Check for updates

# Hypergraphs with edge-dependent vertex weights: $p$-Laplacians and spectral clustering

Yu Zhu* and Santiago Segarra

Department of Electrical and Computer Engineering, Rice University, Houston, TX, United States

We study $p$-Laplacians and spectral clustering for a recently proposed hypergraph model that incorporates edge-dependent vertex weights (EDVW). These weights can reflect different importance of vertices within a hyperedge, thus conferring the hypergraph model higher expressivity and flexibility. By constructing submodular EDVW-based splitting functions, we convert hypergraphs with EDVW into submodular hypergraphs for which the spectral theory is better developed. In this way, existing concepts and theorems such as $p$-Laplacians and Cheeger inequalities proposed under the submodular hypergraph setting can be directly extended to hypergraphs with EDVW. For submodular hypergraphs with EDVW-based splitting functions, we propose an efficient algorithm to compute the eigenvector associated with the second smallest eigenvalue of the hypergraph 1-Laplacian. We then utilize this eigenvector to cluster the vertices, achieving higher clustering accuracy than traditional spectral clustering based on the 2-Laplacian. More broadly, the proposed algorithm works for all submodular hypergraphs that are graph reducible. Numerical experiments using real-world data demonstrate the effectiveness of combining spectral clustering based on the 1-Laplacian and EDVW.

KEYWORDS

submodular hypergraphs, *p*-Laplacian, spectral clustering, edge-dependent vertex weights, decomposable submodular function minimization

## 1. Introduction

Spectral clustering makes use of eigenvalues and eigenvectors of graph Laplacians to group vertices in a graph. It is one of the most popular clustering methods due to its generality, efficiency, and strong theoretical basis. Standard graph Laplacians were first adopted to obtain relaxations of balanced graph cut criteria (Von Luxburg, 2007). Later, these were generalized to $p$-Laplacians, which are able to provide better approximations of the Cheeger constant (Amghibech, 2003; Bühler and Hein, 2009). Especially, the second smallest eigenvalue of the 1-Laplacian is identical to the Cheeger constant and the partition that achieves the optimal Cheeger cut can be obtained by thresholding the corresponding eigenvector (Szlam and Bresson, 2010).

Graphs are widely used to model pairwise interactions, but in many real-world applications the entities engage in higher-order relationships (Benson et al., 2016; Schaub et al., 2021). For instance, in co-authorship networks multiple authors may interact in writing an article together (Chitra and Raphael, 2019). In an e-commerce system, multiple customers can be associated if they once purchased the same product (Li et al., 2018). In text mining, multiple documents are related to each other if they contain the same

keywords (Hayashi et al., 2020; Zhu et al., 2021). Such multi-way relations can be modeled by hypergraphs, where the notion of an edge is generalized to a hyperedge that can connect more than two vertices.

In graphs, there is only one way to cut an edge, thus, a scalar weight is enough to characterize the cut. But in hypergraphs, there may exist multiple ways to split a hyperedge. Consequently, a splitting function $w_e$ is introduced for each hyperedge $e$ in the hypergraph, assigning a cost to every possible cut of $e$. For any $\mathcal{S} \subseteq e$, $w_e(\mathcal{S})$ indicates the penalty of partitioning $e$ into $\mathcal{S}$ and $e \setminus \mathcal{S}$. In particular, when $w_e$ is submodular for every hyperedge $e$, the corresponding model is termed as a submodular hypergraph which has desirable mathematical properties, making it convenient for theoretical analysis (Li and Milenkovic, 2017, 2018). A series of results in graph spectral theory including $p$-Laplacians, nodal domain theorems, and Cheeger inequalities have been generalized to submodular hypergraphs (Li and Milenkovic, 2018; Yoshida, 2019).

The choice of hyperedge splitting functions has a large practical effect on the hypergraph clustering performance. There are mainly two types of splitting functions in existing work. One is the so-called all-or-nothing splitting function in which an identical penalty is charged if the hyperedge is split regardless of how its vertices are separated (Hein et al., 2013). Another slightly more general type is the class of cardinality-based splitting functions where the splitting penalty depends only on the number of vertices placed on each side of the split (Veldt et al., 2020).

The limitation of existing splitting functions is that they treat all the vertices in a hyperedge equally while in practice these vertices may have different degrees of contribution to the hyperedge. Such information can be captured by edge-dependent vertex weights (EDVW): every vertex $v$ is associated with a weight $\gamma_e(v)$ for each incident hyperedge $e$ that reflects the contribution or importance of $v$ to $e$ (Chitra and Raphael, 2019). Going back to the aforementioned examples, EDVW can be used to model the author positions in a co-authored article, the quantity of a product bought by a customer, as well as the frequency of a word in a document.

Spectral theory on the hypergraph model with EDVW is much less developed than on submodular hypergraphs. Existing works studying hypergraphs with EDVW have only focused on random walk-based Laplacian matrices (Chitra and Raphael, 2019; Hayashi et al., 2020; Zhu et al., 2021), thus raising the question: *How to define non-linear $p$-Laplacians for the hypergraph model that incorporates EDVW?* Our basic idea for solving this problem is to convert a hypergraph with EDVW into a submodular hypergraph then $p$-Laplacians and related theorems developed for submodular hypergraphs can be directly leveraged. Based on our earlier work (Zhu et al., 2022), the model conversion can be achieved by defining submodular EDVW-based splitting functions in the form of $w_e(\mathcal{S}) = g_e(\sum_{v \in \mathcal{S}} \gamma_e(v))$ where $g_e$ is a concave function and the splitting penalty $w_e(\mathcal{S})$ is dependent only on the sum of EDVW in $\mathcal{S}$. Moreover, hypergraphs with such splitting functions are proved to be graph reducible, meaning that there exists some graph sharing the same cut properties (Zhu and Segarra, 2022).

Since the 1-Laplacian provides the tightest approximation of the Cheeger constant, a follow-up question is: *How to apply 1-spectral clustering to submodular hypergraphs with EDVW-based splitting functions?* To this end, we develop an algorithm to compute the second eigenvector of the 1-Laplacian for EDVW-based submodular hypergraphs based on the inverse power method (IPM). The IPM was initially proposed for undirected graphs (Hein and Bühler, 2010), then generalized to submodular hypergraphs with cardinality-based splitting functions (Li and Milenkovic, 2018). A key to the success of IPM is an efficient solution to the inner-loop optimization problem in it. In this paper, we derive an equivalent definition of graph reducibility based on which we further propose an efficient solution to the inner problem that works for all graph reducible submodular hypergraphs including those with EDVW. The proposed solution can also be used to solve submodular function minimization (SFM) problems (Bach, 2013) when the submodular function can be represented as sums of concave functions applied to modular functions.

The major contributions of this paper can be summarized as follows:

(1) We present an equivalent definition of graph reducibility in terms of the Lovász extension of the cut function (see Theorem 1), which is helpful for understanding the relations between graph Laplacians and hypergraph Laplacians.

(2) We propose an algorithm to compute the eigenvector of the 1-Laplacian associated with the second smallest eigenvalue for all graph reducible submodular hypergraphs including those with EDVW-based splitting functions, and use the eigenvector for 1-spectral clustering.

(3) We validate the effectiveness of the proposed algorithm which leverages both of EDVW and 1-spectral clustering *via* numerical experiments on real-world datasets.

## 1.1. Paper outline

The rest of this paper is structured as follows. Preliminary mathematical concepts and submodular hypergraphs are reviewed in section 2. Section 3 introduces the hypergraph model with EDVW and shows how to convert it to graph reducible submodular hypergraphs by constructing EDVW-based splitting functions. The section also presents two equivalent definitions for graph reducibility. The proposed 1-spectral clustering algorithm is described in section 4. Section 5 presents experimental results. The relation between hypergraph Laplacians defined in different ways and the application of the proposed algorithm in SFM are discussed in section 6. Closing remarks are included in section 7.

## 1.2. Notation

For a vector $\mathbf{x}$ and a set $\mathcal{S}$, $\mathbf{x}_{\mathcal{S}}$ denotes the vector formed by the entries of $\mathbf{x}$ indexed by $\mathcal{S}$ and $\mathbf{x}(\mathcal{S}) = \sum_{v \in \mathcal{S}} x_v$. The operator $\mathcal{P}_{a,b}(\mathbf{x})$ projects every entry of $\mathbf{x}$ onto the range $[a, b]$.

