Graph Cuts with Arbitrary Size Constraints Through Optimal Transport

Chakib Fettal, Lazhar Labiod, Mohamed Nadif

GRAPH CUTS WITH ARBITRARY SIZE CONSTRAINTS THROUGH OPTIMAL TRANSPORT

Chakib Fettal
Centre Borelli UMR 9010
Université Paris Cité

Lazhar Labiod
Centre Borelli UMR 9010
Université Paris Cité

Mohamed Nadif
Centre Borelli UMR 9010
Université Paris Cité

ABSTRACT

Clustering is an important task in computer vision and machine learning in general, and new applications are constantly appearing. A common way of obtaining an image dataset partition is through graph cuts, which are also used as a component in more complex clustering paradigms such as subspace clustering. One drawback of classical min-cut algorithms is that they tend to produce small groups, which is why more balanced variants have risen, including normalized and ratio cuts. We believe, however, that with these variants the balance constraints are too restrictive for some applications (long-tailed clustering), while not being restrictive enough for others (when searching for perfectly balanced partitions), since the constraint is not hard. Here, we propose a new graph cut algorithm for partitioning with arbitrary size constraints. We formulate the graph cut problem as a constrained Gromov-Wasserstein problem, and our algorithm is slower than the classical spectral clustering algorithm by only a ratio of $\mathcal{O}(\log(n))$ while being more efficient in practice. We demonstrate the performance of our approach on several balanced and imbalanced (long-tail) datasets.

1 Introduction

Clustering is an important task in machine learning and computer vision. Intuitively, the task of image clustering boils down to grouping images into clusters such that the images within the same clusters are similar to each other, while those in different clusters are dissimilar. Applications are diverse and wide ranging, including, for example, content-based image retrieval [Bhunia et al., 2020, Lee et al., 2022, Bhunia et al., 2021], image annotation [Cheng et al., 2018, Cai et al., 2013], and image indexing [Cao et al., 2013]. Consequently, much research has been dedicated to image clustering [Chang et al., 2017, Ji et al., 2017, Elhamifar and Vidal, 2013, Ji et al., 2019].

A popular way of formulating the image clustering problem is through the minimum graph cut (min-cut) problem where the graph is created based on the input images. However, in practice, the min-cut problem suffers from the formation of some small groups which leads to bad performance. As a result, other versions of min-cut were proposed that take into account the size of the resulting groups, in order to make the partition more balanced. This notion of size is variable, for example, in the Normalized Cut (`ncut`) problem [Shi and Malik, 2000], size refers to the total volume of a group, while in the Ratio Cut (`rcut`) problem [Hagen and Kahng, 1992], it refers to the cardinality of a group. A common method for solving the `ncut` and `rcut` problems is that of the spectral clustering algorithm [Von Luxburg, 2007, Ng et al., 2001] which is popular due to it often showing good empirical performance and being somewhat efficient. The spectral clustering algorithm variants are present in many image clustering frameworks, such as for subspace clustering [Agrawal et al., 1998] where a spectral clustering algorithm is applied to a learned subspace affinity matrix to obtain a partition of the points according to the subspaces in which they lie.

However, some restrictions that apply to the spectral clustering algorithms and to most approaches tackling the `ncut` and `rcut` problems in general do exist. A first one is that the balance constraint is not strict enough, meaning that even if we include the size regularization into the min-cut problem, the groups are still not necessarily of similar size, which is why several truly balanced clustering algorithm have been proposed in the literature [Chen et al., 2017, Chen et al., 2019, Li et al., 2018]. Another problem is that the balance constraint is too restrictive for many real world datasets, for example, a recent trend in computer vision is to propose approaches dealing with long-tailed datasets which are datasets that contain head classes that represent most of the overall dataset and then have tail classes that represent a

small fraction of the overall dataset [Xu et al., 2022, Zhu et al., 2014]. Some approaches propose integrating generic size constraints to the objective like in [Genevay et al., 2019, Höppner and Klawonn, 2008, Zhu et al., 2010], however these approaches directly deal with the input images (or data in general) instead of graphs.

In this paper, we propose a novel framework that introduces generic and at the same time stricter size constraints to the min-cut problem using Optimal Transport. To sum up, the main contributions of this work are :

- We formulate a problem for obtaining graph cuts that are balanced for an arbitrarily defined notion of size instead of specifically the volume or cardinality as is traditionally done in spectral clustering. We also propose a more general formulation of graph cuts with cluster size constraints which can help when dealing with perfectly balanced datasets and heavily imbalanced datasets such as long-tailed datasets which follow an exponential decay in sample sizes across different classes.
- We then propose a solution for said problem through optimal transport using an approach reminiscent of the simplex algorithm and analyze its computational complexity. Links with existing works are also studied.
- Comprehensive experiments on balanced and long-tailed data sets using two variants we named `OT-ncut` and `OT-rcut` showcase the effectiveness of the proposed method compared to the most common min-cut algorithms (that use spectral clustering) both in terms of obtaining the desired cluster sizes as well as clustering performance. We release the code of our algorithm [1] for reproducibility.

The rest of this paper is organized as follows : Preliminaries are presented in Section 2. Some related work is discussed in section 3. The `OT-cut` problem and algorithm along with their analysis and links to prior research are given in section 4. We present experimental results and analysis in section 5. Conclusions are then given in section 6.

2 Related Work

Our work is related with balanced clustering, as the latter is a special case of it, as well as with the more generic problem of constrained clustering.

Balanced Clustering. A common class of constrained clustering problems is balanced clustering where we wish to obtain a partition with clusters of the same size. For example, [DeSieno, 1988] introduced a conscience mechanism which penalizes clusters relative to their size, [Ahalt et al., 1990], then employed it to develop the Frequency Sensitive Competitive Learning (FSCL) algorithm. In [Li et al., 2018], authors proposed to leverage the exclusive lasso on the k-means and min-cut problems to regulate the balance degree of the clustering results. [Lin et al., 2019] proposed a simplex algorithm to solve a minimum cost flow problem similar to k-means. In [Chen et al., 2017], authors proposed a self-balanced min-cut algorithm for image clustering implicitly using exclusive lasso as a balance regularizer in order to produce balanced partitions.

