsymbol{x}\in K} h_{\boldsymbol{w}}(\boldsymbol{x}), \qquad \text{where } h_{\boldsymbol{w}}(\boldsymbol{x}) \coloneqq f_{\boldsymbol{w}}(\boldsymbol{x}) + g_{\boldsymbol{w}}(\boldsymbol{x}).$$

Here for $L, \sigma > 0$, we say that a differentiable function $g : K \to \mathbb{R}$ is *$\sigma$-strongly convex* and *$L$-smooth* if for any $x, y \in K$,

$$g(y) \geq g(x) + \langle \nabla g(x), y - x\rangle + \frac{\sigma}{2}\|y - x\|^2,$$
$$g(y) \leq g(x) + \langle \nabla g(x), y - x\rangle + \frac{L}{2}\|y - x\|^2$$

hold, respectively.

Let $\boldsymbol{w}, \tilde{\boldsymbol{w}}$ be the original and perturbed weight parameters and let $\boldsymbol{x}^*, \tilde{\boldsymbol{x}}^*$ be the unique optimal solutions with respect to $h_{\boldsymbol{w}}, h_{\tilde{\boldsymbol{w}}}$, respectively. In order to analyze $\|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}^*\|_2$, we can instead analyze a sequence of iterates $(\boldsymbol{x}^k)_{k\geq 0}, (\tilde{\boldsymbol{x}}^k)_{k\geq 0}$ such that $\boldsymbol{x}^k \to \boldsymbol{x}^*, \tilde{\boldsymbol{x}}^k \to \tilde{\boldsymbol{x}}^*$ and analyze

$$\lim_{k\to\infty} \|\boldsymbol{x}^k - \tilde{\boldsymbol{x}}^k\|_2.$$

[9]We can alternatively assume that $f_{\boldsymbol{w}}$ is lower semi-continuous, proper, convex, and takes value in the extended reals.

To generate $(\boldsymbol{x}^k)$ and $(\tilde{\boldsymbol{x}}^k)$, we use the *proximal gradient method* (PGM; Definition 3). We emphasize that *PGM is only used in the stability analysis* and the algorithm designer is free to use their favorite optimization algorithm, provided the produced solution is sufficiently close to optimality.

It is known that PGM converges at an exponential rate even if we only compute an inexact proximal step Schmidt et al. [2011]. Moreover, it is known that the proximal operator is *strictly non-expansive* Rockafellar [1976], meaning that

$$\|\mathrm{prox}_{L,f}(\boldsymbol{y}) - \mathrm{prox}_{L,f}(\boldsymbol{z})\|_2^2 \leq \langle \mathrm{prox}_{L,f}(\boldsymbol{y}) - \mathrm{prox}_{L,f}(\boldsymbol{z}), \boldsymbol{y} - \boldsymbol{z} \rangle .$$

By an application of the Cauchy-Schwartz inequality, this immediately implies that

$$\|\mathrm{prox}_{L,f}(\boldsymbol{y}) - \mathrm{prox}_{L,f}(\boldsymbol{z})\|_2 \leq \|\boldsymbol{y} - \boldsymbol{z}\|_2. \tag{1}$$

We underline again that any algorithm which solves the convex program to some sufficiently small error enjoys the stability properties we prove next and PGM is only used to establish the stability of the unique optimal solution through an analysis of how its trajectory differs on two distinct executions under $\boldsymbol{w}, \tilde{\boldsymbol{w}}$.

To bound the fractional pointwise Lipschitz constant, we require the following assumptions:

**Assumption 9.** Let $0 \neq \delta \in \mathbb{R}$ be a real number with sufficiently small absolute value and suppose that we obtain $\tilde{\boldsymbol{w}}$ from $\boldsymbol{w}$ by perturbing a single coordinate of $\boldsymbol{w}$ by $\delta$.

1. $f_{\boldsymbol{w}}, f_{\tilde{\boldsymbol{w}}}$ are convex and continuous over $K$ ;
2. Assuming $|\delta| \leq \delta_{\boldsymbol{w}}$ for some $\delta_{\boldsymbol{w}}$ depending only on $\boldsymbol{w}$, both $g_{\boldsymbol{w}}$ and $g_{\tilde{\boldsymbol{w}}}$ are $\sigma_{\boldsymbol{w}}$-strongly convex over $K$ for the same $\sigma_{\boldsymbol{w}}$ that depends only on $\boldsymbol{w}$;
3. Assuming $|\delta| \leq \delta_{\boldsymbol{w}}$ for some $\delta_{\boldsymbol{w}}$ depending only on $\boldsymbol{w}$, both $g_{\boldsymbol{w}}$ and $g_{\tilde{\boldsymbol{w}}}$ are $L_{\boldsymbol{w}}$-smooth over $K$ for the same $L_{\boldsymbol{w}} > \sigma_{\boldsymbol{w}}$ that depends only on $\boldsymbol{w}$;
4. for every $\boldsymbol{x} \in K$, $\frac{1}{L_{\boldsymbol{w}}}\|\boldsymbol{\nabla} g_{\boldsymbol{w}}(\boldsymbol{x}) - \boldsymbol{\nabla} g_{\tilde{\boldsymbol{w}}}(\boldsymbol{x})\|_2 \leq C_{\boldsymbol{w}}|\delta|$ for some constant $C_{\boldsymbol{w}}$ that depends only on $\boldsymbol{w}$;
5. for every $\boldsymbol{x} \in K$, $\|\mathrm{prox}_{L_{\boldsymbol{w}},f_{\boldsymbol{w}}}(\boldsymbol{x}) - \mathrm{prox}_{L_{\boldsymbol{w}},f_{\tilde{\boldsymbol{w}}}}(\boldsymbol{x})\|_2 \leq D_{\boldsymbol{w}}|\delta|$ for some $D_{\boldsymbol{w}}$ that depends only on $\boldsymbol{w}$.

Some remarks are in order regarding Condition 4 and Condition 5, which may not come across as natural at first glance. Consider a regularizer of the form $g_{\boldsymbol{w}}(\boldsymbol{x}) = \boldsymbol{w}^\top F(\boldsymbol{x})$ where $F : K \to \mathbb{R}^m$ is sufficiently "well-behaved". All of our regularizers fall under this form. Then $\boldsymbol{\nabla} g(\boldsymbol{x}) = \boldsymbol{\nabla} F(\boldsymbol{x})^\top w \in \mathbb{R}^n$ so that

$$\|\boldsymbol{\nabla} g_{\boldsymbol{w}}(\boldsymbol{x}) - \boldsymbol{\nabla} g_{\tilde{\boldsymbol{w}}}(\boldsymbol{x})\|_2 = \left\|\boldsymbol{\nabla} F(x)^\top (\boldsymbol{w} - \tilde{\boldsymbol{w}})\right\|_2 \leq \left\|\boldsymbol{\nabla} F(\boldsymbol{x})^\top\right\| \cdot \|\boldsymbol{w} - \tilde{\boldsymbol{w}}\|_2 = \left\|\boldsymbol{\nabla} F(\boldsymbol{x})^\top\right\| \cdot \delta,$$

where $\|\boldsymbol{\nabla} F(\boldsymbol{x})^\top\|$ denotes the operator norm of the matrix $\boldsymbol{\nabla} F(\boldsymbol{x})^\top$ with respect to the Euclidean norm. When $K$ is compact and $\boldsymbol{\nabla} F$ is continuous, we can always obtain a trivial bound to satisfy Condition 4. However, an important detail in our applications is that this bound can be improved significantly using problem-specific structure to yield better guarantees.

On the other hand, $\mathrm{prox}_{L_{\boldsymbol{w}},f_{\boldsymbol{w}}}(\boldsymbol{x})$ is the optimal solution of a strongly convex objective, meaning we can translate the closeness of the objective value to a proximity in the variable space. If $f_{\boldsymbol{w}}, f_{\tilde{\boldsymbol{w}}}$ are sufficiently close in function value, we would thus expect their optimal solutions to be close. In our applications, extracting a bound for Condition 5 that depends only on $\boldsymbol{w}$ requires exploiting the specific structure of $f_{\boldsymbol{w}}$.

**Theorem 10.** *Let $h_{\boldsymbol{w}}(x) = f_{\boldsymbol{w}}(x) + g_{\boldsymbol{w}}(x)$ be a composite function parameterized by weight vector $\boldsymbol{w}$, where $f_{\boldsymbol{w}} : K \to \mathbb{R}$ is convex and $g_{\boldsymbol{w}} : K \to \mathbb{R}$ is $\sigma_{\boldsymbol{w}}$-strongly convex and $L_{\boldsymbol{w}}$-smooth. Moreover, let $\tilde{\boldsymbol{w}}$ be obtained from $\boldsymbol{w}$ by perturbing one coordinate by $\delta > 0$. Suppose Assumption 9 holds. Then the optimal solutions $\boldsymbol{x}^*, \tilde{\boldsymbol{x}}^*$ corresponding to $h_{\boldsymbol{w}}$ and $h_{\tilde{\boldsymbol{w}}}$ satisfy*

$$\|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}^*\|_2 \leq \frac{L_{\boldsymbol{w}}(C_{\boldsymbol{w}} + D_{\boldsymbol{w}})}{\sigma_{\boldsymbol{w}}}|\delta|.$$

To prove Theorem 10, we use the following technical lemma, whose proof is deferred to Appendix G.1. Before we state it formally, let us define the notion of $\eta$-expansiveness.

**Definition 11.** Let $\eta > 0$, $K$ be some set endowed with some norm $\|\cdot\|$, and $\mathcal{T} : K \to K$. We say that $\mathcal{T}$ is an $\eta$-expansive operator with respect to $\|\cdot\|$ if $\forall \boldsymbol{x}, \boldsymbol{y} \in K$ it holds that $\|\mathcal{T}(\boldsymbol{x}) - \mathcal{T}(\boldsymbol{y})\| \leq \eta \cdot \|\boldsymbol{x} - \boldsymbol{y}\|$.

**Lemma 12.** *The gradient step of a $L$-smooth function $g$ that is $\sigma$-strongly convex over its convex feasible region $K$ with step size $\eta \leq 1/L$ is $(1 - \eta\sigma)$-expansive with respect to the Euclidean norm.*

For completeness, we include a proof of Lemma 12 in Appendix G.1.

The proof of Theorem 10 presented below takes inspiration from the notion of algorithmic stability introduced by Hardt et al. [2016]. Specifically, the authors analyze the stochastic gradient descent trajectory of an arbitrary objective function $h$ that is $L$-smooth and $\sigma$-strongly convex (which includes our composite objectives as a special case) and derive an $\ell_2$ solution perturbation bound of roughly $O(|\delta| \cdot \frac{B^2}{\sigma})$, where $B$ is a bound on the maximum norm of $\boldsymbol{\nabla} h$. Note that the bound depends on $B$ rather than $L$. While Hardt et al. [2016] observed that the proximal operator is a "stability-inducing" operation, they do not explicitly analyze composite objectives nor the PGM. On the other hand, analyzing the trajectory of PGM is a natural choice for composite objectives $h = f + g$ such as our own, and we develop a more fine-grained bound of the form $O(|\delta| \cdot \frac{L(C+D)}{\sigma})$, where $C, D$ bound deviations of $\boldsymbol{\nabla} g$ and $\mathrm{prox}_{L,f}$ (see Conditions 4 and 5). For our settings, $\frac{B^2}{\sigma}$ can be $\Omega(n^2)$ while $\frac{L(C+D)}{\sigma} = o(n)$. Thus, we develop stronger perturbation bounds for our particular setting by analyzing the trajectory of a more suitable process (PGM) and measuring the perturbation using carefully chosen quantities of the objective (smoothness $L$, gradient/proximal deviations $C, D$).

We now present the proof of Theorem 10.

*Proof of Theorem 10.* We assume that the neighborhood around $\boldsymbol{w}$ is small enough so that the conditions in Assumption 9 hold. Let

- $\boldsymbol{x}^t, \tilde{\boldsymbol{x}}^t$ denote the adjacent runs of PGM,
- $L = L_{\boldsymbol{w}}$ the common smoothness parameter,
- $\sigma = \sigma_{\boldsymbol{w}}$ the common strong convexity parameter,
- and $G(\boldsymbol{x}^t) \coloneqq \boldsymbol{x}^t - \frac{1}{L}\boldsymbol{\nabla} g_{\boldsymbol{w}}(\boldsymbol{x}^t)$ the gradient update ($\tilde{G}(\tilde{\boldsymbol{x}}^t)$ similarly).

Consider a single iteration:

$$
\begin{aligned}
&\|\boldsymbol{x}^{t+1} - \tilde{\boldsymbol{x}}^{t+1}\|_2 \\
&= \|\mathrm{prox}_{L,f_{\boldsymbol{w}}}(G(\boldsymbol{x}^t)) - \mathrm{prox}_{L,f_{\tilde{\boldsymbol{w}}}}(\tilde{G}(\tilde{\boldsymbol{x}}^t))\|_2 \\
&\leq \|\mathrm{prox}_{L,f_{\boldsymbol{w}}}(G(\boldsymbol{x}^t)) - \mathrm{prox}_{L,f_{\boldsymbol{w}}}(\tilde{G}(\tilde{\boldsymbol{x}}^t))\|_2 + \|\mathrm{prox}_{L,f_{\boldsymbol{w}}}(\tilde{G}(\tilde{\boldsymbol{x}}^t)) - \mathrm{prox}_{L,f_{\tilde{\boldsymbol{w}}}}(\tilde{G}(\tilde{\boldsymbol{x}}^t))\|_2 \\
&\leq \|G(\boldsymbol{x}^t) - \tilde{G}(\tilde{\boldsymbol{x}}^t)\|_2 + \|\mathrm{prox}_{L,f_{\boldsymbol{w}}}(\tilde{G}(\tilde{\boldsymbol{x}}^t)) - \mathrm{prox}_{L,f_{\tilde{\boldsymbol{w}}}}(\tilde{G}(\tilde{\boldsymbol{x}}^t))\|_2 && \text{(by Inequality 1)} \\
&\leq \|G(\boldsymbol{x}^t) - G(\tilde{\boldsymbol{x}}^t)\|_2 + \|\tilde{G}(\tilde{\boldsymbol{x}}^t) - G(\tilde{\boldsymbol{x}}^t)\|_2 + \|\mathrm{prox}_{L,f_{\boldsymbol{w}}}(\tilde{G}(\tilde{\boldsymbol{x}}^t)) - \mathrm{prox}_{L,f_{\tilde{\boldsymbol{w}}}}(\tilde{G}(\tilde{\boldsymbol{x}}^t))\|_2 \\
&\leq \left(1 - \frac{\sigma}{L}\right)\|\boldsymbol{x}^t - \tilde{\boldsymbol{x}}^t\|_2 + \frac{1}{L}\|\boldsymbol{\nabla} g_{\boldsymbol{w}}(\tilde{\boldsymbol{x}}^t) - \boldsymbol{\nabla} g_{\tilde{\boldsymbol{w}}}(\tilde{\boldsymbol{x}}^t)\|_2 + \|\mathrm{prox}_{L,f_{\boldsymbol{w}}}(\tilde{G}(\tilde{\boldsymbol{x}}^t)) - \mathrm{prox}_{L,f_{\tilde{\boldsymbol{w}}}}(\tilde{G}(\tilde{\boldsymbol{x}}^t))\|_2 \\
&&& \text{(by Lemma 12)} \\
&\leq \left(1 - \frac{\sigma}{L}\right)\|\boldsymbol{x}^t - \tilde{\boldsymbol{x}}^t\|_2 + C_{\boldsymbol{w}}|\delta| + D_{\boldsymbol{w}}|\delta| && \text{(by Condition 4, Condition 5)} \\
&= \left(1 - \frac{\sigma}{L}\right)^{t+1}\|\boldsymbol{x}^0 - \tilde{\boldsymbol{x}}^0\|_2 + \sum_{\tau=1}^{t+1}\left(1 - \frac{\sigma}{L}\right)^{t+1-\tau}(C_{\boldsymbol{w}} + D_{\boldsymbol{w}})|\delta| \\
&= \sum_{\tau=1}^{t+1}\left(1 - \frac{\sigma}{L}\right)^{t+1-\tau}(C_{\boldsymbol{w}} + D_{\boldsymbol{w}})|\delta|.
\end{aligned}
$$

Taking the limit as $t \to \infty$ yields the following bound on $\ell_2$-distance between optimal solutions $\boldsymbol{x}^*, \tilde{\boldsymbol{x}}^*$ respectively.

$$\|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}^*\|_2 \leq \frac{L(C_{\boldsymbol{w}} + D_{\boldsymbol{w}})}{\sigma}|\delta|.$$

□

## D Minimum Vertex $S$-$T$ Cut

In this section, we provide two pointwise Lipschitz continuous algorithms for the minimum vertex $S$-$T$ cut problem.

