Submodular Hypergraph Partitioning: Metric Relaxations and Fast Algorithms via an Improved Cut-Matching Game

Antares Chen
antaresc@uchicago.edu
University of Chicago

Lorenzo Orecchia
orecchia@uchicago.edu
University of Chicago

Erasmo Tani
etani@uchicago.edu
University of Chicago

September 14, 2025

Abstract

Despite there being significant work on developing spectral [16, 39, 38], and metric embedding [48] based approximation algorithms for hypergraph generalizations of conductance, little is known regarding the approximability of hypergraph partitioning objectives beyond this.

This work proposes algorithms for a general model of hypergraph partitioning that unifies both undirected and directed versions of many well-studied partitioning objectives. The first contribution of this paper introduces *polymatroidal cut functions*, a large class of cut functions amenable to approximation algorithms via metric embeddings and routing multicommodity flows. We demonstrate an $O(\sqrt{\log n})$-approximation, where n is the number of vertices in the hypergraph, for these problems by rounding relaxations to metrics of negative-type.

The second contribution of this paper generalizes the cut-matching game framework of Khandekar *et al.* [35] to tackle polymatroidal cut functions. This yields the first almost-linear time $O(\log n)$-approximation algorithm for standard versions of undirected and directed hypergraph partitioning [38]. A technical consequence of our construction is that a cut-matching game which greatly relaxes the set of allowed actions for both players can be used to partition hypergraphs with negligible impact on the approximation ratio. We believe this to be of independent interest.

1 Introduction

In the past 20 years, increasing complexity in real-world data has necessitated generalizing graph models to represent higher order relations [3, 15]. Hypergraphs have served as the quintessential object to model relations involving multiple objects, and, as a consequence, the study of hypergraph algorithms has featured prominently in recent practical [12, 57] and theoretical developments [16, 17, 47, 48].

Hypergraph partitioning is a fundamental algorithmic problem in this broader landscape. From an unsupervised machine learning perspective, hypergraph partitioning enables detecting significant clusters in complex higher-order networks, such as social [57, 61] or biological networks [28, 37]. From a theoretical perspective, partitioning algorithms are a fundamental primitive to be exploited in the development of divide-and-conquer methods for other hypergraphs problems. In contrast to graphs, where there is only one way for a partition to intersect an edge, hypergraphs admit multiple definitions for the cost of cutting a hyperedge. This yields a broad spectrum partitioning objectives, and allows the end-user to better design partitioning problems that fit specific applications. While there has been significant work on developing spectral approximation algorithms for hypergraph generalizations of conductance [16, 39, 38], with few exceptions [48], little is known about the approximability of large classes of hypergraph partitioning objectives beyond this. Even less is known regarding fast algorithms for these tasks.

This paper proposes *a general model for partitioning hypergraphs.* The first half of this paper identifies a large class of hypergraph partitioning objectives, which we call *polymatroidal cut functions.* This class is a subset of submodular cut functions previously considered by [43, 44] and captures both undirected and directed versions of almost all partitioning measures found in prior theoretical work. We show that these objectives possess properties that imply they are particularly well-suited for *approximation methods based on metric embeddings and routing multi-commodity flows.* This leads us to polynomial time $O(\sqrt{\log n})$-approximations for these objectives based on rounding ℓ_2^2-metric relaxations. The second half of this paper constructs fast algorithms that achieve similar $O(\log n)$-approximations by *generalizing the cut-matching framework* of Khandekar *et al.* [35]. This produces the first almost-linear time polylogarithmic approximation for both undirected and directed hypergraph conductance.

Let us now define concepts relevant to hypergraph partitioning. A *hypergraph* is a tuple $G = (V, E)$, where $E \subseteq 2^V$ is a set of *hyperedges.* The *rank* of a hyperedge $h \in E$ is its cardinality $|h|$. If every $h \in E$ satisfies $|h| = k$, the hypergraph G is called *k-uniform.* Hence, an undirected graph is a 2-uniform hypergraph. A key step in formalizing hypergraph cut problems is to define a suitable *cut function* $\delta_h : 2^h \to [0, 1]$ on each hyperedge $h \in E$. This expresses the cost incurred when h is cut by $(A, \bar{A})$ into $A \cap h$ and $\bar{A} \cap h$. Central to our work are submodular cut functions, defined by [43, 44]:

Definition 1 (Submodular Cut Function)**.** A function $\delta_h : 2^h \to [0, 1]$ is a *submodular cut function* if it is submodular and satisfies $\delta_h(\varnothing) = \delta_h(h) = 0$.

The space of cut functions for an edge $\{i, j\}$ is spanned by the undirected edge cut function $\delta_{\{i,j\}}(S) = |\mathbf{1}_i^S - \mathbf{1}_j^S|$ and the directed edge cut function $\delta_{(i,j)}(S) = (\mathbf{1}_i^S - \mathbf{1}_j^S)_+$, giving a complete classification of cut functions for rank-2 hyperedges. Higher-rank hyperedges support broader choices of cut functions. The most common example is the *standard hypergraph cut function* [16, 47]: $\delta_h^{\mathsf{cut}}(S) \stackrel{\text{def}}{=} \min\{1, |S|, |h \setminus S|\}$, which takes value 1 if h is cut by S and 0 otherwise. If the hypergraph $G = (V, E)$ is associated with positive hyperedge weights $\mathbf{w} \in \mathbb{Z}_{>0}^E$, we can succintly write the cut function for the entire graph as $\delta_G(S) \stackrel{\text{def}}{=} \sum_{h \in E} w_h \cdot \delta_h(S)$.

With these definitions, we are interested in minimizing the *submodular hypergraph ratio-cut objective.*

Definition 2 (Submodular Hypergraph Ratio-Cut)**.** Given a hypergraph $G = (V, E)$ with positive hyperedge weights $\mathbf{w} \in \mathbb{Z}_{>0}^E$, non-negative vertex weights $\boldsymbol{\mu} \in \mathbb{Z}_{\geq 0}^V$, and a collection of submodular cut functions $\{\delta_h\}_{h \in E}$, the *ratio-cut objective* on G is defined for all $S \subseteq V$ as:

$$\Psi_G(S) \stackrel{\text{def}}{=} \frac{\delta_G(S)}{\min\{\mu(S), \mu(\bar{S})\}} = \frac{\sum_{h \in E} w_h \cdot \delta_h(S \cap h)}{\min\{\mu(S), \mu(\bar{S})\}} .$$

The *ratio-cut* of G is $\Psi_G^* \stackrel{\text{def}}{=} \min_{S \subset V} \Psi_G(S)$.

When specialized to graphs[1] ratio-cut objectives include both graph expansion ($\boldsymbol{\mu} = \mathbf{1}$), graph conductance ($\forall i \in V, \mu_i = \sum_{h \in E : i \in h} w_h$). It also captures their directed counterparts, by replacing the undirected cut function $\delta_{\{i,j\}}$ with its directed analogue $\delta_{(i,j)}$.

Previous Work on Submodular Hypergraph Partitioning The standard hypergraph partitioning problem was first studied in the contexts of parallel numerical algorithms [15] and scientific computing [25].

Since then, many high-quality approximation algorithms have been developed for approximating hypergraph expansion and conductance. Louis and Makarychev [48] gave the first randomized polynomial-time $O\big(\sqrt{\log n}\big)$-approximation algorithm for hypergraph expansion. This matches the best known approximation for the graph expansion [10], and uses the same relaxation to metrics of negative-type. Louis [47] and Chan *et al.* [16] developed a spectral approach towards approximation hypergraph conductance, achieving Cheeger-like guarantees. Their algorithms require solving semidefinite programs (SDP) to approximate the hypergraph spectral gap, returning a partition with conductance at most $O\big(\sqrt{\Psi_G^* \log \max_{h \in E} |h|}\big)$. Their results are known to be tight under the small-set expansion conjecture. In more recent work, Lau *et al.* used a similar approach to obtain a $O(\sqrt{\Psi_G^* \log(1/\Psi_G^*)})$ guarantee for directed graph conductance [39].

Much less is known about the approximability of hypergraph ratio-cut objectives for general submodular cut functions. Following the work of Yoshida [62], Li and Milenkovic [43, 44] proposed submodular cut functions. For submodular hypergraph conductance, rounding an SDP relaxation yields algorithms with conductance guarantees $O\big(\sqrt{\Psi_G^* \cdot \log |V|}\big)$ [62] and $O\big(\sqrt{\Psi_G^* \cdot \max_{h \in E} |h|}\big)$ [44]. Li and Milenkovic [44] further conjecture that improving the dependence on hypergraph rank is NP-hard. The work of Yoshida [62] and the survey by Veldt *et al.* [60] provide excellent discussions of submodular cut functions and their applications.

Previous Work on Cut-Matching Games Algorithms for the above settings require solving SDP relaxations over $|V| \times |V|$ symmetric matrices, with at least $|E|$ constraints. No faster algorithms are currently known beyond generic SDP solvers [32], which run in $\Omega\big(|V|^2 |E|\big)$ time. Even for the simplest form of standard hypergraph partitioning, the resulting SDP relaxation is a mixed packing-and-covering program for which no almost-linear time algorithm is currently known [31].

In order to develop fast approximation algorithms, one then considers primal-dual methods that solve the dual to metric relaxations for ratio-cut problems. This poses an immediate issue; the dual to these relaxations are given by multi-commodity flow problems whose demands are encoded by a $\boldsymbol{\mu}$-weighted complete graph. Such problems may require quadratic time to solve [8] due to the large number of demands. The cut-matching game [36] was originally a method of approximately reducing this multi-commodity flow to a polylogarithmic number of exact single-commodity flow computations. Each single-commodity flow problem would route a perfect matching in the graph, until the union of matchings approximates the desired demand graph.

Through the cut-matching game, Khandekar *et al.* [36] obtained a $O(\log^2 |V|)$-approximation to undirected graph expansion by exactly solving $O(\log^3 |V|)$ maximum flow problems. Later, Orecchia *et al.* [54] reduced both of these measures by $O(\log n)$. Subsequent works have vastly generalized the applicability of the cut-matching game: Louis [46] extends [36] to approximate directed graph expansion, Long and Saranurak extend [35] to approximate vertex and hypergraph expansion, and [49] demonstrate how to use the cut-matching game using approximate single-commodity flow solves. The cut-matching game has since become ubiquitous in designing fast deterministic algorithms for various static, and dynamic graph problems [13, 21, 22, 23].

Our Contributions In this paper, we achieve the following.

- We identify the class of polymatroidal cut functions, a subset of submodular cut functions amenable to metric and flow techniques. For this class, we develop notions of hypergraph flows and flow-embeddings of graphs into hypergraphs which certify lower bounds to the ratio-cut objective.

- Using metric embedding techniques, we give polynomial-time $O(\sqrt{\log n})$-approximation algorithms for the minimum ratio-cut problem over submodular hypergraphs with polymatroidal cut functions. Our

[1] In the rest of the paper, for any undirected (resp. directed) graphs G, unless otherwise specified, we assume that δ_G and Ψ_G are formed using the undirected (resp. directed) cut functions.

algorithms generalize the seminal result of Arora, Rao and Vazirani [8] for sparsest cut to the largest class of problems to date.

- We extend the cut-matching game framework to approximate minimum ratio-cuts on submodular hypergraphs with polymatroidal cut functions. Our approach is based on approximating a continuous non-convex formulation of the minimum ratio-cut via a family of "local" convex submodular minimization problems. We then use the cut-matching framework to "boost" solutions to problems in this family, producing an $O(\log n)$-approximation algorithm for minimum ratio-cut that only requires a polylogarithmic number of approximate submodular minimization solves. This amounts to the first almost-linear time polylogarithmic approximation for both undirected, and directed hypergraph conductance.

A technical byproduct of our cut-matching game is that it allows for a cut player to play arbitrary disjoint vertex sets, and requires only a single $O(1)$-approximate submodular minimization solve per round. This may impact the deployment of cut-matching games in practice by lowering the required precision for the current running-time bottleneck of algorithms based on this framework [52, 58] as previous applications required either a single $o_n(1)$-approximate solves, or a polylogarithmic number of $O(1)$-approximate solves.

Our cut-matching game additionally avoids combinatorial restrictions found in previous applications of the framework. For example, the submodular minimization problem present in approximating hypergraph conductance is equivalent to a single-commodity flow whose demand graph is neither a perfect matching in the undirected case, nor Eulerian in the directed case. This may simplify the analyses of down-stream applications of the cut-matching game such as those found in [13, 21, 22, 23, 45, 49].

Concurrent Work Subsequent to initial versions of this paper [6], Veldt [58] provided a generalization of the cut-matching framework to approximate minimum ratio-cut problems specified by cardinality-based symmetric submodular cut functions, a subset of polymatroidal cut functions. Their approach utilizes exact maximum flow solves to compute hypergraph cut preservers, a gadget which lower bounds the ratio-cut objective only when the cut function is cardinality-based [60, 59]. Unlike this work, [58] provides compelling empirical evaluations which should certainly be considered.

Independent of this work, Lau *et al.* [40] give an almost-linear time $O(\sqrt{\log n})$-approximation for standard undirected and directed hypergraph cut functions. Their algorithm solves the dual to a relaxation of hypergraph conductance given by adding ℓ_2^2-metric constraints to a minimum reweighted eigenvalue problem [39]. This yields a multi-commodity flow problem, which they demonstrate to be approximable using a sequence of single-commodity flows via an interesting generalization of Sherman's algorithmic chaining [55]. Their algorithm is not based on the cut-matching game, and require similar combinatorial restrictions when routing the single-commodity flow as found in previous works.

Paper Organization The paper is organized as follows:

- Section 2 provides a technical overview of the results, and proof techniques found in this paper.
- Section 3 outlines preliminary tools and notation.
- Section 4.1 introduces polymatroidal cut functions, and provides key examples of functions in this class.
- Sections 4.2 and 4.3 introduce hypergraph flow, and hypergraph flow embeddings. These serve as fundamental objects in constructing certificates of approximate optimality in the cut-matching game.
- Section 5 details our $O(\sqrt{\log n})$-approximation for minimum ratio-cut given polymatroidal cut functions, describing the ℓ_2^2-metric relaxation, and embedding results required to round an integral solution.
- Section 6 describes our non-convex approach to the minimum ratio-cut problem and its connection to the cut-matching game.
- Section 7 describes our cut-matching game framework and uses it to construct an $O(\log n)$-approximation algorithm to the minimum ratio-cut problem with polymatroidal cut functions.

The proof of several technical statements encountered throughout the paper are postponed to the appendix.

2 Our Results

2.1 The Metric Approach to Submodular Hypergraph Ratio-Cut

The best polynomial-time approximation algorithms for graph ratio-cut problems are obtained by combining metric relaxations, either to general metrics [42] or ℓ_2^2-metrics [10], with metric embedding results for rounding. They achieve respectively an $O(\log n)$ and a $O(\sqrt{\log n})$-approximation ratio. We study the applicability of this approach to the submodular hypergraph ratio-cut problem. At the outset, we remark that we should not hope to obtain polylogarithmic approximation for general submodular cut functions without further assumptions, as Svitkina and Fisher (Section 3 in [56]) exhibit a submodular cut function for which any $o(\sqrt{n})$-approximation requires exponentially many queries to a value oracle.

Given this negative result, it is necessary to restrict the problem to a smaller class of cut functions. At a high level, for a metric relaxation to work, the cut function must exhibit some kind of monotonicity, so that its value is preserved under low-distortion metric embeddings. However, a cut function cannot be monotone without being equal to 0, so the monotone structure must appear in some other way. To solve this issue, we introduce the class of *polymatroidal cut functions*, which are the infimal symmetrization of monotone non-decreasing submodular functions (see below). We believe that this new class captures most cut functions that are amenable to approximation via metric relaxation. As partial evidence to this statement, we show in Section 4 that the fractional dual objects associated with polymatroidal cut functions are closely related to polymatroidal flows, a well-studied generalization of network flows with many of the same favorable flow-cut gap properties [19].

Polymatroidal Cut Functions and Metric Embeddings Our first contribution is to define *polymatroidal cut functions*, a subclass of submodular cut functions given by the infimal convolutions of non-decreasing submodular functions.

Definition 3 (Polymatroidal Cut Function)**.** A cut function $\delta_h : 2^h \to \mathbb{R}_{\geq 0}$ is *polymatroidal* if, for all $S \subseteq h$, it can be expressed as

$$\delta_h(S) = \min\left\{F_h^-(S),\ F_h^+(h \setminus S)\right\},$$

where the associated functions $F_h^-, F_h^+ : 2^h \to \mathbb{R}$ are non-decreasing submodular functions such that $F_h^-(\varnothing) = F_h^+(\varnothing) = 0$. When the associated functions F_h^- and F_h^+ are identical, we refer to the cut function δ_h as *symmetric polymatroidal.*

Note that δ_h penalizes a cut $(A, \overline{A})$ intersecting h in a directed manner, with F^- penalizing $A \cap h$ and F^+ penalizing $\overline{A} \cap h$. For a symmetric polymatroidal cut function δ_h, such penalty is the same, so that $\delta_h(A \cap h) = \delta_h(\overline{A} \cap h)$, generalizing the cut function of undirected graphs and hypergraphs. We show in Section 4 that Definition 3 captures most of the cut functions proposed in previous work, including all directed and undirected standard hypergraph cut functions. Moreover, our setup also captures possible combinations of all these types of cut functions into a single framework. We highlight two additional examples that are relevant in applications:

- F_h is a cardinality-based non-decreasing submodular function [59], i.e., a non-decreasing concave function of the cardinality. This is useful when we wish to modify the star-expansion cut function by diminishing the penalty on balanced partitions of h, e.g., by taking $\delta_h(S) = \min\left\{|S|^p, |h \setminus S|^p\right\}$ for $p \in (0, 1)$.

- F_h is the entropy of a subset of random variables associated with vertices of h. When the hypergraph G is the factor graph of an undirected graphical model, this yields a variant of the minimum information partition [30, 50].

In Section 5, we investigate the polynomial-time approximability of ratio-cut problems over submodular hypergraphs with polymatroidal cut functions. In particular, we prove the following result.

Theorem 1. *There exists a randomized polynomial-time $O\big(\sqrt{\log|V|}\big)$-approximation algorithm for solving the minimum ratio-cut problem on weighted submodular hypergraphs equipped with polymatroidal cut functions.*

This result extends the $O\big(\sqrt{\log n}\big)$-approximation of Louis and Makarychev [48] for standard hypergraph sparsest cut to all ratio-cut problems for the more general class of polymatroidal functions. Our algorithm for Theorem 1 solves a novel ℓ_2^2-metric relaxation of the ratio-cut problem for weighted submodular hypergraphs (see the program `RC-Directed-Semimetric` in Section 5). This relaxation crucially exploits the form of the Lovász extension of polymatroidal cut functions. To the best of our knowledge, our results constitutes the broadest known generalization of the seminal result of Arora, Rao and Vazirani for sparsest cut, capturing a wealth of partitioning objectives, including directed and undirected graph ratio-cut, vertex-based ratio cut and hypergraph ratio-cut within a single simple analysis. It also improves on the best known approximation ratio for many other objectives in this class, such as cardinality-based cut functions [59].

Polymatroidal Cut Functions and Hypergraph Flows In the second part of the paper, we explore the efficient solution of the minimum submodular hypergraph ratio cut problem via flow methods based on the cut-matching game. We start by developing the notions of flow-cut duality and flow embeddings over hypergraphs. For this purpose, in Section 4.2, we define a novel notion of *hypergraph flows* for weighted submodular hypergraphs equipped with polymatroidal cut functions. Specifically, for a weighted submodular hypergraph $G = (V, E, \mathbf{w}, \boldsymbol{\mu})$, we show that the subgradients of polymatroidal cut functions behave like network flows over the *factor graph*, a graph analogue of the input submodular hypergraph that includes an auxiliary vertex for each hyperedge $h \in E$ and an edge $\{i, h\}$ for each $i \in h$. For each hyperedge $h \in E$, the monotone functions F_h^- and F_h^+ define capacity constraints on the flows entering and leaving the corresponding auxiliary node. In this process, we highlight new connections between submodular hypergraphs and the polymatroidal flows described by Lawler and Martell [41] and extended to the multi-commodity flow setting by Chekuri *et al.* [19]. The main algorithmic result in this part of the paper is a *flow decomposition result for hypergraph flows* (Theorem 6) which allows us to construct flow embeddings of directed and undirected graphs into hypergraphs while lower bounding the size of the hypergraph cut with the size of the corresponding graph cut. This will become a fundamental building block for our cut-matching game construction.