Throughout the paper we assume that the considered hypergraphs are connected.

## 2. Preliminaries

### 2.1. Mathematical preliminaries

For a finite set $\mathcal{V}$, a set function $F: 2^{\mathcal{V}} \to \mathbb{R}$ is called submodular if $F(\mathcal{S}_1) + F(\mathcal{S}_2) \geq F(\mathcal{S}_1 \cup \mathcal{S}_2) + F(\mathcal{S}_1 \cap \mathcal{S}_2)$ for every $\mathcal{S}_1, \mathcal{S}_2 \subseteq \mathcal{V}$. Considering a set function $F: 2^{\mathcal{V}} \to \mathbb{R}$ such that $F(\emptyset) = 0$ where $\mathcal{V} = [N] = \{1, 2, \cdots, N\}$, its Lovász extension $f: \mathbb{R}^N \to \mathbb{R}$ is defined as follows. For any $\mathbf{x} \in \mathbb{R}^N$, sort its entries in non-increasing order $x_{i_1} \geq x_{i_2} \geq \cdots \geq x_{i_N}$, where $(i_1, i_2, \cdots, i_N)$ is a permutation of $(1, 2, \cdots, N)$, and set

$$f(\mathbf{x}) = \sum_{j=1}^{N-1} F(\mathcal{S}_j)(x_{i_j} - x_{i_{j+1}}) + F(\mathcal{V})x_{i_N}, \qquad (1)$$

where $\mathcal{S}_j = \{i_1, \cdots, i_j\}$ for $1 \leq j < N$. A set function $F$ is submodular if and only if its Lovász extension $f$ is convex (Lovász, 1983). For any $\mathcal{S} \subseteq \mathcal{V}$, $F(\mathcal{S}) = f(\mathbf{1}_{\mathcal{S}})$ where $\mathbf{1}_{\mathcal{S}}$ is the indicator vector of $\mathcal{S}$. If $F(\mathcal{V}) = 0$, $f(a\mathbf{x} + b\mathbf{1}) = af(\mathbf{x})$ for any $a \in \mathbb{R}_{\geq 0}, b \in \mathbb{R}$. More properties of submodular functions can be found in Appendix A.

### 2.2. Submodular hypergraphs

Let $\mathcal{H} = (\mathcal{V}, \mathcal{E}, \mu, \{w_e\})$ denote a submodular hypergraph (Li and Milenkovic, 2018) where $\mathcal{V} = [N]$ is the vertex set and $\mathcal{E}$ is the set of hyperedges. The function $\mu: \mathcal{V} \to \mathbb{R}_+$ assigns positive weights to vertices. Each hyperedge $e \in \mathcal{E}$ is associated with a submodular splitting function $w_e: 2^e \to \mathbb{R}_{\geq 0}$ that assigns non-negative penalties to every possible split of $e$. Moreover, $w_e$ is required to satisfy $w_e(\emptyset) = 0$ and be symmetric so that $w_e(\mathcal{S}) = w_e(e \setminus \mathcal{S})$ for any $\mathcal{S} \subseteq e$. The domain of $w_e$ can be extended from $2^e$ to $2^{\mathcal{V}}$ by setting $w_e(\mathcal{S}) = w_e(\mathcal{S} \cap e)$ for any $\mathcal{S} \subseteq \mathcal{V}$, guaranteeing that the submodularity is maintained.

A cut is a partition of the vertex set $\mathcal{V}$ into two disjoint, non-empty subsets denoted by $\mathcal{S}$ and its complement $\mathcal{V} \setminus \mathcal{S}$. The weight of the cut is defined as the sum of splitting penalties associated with each hyperedge (Li and Milenkovic, 2018; Veldt et al., 2020), i.e.,

$$\mathrm{cut}_{\mathcal{H}}(\mathcal{S}) = \sum_{e \in \mathcal{E}} w_e(\mathcal{S}). \qquad (2)$$

The normalized Cheeger cut (NCC) is defined as

$$\mathrm{NCC}(\mathcal{S}) = \frac{\mathrm{cut}_{\mathcal{H}}(\mathcal{S})}{\min\{\mathrm{vol}(\mathcal{S}), \mathrm{vol}(\mathcal{V} \setminus \mathcal{S})\}} \qquad (3)$$

where $\mathrm{vol}(\mathcal{S}) = \sum_{v \in \mathcal{S}} \mu(v)$ denotes the volume of $\mathcal{S}$. The 2-way Cheeger constant is defined as the minimum NCC over all non-empty subsets of $\mathcal{V}$ except itself, i.e.,

$$h_2 = \min_{\emptyset \subset \mathcal{S} \subset \mathcal{V}} \mathrm{NCC}(\mathcal{S}). \qquad (4)$$

The solution to (4) provides an optimal partitioning in the sense that we obtain two clusters which are balanced in terms of volume

and loosely connected as captured by a small cut weight. In this paper, we adopt the minimization of NCC as our objective. There exist other clustering measures such as ratio cut, normalized cut and ratio Cheeger cut, which are closely related (Von Luxburg, 2007; Bühler and Hein, 2009).

Optimally solving (4) has been shown to be NP-hard for graphs (Wagner and Wagner, 1993), let alone hypergraphs. In spectral graph theory, Cheeger inequalities are derived to bound and approximate the Cheeger constant using graph $p$-Laplacians (Amghibech, 2003; Bühler and Hein, 2009; Szlam and Bresson, 2010; Chang, 2016; Chang et al., 2017; Tudisco and Hein, 2018). The results have been generalized to submodular hypergraphs (Li and Milenkovic, 2018; Yoshida, 2019). The $p$-Laplacian $\triangle_p$ of a submodular hypergraph is defined as an operator that, for all $\mathbf{x} \in \mathbb{R}^N$, induces

$$\langle \mathbf{x}, \triangle_p(\mathbf{x}) \rangle = \sum_{e \in \mathcal{E}} \vartheta_e f_e(\mathbf{x})^p \triangleq Q_p(\mathbf{x}), \qquad (5)$$

where $\vartheta_e = \max_{\mathcal{S} \subseteq e} w_e(\mathcal{S})$ and $f_e$ is the Lovász extension of the normalized splitting function $\vartheta_e^{-1} w_e$. Notice that $\triangle_p$ can be alternatively defined in terms of the subdifferential of $f_e$ (Li and Milenkovic, 2018), while we keep the definition (5) since it is more instrumental to our development. In particular, when $p = 1$, $Q_1(\mathbf{x})$ turns out to be the Lovász extension of the cut function defined in (2). It has been proved that the second smallest eigenvalue of the 1-Laplacian is identical to the Cheeger constant $h_2$ (Li and Milenkovic, 2018).

## 3. EDVW-based submodular hypergraphs

### 3.1. The hypergraph model with EDVW

Let $\mathcal{H} = (\mathcal{V}, \mathcal{E}, \mu, \kappa, \{\gamma_e\})$ represent a hypergraph with EDVW (Chitra and Raphael, 2019) where $\mathcal{V}$, $\mathcal{E}$, and $\mu$ respectively denote the vertex set, the hyperedge set, and positive vertex weights. The function $\kappa: \mathcal{E} \to \mathbb{R}_+$ assigns positive weights to hyperedges, and those weights can reflect the strength of connection. Each hyperedge $e \in \mathcal{E}$ is associated with a function $\gamma_e: \mathcal{V} \to \mathbb{R}_{\geq 0}$ to assign edge-dependent vertex weights. For $v \in e$, $\gamma_e(v)$ is positive and measures the importance of the vertex $v$ to the hyperedge $e$; for $v \notin e$, $\gamma_e(v)$ is set to zero. For convenience, we define $\gamma_e(\mathcal{S}) = \sum_{v \in \mathcal{S}} \gamma_e(v)$.