**Theorem 13.** *For any $\gamma > 0$, there is a randomized polynomial-time approximation algorithm for undirected minimum vertex $S$-$T$ cut that outputs a vertex subset $A \subseteq V$ with the following guarantees.*

*(a) $A$ is a $\left(1 + O\left(\frac{1}{\gamma\sqrt{n}}\right), O\left(\frac{\lambda_2 \log(1/\gamma)}{\gamma}\right)\right)$-approximate $S$-$T$ cut with probability at least $1 - \gamma$.*

*(b) The algorithm has a pointwise Lipschitz constant of $O\left(n/\lambda_2\right)$.*

*Here $\lambda_2$ denotes the algebraic connectivity of $(G, \boldsymbol{w})$.*

For graphs undergoing edge insertions/deletions, the recourse of a dynamic algorithm is defined as the size of the symmetric difference of output sets between consecutive updates, *e.g.*, the change in the output between consecutive updates. It is known that a $(\zeta, \alpha)$-approximate (pointwise) Lipschitz algorithm with (pointwise) Lipschitz constant $K$ can be black-box transformed to a dynamic $(\zeta, \alpha)$-approximation algorithm where the expected recourse is at most $K$ [Kumabe and Yoshida, 2023, Theorem 1.6]. Thus, as a direct corollary of our minimum $s$-$t$ cut algorithm, we obtain the first dynamic minimum $s$-$t$ cut algorithm with non-trivial recourse.

**Corollary 14.** *There is a dynamic $(1 + O(1/\sqrt{n}), O(\lambda_2))$-approximation algorithm for minimum $S$-$T$ cut with expected recourse $O(n/\lambda_2)$, where $\lambda_2$ is the second smallest eigenvalue of the unnormalized Laplacian matrix of the current input graph.*

**Theorem 15.** *For any $\beta \in (0, 1/2)$ and $\gamma \in (0, 1)$, there exists a randomized polynomial-time approximation algorithm for the minimum vertex $S$-$T$ cut problem that outputs a vertex subset $A \subseteq V$ with the following guarantees.*

*(a) $\mathbb{E}[\mathrm{cut}_{G,\boldsymbol{w}}(A)] = O(\beta^{-1}\mathrm{OPT}_\beta)$.*

*(b) $A$ induces an $S$-$T$ cut with probability $1 - \gamma$.*

*(c) The algorithm has a pointwise Lipschitz constant of $O\left(\frac{\sqrt{n}\log(1/\gamma)}{\beta^2\lambda_2}\right)$.*

*Here $\mathrm{OPT}_\beta$ is the minimum weight of a $\beta$-balanced cut and $\lambda_2$ is the algebraic connectivity of the weighted graph $(G, \boldsymbol{w})$.*

We also provide a lower bound whichs shows that our Lipschitz-approximation tradeoff in Theorem 13 is tight.

**Theorem 16.** *For any sufficiently large integer $n$ and any $\lambda < n/c$ for a sufficiently large constant $c > 0$, there exists a weighted graph $(G, w)$ with algebraic connectivity at least $\lambda$ such that any algorithm that outputs a $(O(1), \lambda)$-approximate minimum $s$-$t$ cut on $(G, w)$ with probability at least $0.99$ has pointwise Lipschitz constant $\Omega(n/\lambda)$.*

In Appendix D.1, we describe our approach to obtain fractional pointwise Lipschitz continuous solutions. This part is identical between the two algorithms. We make the distinction at the rounding steps (Appendices D.2 to D.3). Finally, we prove the lower bound in Appendix D.4.

### D.1 Obtaining a Fractional Lipschitz Solution

The first step in our algorithm is to obtain a fractional Lipschitz algorithm. Fix any $s_0 \in S, t_0 \in T$, and consider the following convex optimization problem.

$$\begin{array}{lll} \text{minimize} & \displaystyle\sum_{uv \in E} w_{uv}|y_u - y_v| & \\ \text{subject to} & y_{t_0} - y_{s_0} = 1 & \\ & y_s = y_{s_0} & \forall s \in S \\ & y_t = y_{t_0} & \forall t \in T \\ & \langle \mathbb{1}, y \rangle = 0 & \\ & y \in [-1,1]^V \cap [y_{t_0}, y_{s_0}]^V & \end{array} \tag{2}$$

To see that this is a valid relaxation of the minimum $S$-$T$ cut problem, let $\mathbb{1}_A$ denote the indicator of any vertex set $A \subseteq V$ with $S \subseteq A$ and $A \cap T = \varnothing$. Then $\mathbb{1}_A - \frac{|A|}{n}\mathbb{1}$ is a feasible solution to the program above with the exact same objective as $\mathbb{1}_A$.

The goal of this section is to show the following:

**Theorem 17.** *For any $\varepsilon > 0$, there exists a polynomial-time algorithm that, given a graph $G = (V, E)$ and a weight vector $\boldsymbol{w} \in \mathbb{R}^E_{\geq 0}$, outputs a feasible solution $\boldsymbol{y}$ to LP 2 with the following properties:*

- *$f_{\boldsymbol{w}}(\boldsymbol{y}) \leq (1+\varepsilon) f_{\boldsymbol{w}}(\boldsymbol{y}^*)$, where $\boldsymbol{y}^*$ is an optimal solution to LP 2.*
- *The fractional pointwise Lipschitz constant with respect to the $\ell_2$ norm is $O(\varepsilon^{-1}\lambda_2^{-1})$.*

To guarantee fractional pointwise Lipschitz continuity, we add a smooth and strongly convex regularizer as follows.

$$\begin{array}{lll} \text{minimize} & \displaystyle\sum_{uv \in E} w_{uv}|y_u - y_v| + \frac{\varepsilon}{2}\sum_{uv \in E} w_{uv}(y_u - y_v)^2 & \\ \text{subject to} & y_{t_0} - y_{s_0} = 1 & \\ & y_s = y_{s_0} & \forall s \in S \\ & y_t = y_{t_0} & \forall t \in T \\ & \langle \mathbb{1}, y \rangle = 0 & \\ & y \in [-1,1]^V \cap [y_{t_0}, y_{s_0}]^V & \end{array}$$

This ensures that the optimal objective has a multiplicative error of $1 + \varepsilon$ with respect to the unregularized problem.

Let $\mathcal{L}_{\boldsymbol{w}}$ denote the unnormalized Laplacian matrix with respect to the edge weights $\boldsymbol{w}$, *i.e.*,

$$\mathcal{L}_{\boldsymbol{w}} = BWB^\top,$$

where $B \in \mathbb{R}^{V \times E}$ is the signed edge-vertex incidence matrix (under some arbitrary orientation of the edges) and $W \in \mathbb{R}^{E \times E}$ is the matrix of edge weights. Let $f_{\boldsymbol{w}}(\boldsymbol{y}) = \sum_{uv \in E} w_{uv}|y_u - y_v|$ and $g_{\boldsymbol{w}}(\boldsymbol{y}) \coloneqq \frac{\varepsilon}{2}\boldsymbol{y}^\top \mathcal{L}_{\boldsymbol{w}} \boldsymbol{y}$. Then we can rewrite the problem as

$$\begin{array}{lll} \text{minimize} & f_{\boldsymbol{w}}(\boldsymbol{y}) + g_{\boldsymbol{w}}(\boldsymbol{y}) & \\ \text{subject to} & y_{t_0} - y_{s_0} = 1 & \\ & y_s = y_{s_0} & \forall s \in S \\ & y_t = y_{t_0} & \forall t \in T \\ & \langle \mathbb{1}, y \rangle = 0 & \\ & y \in [-1,1]^V \cap [y_{t_0}, y_{s_0}]^V & \end{array}$$

Let $\lambda_1 \leq \cdots \leq \lambda_n$ denote the eigenvalues of $\mathcal{L}_{\boldsymbol{w}}$. It is known that the first eigenvalue $\lambda_1 = 0$ corresponds to the all-ones vector $\mathbb{1}$, which can break the strong convexity of the objective function if we optimize over vectors in $[-1,1]^V$. However, since we restrict ourselves to work in the subspace $\langle \mathbb{1}, \boldsymbol{y} \rangle = 0$, we see that $g_{\boldsymbol{w}}(\boldsymbol{y}) = \frac{\varepsilon}{2}\boldsymbol{y}^\top \mathcal{L}_{\boldsymbol{w}} \boldsymbol{y}$ is $\varepsilon\lambda_n$-smooth and $\varepsilon\lambda_2$-strongly convex.

We now prove Theorem 17.

**Remark 18.** *We assume without loss of generality that $\lambda_2 > 0$. Indeed, if $\lambda_2 = 0$ then our bounds trivially hold under the convention $1/0 = \infty$.*

We wish to apply Theorem 10 and Lemma 2 over a small neighborhood of $\boldsymbol{w}$. Recall by Lemma 2 that it does not suffice to apply Theorem 10 just for $\boldsymbol{w}, \tilde{\boldsymbol{w}}$. We must perform the analysis for all $\hat{\boldsymbol{w}}$ in a neighborhood of $\boldsymbol{w}$ and $\tilde{\boldsymbol{w}}$ obtained from $\hat{\boldsymbol{w}}$ by perturbing a single coordinate. To do so, we first check that $f_{\boldsymbol{w}}$ and $g_{\boldsymbol{w}}$ satisfy Assumption 9. Fix $\hat{\boldsymbol{w}}, \tilde{\boldsymbol{w}}$ in a small neighborhood of $w$. Let $0 \neq \delta \in \mathbb{R}$ be a real number arbitrarily in small absolute value and suppose that we obtain $\tilde{\boldsymbol{w}}$ from $\hat{\boldsymbol{w}}$ by perturbing a single coordinate of $\hat{\boldsymbol{w}}$. We have the following properties:

1. $f_{\hat{\boldsymbol{w}}}, f_{\tilde{\boldsymbol{w}}}$ are convex and continuous over $K$;
2. $g_{\hat{\boldsymbol{w}}}, f_{\tilde{\boldsymbol{w}}}$ are $\sigma_{\boldsymbol{w}}$-strongly convex over $K$ for $\sigma_{\boldsymbol{w}} = \varepsilon\lambda_2(\mathcal{L}_{\boldsymbol{w}})/2$, that depends only on $\boldsymbol{w}$ (assuming $\hat{\boldsymbol{w}}, \tilde{\boldsymbol{w}}$ lies in the neighborhood of $\boldsymbol{w}$);
3. for sufficiently small $\delta$, $g_{\hat{\boldsymbol{w}}}, g_{\tilde{\boldsymbol{w}}}$ is $L_{\boldsymbol{w}}$-smooth over $K$ for $L_{\boldsymbol{w}} = 2\lambda_n(\mathcal{L}_{\boldsymbol{w}})$ (assuming $\hat{\boldsymbol{w}}, \tilde{\boldsymbol{w}}$ lies in the neighborhood of $w$);
4. for every $x \in K$, $\frac{1}{L_{\boldsymbol{w}}}\|\boldsymbol{\nabla} g_{\hat{\boldsymbol{w}}}(\boldsymbol{x}) - \boldsymbol{\nabla} g_{\tilde{\boldsymbol{w}}}(\boldsymbol{x})\|_2 \leq C_{\boldsymbol{w}}|\delta|$ for $C_{\boldsymbol{w}} = O(\varepsilon/L_{\boldsymbol{w}})$; (Lemma 19)
5. for every $\boldsymbol{x} \in K$, $\|\mathrm{prox}_{L_{\boldsymbol{w}}, f_{\hat{\boldsymbol{w}}}}(\boldsymbol{x}) - \mathrm{prox}_{L_{\boldsymbol{w}}, f_{\tilde{\boldsymbol{w}}}}(\boldsymbol{x})\|_2 \leq D_{\boldsymbol{w}}|\delta|$ for $D_{\boldsymbol{w}} = O(1/L_{\boldsymbol{w}})$. (Lemma 20)

**Lemma 19.** *Suppose we obtain $\tilde{\boldsymbol{w}}$ from $\hat{\boldsymbol{w}}$ by perturbing the $uv$-th coordinate by some $\delta \neq 0$. Then*

$$\frac{1}{L_{\boldsymbol{w}}}\|\boldsymbol{\nabla} g_{\hat{\boldsymbol{w}}}(\boldsymbol{y}) - \boldsymbol{\nabla} g_{\tilde{\boldsymbol{w}}}(\boldsymbol{y})\|_2 = O\left(\frac{|\delta|}{L_{\boldsymbol{w}}}\right).$$

*Proof.* We have

$$\frac{1}{L_{\boldsymbol{w}}}\|\boldsymbol{\nabla} g_{\hat{\boldsymbol{w}}}(\boldsymbol{y}) - \boldsymbol{\nabla} g_{\tilde{\boldsymbol{w}}}(\boldsymbol{y})\|_2 = \frac{\varepsilon}{L_{\boldsymbol{w}}}\|(\mathcal{L}_{\hat{w}} - \mathcal{L}_{\tilde{w}})y\|_2 = \frac{\varepsilon}{L_{\boldsymbol{w}}}\sqrt{(\delta y_u)^2 + (\delta y_v)^2} \leq \frac{\sqrt{2}\varepsilon|\delta|}{L_{\boldsymbol{w}}},$$

where we used the fact $\boldsymbol{y} \in [-1, 1]^V$ in the inequality. □

**Lemma 20.** *Suppose we obtain $\tilde{\boldsymbol{w}}$ from $\hat{\boldsymbol{w}}$ by perturbing the $uv$-th coordinate by some $\delta \neq 0$. Then for every $\boldsymbol{x} \in K$,*

$$\|\mathrm{prox}_{L_{\boldsymbol{w}}, f_{\hat{\boldsymbol{w}}}}(\boldsymbol{x}) - \mathrm{prox}_{L_{\boldsymbol{w}}, f_{\tilde{\boldsymbol{w}}}}(\boldsymbol{x})\|_2 = O\left(\frac{|\delta|}{L_{\boldsymbol{w}}}\right).$$