2.2 A Non-Convex Optimization Approach To Minimum Ratio-Cut

To generalize the cut-matching game framework to the setting of submodular hypergraphs with polymatroidal cut functions, we devise a novel optimization approach based on a continuous non-convex formulation (`RC-NonCvx`) of the submodular hypergraph ratio-cut problem. Given a weighted submodular hypergraph $G = (V, E, \mathbf{w}, \boldsymbol{\mu})$, and cut functions $\{\delta_h\}_{h\in E}$ with Lovász extensions $\{\bar{\delta}_h\}_{h\in E}$, we define (`RC-NonCvx`) as the following non-convex optimization problem over $\mathbb{R}^V$:

$$\bar{\Psi}_G(\mathbf{x}) \overset{\text{def}}{=} \frac{\sum_{h\in E} w_h \bar{\delta}_h(\mathbf{x})}{\min_u \|\mathbf{x} - u\mathbf{1}\|_{\boldsymbol{\mu},1}}$$

$$\bar{\Psi}_G^* \overset{\text{def}}{=} \min_{\mathbf{x}\in\mathbb{R}^V} \bar{\Psi}_G(\mathbf{x}) \qquad (\texttt{RC-NonCvx})$$

The fact that solving (`RC-NonCvx`) is equivalent to solving the minimum ratio-cut problem (expressed in the following lemma) is a direct consequence of the submodularity of the cut functions. It is proved in full in Appendix A.

Lemma 1. *Given a weighted hypergraph $G^* = (V, E, \boldsymbol{\mu}, \mathbf{w})$, we have $\Psi_G^* = \bar{\Psi}_G^*$. Furthermore, there exists an algorithm that, given any $\mathbf{x} \in \mathbb{R}^V$, recovers a cut $S \subseteq V$ satisfying*

$$\Psi_G(S) \le \bar{\Psi}_G(\mathbf{x})$$

in time $O\big(|V|\log|V| + \mathsf{sp}(G)\big)$, where the sparsity of G is defined as $\mathsf{sp}(G) \overset{\text{def}}{=} \sum_{h\in E}|h|$.

The main idea in our approach is to address the non-convexity of the $\bar{\Psi}_G$ objective by replacing the non-concave denominator with a linear lower bound. In particular, we can exploit the dual characterization of the ℓ_1-norm in the lower bound to write:

$$\min_{u\in\mathbb{R}} \|\mathbf{x} - u\mathbf{1}\|_{1,\boldsymbol{\mu}} = \min_{u\in\mathbb{R}} \max_{\|\mathbf{y}\|_\infty \le 1} \langle \mathbf{y}, \mathbf{x} - u\mathbf{1}\rangle_{\boldsymbol{\mu}} = \max_{\substack{\|\mathbf{y}\|_\infty \le 1 \\ \langle \mathbf{y}, \mathbf{1}\rangle_{\boldsymbol{\mu}} = 0}} \langle \mathbf{y}, \mathbf{x}\rangle_{\boldsymbol{\mu}} \tag{2}$$

This calculation directly leads us to define a novel *localized, convex formulation* of the (RC-NonCvx) problem for each vector $\mathbf{s} \in \mathbb{R}^V$ with $\|\mathbf{s}\|_\infty \le 1$ and $\langle \mathbf{s}, \mathbf{1}\rangle = 0$, which we call a *seed*:

$$\begin{aligned} \bar{\Psi}_{G,\mathbf{s}}(\mathbf{x}) &\overset{\text{def}}{=} \frac{\sum_{h\in E} w_h \bar{\delta}_h(\mathbf{x})}{\max\{0, \langle \mathbf{s}, \mathbf{x}\rangle_{\boldsymbol{\mu}}\}} \\ \bar{\Psi}^*_{G,\mathbf{s}} &\overset{\text{def}}{=} \min_{\mathbf{x}\in\mathbb{R}^V} \bar{\Psi}_{G,\mathbf{s}}(\mathbf{x}) \end{aligned} \tag{Local-RC}$$

Intuitively, the program (Local-RC) seeks a distribution over cuts $\mathbf{x}$ with small expected cut size and high correlation with the seed $\mathbf{s}$. In Section 6, we investigate the properties of this formulation, including its equivalence with an integral cut problem, the ratio-cut improvement problem (Definition 10 in Section 6), which generalizes the cut improvement problem of Andersen and Lang [7] to submodular hypergraphs. For polymatroidal cut functions, we show that the natural notion of a dual solution for the convex problem Local-RC is exactly a hypergraph flow with demand vector $\mathrm{diag}(\boldsymbol{\mu})\mathbf{s}$. By applying our flow decomposition result, we can turn such a hypergraph flow into a *dual graph certificate*, a graph D such that $\Psi^*_{G,\mathbf{s}} \cdot \delta_D(S) \le \delta_G(S)$ for all $S \subseteq V$. To construct these objects, we prove the following algorithmic result:

Theorem 2 (Informal. See Theorems 9 and 10)**.** *An approximate primal-dual solution to* (Local-RC) *can be computed by solving $O(\log|V|)$-many decomposable submodular minimization problems for general polymatroidal cut functions or $O(\log|V|)$-many $^1/_2$-approximate maximum flow problems over a graph of size $O(\mathsf{sp}(G))$.*

Finally, and crucially for our purposes, for all $\mathbf{x} \in \mathbb{R}^V$, we have $\bar{\Psi}_G(\mathbf{x}) \le \bar{\Psi}_{G,\mathbf{s}}(\mathbf{x})$ by Equation 2. Hence, any (not necessarily optimal) solution $\mathbf{x}$ to (Local-RC) for a seed $\mathbf{s}$ yields a solution to (RC-NonCvx) of value at most $\bar{\Psi}_{G,\mathbf{s}}(\mathbf{x})$ and, via Lemma 1, a cut S with $\Psi_G(S) \le \bar{\Psi}_{G,\mathbf{s}}(\mathbf{x})$. In other words, if we can somehow find a seed $\mathbf{s}$ such that $\bar{\Psi}^*_{G,\mathbf{s}} \le \alpha\Psi^*_G$, then solving the (Local-RC) will yield an α-approximation algorithm for Ψ^*_G.

This reduction may not seem very useful, as finding the required seed may be as hard as solving the original problem. However, we can now exploit the dual feedback obtained when solving (Local-RC) for a sequence of seeds $\mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_T)$ to effectively guide our search for a good seed and boost our approximation ratio. In particular, suppose we can adaptively construct a sequence of T seeds $\mathbf{S}$ such that the sum $H = \sum_{t=1}^T D_t$ of the corresponding dual demand graphs D_t has large ratio-cut, i.e., $\Psi^*_H \ge \beta$. Then, it is easy to show (see Theorem 11 and its proof) that we must have $\min_{t=1}^T \Psi^*_{G,\mathbf{s}_t} \le {}^T/_\beta \cdot \Psi^*_G$. Hence, solving (Local-RC) for one of the seeds in the sequence $\mathbf{S}$ must yield a ${}^T/_\beta$-approximation to the minimum ratio-cut problem. It turns out that the problem of constructing such sequence $\mathbf{S}$ is precisely the task addressed by the cut-matching framework of Khandekar *et al.* [36], which we generalize and strengthen in our next contribution.

2.3 An Improved Cut-Matching Game

We now describe our new definition of the cut-matching game. A key distinction between our setup and the previous lies in what constitutes admissible actions played by the cut and matching players.

Definition 4. Let V be a set of vertices with vertex weights $\boldsymbol{\mu} \in \mathbb{Z}^V_{\ge 0}$. A *cut action* is a pair (A, B) of non-empty disjoint subsets $A, B \subseteq V$ such that $\boldsymbol{\mu}(A) \le \boldsymbol{\mu}(B)$.

The cut player is no longer restricted to returning a partition of the vertex set, let alone a bisection. The set of valid actions for a matching player is similarly relaxed, as the edges of the response do not need to form a perfect matching. It is only required that enough of the available degree is routed across the cut action.

Definition 5. An *approximate matching response* to a cut action (A, B) is an weighted bipartite graph $D = \big(A \cup B, E_D, \mathbf{w}^D, \boldsymbol{\mu}\big)$ satisfying the following conditions.

1. Bounded degree: for all $i \in V$
$$\deg_D(i) \leq \begin{cases} \mu_i & i \in A \\ \frac{\boldsymbol{\mu}(A)}{\boldsymbol{\mu}(B)} \cdot \mu_i & i \in B \end{cases}.$$
2. Largeness: $\mathbf{w}^D\big(E(A, B)\big) \geq \frac{\boldsymbol{\mu}(A)}{2}$.

We may emphasize that D is a *directed approximate matching response*, if the edges of D are directed from A to B.

This definition is motivated by the fact that an approximate matching response naturally arises from the dual graph certificate obtained by approximately solving the (`Local-RC`) program above. The matching response will be undirected when we wish to approximate the minimum ratio-cut of a hypergraph with symmetric cut functions and directed in the general case of asymmetric cut functions. Let us now define the cut-matching game.

Definition 6. Let $n > 0$, and $\boldsymbol{\mu} \in \mathbb{Z}_{\geq 0}^V$ be a weighting over a set of vertices V where $|V| = n$. A $(n, \boldsymbol{\mu})$-*generalized cut-matching game* is a multi-round, two-player game between a cut player $\mathcal{C}$, and a matching player $\mathcal{M}$ that proceeds as follows. In the t-th round of the game, following definitions 4 and 5:

1. $\mathcal{C}$ plays a cut action (A_t, B_t),
2. $\mathcal{M}$ responds with an approximate matching response D_t to (A_t, B_t).

The *state graph* after t rounds is $H_t = \sum_{s=1}^{t} D_s$ the edge-wise sum of the matching response seen thus far.

For comparison, the original cut-matching game of [36] is captured by the above definition when one specifically considers graph expansion $\boldsymbol{\mu} = \mathbf{1}$, restricts cut actions to be exact bisections $(A, \bar{A})$ where $|A| = |\bar{A}|$, and requires exact matching responses: $\mathbf{w}^D\big(E(A, \bar{A})\big) = \boldsymbol{\mu}(A)$.

To approximate minimum ratio-cut using our cut-matching games, we seek a cut player $\mathcal{C}$ that outputs a sequence of cuts which, in as few cuts as possible, forces the matching player $\mathcal{M}$ to place edges that make the minimum ratio-cut objective for the state graph H_T large. We call a cut player *good* if it can achieve this.

Definition 7. A *cut strategy* is a randomized algorithm $\mathcal{A}_{\mathsf{cut}}$ that takes as input the current state graph H, and outputs a cut action (A, B). A cut strategy $\mathcal{A}_{\mathsf{cut}}$ is $\big(f(n), g(n)\big)$-*good* if a cut player $\mathcal{C}$ using $\mathcal{A}_{\mathsf{cut}}$ at every iteration, achieves:
$$\Psi_{H_{g(n)}} \geq f(n)$$
with constant probability, for any (potentially adaptive) sequence of approximate matching responses $D_1, \ldots, D_{g(n)}$.

Note that, when dealing with symmetric cut functions, H will be undirected, so that Ψ_H refers to the undirected ratio-cut objective. If the cut functions are asymmetric, then Ψ_H is naturally replaced with the directed ratio-cut objective.

These definitions reveal the mechanism by which the cut-matching framework adaptively constructs a sequence of seeds to (`Local-RC`): the seed we choose at every iteration of cut-matching game is just a $\boldsymbol{\mu}$-weighted indicator vector for the cut action of that round. A good strategy guarantees that, no matter what dual graph certificates we receive for each seed, at the end of the game at $T = g(n)$ iterations, we can guarantee H_T has ratio-cut objective at least $\beta = f(n)$. This shows that a good strategy yields an approximation ratio of $O(g(n)/f(n))$ for minimum ratio cut. In Section 7, we formalize this result as Theorem 11. Our main technical theorem (Theorem 12) regarding the new cut-matching game shows the existence of good strategies for the cut-player that yield $O(\log n)$-approximation algorithms for the minimum ratio-cut problem over submodular hypergraphs, for both the cases of symmetric and general polymatroidal cut functions. The proof of this theorem is based on a new geometric result about separated sets in vector embeddings, which we discuss below. Together with Theorem 2, our results on good cut-player strategies directly imply the following results for the solution of the minimum ratio-cut problem. Their short proof appear in Appendix A.5.

Corollary 1. *There exists an $O(\log n)$-approximation algorithm for the minimum ratio-cut problem over submodular hypergraphs with polymatroidal cut functions whose running time is dominated by the solution of a polylogarthmic number of decomposable submodular minimization problems over the hypergraph cut function.*

Corollary 2. *There exists an $O(\log n)$-approximation algorithm for the minimum ratio-cut problem over submodular hypergraphs equipped with the directed or standard hypergraph cut function whose running time is almost linear in the hypergraph sparsity.*

Separated Sets In our new cut-matching game, the cut player must pay considerable more care in choosing a cut action (A, B) than in the original version, as the approximate nature of the matching response means that the matching player may easily keep some small subset of vertices disconnected from the rest of the graph. Suppose for instance that the cut player restricted itself to play μ-bisections. Then, the matching player could select a small cut T, with $\mu(T) \leq 1/10 \cdot \mu(V)$ and never place any edge across the boundary of T. The only way the cut player can force progress on T is to depart from playing bisections and play a cut action (A, B) where A is close or equal to T.

Let us see how this intuition is formalized in our design of cut-player strategies. As in the analysis of Orecchia *et al.* [54], our cut strategy maintains at each iteration t a spectral embedding for the current state graph H_t. We make progress if we can force the matching player to connect vertices that are far away in this embedding. We show that we can achieve this, even against an approximate matching player, by choosing a cut action coming from the following notion of separated sets:

Definition 8 (σ-separated sets)**.** Given an embedding $\{\mathbf{v}_i\}_{i\in[n]}$ and a measure $\boldsymbol{\mu} \in \mathbb{R}^n_{\geq 0}$ we say two sets $S, T \subseteq [n]$ are *σ-separated* if:

$$\boldsymbol{\mu}(S) \cdot \boldsymbol{\mu}(T) \cdot \min_{\substack{i \in S \\ j \in T}} \|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq \sigma \cdot \sum_{i,j \in \binom{V}{2}} \mu_i \mu_j \cdot \|\mathbf{v}_i - \mathbf{v}_j\|^2.$$

This definition generalizes the idea of well-separated sets [10], which was applied only to large balanced sets, to possibly unbalanced sets, as long as S and T contribute a σ-fraction to the total variance of the embedding. In Section 8, we show how to efficiently find separated sets and prove the following result.

Theorem 3. *Given an embedding $\{\mathbf{v}_i\}_{i\in[n]} \subseteq \mathbb{R}^d$ and a measure $\boldsymbol{\mu} \in \mathbb{Z}^n_{\geq 0}$, with $\mu(V) = \mathsf{poly}(n)$, there exists a randomized algorithm running in time $\tilde{O}(nd)$ that outputs σ-separated sets S and T for $\sigma = \Omega(1/\log n)$ with high probability.*

3 Preliminaries

Hypergraphs The sparsity $\mathsf{sp}(G)$ of a hypergraph is defined as $\mathsf{sp}(G) \overset{\text{def}}{=} \sum_{h \in E} |h|$. Throughout the paper, we assume that the edge weights $\mathbf{w}$ and vertex weights $\boldsymbol{\mu}$ of our instance submodular hypergraph $G = (V, E, \mathbf{w}, \boldsymbol{\mu})$ are polynomials in $|V|$. In particular, we have $\boldsymbol{\mu}(V) \in \mathsf{poly}(|V|)$.

Linear Algebra Notation Throughout this paper, we adopt the convention of representing scalars x in unbolded lowercase, vectors $\mathbf{x}$ in boldface lowercase, and matrices $\mathbf{X}$ as boldface uppercase. Given vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, we will use $\mathbf{x} \circ \mathbf{y}$ to denote the Hadamard (entry-wise) product between $\mathbf{x}$ and $\mathbf{y}$. We will use $|\mathbf{x}|$ to denote the vector whose entries are the absolute value of the entries in $\mathbf{x}$. We let $\langle \mathbf{x}, \mathbf{y} \rangle$ denote the standard Euclidean inner product, $\|\cdot\|$ the Euclidean norm, and $\|\cdot\|_p$ the ℓ_p-norm.

We define an inner product between $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ weighted by a non-negative vector $\boldsymbol{\mu} \in \mathbb{R}^n_{\geq 0}$ as $\langle \mathbf{x}, \mathbf{y} \rangle_{\boldsymbol{\mu}} \overset{\text{def}}{=} \sum_{i=1}^n \mu_i \cdot x_i \cdot y_i$. The quantity $\|\mathbf{x}\|_{\boldsymbol{\mu}} \overset{\text{def}}{=} \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle_{\boldsymbol{\mu}}}$ will denote the norm induced by the inner product $\langle \cdot, \cdot \rangle_{\boldsymbol{\mu}}$, while for any $p > 0$, we use $\|\mathbf{x}\|_{\boldsymbol{\mu},p}$ to denote the ℓ_p-norm weighted by $\boldsymbol{\mu}$ $\|\mathbf{x}\|_{\boldsymbol{\mu},p} \overset{\text{def}}{=} \left(\sum_{i=1}^n \mu_i \cdot |x_i|^p\right)^{1/p}$. Note consequently that $\|\mathbf{x}\|_{\boldsymbol{\mu}} = \|\mathbf{x}\|_{\boldsymbol{\mu},2}$. For $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, we write $\mathbf{x} \perp \mathbf{y}$ to indicate the relation: $\langle \mathbf{x}, \mathbf{y} \rangle = 0$.

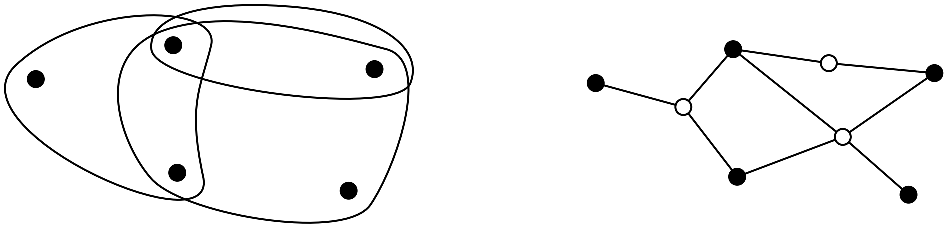

Figure 1: A hypergraph (left) and its factor graph (right). In the factor graph image, the variable vertices are shown in solid black, while the factor vertices are shown as empty circles.

Given any vector $\mathbf{x} \in \mathbb{R}^n$, we define the positive and negative parts of $\mathbf{x}$ as the vectors $\mathbf{x}_+, \mathbf{x}_- \in \mathbb{R}^n$ with $\mathbf{x}_+(i) \stackrel{\text{def}}{=} \max\{0, \mathbf{x}(i)\}$ and $\mathbf{x}_-(i) \stackrel{\text{def}}{=} \max\{0, -\mathbf{x}(i)\}$.

If $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n\times n}$, then $\langle \mathbf{X}, \mathbf{Y} \rangle = \sum_{i,j=1}^n X_{ij} Y_{ij}$ denotes the Frobenius inner product. We let $\mathcal{S}^{n\times n}$ be the set of n by n symmetric matrices. Given $\mathbf{X}, \mathbf{Y} \in \mathcal{S}^{n\times n}$ we denote by $\mathbf{X} \succeq 0$ the statement that $\mathbf{X}$ is positive semidefinite, and by $\mathbf{X} \preceq \mathbf{Y}$ the relation $0 \preceq \mathbf{Y} - \mathbf{X}$. We use $\operatorname{diag}(\mathbf{x}) \in \mathbb{R}^{n\times n}$ to denote the real matrix with the coordinates of $\mathbf{x}$ placed along its diagonal. We will reserve specifically the matrix $\mathbf{M} \stackrel{\text{def}}{=} \operatorname{diag}(\boldsymbol{\mu})$ for the non-negative vector $\boldsymbol{\mu} \in \mathbb{R}^n_{\geq 0}$.