Constrained Clustering. Some clustering approaches with generic size constraints, which can be seen as an extension of balanced clustering, also exist. In [Zhu et al., 2010], a heuristic algorithm to transform size constrained clustering problems into integer linear programming problems was proposed. Authors in [Ganganath et al., 2014] introduced a modified k-means algorithm which can be used to obtain clusters of preferred sizes. Clustering paradigms based on OT generally offer the possibility to set a target distribution for resulting partitions. [Genevay et al., 2019] proposed a deep clustering algorithm through optimal transport with entropic regularization. In [Fettal et al., 2022], authors proposed a way to perform biclustering which is an extension of clustering to bipartite graphs through Optimal Transport while choosing the size of the resulting biclusters.

3 Preliminaries

In what follows, $\Delta^n = \{\mathbf{p} \in \mathbb{R}_+^n | \sum_{i=1}^n p_i = 1\}$ denotes the n-dimensional standard simplex. $\Pi(\mathbf{w}, \mathbf{v}) = \{\mathbf{Z} \in \mathbb{R}_+^{n \times k} | \mathbf{Z1} = \mathbf{w}, \mathbf{Z}^\top \mathbf{1} = \mathbf{v}\}$ denotes the transportation polytope, where $\mathbf{w} \in \Delta^n$ and $\mathbf{v} \in \Delta^k$ are the marginals of the joint distribution $\mathbf{Z}$ and $\mathbf{1}$ is a vector of ones, its size can be inferred from the context. Matrices are denoted with uppercase boldface letters, and vectors with lowercase boldface letters. For a matrix $\mathbf{M}$, its i-th row is $\mathbf{m}_i$. Tr refers to the trace of a square matrix. $\|.\|_0$ is the zero norm that returns the number of nonzero elements in its argument. $\otimes$ denotes tensor-matrix product.

[1] https://github.com/chakib401/ot-cut/

3.1 Graph Cuts

Graph Cut. Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with an weighted adjacency matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$ with $n = |\mathcal{V}|$, a cut is a partition of its vertices $\mathcal{V}$ into two disjoint subsets $\mathcal{A}$ and $\bar{\mathcal{A}}$. The value of a cut is given by

$$\text{cut}(\mathcal{A}) = \sum_{v_i \in \mathcal{A}, v_j \in \bar{\mathcal{A}}} w_{ij}. \tag{1}$$

Minimum *k*-cut Problem. The goal of the minimum k-cut problem is to find a partition $(\mathcal{A}_1, ..., \mathcal{A}_k)$ of the set of vertices $\mathcal{V}$ into k different groups that is minimal in some metric. Intuitively, we wish for the edges between different subsets to have small weights, and for the edges within a subset have large weights. Formally, it is defined as

$$\text{min-cut}(\mathbf{W}, k) = \min_{\mathcal{A}_1, ..., \mathcal{A}_k} \sum_{i=1}^{k} \text{cut}(\mathcal{A}_i). \tag{2}$$

This problem can also be stated as a trace minimization problem by representing the resulting partition $\mathcal{A}_1, \ldots, \mathcal{A}_k$ using an assignment matrix $\mathbf{X}$ such that for each row i, we have that

$$x_{ij} = \begin{cases} 1 & \text{if vertex } i \text{ is in } \mathcal{A}_j, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

This condition is equivalent to introducing two constraints which are $\mathbf{X} \in \{0, 1\}^{n \times k}$ and $\mathbf{X1} = \mathbf{1}$. The minimum k-cut problem can then be formulated as

$$\text{min-cut}(\mathbf{W}, k) = \min_{\substack{\mathbf{X} \in \{0,1\}^{n \times k} \\ \mathbf{X1} = \mathbf{1}}} \text{Tr}(\mathbf{X}^\top \mathbf{L} \mathbf{X}), \tag{4}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ refers to the graph Laplacian of the graph $\mathcal{G}$ and $\mathbf{D}$ is the diagonal matrix of degree of $\mathbf{W}$, i.e., $d_{ii} = \sum_j w_{ij}$.

Normalized Cut Problem. In practice, solutions to the minimum k-cut problem do not yield satisfactory partitions due to the formation of small groups of vertices. Consequently, versions of the problem that take into account some notion of "size" for these groups have been proposed. The most commonly used one is normalized cut [Shi and Malik, 2000] :

$$\texttt{ncut}(\mathbf{W}, k) = \min_{\mathcal{A}_1, ..., \mathcal{A}_k} \sum_{i=1}^{k} \frac{\text{cut}(\mathcal{A}_i)}{\text{vol}(\mathcal{A}_i)}, \tag{5}$$

since $\text{vol}(\mathcal{A}_i) = \mathbf{x}_i^T \mathbf{D} \mathbf{x}_i$, then this problem can also be stated as a trace minimization problem :

$$\texttt{ncut}(\mathbf{W}, k) = \min_{\substack{\mathbf{X1} = \mathbf{1} \\ \mathbf{X} \in \{0,1\}^{n \times k}}} \text{Tr}\left(\frac{\mathbf{X}^\top \mathbf{L} \mathbf{X}}{\mathbf{X}^\top \mathbf{D} \mathbf{X}}\right), \tag{6}$$

where the ratio can be taken as either right or left multiplication of the numerator by the inverse of the denominator, this equivalence is due to the fact that we use the trace operator and that the denominator is a diagonal matrix. A special case of the normalized graph cut is recovered by setting $\mathbf{D} = \mathbf{I}$ in problem 6. This problem is referred to as the ratio cut problem due to the different groups being normalized by their cardinality instead of their volumes :