*Proof.* Let $\hat{\boldsymbol{z}} = \mathrm{prox}_{L_{\boldsymbol{w}}, f_{\hat{\boldsymbol{w}}}}(\boldsymbol{y})$ and $\tilde{\boldsymbol{z}} = \mathrm{prox}_{L_{\boldsymbol{w}}, f_{\tilde{\boldsymbol{w}}}}(\boldsymbol{y})$. By the $L_{\boldsymbol{w}}$-strong convexity of the proximal objective $\bar{f}_{\hat{\boldsymbol{w}}, \boldsymbol{y}}(\cdot) \coloneqq \frac{L_{\boldsymbol{w}}}{2}\|\cdot - \boldsymbol{y}\|_2^2 + f_{\hat{\boldsymbol{w}}}(\cdot)$, we see that

$$\begin{aligned}
\frac{L_{\boldsymbol{w}}}{2}\|\hat{\boldsymbol{z}} - \tilde{\boldsymbol{z}}\|_2^2 &\leq f_{\hat{\boldsymbol{w}}}(\tilde{\boldsymbol{z}}) + \frac{L_{\boldsymbol{w}}}{2}\|\tilde{\boldsymbol{z}} - \boldsymbol{y}\|_2 - f_{\hat{\boldsymbol{w}}}(\hat{\boldsymbol{z}}) - \frac{L_{\boldsymbol{w}}}{2}\|\hat{\boldsymbol{z}} - \boldsymbol{y}\|_2 - \underbrace{\langle \boldsymbol{\nabla} \bar{f}_{\hat{\boldsymbol{w}}, y}(\hat{\boldsymbol{z}}), \tilde{\boldsymbol{z}} - \hat{\boldsymbol{z}}\rangle}_{\geq 0} \\
&\leq f_{\hat{\boldsymbol{w}}}(\tilde{\boldsymbol{z}}) + \frac{L_{\boldsymbol{w}}}{2}\|\tilde{\boldsymbol{z}} - \boldsymbol{y}\|_2 - f_{\hat{\boldsymbol{w}}}(\hat{\boldsymbol{z}}) - \frac{L_{\boldsymbol{w}}}{2}\|\hat{\boldsymbol{z}} - \boldsymbol{y}\|_2
\end{aligned}$$

and vice versa for $\frac{L_{\boldsymbol{w}}}{2}\|\cdot - \boldsymbol{y}\|_2^2 + f_{\tilde{\boldsymbol{w}}}(\cdot)$. The second inequality here follows from the fact that the directional derivatives in every feasible direction is non-negative for any (local) minima of $\bar{f}_{\hat{\boldsymbol{w}}, \boldsymbol{y}}$. Adding these two inequalities leads to the regularization terms canceling out, leaving

$$\begin{aligned}
\frac{L_{\boldsymbol{w}} + L_{\boldsymbol{w}}}{2}\|\hat{\boldsymbol{z}} - \tilde{\boldsymbol{z}}\|_2^2 &\leq f_{\hat{\boldsymbol{w}}}(\tilde{\boldsymbol{z}}) - f_{\hat{\boldsymbol{w}}}(\hat{\boldsymbol{z}}) + f_{\tilde{\boldsymbol{w}}}(\hat{\boldsymbol{z}}) - f_{\tilde{\boldsymbol{w}}}(\tilde{\boldsymbol{z}}) \\
&= [f_{\hat{\boldsymbol{w}}}(\tilde{\boldsymbol{z}}) - f_{\tilde{\boldsymbol{w}}}(\tilde{\boldsymbol{z}})] - [f_{\hat{\boldsymbol{w}}}(\hat{\boldsymbol{z}}) - f_{\tilde{\boldsymbol{w}}}(\hat{\boldsymbol{z}})] \\
&= (\hat{w}_{uv} - \tilde{w}_{uv})|\tilde{z}_u - \tilde{z}_v| - (\hat{w}_{uv} - \tilde{w}_{uv})|\hat{z}_u - \hat{z}_v| \\
&\leq |\delta| \cdot ||\tilde{z}_u - \tilde{z}_v| - |\hat{z}_u - \hat{z}_v|| \\
&\leq |\delta| \cdot |\tilde{z}_u - \tilde{z}_v - \hat{z}_u + \hat{z}_v| && \text{(reverse triangle inequality)} \\
&= |\delta|\sqrt{(\tilde{z}_u - \tilde{z}_v - \hat{z}_u + \hat{z}_v)^2} \\
&= |\delta|\sqrt{[(\tilde{z}_u - \hat{z}_u) + (\hat{z}_v - \tilde{z}_v)]^2} \\
&\leq |\delta|\sqrt{2(\tilde{z}_u - \hat{z}_u)^2 + 2(\hat{z}_v - \tilde{z}_v)^2} && \text{(convexity of } x^2\text{)} \\
&\leq \sqrt{2}\delta\|\hat{\boldsymbol{z}} - \tilde{\boldsymbol{z}}\|_2.
\end{aligned}$$

It follows that

$$\|\hat{\boldsymbol{z}} - \tilde{\boldsymbol{z}}\|_2 \leq \frac{\sqrt{2}|\delta|}{L_{\boldsymbol{w}}}.$$

□

Equipped with the previous results, we see that we can apply Theorem 10. This gives the following guarantee regarding the distance of the optimal solutions under two different executions.

**Corollary 21.** *Let $G = (V, E)$ be a graph and $\boldsymbol{w} \in \mathbb{R}^E_{\geq 0}$ be a weight vector. Let $\hat{\boldsymbol{w}}, \tilde{\boldsymbol{w}} \in \mathbb{R}^E_{\geq 0}$ be two weight vectors in a sufficiently small neighborhood of $\boldsymbol{w}$, where $\tilde{\boldsymbol{w}} = \hat{\boldsymbol{w}} + \delta \mathbb{1}_e$, for some sufficiently small $\delta > 0$ and some edge $e \in E$. Let $\hat{\boldsymbol{y}}^*, \tilde{\boldsymbol{y}}^*$ be the optimal solutions of the regularized objectives $h_{\hat{\boldsymbol{w}}}, h_{\tilde{\boldsymbol{w}}}$, respectively. Then,*

$$\|\hat{\boldsymbol{y}}^* - \tilde{\boldsymbol{y}}^*\|_2 \leq \frac{L_{\boldsymbol{w}}(C_{\boldsymbol{w}} + D_{\boldsymbol{w}})}{\sigma_{\boldsymbol{w}}}|\delta| = O\left(\frac{|\delta|}{\varepsilon\lambda_2}\right).$$

By Corollary 21 and Lemma 2, we obtain Theorem 17.

To extract a cut from a fractional solution $\boldsymbol{y}$, we must round the fractional solution to an integral one. Consider the standard threshold rounding scheme described below: Sample $\tau \sim U[-1, 1]$ and take the threshold cut $A_\tau \coloneqq \{v \in V : y_v \leq \tau\}$. It is not hard to see that the following holds:

**Lemma 22** (Threshold Rounding)**.** *We have $\mathbb{E}_\tau[\mathrm{cut}_{G,\boldsymbol{w}}(A_\tau)] = \frac{1}{2}\sum_{uv\in E} w_{uv}|y_u - y_v| = \frac{1}{2}f_{\boldsymbol{w}}(\boldsymbol{y})$. Moreover, we have $\mathbb{E}_\tau\,|A_\tau \triangle \tilde{A}_\tau| = O(\|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|_1)$.*

We say that a set $A \subseteq V$ is *feasible* if $S \subseteq A \subseteq V \setminus T$ and *infeasible* otherwise. Then, the threshold rounding may output an empty cut or the whole vertex set, which are infeasible, with probability (at most) $1/2$. We develop various techniques for overcoming this issue in the next subsections.

### D.2 Rounding via Exponential Mechanism

We can round a fractional solution through the *exponential mechanism* McSherry and Talwar [2007]. We use a slightly modified version of its guarantees due to Varma and Yoshida [2021].

**Theorem 23** (Exponential Mechanism; Lemma 2.1 in Varma and Yoshida [2021])**.** *Let $\eta > 0$ and $\mathcal{A}_\eta$ be the algorithm that, given a vector $\boldsymbol{x} \in \mathbb{R}^k$, samples an index $i \in [k]$ with probability proportional to $\exp(-\eta x_i)$. Then, we have*

$$\Pr_{i\sim\mathcal{A}_\eta(x)}\left[x_i \geq \min_{j\in[k]} x_j + \frac{\log k}{\eta} + \frac{t}{\eta}\right] \leq \exp(-t).$$

*Moreover for $\tilde{\boldsymbol{x}} \in \mathbb{R}^k$,*

$$\mathrm{TV}(\mathcal{A}_\eta(\boldsymbol{x}), \mathcal{A}_\eta(\tilde{\boldsymbol{x}})) = O(\eta\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|_1).$$

Let $\gamma \in (0, 1)$ be a parameter such that $1/\gamma$ is an integer. If we know that the optimal $S$-$T$ cut $A$ has cardinality $[\gamma(i-1)n, \gamma in]$ for an integer $i \in [1/\gamma]$, then we have $\mathbb{1}_A - \frac{|A|}{n}\mathbb{1} \in [-\gamma i, 1 - (i-1)\gamma]^V$. Hence, we consider the following (regularized) convex relaxation of the problem.

$$\begin{array}{lll} \text{minimize} & f_{\boldsymbol{w}}(\boldsymbol{y}) + g_{\boldsymbol{w}}(\boldsymbol{y}) & \\ \text{subject to} & y_{s_0} - y_{t_0} = 1 & \\ & y_s = y_{s_0} & \forall s \in S \\ & y_t = y_{t_0} & \forall t \in T \\ & \langle \mathbb{1}, y\rangle = 0 & \\ & \boldsymbol{y} \in [-\gamma i, 1 - (i-1)\gamma]^V \cap [y_{t_0}, y_{s_0}]^V & \end{array} \tag{3}$$

We solve this LP for each $i \in [1/\gamma]$ and then apply the exponential mechanism to select a solution. The algorithm is summarized as Algorithm 1.

This leads to a proof of Theorem 13, our first main result in this section.

*Proof of Theorem 13.* We first analyze the solution quality. By combining Theorem 17 and Theo-

**Algorithm 1:** Minimum Vertex $S$-$T$ Cut via Exponential Mechanism

**Input:** A graph $G = (V, E)$, a weight vector $\boldsymbol{w} \in \mathbb{R}^E_{\geq 0}$, and $\gamma \in (0, 1)$

**for** $i \in [\gamma^{-1}]$ **do**
  Solve LP 3 for $i$ using Theorem 17 with $\varepsilon \coloneqq 1/\sqrt{n}$;
  Let $\theta^{(i)} \in \mathbb{R}$ and $\boldsymbol{y}^{(i)} \in \mathbb{R}^V$ be the obtained objective value and solution, respectively.
Sample $\Lambda_2 \sim U[\lambda_2/2, \lambda_2]$;
Apply the exponential mechanism on $\boldsymbol{\theta} = (\theta^{(1)}, \ldots, \theta^{(1/\gamma)})$ with $\eta = \gamma\varepsilon^{-1}\Lambda_2^{-1}/\sqrt{n}$;
Let $i^* \in [n]$ be the chosen index;
Apply the threshold rounding on $\boldsymbol{y}^{(i^*)}$, where we sample the threshold $\tau$ uniformly from $[-\gamma i^*, 1 - (i^* - 1)\gamma]$;

rem 23 with $t = \log 1/\gamma$, we see that with probability at least $1 - \gamma$, the selected fractional solution has objective at most

$$(1+\varepsilon)\mathrm{OPT}+O\left(\frac{\log\gamma^{-1}}{\eta}\right) = (1+\varepsilon)\mathrm{OPT}+O\left(\frac{\varepsilon\lambda_2\sqrt{n}\log\gamma^{-1}}{\gamma}\right) = \left(1+\frac{1}{\sqrt{n}}\right)\mathrm{OPT}+O\left(\frac{\lambda_2\log\gamma^{-1}}{\gamma}\right).$$

Threshold rounding yields a feasible $S$-$T$ cut with probability at least $1 - \gamma$ and conditioned on feasibility, the expected cost is the same as the objective of the fractional solution. An application of Markov's inequality on the non-negative random variable $\mathrm{cut}_{G,\boldsymbol{w}}(A) - \mathrm{OPT}$ yields the desired approximation guarantees.

Next, we analyze the pointwise Lipschitz constant. By Weyl's inequality (Theorem 45), we have $|\lambda_2 - \tilde{\lambda}_2| \leq 2\delta$. But then we can verify with a straightforward computation that

$$\mathrm{TV}(\Lambda_2, \tilde{\Lambda}_2) = \mathrm{TV}(U[\lambda_2/2, \lambda_2], U[\tilde{\lambda}_2/2, \tilde{\lambda}_2]) = O(\delta/\lambda_2)$$

for a sufficiently small $\delta$. Let $\boldsymbol{\theta} \in \mathbb{R}^{\gamma^{-1}}$ denote the vector consisting of optimal fractional values of the $\gamma^{-1}$ different convex relaxations we consider. Noting that Theorem 17 yields

$$\|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|_1 = O\left(\frac{\sqrt{n}\cdot|\delta|}{\varepsilon\lambda_2}\right),$$

and that for every index $i \in [1/\gamma]$ we have that $\|\boldsymbol{y}^{(i)} - \tilde{\boldsymbol{y}}^{(i)}\|_1 = O\left(\frac{\sqrt{n}\cdot|\delta|}{\varepsilon\lambda_2}\right)$, the pointwise Lipschitz constant is bounded as

$$\begin{aligned}
&[1 - \mathrm{TV}(\Lambda_2, \tilde{\Lambda}_2) - \mathrm{TV}(\mathcal{A}(\boldsymbol{\theta}), \mathcal{A}(\tilde{\boldsymbol{\theta}}))] \cdot O\left(\frac{\sqrt{n}}{\varepsilon\lambda_2}\right) + [\mathrm{TV}(\Lambda_2, \tilde{\Lambda}_2) + \mathrm{TV}(\mathcal{A}(\boldsymbol{\theta}), \mathcal{A}(\tilde{\boldsymbol{\theta}}))] \cdot n \\
&= O\left(\frac{\sqrt{n}}{\varepsilon\lambda_2}\right) + O\left(\frac{\eta}{\gamma} + \frac{1}{\lambda_2}\right) \cdot n = O\left(\frac{n}{\lambda_2}\right) + O\left(\frac{1}{\lambda_2}\right) \cdot n = O\left(\frac{n}{\lambda_2}\right).
\end{aligned}$$

The second last inequality follows from the choices $\varepsilon = 1/\sqrt{n}$ and $\eta = \gamma\varepsilon^{-1}\Lambda_2^{-1}/\sqrt{n} = \Theta(\gamma/\lambda_2)$. □

### D.3 Rounding via $k$-Way Submodularity

Recall that the cut function $\mathrm{cut}_{G,\boldsymbol{w}} : 2^V \to \mathbb{R}$ is a *submodular* function, *i.e.*, for every $A, B \subseteq V$,

$$\mathrm{cut}_{G,\boldsymbol{w}}(A) + \mathrm{cut}_{G,\boldsymbol{w}}(B) \geq \mathrm{cut}_{G,\boldsymbol{w}}(A \cup B) + \mathrm{cut}_{G,\boldsymbol{w}}(A \cap B).$$

To boost the probability of outputting feasible sets, we use the following result regarding submodular functions due to Harvey et al. [2006].