Given a finite set V, a vector $\mathbf{x} \in \mathbb{R}^V$, and a subset $S \subseteq V$, we let $\mathbf{x}(S) = \sum_{v\in S} x_v$. We also denote by $\mathbf{x}_S \in \mathbb{R}^V$, the restriction of $\mathbf{x}$ to coordinates in S. In this way, the vector $\mathbf{1}_S$ denotes the $\{0,1\}$-indicator of set S.

Preliminaries on Graphs Given a graph $H = (V, E, \mathbf{w})$, denote its combinatorial Laplacian as $\mathbf{L}(H)$:

$$\mathbf{L}(H) \stackrel{\text{def}}{=} \mathbf{D}(H) - \mathbf{A}(H),$$

If H is directed, then we will associate with it a directed Laplacian $\mathbf{L}_+$ given by

$$\mathbf{L}_+(H) = \sum_{(i,j)\in E} w_{ij} \cdot \left(\mathbf{L}_{ij} + \mathbf{E}_{ii} - \mathbf{E}_{jj}\right)$$

where $\mathbf{L}_{ij}$ is the Laplacian of an undirected graph $\left(V, \left\{\{i,j\}\right\}\right)$, and $\mathbf{E}_{ii}$ is 1 on ii-th entry and 0 everywhere else. We will denote $\tilde{H}$ as the symmetrization of H, i.e. the weighted undirected graph formed by taking the edge-wise sum $\mathbf{A}(G) + \mathbf{A}(G)^\top$ of H and its reverse graph.

Factor Graphs A hypergraph $G = (V, E_G, \mathbf{w}, \boldsymbol{\mu})$ is naturally associated with its *factor graph*, an unweighted graph $\hat{G} = (V \cup E_G, \hat{E}_G)$ that is bipartite between the *variable* vertices representing vertices $i \in V$, and *factor* vertices representing hyperedges $h \in E_G$. For $i \in V$ and $h \in E_G$, the edge $\{i, h\}$ belongs to $\hat{E}_G$ if and only if $i \in h$, i.e.:

$$\hat{E}_G = \{\{i,h\} \in V \times E_G : i \in h\}.$$

Submodular Functions A set function $F : 2^V \to \mathbb{R}$ is submodular if for all $A, B \subseteq V$, we have:

$$F(A \cup B) \leq F(A) + F(B) - F(A \cap B).$$

From the above, one can easily see that the set of submodular functions is closed under taking conic combinations. Following standard conventions, we will always assume that $F(\varnothing) = 0$ unless specified otherwise.

Any submodular set function F can be associated with a convex body $\mathcal{B}(F)$, known as its base polytope:

$$\mathcal{B}(F) \overset{\text{def}}{=} \big\{\mathbf{x} \in \mathbb{R}^V \;:\; \langle \mathbf{x}, \mathbf{1}\rangle = F(V) \;\wedge\; \forall S \subseteq V,\, \langle \mathbf{x}, \mathbf{1}_S\rangle \leq F(S)\big\}. \tag{4}$$

When F is monotone non-decreasing and $F(\emptyset) = 0$, it is convenient to employ the positive submodular polyhedron:

$$\mathcal{P}_+(F) \overset{\text{def}}{=} \big\{\mathbf{x} \in \mathbb{R}^V_{\geq 0} \;:\; \forall S \subseteq V,\, \langle \mathbf{x}, \mathbf{1}_S\rangle \leq F(S)\big\}.$$

Crucially, a submodular set function F can be extended to its convex closure, known as its Lovász extension and denoted by $\bar{F} : \mathbb{R}^V \to \mathbb{R}$. There are a number of equivalent characterizations of the Lovász extension and its connection to the base polytope $\mathcal{B}$ and the positive polyhedron $\mathcal{P}_+$. Rather than listing them here, we will refer directly to the results in Chapter 3 and 4 of the excellent reference by Bach [11]. We make an exception for the following fact, which plays a crucial role in the analysis of our approximation algorithms based on metric embeddings. See Proposition 4.8 in [11] for a proof.

Fact 1. *Let $F : 2^V \to \mathbb{R}_{\geq 0}$ be a monotone non-decreasing submodular function with $F(\emptyset) = 0$. Its Lovász extension $\bar{F}$ is monotone, non-decreasing over $\mathbb{R}^V$.*

Online Linear Optimization and Regret Minimization Fix a convex set of *actions* $\mathcal{X} \subseteq \mathcal{S}^{n\times n}$. The setting of *online linear optimization* over symmetric matrices considers the following multi-round game: in each round t, a player plays an action $\mathbf{X}_t \in \mathcal{X}$, receives a feedback matrix $\mathbf{F}_t \in \mathcal{S}^{n\times n}$, and suffers a loss of $\langle \mathbf{F}_t, \mathbf{X}_t\rangle$. The goal of the player is to play actions which minimize total loss incurred. In general, one cannot hope for a strategy that is competitive against an arbitrary, adaptive adversary. Thus, one instead seeks to *minimize regret*: the difference, up to a time horizon $T > 0$, between total loss and the loss suffered by the best fixed action in hindsight.

We will require a regret minimization strategy to produce a sequence of $\mathbf{X}_t$ for this setting. Fortunately, many prior results [9, 5, 29, 34, 51, 53] have studied this, and a Matrix Multiplicative Weight Update algorithm works off-the-shelf.

Theorem 4 (Theorem 3.3.3 in [51])**.** *Fix $T > 0$, and a weight vector $\boldsymbol{\mu} \in \mathbb{Z}^n_{\geq 0}$. Let $\mathbf{F}_1, \ldots, \mathbf{F}_T$ be a sequence of feedback matrices such that there exists $\rho > 0$ where $\mathbf{F}_t \succeq -\frac{1}{\rho} \cdot \mathbf{M}$ for every $t > 0$. For any step size $0 < \eta < \rho$, define $\mathbf{X}_t$ to be given by the matrix multiplicative weight update*

$$\begin{aligned} \mathbf{W}_{t+1} &= \mathtt{MMWU}_{\eta,\boldsymbol{\mu}}(\mathbf{F}_1, \ldots, \mathbf{F}_t) \\ &\overset{\text{def}}{=} \mathbf{M}^{-1/2} \exp\left(-\eta \cdot \sum_{k=1}^{t} \mathbf{M}^{-1/2}\mathbf{F}_k\mathbf{M}^{-1/2}\right)\mathbf{M}^{-1/2} \\ \mathbf{X}_{t+1} &= \frac{\mathbf{W}_{t+1}}{\langle \mathbf{M}, \mathbf{W}_{t+1}\rangle} \end{aligned} \tag{matrix-mwu}$$

Then, the following inequalities hold for any $\mathbf{U} \in \Delta^{n\times n}_{\boldsymbol{\mu}}$. We have:

$$\frac{1}{T}\sum_{t=1}^{T}\langle \mathbf{F}_t, \mathbf{U}\rangle \geq \frac{1}{T}\sum_{t=1}^{T}\langle \mathbf{F}_t, \mathbf{X}_t\rangle - \frac{\eta}{T}\sum_{t=1}^{T} \big\|\mathbf{M}^{-1/2}\mathbf{F}_t\mathbf{M}^{-1/2}\big\|^2_{\mathsf{spec}} - \frac{\log n}{\eta T}\,. \tag{6}$$

Furthermore, if $\mathbf{F}_t \succeq -\frac{1}{\rho} \cdot \mathbf{M}$ for every $\rho > 0$ (i.e. $\mathbf{F}_t \succeq \mathbf{0}$), and there exist $\epsilon > 0$ such that $\mathbf{F}_t \preceq \epsilon \cdot \mathbf{M}$, then:

$$\frac{1}{T}\sum_{t=1}^{T} \big\langle \mathbf{F}_t, \mathbf{U}\big\rangle \geq \frac{1-\epsilon\eta}{T} \cdot \sum_{t=1}^{T} \big\langle \mathbf{F}_t, \mathbf{X}_t\big\rangle - \frac{\log n}{\eta T}\,. \tag{7}$$

We remark that equation (6) is the standard regret bound for the Matrix Multiplicative Weight Update algorithm, while (7) is the regret bound proven using local norm convergence.

4 Polymatroidal Cut Functions and Hypergraph Flows

In this section, we establish the main properties of the polymatroidal cut functions in Definition 3. In the second part of the section, we introduce the dual notion of hypergraph flows as natural lower bounds to polymatroidal cut functions and highlight their connection with previously studied polymatroidal flows [19]. Finally, we define the notion of flow embedding of a graph into a hypergraph.

4.1 Properties and Examples

We start by verifying that any polymatroidal cut function is submodular. A proof of the following fact can be found in Appendix A:

Fact 2. *Let V be an arbitrary finite set and $F, G : 2^V \to \mathbb{R}$ be monotone non-decreasing submodular functions. If $\delta : 2^V \to \mathbb{R}$ is the function defined by:*

$$\delta(S) \overset{\text{def}}{=} \min\{F(S), G(V \setminus S)\},$$

for all $S \subseteq V$, then δ is submodular.

Examples of Polymatroidal Cut Functions We now show that the class of polymatroidal cut functions encompasses many popular cut objectives. Our first example is the directed edge cut function in graphs. This function is defined on a directed edge (i, j) as:

$$\delta_{(i,j)}(S) = \begin{cases} 1 & \text{if } i \in S \text{ and } j \in \overline{S}, \\ 0 & \text{otherwise.} \end{cases}$$

It is easy to check that this function is polymatroidal, since it can be rewritten as:

$$\delta_{(i,j)}(S) = \min\{|S \cap \{i\}|, |(h \setminus S) \cap \{j\}|\}$$

for all $S \subseteq h$. Next, we consider the standard hyperedge cut function. Given a hyperedge $h \in E$, this function is defined as:

$$\delta_h^{\mathsf{cut}}(S) \overset{\text{def}}{=} \begin{cases} 0 & \text{if } S \cap h \in \{\emptyset, h\}, \\ 1 & \text{otherwise.} \end{cases}$$

Note that the undirected cut function in graphs is a special case of δ^{cut}. This function is symmetric polymatroidal, as we can let $F_h(S) = \min\{1, |h \cap S|\} = G_h(S)$ for any $h \in E$ to obtain:

$$\delta_h^{\mathsf{cut}}(S) = \min\{1, |h \cap S|, |h \cap \overline{S}|\},$$

which is 1 if and only if S intersects h non-trivially, and zero otherwise.

Moreover, the class of directed hypergraph cut functions recently introduced by Lau et al. (Section 6.2 of [40]) to define directed hypergraph expansion is also a sub-class of polymatroidal cut functions, as we now show. Given a hyperedge $h \in E$ and a partition of h into a tail set T_h and a headset H_h, we can capture the directed cut function of Lau et al., by letting $F_h(S) = \min\{1, |T_h \cap S|\}$ and $G_h(\overline{S}) = \min\{1, |H_h \cap \overline{S}|\}$.

Finally, Veldt *et al.* [59] consider a class of cut functions which are cardinality-based and submodular. In Appendix B.2 of their paper, they argue that every such cut function δ_h can be written as:

$$\delta_h(S) = g_h(\min\{|S \cap h|, |h \cap \overline{S}|\}),$$

for some concave and monotonically increasing function g_h. Since g_h is monotonically increasing, the above gives:

$$\delta_h(S) = \min\{g_h(|S \cap h|), g_h(|h \cap \overline{S}|)\},$$

and hence all functions in this class are also examples of polymatroidal cut functions (note that the composition of a monotone, concave function with the cardinality function is always monotone and submodular).

Lovász Extensions of Polymatroidal Cut Functions We next establish the form of the Lovász extension of a polymatroidal cut function. We will make heavy use of this form in building our metric relaxations and in rounding them via low-distortion metric embeddings in Section 5. The proof of the following fact is found in Appendix A:

Fact 3. *Given a polymatroidal cut function δ_h, we have, for all $\mathbf{x} \in \mathbb{R}^h$:*

$$\overline{\delta}_h(\mathbf{x}) = \min_{\nu \in \mathbb{R}} \bar{F}_h^-((\mathbf{x} - \nu \mathbf{1}_h)_+) + \bar{F}_h^+((\mathbf{x} - \nu \mathbf{1}_h)_-).$$

It is instructive to consider how the previous results are instantiated for the two canonical examples of the directed edge cut function and the standard hypergraph cut function. For the former on an arc (i, j), we have $\bar{F}^-(\mathbf{x}) = \mathbf{x}_i$ and $\bar{F}^+(\mathbf{x}) = \mathbf{x}_j$, so that:

$$\forall \mathbf{x} \in \mathbb{R}^{\{i,j\}},\ \overline{\delta}_{(i,j)}(\mathbf{x}) = \min_{\nu \in \mathbb{R}} (\mathbf{x}(i) - \nu)_+ + (\mathbf{x}(j) - \nu)_- = (\mathbf{x}(i) - \mathbf{x}(j))_+.$$

For the latter, on a hyperedge h:

$$\forall \mathbf{x} \in \mathbb{R}^h,\ \overline{\delta}_h^{\mathsf{cut}}(\mathbf{x}) = \min_{\nu \in \mathbb{R}} \|(\mathbf{x} - \nu \mathbf{1})_+\|_\infty + \|(\mathbf{x} - \nu \mathbf{1}\|)_-\|_\infty = .$$

In the case of a symmetric polymatroidal cut function, the characterization of Fact 3 can take a more compact form, as shown in Appendix A:

Fact 4. *Given a symmetric polymatroidal cut function δ_h with $F_h \overset{\text{def}}{=} F_h^+ = F_h^-$, we have, for all $\mathbf{x} \in \mathbb{R}^h$:*

$$\min_{\nu \in \mathbb{R}} \bar{F}_h(|\mathbf{x} - \nu \mathbf{1}_h|) \leq \overline{\delta}_h(\mathbf{x}) \leq 2 \cdot \min_{\nu \in \mathbb{R}} \bar{F}_h(|\mathbf{x} - \nu \mathbf{1}_h|).$$

4.2 Base Polytope, Hypergraph Flows and Polymatroidal Networks

It is natural to attempt to lower bound the cost of cuts in a submodular hypergraph $G = (V, E_G, \mathbf{w}, \boldsymbol{\mu})$ by considering collections of vectors $\mathbf{Y} = (\mathbf{y}_h \in \mathcal{B}(\delta_h))_{h \in E_G}$, where each vector $\mathbf{y}_h$ belongs to the base polytope of the polymatroidal cut function δ_h under consideration. In this section, we interpret these vectors as hypergraph flows over G and show that they take the form of polymatroidal flows [41, 19] over the factor graph $\hat{G}$.

Base Polytope of Polymatroidal Cut Function We start by characterizing the base polytope $\mathcal{B}(\delta_h)$ for a polymatroidal cut function δ_h. By the definition of base polytope in Equation 8, we have

$$\mathcal{B}(\delta_h) = \{\mathbf{y} \in \mathbb{R}^h : \langle \mathbf{y}, \mathbf{1}_h \rangle = 0 \wedge \forall S \subset h,\ \langle \mathbf{y}, \mathbf{1}_S \rangle \leq \min\{F_h^-(S), F_h^+(h \setminus S)\}\}.$$

The first constraint implies that for all $S \subset h$, we must have $\langle \mathbf{y}, \mathbf{1}_S \rangle = -\langle \mathbf{y}, \mathbf{1}_{h \setminus S} \rangle$. Hence, we can rewrite:

$$\begin{aligned} \mathcal{B}(\delta_h) &= \{\mathbf{y} \in \mathbb{R}^h \perp \mathbf{1}_h : \forall S \subset h,\ \langle \mathbf{y}, \mathbf{1}_S \rangle \leq F_h^-(S) \wedge -\langle \mathbf{y}, \mathbf{1}_{h \setminus S} \rangle \leq F_h^+(h \setminus S)\} \\ &= \{\mathbf{y} \in \mathbb{R}^h \perp \mathbf{1}_h : \forall S \subset h,\ -F_h^+(S) \leq \langle \mathbf{y}, \mathbf{1}_S \rangle \leq F_h^-(S)\} \\ &= \{\mathbf{y} \in \mathbb{R}^h \perp \mathbf{1}_h : (\mathbf{y})_+ \in \mathcal{P}_+(F_h^-) \wedge (\mathbf{y})_- \in \mathcal{P}_+(F_h^+)\}. \end{aligned} \tag{8}$$

We will interpret an element $\mathbf{y}$ of $\mathcal{B}(\delta_h)$ as a flow over the neighborhood of factor vertex h in the factor graph $\hat{G}$, with $\mathbf{y}_i > 0$ indicating flow going from i to h and $\mathbf{y}_i < 0$ flow going from h to i. The constraint $\mathbf{y} \perp \mathbf{1}_h$ simply represents flow conservation at h. With this setup, for each set $S \subset h$, the function F_h^- (resp. F_h^+) constrains the amount of flow that can be routed from the set S to h (resp. to the set S from h).

Example: Base Polytope for Standard Hypergraph Cut Function Consider the case of the standard hypergraph cut function $\delta_h^{\mathsf{cut}}(S)$ defined in the previous section. Then, it is easy to see that:

$$\mathcal{B}(\delta_h^{\mathsf{cut}}) = \{\mathbf{y} \in \mathbb{R}^h \perp \mathbf{1}_h \mid \|\mathbf{y}\|_1 \leq 2\}.$$

As discussed above, we can think of each entry $\mathbf{y}(i)$ of $\mathbf{y}$ for $i \in h$, as the amount of flow flowing into h from vertex i, and the constraint $\|\mathbf{y}\|_1 \leq 2$ enforces that the total amount of flow through the hyperedge h is no more than 1. This is equivalent to a vertex capacity constraint on the factor vertex corresponding to the hyperedge h in the factor graph $\hat{G}$.

Example: Base Polytope for Directed Hypergraph Cut Functions Consider the class of directed hypergraph cut functions defined by Lau *et al.* in Section 6.2 of [40]. Each of these cut functions δ_h has an associated head set $H_h \subseteq h$ and an associated tail set $T_h \subseteq h$ and $\delta_h(S)$ has value one if and only if the head set intersects $H_h \cap S \neq \emptyset$ and $\overline{S} \cap \emptyset \neq \emptyset$. Their base polytope is then given by:

$$\mathcal{B}(\delta_h) = \{\mathbf{y} \in \mathbb{R}^h \perp \mathbf{1}_h \mid \mathbf{y}(H_h) \leq 1 \wedge \forall i \in T_h \setminus H_h : \mathbf{y}(i) \leq 0\}.$$

The notion of hypergraph flow, which we introduce next, formalizes the network flow interpretation for all hyperedges $h \in E_G$.

Hypergraph Flows Given a weighted submodular hypergraph $G = (V, E, \mathbf{w} \in \mathbb{Z}^E_{>0}, \boldsymbol{\mu} \in \mathbb{Z}^V_{\geq 0})$, a *hyperedge flow* over a hyperedge $h \in E$ is a vector $\mathbf{y}_h \in \mathbb{R}^h \perp \mathbf{1}_h$. A *hypergraph flow* $\mathbf{Y} \in \bigoplus_{h \in E} \mathbb{R}^h \perp \mathbf{1}_h$ is a direct sum of hyperedge flows $\mathbf{Y} = (\mathbf{y}_h)_{h \in E}$. Analogous to the case for graph flows, we define a notion of congestion based on the maximum violation of the polymatroidal constraints defining each $\mathcal{B}(\delta_h)$:

$$\mathrm{cong}_G(\mathbf{Y}) = \min\{\rho \geq 0 : \forall h \in E_G, \, \mathbf{y}_h \in \rho \cdot \mathbf{w}_h \cdot \mathcal{B}(\delta_h)\}. \tag{9}$$

The demand vector $\mathrm{dem}(\mathbf{Y}) \in \mathbb{R}^V \perp \mathbf{1}$ of a hypergraph flow $\mathbf{Y} = (\mathbf{y}_h)_{h \in E}$ is

$$\mathrm{dem}_i(\mathbf{Y}) = \sum_{h \in E : h \ni i} \mathbf{y}_h(i). \tag{10}$$

We will make use of the vector space structure inherited from $\mathbf{Y} \in \bigoplus \mathbb{R}^h \perp \mathbf{1}_h$ to define linear combinations of hypergraph flows $c\mathbf{Y}_1 + \mathbf{Y}_2$ for any $c \in \mathbb{R}$. The demand vector also behaves linearly as $\mathrm{dem}(c\mathbf{Y}_1 + \mathbf{Y}_2) = c \cdot \mathrm{dem}(\mathbf{Y}_1) + \mathrm{dem}(\mathbf{Y}_2)$.