$$\texttt{rcut}(W, k) = \min_{\mathcal{A}_1, ..., \mathcal{A}_k} \sum_{i=1}^{k} \frac{\text{cut}(\mathcal{A}_i)}{|\mathcal{A}_i|}, \tag{7}$$

and similarly to the normalized cut, since $|\mathcal{A}_i| = \mathbf{x}_i^T \mathbf{x}_i$, we can formulate the ratio cut problem as a trace minimization problem :

$$\texttt{rcut}(\mathbf{W}, k) = \min_{\substack{\mathbf{X} \in \{0,1\}^{n \times k} \\ \mathbf{X1} = \mathbf{1}}} \text{Tr}\left(\frac{\mathbf{X}^\top \mathbf{L} \mathbf{X}}{\mathbf{X}^\top \mathbf{X}}\right). \tag{8}$$

Spectral Clustering for the Normalized & Ratio Cuts. A common approach to solving the normalized graph cut problems, spectral clustering, replaces the partition constraints on $\mathbf{X}$ with a form of semi-orthogonality constraints. In the case of `rcut`, we have

$$\texttt{ncut}(\mathbf{W}, k) = \min_{\substack{\mathbf{X}\in\mathbb{R}^{n\times k}\\ \mathbf{X}^\top\mathbf{X}=\mathbf{I}}} \operatorname{Tr}\left(\mathbf{X}^\top\mathbf{L}\mathbf{X}\right). \tag{9}$$

On the other hand for `ncut`, the partition matrix $\mathbf{X}$ is substituted with $\mathbf{H} = \mathbf{D}^{1/2}\mathbf{X}$ and a semi-orthogonality constraint is placed on this $\mathbf{H}$, i.e.,

$$\texttt{ncut}(\mathbf{W}, k) = \min_{\substack{\mathbf{H}\in\mathbb{R}^{n\times k}\\ \mathbf{H}^\top\mathbf{H}=\mathbf{I}}} \operatorname{Tr}\left(\mathbf{H}^\top\mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}\mathbf{H}\right). \tag{10}$$

A solution $\mathbf{H}$ for the `ncut` problem is formed by stacking the first k-eigenvectors of the symmetrically normalized Laplacian $\mathbf{L}_s = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$ as its columns, and then applying a clustering algorithm such as k-means on its rows and assign the original data points accordingly [Ng et al., 2001]. The principle is the same for the spectral `rcut` algorithm.

3.2 Optimal Transport

Discrete optimal transport. The goal of the optimal transport problem is to find a minimal cost transport plan $\mathbf{X}$ between a source probability distribution of $\mathbf{w}$ and a target probability distribution $\mathbf{v}$. Here we are interested in the discrete Kantorovich formulation of OT. When dealing with discrete probability distributions, said formulation is

$$\texttt{OT}(\mathbf{M}, \mathbf{w}, \mathbf{v}) \triangleq \min_{\mathbf{X}\in\Pi(\mathbf{w},\mathbf{v})} \langle \mathbf{M}, \mathbf{X}\rangle, \tag{11}$$

where $\mathbf{M} \in \mathbb{R}^{n\times k}$ is the cost matrix, and c_{ij} quantifies the effort needed to transport a probability mass from $\mathbf{w}_i$ to $\mathbf{v}_j$.

Discrete Gromov-Wasserstein Discrepancy. The generic discrete Gromov-Wasserstein (GW) discrepancy [Peyré et al., 2016] is an extension of optimal transport to the case where the source and target distributions are defined on different metric spaces :

$$\mathrm{GW}(\mathbf{M}, \mathbf{M}', \mathbf{w}, \mathbf{v}) \triangleq \min_{\mathbf{X}\in\Pi(\mathbf{w},\mathbf{v})} \langle L(\mathbf{M}, \mathbf{M}') \otimes \mathbf{X}, \mathbf{X}\rangle \tag{12}$$

where $\mathbf{M} \in \mathbb{R}^{n\times n}$ and $\mathbf{M}' \in \mathbb{R}^{k\times k}$ are similarity matrices defined on the source space and target space respectively, and $L : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$ is a divergence measure between scalars, $L(\mathbf{M}, \mathbf{M}')$ is the $n \times n \times k \times k$ tensor of all pairwise divergences between the elements of $\mathbf{M}$ and $\mathbf{M}'$.

Gromov-Wasserstein Learning for Graphs. The Gromov-Wasserstein partitioning paradigm S-GWL [Xu et al., 2019] supposes that the Gromov-Wasserstein discrepancy can uncover the clustering structure of the observed source graph $\mathcal{G}$ when the target graph $\mathcal{G}_{dc}$ only contains weighted self-connected isolated nodes, this means that its adjacency matrix is diagonal. The weights of this diagonal matrix as well as the source and target distribution are a special function of the node degrees. Their approach uses a regularized proximal gradient method as well as a recursive partitioning scheme and can be used in a multi-view clustering setting. The problem with this approach is that it is extremely sensitive the hyperparameter setting which is problematic since it is an unsupervised method. Another approach, SpecGWL, introduced in [Chowdhury and Needham, 2021] which generalizes spectral clustering using Gromov-Wasserstein discrepancy and heat kernels but suffers from very high computational complexity since, given a graph with n node, its optimization procedure involves the computation a gradient which is in $O(n^3\log(n))$ and an eigendecompostion $O(n^3)$ and therefore is not usable for large scale graphs.

4 Graph Cuts with Size Constraints via OT

In this section, we derive our optimal transport-based constrained graph cut problem and propose a simple iterative algorithm for its resolution.