**Theorem 24** ($k$-way submodularity; Theorem 2 in Harvey et al. [2006])**.** *Let $f : 2^{\mathcal{X}} \to \mathbb{R}$ be a submodular function and $A_1, \ldots, A_k \subseteq \mathcal{X}$ be a collection of $k$ sets. For $r \in [k]$, define*

$$B_{r,k} \coloneqq \bigcup_{\{i_1,\ldots,i_r\} \subseteq \binom{[k]}{r}} \bigcap_{j=1}^{r} A_{i_j}.$$

*Then, we have*

$$\sum_{r=1}^{k} f(A_r) \geq \sum_{r=1}^{k} f(B_{r,k}).$$

*Note that the case of $k = 2$ is simply the definition of submodularity.*

First, we consider the following convex relaxation for the *$\beta$-balanced $S$-$T$* cut problem for some $\beta \in [0, 1/2]$.

$$\begin{array}{lll} \text{minimize} & f_{\boldsymbol{w}}(\boldsymbol{y}) + g_{\boldsymbol{w}}(\boldsymbol{y}) & \\ \text{subject to} & y_{s_0} - y_{t_0} = 1 & \\ & y_s = y_{s_0} & \forall s \in S \\ & y_t = y_{t_0} & \forall t \in T \\ & \langle \mathbb{1}, y \rangle = 0 & \\ & y \in [-1+\beta, 1-\beta]^V \cap [y_{t_0}, y_{s_0}]^V & \end{array} \tag{4}$$

**Algorithm 2:** Rounding for Minimum Vertex $S$-$T$ Cut via $k$-Way Submodularity

**Input:** A graph $G = (V, E)$, a vector $\boldsymbol{y} \in \mathbb{R}^V_{\geq 0}$, $\beta \in [0, 1/2]$, and $\gamma \in (0, 1)$

**for** $i = 1$ *to* $k := \Theta(\beta^{-2} \log(1/\gamma))$ **do**
  Sample $\tau_i \in [-1+\beta, 1-\beta]$ uniformly at random;
  $A_i \leftarrow \{v \in V : y_v \leq \tau_i\}$;
Sample an integer $r$ from $\{k/2, \ldots, (1/2 + \beta/4)k\}$ uniformly at random;
$B_{r,k} \leftarrow \bigcup_{\{i_1,\ldots,i_r\} \in \binom{[k]}{r}} \bigcap_{j=1}^{r} A_{i_j}$;
**return** $B_{r,k}$;

For $\beta \in [0, 1/2]$, we say that a set $A \subseteq V$ is *$\beta$-balanced* if $\min\{|A|, |V \setminus A|\} \geq \beta n$, or in other words, $\beta n \leq |A| \leq (1-\beta)n$. Let $\mathrm{OPT}_\beta$ denote the minimum weight of a $\beta$-balanced $S$-$T$ cut. Using $k$-way submodularity, we provide an algorithm with small pointwise Lipschitz constant that outputs a cut of weight comparable to $\mathrm{OPT}_\beta$. Note that, for a $\beta$-balanced set $A$, we have $\mathbb{1}_A - \frac{|A|}{n}\mathbb{1} \in [-1+\beta, 1-\beta]^V$. Then, we consider Algorithm 2, which first applies threshold rounding multiple times, where the threshold is sampled from a smaller range than $[-1, 1]$, and then applies $k$-way submodularity to aggregate the obtained sets.

**Lemma 25.** *Suppose that a vector $\boldsymbol{y} \in \mathbb{R}^V_{\geq 0}$ is a feasible solution to LP 4. Then, the set $B_{r,k}$ in Algorithm 2 is feasible for $r \in \{k/2, \ldots, (1/2 + \beta/4)k\}$ with probability at least $1 - \gamma$.*

*Proof.* Because $y_{s_0} - y_{t_0} = 1$, for each $i \in [k]$, the set $A_i$ is feasible with probability at least

$$p := \frac{1}{(1-\beta) - (-1+\beta)} = \frac{1}{2-2\beta}.$$

Let $X$ be the number of $i$'s such that $A_i$ is feasible. By a multiplicative Chernoff bound, we have

$$\Pr[X \leq (1-\phi)pk] \leq \exp\left(-\frac{\phi^2 pk}{2}\right)$$

for any $\phi \in (0, 1)$. By setting $\phi = \beta/3$, we have

$$\begin{aligned} (1-\phi)p &= \frac{1}{2}\left(1 - \frac{\beta}{3}\right)\left(\frac{1}{1-\beta}\right) \\ &\geq \frac{1}{2}\left(1 - \frac{\beta}{3}\right)(1+\beta) && (1 + x \leq \tfrac{1}{1-x}, x \in [-1, 1)) \\ &= \frac{1}{2}\left(1 + \frac{2\beta}{3} - \frac{\beta^2}{3}\right) \\ &\geq \frac{1}{2} + \frac{\beta}{4}. && (\beta \in [0, 1/2]) \end{aligned}$$

It follows that

$$\Pr\left[X \le \left(\frac{1}{2}+\frac{\beta}{4}\right)k\right] \le \exp\left(-\Omega(\beta^2 k)\right).$$

By choosing the hidden constant in $k = \Theta(\beta^{-2}\log(1/\gamma))$ to be large enough, we have

$$\Pr\left[X \le \left(\frac{1}{2}+\frac{\beta}{4}\right)k\right] \le \gamma.$$

Suppose the above event does not occur so that at least $(1/2+\beta/4)k+1$ sets are feasible. Reorder the $A_i$'s so that $A_1,\dots,A_{(1/2+\beta/4)k+1}$ are feasible, and let $k/2 \le r \le (1/2+\beta/4)k$. Then, the set $\bigcap_{j=1}^{r} A_j$ is feasible and in particular, non-empty. This is because intersections of feasible cuts are feasible, and under the good event we have conditioned on, the sets $A_1,\dots,A_r$ are feasible. Similarly, it is not hard to see that every intersection $\bigcap_{j=1}^{r} A_{i_j}$ contains at least one feasible set, so this intersection does not contain any elements from $T$. Thus, these two properties show that by taking a union over these intersections $\bigcup_{\{i_1,\dots,i_r\}\in\binom{[k]}{r}} \bigcap_{j=1}^{r} A_{i_j}$ the final set will contain all elements of $S$ and no elements from $T$. □

We first analyze the approximation guarantee of Algorithm 2.

**Lemma 26.** *The output $A \subseteq V$ of Algorithm 2 satisfies*

$$\mathbb{E}[\mathrm{cut}_{G,\boldsymbol{w}}(A)] = O(\beta^{-1} f_{\boldsymbol{w}}(\boldsymbol{y})).$$

*Proof.* By the $k$-way submodularity of $\mathrm{cut}_{G,\boldsymbol{w}}$ (Theorem 24), we have

$$\sum_{r=1}^{k} \mathrm{cut}_{G,\boldsymbol{w}}(A_r) \ge \sum_{r=1}^{k} \mathrm{cut}_{G,\boldsymbol{w}}(B_{r,k}) \ge \sum_{r=k/2}^{(1/2+\beta/4)k} \mathrm{cut}_{G,\boldsymbol{w}}(B_{r,k}).$$

Hence, we have

$$\mathbb{E}[\mathrm{cut}_{G,\boldsymbol{w}}(A)] = \frac{4}{\beta k}\sum_{r=k/2}^{(1/2+\beta/4)k} \mathbb{E}[\mathrm{cut}_{G,\boldsymbol{w}}(B_{r,k})] \le \frac{4}{\beta k}\sum_{r=1}^{k} \mathbb{E}[\mathrm{cut}_{G,\boldsymbol{w}}(A_r)] = O(\beta^{-1} f_{\boldsymbol{w}}(\boldsymbol{y})).$$

□

We use the following simple fact to analyze the pointwise Lipschitz constant of Algorithm 2.

**Proposition 27.** *Let $\mathcal{I}$ be an arbitrary indexing set and $A\Delta B$ denote the symmetric difference between sets $A$ and $B$. Then for any collections of sets indexed by $\mathcal{I}$, $\{A_\alpha\}_{\alpha\in\mathcal{I}}, \{B_\alpha\}_{\alpha\in\mathcal{I}}$, the following hold:*

*(a)* $\left(\bigcup_{\alpha\in\mathcal{I}} A_\alpha\right) \Delta \left(\bigcup_{\alpha\in\mathcal{I}} B_\alpha\right) \subseteq \bigcup_{\alpha\in\mathcal{I}} (A_\alpha \Delta B_\alpha)$

*(b)* $\left(\bigcap_{\alpha\in\mathcal{I}} A_\alpha\right) \Delta \left(\bigcap_{\alpha\in\mathcal{I}} B_\alpha\right) \subseteq \bigcup_{\alpha\in\mathcal{I}} (A_\alpha \Delta B_\alpha)$.

**Lemma 28.** *Let $A$ and $\tilde{A}$ be the outputs of Algorithm 2 for $\boldsymbol{y}$ and $\tilde{\boldsymbol{y}}$, respectively. Then, we have*

$$\mathbb{E}[|A \triangle \tilde{A}|] \le O(\beta^{-2}\log(1/\gamma)\cdot\|\boldsymbol{y}-\tilde{\boldsymbol{y}}\|_1).$$

*Proof.* For any $r \in [k]$, we have

$$\begin{aligned} B_{r,k}\Delta\tilde{B}_{r,k} &= \left(\bigcup_{I\in\binom{[k]}{r}} \bigcap_{i\in I} A_i\right) \Delta \left(\bigcup_{I\in\binom{[k]}{r}} \bigcap_{i\in I} \tilde{A}_i\right) \\ &\subseteq \bigcup_{I\in\binom{[k]}{r}} \left(\left(\bigcap_{i\in I} A_i\right) \Delta \left(\bigcap_{i\in I} \tilde{A}_i\right)\right) \qquad \text{(by Proposition 27 (a))} \end{aligned}$$

$$\subseteq \bigcup_{I \in \binom{[k]}{r}} \bigcup_{i \in I} (A_i \Delta \tilde{A}_i) \qquad \text{(by Proposition 27 (b))}$$
$$\subseteq \bigcup_{j=1}^{k} (A_j \Delta \tilde{A}_j).$$

For each $i \in [k]$, we have $\mathbb{E}[|A_i \triangle \tilde{A}_i|] \leq \|\boldsymbol{y}^* - \tilde{\boldsymbol{y}}^*\|_1$. But then by the calculations above, we have $\mathbb{E}[|A \triangle \tilde{A}|] \leq k \cdot \|\boldsymbol{y}^* - \tilde{\boldsymbol{y}}^*\|_1 = O(\beta^{-2} \log(1/\gamma) \cdot \|\boldsymbol{y}^* - \tilde{\boldsymbol{y}}^*\|_1)$. □

We are now ready to prove Theorem 15

*Proof of Theorem 15.* The approximation guarantee is clear from Theorem 17 and Lemma 26. Lemma 25 gives the feasibility guarantee. To analyze the pointwise Lipschitz constant, first note that Theorem 17 yields

$$\|\boldsymbol{y} - \tilde{\boldsymbol{y}}\|_1 = O\left(\frac{\sqrt{n} \cdot |\delta|}{\lambda_2}\right).$$

Combined with Lemmas 26 and 28, we obtain the claimed bound. □

### D.4 A Lower Bound

In this section, we prove Theorem 16, a lower bound on the pointwise Lipschitz constant for the minimum vertex $S$-$T$ cut problem. This shows that our Lipschitz-approximation tradeoff in Theorem 13 is tight.

*Proof of Theorem 16.* Let $\mathcal{A}$ be an arbitrary $(C, f(n))$-approximation algorithm for the minimum $s$-$t$ cut problem for $C = O(1)$ and let $f(n) < n/2(C+2)$ be an increasing function of $n$. Let $G = (U \cup R, E)$ be a complete bipartite graph, where $|U| = (C+2)f(n)$ and $|R| = n - (C+2)f(n)$. Let $s, t$ be two distinct vertices in $R$. We define two weight vectors $\boldsymbol{w}, \tilde{\boldsymbol{w}} \in [0,1]^E$ as follows:

$$w_e = \begin{cases} 1/4(C+2) & \text{if } s \in e, \\ 1 & \text{otherwise,} \end{cases} \qquad \tilde{w}_e = \begin{cases} 1/4(C+2) & \text{if } t \in e, \\ 1 & \text{otherwise.} \end{cases}$$

The minimum $s$-$t$ cuts of $(G, \boldsymbol{w})$ and $(G, \tilde{\boldsymbol{w}})$ are $\{s\}$ and $V \setminus \{t\}$, respectively, and both have cut weights of $\mathrm{OPT} = |U|/4(C+2) = f(n)/4$. Moreover, for both weighted graphs, any other $s$-$t$ cut has weight at least $|U| = (C+2)f(n) > C \cdot \mathrm{OPT} + f(n)$. Thus $\mathcal{A}$ outputs $\{s\}$ with probability at least $0.99$ given $G, \boldsymbol{w}$ and $V \setminus \{t\}$ with probability at least $0.99$ given $G, \tilde{\boldsymbol{w}}$. Hence, the earth mover's distance between the output distributions of $\mathcal{A}$ on $\boldsymbol{w}$ and that on $\tilde{\boldsymbol{w}}$ is at least

$$0.98 \cdot (n-2) = \Omega(n)$$

Consider a transition of the weight vector from $\boldsymbol{w}$ to $\tilde{\boldsymbol{w}}$, where the edge weights never go below $1/4(C+2)$. It is well known that $\lambda_2(G, \mathbb{1}) = \min\{|U|, |R|\} = |U| = (C+2)f(n)$ because $|U| = (C+2)f(n) < n/2$. Since $\lambda_2$ scales proportionally with weights, $\lambda_2(G, \mathbb{1}/4(C+2)) = |U|/4(C+2) = f(n)/4$. As $\lambda_2$ is increasing in the edge weight of an any edge, the algebraic connectivity of every graph in the transition is at least $f(n)/4$. Hence by Theorem 46, there exists $\boldsymbol{w}^*$ such that the pointwise Lipschitz constant at $\boldsymbol{w}^*$ is at least

$$\Omega\left(\frac{n}{\|\boldsymbol{w} - \tilde{\boldsymbol{w}}\|_1}\right) = \Omega\left(\frac{n}{|U|}\right) = \Omega\left(\frac{n}{f(n)}\right) = \Omega\left(\frac{n}{\lambda_2(G, \boldsymbol{w}^*)}\right).$$

□

## E Maximum Weight Bipartite *b*-Matching

In this section, we design a bipartite matching algorithm which cleanly extends to handle ***b***-matchings. Our goal is to prove the following results.