Finally, notice that hypergraph flows over hyperedges of rank 2 are equivalent to graph flows as, for an edge $\{i, j\} \in E$, $\mathbf{y}_{ij}(i)$ represents the flow from i to j, while $\mathbf{y}_{ij}(j) = -\mathbf{y}_{ij}(j)$ is the flow from j to i.

Connection to Polymatroidal Flows In the polymatroidal network flow model [19, 41], one has a directed (2-uniform) graph H representing a flow network where the edges are not individually capacitated, but rather for every vertex $v \in V(H)$, there are monotone submodular functions ρ^- and ρ^+ constraining the amount of flow entering and exiting a vertex respectively.

Given an hypergraph flow $\mathbf{Y} = \{\mathbf{y}_h\}_{h \in E}$ over G, one can interpret it as a graph flow over the factor graph $\hat{G}$, where the flow from vertex i to factor vertex h over the edge $\{i, h\}$ is given by $\mathbf{y}_h(i)$. When the congestion (as defined in (9)) is required to be at most one, the constraints imposed on this flow on $\hat{G}$ can be cast as constraints defining a flow in a polymatroidal network supported on $\hat{G}$. Here, each constraint on the hyperedge flow for hyperedge $h \in E(G)$ defined by its base polytope, can be encoded as polymatroidal constraints on the graph flow into and out of the vertex h in $\hat{G}$.

Despite this connection with polymatroidal network, we remark that, to the best of our knowledge, the hypergraph partitioning problems studied in this paper cannot be expressed in the setup proposed by Chekuri [19] for defining sparsest cut problems over polymatroidal networks.

4.3 Hypergraph Flow Embeddings

To construct efficient flow-based algorithms for the minimum ratio-cut problem in the second part of this paper, we will need a way to generalize the notion of flow embedding of a demand graph into a graph to the setting of hypergraphs. In this section, we leverage the definition of hypergraph flows to provide this generalization. In particular, we define the flow embedding of a (directed) graph into a weighted submodular hypergraphs with polymatroidal cut functions. We also show that these embeddings yield lower bounds on the hypergraph cut functions in terms of the graph cuts of the embedded graph. Analogously to graphs, we also show that any hypergraph flow can be converted into a flow embedding via flow-path decompositions.

Definition 9 (Hypergraph Flow Embedding)**.** Let $G = \big(V, E_G, \mathbf{w}^G, \boldsymbol{\mu}\big)$ be a weighted submodular hypergraph equipped with a collection of polymatroidal cut functions $\{\delta_h\}_{h\in E_G}$ and let $H = \big(V, E_H, \mathbf{w}^H\big)$ be a weighted directed graph on the same vertex set. We say that the graph H embeds into G as a flow with congestion $\rho \geq 0$, denoted $H \preceq_\rho G$ if there exist hypergraph flows $\{\mathbf{Y}^e\}_{e\in E_H}$ such that:

1. For all $e = (i, j) \in E_H$, the flow $\mathbf{Y}^e$ routes w_e^H units of flow from vertex i to vertex j, i.e.,
$$\mathrm{dem}(\mathbf{Y}^e) = w_e^H \cdot (\mathbf{1}_i - \mathbf{1}_j)\,.$$
2. For each $h \in E_G$, the flows $\mathbf{Y}^e{}_{e\in E_H}$ respect the polymatroidal capacity constraints, i.e., for all $h \in E_G$:
$$\sum_{e\in E_h} (\mathbf{y}_h^e)_+ \in \rho \cdot w_h \cdot \mathcal{P}_+(F_h^-),$$
$$\sum_{e\in E_h} (\mathbf{y}_h^e)_- \in \rho \cdot w_h \cdot \mathcal{P}_+(F_h^+).$$

The capacity constraints can be understood by analogy with the capacity constraints for multi-commodity flows over a graph. As in that case, the constraints in the definition ensure that flows from different commodity along the same arc do not cancel out.

We will use this notion of embeddability to certify lower bounds to the ratio-cut objective for general polymatroidal cut functions. We will crucially make use of the following theorem in our development of the cut-matching game for solving ratio-cut problems over weighted submodular hypergraphs. Its proof is in Section A.

Theorem 5. *Let $G = \big(V, E_G, \mathbf{w}^G, \boldsymbol{\mu}\big)$ be a weighted hypergraph equipped with a collection of polymatroidal cut functions $\{\delta_h\}_{h\in E_G}$, and $H = \big(V, E_H, \mathbf{w}^H\big)$ be a weighted directed graph. If $H \preceq_\rho G$, then*

$$\forall \mathbf{x} \in \mathbb{R}^V\,,\ \sum_{(i,j)\in E_H} w_{ij}^H \cdot (\mathbf{x}_i - \mathbf{x}_j)_+ \leq \rho \cdot \sum_{h\in E_G} w_h^G \cdot \overline{\delta}_h(\mathbf{x}) \ \text{ and } \ \forall S \subseteq V\,,\ \delta_H(S) \leq \rho \cdot \delta_G(S).$$

For symmetric polymatroidal cut functions $\{\delta_h\}_{h\in E_G}$, then $H \preceq_\rho G$ implies the following bound on the undirected symmetrization $\tilde{H} = (V, E_{\tilde{H}}, \mathbf{w}^{\tilde{H}})$:

$$\forall \mathbf{x} \in \mathbb{R}^V\,,\ \sum_{\{i,j\}\in E_{\tilde{H}}} w_{ij}^{\tilde{H}} \cdot |\mathbf{x}_i - \mathbf{x}_j| \leq 2 \cdot \rho \cdot \sum_{h\in E_G} w_h^G \cdot \overline{\delta}_h(\mathbf{x}) \ \text{ and } \ \forall S \subseteq V\,,\ \delta_{\tilde{H}(S)} \leq 2 \cdot \rho \cdot \delta_G(S).$$

We now state the main algorithmic result of this section, which allows us to construct hypergraph flow embedding from any single hypergraph flow. Just like in the case of graphs, this result is based on constructing a flow-path decomposition of the original flow. The particular form of polymatroidal cut functions plays monotonicity of the functions F^+ and F^- plays a crucial role in bounding the congestion of the embedding, which cannot be controlled in the same way for general submodular cut functions. The proof appears in Appendix A.

Theorem 6. *Let $G = \big(V, E_G, \mathbf{w}^G, \boldsymbol{\mu}\big)$ be a weighted hypergraph equipped with a collection of polymatroidal cut functions $\{\delta_h\}_{h\in E_G}$. Given a hypergraph flow $\mathbf{Y}$ over G, there is an algorithm that, in time $\tilde{O}\big(\mathsf{sp}(G)\big)$, computes a weighted directed bipartite graph $H = \big(V, E_h, \mathbf{w}^H\big)$ such that $H \preceq_{\mathrm{cong}_G(\mathbf{Y})} G$ and $|E_H| = \tilde{O}\big(\mathsf{sp}(G)\big)$. Furthermore, all vertices $i \in V$ with non-negative (resp. negative) demand $\mathrm{dem}_i(\mathbf{Y})$ are source nodes (resp. sink nodes) in H, each with out-degree (resp. in-degree) equal $\mathrm{dem}_i(\mathbf{Y})$.*

5 An $O(\sqrt{\log n})$-Approximation via ℓ_2^2-Metric Embeddings

In this section, we prove Theorem 1 by giving an ℓ_2^2-metric relaxation and rounding argument yielding a randomized polynomial-time $O(\sqrt{\log n})$-approximation algorithm for the minimum ratio-cut problem over submodular hypergraphs equipped with polymatroidal cut functions. As a warm-up, we give the simpler argument for symmetric polymatroidal cut functions, before proving the result for general polymatroidal cut functions in Section 5.2

5.1 Warm-up: Symmetric Polymatroidal Cut Functions

We begin by providing a relaxation for the symmetric version of the problem, in which $F_h^+ = F_h^-$ for every $h \in E$. All the results in this subsection will focus exclusively on this special case, but they will be extended to the general case of all polymatroidal cut functions in the following subsection.

Given a weighted submodular hypergraph $G = (V, E, \mathbf{w}, \boldsymbol{\mu})$, we construct a vector-program relaxation of the continuous non-convex formulation in Equation (`RC-NonCvx`) by associating to each vertex $i \in V \cup E$ of the factor graph $\hat{G}$ a vector $\mathbf{v}_i$. For a hyperedge $h \in E$ and $i \in h$, we can then relax the terms $|\mathbf{x}_i - \nu_h \mathbf{1}_h|$, which make up the argument of $\bar{F}_h$ in Fact 5, with the distance $\mathbf{d}_i^h \overset{\text{def}}{=} \|\mathbf{v}_i - \mathbf{v}_h\|^2$. We obtain the following semidefinite program in its vector embedding form:

$$
\begin{aligned}
\min_{\mathbf{v}_i} \quad & \sum_{h\in E} w_h \cdot \bar{F}_h(\mathbf{d}^h) \\
\text{s.t.} \quad & \sum_{\{i,j\}\subseteq V} \frac{\mu(i)\mu(j)}{\mu(V)} \cdot \|\mathbf{v}_i - \mathbf{v}_j\|_2^2 \geq 1 \\
& \|\mathbf{v}_i - \mathbf{v}_j\|_2^2 \leq \|\mathbf{v}_i - \mathbf{v}_k\|_2^2 + \|\mathbf{v}_k - \mathbf{v}_j\|_2^2 && \forall\, i,j,k \in V \cup E \\
& \mathbf{d}_i^h = \|\mathbf{v}_i - \mathbf{v}_h\|^2 && \forall h \in E, i \in h \\
& \mathbf{v}_i \in \mathbb{R}^n, \mathbf{d}^h \in \mathbb{R}^h_{\geq 0} && \forall\, i \in V \cup E, \forall\, h \in E
\end{aligned}
\tag{RC-Metric}
$$

The following simple lemma is proved in Appendix A.

Lemma 2. *The* (`RC-Metric`) *vector program is, up to a constant, a relaxation of the minimum ratio-cut program for the hypergraph $G = (V, E, \mathbf{w}, \boldsymbol{\mu})$ with symmetric polymatroidal cut functions $\{\delta_h\}_{h\in E}$.*

Rounding via Metric Embedding To round the convex program (`RC-Metric`), we apply the following well-known embedding result, which is implicit in the seminal work of Arora, Rao and Vazirani [10].

Theorem 7 ([10])**.** *Let $\{\mathbf{v}_i \in \mathbb{R}^d\}_{i\in V\cup E}$ be a feasible solution to the ℓ_2^2-metric relaxation* (`RC-Metric`)*. Then, there exists an efficiently computable 1-Lipschitz[2] map $\phi : \mathbb{R}^d \to \mathbb{R}$ satisfying:*

$$
\frac{\sum_{i,j\in V} \mu(i)\mu(j) \cdot |\phi(\mathbf{v}_i) - \phi(\mathbf{v}_j)|}{\sum_{i,j\in V} \mu(i)\mu(j) \cdot \|\mathbf{v}_i - \mathbf{v}_j\|^2} \geq \Omega\left(\frac{1}{\sqrt{\log |V|}}\right).
$$

[2] Recall that, given a metric space (X, d), a function $f : X \to \mathbb{R}$ is said to be K-Lipschitz for some constant K if $|f(x) - f(y)| \leq K \cdot d(x, y)$ for every $x, y \in X$.

To complete the proof of Theorem 1 for the special case of symmetric polymatroidal cut functions, we make use of the embedding in Theorem 7. The proof crucially relies on the monotonicity of the Lovász extension $\bar{F}$ (Fact 1). The ability to perform this step can be taken as a justification for our definition of polymatroidal cut functions.

Lemma 3. *There is a polynomial-time algorithm that given a solution $\{\mathbf{v}_i\}_{i\in V\cup E}$ to (`RC-Metric`) of objective value $\kappa := \sum_{h\in E} w_h \cdot \bar{F}_h(\mathbf{d}^h)$, returns a cut $S \subseteq V$ such that:*

$$\Psi_G(S) \le O\big(\kappa \cdot \sqrt{\log |V|}\big)\,.$$

Proof. TOPROVE 0 □

5.2 General Polymatroidal Cut Functions

For the case of general polymatroidal cut function, our metric relaxation must be able to capture the signed terms $(\mathbf{x}_h - \nu_h \mathbf{1}_h)_+$ and $(\mathbf{x}_h - \nu_h \mathbf{1}_h)_-$ that appear in the Lovász extension in Fact 3. For this purpose, we will use the directed ℓ_2^2-semimetrics introduced by Charikar, Makarychev and Makarychev [18] and, for every $h \in E$ and $i \in h$, relax $(\mathbf{x}_i - \nu_h)_+$ to $\mathbf{d}_i^{h,-} \overset{\text{def}}{=} \|\mathbf{v}_i - \mathbf{v}_h\|^2 + \|\mathbf{v}_i\|_2^2 - \|\mathbf{v}_h\|_2^2$ and $(\mathbf{x}_i - \nu_h)_-$ to $\mathbf{d}_i^{h,+} \overset{\text{def}}{=} \|\mathbf{v}_i - \mathbf{v}_h\|^2 + \|\mathbf{v}_h\|_2^2 - \|\mathbf{v}_i\|_2^2$. We obtain the following semidefinite program:

$$\begin{aligned}
\min_{\mathbf{v}_i, \boldsymbol{\ell}^h} \quad & \sum_{h\in E} w_h \cdot \big(\bar{F}_h^-(\mathbf{d}_h^-) + \bar{F}_h^+(\mathbf{d}_h^+)\big) \\
\text{s.t.} \quad & \sum_{\{i,j\}\subseteq V} \frac{\mu(i)\mu(j)}{\mu(V)} \cdot \|\mathbf{v}_i - \mathbf{v}_j\|_2^2 \ge 1 \\
& \|\mathbf{v}_i - \mathbf{v}_j\|_2^2 \le \|\mathbf{v}_i - \mathbf{v}_k\|_2^2 + \|\mathbf{v}_k - \mathbf{v}_j\|_2^2 && \forall\, i,j,k \in V \cup E \\
& \mathbf{d}_i^{h,-} = \|\mathbf{v}_i - \mathbf{v}_h\|^2 + \|\mathbf{v}_i\|_2^2 - \|\mathbf{v}_h\|_2^2 && \forall h \in E, i \in h \\
& \mathbf{d}_i^{h,+} = \|\mathbf{v}_i - \mathbf{v}_h\|^2 + \|\mathbf{v}_h\|_2^2 - \|\mathbf{v}_i\|_2^2 && \forall h \in E, i \in h \\
& \mathbf{v}_i \in \mathbb{R}^n, \mathbf{d}^{h,+}, \mathbf{d}^{h,-} \in \mathbb{R}^h_{\ge 0} && \forall\, i \in V \cup E,\ \forall\, h \in E
\end{aligned} \qquad (\texttt{RC-Directed-Semimetric})$$

We have the following lemma, which we prove in Appendix A.

Lemma 4. *The* (`RC-Directed-Semimetric`) *vector program is, up to a constant, a relaxation of the minimum submodular hypergraph ratio-cut problem for the hypergraph $G = (V, E, \mathbf{w}, \boldsymbol{\mu})$ with polymatroidal cut functions $\{\delta_h\}_{h\in E}$.*

Rounding via Metric Embedding To round a solution to the program (`RC-Directed-Semimetric`) to a solution of the original ratio-cut problem (`RC-NonCvx`), we require a version of the embedding result of Theorem 7 for directed semimetrics. The following result is a simple consequence of the rounding algorithm of Agarwal *et al.* [2] for a similar relaxation of directed expansion. We prove it for completeness in Appendix A.

Theorem 8. *Given a feasible solution $\{\mathbf{v}_i \in \mathbb{R}^d\}_{i\in V\cup E}$ to `RC-Directed-Semimetric` and a measure $\boldsymbol{\mu} \in \mathbb{R}^V_{\ge 0}$, there exists a polynomial-time computable map $\phi : \mathbb{R}^d \to \mathbb{R}$ such that:*

1. $(\phi(\mathbf{v}) - \phi(\mathbf{w}))_+ \le \|\mathbf{v} - \mathbf{w}\|^2 + \|\mathbf{v}\|^2 - \|\mathbf{w}\|^2$ *for all* $\mathbf{v}, \mathbf{w} \in \mathbb{R}^d$,
2. *and:*

$$\frac{\sum_{i,j\in V} \mu(i)\mu(j)|\phi(\mathbf{v}_i) - \phi(\mathbf{v}_j)|}{\sum_{i,j\in V} \mu(i)\mu(j)\|\mathbf{v}_i - \mathbf{v}_j\|_2^2} \ge \Omega\left(\frac{1}{\sqrt{\log |V|}}\right)$$

We can now complete the proof of Theorem 1 for the general polymatroidal case by showing that the map ϕ in Theorem 8 yields a $O(\sqrt{\log n})$-approximate solution to the minimum ratio-cut problem.

Lemma 5. *Given a solution* $(\{\mathbf{v}_i\}_{i \in V \cup \in E})$ *to* `RC-Directed-Semimetric` *of value* κ*, we can produce in polynomial time a set* $S \subseteq V$ *such that:*

$$\Psi_G(S) \leq O(\kappa \cdot \sqrt{\log |V|}).$$

Proof. TOPROVE 1 □

6 Localized Convex Formulations and Ratio-Cut Improvement

In this section, we consider the localized formulation (`Local-RC`) introduced in Section 2.2, which we rewrite as a convex program. We show that the dual solutions to this problem are hypergraph flows, which can be used to obtain hypergraph flow embeddings into the input graph. Finally, we give algorithms that construct approximate primal-dual solutions for the (`Local-RC`) problem. These algorithms will play an important part in our cut-matching game.

We start by introducing the *ratio-cut improvement* objective, a local partitioning objective parametrized by a seed $\mathbf{s} \in R^V$, which is the integral analogue of the (`Local-RC`) problem:

Definition 10 (Ratio-Cut Improvement Problem)**.** Given a weighted submodular hypergraph $G = (V, E, \mathbf{w}, \boldsymbol{\mu})$, cut functions $\{\delta_h\}_{h \in E}$, and a *seed vector* $\mathbf{s} \in \mathbb{R}^V$ with $\langle \mathbf{s}, \mathbf{1} \rangle_\mu = 0$ and $\|\mathbf{s}\|_\infty \leq 1$, the ratio-cut improvement objective $\Psi_{G,s}$ is defined for all $S \subset V$ as:

$$\Psi_{G,\mathbf{s}}(S) \stackrel{\text{def}}{=} \frac{\sum_{h \in E} w_h \cdot \delta_h(S \cap h)}{\max\{0, \langle \mathbf{s}, \mathbf{1}^S \rangle_\mu\}}$$

$$\Psi^*_{G,\mathbf{s}} \stackrel{\text{def}}{=} \min_{S \subseteq V} \Psi_{G,\mathbf{s}}(S)$$

The ratio-cut improvement problem asks to find $S \subseteq V$ achieving $\Psi^*_{G,\mathbf{s}} = \Psi_{G,\mathbf{s}}(S)$.

The ratio-cut improvement problem can be thought of as attempting to find a cut $(S, \overline{S})$ which has both a small boundary and high correlation with the input seed $\mathbf{s}$ simultaneously. In Appendix B, we demonstrate that the problem of minimizing $\Psi_{G,\mathbf{s}}$ is precisely the generalization of the cut-improvement problem of Andersen and Lang [7] to weighted submodular hypergraphs and fractional seeds.

The relation between the minimum ratio-cut problem, the ratio-cut improvement problem and their continuous formulations (`RC-NonCvx`) and (`Local-RC`) is formalized in the following lemma.

Lemma 6. *For all* $\mathbf{x} \in \mathbb{R}^V$*, and seeds* $\mathbf{s} \in \mathbb{R}^V$ *satisfying* $\|\mathbf{s}\|_\infty \leq 1$ *and* $\langle \mathbf{s}, \mathbf{1} \rangle_{\boldsymbol{\mu}} = 0$*, we have:*

$$\bar{\Psi}_G(\mathbf{x}) \leq \bar{\Psi}_{G,\mathbf{s}}(\mathbf{x}) \,.$$

In particular, setting $\mathbf{x} = \mathbf{1}^S$ *yields* $\Psi^*_{G,\mathbf{s}}$ *is a relaxation of* Ψ^*_G *for any* $S \subset V$*:*

$$\Psi_G(S) \leq \Psi_{G,\mathbf{s}}(S) \,.$$

Proof. TOPROVE 2 □

Of critical, algorithmic, consequence is the fact that computing the value of $\bar{\Psi}^*_{G,\mathbf{s}}$ can be written as a convex program. We refer to this program as (`CI-Primal`):

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^V} \quad & \sum_{h \in E} w_h \cdot \bar{\delta}_h(\mathbf{x}) \\ \text{s.t.} \quad & \langle \mathbf{s}, \mathbf{x} \rangle_\mu = 1 \end{aligned} \tag{CI-Primal}$$

The following lemma is a simple consequence of the positive homogeneity of the Lovász extensions of submodular functions (Proposition 3.1 in [11]).