4.1 Graph Cuts via Optimal Transport

As already mentioned, the good performance of the normalized cut algorithm comes from the normalization by the volume of each group in the cut. However, the size constraint is not a hard one, meaning that obtained groups are not of

exactly the same volume. This leads us to propose to replace the volume normalization by a strict balancing constraint as follows :

$$\min_{\mathcal{A}_1,\ldots,\mathcal{A}_k} \sum_{i=1}^{k} \text{cut}(\mathcal{A}_i) \quad \text{such that} \quad \text{vol}(\mathcal{A}_1) = \ldots = \text{vol}(\mathcal{A}_k). \tag{13}$$

Similarly to the `ncut` problem, this problem can be formulated as a trace minimization problem :

$$\min_{\mathbf{X}\mathbf{1}=\mathbf{D}\mathbf{1}, \quad \mathbf{X}^\top\mathbf{1}=\frac{\sum_i d_{ii}}{k}\mathbf{1}, \quad \forall_i \|\mathbf{x}_i\|_0=1} \text{Tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X}). \tag{14}$$

This problem is hard and may not have feasible solutions. However, this problem can be slightly modified to become an instance of the Gromov-Wasserstein problem, to which relatively efficient heuristics exist. Specifically, the volume constraint can be implicitly satisfied by defining $\mathbf{X}$ to be an element of the transportation polytope with a uniform target distribution instead of being a partition matrix. The degrees are also normalized by dividing them by their total sum and then representing them as proportions instead of absolute quantities, yielding the following problem :

$$\min_{\mathbf{X}\in\Pi\left(\frac{1}{\sum_i d_{ii}}\mathbf{D}\mathbf{1}, \frac{1}{k}\mathbf{1}\right)} \text{Tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X}) \tag{15}$$

This formulation is a special case of the Gromov-Wasserstein problem for a source space whose similarity matrix in the initial space is $\mathbf{M} = \mathbf{L}$ and whose similarity matrix in the destination space is $\mathbf{M}' = \mathbf{I}$. Note that a ratio cut version can easily be obtained by replacing the volume constraint with

$$|\mathcal{A}_1| = \ldots = |\mathcal{A}_k| \tag{16}$$

in problem 14, and similarly in problem 15, by setting $\mathbf{D} = \mathbf{I}$, giving rise to :

$$\min_{\mathbf{X}\in\Pi\left(\frac{1}{n}\mathbf{1}, \frac{1}{k}\mathbf{1}\right)} \text{Tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X}) \tag{17}$$

4.2 Graph Cuts with Size Constraints

From the previous problem, it is easy to see that target distribution does not need to be uniform, and as such, any distribution can be considered, leading to further applications like long-tailed dataset clustering. Another observation is that any notion of size can be considered and not only the volume or cardinality of the formed node groups. We formulate an initial version of the generic optimal transport graph cut problem as :

$$\min_{\mathbf{X}\in\Pi(\boldsymbol{\pi}^s,\boldsymbol{\pi}^t)} \text{Tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X}) \equiv \min_{\mathbf{X}\in\Pi(\boldsymbol{\pi}^s,\boldsymbol{\pi}^t)} \langle \mathbf{L}\mathbf{X}, \mathbf{X}\rangle, \tag{18}$$

where $\boldsymbol{\pi}^s_i$ is the relative 'size' of the element i and $\boldsymbol{\pi}^t_j$ is the desired relative 'size' of the group j. Through the form that uses the Frobenius product, it is easy to see how our problem is related to the Gromov-Wasserstein problem. These size parameters can either be set using domain knowledge by the expert using our algorithm or by trying multiple guesses and selecting the best one via internal clustering quality metrics such as Davies-Bouldin index [Davies and Bouldin, 1979], Silhouette score [Rousseeuw, 1987], etc.

4.3 Transport Plans as Partition Matrices

The proposed approach relies on the fact that the transport plan $\mathbf{X}$ can be interpreted as a partition matrix. Fortunately, this interpretation can be made through the concept of h-almost hard clustering [Fettal et al., 2022] :

Definition 1 (h-almost hard clustering)**.** *An h-almost hard clustering is a clustering whose partition matrix is $\otimes \in \mathbb{R}^{n\times k}$ such that $\| \otimes \|_0 = n + h$ and for each row $\boldsymbol{c}$ of $\otimes$ we have that $\|\boldsymbol{c}\|_0 > 0$. When $h = 0$, we obtain a standard hard clustering with one non-zero element per row.*

The extreme points of the transportation polytope are always h-almost hard clustering (see [Peyré et al., 2019, Fettal et al., 2022] for a proof), so we add a boundary condition to our problem in order to always obtain a transport plan $\mathbf{X}$ that can be interpreted easily as a hard partition matrix :

$$\texttt{OT-cut}(\mathbf{L}, \boldsymbol{\pi}^s, \boldsymbol{\pi}^t) \triangleq \min_{\mathbf{X}\in\texttt{ext}(\Pi(\boldsymbol{\pi}^s,\boldsymbol{\pi}^t))} \text{Tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X}) \tag{19}$$

where ext is the set of extreme points of its argument. Consequently we have that :

Proposition 1. *A solution* $\mathbf{X}$ *to the* `OT-cut` *problem is an* h*-almost hard clustering with* $h \in \{0, \dots, k-1\}$.