**Theorem 29.** *For any bipartite graph $G = (V = U \cup R, E)$, weight vector $\boldsymbol{w} \in \mathbb{R}_{>0}$, and approximation parameter $\varepsilon > 0$, Algorithm 3 runs in polynomial time and returns a $2(1+\varepsilon)$-approximate maximum weighted matching in expectation, with pointwise Lipschitz constant $O\left(\frac{\sqrt{m}}{\varepsilon w_{\min}}\right)$.*

**Theorem 30.** *Fix $\varepsilon > 0$. Algorithm 4 terminates in $\mathrm{poly}(n)$ time and outputs a $2(1+\varepsilon)e/(e-1)$-approximate maximum weight $\boldsymbol{b}$-matching in expectation for bipartite graphs. Moreover, its pointwise Lipschitz constant is $O\left(\frac{\sqrt{m}}{\varepsilon w_{\min}}\right)$.*

In Appendix E.1, we design and analyze our bipartite matching algorithm. Having established the general analysis of PGTA, the analysis of the fractional Lipschitz solution is relatively painless. We directly use the Kumabe and Yoshida [2023] rounding algorithm for bipartite matchings.

In Appendix E.2, we describe how to obtain fractional Lipschitz solutions for bipartite $\boldsymbol{b}$-matchings. We can no longer directly use the Kumabe and Yoshida [2023] rounding scheme in this setting and we design a more involved rounding scheme inspired by their rounding algorithm in Appendix E.3.

### E.1 Warmup for Bipartite 1-Matching

The LP that obtains a bipartite maximum (unweighted and weighted) matching is as follows. Let $U$ and $R$ be the sets of the vertices on the left hand side and right hand sides of the bipartite graph and $\boldsymbol{w} = \{w_{uv}\}_{uv \in E}$ is the vector representing the weights of every edge. We regularize the standard LP relaxation using the simple strongly convex regularizer of $\sum_{uv \in E} w_{uv} x_{uv}^2$.

$$\begin{aligned} \text{minimize} \quad & \sum_{uv \in E} -w_{uv} x_{uv} + \frac{\varepsilon}{2} \cdot \sum_{uv \in E} w_{uv} x_{uv}^2 & \\ \text{subject to} \quad & \sum_{v \in R} x_{uv} \leq 1 & \forall u \in U \\ & \sum_{u \in L} x_{uv} \leq 1 & \forall v \in R \\ & 0 \leq x_{uv} \leq 1 & \forall uv \in E. \end{aligned} \tag{5}$$

We now proceed with the framework introduced in Appendix C and the notation used within that section. The next result shows that the conditions in Assumption 9 are satisfied.

**Lemma 31.** *The convex program in LP 5 satisfies the conditions in Assumption 9 with $L_{\boldsymbol{w}} = 2\varepsilon w_{\max}, \sigma_{\boldsymbol{w}} = \varepsilon w_{\min}/2, C_{\boldsymbol{w}} = 1/(2w_{\max}), D_{\boldsymbol{w}} = 1/(2\varepsilon w_{\max})$*

*Proof.* Let $\boldsymbol{w}$ be a weight vector and let $\tilde{\boldsymbol{w}} = \boldsymbol{w} + \delta \cdot \mathbb{1}_e$, for some $e \in E$ and $0 \neq \delta \in \mathbb{R}$ that is sufficiently small. We set $f_{\boldsymbol{w}}(\boldsymbol{x}) = -\langle \boldsymbol{w}, \boldsymbol{x} \rangle$ and $g_{\boldsymbol{w}}(\boldsymbol{x}) = \varepsilon \cdot \langle \boldsymbol{w}, \boldsymbol{x}^{\circ 2} \rangle$, where $\boldsymbol{x}^{\circ 2}$ denotes the vector obtained from $\boldsymbol{x}$ by multiplying each entry with itself. Let $L = 2\varepsilon w_{\max}, \sigma = \varepsilon w_{\min}/2$. Then, for $|\delta|$ that is sufficiently small, we see that $g_{\boldsymbol{w}}(\cdot)$ and $g_{\tilde{\boldsymbol{w}}}(\cdot)$ are both $L$-smooth, $\sigma$-strongly convex.

Furthermore, for every $\boldsymbol{y} \in K$, since $K \subseteq [0,1]^E$,

$$\frac{1}{L} \|\boldsymbol{\nabla} g_{\boldsymbol{w}}(\boldsymbol{y}) - \boldsymbol{\nabla} g_{\tilde{\boldsymbol{w}}}(\boldsymbol{y})\|_2 \leq \frac{\varepsilon}{L} |\delta|.$$

Thus our functions satisfy Condition 4 with $C_{\boldsymbol{w}} = \varepsilon/L$.

Finally, for every $\boldsymbol{y} \in K$, let $\boldsymbol{z} = \mathrm{prox}_{L, f_{\boldsymbol{w}}}(\boldsymbol{y})$ and $\tilde{\boldsymbol{z}} = \mathrm{prox}_{L, f_{\tilde{\boldsymbol{w}}}}(\boldsymbol{y})$. A similar calculation we made in the proof of Lemma 20 using the $L$-strong convexity of the proximal objective value yields the following, where $e \in E$ is the edge that differs in weight,

$$L\|\boldsymbol{z} - \tilde{\boldsymbol{z}}\|_2^2 \leq f_{\boldsymbol{w}}(\tilde{\boldsymbol{z}}) - f_{\boldsymbol{w}}(\boldsymbol{z}) + f_{\tilde{\boldsymbol{w}}}(\boldsymbol{z}) - f_{\tilde{\boldsymbol{w}}}(\tilde{\boldsymbol{z}}) = |\delta|(\tilde{z}_e - z_e) \leq |\delta| \|\boldsymbol{z} - \tilde{\boldsymbol{z}}\|_2.$$

In other words,

$$\|\mathrm{prox}_{L, f_{\boldsymbol{w}}}(\boldsymbol{y}) - \mathrm{prox}_{L, f_{\tilde{\boldsymbol{w}}}}(\boldsymbol{y})\|_2 \leq \frac{1}{L} |\delta|$$

and we satisfy Condition 5 with $D_{\boldsymbol{w}} = 1/L$. □

Since the regularized objective satisfies the conditions of Assumption 9, we get the following result as a corollary.

**Corollary 32.** *Let $\boldsymbol{x}^*$ be the optimal solution of LP 5 under weight vector $\boldsymbol{w}$ and $\tilde{\boldsymbol{x}}^*$ be the optimal solution of LP 5 under weight vector $\tilde{\boldsymbol{w}} = \boldsymbol{w} + \delta \cdot \mathbb{1}_e$, for some $e \in E$ and $0 \neq \delta \in \mathbb{R}$ that has sufficiently small absolute value. Then, $\|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}^*\|_1 \leq \frac{2\sqrt{m}}{w_{\min}}(1 + 1/\varepsilon)|\delta|$.*

*Proof.* Since LP 5 satisfies Assumption 9 with $L_{\boldsymbol{w}} = 2\varepsilon w_{\max}, \sigma_{\boldsymbol{w}} = \varepsilon w_{\min}/2, C_{\boldsymbol{w}} = 1/(2w_{\max}), D_{\boldsymbol{w}} = 1/(2\varepsilon w_{\max})$, Theorem 10 shows that for the optimal solutions $\boldsymbol{x}^*, \tilde{\boldsymbol{x}}^*$,

$$\begin{aligned}\|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}^*\|_1 &\leq \sqrt{m}\|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}^*\|_2 \\ &\leq \sqrt{m}\frac{L_{\boldsymbol{w}}(C_{\boldsymbol{w}} + D_{\boldsymbol{w}})}{\sigma_{\boldsymbol{w}}}|\delta| \\ &= \frac{2\sqrt{m}}{w_{\min}}(1 + 1/\varepsilon)|\delta|.\end{aligned}$$

□

We use the above calculations to show the following lemma on the Lipschitz constant and approximation factor of our algorithm.

**Lemma 33.** *For any bipartite graph $G = (V = U \cup R, E)$ and weight vector $\boldsymbol{w} \in R^E_{>0}$ the optimal solution of LP 5 is a $(1 + \varepsilon/2)$-approximation to the fractional maximum weight bipartite matching of $G, \boldsymbol{w}$. Moreover, let $\boldsymbol{x}^*$ be the optimal solution of LP 5 under weight vector $\boldsymbol{w}$ and $\tilde{\boldsymbol{x}}^*$ be the optimal solution of LP 5 under weight vector $\tilde{\boldsymbol{w}} = \boldsymbol{w} + \delta \cdot \mathbb{1}_e$, for some $e \in E$ and $0 \neq \delta \in \mathbb{R}$ that has sufficiently small absolute value. Then, $\|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}^*\|_1 \leq \frac{2\sqrt{m}}{w_{\min}}(1 + 1/\varepsilon)|\delta|$.*

*Proof.* The Lipschitz constant directly follows from Corollary 32. To get our approximation factor, notice that $\sum_{uv \in E} w_{uv} x_{uv}^2 \leq \sum_{uv \in E} w_{uv} x_{uv}$ for any $x_{uv} \in [0, 1]$ and $uv \in E$. Then, it holds that

$$\begin{aligned}\sum_{uv} w_{uv} x_{uv} &\geq \sum_{uv} w_{uv} x_{uv} - \frac{\varepsilon}{2}\sum_{uv} w_{uv} x_{uv}^2 \\ &\geq \sum_{uv} w_{uv} x^*_{uv} - \frac{\varepsilon}{2}\sum_{uv} w_{uv} (x^*_{uv})^2 \\ &\geq \left(1 - \frac{\varepsilon}{2}\right)\sum_{uv} w_{uv} x^*_{uv}.\end{aligned}$$

which gives a $(1 + \varepsilon/2)$-approximation of the maximum fractional matching. □

We use the rounding scheme given by Kumabe and Yoshida [2023] (see Algorithm 6, Lines 4-13 in Kumabe and Yoshida [2023]) to round our fractional solution to an integral solution. Specifically, their paper implies the following approximation and Lipschitz constant guarantees about rounding a fractional matching solution to an integral matching.

**Theorem 34** (Sections 7.2, 8.3.5 in Kumabe and Yoshida [2023])**.** *Given any feasible fractional matching $\boldsymbol{x} = \{x_{ij}\}_{i \in U, j \in V}$ satisfying the bipartite matching LP, there exists a rounding scheme $\mathcal{A}_\pi$ that produces a matching $\mathbf{M}_\pi = \mathcal{A}_\pi(\boldsymbol{x})$ such that*

$$\mathbb{E}\left[\sum_{e \in \mathbf{M}_\pi} w_e\right] \geq \frac{1}{2}\sum_{ij \in E} w_{ij} x_{ij},$$

*where $w_e$ is the weight of edge $e$. Furthermore, it holds that under the optimal coupling $\pi, \tilde{\pi} \sim \mathcal{D}$,*

$$\mathbb{E}_{(\pi, \tilde{\pi}) \sim \mathcal{D}}\left[\|\mathbb{1}_{\mathbf{M}_\pi} - \mathbb{1}_{\tilde{\mathbf{M}}_{\tilde{\pi}}}\|_1\right] = O\left(\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|_1\right).$$

Our approach is summarized in Algorithm 3.

Combining the above theorem with Lemma 33 yields the proof of Theorem 29.

### E.2 Obtaining a Fractional Lipschitz Solution

We solve maximum weight bipartite $\boldsymbol{b}$-matching in a similar way to our solution for maximum weight bipartite matching, with the main difference being the rounding schemes we use for these two problems. The LP for maximum matching with the corresponding regularizer is given below, where $b_v$ is the constraint on the maximum number of matched neighbors to vertex $v$.

**Algorithm 3:** Maximum Bipartite Matching

**Input:** A graph $G = (V = U \cup R, E)$, a weight vector $\boldsymbol{w} \in \mathbb{R}^E_{>0}$, an approximation parameter $\varepsilon > 0$.

Solve LP 5 and let $\boldsymbol{x}^* \in \mathbb{R}^E$ be the obtained solution;

$\hat{\boldsymbol{x}} \leftarrow$ rounding scheme from Kumabe and Yoshida [2023] ; `// See Algorithm 6, Lines 4-13 of Kumabe and Yoshida [2023]`

Return $\hat{\boldsymbol{x}}$

$$\begin{aligned} \text{minimize} \quad & \sum_{uv \in E} -w_{uv} x_{uv} + \frac{\varepsilon}{2} \cdot \sum_{uv \in E} w_{uv} x_{uv}^2 & \\ \text{subject to} \quad & \sum_{v \in R} x_{uv} \leq b_u & \forall u \in U \\ & \sum_{u \in U} x_{uv} \leq b_v & \forall v \in R \\ & 0 \leq x_{uv} \leq 1 & \forall u, v \in E. \end{aligned} \tag{6}$$

As in the maximum weight bipartite matching case, it is classically known that the optimum solution to the $\boldsymbol{b}$-matching LP equals the maximum solution of the integer program. Since we use the same regularizer as in Appendix E.1 and we are still optimizing over a subset of $[0,1]^E$, we obtain the same Lipschitz constant using the PGM framework summarized below. The proof of the approximation factor also follows from the proof of Lemma 33.

**Lemma 35.** *For any $\varepsilon > 0$, the optimal solution of LP 6 gives a $(1 + \varepsilon/2)$-approximate maximum fractional $\boldsymbol{b}$-matching. Moreover, let $\boldsymbol{x}^*, \tilde{\boldsymbol{x}}^*$ be the optimal solutions of LP 6 under $\boldsymbol{w}, \tilde{\boldsymbol{w}} = \boldsymbol{w} + \delta \mathbb{1}_e$, for some $e \in E$ and $0 \neq \delta \in \mathbb{R}$, with sufficiently small absolute value. Then, $\|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}^*\|_1 = O\left(\frac{\sqrt{m}}{\varepsilon w_{\min}}\right)$.*

### E.3 Rounding via Multi-Item Cooperative Auction

We extend the fractional bipartite matching rounding scheme from [Kumabe and Yoshida, 2023, Section 7.2] (Theorem 34) for the case of fractional bipartite $\boldsymbol{b}$ matchings. We can view the following rounding algorithm as a *contention resolution scheme* Vondrák et al. [2011].