Lemma 7. *The optimum value of* (`CI-Primal`) *equals* $\Psi^*_{G,\mathbf{s}}$.

With this convex program, one can leverage convex duality to construct lower bounds for $\bar{\Psi}^*_{G,\mathbf{s}}$, and consequently $\Psi^*_{G,\mathbf{s}}$. We consider a dual formulation of (`CI-Primal`) as a problem over hypergraph flows:

$$\begin{aligned} \max \quad & \alpha \\ \text{s.t.} \quad & \operatorname{dem}(\mathbf{Y}) = \alpha \cdot \big(\boldsymbol{\mu} \circ \mathbf{s}\big) \\ & \operatorname{cong}_G(\mathbf{Y}) \leq 1 \\ & \alpha \in \mathbb{R}, \mathbf{Y} = \{\mathbf{y}_h \in \mathbb{R}^h \perp \mathbf{1}_h\}_{h \in E} \end{aligned} \tag{\texttt{CI-Dual}}$$

In this setting, strong duality between (`CI-Primal`) and (`CI-Dual`) holds by virtue of the (`CI-Primal`) having a finite number of polyhedral constraints. The full proof of the following lemma can be found in Appendix A.

Lemma 8. *The programs* (`CI-Primal`) *and* (`CI-Dual`) *are dual to each other. Strong duality holds.*

From this point onwards, we restrict our attention to the kind of seed vectors that we will encounter in our application to the cut-matching game. For this purpose, we define the following notation for disjoint sets $A, B \subseteq V$ with $\mu(A) \leq \mu(B)$:

$$\mathbf{1}^{A,B} = \mathbf{1}_A - \frac{\mu(A)}{\mu(B)}\mathbf{1}_B.$$

Notice that $\mathbf{1}^{A,B}$ is a valid seed for the local formulation of the minimum ratio-cut problem as $\|\mathbf{1}^{A,B}\|_\infty \leq 1$ and $\langle \mathbf{1}^{A,B}, \mathbf{1}\rangle_{\boldsymbol{\mu}} = 0$. For this kind of seed, the dual problem can be interpreted as a maximum concurrent single-commodity flow over the hypergraph G, in which we each vertex $i \in A$ is asked to route demand μ_i to B and each vertex j in B is asked to receive demand ${}^{\mu(A)}/_{\mu(B)} \cdot \mu_j$ from A. For a solution of value α, each vertex will be able to route an α fraction of its demand and the total demand routed from A to B will be $\alpha\mu(A)$. Notice that, in the graph case, for $B = \bar{A}$, this construction specializes exactly to the undirected maximum concurrent flow problem used by Andersen and Lang [7] to solve the ratio-cut improvement problem.

Approximate Solutions In Section 6.2, we will discuss how, for many kinds of polymatroidal cut functions, approximately feasible dual solutions to (`CI-Dual`) can be obtained by approximately solving a maximum flow problem over the factor graph or an augmented version of it. For this purpose, it is convenient to introduce a notion of approximate dual solution of value α to (`CI-Dual`).

Definition 11 (Approximate dual solution)**.** Let $G = \big(V, E_G, \mathbf{w}^G, \boldsymbol{\mu}\big)$ be a weighted submodular hypergraph equipped with polymatroidal cut functions $\{\delta_h\}_{h \in E}$, and $A, B \subseteq V$ disjoint sets with $\mu(A) \leq \mu(B)$. An *approximate dual solution of value* $\alpha \geq 0$ for $(G, \mathbf{1}^{A,B})$ is a hypergraph flow $\mathbf{Y} = \big(\mathbf{y}^h \in \mathbb{R}^h \perp \mathbf{1}_h\big)$ such that the following hold simultaneously:

1. $\mathbf{Y}$ has unit congestion, i.e., $\operatorname{cong}_G(\mathbf{Y}) \leq 1$,
2. $\mathbf{Y}$ routes a ${}^1/_2$-fraction of the required flow from A to B, i.e., at least ${}^1/_2 \cdot \alpha \cdot \mu(A)$ flow,
3. $\mathbf{Y}$ does not exceed the require demand at every vertex, i.e., $|\operatorname{dem}(\mathbf{Y})| \leq \alpha \cdot |\boldsymbol{\mu} \circ \mathbf{1}^{A,B}|$.

6.1 Graph Certificates from Approximate Dual Solutions

As discussed in Section 2.2, our plan is to apply the cut-matching game to combine local lower bounds given by approximately optimal solutions to (`CI-Dual`) into global lower bounds for the non-convex problem $\bar{\Psi}^*_G$. To carry out this plan, we need to express our local lower bounds, i.e. dual solutions to (`CI-Dual`) in terms of hypergraph embeddings of graphs into the instance hypergraph.

Definition 12. Let $G = \big(V, E_G, \mathbf{w}^G, \boldsymbol{\mu}\big)$ be a weighted submodular hypergraph equipped with polymatroidal cut functions $\{\delta_h\}_{h\in E}$, and $A, B \subseteq V$ disjoint sets with $\mu(A) \leq \mu(B)$. Let $\alpha = \Psi^*_{G,\mathbf{1}^{A,B}}$. An *approximate dual graph certificate of value* $\alpha \geq 0$ for the (`Local-RC`) problem with seed $\mathbf{1}^{A,B}$ is a directed graph $D = (V, E_D, \mathbf{w}^D)$ with the following properties:

1. D is sparse, i.e., $|E_D| \leq \tilde{O}(\mathsf{sp}(G))$,
2. $\alpha \cdot D$ embeds into G with congestion 1, or equivalently $D \preceq_{1/\alpha} G$,
3. D is bipartite from A to B, i.e. all arcs of D go from A to B,
4. for all $i \in A, j \in B$, we have the degree bounds $\deg_D(i) \leq \mu_i$ and $\deg_D(j) \leq {}^{\mu(A)}/_{\mu(B)} \cdot \mu_j$,
5. D is large, i.e., $w^D(E(A,B)) \geq {}^1/_2 \cdot \mu(A)$.

Following Theorem 5, we may assume that D is undirected if all cut functions are symmetric.

In the cut-matching game framework, we will use dual graph certificates to iteratively construct lower bounds to Ψ^*_G through the application of Theorem 5. For example, given an approximate dual graph certificate D of value α, Theorem 5 yields the lower bound:

$$\forall S \subseteq V\,, \delta_G(S) \geq \alpha \cdot \delta_D(S) = \alpha \cdot w^D(E(S \cap A, \bar{S} \cap B)).$$

As we built D from a dual solution for (`CI-Dual`) with seed $\mathbf{1}^{A,B}$, this lower bound is tighter for cuts S that are well-correlated with $\mathbf{1}^{A,B}$, reaching its maximum when $A \subseteq S \subseteq \bar{B}$.

We can now apply the flow-path decomposition of Theorem 6 to show how to efficiently construct an approximate dual graph certificate from an approximate dual solution. The straightforward proof is found in Appendix A.

Lemma 9. *Under the same assumptions of Definition 12, consider an approximate dual solution* $\mathbf{Y}$ *of value* α *to the ratio-cut improvement problem on* $(G, \mathbf{1}^{A,B})$*. Let* H *be the directed graph obtained by applying the flow-path decomposition algorithm of Theorem 6 to the hypergraph flow* $\mathbf{Y}$*. Then, the scaled graph* $\frac{1}{\alpha} \cdot H$ *is an approximate dual graph certificate of value* α*.*

6.2 Approximately Solving (`Local-RC`)

In this section, we discuss algorithms that construct approximate solutions to (`CI-Primal`) and (`CI-Dual`), the primal and dual convex formulations of (`Local-RC`) . Our specific focus is to describe algorithms that implement the following specification:

Definition 13. Given a weighted submodular hypergraph $G = (V, E, \mathbf{w}, \boldsymbol{\mu})$ with polymatroidal cut functions $\{\delta_h\}_{h\in E}$ and disjoint sets $A, B \subset V$ with $\mu(A) \leq \mu(B)$, an *approximate primal-dual oracle* problem is an algorithm $\mathcal{A}_{\mathtt{CI}}$ that takes as input G and the seed vector $\mathbf{1}^{A,B}$. For some $\alpha \geq 0$, the algorithm $\mathcal{A}_{\mathtt{CI}}$ outputs both:

1. a cut $S \subseteq V$ with $\Psi_{G,\mathbf{1}^{A,B}}(S) \leq 3\alpha$,
2. an approximate dual graph certificate of value α.

We prove the following two theorems in Appendix C by a simple application of binary search over α. The first thereom bounds the complexity of constructing an approximate primal-dual oracle for general polymatroidal cut functions.

Theorem 9. *For submodular hypergraphs with general polimatroidal cut functions, an approximate primal-dual oracle can be implemented in time* $\tilde{O}\big(|V| \cdot \sum_{h\in E} \Theta_h\big)$*, where* Θ_h *is the running time for a quadratic optimization oracle for* δ_h*, by solving* $O(\log|V|)$*-many decomposable submodular minimization problems.*

Our second theorem shows how this result can be significantly improved in the case of the standard hypergraph cut functions by relying on almost-linear-time maximum flow solvers [13, 20].

Theorem 10. *For a submodular hypergraph G equipped with the directed or standard hypergraph cut functions, an approximate primal-dual oracle can be implemented in almost linear-time by solving $O(\log |V|)$-many $1/2$-approximate maximum flow problems over a directed, vertex-capacitated version of the factor graph $\hat{G}$.*

7 An $O(\log n)$-Approximation via Generalized Cut-Matching Games

We now describe how to use our cut-matching game to approximate minimum ratio-cut. Let us begin by stating the precise approximation guarantee garnered by running a game between a good cut strategy, and an approximate primal-dual solver for the ratio-cut improvement problem (`Local-RC`).

Theorem 11. *Let $G = (V, E, \mathbf{w}, \boldsymbol{\mu})$ be an input submodular hypergraph equipped with polymatroidal cut functions. Consider an execution of the cut-matching game in which the cut player $\mathcal{C}$ plays an $\big(f(n), g(n)\big)$-good cut strategy, while the matching player plays the dual graph certificates output by an approximate primal-dual oracle $\mathcal{A}_{\mathtt{CI}}$ (Definition 13) on G. Let C_t be the cuts output by $\mathcal{A}_{\mathtt{CI}}$. Then, with constant probability, we have the following approximation result:*

$$\min \Big\{ \Psi_G(C_t) \Big\}_{t=1}^{g(n)} \leq O\bigg(\frac{g(n)}{f(n)}\bigg) \cdot \Psi_G^*.$$

This is a standard result for cut-matching games and its proof does not deviate too far from that in previous work. We give its proof in Appendix D for completeness.

Theorem 11 reduces the task of approximating minimum ratio-cut to constructing a good cut strategy. Hence, this section is devoted towards proving the following theorem.

Theorem 12. *Let $H = \big(V, E_H, \mathbf{w}^H, \boldsymbol{\mu}\big)$ be the state graph for an $(n, \boldsymbol{\mu})$-generalized cut-matching game. There exists a cut strategy $\mathcal{A}_{\mathsf{cut}}$ satisfying the following:*

1. *If H is undirected, then $\mathcal{A}_{\mathsf{cut}}$ is $\big(\Omega(\log n), O(\log^2 n)\big)$-good with probability $O(1)$.*
2. *If H is directed, then $\mathcal{A}_{\mathsf{cut}}$ is $\big(\Omega(\log^2 n), O(\log^3 n)\big)$-good with probability $O(1)$.*

Furthermore, $\mathcal{A}_{\mathsf{cut}}$ outputs a cut action in time $\tilde{O}\big(\mathsf{sp}(H)\big)$.

To prove this theorem, we give an optimization-based analysis of the cut-matching game by using regret minimization techniques from online optimization to construct and analyze $\mathcal{A}_{\mathsf{cut}}$. This approach is very natural: the goal of online optimization is to adaptively produce a sequence of actions that will lead to some terminal outcome despite the presence of adversarial response. For the cut-matching game, the cut player's goal is to produce a short sequence of cut actions such that, no matter what approximate dual responses are given by $\mathcal{M}$, the state graph H will have large ratio-cut objective.

This approach also provides insight into why earlier combinatorial restrictions on valid cut and matching responses are not necessary when using the cut-matching game to partition (hyper)graphs. An interesting byproduct of our construction, in the case where $\{\delta_h\}_{h \in E}$ are asymmetric, is that $\mathcal{M}$ does not necessarily route Eulerian demand graphs. Additionally, no further reductions are required to enforce these combinatorial restrictions when approximating ratio-cut objectives beyond expansion.

To prove Theorem 12, we first describe our construction of $\mathcal{A}_{\mathsf{cut}}$ for the case where the cut functions defining the ratio-cut objective are symmetric, and H is undirected. Simply considering this case already demonstrates a large fraction of our technical contributions. We then consider asymmetric cut functions and directed H, highlighting key differences from the symmetric case.

7.1 An $\big(\Omega(\log n), O\big(\log^2 n\big)\big)$-good Cut Strategy for Symmetric Cut Functions

We first consider the case where the polymatroidal cut functions $\{\delta_h\}_{h\in E}$ are symmetric.

Let us start by instantiating the online linear optimization setting in the context of our cut-matching game. For an instance of a $(n, \boldsymbol{\mu})$-generalized cut-matching game, fix the action set to be $\mathcal{X} = \Delta_{\boldsymbol{\mu}}^{n\times n}$ where

$$\Delta_{\boldsymbol{\mu}}^{n\times n} \stackrel{\text{def}}{=} \big\{\mathbf{X} \succeq \mathbf{0} : \big\langle \mathbf{L}(K_{\boldsymbol{\mu}}), \mathbf{X}\big\rangle \geq 1\big\}\,.$$

At the end of round t, the matching player $\mathcal{M}$ will have produced approximate matching responses $D_1, \ldots, D_t$. Choose the t-th feedback matrix to be $\mathbf{F}_t = \mathbf{L}(D_t)$. The loss incurred during round t is then

$$\big\langle \mathbf{L}(D_t), \mathbf{X}\big\rangle\,.$$

We instantiate online linear optimization in this way because any bound on regret now translates to a lower bound on the ratio-cut objective for the state graph H. When $\{\delta_h\}_{h\in E}$ are symmetric, H is an undirected graph, and Ψ_H^* is simply the undirected ratio-cut objective, which has the following trivial SDP relaxation:

Lemma 10. *Given any undirected graph $G = \big(V, E_G, \mathbf{w}^G, \boldsymbol{\mu}\big)$, we have $\frac{\mathsf{opt}}{2} \leq \Psi_G^*$ where* opt *is the optimal objective value for the following semidefinite program.*

$$\begin{aligned} \min \quad & \big\langle \mathbf{L}(G), \mathbf{X}\big\rangle \\ \text{s.t.} \quad & \big\langle \mathbf{L}(K_{\boldsymbol{\mu}}), \mathbf{X}\big\rangle \geq 1 \\ & \mathbf{X} \succeq \mathbf{0} \end{aligned} \qquad (\texttt{cond-sdp})$$

Our choice of action set thus coincides with the feasible set for (`cond-sdp`), while the feedback matrices are chosen so that the sum of the losses give the SDP objective for the state graph. This is as intended; we want the cut player to force a state graph with large ratio-cut objective, thus it is crucial to have some lower bound on the ratio-cut objective that is a function of both cut, and matching responses. This is also to be expected; if we wanted regret minimizing strategies to produce approximate solutions for (`cond-sdp`), then we would have defined the action set and losses in exactly the same way.

Three issues remain if we wish to use a regret bound from Theorem 4 to analyze a good cut strategy:

(1) *Ensuring Small Width*: We need to ensure that the width term $\big\|\mathbf{M}^{-1/2}\mathbf{L}(D_t)\mathbf{M}^{-1/2}\big\|_{\mathsf{spec}}$ is small given any approximate matching response D_t.

(2) *Ensuring Large Loss*: For our cut strategy to be good, we will need to specify how to produce a cut action such that the loss $\big\langle \mathbf{L}(D_t), \mathbf{X}_t\big\rangle$ regardless of what approximate matching response D_t is played.

(3) *Implementing* $\mathtt{MMWU}_{\eta,\boldsymbol{\mu}}$ *in Nearly-linear Time*: If we want our cut player to run in nearly-linear time, then we will (a) need to compute the update $\mathtt{MMWU}_{\eta,\boldsymbol{\mu}}$ in nearly-linear time, and (b) ensure that queries on the vector embedding given by the Gram matrix of the update can be computed in sublinear time.

Let us address these issues now.

7.1.1 Ensuring Small Width

The width term is constant for any approximate matching response. This follows simply from the fact that the degree of any vertex i is bounded by μ_i.

Lemma 11. *Any approximate matching response $D = \big(V, E_D, \mathbf{w}^D, \boldsymbol{\mu}\big)$ satisfies $\mathbf{0} \preceq \mathbf{L}(D) \preceq 2\cdot\mathbf{M}$.*

Notice that, since $\mathbf{L}(D) \succeq \mathbf{0}$, the regret guarantee in Theorem 4 holds for any step-size $\eta > 0$.

7.1.2 Ensuring Large Loss

Let $\{\mathbf{v}_i^{(t)}\}_{i\in V} \subseteq \mathbb{R}^d$ be a vector embedding of V whose Gram matrix is $\mathbf{X}_t$. If $D_t = \big(V, E_{D_t}, \mathbf{w}^{D_t}\big)$ is the approximate matching response at time t, then the loss $\langle \mathbf{L}_{D_t}, \mathbf{X}_t\rangle$ is equivalent to

$$\langle \mathbf{L}_{D_t}, \mathbf{X}_t\rangle = \sum_{(i,j)\in E_{D_t}} w_{ij}^{D_t} \cdot \big\|\mathbf{v}_i^{(t)} - \mathbf{v}_j^{(t)}\big\|^2 .$$

For our cut strategy to be good, we need to produce a cut action (A_t, B_t) such that, no matter how D_t is played, the edges of D_t will always be placed across pairs of embedding vectors $\mathbf{v}_i^{(t)}$, $\mathbf{v}_j^{(t)}$ that are well-separated under the ℓ_2^2-norm. This can be difficult to guarantee if the flow routed is approximate. If the primal-dual solver can only route a constant factor approximate flow, then an adversarially chosen constant fraction of the demand may be dropped. If one naively rounds (A_t, B_t) using a random 1-dimensional projection, as in [9, 53], then the rounding can produce an unbalanced cut, upon which the adversary can choose to drop a fraction of demand that routes between *all* vertex pairs whose embedding vectors are far apart in ℓ_2^2-distance.

Previous works [23, 22, 49] have addressed this issue by routing a polylogarithmic number of $O(1)$-factor approximate maximum flows during a single iteration of the cut-matching game However, a technical contribution of this construction is that a *single* approximate primal-dual solve of (`Local-RC`) suffices, provided (A_t, B_t) are σ-separated in the sense of Definition (8).

The following lemma shows that any approximate matching response routing demand across σ-separated (A_t, B_t) must always route a large amount of flow between embedding vectors that are well separated in ℓ_2^2-distance. Using (A_t, B_t) as the cut action will thus always result in large loss.

Lemma 12. *Given $\boldsymbol{\mu} \in \mathbb{Z}_{\geq 0}$, let $\mathbf{X} \in \Delta_{\boldsymbol{\mu}}^{n\times n}$ and $\{\mathbf{v}_i\}_{i\in V} \subseteq \mathbb{R}^d$ be the vector embedding whose Gram matrix is $\mathbf{X}$. If (A, B) is σ-well separated with respect to $\boldsymbol{\mu}$ and $\{\mathbf{v}_i\}_{i\in V}$, then for any approximate matching response $D = \big(A \cup B, E_D, \mathbf{w}^D\big)$ with respect to (A, B), we have:*

$$\sum_{\{i,j\}\in E_D} w_{ij}^D \cdot \|\mathbf{v}_i - \mathbf{v}_j\|_2^2 \geq \frac{\sigma}{2} .$$

Proof. TOPROVE 3 □

In section 8, we give an algorithm `round-cut` that produces $\Omega\big(\frac{1}{\log n}\big)$-separated (A_t, B_t). This algorithm will subsequently be used in our cut strategy construction for the symmetric case.