The motivation of introducing EDVW is to enable the hypergraph model to describe the cases when the vertices in the same hyperedge contribute differently to this hyperedge. This information cannot be captured by hypergraphs adopting the all-or-nothing or cardinality-based splitting functions and is also hard to be directly described by submodular hypergraphs, but it can be conveniently represented *via* EDVW. For example, Chitra and Raphael (2019) studies the application of ranking authors in an academic citation network where authors and papers are respectively modeled as vertices and hyperedges. For a paper $e$ and any author $v$ of the paper, the corresponding EDVW $\gamma_e(v)$ is set to 2 if $v$ is the first or last author, otherwise the weight is set to 1.

## 3.2. Building submodular hypergraphs from EDVW

In order to effectively handle EDVW while still leveraging existing results obtained for submodular hypergraphs, we consider the conversion from a hypergraph with EDVW $\mathcal{H} = (\mathcal{V}, \mathcal{E}, \mu, \kappa, \{\gamma_e\})$ to a submodular hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E}, \mu, \{w_e\})$. The basic idea is to keep $\mathcal{V}$, $\mathcal{E}$, and $\mu$ unchanged, and construct submodular splitting functions $\{w_e\}$ from EDVW $\{\gamma_e\}$ and hyperedge weights $\kappa$. After such a transformation, we can directly extend concepts such as $p$-Laplacians and related theorems proposed for the submodular hypergraph model (Li and Milenkovic, 2018) to the EDVW-based hypergraph model.

In our preliminary works (Zhu and Segarra, 2022; Zhu et al., 2022), we have proposed a class of submodular EDVW-based splitting functions in the following form

$$w_e(\mathcal{S}) = h_e(\kappa(e)) \cdot g_e(\gamma_e(\mathcal{S})), \qquad (6)$$

where $h_e : \mathbb{R}_+ \to \mathbb{R}_+$ is an arbitrary function and $g_e : [0, \gamma_e(e)] \to \mathbb{R}_{\geq 0}$ is concave, symmetric with respect to $\gamma_e(e)/2$, and satisfies $g_e(0) = 0$. The resulting $w_e$ is a valid splitting function that is non-negative, submodular, symmetric, and satisfies $w_e(\emptyset) = 0$. In practice, it is reasonable to select a non-decreasing function for $h_e$ such as $h_e(x) = 1$ and $h_e(x) = x$ since a larger hyperedge weight $\kappa(e)$ is expected to lead to a larger splitting penalty for the same split of the hyperedge. Possible choices of $g_e$ include $g_e(x) = x \cdot (\gamma_e(e) - x)$, $g_e(x) = \min\{x, \gamma_e(e) - x\}$, and $g_e(x) = \min\{x, \gamma_e(e) - x, b\}$ where $b$ is a positive parameter. Also notice that for trivial EDVW, namely $\gamma_e(v) = 1$ for all $v \in e$, the splitting functions defined as (6) reduce to cardinality-based ones (Veldt et al., 2020).

## 3.3. Hypergraph-to-graph reductions

Submodular hypergraphs with splitting functions defined as (6) have a desirable property that they are graph reducible. In other words, they can project onto some graph which shares identical cut properties. Following Veldt et al. (2020), we consider the reduction of a hypergraph to a (possibly directed) graph with a potentially augmented vertex set. For a directed graph (digraph) $\mathcal{G}$ with vertex set $\mathcal{V}_{\mathcal{G}}$ and weighted adjacency matrix $\mathbf{A}$ whose entry $A_{uv}$ denotes the weight of the directed edge pointing from $u$ to $v$, its cut function is defined as

$$\text{cut}_{\mathcal{G}}(\mathcal{S}) = \sum_{u \in \mathcal{S}, v \in \mathcal{V}_{\mathcal{G}} \setminus \mathcal{S}} A_{uv}. \qquad (7)$$

In the following, we state a formal definition for graph reducibility in terms of cut weights, which is a variant of the definition in terms of hyperedge splitting functions stated in Veldt et al. (2020).

Definition 1. For a submodular hypergraph $\mathcal{H}$ with vertex set $\mathcal{V}$, we say that its cut function $\text{cut}_{\mathcal{H}}(\mathcal{S})$ is reducible to the cut function of a directed graph $\mathcal{G}$ with vertex set $\mathcal{V}_{\mathcal{G}} = \mathcal{V} \cup \bar{\mathcal{V}}$ where $\bar{\mathcal{V}}$ is a set of auxiliary vertices if the following equality holds

$$\text{cut}_{\mathcal{H}}(\mathcal{S}) = \min_{\mathcal{T} \subseteq \bar{\mathcal{V}}} \text{cut}_{\mathcal{G}}(\mathcal{S} \cup \mathcal{T}), \quad \forall \mathcal{S} \subseteq \mathcal{V}. \qquad (8)$$

```
1: Input: hypergraph with EDVW H = (V,E,μ,κ,{γₑ})
2: Convert H to a submodular hypergraph by
   constructing submodular splitting functions based
   on (6)
3: Compute the second eigenvector of the hypergraph
   1-Laplacian via the minimization of R₁(x) in (11)
4: Threshold the obtained eigenvector to get the
   bipartition of V where we choose the threshold
   value as the one that minimizes the NCC in (3)
```

Algorithm 1. 1-spectral clustering for hypergraphs with EDVW.

In the following theorem, we show an equivalent condition for graph reducibility regarding the Lovász extension of the cut function, which is beneficial for our later development of the 1-spectral clustering algorithm. The Lovász extension of the graph cut function can be written as

$$Q_1^{(g)}(\mathbf{y}) = \sum_{u,v \in \mathcal{V}_{\mathcal{G}}} A_{uv} \max\{y_u - y_v, 0\} \qquad (9)$$

where $\mathbf{y}$ is a vector of length $\mathcal{V}_{\mathcal{G}}$.

Theorem 1. The equality presented in (8) is equivalent to

$$Q_1(\mathbf{x}) = \min_{\bar{\mathbf{x}} \in \mathbb{R}^M} Q_1^{(g)}(\mathbf{y}), \quad \forall \mathbf{x} \in \mathbb{R}^N \qquad (10)$$

where $Q_1(\mathbf{x})$ and $Q_1^{(g)}(\mathbf{y})$ are respectively defined in (5) and (9), $|\mathcal{V}| = N$, $|\bar{\mathcal{V}}| = M$, and $\mathbf{y} \in \mathbb{R}^{N+M}$ is composed of $\mathbf{y}_{\mathcal{V}} = \mathbf{x}$ and $\mathbf{y}_{\bar{\mathcal{V}}} = \bar{\mathbf{x}}$.

Proof. The equivalence between (8) and (10) is proved in Appendix B.

It has been shown in Veldt et al. (2020) that all hypergraphs with submodular cardinality-based splitting functions are graph reducible. Our earlier work (Zhu and Segarra, 2022) has generalized the conclusion to hypergraphs with submodular EDVW-based splitting functions. In the following section, we propose a 1-spectral clustering algorithm for all submodular hypergraphs that are graph reducible including those EDVW-based ones, which are the focus of this paper.

# 4. Spectral clustering based on the 1-Laplacian

## 4.1. IPM-based 1-spectral clustering

We study spectral clustering algorithms for EDVW-based submodular hypergraphs leveraging the 1-Laplacian. As mentioned in section 2.2, for submodular hypergraphs the Cheeger constant $h_2$ is equal to the second smallest eigenvalue $\lambda_2$ of the 1-Laplacian $\triangle_1$. The corresponding optimal bipartition can be obtained by thresholding the eigenvector of $\triangle_1$ associated with $\lambda_2$ (Li and Milenkovic, 2018). This eigenvector can be computed by minimizing

$$R_1(\mathbf{x}) = \frac{Q_1(\mathbf{x})}{\min_{c \in \mathbb{R}} \|\mathbf{x} - c\mathbf{1}\|_{1,\mu}}, \qquad (11)$$

1: **Input:** submodular hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E}, \mu, \{w_e\})$ with $N$ vertices, accuracy $\epsilon$

2: **Initialization:** non-constant $\mathbf{x} \in \mathbb{R}^N$ subject to $0 \in$ $\mathrm{argmin}_c \|\mathbf{x} - c\mathbf{1}\|_{1,\mu}$, $\lambda \leftarrow R_1(\mathbf{x})$