**Algorithm 4:** Maximum Bipartite $\boldsymbol{b}$-Matching

**Input:** A graph $G = (V = U \cup R, E)$, a weight vector $\boldsymbol{w} \in \mathbb{R}^E_{>0}$, approximation error $\varepsilon$

Solve LP 6 and let $\boldsymbol{x}^* \in \mathbb{R}^E$ be the obtained solution;

$p(u, v_j) \leftarrow 0$ for all $u \in U, v \in R$ and $j \in [b_v]$ ; `// Bidding Phase`

**for** *each buyer* $u \in U$ **do**

  Sample sellers $v(u,1), \ldots, v(u,b_u)$ *i.i.d.* from

$$\begin{cases} v \in N(u) \cup \{\perp\}, & \text{with probability } \frac{x^*_{uv}}{b_u} \\ \perp, & \text{with remaining probability} \end{cases}$$

  ;

  **for** *each unique seller* $\perp \neq v \in \{v(u,i), \ldots, v(u,b_u)\}$ **do**

    $u$ bids on an item $v_j \in \{v_1, \ldots, v_{b_v}\}$ uniformly at random;

    $p(u, v_j) \leftarrow 1$;

$q(u, v_j) \leftarrow 0$ for all $u \in U, v \in R$ and $j \in [b_v]$ ; `// Accepting Phase`

**for** *each seller* $v \in R$ *and* $j \in [b_v]$ **do**

  $v$ accepts a bid for $v_j$ uniformly at randomly from $\{u \in U : p(u, v_j) = 1\}$;

  $q(u, v_j) \leftarrow 1$;

Return $M \leftarrow \{uv : \exists j \in [b_v], q(u, v_j) = 1\}$

We interpret the rounding algorithm as a cooperative auction between buyers $u \in U$ and sellers $v \in R$, where each buyer $u$ can buy at most $b_u$ unique items from its neighboring sellers $N(u)$ and each seller $v$ has $b_v$ identical items to sell. Each transaction between $u, v$ yields a gain of $w_{uv}$ and buyers and sellers attempt to maximize the total gain. In the bidding phase, each buyer $u \in U$ selects a set of at most $b_u$ sellers and bids on one item from each selected seller. In the accepting phase, each seller $v \in R$ accepts a bid uniformly at random for each item $v_j, j \in [b_v]$ whose set of bidders is non-empty.

**Lemma 36.** *Let edge weights $\tilde{\boldsymbol{w}} \in \mathbb{R}^E_{>0}$ be obtained by perturbing $\boldsymbol{w} \in \mathbb{R}^E_{>0}$ at a single coordinate by some sufficiently small $\delta > 0$. Let $\boldsymbol{x}^*, \tilde{\boldsymbol{x}}^* \in [0,1]^E$ be the optimal solutions to LP 6 resulting from $\boldsymbol{w}, \tilde{\boldsymbol{w}}$, respectively. Let $p(u, v_j) \coloneqq \mathbb{1}\{u \text{ bids on } v_j\}$ be the indicator variable of bids obtained when Algorithm 4 is executed on $\boldsymbol{w}$ and similarly $\tilde{p}(u, v_j)$ when it is executed on $\tilde{\boldsymbol{w}}$. Then the following holds:*

*(a)* $\mathbb{E}\left[\sum_{u \in U, v \in N(u), j \in [b_v]} w_{uv} p(u, v_j)\right] \geq (1 - 1/e)\mathrm{OPT}_{\mathrm{LP}}$

*(b)* $\mathrm{EMD}_d(\boldsymbol{p}, \tilde{\boldsymbol{p}}) = \|\boldsymbol{x}^* - \tilde{\boldsymbol{x}}^*\|_1$

*Proof of Lemma 36.* We prove each statement separately.

**Proof of (a).** It suffices to compute the marginal probability $\mathbb{E}[p(u, v_j)]$ of a buyer $u \in U$ bidding on the item $v_j$ from a seller $v \in N(u)$ and lower bound it by $(1 - 1/e)\frac{x^*_{uv}}{b_v}$. Then the result follows from the linearity of expectation. In fact, we show that the probability that $u$ bids on some item from $v \in N(v)$ is at least $(1 - 1/e)x^*_{uv}$. Since the specific item is chosen uniformly at random, the result follows.

Consider the equivalent experiment for generating a set of at most $b_u$ sellers as follows: For $i = 1, 2, \ldots, b_u$, we sample a seller $v \in N(u)$ with probability $\frac{x^*_{uv}}{b_v}$ and do nothing with the remaining probability. Notice that since $\sum_{v \in N(u)} x^*_{uv} \leq b_u$, it is indeed the case that $\sum_{v \in N(u)} \frac{x^*_{uv}}{b_u} \leq 1$. If we have not yet chosen $v$ in a prior iteration $i' < i$, then add $v$ to the set of sellers. The event that the neighbor $v \in N(u)$ is drawn at least once is the complement of the event that it is never picked after $b_u$ draws. Hence

$$\begin{aligned}
\Pr[u \text{ selects } v] &= 1 - \left(1 - \frac{x^*_{uv}}{b_u}\right)^{b_u} \\
&\geq 1 - e^{-x^*_{uv}} && (1 + z \leq e^z) \\
&\geq \left(1 - \frac{1}{e}\right) x^*_{uv}. && \text{(concavity)}
\end{aligned}$$

The last inequality can be verified by noting that we have equality at $x^*_{uv} = 0, 1$ and that $1 - e^{-x^*_{uv}}$ is a concave function of $x^*_{uv}$.

**Proof of (b).** It suffices to show that for each $u \in U$, under the optimal coupling $(\boldsymbol{p}, \tilde{\boldsymbol{p}}) \sim \mathcal{D}$,

$$\mathop{\mathbb{E}}_{(\boldsymbol{p}, \tilde{\boldsymbol{p}}) \sim \mathcal{D}} \left[\sum_{v \in N(u), j \in [b_v]} |p(u, v_j) - \tilde{p}(u, v_j)|\right] \leq \sum_{v \in N(u)} |x^*_{uv} - \tilde{x}^*_{uv}|.$$

Then the conclusion follows by summing over all $u \in U$ and the linearity of expectation.

First, we analyze

$$\begin{aligned}
&\sum_{v \in N(u)} \left| \sum_{j \in [b_v]} p(u, v_j) - \sum_{j \in [b_v]} \tilde{p}(u, v_j) \right| \\
&= \sum_{v \in N(u)} |\mathbb{1}\{u \text{ selects seller } v \text{ under weights } w\} - \mathbb{1}\{u \text{ selects seller } v \text{ under weights } \tilde{\boldsymbol{w}}\}|.
\end{aligned}$$

This is precisely the cardinality of the symmetric difference of the set of sellers selected by $u$. For the

final set of sellers to differ at some $v \in N(u)$, the number of times $v$ was selected in the multiset of selected sellers must have differed. The total variation distance of a single selected seller is $\sum_{v \in N(u)} \left| \frac{x^*_{uv}}{b_u} - \frac{\tilde{x}^*_{uv}}{b_u} \right|$. Hence the expected cardinality of the symmetric difference of the multiset within the optimal coupling is $\sum_{v \in N(u)} |x^*_{uv} - \tilde{x}^*_{uv}|$ and we have the upper bound

$$\sum_{v \in N(u)} \left| \sum_{j \in [b_v]} p(u, v_j) - \sum_{j \in [b_v]} \tilde{p}(u, v_j) \right| \leq \sum_{v \in N(u)} |x^*_{uv} - \tilde{x}^*_{uv}|.$$

Now, if we consider a coupling where we are sharing the internal randomness, $p(u, v_j), \tilde{p}(u, v_j)$ can only differ if $u$ was inconsistent in including $v$ in the set of sellers. In other words, for any $u \in U, v \in N(u)$, we have

$$\sum_{j \in [b_v]} |p(u, v_j) - \tilde{p}(u, v_j)| = \left| \sum_{j \in [b_v]} p(u, v_j) - \sum_{j \in b_v} \tilde{p}(u, v_j) \right|.$$

It follows that

$$\begin{aligned}
&\mathop{\mathbb{E}}_{(\boldsymbol{p}, \tilde{\boldsymbol{p}}) \sim \mathcal{D}} \left[ \sum_{v \in N(u), j \in [b_v]} |p(u, v_j) - \tilde{p}(u, v_j)| \right] \\
&= \mathop{\mathbb{E}}_{(\boldsymbol{p}, \tilde{\boldsymbol{p}} \sim \mathcal{D})} \left[ \sum_{v \in N(u)} \left| \sum_{j \in [b_v]} p(u, v_j) - \sum_{j \in [b_v]} \tilde{p}(u, v_j) \right| \right] \\
&\leq \sum_{v \in N(u)} |x^*_{uv} - \tilde{x}^*_{uv}|.
\end{aligned}$$

□

For the accepting phase, we observe that it is identical to the case of bipartite matching since each item is sold to its set of bidders uniformly at random. The approximation and Lipschitz guarantees follow almost immediately from the guarantees for bipartite matching Kumabe and Yoshida [2023]. We write $q(v_j, u)$ to be the indicator variable that item $v_j$ is sold to $u \in N(v)$.

For the approximation guarantees, an important observation from Kumabe and Yoshida [2023] is that the number of bids for an item in Algorithm 4 follows a *Poisson Binomial* distribution. Recall $\mathrm{PoBin}(y_1, \ldots, y_n)$ is the number of heads from flipping $n$ independent coins with biases $y_1, \ldots, y_n$. The following lemma ensures that we lose at most half the bids in expectation within the acceptance phase.

**Lemma 37** (Lemmas 7.10, 7.11, 7.12 in Kumabe and Yoshida [2023])**.** *Suppose $u \in U$ bids on an item $v_j$ for $v \in N(u), j \in [b_v]$ and the total number of other bids for $v_j$ follows a Poisson Binomial distribution $\mathrm{PoBin}(y_{u'v_j})_{u \neq u' \in N(v)}$ whose parameters $y_{u'v_j} \in [0, 1], u \neq u' \in N(v)$ satisfy $\sum_{u \neq u' \in N(v)} y_{u'v_j} \leq 1$. Then, if $v_j$ is sold to a bidder uniformly at random, $v_j$ is sold to $u$ with probability at least $1/2$.*

For $p(u, v_j) \coloneqq \mathbb{1}\{u \text{ bids on } v_j\}$ and $q(u, v_j) \coloneqq \mathbb{1}\{v_j \text{ is sold to } u\}$ from Algorithm 4, Lemma 37 states that

$$\Pr[q(u, v_j) = 1 \mid p(u, v_j) = 1] \geq \frac{1}{2}.$$

This is because for each item $v_j$ we flip $|N(v)|$ different independent coins in total and for every $u \in N(v)$, the coin corresponding to $u$ has bias at most $x_{uv}/b_v$. Notice that for any $u \in N(v)$,

$$\sum_{u \neq u' \in N(v)} \frac{x_{uv}}{b_v} \leq 1,$$

so the conditions of Lemma 37 are satisfied.

For the Lipschitz guarantees, the following lemma reduces bounding the Lipschitz constant of the final $\boldsymbol{b}$-matching to bounding the Lipschitz constant of the bids themselves.

**Lemma 38** (Lemma 7.18 in Kumabe and Yoshida [2023]). *Let $p(u, v_j), \tilde{p}(u, v_j), u \in U, v \in N(u), j \in [b_v]$ be two realizations of indicator variables that a buyer $u \in U$ bids on some item $v_j, j \in [b_v]$ from a seller $v \in N(u)$. Suppose each item is sold uniformly at random to one of its bidders (if there are any), and let $q(u, v_j), \tilde{q}(u, v_j), u \in U, v \in N(u), j \in [b_v]$ be the indicator variables that $v_j$ is sold to $u$ conditioned on $\boldsymbol{p}, \tilde{\boldsymbol{p}}$, respectively. Then under the conditional optimal coupling $(\boldsymbol{q}, \tilde{\boldsymbol{q}}) \sim \mathcal{D}_{p,\tilde{p}}$,*

$$\mathop{\mathbb{E}}_{(\boldsymbol{q},\tilde{\boldsymbol{q}})\sim\mathcal{D}_{\boldsymbol{p},\tilde{\boldsymbol{p}}}}[\|\boldsymbol{q} - \tilde{\boldsymbol{q}}\|_1 \mid \boldsymbol{p}, \tilde{\boldsymbol{p}}] \leq 2\|p - \tilde{p}\|_1.$$

Let edge weights $\tilde{\boldsymbol{w}} \in \mathbb{R}^E_{>0}$ be obtained by perturbing $\boldsymbol{w} \in \mathbb{R}^E_{>0}$ at a single coordinate by some small $\delta > 0$. Let $p(u, v_j) \coloneqq \mathbb{1}\{u \text{ bids on } v_j\}$ to be the indicator variable that $u$ bids on the item $v_j$ from seller $v \in N(u)$ when Algorithm 4 is executed on $\boldsymbol{w}$ and similarly $\tilde{p}(u, v_j)$ when executed on $\tilde{\boldsymbol{w}}$. Define $q(u, v_j) \coloneqq \mathbb{1}\{v_j \text{ is sold to } u\}$ to be the indicator variable that the item $v_j$ is sold to $u$ when Algorithm 4 is executed on $\boldsymbol{w}$ and similarly $\tilde{q}(u, v_j)$ when executed on $\tilde{\boldsymbol{w}}$. Fix a realization $\boldsymbol{p}, \tilde{\boldsymbol{p}}$. Lemma 38 states that under the conditional optimal coupling $(\boldsymbol{q}, \tilde{\boldsymbol{q}}) \sim \mathcal{D}_{\boldsymbol{p},\tilde{\boldsymbol{p}}}$,

$$\mathop{\mathbb{E}}_{(\boldsymbol{q},\tilde{\boldsymbol{q}})\sim\mathcal{D}_{\boldsymbol{p},\tilde{\boldsymbol{p}}}}[\|\boldsymbol{q} - \tilde{\boldsymbol{q}}\|_1 \mid \boldsymbol{p}, \tilde{\boldsymbol{p}}] \leq 2\|\boldsymbol{p} - \tilde{\boldsymbol{p}}\|_1.$$

We are now ready to prove Theorem 30.

*Proof of Theorem 30.* We show the approximation and Lipschitz guarantees separately.

**Approximation Guarantee.** Let $\boldsymbol{x}$ denote the optimal solution of LP 6 and $M$ denote an output $\boldsymbol{b}$-matching. By the linearity of expectation,

$$\begin{aligned}
&\mathbb{E}\left[\sum_{e\in E} w_e \mathbb{1}\{e \in M\}\right] \\
&= \sum_{u\in U}\sum_{v\in N(v)}\sum_{j\in[b_v]} w_{uv} \Pr[q(u, v_j) = 1 \mid p(u, v_j) = 1] \cdot \Pr[p(u, v_j) = 1] \\
&\geq \frac{1}{2}\sum_{u\in U}\sum_{v\in N(v)}\sum_{j\in[b_v]} w_{uv} \Pr[p(u, v_j) = 1] && \text{(Lemma 37)} \\
&\geq \frac{1}{2}\left(1 - \frac{1}{e}\right)\mathrm{OPT}_{\mathrm{LP}}. && \text{(Lemma 36)}
\end{aligned}$$

Since $\mathrm{OPT}_{\mathrm{LP}} \geq (1 - \varepsilon)\mathrm{OPT}$ by Lemma 35, we have shown the approximation guarantees as desired.