7.1.3 Implementing `MMWU` in Nearly-linear Time

Computing $\mathtt{MMWU}_{\eta,\boldsymbol{\mu}}$ in nearly-linear time, along with ℓ_2^2-distances and projections in logarithmic time are standard techniques. We briefly sketch them here.

Lemma 6 in [9] states one can compute a matrix-vector product with a matrix exponential by truncating its Taylor expansion. Given any $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\mathbf{v} \in \mathbb{R}^n$, one can compute $\mathbf{u} \in \mathbb{R}^n$ such that

$$\big\|\mathbf{u} - \exp(\mathbf{A})\mathbf{v}\big\| \leq \epsilon$$

in time $O\big(\mathcal{T}_{\mathsf{mv}} \cdot \max\big\{\|\mathbf{M}\|, \log\big(\frac{1}{\epsilon}\big)\big\}\big)$ where $\mathcal{T}_{\mathsf{mv}}$ is matrix-vector product time. To compute $\mathtt{MMWU}_{\eta,\boldsymbol{\mu}}$ in nearly-linear time, we take advantage of two details.

First, each loss matrix provided to `MMWU` is given by the Laplacian of a graph D_t. Consequently, the sparsity of the matrix being exponentiated in `MMWU` is bounded by $\mathsf{sp}(H_t)$. Consequently, matrix-vector product time satisfies $\mathcal{T}_{\mathsf{mv}} = O\big(\mathsf{sp}(H_T)\big)$ where T is the total number of cut-matching game rounds. Second, Lemma 11 implies that $\big\|\mathbf{M}^{-1/2}\mathbf{L}(D_t)\mathbf{M}^{-1/2}\big\|_{\mathsf{spec}} \leq O(1)$ for every $t \in T$, and thus

$$\Big\|\sum_{k=1}^{t} \mathbf{M}^{-1/2}\mathbf{L}(D_t)\mathbf{M}^{-1/2}\Big\|_{\mathsf{spec}} \leq O(t) .$$

If we run at most $\mathsf{polylog}(n)$ iterations of the cut-matching game, we can compute an $O\big(\frac{1}{\mathsf{poly}(n)}\big)$-additive approximation to $\texttt{MMWU}$ by truncating the Taylor expansion up to a $\mathsf{polylog}(n)$ number of terms.

To compute ℓ_2^2-distances, and projections in logarithmic time, one can use the Johnson–Lindenstrauss transform [33] to project the embedding into $O\big(\frac{\log n}{\delta^2}\big)$ dimensions for $\delta = O(1)$ (See, e.g. [24, 1, 14]). Since the error is multiplicative in terms of pairwise ℓ_2^2-distances, picking δ to be constant only causes a constant factor loss in the number of cut strategy plays, and the final approximation ratio. We will ignore this constant in the subsequent calculations [3].

7.1.4 Completing the Analysis for Symmetric Cut Functions

We are now ready to describe the cut strategy which we call $\mathcal{A}^{\leftrightarrow}_{\mathsf{cut}}$. Fix a step size $\eta > 0$ to be chosen subsequently, and consider round t of a $(n, \boldsymbol{\mu})$-generalized cut-matching game. Up to this point, the matching player $\mathcal{M}$ will have produced approximate matching responses $D_1, \ldots, D_t$.

The cut strategy is straightforward: use Johnson-Lindenstrauss and the truncated Taylor expansion, as in Lemma 6 of [9], to compute a low-dimensional projection of the embedding whose Gram matrix is given by $\texttt{MMWU}_{\eta,\boldsymbol{\mu}}\big(\mathbf{L}(D_1), \ldots, \mathbf{L}(D_t)\big)$, execute $\texttt{round-cut}$ on the resulting embedding to produce σ-separated sets (A_t, B_t), and output (A_t, B_t) as the cut action. Our strategy is summarized in Algorithm 1.

```
Algorithm 1 $\mathcal{A}^{\leftrightarrow}_{\mathsf{cut}}$.
Input: vertex measure $\boldsymbol{\mu} \in \mathbb{Z}^V_{\geq 0}$, and approximate matching responses $D_1, \ldots, D_t$.
Parameters: step size $\eta > 0$.
Do: The following.
    1. Sample $d = O\big(\log n\big)$ random unit vectors $\mathbf{u}_1, \ldots, \mathbf{u}_d \sim \mathcal{S}^{n-1}$.
    2. Denote $\mathbf{W}_t = \texttt{MMWU}_{\eta,\boldsymbol{\mu}}\big(\mathbf{L}(D_1), \ldots, \mathbf{L}(D_t)\big)$, and compute $\big\{\hat{\mathbf{v}}_i^{(t)}\big\}_{i \in V} \subseteq \mathbb{R}^d$ given by
           $\big(\hat{\mathbf{v}}_i^{(t)}\big)_j = \langle \mathbf{W}_t \mathbf{u}_j, \mathbf{e}_i \rangle \qquad \forall\, i \in V\ j \in [d]$
       using the truncated Taylor series via [9].
    3. Execute $\texttt{round-cut}\big(d, \big\{\hat{\mathbf{v}}_i^{(t)}\big\}_{i \in V}\big)$ to produce the cut $(A_t, B_t)$.
Output: The cut $(A_t, B_t)$.
```

Figure 2: The cut strategy $\mathcal{A}^{\leftrightarrow}_{\mathsf{cut}}$ for symmetric polymatroidal cut functions $\{\delta_h\}_{h \in E}$.

Let us now demonstrate that $\mathcal{A}^{\leftrightarrow}_{\mathsf{cut}}$ is a $\big(\Omega(\log n), O\big(\log^2 n\big)\big)$-good cut strategy.

Lemma 13. *Let $G = (V, E, \mathbf{w}, \boldsymbol{\mu})$ be an input submodular hypergraph equipped with symmetric polymatroidal cut functions $\{\delta_h\}_{h \in E}$ such that $|V| = n$. Then $\mathcal{A}^{\leftrightarrow}_{\mathsf{cut}}$ is $\Big(\Omega(\log n), O\big(\log^2 n\big)\Big)$-good with probability $O(1)$.*

Proof. TOPROVE 4 □

7.2 An $\big(\Omega(\log^2 n), O(\log^3 n)\big)$-good Cut Strategy for Asymmetric Cut Functions

Let us now consider the case where $\{\delta_h\}_{h \in E}$ are asymmetric. We describe our setup for the online linear optimization setting.

[3] Picking $\delta > \frac{1}{10}$ loses a factor 2 in the final bound on $g(n)$ when proving Theorem 12

Given an instance of a $(n, \boldsymbol{\mu})$-generalized cut-matching game, we again fix the set of actions to be $\mathcal{X} = \Delta_{\boldsymbol{\mu}}^{n \times n}$. For round t of the online optimization setting, we now define the feedback matrix to be the directed Laplacian of the approximate matching response D_t: $\mathbf{F}_t = \mathbf{L}_+(D_t)$ so that the losses become

$$\left\langle \mathbf{L}_+(D_t), \mathbf{X} \right\rangle$$

Similar to the symmetric case, our choices of action set and loss functions are guided by the trivial spectral relaxation for the directed ratio-cut of a graph.

Lemma 14. *Given any weighted, directed graph $G = \left(V, E_G, \mathbf{w}^G, \boldsymbol{\mu}\right)$, we have $\frac{\mathsf{opt}}{2} \leq \Psi_G^*$ where opt is the optimal objective value for the following semidefinite program.*

$$\begin{aligned} \min \quad & \left\langle \mathbf{L}_+(G), \mathbf{X} \right\rangle \\ \text{s.t.} \quad & \left\langle \mathbf{L}(K_{\boldsymbol{\mu}}), \mathbf{X} \right\rangle \geq 1 \\ & \mathbf{X} \succeq \mathbf{0} \end{aligned} \qquad (\texttt{dir-cond-sdp})$$

In order to use a regret bound from Theorem 4, we need only address the first two issues presented in the symmetric case:

(1) *Ensuring Small Width*: We need to ensure that the width $\left\|\mathbf{M}^{-1/2}\mathbf{L}_+(D_t)\mathbf{M}^{-1/2}\right\|_{\mathsf{spec}}$ is small given any approximate matching response D_t. Similar to the symmetric case, we show this is bounded by a constant.

(2) *Ensuring Large Loss*: We now need to ensure that $\left\langle \mathbf{L}_+(D_t), \mathbf{X}_t \right\rangle$ is large. A key difference present in the cut strategy for the asymmetric case is that ensuring large loss requires an additional step in rounding algorithm to control for terms in the loss that correspond to the embedding vector lengths.

To compute the matrix exponential update $\texttt{MMWU}_{\eta,\boldsymbol{\mu}}\left(\mathbf{L}_+(D_1), \ldots, \mathbf{L}_+(D_t)\right)$ in nearly-linear time, we can reuse the sketch provided in Section 7.1.3.

7.2.1 Ensuring Small Width

Whe width in the asymmetric case is still bounded by an absolute constant given any directed approximate matching response. We prove the following Lemma in Appendix A.5

Lemma 15. *Suppose $D = \left(A \cup B, E_D, \mathbf{w}^D, \boldsymbol{\mu}\right)$ is an approximate matching response with respect to cut action (A, B). Then $-\mathbf{M} \preceq \mathbf{L}_+(D) \preceq 3 \cdot \mathbf{M}$.*

This lemma implies that the regret guarantee of Theorem 4 only for step-size $0 < \eta < 1$ when the losses are $\left\langle \mathbf{L}_+(D), \mathbf{X} \right\rangle$. We will subsequently choose an η satisfying this.

7.2.2 Ensuring Large Loss

In the asymmetric case, we wish to produce a cut action (A_t, B_t) such that no matter how the matching player $\mathcal{M}$ plays D, the edges of D will be weighted such that the loss

$$\left\langle \mathbf{L}_+(D_t), \mathbf{X} \right\rangle = \sum_{(i,j) \in E_{D_t}} w_{ij}^{D_t} \cdot \left(\|\mathbf{v}_i - \mathbf{v}_j\|_2^2 + \|\mathbf{v}_i\|_2^2 - \|\mathbf{v}_j\|_2^2 \right)$$

is large. The critical distinction from the symmetric case is the presence of the embedding vector lengths.

This is where one might be inclined to require that valid matching responses be Eulerian directed graphs. If the matching response D is Eulerian, then the graph can be split into two parts where the edges are all oriented from A_t to B_t, and from B_t to A_t. One can then cancel the embedding vector lengths in the loss using the fact that the in- and out-degrees are identical for each vertex. This reduces the problem of constructing a good cut strategy back to that present in the symmetric case: one need only round a cut

(A_t, B_t) such that a large amount of flow is always routed between vertex pairs whose embedding vectors are far in ℓ_2^2-distance.

Absent of this trick, one then needs to control the length of the embedding vectors. A priori this seems difficult. To ensure that an approximation algorithm based on cut-matching games runs in nearly-linear time, some low-dimensional projection needs to be applied to the embedding. One then cannot hope to simultaneously control for both the distortion of the projected embedding, and the length of the projected vectors, suggesting that the matching response *must* be an Eulerian graph. We remark that if one tries to directly adapt the potential analysis of [36] or [54], then the same terms in the loss appear in lower bounding the potential progress made when the matching player routes a large flow. This might suggest why current works generalizing the cut-matching game to directed settings [46, 40] require the matching player to play Eulerian graphs.

However, if one is cognizant of the underlying SDP being used to certify lower bounds on the directed conductance of the state graph, then it is natural to consider rounding algorithms which directly target the SDP. Indeed, applying the rounding algorithm present in [2] (Algorithm 4) augments a σ-separated cut (S, T) into a cut action (A, B) such that any matching response across (A, B) must have large loss. We describe this rounding procedure in Algorithm 2 `directed-round-cut`.

```
Algorithm 2 directed-round-cut.
Input: A cut $(S, T)$ of $V$, and vector embedding $\{\mathbf{v}_i\}_{i \in V} \subseteq \mathbb{R}^d$.

    1. Find $r > 0$ such that $\boldsymbol{\mu}\big(A \cap \mathcal{B}_r(0)\big) \geq \frac{\boldsymbol{\mu}(S)}{2}$ and $\boldsymbol{\mu}\big(A \cap \overline{\mathcal{B}_r(0)}\big) \geq \frac{\boldsymbol{\mu}(S)}{2}$.
    2. Define the following sets.
           $S^+ \stackrel{\text{def}}{=} \left\{i \in S : \|\mathbf{v}_i\|_2^2 \leq r^2\right\}$    $S^- \stackrel{\text{def}}{=} \left\{i \in S : \|\mathbf{v}_i\|_2^2 \geq r^2\right\}$
           $T^+ \stackrel{\text{def}}{=} \left\{i \in T : \|\mathbf{v}_i\|_2^2 \leq r^2\right\}$    $T^- \stackrel{\text{def}}{=} \left\{i \in T : \|\mathbf{v}_i\|_2^2 \geq r^2\right\}$
    3. If $\boldsymbol{\mu}(T^+) \leq \boldsymbol{\mu}(T^-)$, then let $A = S^+$ and $B = T^-$. Else, let $A = T^+$ and $B = S^-$.

Output: the cut $(A, B)$
```

The intuition for `directed-round-cut` is this: one does not need to route the flow bidirectionally across (S, T), and subsequently produce an Eulerian demand graph. The SDP solution, and specifically the position of the median embedding vector, should dictate which direction flow needs to be routed in a given round the cut-matching game. We now prove our lower bound on the loss

Lemma 16. *Let $\boldsymbol{\mu} \in \mathbb{Z}_{\geq 0}^V$, $\mathbf{X} \in \Delta_{\boldsymbol{\mu}}^{n \times n}$, and $\{\mathbf{v}_i\}_{i \in V} \subseteq \mathbb{R}^d$ be a vector embedding whose Gram matrix is $\mathbf{X}$. For $\sigma > 0$, suppose* `directed-round-cut` *is given (S, T) that are σ-separated with respect to $\boldsymbol{\mu}$ and $\{\mathbf{v}_i\}_{i \in V}$ as input, and outputs (A, B). Then, any approximate directed matching response $D = \left(A \cup B, E_D, \mathbf{w}^D\right)$ with respect to A, B satisfies the following:*

$$\sum_{(i,j) \in E_D} w_{ij}^D \cdot \left(\|\mathbf{v}_i - \mathbf{v}_j\|_2^2 + \|\mathbf{v}_i\|_2^2 - \|\mathbf{v}_j\|_2^2\right) \geq \frac{\sigma}{8}$$

Proof. TOPROVE 5 □

We also bound the running time of `directed-round-cut`.

Lemma 17. `directed-round-cut` *runs in time $O(nd + n \log n)$ for $n = |V|$.*

Proof. TOPROVE 6 □

7.2.3 Completing the Analysis for Asymmetric Cut Functions

We are now ready to give our good cut strategy

```
Algorithm 3 $\mathcal{A}^{\rightarrow}_{\mathsf{cut}}$.
Input: vertex measure $\boldsymbol{\mu} \in \mathbb{Z}^V_{\geq 0}$, and directed approximate matching responses $D_1, \ldots, D_t$.
Parameters: step size $\eta > 0$.
Do: The following.
    1. Sample $d = O\big(\log n\big)$ random unit vectors $\mathbf{u}_1, \ldots, \mathbf{u}_d \sim \mathcal{S}^{n-1}$.
    2. Denote $\mathbf{W}_t = \texttt{MMWU}_{\eta,\boldsymbol{\mu}}\big(\mathbf{L}_+(D_1), \ldots, \mathbf{L}_+(D_t)\big)$, and compute $\big\{\hat{\mathbf{v}}_i^{(t)}\big\}_{i\in V} \subseteq \mathbb{R}^d$ given by
           $\big(\hat{\mathbf{v}}_i^{(t)}\big)_j = \langle \mathbf{W}_t \mathbf{u}_j, \mathbf{e}_i \rangle \qquad \forall\, i \in V\ j \in [d]$
       using the truncated Taylor series via [9].
    3. Execute round-cut$\big(d, \big\{\hat{\mathbf{v}}_i^{(t)}\big\}_{i\in V}\big)$ to produce the cut $(S, T)$
    4. Execute directed-round-cut$\big((S,T), \big\{\hat{\mathbf{v}}_i^{(t)}\big\}_{i\in V}\big)$ to produce the cut $(A_t, B_t)$.
Output: The cut $(A_t, B_t)$.
```

Figure 3: The cut strategy $\mathcal{A}^{\rightarrow}_{\mathsf{cut}}$ for asymmetric polymatroidal cut functions $\{\delta_h\}_{h\in E}$.

Lemma 18. *Let $G = (V, E, \mathbf{w}, \boldsymbol{\mu})$ be an input submodular hypergraph equipped with asymmetric polymatroidal cut functions $\{\delta_h\}_{h\in E}$ such that $|V| = n$. Then $\mathcal{A}^{\rightarrow}_{\mathsf{cut}}$ is $\Big(\Omega\big(\log^2 n\big), O\big(\log^3 n\big)\Big)$-good with probability $O(1)$.*

Proof. TOPROVE 7 □

7.3 Completing the Cut Strategy: Proof of Theorem 12

For the sake of completion, let us now prove Theorem 12.

Theorem 12. *Let $H = \big(V, E_H, \mathbf{w}^H, \boldsymbol{\mu}\big)$ be the state graph for an $(n, \boldsymbol{\mu})$-generalized cut-matching game. There exists a cut strategy $\mathcal{A}_{\mathsf{cut}}$ satisfying the following:*

1. *If H is undirected, then $\mathcal{A}_{\mathsf{cut}}$ is $\big(\Omega(\log n), O(\log^2 n)\big)$-good with probability $O(1)$.*
2. *If H is directed, then $\mathcal{A}_{\mathsf{cut}}$ is $\big(\Omega(\log^2 n), O(\log^3 n)\big)$-good with probability $O(1)$.*

Furthermore, $\mathcal{A}_{\mathsf{cut}}$ outputs a cut action in time $\tilde{O}\big(\mathsf{sp}(H)\big)$.

Proof. TOPROVE 8 □

8 Constructing Separated Sets: Proof of Theorem 3

This section is devoted to proving Theorem 3 on the construction of separated sets. We will show that the algorithm `round-cut` in Figure 4 satisfies the requirements of the theorem. We denote by $V = [n]$ the index set for the vector embedding $\{\mathbf{v}_i\}_{i\in V} \subseteq \mathbb{R}^d$ in the theorem statement and define the following notions of expected vector and total variance of the embedding:

$$\begin{aligned}\mathbf{v}_{\mathsf{avg}} &\stackrel{\text{def}}{=} \frac{1}{\mu(V)}\sum_{i\in V}\mu_i\mathbf{v}_i\\ \operatorname*{Var}_{\boldsymbol{\mu}}(\mathbf{v}_i) &\stackrel{\text{def}}{=} \frac{1}{\mu(V)}\sum_{i\in V}\mu_i\cdot\|\mathbf{v}_i-\mathbf{v}_{\mathsf{avg}}\|_2^2 = \frac{1}{(\mu(V))^2}\sum_{\{i,j\}\in\binom{V}{2}}\mu_i\mu_j\|\mathbf{v}_i-\mathbf{v}_j\|^2.\end{aligned}$$

To simplify our proof, we will assume, without loss of generality, that our embedding is both *centered*, i.e., $\mathbf{v}_{\mathsf{avg}} = 0$, and *normalized*, i.e., $\mathrm{Var}_{\boldsymbol{\mu}}(\mathbf{v}_i) = 1$. Under these assumptions, we will make use of the following lemma, which relates the contribution to the total variance of a set $\bar{S}$ to the that of its complement S.