We can obtain a size constrained variant of the ncut problem by setting $\boldsymbol{\pi}^s = \frac{1}{\sum_i d_{ii}}\mathbf{D1}$:

$$\texttt{OT-ncut}(\mathbf{L}, \boldsymbol{\pi}^t) \triangleq \min_{\mathbf{X} \in \texttt{ext}\left(\Pi\left(\frac{1}{\sum_i d_{ii}}\mathbf{D1}, \boldsymbol{\pi}^t\right)\right)} \operatorname{Tr}(\mathbf{X}^\top \mathbf{L}\mathbf{X}). \tag{20}$$

Analogously, the variant of the rcut problem is obtained by setting $\boldsymbol{\pi}^s = \frac{1}{n}\mathbf{1}$, yielding :

$$\texttt{OT-rcut}(\mathbf{L}, \boldsymbol{\pi}^t) \triangleq \min_{\mathbf{X} \in \texttt{ext}\left(\Pi\left(\frac{1}{n}\mathbf{1}, \boldsymbol{\pi}^t\right)\right)} \operatorname{Tr}(\mathbf{X}^\top \mathbf{L}\mathbf{X}), \tag{21}$$

4.4 Optimization and Complexity

Problem 19 is nonconvex due to the boundary constraint. We propose to use proximal gradient descent with constant stepsize to search for a stationary point. We add an l2-norm regularizer to simplify the update rule obtained with the proximal gradient method. The resulting update rule is :

$$\mathbf{X}^{(t+1)} := \arg \texttt{OT}\left((\mathbf{L} - \mathbf{I})\mathbf{X}^{(t)}, \boldsymbol{\pi}^s, \boldsymbol{\pi}^t\right). \tag{22}$$

Note that when using the symmetrically normalized Laplacian matrix $\mathbf{L}_{sym}$, we have that $\mathbf{W}_{sym} = \mathbf{I} - \mathbf{L}_{sym}$ and the update rule becomes :

$$\mathbf{X}^{(t+1)} := \arg \texttt{OT}\left(-\mathbf{W}_{sym}\mathbf{X}^{(t)}, \boldsymbol{\pi}^s, \boldsymbol{\pi}^t\right). \tag{23}$$

The resolution of this problem is possible by stating it as the earth-mover's distance (EMD) linear program [Hitchcock, 1941] which can be solved via the network simplex algorithm. This algorithm has been empirically observed to converge to some stationary point in few iteration based on the initial guess $\mathbf{X}^{(0)}$. To illustrate this, in figure 1, we can report the evolution of loss function OT-rcut on MNIST, OT-rcut on FMNIST and OT-ncut on CIFAR-10 ($\rho = 10$). The pseudocode for the optimization procedure is presented in algorithm 1. Similarly to the algorithm proposed in [Peyré et al., 2016] for solving the GW problem with an arbitrary loss and cost matrices, there are no convergence guarantees. Possible heuristics to improve the quality of the final solution would be doing multiple runs with different initializations, or initializing the algorithm with a partition matrix obtained from a spectral-cut algorithm projected onto the transportation polytope.

Proof. We formulate the l_2-norm regularized `OT-cut` problem as

$$\min_{\mathbf{X}} \quad \underbrace{\operatorname{Tr}(\mathbf{X}^\top \mathbf{L}\mathbf{X})}_{f(\mathbf{X})} + \underbrace{I_{\texttt{ext}(\Pi(\boldsymbol{\pi}^s, \boldsymbol{\pi}^t))}(\mathbf{X}) - \|\mathbf{X}\|^2}_{g(\mathbf{X})}$$

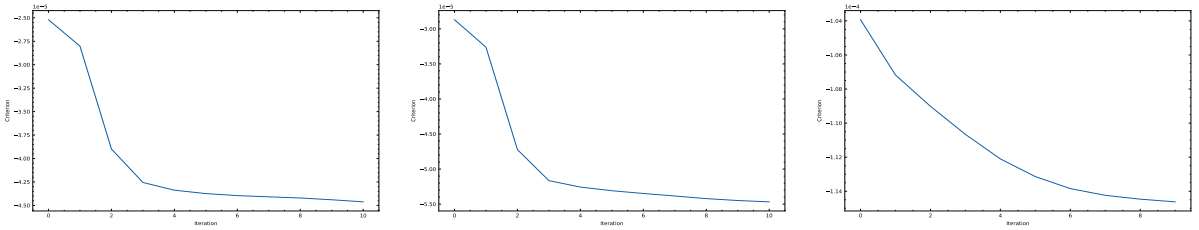
where $I_{\mathcal{C}}$ is the indicator function of set $\mathcal{C}$. The proximal gradient update rule with respect to this problem is:

$$\begin{aligned}
\mathbf{X}^{(t+1)} &:= \texttt{prox}_{\lambda g}\left(\mathbf{X}^{(t)} - \lambda \nabla f(\mathbf{X}^{(t)})\right) \\
&:= \texttt{prox}_{\lambda g}\left(\mathbf{X}^{(t)} - \lambda \nabla \operatorname{Tr}\left(\mathbf{X}^{(t)}\mathbf{L}\mathbf{X}^{(t)}\right)\right) \\
&:= \texttt{prox}_{\lambda(I_{\texttt{ext}(\Pi(\boldsymbol{\pi}^s, \boldsymbol{\pi}^t))} - \|\cdot\|^2)}\left((\mathbf{I} - 2\lambda\mathbf{L})\mathbf{X}^{(t)}\right) \\
&:= \operatorname*{arg\,min}_{\mathbf{Z} \in \texttt{ext}(\Pi(\boldsymbol{\pi}^s, \boldsymbol{\pi}^t))} \frac{1}{2\lambda}\left\|\mathbf{Z} - (\mathbf{I} - 2\lambda\mathbf{L})\mathbf{X}^{(t)}\right\|^2 - \|\mathbf{Z}\|^2 \\
&:= \operatorname*{arg\,min}_{\mathbf{Z} \in \texttt{ext}(\Pi(\boldsymbol{\pi}^s, \boldsymbol{\pi}^t))} \frac{1}{2\lambda}\|\mathbf{Z}\|^2 + \frac{1}{2\lambda}\left\|(\mathbf{I} - 2\lambda\mathbf{L})\mathbf{X}^{(t)}\right\|^2 \\
&\qquad - \frac{1}{\lambda}\operatorname{Tr}\left(\mathbf{Z}^\top(\mathbf{I} - 2\lambda\mathbf{L})\mathbf{X}^{(t)}\right) - \|\mathbf{Z}\|^2,
\end{aligned} \tag{24}$$