3: **repeat**

4: $\forall v \in \mathcal{V}$, $g_v \leftarrow \begin{cases} \mathrm{sgn}(x_v) \cdot \mu(v), & \text{if } x_v \neq 0 \\ \frac{\mu_-(\mathbf{x}) - \mu_+(\mathbf{x})}{\mu_0(\mathbf{x})} \cdot \mu(v), & \text{if } x_v = 0 \end{cases}$

5: $\mathbf{x} \leftarrow \mathrm{argmin}_{\|\mathbf{x}\| \leq 1} Q_1(\mathbf{x}) - \lambda \langle \mathbf{x}, \mathbf{g} \rangle$          **(inner problem)**

6: $c \leftarrow \mathrm{argmin}_c \|\mathbf{x} - c\mathbf{1}\|_{1,\mu}$

7: $\mathbf{x} \leftarrow \mathbf{x} - c\mathbf{1}$

8: $\lambda' \leftarrow \lambda, \lambda \leftarrow R_1(\mathbf{x})$

9: **until** $\frac{|\lambda - \lambda'|}{\lambda'} < \epsilon$

10: **Output:** $\mathbf{x}$

Algorithm 2. IPM-based minimization of $R_1(\mathbf{x})$ (Hein and Bühler, 2010; Li and Milenkovic, 2018).

where $\|\mathbf{x}\|_{1,\mu} = \sum_{v \in \mathcal{V}} \mu(v)|x_v|$. Given the eigenvector $\mathbf{x}$, a partitioning can be defined as $\mathcal{S} = \{v \in \mathcal{V} | x_v > t\}$ and its complement, where $t$ is a threshold value. The optimal $t$ can be determined as the one that minimizes the NCC in (3). The pipeline for the 1-spectral clustering algorithm is summarized in Algorithm 1.

The minimization of $R_1(\mathbf{x})$ can be solved based on the inverse power method (Hein and Bühler, 2010; Li and Milenkovic, 2018), as outlined in Algorithm 2. Three functions are introduced: $\mu_+(\mathbf{x}) = \sum_{v \in \mathcal{V}: x_v > 0} \mu(v)$, $\mu_-(\mathbf{x}) = \sum_{v \in \mathcal{V}: x_v < 0} \mu(v)$ and $\mu_0(\mathbf{x}) = \sum_{v \in \mathcal{V}: x_v = 0} \mu(v)$. Although this algorithm cannot guarantee convergence to the second eigenvector, the objective $R_1(\mathbf{x})$ is guaranteed to decrease and converge in the iterative process. Moreover, if we start from some point $\mathbf{x} = \mathbf{1}_{\mathcal{S}}$ in Algorithm 2 where $(\mathcal{S}, \mathcal{V} \setminus \mathcal{S})$ is a given partition of $\mathcal{V}$, then each step of the IPM-based method gives a partition that has a smaller (or at least equal) NCC value (see theorem 4.2 in Hein and Setzer, 2011). This implies that we can leverage the partition obtained *via* other methods as initialization. The algorithm was first proposed for the undirected graph setting (Hein and Bühler, 2010), then generalized to submodular hypergraphs with cardinality-based splitting functions (Li and Milenkovic, 2018). It is actually a special case of a more general class of minimization algorithms called RatioDCA and generalized RatioDCA proposed in Hein and Setzer (2011) and Tudisco et al. (2018) in order to handle more types of balanced graph cuts and modularity measures, respectively. The major difference between the graph setting and the hypergraph setting lies in how the inner-loop optimization problem (cf. line 5 in Algorithm 2) is solved.

In Li and Milenkovic (2018), the authors solved the inner problem using a random coordinate descent method (Ene and Nguyen, 2015) together with a divide-and-conquer algorithm proposed in Jegelka et al. (2013). The computational complexity of the divide-and-conquer algorithm depends on the time of solving the problem $\min_{\mathcal{S} \subseteq e} F(\mathcal{S}) \triangleq w_e(\mathcal{S}) + \mathbf{z}(\mathcal{S})$ for an arbitrary vector $\mathbf{z} \in \mathbb{R}^{|e|}$. For a cardinality-based splitting function, the solution to this problem can be found efficiently *via* a line search even when $|e|$ is large. In the line search, we create a series of sets $\mathcal{S}_0 = \emptyset, \mathcal{S}_1, \cdots, \mathcal{S}_{|e|}$, where $\mathcal{S}_i$ contains $i$ vertices corresponding

to the first $i$ smallest entries in vector $\mathbf{z}$. We compare their objective values $F(\mathcal{S}_i)$ and identify the solution $\mathcal{S}_{i*}$ leading to the minimum objective value. However, such solution does not work for EDVW-based splitting functions. In the following section, we study the inner problem considering the EDVW-based case.

## 4.2. Solution to the inner problem

We propose efficient solutions to the inner problem for EDVW-based submodular hypergraphs leveraging the property that they are graph reducible. More generally speaking, the proposed solutions work for all graph reducible submodular hypergraphs. We first show that the inner problem is equivalent to another optimization problem (12) defined on the digraph obtained *via* the reduction.

Theorem 2. For any submodular hypergraph $\mathcal{H}$ with vertex set $\mathcal{V}$, if it is reducible to a digraph $\mathcal{G}$ with vertex set $\mathcal{V}_{\mathcal{G}} = \mathcal{V} \cup \bar{\mathcal{V}}$ and edge set $\mathcal{E}_{\mathcal{G}}$, i.e., (8) or (10) holds, then the solution $\mathbf{x}$ to the inner problem in Algorithm 2 can be obtained (up to a scalar multiple) by setting $\mathbf{x} = \mathbf{y}_{\mathcal{V}}$ and $\mathbf{y} \in \mathbb{R}^{|\mathcal{V}_{\mathcal{G}}|}$ is the solution to

$$\min_{\|\mathbf{y}\|_2 \leq 1} Q_1^{(g)}(\mathbf{y}) - \langle \mathbf{y}, \tilde{\mathbf{g}} \rangle \qquad (12)$$

where $\tilde{\mathbf{g}} \in \mathbb{R}^{|\mathcal{V}_{\mathcal{G}}|}$ is a vector composed of $\tilde{\mathbf{g}}_{\mathcal{V}} = \lambda \mathbf{g}$ and $\tilde{\mathbf{g}}_{\bar{\mathcal{V}}} = \mathbf{0}$.

*Proof.* The proof can be found in Appendix C where we have used the equivalent definition for graph reducibility proposed in **Theorem 1**.

We present two ways for solving problem (12) in sections 4.2.1 and 4.2.2, respectively.

### 4.2.1. Solving the inner problem *via* FISTA

Both of the original inner problem and its equivalent problem (12) are convex but non-smooth. Inspired by Hein and Bühler (2010), we derive a dual formulation (13) of problem (12). Compared to the primal problem, the objective function $\Psi$ of the dual problem is smooth. Moreover, problem (13) can be efficiently solved using a fast iterative shrinkage-thresholding algorithm (FISTA) (Nesterov, 1983; Beck and Teboulle, 2009), which has a guaranteed convergence rate $O(1/k^2)$ where $k$ is the number of iterations. FISTA requires an upper bound on the Lipschitz constant of the gradient of $\Psi$ as the input, which is provided in (15). The steps of FISTA are summarized in Algorithm 3.

To make it clear, in the theorem below, the set $\tilde{\mathcal{E}}_{\mathcal{G}}$ (as well as the directed edge set $\mathcal{E}_{\mathcal{G}}$) contains *ordered* node pairs, meaning that $(u, v)$ and $(v, u)$ are different. The parameter $m$ can be understood as the number of connected node pairs where the connection might be unidirectional or bidirectional.