Lemma 19 (Fact 5.19 from [53]). *Let $\{\mathbf{v}_i\}_{i\in V}$ be a centered and normalized vector embedding. We have:*

$$\sum_{i\in\overline{S}}\mu_i\cdot\|\mathbf{v}_i\|_2^2 \geq \mu(S)\cdot\left(1-\frac{1}{(\mu(V))^2}\cdot\sum_{i,j\in\binom{S}{2}}\mu_i\mu_j\cdot\|\mathbf{v}_i-\mathbf{v}_j\|_2^2\right).$$

Our algorithm handles two cases, depending on whether there exist a set of small measure that contributes most of the variance of the embedding or not. To distinguish these two cases, we consider long vectors, i.e. vectors with large contributions to the variance of the embedding, and in particular we define the set $R_t \subseteq V$ of vertices with long vector embeddings:

$$R_t \stackrel{\text{def}}{=} \left\{i\in V : \|\mathbf{v}_i\|_2^2 \leq t\right\}$$

We then say that an embedding is (t,b)-balanced if the variance due to vertices in R_t captures a b-fraction of the total embedding variance, as per the following definition.

Definition 14 ((t,b)-Balanced Embedding). Given $t > 0$, and $b > 0$, a centered normalized vector embedding $\{\mathbf{v}_i \in \mathbb{R}^d\}_{i\in V}$ is (t,b)-*balanced* if the set R_t satisfies:

$$\frac{1}{(\mu(V))^2}\cdot\sum_{i,j\in\binom{R_t}{2}}\boldsymbol{\mu}(i)\boldsymbol{\mu}(j)\cdot\|\mathbf{v}_i-\mathbf{v}_j\|_2^2 \geq b\,.$$

We distinguish the two cases based on whether the vector embedding is $(3, 1/10)$-balanced. We remark that, given t and b, one can check if the embedding is (t,b)-balanced in $O(nd)$ time by computing distances to the average vector of R_t. We start by describing the unbalanced case, which is where our main technical contribution lies.

The unbalanced case In the unbalanced case, the `round-cut` algorithm sorts the embedding vectors according to their length and outputs separated sets S and T from the set of threshold cuts in this ordering. In particular, the set T is always chosen to equal $R_{3/2}$, while the set S is chosen by sweeping through all cuts in the ordering to find one that satisfies the required separation guarantee. We show that such a cut always exists. The following lemma, which is implicit in previous works [10, 2], follows from Markov's Inequality and Lemma 19. We prove it for completeness in Appendix A.

Lemma 20. *Let $\{\mathbf{v}_i \in \mathbb{R}^d\}_{i\in V}$ be a centered normalized vector embedding that is not $(3, 1/10)$-balanced. Then:*

$$\mu(R_{3/2}) \geq \frac{1}{3}\cdot\mu(V) \;\; \textit{and} \;\; \sum_{i\in\overline{R_3}}\mu_i\cdot\|\mathbf{v}_i\|^2 \geq \frac{3}{5}\cdot\mu(V).$$

```
Algorithm 4 round-cut.
Input: A centered normalized vector embedding $\{\mathbf{v}_i \in \mathbb{R}^d\}_{i\in V}$.

  1. If $\{\mathbf{v}_i\}_{i\in V} \subseteq \mathbb{R}^d$ is $\left(3, \frac{1}{10}\right)$-balanced:
       • Sample $\mathbf{g}$ uniformly at random from $\mathcal{S}^{d-1}$ and compute $r_i = \sqrt{d} \cdot \langle \mathbf{g}, \mathbf{v}_i \rangle$ for each $i \in V$.
       • Sort $r_i$. Assume without loss of generality that $r_1 \geq \ldots \geq r_n$ and define sweep cuts
             $S_t^{up} = \{i \in V : r_i \geq r_t\}$ and $S_t^{down} = \{i \in V : r_i \leq r_t\}$
       • Let $t_{start} = \min\{t : 3 \cdot \boldsymbol{\mu}(S_t^{up}) \geq \boldsymbol{\mu}(V)\}$ and $t_{end} = \max\{t : 3 \cdot \boldsymbol{\mu}(S_t^{down}) \geq \boldsymbol{\mu}(V)\}$
       • Output $\left(S_{t_{start}}^{up}, S_{t_{end}}^{down}\right)$.
  2. If $\{\mathbf{v}_i\}_{i\in V} \subseteq \mathbb{R}^d$ is not $\left(3, \frac{1}{10}\right)$-balanced:
       • Sort the vectors by their length. Relabel vertices so that $\|\mathbf{v}_1\| \geq \|\mathbf{v}_2\| \geq \ldots \geq \|\mathbf{v}_n\|$.
       • Let $T = \{i \in V : \|\mathbf{v}_i\|^2 \leq \frac{3}{2}\}$.
       • Define sweep cuts $S_t \stackrel{\text{def}}{=} \{i \in V : \|\mathbf{v}_i\| \geq \|\mathbf{v}_t\|\}$. For each sweep cut $S_t$ with $\|\mathbf{v}_t\| \geq 3$:
           – if the following condition holds(see Equation 17 and Equation 18),
                 $\frac{\boldsymbol{\mu}(S_t)}{\boldsymbol{\mu}(V)} \cdot \frac{\|\mathbf{v}_t\|^2}{8} \geq \frac{3}{100\log(\boldsymbol{\mu}(V))}$,
             output $\frac{1}{100\log(\boldsymbol{\mu}(V))}$-separated sets $(S_t, T)$.
```

Figure 4: The round-cut algorithm.

Setting $T = R_{3/2}$, it remains to that there exists some $t \geq 3$ such that all pairs $i \in \overline{R_t}$, $j \in T$ satisfy

$$\|\mathbf{v}_i - \mathbf{v}_j\|_2^2 \geq \sigma \cdot \frac{3\mu(V)}{\mu(\overline{R_t})}, \tag{17}$$

where $\sigma = 1/100\log(\mu(V)) = \Omega(1/\log n)$. For sake of contradiction, assume that no such t exists. Then, for every $t \geq 3$, there exists a pair $i \in \overline{R_t}, j \in T$ such that the squared distance between $\mathbf{v}_i$ and $\mathbf{v}_j$ is less than $\sigma \cdot 3\mu(V)/\mu(\overline{R_t})$. At the same time, we can lower bound such squared distance as follows:

$$\|\mathbf{v}_i - \mathbf{v}_j\|^2 \geq (\|\mathbf{v}_i\| - \|\mathbf{v}_j\|)^2 \geq (\sqrt{t} - \sqrt{3/2})^2 \geq \frac{t}{8}\,. \tag{18}$$

This yields, for all $t \geq 3$:

$$\mu(\overline{R_t}) \leq \frac{24\sigma}{t} \cdot \mu(V).$$

Integrating over $t \in [3, M]$ where $M \stackrel{\text{def}}{=} \max_{i\in V, \mu_i > 0} \|\mathbf{v}_i\|^2$, we have:

$$\sum_{i\in\overline{R_3}} \mu_i \|\mathbf{v}_i\|^2 = \int_3^M \mu(\overline{R_t})\, dt \,+\, 3 \cdot \mu(\overline{R_3}) \leq 24\sigma \cdot \mu(V) \cdot \int_3^M \frac{dt}{t} \,+ 8\sigma \cdot \mu(V) \leq (24\log(M) + 8) \cdot \sigma\mu(V).$$

At the same time, Lemma 20 lower bounds the right hand side:

$$\frac{3}{5} \cdot \mu(V) \leq \sum_{i \in \overline{R_3}} \mu_i \|\mathbf{v}_i\|^2 \leq 32 \cdot \log(\mu(V)) \cdot \sigma \cdot \mu(V)\,,$$

which implies $\sigma \geq {}^{1}/_{54\log(\mu(V))}$, yielding the required contradiction.

On the right hand side, by the normalization assumption and the fact that $\boldsymbol{\mu} \in \mathbb{Z}^n_{\geq 0}$, we have $M \leq \sum_{i \in V} \mu_i \|\mathbf{v}_i\|^2 \leq \mu(V) = \mathsf{poly}(n)$. This completes the proof that a set $\overline{R_t}$ with the required separation guarantee exists. To find such cut, the `round-cut` algorithm avoids computing the minimum distance between $\overline{R_t}$ and T for each threshold cut by noticing that the guarantee holds if we replace such minimum distance with the quantity ${}^{t}/_{8}$, as we did in Equation 18.

The balanced case In the balanced case, the set R_3 of short vectors captures a constant fraction of the vector embedding's variance and a large fraction of the total volume $\boldsymbol{\mu}(V)$ via Markov's inequality. One can subsequently expect that random projection coupled with a search for a balanced sweep cut will work as a bulk of the vectors will be well separated from one another. The balanced case analysis is thus very similar to the analysis of SDP-based approximation algorithms for balanced separator [53]. In fact, it is simpler as one does not need to bound the conductance of such a cut and so we leave the proof of the following lemma to Appendix A.6.

Lemma 21. *Given $V = [n]$, let $\{\mathbf{v}_i \in \mathbb{R}_d\}_{i \in V} \subseteq \mathbb{R}^d$ be a centered normalized vector embedding. If $\{\mathbf{v}_i\}_{i \in V}$ is a $\left(3, \frac{1}{10}\right)$-balanced embedding, then algorithm* `round-cut` *outputs a pair (S, T) of $\Omega\left(\frac{1}{\log n}\right)$-separated sets with high probability. m*

Running Time To bound the running time, consider the following. Checking if the embedding is $(3, {}^{1}/_{10})$-balanced takes $O(nd)$ time. This can seen by noting that the inequality in Definition 14 can be rewritten as:

$$\frac{1}{\mu(R_t)} \sum_{i \in R_t} \mu(i) \left\| \mathbf{v}_i - \frac{1}{\mu(R_t)} \sum_{j \in R_t} \mu(j) \mathbf{v}_j \right\|_2^2 \geq b.$$

Computing ℓ_2^2-distances and vector projections takes $O(d)$ time per vector, of which there are n. Sorting takes $O(n \log n)$ time. There are at most $O(n)$ sweep cuts to check in both the balanced and unbalanced case, and finding the appropriate cut in both cases takes $O(1)$ time per cut. Checking the condition:

$$\frac{\mu(S_t)}{\mu(V)} \cdot \frac{\|\mathbf{v}_t\|^2}{8} \geq \frac{3}{100\log(\mu(V))},$$

requires time $O(d)$ per cut. Hence the overall running time is at most $O(nd + n\log n)$.

This completes the proof of Theorem 3.

9 Acknowledgements

AC is supported by NSF DGE 2140001. Part of this work was also done while AC was a visiting student at Bocconi University. LO is supported by NSF CAREER 1943510. The authors would like to thank anonymous reviewers for suggesting related work on polymatroidal networks, Thatchaphol Saranurak for suggesting related work on the use of cut-matching games beyond partitioning graphs, and Jeffrey Negrea for assisting on topics related to online learning. AC would like to thank Luca Trevisan for discussion regarding HDX construction, flow embedding techniques for certifying expansion of random regular graphs, and support while visiting Bocconi.

References

[1] Dimitris Achlioptas. Database-friendly random projections. In *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 274–281, 2001.

[2] Amit Agarwal, Moses Charikar, Konstantin Makarychev, and Yury Makarychev. $\mathcal{O}(\sqrt{\log n})$-approximation algorithms for Min UnCut, Min 2CNF Deletion, and directed cut problems. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 573–581, 2005.

[3] Sameer Agarwal, Kristin Branson, and Serge Belongie. Higher order learning with graphs. In *Proceedings of the 23rd international conference on machine learning*, pages 17–24. Association for Computing Machinery, 2006.

[4] Ravindra K Ahujia, Thomas L Magnanti, and James B Orlin. *Network Flows: Theory, Algorithms and Applications*. Prentice-Hall, 1993.

[5] Zeyuan Allen-Zhu, Zhenyu Liao, and Lorenzo Orecchia. Spectral sparsification and regret minimization beyond matrix multiplicative updates. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 237–245, 2015.

[6] Konstantinos Ameranis, Antares Chen, Lorenzo Orecchia, and Erasmo Tani. Efficient flow-based approximation algorithms for submodular hypergraph partitioning via a generalized cut-matching game. *arXiv preprint arXiv:2301.08920*, 2023.

[7] Reid Andersen and Kevin J. Lang. An algorithm for improving graph partitions. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '08, page 651–660, USA, 2008. Society for Industrial and Applied Mathematics.

[8] Sanjeev Arora, Elad Hazan, and Satyen Kale. $O(\sqrt{\log n})$-Approximation to SPARSEST CUT in $\tilde{O}(n^2)$ Time. *SIAM Journal on Computing*, 39(5):1748–1771, January 2010.

[9] Sanjeev Arora and Satyen Kale. A combinatorial, primal-dual approach to semidefinite programs. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 227–236, 2007.

[10] Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM (JACM)*, 56(2):1–37, 2009.

[11] Francis Bach et al. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends in Machine Learning*, 6(2-3):145–373, 2013.

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

[13] Aaron Bernstein, Maximilian Probst Gutenberg, and Thatchaphol Saranurak. Deterministic decremental sssp and approximate min-cost flow in almost-linear time. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1000–1008. IEEE, 2022.

[14] Avrim Blum, John Hopcroft, and Ravindran Kannan. *Foundations of data science*. Cambridge University Press, 2020.

[15] Umit V Catalyurek and Cevdet Aykanat. Hypergraph-partitioning-based decomposition for parallel sparse-matrix vector multiplication. *IEEE Transactions on parallel and distributed systems*, 10(7):673–693, 1999.

[16] T-H Hubert Chan, Anand Louis, Zhihao Gavin Tang, and Chenzi Zhang. Spectral properties of hypergraph laplacian and approximation algorithms. *Journal of the ACM (JACM)*, 65(3):1–48, 2018.

[17] Yuk Hei Chan and Lap Chi Lau. On linear and semidefinite programming relaxations for hypergraph matching. *Mathematical programming*, 135(1-2):123–148, 2012.

[18] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Directed metrics and directed graph partitioning problems. In *SODA*, volume 6, pages 51–60, 2006.

[19] Chandra Chekuri, Sreeram Kannan, Adnan Raja, and Pramod Viswanath. Multicommodity flows and cuts in polymatroidal networks. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 399–408, 2012.

[20] Li Chen, Rasmus Kyng, Yang P Liu, Richard Peng, Maximilian Probst Gutenberg, and Sushant Sachdeva. Maximum flow and minimum-cost flow in almost-linear time. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 612–623. IEEE, 2022.

[21] Julia Chuzhoy. A Distanced Matching Game, Decremental APSP in Expanders, and Faster Deterministic Algorithms for Graph Cut Problems. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Proceedings, pages 2122–2213. Society for Industrial and Applied Mathematics, January 2023.

[22] Julia Chuzhoy, Yu Gao, Jason Li, Danupon Nanongkai, Richard Peng, and Thatchaphol Saranurak. A deterministic algorithm for balanced cut with applications to dynamic connectivity, flows, and beyond. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1158–1167. IEEE, 2020.

[23] Julia Chuzhoy and Sanjeev Khanna. A new algorithm for decremental single-source shortest paths with applications to vertex-capacitated flow and cut problems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 389–400, 2019.

[24] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of johnson and lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.

[25] Karen D Devine, Erik G Boman, Robert T Heaphy, Rob H Bisseling, and Umit V Catalyurek. Parallel hypergraph partitioning for scientific computing. In *Proceedings 20th IEEE International Parallel & Distributed Processing Symposium*, pages 10–pp. IEEE, 2006.

[26] Alina Ene and Huy Nguyen. Random coordinate descent methods for minimizing decomposable submodular functions. In *International Conference on Machine Learning*, pages 787–795. PMLR, 2015.

[27] Alina Ene, Huy Nguyen, and László A Végh. Decomposable submodular function minimization: discrete and continuous. *Advances in neural information processing systems*, 30, 2017.

[28] Song Feng, Emily Heath, Brett Jefferson, Cliff Joslyn, Henry Kvinge, Hugh D Mitchell, Brenda Praggastis, Amie J Eisfeld, Amy C Sims, Larissa B Thackray, et al. Hypergraph models of biological networks to identify genes critical to pathogenic viral response. *BMC bioinformatics*, 22(1):1–21, 2021.

[29] Elad Hazan, Satyen Kale, and Shai Shalev-Shwartz. Near-optimal algorithms for online matrix prediction. In *Conference on Learning Theory*, pages 38–1. JMLR Workshop and Conference Proceedings, 2012.

[30] Shohei Hidaka and Masafumi Oizumi. Fast and exact search for the partition with minimal information loss. *PLoS One*, 13(9):e0201126, 2018.

[31] Arun Jambulapati, Yin Tat Lee, Jerry Li, Swati Padmanabhan, and Kevin Tian. Positive semidefinite programming: mixed, parallel, and width-independent. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 789–802, 2020.

[32] Haotian Jiang, Tarun Kathuria, Yin Tat Lee, Swati Padmanabhan, and Zhao Song. A faster interior point method for semidefinite programming. In *2020 IEEE 61st annual symposium on foundations of computer science (FOCS)*, pages 910–918. IEEE, 2020.

[33] William Johnson and J. Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Conference in Modern Analysis and Probability*, 26:189–206, 01 1982.

[34] Satyen Kale. *Efficient algorithms using the multiplicative weights update method.* Princeton University, 2007.

[35] Rohit Khandekar, Subhash Khot, Lorenzo Orecchia, and Nisheeth K Vishnoi. On a cut-matching game for the sparsest cut problem. *Univ. California, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2007-177*, 6(7):12, 2007.

[36] Rohit Khandekar, Satish Rao, and Umesh Vazirani. Graph partitioning using single commodity flows. *Journal of the ACM (JACM)*, 56(4):1–15, 2009.

[37] Steffen Klamt, Utz-Uwe Haus, and Fabian Theis. Hypergraphs and cellular networks. *PLoS computational biology*, 5(5):e1000385, 2009.

[38] Tsz Chiu Kwok, Lap Chi Lau, and Kam Chuen Tung. Cheeger inequalities for vertex expansion and reweighted eigenvalues. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 366–377. IEEE, 2022.

[39] Lap Chi Lau, Kam Chuen Tung, and Robert Wang. Cheeger inequalities for directed graphs and hypergraphs using reweighted eigenvalues. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1834–1847, 2023.

[40] Lap Chi Lau, Kam Chuen Tung, and Robert Wang. Fast algorithms for directed graph partitioning using flows and reweighted eigenvalues. *arXiv preprint arXiv:2306.09128*, 2023.

[41] Eugene L Lawler and Charles U Martel. Computing maximal "polymatroidal" network flows. *Mathematics of Operations Research*, 7(3):334–347, 1982.

[42] Tom Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM (JACM)*, 46(6):787–832, 1999.

[43] Pan Li and Olgica Milenkovic. Inhomogeneous hypergraph clustering with applications. *Advances in neural information processing systems*, 30, 2017.

[44] Pan Li and Olgica Milenkovic. Submodular hypergraphs: p-laplacians, cheeger inequalities and spectral clustering. In *International Conference on Machine Learning*, pages 3014–3023. PMLR, 2018.

[45] Yaowei Long and Thatchaphol Saranurak. Near-optimal deterministic vertex-failure connectivity oracles. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1002–1010. IEEE, 2022.

[46] Anand Louis. Cut-matching games on directed graphs. *arXiv preprint arXiv:1010.1047*, 2010.

[47] Anand Louis. Hypergraph markov operators, eigenvalues and approximation algorithms. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 713–722, 2015.

[48] Anand Louis and Yury Makarychev. Approximation algorithms for hypergraph small-set expansion and small-set vertex expansion. *Theory of Computing*, 12(1):1–25, 2016.

[49] Danupon Nanongkai and Thatchaphol Saranurak. Dynamic spanning forest with worst-case update time: adaptive, las vegas, and $\mathcal{O}(n^{1/2-\varepsilon})$-time. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1122–1129, 2017.

[50] Mukund Narasimhan, Nebojsa Jojic, and Jeff A Bilmes. Q-clustering. *Advances in Neural Information Processing Systems*, 18, 2005.

[51] Lorenzo Orecchia. *Fast Approximation Algorithms for Graph Partitioning using Spectral and Semidefinite-Programming Techniques.* PhD thesis, EECS Department, University of California, Berkeley, May 2011.

[52] Lorenzo Orecchia, Konstantinos Ameranis, Charalampos Tsourakakis, and Kunal Talwar. Practical almost-linear-time approximation algorithms for hybrid and overlapping graph clustering. In *International Conference on Machine Learning*, pages 17071–17093. PMLR, 2022.

[53] Lorenzo Orecchia, Sushant Sachdeva, and Nisheeth K Vishnoi. Approximating the exponential, the lanczos method and an $\tilde{\wr}(m)$-time spectral algorithm for balanced separator. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 1141–1160, 2012.