then by setting $\lambda = \frac{1}{2}$:

$$\begin{aligned}
\mathbf{X}^{(t+1)} &:= \operatorname*{arg\,min}_{\mathbf{Z} \in \texttt{ext}(\Pi(\boldsymbol{\pi}^s, \boldsymbol{\pi}^t))} \operatorname{Tr}\left(\mathbf{Z}^\top(\mathbf{L} - \mathbf{I})\mathbf{X}^{(t)}\right), \\
&:= \operatorname*{arg\,min}_{\mathbf{Z} \in \texttt{ext}(\Pi(\boldsymbol{\pi}^s, \boldsymbol{\pi}^t))} \left\langle \mathbf{Z}, (\mathbf{L} - \mathbf{I})\mathbf{X}^{(t)} \right\rangle,
\end{aligned} \tag{25}$$

here, we can drop the boundary constraint as the solution is guaranteed to be an extreme point of the transportation polytope. This results in the classical OT problem with cost matrix $(\mathbf{L} - \mathbf{I})\mathbf{X}^{(t)}$ and marginals π^s and π^t. □



(a) OT-rcut on MNIST (b) OT-rcut on FMNIST (c) OT-ncut on CIFAR-10 (ρ=10)

Figure 1: Evolution of loss as function of the number of iterations.

Proposition 2. *For a graph with $|\mathcal{E}|$ edges and n nodes, the complexity of an iteration of the proposed algorithm is $\mathcal{O}\left(kn^2 \log n\right)$.*

Proof. We note that in practice $n >> k$ and that the complexity of the network simplex algorithm for some graph $\mathcal{G}_{EMD} = (\mathcal{V}_{EMD}, \mathcal{V}_{EMD})$ is in $\mathcal{O}(|\mathcal{V}_{EMD}||\mathcal{E}_{EMD}| \log |\mathcal{E}_{EMD}|)$ [Orlin, 1997]. In our case, this graph has $|\mathcal{V}_{EMD}| = n + k$ (since $n >> k$, we can drop the k) and $|\mathcal{E}_{EMD}| = nk$. The other operation that is performed during each iteration is the matrix multiplication $(\mathbf{L} - \mathbf{I})\mathbf{X}^{(t)}$ whose complexity is in $\mathcal{O}(k|\mathcal{E}|)$, in the worst case when matrix $\mathbf{L}$ is fully dense, we have that $|\mathcal{E}| = n^2$. Note that the complexity of the spectral clustering algorithm is in $\mathcal{O}(kn^2)$. □

Algorithm 1: Proximal Gradient Descent for `OT-cut`

Input : $\mathbf{L}$ Laplacian matrix,
$\boldsymbol{\pi}^s$ node size distribution,
$\boldsymbol{\pi}^t$ cluster size distribution,
$\mathbf{G}_{init}$ initial partition matrix,
maxIter maximum number of iterations.
Output : $\mathbf{G}$ partition of the graph.
$\mathbf{X}^{(0)} := \arg \texttt{OT}\left(\mathbf{G}_{init}, \boldsymbol{\pi}^s, \boldsymbol{\pi}^t\right)$;
while $t <$ *maxIter* **do**
  $\mathbf{X}^{(t+1)} := \arg \texttt{OT}\left((\mathbf{L} - \mathbf{I})\mathbf{X}^{(t)}, \boldsymbol{\pi}^s, \boldsymbol{\pi}^t\right)$;
end
Generate partition matrix $\mathbf{G}$ such that each node v_i is assigned it to partition $\arg\max_i x_i$;

5 Links to Prior Works

In this section we discuss how our approach generalizes and can used in conjunction with other approaches.

Optimal-Transport Based Biclustering Biclustering is the extension of clustering to bipartite graphs. Here, we recover the BCOT [Fettal et al., 2022] problem as a special case of `OT-cut`. Given a bipartite adjacency matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{n\times n} & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{0}_{d\times d} \end{bmatrix},$$

we recover their formulation through ours by considering this anti-adjacency matrix $\bar{\mathbf{A}}$:

$$\bar{\mathbf{A}} = \begin{bmatrix} \infty_{n\times n} & L(\mathbf{B}) \\ L(\mathbf{B})^\top & \infty_{d\times d} \end{bmatrix}.$$

Then setting $\boldsymbol{\pi}^s = [\mathbf{v}, \mathbf{w}]^\top$ and $\boldsymbol{\pi}^s = [\mathbf{v}, \mathbf{w}]^\top$ and omitting the boundary condition. All in all, we have that

$$\text{BCOT}(L(\mathbf{B}), \mathbf{w}, \mathbf{v}) \equiv \texttt{OT-cut}\left(\bar{\mathbf{A}}, [\mathbf{v}, \mathbf{w}]^\top, [\mathbf{w}, \mathbf{v}]^\top\right)$$

OT Kernel *k*-Means. In [Genevay et al., 2019], authors proposed an algorithm for k-means with cluster size constraints and entropic regularization. By dropping the regularization and adding a boundary constraint, one can think of the case where the adjacency matrix in our formulation is a kernel matrix and use the same principles that were used with kernel k-means [Dhillon et al., 2004] to optimize the OT graph cut criterion.

Table 1: Characteristics of the datasets from which we construct the graphs. The balance score ρ is the ratio of the number of occurrences of the most frequent class over that of the least frequent class.

Dataset	#Images	#Classes	Balance
MNIST	60,000	10	1.0
FMNIST	60,000	10	1.0
KMNIST	60,000	10	1.0
CIFAR-10	50,000	10	1.0
CIFAR-10 ($\rho = 5$)	25,423	10	5.0
CIFAR-10 ($\rho = 10$)	20,431	10	10.0
CIFAR-10 ($\rho = 20$)	17,023	10	20.0
CIFAR-10 ($\rho = 100$)	12,406	10	100.0

Table 2: Image clustering performance on the imbalanced (long-tail) datasets. Values are the averages over five runs. Standard deviations were not reported due to being negligible (≤ 0.1). Best results are highlighted in bold font.