Theorem 3. Define a set $\tilde{\mathcal{E}}_{\mathcal{G}} = \{(u, v) \in \mathcal{V}_{\mathcal{G}} \times \mathcal{V}_{\mathcal{G}} | (u, v) \in \mathcal{E}_{\mathcal{G}} \text{ or } (v, u) \in \mathcal{E}_{\mathcal{G}}\}$ and set $m = |\tilde{\mathcal{E}}_{\mathcal{G}}|/2$. The dual of problem (12) is

$$\min_{\substack{\boldsymbol{\alpha} \in [0,1]^m \\ \alpha_{uv} + \alpha_{vu} = 1}} \Psi(\boldsymbol{\alpha}) \triangleq \|f_A(\boldsymbol{\alpha}) - \tilde{\boldsymbol{g}}\|_2^2 \qquad (13)$$

```
1: Input: A Lipschitz constant L of ∇Ψ
2: Initialization: α = β ∈ ℝᵐ , t = 1
3: repeat
4: α′ ← α , α ← β − 1/L ∇Ψ(β)
5: α ← 𝒫₀,₁(α)
6: t′ ← t , t ← (1+√(1+4t²))/2
7: β ← α + (t′−1)/t (α − α′)
8: until convergence or a predefined maximum number
    of iterations is reached
9: Output: α
```

Algorithm 3.  Solution of problem (13) with FISTA.

```
1: Initialization: τσ ≤ 1/‖B‖₂² ,  y = ȳ ∈ ℝ^{|𝒱_𝒢|} ,  z ∈ ℝ^{|ℰ_𝒢|}
2: repeat
3: z ← 𝒫₀,₁(z + σBȳ)
4: y′ ← y ,  y ← 1/(1+τ) (y − τ(Bᵀz − g̃))
5: θ ← 1/√(1+τ) ,  τ ← θτ ,  σ ← σ/θ
6: ȳ ← y + θ(y − y′)
7: until convergence or a predefined maximum number
    of iterations is reached
8: Output: y
```

Algorithm 4.  Solution of problem (18) with accelerated PDHG.

where $\boldsymbol{\alpha}$ is a vector of length $m$ collecting all $\alpha_{uv}$ satisfying $(u, v) \in \tilde{\mathcal{E}}_{\mathcal{G}}$ and $u < v$. For the function $f_A : \mathbb{R}^m \to \mathbb{R}^{N+M}$, the $u$th element of $f_A(\boldsymbol{\alpha})$ is

$$[f_A(\boldsymbol{\alpha})]_u = \sum_{v \mid (u,v) \in \tilde{\mathcal{E}}_{\mathcal{G}}} A_{uv}\alpha_{uv} - A_{vu}\alpha_{vu}.$$

The primal and dual variables are related as

$$\mathbf{y} = -\frac{f_A(\boldsymbol{\alpha}) - \tilde{\boldsymbol{g}}}{\|f_A(\boldsymbol{\alpha}) - \tilde{\boldsymbol{g}}\|_2}. \qquad (14)$$

The Lipschitz constant of the gradient of $\Psi$ is upper bounded by

$$L = 4 \max_{u \in \mathcal{V}_{\mathcal{G}}} \sum_{v \in \mathcal{V}_{\mathcal{G}}} (A_{uv} + A_{vu})^2. \qquad (15)$$

*Proof.*  The proof is given in Appendix D.

In a nutshell, to solve the inner problem, we first compute the adjacency matrix of the digraph according to the graph reduction procedure proposed in Zhu and Segarra (2022). Then we solve the dual problem (13) using FISTA, and get the solution $\mathbf{y}$ to the primal problem (12) according to the relation between the primal and dual variables as shown in (14). Finally, we obtain the solution $\mathbf{x}$ to the original problem by taking the entries of $\mathbf{y}$ indexed by $\mathcal{V}$.

### 4.2.2. Solving the inner problem *via* PDHG

Problem (12) can also be solved using a primal-dual hybrid gradient (PDHG) algorithm (Chambolle and Pock, 2011, 2016a,b). Although both FISTA and PDHG ensure a quadratic convergence rate, it has been observed that PDHG can outperform FISTA in practice for clustering applications (Hein and Setzer, 2011).

PDHG is able to solve problems in the following form:

$$\min_{\mathbf{y} \in \mathbb{R}^{N_1}} f_1(\mathbf{y}) + f_2(\mathbf{B}\mathbf{y}), \qquad (16)$$

whose dual problem is

$$\max_{\mathbf{z} \in \mathbb{R}^{N_2}} -f_1^*(-\mathbf{B}^\top \mathbf{z}) - f_2^*(\mathbf{z}), \qquad (17)$$

where $f_1 : \mathbb{R}^{N_1} \to (-\infty, +\infty]$ and $f_2 : \mathbb{R}^{N_2} \to (-\infty, +\infty]$ are proper, convex, lower semicontinuous functions, $\mathbf{B} : \mathbb{R}^{N_1} \to \mathbb{R}^{N_2}$

is a bounded linear operator, and $f_1^*$ and $f_2^*$ are the corresponding conjugate functions of $f_1$ and $f_2$ (Boyd et al., 2004).

The solution to problem (12) can be obtained *via* normalizing the solution to problem (18) below.

$$\min_{\mathbf{y} \in \mathbb{R}^{|\mathcal{V}_{\mathcal{G}}|}} Q_1^{(g)}(\mathbf{y}) + \frac{1}{2}\|\mathbf{y} - \tilde{\boldsymbol{g}}\|_2^2. \qquad (18)$$

We can fit (18) into the form (16) by setting $N_1 = |\mathcal{V}_{\mathcal{G}}|$, $N_2 = |\mathcal{E}_{\mathcal{G}}|$, $f_1(\mathbf{y}) = \frac{1}{2}\|\mathbf{y} - \tilde{\boldsymbol{g}}\|_2^2$, $f_2(\mathbf{z}) = \sum_{i=1}^{N_2} \max\{z_i, 0\}$, and $\mathbf{B}$ is a $|\mathcal{E}_{\mathcal{G}}| \times |\mathcal{V}_{\mathcal{G}}|$ matrix. For the row of $\mathbf{B}$ corresponding to edge $u \to v$, the $u$th and $v$th elements in the row are respectively equal to $A_{uv}$ and $-A_{uv}$, and the other elements are zero. We can show that $f_1^*(\mathbf{y}) = \frac{1}{2}\|\mathbf{y}\|_2^2 + \langle \mathbf{y}, \tilde{\boldsymbol{g}}\rangle$, $f_2^*(\mathbf{z}) = 0$ with the domain that $0 \le z_i \le 1$ for any $0 \le i \le N_2$. Since $f_1$ is 1-strongly convex, we can leverage an accelerated variant of the PDHG algorithm. The algorithm tailored for problem (18) is given in Algorithm 4 (cf. algorithm 8 in Chambolle and Pock, 2016a).

## 4.3. A special case: Reduction to an undirected graph

If the submodular hypergraph is reducible to an undirected graph (e.g., see theorem 3.3 in Zhu and Segarra, 2022), then there exists another dual formulation of problem (12) that bears a similar form to (13) while the gradient of its objective has a smaller Lipschitz constant (cf. lemma 4.3 in Hein and Bühler, 2010). In fact, for this case, the hypergraph 1-spectral clustering can be implemented *via* its graph counterpart.

Following **Theorem 1**, we further define a vertex weight function $\mu_{\mathcal{G}}$ for the graph $\mathcal{G}$ such that $\mu_{\mathcal{G}}(v) = \mu(v)$ for $v \in \mathcal{V}$ and $\mu_{\mathcal{G}}(v) = 0$ for each auxiliary vertex $v \in \bar{\mathcal{V}}$. Then it follows from (10) that $R_1(\mathbf{x}) = \min_{\bar{\mathbf{x}} \in \mathbb{R}^M} R_1^{(g)}(\mathbf{y})$ for any $\mathbf{x} \in \mathbb{R}^N$ where $R_1^{(g)}(\mathbf{y}) = \frac{Q_1^{(g)}(\mathbf{y})}{\min_{c \in \mathbb{R}} \|\mathbf{y} - c\mathbf{1}\|_{1,\mu_{\mathcal{G}}}}$ and $\mathbf{y}$ is composed of $\mathbf{y}_{\mathcal{V}} = \mathbf{x}$ and $\mathbf{y}_{\bar{\mathcal{V}}} = \bar{\mathbf{x}}$. Minimizing both sides of the equation over $\mathbf{x} \in \mathbb{R}^N$ leads to the same minimum NCC value. When $\mathcal{G}$ is undirected, the second eigenvector of the graph 1-Laplacian can be computed by minimizing $R_1^{(g)}(\mathbf{y})$ and then the second eigenvector of the hypergraph 1-Laplacian can be computed as $\mathbf{x} = \mathbf{y}_{\mathcal{V}}$. This can also be understood in terms of the cut function. Following (8), it is easy to show that the Cheeger constant of the graph $\mathcal{G}$ is identical to that

of the hypergraph $\mathcal{H}$. This coincides with theorem 3.3 in (Liu et al., 2021), which further proves that if $\mathcal{S}$ is the set in $\mathcal{G}$ leading to the minimum NCC for the vertex weight function $\mu_{\mathcal{G}}$, then $\mathcal{S} \cap \mathcal{V}$ is the minimum NCC set in $\mathcal{H}$.