[54] Lorenzo Orecchia, Leonard J Schulman, Umesh V Vazirani, and Nisheeth K Vishnoi. On partitioning graphs via single commodity flows. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 461–470, 2008.

[55] Jonah Sherman. Breaking the multicommodity flow barrier for $\mathcal{O}(\sqrt{\log n})$-approximations to sparsest cut. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 363–372. IEEE, 2009.

[56] Zoya Svitkina and Lisa Fleischer. Submodular approximation: Sampling-based algorithms and lower bounds. *SIAM Journal on Computing*, 40(6):1715–1737, 2011.

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

[58] Nate Veldt. Cut-matching games for generalized hypergraph ratio cuts. In *Proceedings of the ACM Web Conference 2023*, WWW '23, page 694–704, New York, NY, USA, 2023. Association for Computing Machinery.

[59] Nate Veldt, Austin R Benson, and Jon Kleinberg. Approximate decomposable submodular function minimization for cardinality-based components. *Advances in Neural Information Processing Systems*, 34:3744–3756, 2021.

[60] Nate Veldt, Austin R Benson, and Jon Kleinberg. Hypergraph cuts with general splitting functions. *SIAM Review*, 64(3):650–685, 2022.

[61] Wenyin Yang, Guojun Wang, Md Zakirul Alam Bhuiyan, and Kim-Kwang Raymond Choo. Hypergraph partitioning for social networks based on information entropy modularity. *Journal of Network and Computer Applications*, 86:59–71, 2017.

[62] Yuichi Yoshida. Cheeger inequalities for submodular transformations. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2582–2601. SIAM, 2019.

A Omitted Proofs

In this section, we prove some technical lemmata used in the body of the paper.

A.1 Our Results

Lemma 1. *Given a weighted hypergraph $G^* = (V, E, \boldsymbol{\mu}, \mathbf{w})$, we have $\Psi^*_G = \bar{\Psi}^*_G$. Furthermore, there exists an algorithm that, given any $\mathbf{x} \in \mathbb{R}^V$, recovers a cut $S \subseteq V$ satisfying*

$$\Psi_G(S) \leq \bar{\Psi}_G(\mathbf{x})$$

in time $O\big(|V| \log |V| + \mathsf{sp}(G)\big)$, where the sparsity of G is defined as $\mathsf{sp}(G) \overset{\text{def}}{=} \sum_{h \in E} |h|$.

Proof. TOPROVE 9 □

A.2 Polymatroidal Cut Functions and Hypergraph Flows

Fact 2. *Let V be an arbitrary finite set and $F, G : 2^V \to \mathbb{R}$ be monotone non-decreasing submodular functions. If $\delta : 2^V \to \mathbb{R}$ is the function defined by:*

$$\delta(S) \overset{\text{def}}{=} \min\{F(S), G(V \setminus S)\},$$

for all $S \subseteq V$, then δ is submodular.

Proof. TOPROVE 10 □

Fact 3. *Given a polymatroidal cut function δ_h, we have, for all $\mathbf{x} \in \mathbb{R}^h$:*

$$\overline{\delta}_h(\mathbf{x}) = \min_{\nu \in \mathbb{R}} \bar{F}^-_h((\mathbf{x} - \nu \mathbf{1}_h)_+) + \bar{F}^+_h((\mathbf{x} - \nu \mathbf{1}_h)_-).$$

Proof. TOPROVE 11 □

Fact 5. *Given a symmetric polymatroidal cut function δ_h with $F_h \overset{\text{def}}{=} F^+_h = F^-_h$, we have, for all $\mathbf{x} \in \mathbb{R}^h$:*

$$\min_{\nu \in \mathbb{R}} \bar{F}_h(|\mathbf{x} - \nu \mathbf{1}_h|) \leq \overline{\delta}_h(\mathbf{x}) \leq 2 \cdot \min_{\nu \in \mathbb{R}} \bar{F}_h(|\mathbf{x} - \nu \mathbf{1}_h|).$$

Proof. TOPROVE 12 □

Theorem 5. *Let $G = \big(V, E_G, \mathbf{w}^G, \boldsymbol{\mu}\big)$ be a weighted hypergraph equipped with a collection of polymatroidal cut functions $\{\delta_h\}_{h \in E_G}$, and $H = \big(V, E_H, \mathbf{w}^H\big)$ be a weighted directed graph. If $H \preceq_\rho G$, then*

$$\forall \mathbf{x} \in \mathbb{R}^V\,, \sum_{(i,j) \in E_H} w^H_{ij} \cdot (\mathbf{x}_i - \mathbf{x}_j)_+ \leq \rho \cdot \sum_{h \in E_G} w^G_h \cdot \overline{\delta}_h(\mathbf{x}) \quad \text{and} \quad \forall S \subseteq V\,, \delta_H(S) \leq \rho \cdot \delta_G(S).$$

For symmetric polymatroidal cut functions $\{\delta_h\}_{h \in E_G}$, then $H \preceq_\rho G$ implies the following bound on the undirected symmetrization $\tilde{H} = (V, E_{\tilde{H}}, \mathbf{w}^{\tilde{H}})$:

$$\forall \mathbf{x} \in \mathbb{R}^V\,, \sum_{\{i,j\} \in E_{\tilde{H}}} w^{\tilde{H}}_{ij} \cdot |\mathbf{x}_i - \mathbf{x}_j| \leq 2 \cdot \rho \cdot \sum_{h \in E_G} w^G_h \cdot \overline{\delta}_h(\mathbf{x}) \quad \text{and} \quad \forall S \subseteq V\,, \delta_{\tilde{H}(S)} \leq 2 \cdot \rho \cdot \delta_G(S).$$

Proof. TOPROVE 13 □

Theorem 6. *Let $G = \big(V, E_G, \mathbf{w}^G, \boldsymbol{\mu}\big)$ be a weighted hypergraph equipped with a collection of polymatroidal cut functions $\{\delta_h\}_{h \in E_G}$. Given a hypergraph flow $\mathbf{Y}$ over G, there is an algorithm that, in time $\tilde{O}\big(\mathsf{sp}(G)\big)$, computes a weighted directed bipartite graph $H = \big(V, E_h, \mathbf{w}^H\big)$ such that $H \preceq_{\mathrm{cong}_G(\mathbf{Y})} G$ and $|E_H| = \tilde{O}\big(\mathsf{sp}(G)\big)$. Furthermore, all vertices $i \in V$ with non-negative (resp. negative) demand $\mathrm{dem}_i(\mathbf{Y})$ are source nodes (resp. sink nodes) in H, each with out-degree (resp. in-degree) equal $\mathrm{dem}_i(\mathbf{Y})$.*

Proof. TOPROVE 14 □

A.3 An $O(\sqrt{\log n})$-Approximation via ℓ_2^2-Metric Embeddings

Lemma 2. *The* (RC-Metric) *vector program is, up to a constant, a relaxation of the minimum ratio-cut program for the hypergraph* $G = (V, E, \mathbf{w}, \boldsymbol{\mu})$ *with symmetric polymatroidal cut functions* $\{\delta_h\}_{h\in E}$.

Proof. TOPROVE 15 □

Lemma 4. *The* (RC-Directed-Semimetric) *vector program is, up to a constant, a relaxation of the minimum submodular hypergraph ratio-cut problem for the hypergraph* $G = (V, E, \mathbf{w}, \boldsymbol{\mu})$ *with polymatroidal cut functions* $\{\delta_h\}_{h\in E}$.

Proof. TOPROVE 16 □

Theorem 8. *Given a feasible solution* $\{\mathbf{v}_i \in \mathbb{R}^d\}_{i\in V\cup E}$ *to* ***RC-Directed-Semimetric*** *and a measure* $\boldsymbol{\mu} \in \mathbb{R}^V_{\geq 0}$*, there exists a polynomial-time computable map* $\phi : \mathbb{R}^d \to \mathbb{R}$ *such that:*

1. $(\phi(\mathbf{v}) - \phi(\mathbf{w}))_+ \leq \|\mathbf{v} - \mathbf{w}\|^2 + \|\mathbf{v}\|^2 - \|\mathbf{w}\|^2$ *for all* $\mathbf{v}, \mathbf{w} \in \mathbb{R}^d$,
2. *and:*
$$\frac{\sum_{i,j\in V} \mu(i)\mu(j)|\phi(\mathbf{v}_i) - \phi(\mathbf{v}_j)|}{\sum_{i,j\in V} \mu(i)\mu(j)\|\mathbf{v}_i - \mathbf{v}_j\|_2^2} \geq \Omega\left(\frac{1}{\sqrt{\log |V|}}\right)$$

Proof. TOPROVE 17 □

A.4 Localized Convex Formulations and Ratio-Cut Improvement

Lemma 8. *The programs* (CI-Primal) *and* (CI-Dual) *are dual to each other. Strong duality holds.*

Proof. TOPROVE 18 □

Lemma 9. *Under the same assumptions of Definition 12, consider an approximate dual solution* $\mathbf{Y}$ *of value* α *to the ratio-cut improvement problem on* $(G, \mathbf{1}^{A,B})$*. Let* H *be the directed graph obtained by applying the flow-path decomposition algorithm of Theorem 6 to the hypergraph flow* $\mathbf{Y}$*. Then, the scaled graph* $\frac{1}{\alpha} \cdot H$ *is an approximate dual graph certificate of value* α.

Proof. TOPROVE 19 □

A.5 An $O(\log n)$-Approximation via Generalized Cut-Matching Games

Corollary 1. *There exists an* $O(\log n)$*-approximation algorithm for the minimum ratio-cut problem over submodular hypergraphs with polymatroidal cut functions whose running time is dominated by the solution of a polylogarthmic number of decomposable submodular minimization problems over the hypergraph cut function.*

Corollary 2. *There exists an* $O(\log n)$*-approximation algorithm for the minimum ratio-cut problem over submodular hypergraphs equipped with the directed or standard hypergraph cut function whose running time is almost linear in the hypergraph sparsity.*

Proof. TOPROVE 20 □

Lemma 15. *Suppose* $D = \left(A \cup B, E_D, \mathbf{w}^D, \boldsymbol{\mu}\right)$ *is an approximate matching response with respect to cut action* (A, B)*. Then* $-\mathbf{M} \preceq \mathbf{L}_+(D) \preceq 3 \cdot \mathbf{M}$.

Proof. TOPROVE 21 □

Lemma 20. *Let* $\{\mathbf{v}_i \in \mathbb{R}^d\}_{i\in V}$ *be a centered normalized vector embedding that is not* $(3, {}^1\!/_{10})$*-balanced. Then:*
$$\mu(R_{3/2}) \geq \frac{1}{3} \cdot \mu(V) \ \text{ and } \sum_{i\in \overline{R_3}} \mu_i \cdot \|\mathbf{v}_i\|^2 \geq \frac{3}{5} \cdot \mu(V).$$

Proof. TOPROVE 22 □

A.6 Constructing Separated Sets: Proof of Theorem 3

We now complete the argument that `round-cut` outputs separated sets when the given vector embedding is balanced. We begin by recalling a fact about uniformly sampled vectors on the sphere.

Lemma 22. *Given $\mathbf{v} \in \mathbb{R}^d$, and $\mathbf{g}$ a uniformly sampled random vector in $\mathcal{S}^{d-1}$, the following hold.*

1. $\mathbb{E}_{\mathbf{g}} \langle \mathbf{g}, \mathbf{v} \rangle^2 = \frac{\|\mathbf{v}\|^2}{d}$
2. *For any $\delta > \frac{d}{16}$, we have* $\Pr_{\mathbf{g}} \left(\langle \mathbf{g}, \mathbf{v} \rangle^2 \geq \delta \cdot \frac{\|\mathbf{v}\|^2}{d} \right) \leq e^{-\delta/4}$.

We also require a lemma proved in [51] which states that, with constant probability, one can extract two balanced partitions where projection lengths are well separated when projecting along a random direction.

Lemma 23 (Lemma 5.5.6 item (3) of [51])**.** *Given a set V such that $|V| = n$, and a $\left(3, \frac{1}{10}\right)$-balanced embedding $\{\mathbf{v}_i\}_{i \in V} \subseteq \mathbb{R}^d$, let $r_i = \sqrt{d} \cdot \langle \mathbf{g}, \mathbf{v}_i \rangle$ for a $\mathbf{g} \in \mathcal{S}^{d-1}$ and each $i \in V$. Assume, without loss of generality, that $r_1 \geq \ldots \geq r_n$. Then, there exists an absolute constant $r > 0$ such that, with $O(1)$ probability over a uniformly random choice of $\mathbf{g} \in \mathcal{S}^{d-1}$, there exists $1 \leq a < b \leq n$ satisfying the following:*

1. $\mu(\{1, \ldots, a\}) \geq \frac{1}{3} \cdot \mu(V)$,
2. $\mu(\{b, \ldots, n\}) \geq \frac{1}{3} \cdot \mu(V)$,
3. $\left(r_a - r_b\right)^2 \geq r \cdot \mathrm{Var}_\mu(\mathbf{v}_i)$.

Using the above two lemmata, we can now prove that `round-cut` produces a robustly separated set in the balanced case.

Lemma 21. *Given $V = [n]$, let $\{\mathbf{v}_i \in \mathbb{R}_d\}_{i \in V} \subseteq \mathbb{R}^d$ be a centered normalized vector embedding. If $\{\mathbf{v}_i\}_{i \in V}$ is a $\left(3, \frac{1}{10}\right)$-balanced embedding, then algorithm `round-cut` outputs a pair (S, T) of $\Omega\left(\frac{1}{\log n}\right)$-separated sets with high probability. m*

Proof. TOPROVE 23 □

B Comparison to the cut-Improvement of Andersen & Lang

In this section, we briefly show that our formulation of the ratio-cut improvement problem in Definition 10 generalizes the definition of the modified quotient cut score, introduced by Andersen and Lang [7], to potentially non-integral seed vectors (rather than cut seeds) and to arbitrary hypergraphs (rather than graphs). Consider the following seed vector. Let $A \subseteq V$ be a cut, and define the vector $\mathbf{s}^{A,\bar{A}} \in \mathbb{R}^V$ to be

$$\mathbf{s}^{A,\bar{A}} \overset{\text{def}}{=} \mathbf{1}^A - {\mu(A)}/{\mu(\bar{A})} \cdot \mathbf{1}^{\bar{A}} \,. \tag{19}$$

Assuming $\mu(A) \leq \mu(\bar{A})$, note that $\langle \mathbf{s}^{A,\bar{A}}, \mathbf{1} \rangle_\mu = 0$ and $\|\mathbf{s}^{A,\bar{A}}\|_\infty \leq 1$, so that $\mathbf{s}^{A,\bar{A}}$ is a valid seed vector. With this choice of seed vector, the ratio-cut improvement objective on a graph G gives:

$$\min \left\{ \Psi_{G,\mathbf{s}^{A,\bar{A}}}(S), \Psi_{G,\mathbf{s}^{A,\bar{A}}}(\bar{S}) \right\} = \frac{\sum_{e \in E} w_h \cdot \delta_e(S)}{\left|\mu(A \cap S) - \frac{\mu(A)}{\mu(\bar{A})} \cdot \mu(\bar{A} \cap S)\right|},$$

which is exactly equal to the *modified quotient cut score* of cut $(S, \bar{S})$ as considered by Andersen and Lang.

C Proofs of Theorems 9 and 10

Suppose we have an algorithm $\mathcal{A}$ that, on input α, either outputs a cut with a cut $S \subseteq V$ with $\Psi_{G,\mathbf{1}^{A,B}}(S) \leq 2\alpha$ or an approximate dual solution of value α. Then, $\mathcal{A}$ can be used to construct an approximate primal-dual oracle by performing binary search on α and converting the final approximate dual solution to an approximate dual graph certificate via the flow decomposition of Theorem 6. Since the range of possible values for α is $[1/\mu(V), \sum_{h\in E} w_h]$, which we assume is polynomial in $|V|$, it suffices to run $\mathcal{A}$ for $O(\log |V|)$ iterations. Hence, in the rest of the section, we focus on implementing $\mathcal{A}$ for both cases in the theorems.

General Polymatroidal Cut Functions For a fixed $\alpha \geq 0$, we consider the integral version of the feasibility problem for (`CI-Primal`):

$$\exists S \subseteq V : \sum_{h\in E} w_h \delta_h(S) - \alpha \cdot \langle \mathbf{1}_S, \mathbf{1}^{A,B}\rangle_{\boldsymbol{\mu}} \leq 0.$$

The left-hand side of this equation can be minimized exactly by the proximal algorithms for decomposable submodular minimization (DSM) of Ene *et al.* [26, 27] when run with $\epsilon = \frac{1}{n}$ and rounded to integral solutions. If the resulting minimum is less than 0, the minimizer S must have $\Psi_{G,\mathbf{1}^{A,B}}(S) \leq \alpha$, as required. Otherwhise, the DSM algorithm produces a matching dual solution, which takes exactly the form of a feasible solution to (`CI-Dual`) with value α. Applying the best result of Ene *et al.* , achieved by the accelerated coordinated descent method under the assumption that a quadratic optimization oracle is given for each cut function δ_h, the running time to implement an approximate primal-dual oracle becomes $\tilde{O}\big(|V| \cdot \sum_{h\in E} \Theta_h\big)$, where Θ_h is the running time for a quadratic optimization oracle for δ_h.

Graph-Reducible Polymatroidal Cut Functions As we have shown in Section 4, for directed and standard hypergraph cut functions, the congestion constraint in the hyperflow problem `CI-Dual` corresponds to vertex capacities on the hyperedge nodes of the factor graph $\tilde{G}$. To implement the primitive $\mathcal{A}$ described above for a fixed α, we construct a network flow problem from this vertex-capacitated factor graph by adding an auxiliary source vertex s and an auxiliary vertex t. The vertex s is connected to every vertex in $i \in A$ with an arc of capacity $\alpha\mu_i$, while the vertex t is connected to every vertex $j \in B$ with an arc of capacity $\alpha \cdot \mu_j \cdot {}^{\mu(A)}/_{\mu(B)}$. It is easy to see that the capacity of a cut of V in this new network is, for every $S \subseteq V$:

$$\operatorname{cap}(\{s\} \cup S) \overset{\text{def}}{=} \sum_{h\in E} w_h \delta_h(S) - \alpha \langle \mathbf{1}_S, \mathbf{1}^{A,B}\rangle_{\mu} + \alpha \cdot \mu(A) \tag{20}$$

Now, apply a $\frac{1}{2}$-approximate maximum flow solver to this network flow problem to obtain a flow a value F and an s-t cut C of capacity $2F$. We have two cases:

1. If $F \geq \frac{\alpha}{2} \cdot \mu(A)$, then the flow routed is an approximate dual solution of value α, as required.
2. If $F < \frac{\alpha}{2} \cdot \mu(A)$, the s-t cut has capacity strictly less then $\alpha \cdot \mu(A)$. As a result, we have by Equation 20:

$$\sum_{h\in E} w_h \delta_h(S) < \alpha \langle \mathbf{1}_S, \mathbf{1}^{A,B}\rangle_{\mu}.$$

   This implies that $\Psi_{G,\mathbf{1}^{A,B}}(S) \leq \alpha \leq 2\alpha$, are required for $\mathcal{A}$.

The almost-linear time result follows by applying the almost-linear time algorithm of Chen *et al.* [20] as our approximate maximum flow solver. For the standard hypergraph cut function, as the edges in the factor graph are undirected, it suffices to use the almost-linear time solver of Bernstein *et al.* [13] for undirected vertex-capacitated graphs.

D Proof of Theorem 11: the cut-matching game reduction

Theorem 11. *Let $G = (V, E, \mathbf{w}, \boldsymbol{\mu})$ be an input submodular hypergraph equipped with polymatroidal cut functions. Consider an execution of the cut-matching game in which the cut player $\mathcal{C}$ plays an $\big(f(n), g(n)\big)$-good cut strategy, while the matching player plays the dual graph certificates output by an approximate primal-dual oracle $\mathcal{A}_{\mathtt{CI}}$ (Definition 13) on G. Let C_t be the cuts output by $\mathcal{A}_{\mathtt{CI}}$. Then, with constant probability, we have the following approximation result:*

$$\min \left\{ \Psi_G(C_t) \right\}_{t=1}^{g(n)} \leq O\left(\frac{g(n)}{f(n)} \right) \cdot \Psi_G^*.$$

Proof. TOPROVE 24 □