	CIFAR-10 ($\rho = 5$)			CIFAR-10 ($\rho = 10$)			CIFAR-10 ($\rho = 20$)			CIFAR-10 ($\rho = 100$)		
	NMI	ARI	CF1	NMI	ARI	CF1	NMI	ARI	CF1	NMI	ARI	CF1
SC-rcut	0.1	-0.0	3.3	0.1	-0.0	4.0	0.1	-0.0	4.6	0.1	-0.0	5.8
OT-rcut	11.6	7.3	20.7	**12.1**	**7.8**	**19.8**	**11.4**	7.5	**17.9**	9.8	5.7	13.7
OT-rcut$_{SC}$	11.1	6.4	20.8	10.6	6.5	18.7	11.3	7.4	17.1	9.8	5.8	13.7
OT-rcut$^{*}_{SC}$	11.2	6.1	19.5	10.5	5.4	16.6	10.8	5.4	14.6	11.6	5.6	14.3
SC-ncut	10.2	5.6	19.1	10.5	6.2	18.0	10.6	5.8	16.4	**12.7**	6.8	14.6
OT-ncut	**12.0**	**8.3**	**21.3**	10.1	7.3	18.9	10.6	**7.9**	17.3	8.4	6.9	13.8
OT-ncut$_{SC}$	10.8	7.5	20.7	10.8	7.5	18.6	10.5	7.8	16.2	10.4	**8.3**	**14.8**
OT-ncut$^{*}_{SC}$	10.4	5.9	20.4	10.4	5.6	18.0	10.6	5.7	16.4	10.9	5.6	13.1

6 Experiments

We ran experiments on balanced and heavily imbalanced (long-tailed) datasets. We evaluated the clustering performance of three variants of each of `OT-ncut` and `OT-rcut` algorithms against the spectral `rcut` and `ncut` algorithms, as well as the ability of our approach to recover the desired partition distribution. We had initially also considered S-GWL as a baseline but its empirical performance was very poor, specifically, it consistently resulted in assigning all the nodes to a single cluster. Another OT-based approach is SpecGWL which was not considered due to its log-cubic complexity which makes it unusable for the datasets we considered. See the supplementary material for more information and additional experiments.

6.1 Benchmark Datasets

We perform experiments on balanced datasets and long-tailed datasets, namey, MNIST [Deng, 2012], FMNIST [Xiao et al., 2017], KMNIST [Clanuwat et al., 2018], CIFAR-10 [Krizhevsky et al., 2009] and unbalanced variants of CIFAR-10 [Cao et al., 2019]. The statistical summaries of these datasets are available in table 5. The CIFAR-10 ($\rho = \frac{\max_i n_i}{\min_i n_i}$) variants, are generated using a long-tailed imbalance sampling method that yields a dataset whose majority class is ρ times more frequent than the minority class following the procedure described in [Cao et al., 2019].

6.2 Performance Metrics

The evaluation is straightforward, we adopt four popular clustering metrics when dealing with the balanced datasets : clustering accuracy (CA), clustering F1 score (CF1), normalized mutual information (NMI), and adjusted rand index (ARI) [Hubert and Arabie, 1985]; multiplied by 100. CA and CF1 are computed by solving a linear assignment problem [Crouse, 2016]. When dealing with the long-tailed dataset, we only use metrics that are sensitive to imbalance NMI, ARI, and CF1. When comparing the concordance of the input cluster distribution $\boldsymbol{\pi}$ and the cluster distribution obtained via one of our algorithms $\hat{\boldsymbol{\pi}}$, we use the Kullback-Leibler divergence [Kullback and Leibler, 1951]. The concordance of two perfectly matching distributions will be equal to zero, otherwise it will be larger.

Table 3: Clustering performance on balanced image datasets. Values are the averages over five runs. Standard deviations were not reported due to being negligible (≤ 0.1). Best results are highlighted in bold font. OT-rcut* has the same results since the ground truth sizes are uniform, similarly, OT-rcut$_{SC}$ also has the same results due to SC-rcut returning a bad guess that is equivalent to a random initialization.

	MNIST				FMNIST				KMNIST				CIFAR 10			
	ACC	NMI	ARI	CF1	ACC	NMI	ARI	CF1	ACC	NMI	ARI	CF1	ACC	NMI	ARI	CF1
SC-ncut	40.2	34.7	17.6	37.6	53.4	53.2	36.5	51.6	37.3	30.4	19.8	35.2	22.2	10.1	5.6	21.5
OT-ncut$_{SC}$	**48.2**	**36.4**	**27.1**	**48.2**	47.8	51.0	35.5	47.0	43.6	33.2	24.2	42.8	21.5	11.5	**6.4**	21.3
OT-ncut$^*_{SC}$	41.5	35.5	25.0	41.1	**56.3**	53.0	**40.7**	**56.0**	**44.7**	**33.6**	**24.2**	**44.5**	23.0	10.8	5.8	23.0
SC-rcut	11.2	0.0	-0.0	2.0	10.0	0.0	0.0	1.8	10.0	0.0	0.0	1.8	10.0	0.0	0.0	1.8
OT-rcut	38.3	32.3	20.7	38.2	54.3	**53.9**	39.0	54.3	41.6	33.1	22.3	41.6	**23.8**	**11.7**	**6.4**	**23.8**
OT-rcut$_{SC}$	Same results as OT-rcut															
OT-rcut$^*_{SC}$	Same results as OT-rcut															

Table 4: The Kullback-Leibler divergence between the imposed target distribution and the one obtained using `OT-cut` variants.