**Pointwise Lipschitz Constant.** Let edge weights $\tilde{\boldsymbol{w}} \in \mathbb{R}^E_{>0}$ be obtained by perturbing $\boldsymbol{w} \in \mathbb{R}^E_{>0}$ at a single coordinate by some small $\delta > 0$. Let $p(u, v_j) \coloneqq \mathbb{1}\{u \text{ bids on } v_j\}$, $q(u, v_j) \coloneqq \mathbb{1}\{v_j \text{ is sold to } u\}$ be the indicator variables that $u$ bids on the item $v_j$ from seller $v \in N(u)$ and the item $v_j$ is sold to $u$, respectively, when Algorithm 4 is executed on $\boldsymbol{w}$. Similarly, define $\tilde{p}(u, v_j)$, $\tilde{q}(u, v_j)$ when the algorithm is executed on $\tilde{\boldsymbol{w}}$.

Consider the coupling $(\boldsymbol{q}, \tilde{\boldsymbol{q}}) \sim \mathcal{D}_{\boldsymbol{p},\tilde{\boldsymbol{p}}} \circ \mathcal{D}$ obtained by sampling from the optimal coupling $(\boldsymbol{p}, \tilde{\boldsymbol{p}}) \sim \mathcal{D}$ (cf. Lemma 36) and then sampling from the conditional optimal coupling $(\boldsymbol{q}, \tilde{\boldsymbol{q}}) \sim \mathcal{D}_{\boldsymbol{p},\tilde{\boldsymbol{p}}}$ (cf. Lemma 38). We have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{(\boldsymbol{q},\tilde{\boldsymbol{q}})\sim\mathcal{D}_{\boldsymbol{p},\tilde{\boldsymbol{p}}}\circ\mathcal{D}}[\|\boldsymbol{q} - \tilde{\boldsymbol{q}}\|_1] \\
&= \mathop{\mathbb{E}}_{(\boldsymbol{p},\tilde{\boldsymbol{p}})\sim\mathcal{D}}\left[\mathop{\mathbb{E}}_{(\boldsymbol{q},\tilde{\boldsymbol{q}})\sim\mathcal{D}_{\boldsymbol{p},\tilde{\boldsymbol{p}}}}[\|\boldsymbol{q} - \tilde{\boldsymbol{q}}\|_1 \mid \boldsymbol{p}, \tilde{\boldsymbol{p}}]\right] \\
&\leq \mathop{\mathbb{E}}_{(\boldsymbol{p},\tilde{\boldsymbol{p}})\sim\mathcal{D}}[2\|\boldsymbol{p} - \tilde{\boldsymbol{p}}\|_1] && \text{(Lemma 38)} \\
&= 2\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|_1 && \text{(Lemma 36)} \\
&= O\left(\frac{\sqrt{m}}{\varepsilon w_{\min}}\right). && \text{(Lemma 35)}
\end{aligned}$$

□

## F Pointwise Lipschitz Continuity under Shared Randomness

In this section, we discuss pointwise Lipschitz continuity under shared randomness, which is formally defined below.

**Definition 39** (Kumabe and Yoshida [2023])**.** Suppose the (randomized) algorithm $\mathcal{A}(G, \boldsymbol{w})$ outputs subsets of some universe $\mathfrak{S}$. Let $d : 2^{\mathfrak{S}} \times 2^{\mathfrak{S}} \to \mathbb{R}_+$ be a metric on $2^{\mathfrak{S}}$. We say that $A$ has *pointwise Lipschitz constant $c_{\boldsymbol{w}}$ under shared randomness with respect to $d$* if

$$\limsup_{\tilde{\boldsymbol{w}} \to \boldsymbol{w}} \mathop{\mathbb{E}}_{\pi} \left[ \frac{d(\mathcal{A}_\pi(G, \boldsymbol{w}) - \mathcal{A}_\pi(G, \tilde{\boldsymbol{w}}))}{\|\boldsymbol{w} - \tilde{\boldsymbol{w}}\|_1} \right] \leq c_{\boldsymbol{w}}.$$

[Kumabe and Yoshida, 2023, Section 8] demonstrated how to implement a pointwise Lipschitz algorithm using shared randomness at a constant factor blowup of the Lipschitz constant when the original algorithm employs internal randomness to sample from a uniform continuous distribution or a discrete distribution. Specifically, let `Sample()` be a sampling process that takes two real values $a, b \in \mathbb{R}$ and a vector $\boldsymbol{p} \in [0,1]^{\mathbb{Z}_{\geq 0}}$ and outputs a real value. For $c \geq 1$, we say that `Sample()` is *$c$-stable* for a pair of functions $(l, r)$ with $l, r : \mathbb{R}^E_{\geq 0} \to \mathbb{R}$ if (i) for any $\boldsymbol{w} \in \mathbb{R}^E_{\geq 0}$, `Sample`$(l(\boldsymbol{w}), r(\boldsymbol{w}), p)$ is uniformly distributed over $[l(\boldsymbol{w}), r(\boldsymbol{w})]$ when $\boldsymbol{p}$ follows the uniform distribution over $[0,1]^{\mathbb{Z}_{\geq 0}}$, and (ii) for any $\boldsymbol{w}, \tilde{\boldsymbol{w}} \in \mathbb{R}^E_{\geq 0}$, we have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\boldsymbol{p} \sim \mathcal{U}\left([0,1]^{\mathbb{Z}_{\geq 0}}\right)} \left[\mathrm{TV}\left(\texttt{Sample}(l(\boldsymbol{w}), r(\boldsymbol{w}), \boldsymbol{p}), \texttt{Sample}(l(\tilde{\boldsymbol{w}}), r(\tilde{\boldsymbol{w}}), \boldsymbol{p})\right)\right] \\
&\leq c \cdot \mathrm{TV}\left(\mathcal{U}([l(\boldsymbol{w}), r(\boldsymbol{w})]), \mathcal{U}([l(\tilde{\boldsymbol{w}}), r(\tilde{\boldsymbol{w}})])\right)
\end{aligned}$$

holds. Then, they gave the following useful sampling processes.

**Lemma 40** (Kumabe and Yoshida [2025])**.** *Let $l$ and $r$ be constant functions. Then, there is a $1$-stable sampling process for $(l, r)$.*

**Lemma 41** (Kumabe and Yoshida [2025])**.** *Let $c > 1$ and suppose $r(w) = cl(w)$ holds for all $w$. Then, there is a $(1 + c)$-stable sampling process for $(l, r)$.*

Using these sampling processes, it is easy to see that most of the sampling processes used in our rounding schemes can be transformed for the shared randomness setting such that the pointwise Lipschitz constant blows up by a constant factor. See Kumabe and Yoshida [2025, 2023] for more details.

The only exception is the exponential mechanism used in Algorithm 1. Recall that the exponential mechanism takes a vector $\boldsymbol{x} \in \mathbb{R}^n$, generates a vector $\boldsymbol{p} \in \mathbb{R}^n$ such that $p_i = \exp(-\eta x_i) / \sum_{j \in [n]} \exp(-\eta x_j)$ for $i \in [n]$, and sample an index $i$ with probability $p_i$. If we can bound $\|\boldsymbol{p} - \tilde{\boldsymbol{p}}\|_1$ by using $\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|_1$, then we can bound the pointwise Lipschitz constant under shared randomness by using [Kumabe and Yoshida, 2025, Lemma 8.4].

**Lemma 42.** *We have*

$$\|\boldsymbol{p} - \tilde{\boldsymbol{p}}\|_1 \leq O(\eta \cdot \|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|_1).$$

*Proof.* We assume that there exists $\hat{\imath}$ and $\delta \neq 0$ such that

$$\tilde{x}_i = \begin{cases} x_i + \delta & \text{if } i = \hat{\imath} \\ x_i & \text{otherwise.} \end{cases}$$

If we can show the claim with this assumption, then we can show the claim for general $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ by triangle inequality. Without loss of generality, we can assume that $\delta > 0$.

Let $Z = \sum_{i \in [n]} \exp(-\eta x_i)$ and $\tilde{Z} = \sum_{i \in [n]} \exp(-\eta \tilde{x}_i)$. We note that $Z \geq \tilde{Z}$. Then, we have

$$\begin{aligned}
\|\boldsymbol{p} - \tilde{\boldsymbol{p}}\|_1 &= \sum_{i=1}^n |p_i - \tilde{p}_i| = \sum_{i=1}^n \left| \frac{\exp(-\eta x_i)}{Z} - \frac{\exp(-\eta \tilde{x}_i)}{\tilde{Z}} \right| \\
&= \sum_{i=1}^n \left| \frac{\exp(-\eta x_i)}{Z} - \left(1 - \frac{\tilde{Z} - Z}{\tilde{Z}}\right) \frac{\exp(-\eta \tilde{x}_i)}{Z} \right|
\end{aligned}$$

$$
\begin{aligned}
&\leq \sum_{i=1}^{n} \left| \frac{\exp(-\eta x_i)}{Z} - \frac{\exp(-\eta \tilde{x}_i)}{Z} \right| + \sum_{i=1}^{n} \left| \left( \frac{\tilde{Z} - Z}{\tilde{Z}} \right) \frac{\exp(-\eta \tilde{x}_i)}{Z} \right| \\
&\leq \frac{1}{Z} \sum_{i=1}^{n} |\exp(-\eta x_i) - \exp(-\eta \tilde{x}_i)| + \frac{|\tilde{Z} - Z|}{Z} \\
&= \frac{1}{Z} \sum_{i=1}^{n} (\exp(-\eta x_i) - \exp(-\eta \tilde{x}_i)) + \frac{Z - \tilde{Z}}{Z}.
\end{aligned}
$$

As the second term is smaller than the first term, we focus on bounding the first term. We have

$$
\begin{aligned}
&\sum_{i=1}^{n} (\exp(-\eta x_i) - \exp(-\eta \tilde{x}_i)) = \exp(-\eta x_i) - \exp(-\eta(x_i + \delta)) = \exp(-\eta x_i)\,(1 - \exp(-\eta \delta)) \\
&\leq \exp(-\eta x_i) \cdot \eta \delta. && \text{(by } 1 - \exp(-x) \leq x.\text{)}
\end{aligned}
$$

Hence, we have $\|\boldsymbol{p} - \tilde{\boldsymbol{p}}\|_1 = O(\eta \|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|_1)$. □

# G Additional Tools

## G.1 Nonexpansiveness of Strongly Convex Gradient

In this section, we prove Lemma 12, which is restated below for convenience.

**Lemma 12.** *The gradient step of a $L$-smooth function $g$ that is $\sigma$-strongly convex over its convex feasible region $K$ with step size $\eta \leq 1/L$ is $(1 - \eta\sigma)$-expansive with respect to the Euclidean norm.*

We note that the case of $K = \mathbb{R}^n$ is stated and proven in [Hardt et al., 2016, Lemma 3.7.3] and the proof also extends to a convex feasible region, as we show below.

We start with the following simple fact:

**Proposition 43.** *Let $g : K \to \mathbb{R}$ be a $\sigma$-strongly convex function over $K$ that is twice continuously differentiable. Then, $g - \frac{\sigma}{2}\|\cdot\|_2^2$ is convex over $K$.*

*Proof.* For $\boldsymbol{x}, \boldsymbol{y} \in K$,

$$
\begin{aligned}
&g(\boldsymbol{y}) - g(\boldsymbol{x}) - \frac{\sigma}{2}(\|\boldsymbol{y}\|_2^2 - \|\boldsymbol{x}\|_2^2) \\
&= \langle \nabla g(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{1}{2}(\boldsymbol{y} - \boldsymbol{x})^\top H(\boldsymbol{\xi})(\boldsymbol{y} - \boldsymbol{x}) - \frac{\sigma}{2}(\|\boldsymbol{y}\|_2^2 - \|\boldsymbol{x}\|_2^2) && (\boldsymbol{\xi} \in [\boldsymbol{x}, \boldsymbol{y}]) \\
&\geq \langle \nabla g(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{\sigma}{2}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2 - \frac{\sigma}{2}(\|\boldsymbol{y}\|_2^2 - \|\boldsymbol{x}\|_2^2) && (H|_K \succeq \sigma I|_K) \\
&= \langle \nabla g(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle - \sigma \langle \boldsymbol{y}, \boldsymbol{x} \rangle + \sigma \|\boldsymbol{x}\|_2^2 \\
&= \langle \nabla g(\boldsymbol{x}) - \sigma \boldsymbol{x}, \boldsymbol{y} - \boldsymbol{x} \rangle \\
&= \langle \nabla \Big[ g(\boldsymbol{x}) - \frac{\sigma}{2}\|\boldsymbol{x}\|_2^2 \Big], \boldsymbol{y} - \boldsymbol{x} \rangle.
\end{aligned}
$$

Hence $g - \frac{\sigma}{2}\|\cdot\|_2^2$ is convex over $K$. □

Recall that convexity and $L$-smoothness implies the *co-coercivity* of gradient:

$$
\langle \nabla g(\boldsymbol{y}) - \nabla g(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle \geq \frac{1}{L}\|\nabla g(\boldsymbol{y}) - \nabla g(\boldsymbol{x})\|_2^2.
$$

Then, we have the following:

**Proposition 44.** *Let $g : K \to \mathbb{R}$ be $\sigma$-strongly convex, $L$-smooth, and twice continuously differentiable. For any $\boldsymbol{x} \neq \boldsymbol{y} \in K$, we have*

$$
\langle \nabla g(\boldsymbol{y}) - \nabla g(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle \geq \frac{1}{\sigma + L}\|\nabla g(\boldsymbol{y}) - \nabla g(\boldsymbol{x})\|_2^2 + \frac{\sigma L}{\sigma + L}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2
$$

*Proof.* $g - \frac{\sigma}{2}\|\cdot\|_2^2$ is convex and remains $(L-\sigma)$-smooth so co-coercivity implies

$$\begin{aligned}
&\langle \nabla g(\boldsymbol{y}) - \nabla g(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle - \sigma\langle \boldsymbol{y} - \boldsymbol{x}, \boldsymbol{y} - \boldsymbol{x}\rangle \\
&\geq \frac{1}{L-\sigma}\|\nabla g(\boldsymbol{y}) - \nabla g(\boldsymbol{x}) - \sigma(\boldsymbol{y} - \boldsymbol{x})\|_2^2 \\
&= \frac{1}{L-\sigma}\|\nabla g(\boldsymbol{y}) - \nabla g(\boldsymbol{x})\|_2^2 - 2\frac{\sigma}{L-\sigma}\langle \nabla g(\boldsymbol{y}) - \nabla g(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle + \frac{\sigma^2}{L-\sigma}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2.
\end{aligned}$$

Rearranging yields the desired inequality. □

We are now ready to prove Lemma 12.