# 5. Experiments

We evaluate the performance of the proposed 1-spectral clustering algorithm for EDVW-based submodular hypergraphs (termed as EDVW-based hereafter) by focusing on the bipartition case. In particular, for hyperedge splitting functions as in (6), we select $h_e(x) = x$ and $g_e(x) = \min\{x, \gamma(e) - x, \beta\gamma(e)\}$ for every hyperedge $e \in \mathcal{E}$, namely

$$w_e(\mathcal{S}) = \kappa(e) \cdot \min\{\gamma_e(\mathcal{S}), \gamma_e(e) - \gamma_e(\mathcal{S}), \beta\gamma_e(e)\} \qquad (19)$$

where $\beta$ is tunable. Notice that we reuse the symbols $\alpha$ and $\beta$ in this section, which are different from and not related to $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ used when describing FISTA. A hypergraph with splitting functions as (19) can be reduced to a digraph that consists of $|\mathcal{V}|+2|\mathcal{E}|$ vertices and $|\mathcal{E}| + 2\sum_{e \in \mathcal{E}} |e|$ edges. More precisely, we first project each hyperedge $e$ onto a small graph which contains $|e| + 2$ vertices including two auxiliary vertices denoted by $e'$ and $e''$. For every $v \in e$, there are two directed edges respectively from $v$ to $e'$ and from $e''$ to $v$, both of weight $\kappa(e)\gamma_e(v)$. There is also a directed edge of weight $\beta\kappa(e)\gamma_e(e)$ from $e'$ to $e''$. Then we concatenate these small graphs for all hyperedges together to form the final graph.

## 5.1. Datasets

We consider three widely used real-world datasets.

*Reuters Corpus Volume 1 (RCV1):* This dataset is a collection of manually categorized newswire stories (Lewis et al., 2004). We consider two categories C14 and C23. A few short documents containing less than 20 words are ignored. We select the 100 most frequent words in the corpus after removing stop words and words appearing in $> 10$ and $< 0.2\%$ of the documents. We then remove documents containing $< 5$ selected words, leaving us with $7,446$ documents. A document (vertex) belongs to a word (hyperedge) if the word appears in the document. The edge-dependent vertex weights are taken as the corresponding tf-idf (term frequency-inverse document frequency) values (Leskovec et al., 2020) to the power of $\alpha$, where $\alpha$ is a tunable parameter.

*20 Newsgroups [1]:* This is also a text dataset. For our two-partition case, we consider the documents in categories "rec.autos" and "rec.sport.hockey." We preprocess the dataset following the same procedure as used in RCV1 above. We finally construct a hypergraph of $1,389$ vertices and 100 hyperedges.

*Covtype:* This dataset contains patches of forest that are in different cover types. We consider two cover types (labeled as 4 and 5) and all numerical features. Each numerical feature is first quantized into 20 bins of equal range and then mapped to hyperedges. The resulting hypergraph has $12,240$ vertices and 196

hyperedges. For each hyperedge (bin), we compute the distance between each feature value in this bin and their median, and then normalize these distances to the range $[0, 1]$. The edge-dependent vertex weights are computed as $\exp(-\alpha \cdot \text{distance})$. Under this setting, larger edge-dependent vertex weights are assigned to vertices whose feature values are close to the typical feature value in the corresponding bin.

Following Hayashi et al. (2020), for all datasets we set the hyperedge weight $\kappa(e)$ to the standard deviation of the EDVW $\gamma_e(v)$ for all $v \in \mathcal{V}$. Following Li and Milenkovic (2018), we set the vertex weight $\mu(v)$ to the vertex degree defined in the submodular hypergraph model, i.e., $\mu(v) = \sum_{e \ni v} \vartheta_e$.

## 5.2. Baselines

We compare the proposed approach with three baseline methods.

*Random walk-based:* The paper Hayashi et al. (2020) defines a hypergraph Laplacian matrix based on random walks with EDVW. We compute the second eigenvector of the normalized hypergraph Laplacian [cf. (6) in Hayashi et al., 2020] and then threshold it to get the partitioning.

*Cardinality-based:* In the description of the datasets above, when $\alpha = 0$, we get the trivial EDVW and the splitting functions reduce to cardinality-based ones.
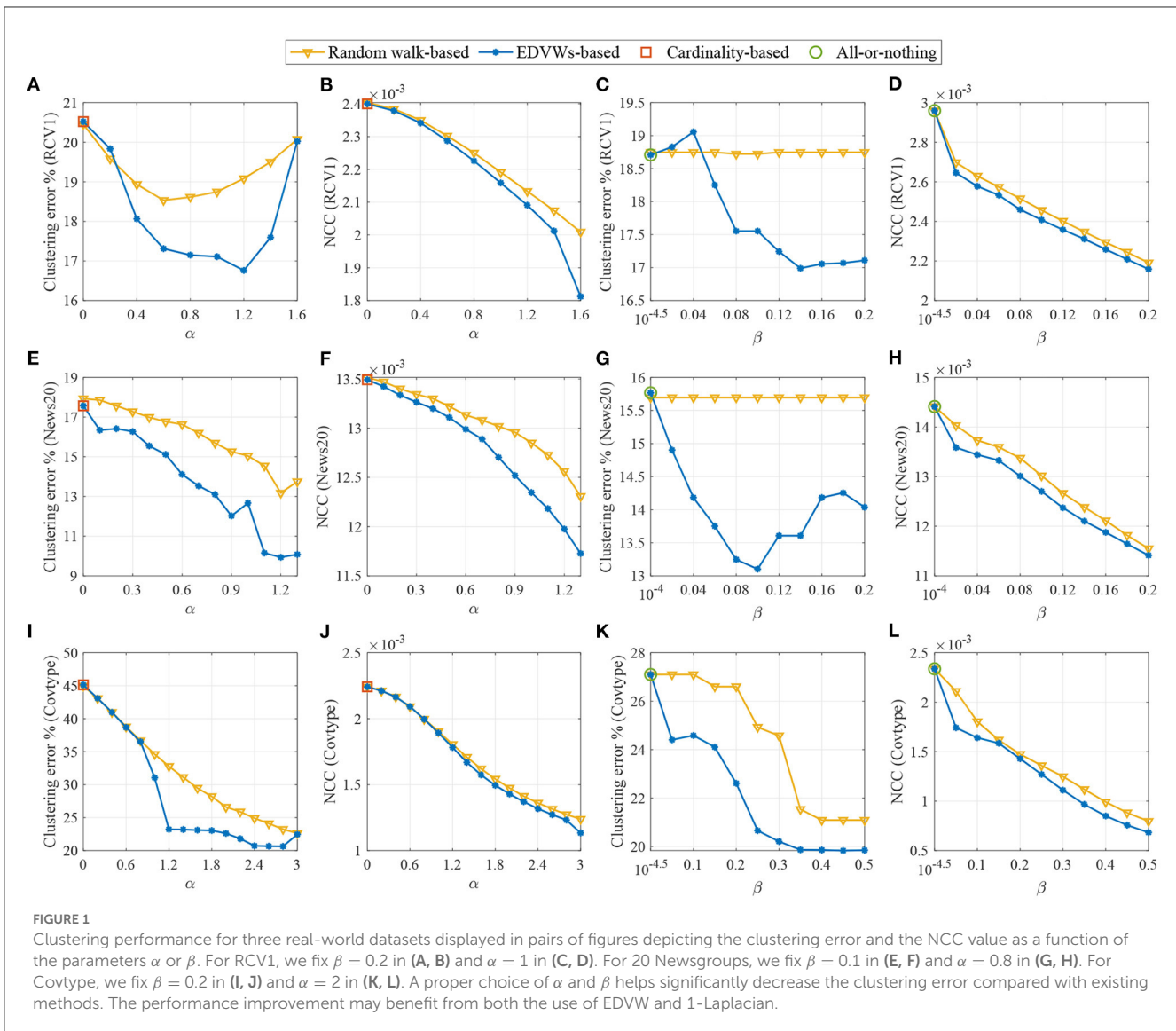
*All-or-nothing:* For the adopted splitting functions (19), they reduce to the all-or-nothing case if $\beta$ is small enough [i.e., $\beta \leq \min_{v \in e} \gamma_e(v)$].

For the proposed method, in Algorithm 2 we adopt the eigenvector obtained in the random walk-based method described above as the starting point. We solve the inner problem using PDHG presented in Algorithm 4. Moreover, in Algorithm 4 we initialize $\tau = \sigma = \frac{0.9}{\|\mathbf{B}\|_2}$ and $\mathbf{z} = \mathbf{0}$.

## 5.3. Results

The results are shown in Figure 1. We present both the clustering error and the NCC where the clustering error is computed as the fraction of incorrectly clustered samples. For the RCV1 dataset (A–D), we fix $\beta = 0.2$ to observe the influence of $\alpha$ in (A, B) and fix $\alpha = 1$ to test $\beta$ in (C, D). For the 20 Newsgroups dataset (E–H), we fix $\beta = 0.1$ in (E, F) and $\alpha = 0.8$ in (G, H) to observe the effects of $\alpha$ and $\beta$, respectively. For the Covtype dataset (I–L), we fix $\beta = 0.2$ in (I, J) and $\alpha = 2$ in (K, L). We can see that for all the considered datasets, when edge-dependent vertex weights are ignored (including $\alpha = 0$ for cardinality-based splitting functions and $\beta$ close to zero for the all-or-nothing splitting function), the clustering performance is severely deteriorated, validating that it is necessary to incorporate EDVW for extra modeling flexibility. It can also be observed that, when appropriate values for $\alpha$ and $\beta$ are selected (such as $\alpha = 1.2, \beta = 0.2$ for RCV1, $\alpha = 1.2, \beta = 0.1$ for 20 Newsgroups and $\alpha = 2.4, \beta = 0.2$ for Covtype), the proposed method performs much better than the random walk-based method which depends on the classical Laplacian.

---

1 http://qwone.com/~jason/20Newsgroups/

**FIGURE 1**
Clustering performance for three real-world datasets displayed in pairs of figures depicting the clustering error and the NCC value as a function of the parameters $\alpha$ or $\beta$. For RCV1, we fix $\beta = 0.2$ in **(A, B)** and $\alpha = 1$ in **(C, D)**. For 20 Newsgroups, we fix $\beta = 0.1$ in **(E, F)** and $\alpha = 0.8$ in **(G, H)**. For Covtype, we fix $\beta = 0.2$ in **(I, J)** and $\alpha = 2$ in **(K, L)**. A proper choice of $\alpha$ and $\beta$ helps significantly decrease the clustering error compared with existing methods. The performance improvement may benefit from both the use of EDVW and 1-Laplacian.

This highlights the value of using the non-linear 1-Laplacian in spectral clustering. To summarize, both the use of EDVW and 1-Laplacian are beneficial for improving the spectral clustering performance.

# 6. Related work

We discuss the related works in this section. We will show the relationship between hypergraph Laplacians introduced *via* random walks and those defined based on submodular splitting functions. We will also show how the proposed solution to the inner problem contributes to SFM.

## 6.1. Random walk-based Laplacians

Here we expand the discussion about the relation between hypergraph Laplacians based on random

walks with EDVW and those studied in this paper considering submodular EDVW-based splitting functions.

Considering the hypergraph model with EDVW (cf. section 3.1), the random walk incorporating EDVW is defined as follows (Chitra and Raphael, 2019). Starting at some vertex $u$, the walker selects a hyperedge $e$ containing $u$ with probability proportional to $\kappa(e)$, then moves to vertex $v$ contained in $e$ with probability proportional to $\gamma_e(v)$. In this process, the probability of moving from $u$ to $v$ *via* $e$ is

$$P_{u \to e \to v} = \begin{cases} \frac{\kappa(e)}{\sum_{e' \ni u} \kappa(e')} \cdot \frac{\gamma_e(v)}{\gamma_e(e)}, & u, v \in e, \\ 0, & \text{else.} \end{cases} \quad (20)$$

Then one can define a transition matrix $\mathbf{P}$ whose $(u, v)$th entry denotes the transition probability from $u$ to $v$ and is computed as $P_{uv} = \sum_{e \in \mathcal{E}} P_{u \to e \to v}$. When the hypergraph is connected, the random walk converges to a unique stationary distribution $\boldsymbol{\pi}$ which is the all-positive dominant left eigenvector of $\mathbf{P}$ scaled to satisfy

$\|\boldsymbol{\pi}\|_1 = 1$. In Chitra and Raphael (2019) and Hayashi et al. (2020), hypergraph Laplacians based on such random walks are proposed and they are actually equal to combinatorial and normalized graph Laplacian matrices (i.e., graph 2-Laplacians) of an undirected graph which is defined on the same vertex set as the hypergraph and has the following adjacency matrix

$$\mathbf{A} = \frac{\boldsymbol{\Phi}\mathbf{P} + \mathbf{P}\boldsymbol{\Phi}}{2} \qquad (21)$$

where $\boldsymbol{\Phi}$ is a diagonal matrix whose $(v, v)$th entry is $\pi_v$.

Consider a hypergraph $\mathcal{H}$ with submodular splitting functions in the following form for each of its hyperedges:

$$w_e(\mathcal{S}) = \sum_{u \in \mathcal{S}, v \in e \setminus \mathcal{S}} \frac{\pi_u P_{u \to e \to v} + \pi_v P_{v \to e \to u}}{2}, \quad \forall \mathcal{S} \subseteq e. \quad (22)$$

It is easy to show that $\mathcal{H}$ is reducible to the graph $\mathcal{G}$ defined by the adjacency matrix (21) since they have the same cut function as $\mathrm{cut}_{\mathcal{H}}(\mathcal{S}) = \mathrm{cut}_{\mathcal{G}}(\mathcal{S}) = \sum_{u \in \mathcal{S}, v \in \mathcal{V} \setminus \mathcal{S}} \frac{\pi_u P_{uv} + \pi_v P_{vu}}{2}$. Notice that there are no auxiliary vertices introduced in the graph reduction. Following **Theorem 1**, we have $Q_1(\mathbf{x}) = Q_1^{(g)}(\mathbf{x})$, implying that the hypergraph 1-Laplacian of $\mathcal{H}$ is identical to the graph 1-Laplacian of $\mathcal{G}$ if we assume that they share the same vertex weight function $\mu$.

## 6.2. Decomposable submodular function minimization

In the inner problem of Algorithm 2, if the norm of $\mathbf{x}$ is $\|\mathbf{x}\|_\infty$, the problem is equivalent to the following SFM problem

$$\min_{\mathcal{S} \subseteq \mathcal{V}} \sum_{e \in \mathcal{E}} w_e(\mathcal{S}) - \lambda \mathbf{g}(\mathcal{S}), \qquad (23)$$

where the primal and dual variables are related as $x_v = 1$ if $v \in \mathcal{S}$ and $x_v = -1$ if $v \notin \mathcal{S}$ (Li and Milenkovic, 2018). Hence, the proposed solution to the inner problem can also be used to solve problems in the form of (23) when each function $w_e$ can be represented as a concave function applied to a modular (additive) function [cf. (6)].

## 7. Conclusion

We presented an equivalent definition of graph reducibility based on which we further proposed a 1-spectral clustering algorithm for submodular hypergraphs that are graph reducible, especially for those with EDVW-based splitting functions. Through experiments on real-world datasets, we showcased the value of combining the hypergraph 1-Laplacian and EDVW. Future research directions include: (1) Developing computation methods for the hypergraph 1-Laplacian's eigenvectors which can work efficiently for all submodular splitting functions, (2) Designing multi-way partitioning algorithms based on non-linear Laplacians (Bühler and Hein, 2009), and (3) Exploring applications of $p$-Laplacians for different values of $p$ (Fu et al., 2022).