*Proof of Lemma 12.* Let $G_{g,\eta}(\boldsymbol{x}) \coloneqq \boldsymbol{x} - \eta\nabla g(\boldsymbol{x})$ be the gradient update . For $\boldsymbol{x}, \boldsymbol{y} \in K$,

$$\begin{aligned}
&\|G_{g,\eta}(\boldsymbol{x}) - G_{g,\eta}(\boldsymbol{y})\|_2^2 \\
&= \|\boldsymbol{x} - \boldsymbol{y}\|_2^2 - 2\eta\langle \nabla g(\boldsymbol{x}) - \nabla g(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y}\rangle + \eta^2\|\nabla g(\boldsymbol{x}) - \nabla g(\boldsymbol{y})\|_2^2 \\
&\leq \left(1 - \frac{2\eta\sigma L}{\sigma + L}\right)\|\boldsymbol{y} - \boldsymbol{x}\|_2^2 + \eta\left(\eta - \frac{2}{\sigma + L}\right)\|\nabla g(\boldsymbol{y}) - \nabla g(\boldsymbol{x})\|_2^2 && \text{(by Proposition 44)} \\
&\leq \left(1 - \frac{2\eta\sigma L}{\sigma + L}\right)\|\boldsymbol{y} - \boldsymbol{x}\|_2^2. && \text{(by } \eta \leq \tfrac{2}{\sigma+L}\text{)}
\end{aligned}$$

Note that $\sigma \leq L$ so that $2L/(\sigma+L) \geq 1$. Hence choosing $\eta \leq 1/L$ implies that $G_{g,\eta}$ is $(1-\eta\sigma)$-expansive. □

### G.2 Weyl's Inequality

**Theorem 45** (Weyl; Corollary 4.3.15 in Horn and Johnson [2012])**.** *Let $A, B$ be $n \times n$ Hermitian (real symmetric) matrices. Then*

$$\lambda_i(A) + \lambda_1(B) \leq \lambda_i(A + B) \leq \lambda_i(A) + \lambda_n(B), \qquad i \in [n].$$

In particular, if $A = \mathcal{L}(G, \boldsymbol{w})$ is the weighted Laplacian matrix of a graph $G$ with edge weights $\boldsymbol{w}$ and $B = \mathcal{L}(uv, \delta)$ is the rank 1 update matrix that increases the edge weight of $uv$ by $\delta > 0$ for some $uv \in E(G)$, then

$$\lambda_i(\mathcal{L}(G, \boldsymbol{w})) \leq \lambda_i(\mathcal{L}(G, \boldsymbol{w} + \delta\mathbb{1}_{uv})) \leq \lambda_i(\mathcal{L}(G, \boldsymbol{w})) + \lambda_n(B) \leq \lambda_i(\mathcal{L}(G, \boldsymbol{w})) + 2\delta.$$

### G.3 Finite Perturbation Bounds for Pointwise Lipschitz Continuity

Although the definition of a pointwise Lipschitz algorithm holds under a limit, we can still obtain finite perturbation bounds, which will be useful to show lower bounds.

**Theorem 46.** *Let $\boldsymbol{w}, \tilde{\boldsymbol{w}} \in \mathbb{R}^E_{\geq 0}$ be two weight vectors for a graph $G = (V, E)$. Let $w : [0,1] \to \mathbb{R}^E_{\geq 0}$ be a continuous path from $\boldsymbol{w}(0) = \boldsymbol{w}$ to $\boldsymbol{w}(1) = \tilde{\boldsymbol{w}}$ that is component-wise monotone,* i.e.*, $r \leq \tilde{r}$ implies $|w_i(r) - w_i(0)| \leq |w_i(\tilde{r}) - w_i(0)|$ for all $i$. If for all $r \in [0,1]$, $\mathcal{A}(G, \boldsymbol{w}(r))$ has pointwise Lipschitz constant of $c_r$ with respect to some metric $d$, then we have*

$$\frac{\mathrm{EMD}_d(\mathcal{A}(G, \boldsymbol{w}), \mathcal{A}(G, \tilde{\boldsymbol{w}}))}{\|\boldsymbol{w} - \tilde{\boldsymbol{w}}\|_1} \leq \sup_{r \in [0,1]} c_r =: c_{\sup}.$$

Recall that a subset of $S \subseteq \mathbb{R}^n$ is said to be *compact* if for any cover of $S$, say $\bigcup_{\iota \in I} C_\iota \supseteq S$, by open subsets $C_\iota \subseteq \mathbb{R}^n$, there exists a finite subcover $\bigcup_{i \in [n]} C_{\iota_i} \supseteq S$ for some $n < \infty$. Recall also the Heine-Borel theorem [Rudin, 1964, Theorem 2.41] from elementary analysis that states a subset of a finite-dimensional vector space is compact if and only if it is closed and bounded.

The intuition of Theorem 46 is that there is a local neighborhood at each point along the path from $\boldsymbol{w} \to \tilde{\boldsymbol{w}}$ where the pointwise Lipschitz constant (approximately) holds. Then by taking small steps forward on the path, we can translate the pointwise Lipschitz bound to a bound for $\boldsymbol{w}, \tilde{\boldsymbol{w}}$. However, in order to ensure a finite number of steps suffice, we crucially rely on compactness and continuity.

*Proof of Theorem 46.* Fix $\varepsilon > 0$. By the definition of a limit superior within the definition of the pointwise Lipschitz constant (Definition 1), for each $\boldsymbol{w}(r), r \in [0,1]$, there is some open ball $B_r \ni \boldsymbol{w}(r)$ such that for every $\boldsymbol{w}(r) \neq \boldsymbol{w} \in B_r$,

$$\frac{\mathrm{EMD}_d(\mathcal{A}(G, \boldsymbol{w}(r)), \mathcal{A}(G, \boldsymbol{w}))}{\|\boldsymbol{w}(r) - \boldsymbol{w}\|_1} \leq c_r + \varepsilon.$$

As the continuous pre-image of open sets is open [Rudin, 1964, Theorem 4.8], $w^{-1}(B_r)$ contains an open interval about $r$. Define $I_r \subseteq w^{-1}(B_r)$ to be such an open interval. Now, $\bigcup_{r\in[0,1]} I_r \supseteq [0,1]$ is an open cover of $[0,1]$, a compact (closed and bounded) set. By the definition of compactness, we can extract a finite subcover, say indexed by the points $0 = r_0 < r_1 < \cdots < r_N = 1$ for some $N < \infty$. By deleting and then adding points if necessary, we may assume without loss of generality that $r_{i+1} \in I_{r_i}$ for all $i = 0, 1, \ldots, N-1$.

It follows that

$$\begin{aligned}
&\mathrm{EMD}_d(\mathcal{A}(G, \boldsymbol{w}(0)), \mathcal{A}(G, \boldsymbol{w}(1))) \\
&\leq \sum_{i=1}^{N} \mathrm{EMD}_d(\mathcal{A}(G, \boldsymbol{w}(r_i)), \mathcal{A}(G, \boldsymbol{w}(r_{i-1}))) \\
&\leq (c_{\sup} + \varepsilon) \sum_{i=1}^{N} \|\boldsymbol{w}(r_i) - \boldsymbol{w}(r_{i-1})\|_1 \\
&= (c_{\sup} + \varepsilon) \|\boldsymbol{w}(0) - \boldsymbol{w}(1)\|_1.
\end{aligned}$$

Note that the last equality crucially uses the fact that our path is monotone. By the arbitrary choice of $\varepsilon > 0$, we must have

$$\mathrm{EMD}_d(\mathcal{A}(G, \boldsymbol{w}), \mathcal{A}(G, \tilde{\boldsymbol{w}})) \leq c_{\sup} \|\boldsymbol{w} - \tilde{\boldsymbol{w}}\|_1,$$

concluding the proof of the theorem. □

# H Naïve Algorithm for Minimum Vertex $S$-$T$ Cut

In this section, we describe a simple warmup for the minimum $S$-$T$ cut algorithm using unweighted regularization techniques from prior works Kumabe and Yoshida [2023] that yields sub-optimal guarantees. In particular, we prove the following theorem.

**Theorem 47.** *Let $\Lambda > 0$. There is a randomized polynomial-time approximation algorithm for undirected minimum vertex $S$-$T$ cut that outputs a vertex subset $A \subseteq V$ with the following guarantees.*

*(a) $A$ is a $(1, O(\Lambda n))$-approximate $S$-$T$ cut in expectation.*

*(b) The algorithm has a pointwise Lipschitz constant of $O\left(\frac{n}{\Lambda}\right)$.*

In comparison with Theorem 13, we must set the regularization parameter $\Lambda = \lambda_2/n$ to match the additive error of $O(\lambda_2)$. However, this yields a Lipschitz constant of $n^2/\lambda_2$, which is strictly worse than the guarantees of Theorem 13.

One technical detail is that in order to correctly set the parameter $\Lambda = \lambda_2/n$, we must estimate the weight-dependent parameter $\lambda_2$ in a stable manner. While Kumabe and Yoshida [2023] demonstrate how to achieve this via a sampling technique, another advantage of our algorithm obtained through PGTA is avoiding the need to smoothly estimate weight-dependent parameters to obtain a fractional Lipschitz solution.

## H.1 Obtaining a Fractional Lipschitz Solution

Let

$$\mathrm{cut}_{G,w}(y) \coloneqq \sum_{uv \in E} w_{uv} |y_u - y_v|$$

denote the fractional cut objective. Recall the LP relaxation LP 2 of minimum $S$-$T$ cut used in Appendix D.1. We restate a simplified version below.

$$\begin{array}{lll} \text{minimize} & \displaystyle\sum_{uv\in E} w_{uv}|y_u - y_v| & \\ \text{subject to} & y_{s_0} = 0 & \\ & y_{t_0} = 1 & \\ & y_s = y_{s_0} & \forall s \in S \\ & y_t = y_{t_0} & \forall t \in T \\ & y_v \in [0,1] & \forall v \in V \end{array} \tag{7}$$

We show the following:

**Lemma 48.** *There is an algorithm that outputs a feasible solution to LP 7 such that* $\mathrm{cut}_G(x) \le \mathrm{OPT}_{\mathrm{LP}} + \frac{\Lambda n}{2}$ *with Lipschitz constant* $O(n/\Lambda)$.

Following the technique of Kumabe and Yoshida [2023], we choose an unweighted $\ell_2$-regularizer with regularization parameter $\Lambda > 0$.

$$\begin{array}{lll} \text{minimize} & \mathrm{cut}_G(y) + \dfrac{\Lambda}{2}\|y\|_2^2 & \\ \text{subject to} & y_{s_0} = 0 & \\ & y_{t_0} = 1 & \\ & y_s = y_{s_0} & \forall s \in S \\ & y_t = y_{t_0} & \forall t \in T \\ & y_v \in [0,1] & \forall v \in V \end{array} \tag{8}$$

This is $(\Lambda/n)$-strongly convex with respect to the $\ell_1$-norm. The proof of Lemma 48 follows from the next two lemmas.

**Lemma 49.** *Let* $y \in \mathbb{R}^V$ *be the minimizer of LP 8. Then, we have* $\mathrm{cut}_G(y) \le \mathrm{OPT}_{\mathrm{LP}} + \frac{\Lambda}{2}n$.

*Proof.* Let $y^\star \in \mathbb{R}^V$ be the minimizer of LP 7. Then, we have

$$\mathrm{cut}_G(x) + \frac{\Lambda}{2}\|y\|_2^2 \le \mathrm{cut}_G(y^\star) + \frac{\Lambda}{2}\|y^*\|_2^2 \le \mathrm{cut}_G(y^\star) + \frac{\Lambda}{2}n.$$

□

As mentioned in Section 4, Kumabe and Yoshida [2023] incur an additive error proportional to the maximum range of the regularizer in order to obtain stable fractional solutions. Moreover, any $\Lambda$-strongly convex regularizer over $[0,1]^n$ has range $\Omega(\Lambda n)$. Thus, this seems like an inherent limitation of the Kumabe and Yoshida [2023] analysis rather than the choice of regularizer. As we see in Appendix D, PGTA allows us to sidestep this limitation by unlocking the use of weighted regularizers. On the other hand, as we demonstrate below, the Kumabe and Yoshida [2023] analysis requires unweighted regularizers.

**Lemma 50.** *The Lipschitz constant of the regularized LP is* $O(n/\Lambda)$.

*Proof.* Let $w \in \mathbb{R}^E$ be a weight vector and $w' \in \mathbb{R}^E$ be the vector obtained from $w$ by increasing the $\hat{e}$-th coordinate by $\delta$, where $\hat{e} = ab \in E$. By Lemma 2, it suffices to analyze this case in order to bound the Lipschitz constant.

Let $h_w(y) = h_{w,G}(y) \coloneqq \mathrm{cut}_{G,w}(y) + \frac{\Lambda}{2}\|y\|_2^2$ be the objective function of (8). Let $y$ and $y'$ be the minimizers of $h_w$ and $h_{w'}$, respectively. Let $D \coloneqq h_w(y') - h_w(y) + h_{y'}(x) - h_{w'}(y')$. As the regularization terms cancel out in $D$, we have

$$\begin{aligned} D &= \sum_{uv\in E} w_{uv}|y'_u - y'_v| - \sum_{uv\in E} w_{uv}|y_u - y_v| + \sum_{uv\in E} w'_{uv}|y_u - y_v| - \sum_{uv\in E} w'_{uv}|y'_u - y'_v| \\ &= (w'_{\hat{e}} - w_{\hat{e}})(|y_a - y_b| - |y'_a - y'_b|) \\ &= \delta(|y_a - y_b| - |y'_a - y'_b|). \end{aligned}$$

By the $(\Lambda/n)$-strong convexity of $x \mapsto \frac{\Lambda}{2}\|\cdot\|_2^2$ with respect to the $\ell_1$ norm and the optimality of $y, y'$ with respect to $h_w, h_{w'}$, respectively, we have

$$h_w(y') - h_w(y) \geq \frac{\Lambda}{n}\|y' - y\|_1^2.$$
$$h_{w'}(y) - h_{w'}(y') \geq \frac{\Lambda}{n}\|y' - y\|_1^2.$$

Summing them up, we get

$$\frac{2\Lambda}{n}\|y' - y\|_1^2 \leq D = \delta(|y_a - y_b| - |y'_a - y'_b|).$$

An application of the reverse triangle inequality yields

$$\frac{2\Lambda}{n}\|y' - y\|_1^2 \leq \delta(|y_a - y'_a| + |y_b - y'_b|) \leq \delta\|y - y'\|_1$$
$$\|y' - y\|_1 \leq \frac{\delta n}{2\Lambda}.$$

□

We remark that in the proof of Lemma 50, the regularization terms must cancel out in the expression $D$. The analysis of Kumabe and Yoshida [2023] does not extend beyond this case. However, PGTA is able to handle the case of a weighted regularizer where this cancellation does not happen, by reducing to the case of an unweighted regularizer at each iteration of PGM.

### H.2 Threshold Rounding

The rounding algorithm is the classic threshold rounding (Lemma 22), where we draw a uniform random threshold $\tau \sim [0, 1]$ and output the vertices with fractional solution $S_\tau = \{v \in V : y_v \leq \tau\}$. Applied to our setting, it preserves the fractional objective function in expectation as well as the fractional Lipschitz constant. Since we study threshold rounding in further detail in Appendix D and obtain stronger results, we omit a formal analysis of the rounding step here.

This concludes the proof of Theorem 47.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We provide complete proofs of the claims.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We explicitly discuss limitations in the conclusion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide all assumptions and proofs for each theoretical result.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [N/A]

   Justification:

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [N/A]

Justification:

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [N/A]

   Justification:

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [N/A]

   Justification:

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [N/A]

   Justification:

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our work confirms with the code of ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: Our contribution is mainly theoretical.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method development in this research does not involve LLMs.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.