## Data availability statement

The original contributions presented in the study are included in the article/Supplementary material, further inquiries can be directed to the corresponding author.

## Author contributions

YZ developed the methodology, conducted the experiments, and wrote the initial manuscript. SS supervised the study, contributed to the discussions, and edited the manuscript. All authors read and approved the final manuscript.

## Funding

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fdata.2023.1020173/full#supplementary-material

# References

Amghibech, S. (2003). Eigenvalues of the discrete p-laplacian for graphs. *Ars Comb.* 67, 283–302.

Bach, F. (2013). Learning with submodular functions: a convex optimization perspective. *Found. Trends Mach. Learn.* 6, 145–373. doi: 10.1561/2200000039

Beck, A., and Teboulle, M. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.* 2, 183–202. doi: 10.1137/0807 16542

Benson, A. R., Gleich, D. F., and Leskovec, J. (2016). Higher-order organization of complex networks. *Science* 353, 163–166. doi: 10.1126/science.aad9029

Boyd, S., Boyd, S. P., and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge: Cambridge University Press. doi: 10.1017/CBO9780511804441

Bühler, T., and Hein, M. (2009). "Spectral clustering based on the graph p-laplacian," in *International Conference on Machine Learning* (Montreal, Canada), 81–88. doi: 10.1145/1553374.1553385

Chambolle, A., and Pock, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.* 40, 120–145. doi: 10.1007/s10851-010-0251-1

Chambolle, A., and Pock, T. (2016a). An introduction to continuous optimization for imaging. *Acta Numer.* 25, 161–319. doi: 10.1017/S096249291600009X

Chambolle, A., and Pock, T. (2016b). On the ergodic convergence rates of a first-order primal-dual algorithm. *Math. Program.* 159, 253–287. doi: 10.1007/s10107-015-0957-3

Chang, K.-C. (2016). Spectrum of the 1-laplacian and cheeger's constant on graphs. *J. Graph Theory* 81, 167–207. doi: 10.1002/jgt.21871

Chang, K.-C., Shao, S., and Zhang, D. (2017). Nodal domains of eigenvectors for 1-laplacian on graphs. *Adv. Math.* 308, 529–574. doi: 10.1016/j.aim.2016.12.020

Chitra, U., and Raphael, B. (2019). "Random walks on hypergraphs with edge-dependent vertex weights," in *International Conference on Machine Learning*, 1172–1181. Available online at: http://proceedings.mlr.press/v97/chitra19a.html (accessed February 9, 2023).

Ene, A., and Nguyen, H. (2015). "Random coordinate descent methods for minimizing decomposable submodular functions," in *International Conference on Machine Learning*, 787–795. Available online at: http://proceedings.mlr.press/v37/ene15.html (accessed February 9, 2023).

Fu, G., Zhao, P., and Bian, Y. (2022). "p-laplacian based graph neural networks," in *International Conference on Machine Learning*, 6878–6917. Available online at: https://proceedings.mlr.press/v162/fu22e.html (accessed February 9, 2023).

Hayashi, K., Aksoy, S. G., Park, C. H., and Park, H. (2020). "Hypergraph random walks, laplacians, and clustering," in *Conference on Information and Knowledge Management* (Galway), 495–504. doi: 10.1145/3340531.3412034

Hein, M., and Bühler, T. (2010). An inverse power method for nonlinear eigenproblems with applications in 1-spectral clustering and sparse PCA. *Adv. Neural Inf. Process. Syst.* 23.

Hein, M., and Setzer, S. (2011). Beyond spectral clustering - tight relaxations of balanced graph cuts. *Adv. Neural Inf. Process. Syst.* 24.

Hein, M., Setzer, S., Jost, L., and Rangapuram, S. S. (2013). The total variation on hypergraphs-learning on hypergraphs revisited. *Adv. Neural Inf. Process. Syst.* 26.

Jegelka, S., Bach, F., and Sra, S. (2013). Reflection methods for user-friendly submodular optimization. *Adv. Neural Inf. Process. Syst.* 26.

Leskovec, J., Rajaraman, A., and Ullman, J. D. (2020). *Mining of Massive Data Sets*. Cambridge: Cambridge University Press. doi: 10.1017/9781108684163

Lewis, D. D., Yang, Y., Russell-Rose, T., and Li, F. (2004). RCV1: a new benchmark collection for text categorization research. *J. Mach. Learn. Res.* 5, 361–397.

Li, J., He, J., and Zhu, Y. (2018). "E-tail product return prediction via hypergraph-based local graph cut," in *International Conference on Knowledge Discovery and Data Mining* (London), 519–527. doi: 10.1145/3219819.3219829

Li, P., and Milenkovic, O. (2017). Inhomogeneous hypergraph clustering with applications. *Adv. Neural Inf. Process. Syst.* 30. doi: 10.1007/978-3-319-70139-4

Li, P., and Milenkovic, O. (2018). "Submodular hypergraphs: p-laplacians, cheeger inequalities and spectral clustering," in *International Conference on Machine Learning*, 3014–3023. Available online at: http://proceedings.mlr.press/v80/li18e.html (accessed February 9, 2023).

Liu, M., Veldt, N., Song, H., Li, P., and Gleich, D. F. (2021). "Strongly local hypergraph diffusions for clustering and semi-supervised learning," in *International World Wide Web Conference* (Ljubljana), 2092–2103. doi: 10.1145/3442381.3449887

Lovász, L. (1983). "Submodular functions and convexity," in *Mathematical Programming The state of the Art*, eds A. Bachem, M. Grötschel, and B. Korte (New York, NY: Springer), 235–257. doi: 10.1007/978-3-642-68874-4_10

Nesterov, Y. E. (1983). A method for solving the convex programming problem with convergence rate o (1/k2). *Dokl. Akad. Nauk SSSR* 269, 543–547.

Schaub, M. T., Zhu, Y., Seby, J.-B., Roddenberry, T. M., and Segarra, S. (2021). Signal processing on higher-order networks: Livin'on the edge…, beyond. *Signal Process.* 187, 108149. doi: 10.1016/j.sigpro.2021.108149

Szlam, A., and Bresson, X. (2010). "Total variation and cheeger cuts," in *International Conference on Machine Learning* (Haifa), 1039–1046.

Tudisco, F., and Hein, M. (2018). A nodal domain theorem and a higher-order cheeger inequality for the graph *p*-laplacian. *J. Spectr. Theory* 8, 883–908. doi: 10.4171/JST/216

Tudisco, F., Mercado, P., and Hein, M. (2018). Community detection in networks via nonlinear modularity eigenvectors. *SIAM J. Appl. Math.* 78, 2393–2419. doi: 10.1137/17M1144143

Veldt, N., Benson, A. R., and Kleinberg, J. (2020). Hypergraph cuts with general splitting functions. *arXiv* [preprint]. doi: 10.48550/arXiv.2001.02817

Von Luxburg, U. (2007). A tutorial on spectral clustering. *Stat. Comput.* 17, 395–416. doi: 10.1007/s11222-007-9033-z

Wagner, D., and Wagner, F. (1993). "Between min cut and graph bisection," in *International Symposium on Mathematical Foundations of Computer Science* (Berlin), 744–750. doi: 10.1007/3-540-57182-5_65

Yoshida, Y. (2019). "Cheeger inequalities for submodular transformations," in *ACM-SIAM Symposium on Discrete Algorithms* (Tokyo), 2582–2601. doi: 10.1137/1.9781611975482.160

Zhu, Y., Li, B., and Segarra, S. (2021). "Co-clustering vertices and hyperedges *via* spectral hypergraph partitioning," in *European Signal Processing Conference* (Dublin), 1416–1420. doi: 10.23919/EUSIPCO54536.2021.9616223

Zhu, Y., Li, B., and Segarra, S. (2022). "Hypergraphs with edge-dependent vertex weights: spectral clustering based on the 1-laplacian," in *International Conference on Acoustics, Speech and Signal Processing* (Singapore), 8837–8841. doi: 10.1109/ICASSP43922.2022.9746363

Zhu, Y., and Segarra, S. (2022). Hypergraph cuts with edge-dependent vertex weights. *Appl. Netw. Sci.* 7:45. doi: 10.1007/s41109-022-00483-x