	OT-ncut*	OT-ncut	OT-rcut*	OT-rcut
MNIST	3.0e-09	2.3e-09	0.0	0.0
FMNIST	1.7e-09	2.6e-09	0.0	0.0
KMNIST	2.5e-09	4.1e-09	0.0	0.0
CIFAR-10	3.7e-09	3.1e-09	0.0	0.0
CIFAR-10 (ρ=5)	0.0	2.5e-7	1.6e-8	0.0
CIFAR-10 (ρ=10)	1.4e-8	6.7e-9	1.1e-8	0.0
CIFAR-10 (ρ=20)	7.0e-9	1.2e-8	7.1e-8	0.0
CIFAR-10 (ρ=100)	8.6e-8	5.1e-8	7.8e-8	0.0

6.3 Experimental settings

We compare two variants of our algorithm, namely, the `OT-ncut` and `OT-rcut` implemented via the Python optimal transport package (POT) [Flamary et al., 2021] to the spectral clustering variants SC-ncut and SC-rcut which were based on the implementation of the spectral clustering in the Scikit-Learn package [Pedregosa et al., 2011]. For each image dataset represented in matrix form as $\mathbf{Y}$, we use subspace Least Squares Regression Subspace Clustering (LSR) [Lu et al., 2012] to create the graph, we get $\mathbf{A} = \mathbf{Y}\mathbf{Y}^\top \left(\mathbf{Y}\mathbf{Y}^\top + \mathbf{I}\right)^{-1}$. Note that all experiments are run five times. In the results tables, base `OT-cuts` variants use random initialization. The variants that end with * use the ground truth target distribution. Finally, for variants ending in $_{SC}$, we choose the initial transport plan $\mathbf{X}^{(0)}$ by first obtaining a partition matrix through the corresponding spectral clustering algorithm, i.e., spectral ncut (SC-ncut) for OT-ncut and spectral rcut (SC-rcut) to OT-rcut. We perform 10 iterations of our algorithm to fine-tune the initial guesses of spectral cuts and perform 20 iterations when using random initialization. All experiments were performed on a 64gb RAM machine with a 12th Gen Intel(R) Core(TM) i9-12950HX (24 CPUs) processor with a frequency of 2.3GHz.

6.4 Experimental Results

We emphasize the fact that the most important quality metric is the relative difference instead of the absolute value of the metrics as our objective is not to learn a better graph over the dataset but rather to get a better cut over the chosen graph.

Performance on balanced datasets. For balanced datasets, the results are reported in table 3. One of our two approaches yields the best results in all 16 cases. Furthermore, each one of them improves over the results of their spectral counterpart, exceeding them in 31 out of 32 cases. Notably, our `OT-rcut` variant achieves a significant improvement over the spectral ratio cut algorithm.

Performance on long-tailed datasets. Table 2 presents the results obtained on the long-tailed datasets. In all cases, `OT-ncut` and `OT-rcut` outperform their spectral clustering counterparts, yielding the best performance in 11 out of 12 cases. Notably, the improvement of `OT-rcut` over SC-rcut is particularly significant, consistent with the findings in the balanced case.

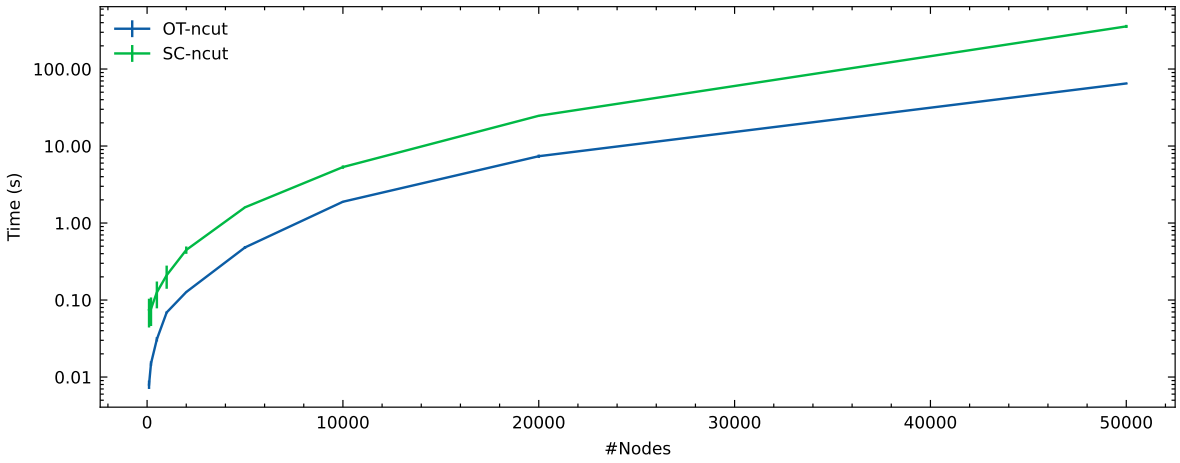


Figure 2: Training times of OT-ncut in seconds (log scale) over subsets of different sizes of MNIST.

Concordance of the Desired & Resulting Cluster Sizes. To evaluate our algorithm's ability to produce a partition with the desired group size distribution, we use the Kullback-Leibler (KL) divergence metric. Specifically, we compare the distribution obtained by our OT-rcut and OT-ncut variants against the target distribution specified as a hyperparameter ($\boldsymbol{\pi}^t$). Table 4 presents the KL divergences for both variants on various datasets. Our approaches achieve near-perfect performance on most datasets. Notably, OT-rcut is able to fully recover the desired group sizes.

Running Time. As shown in figure 2, OT-ncut with random initialization is more efficient than the spectral ncut algorithm, significantly outspeeding it on all subsets of MNIST despite being theoretically more complex. This is due to the fact that our algorithm needs few iterations to converge. Regularization can be introduced to further speed up our algorithms such as low-rank [Scetbon and marco cuturi, 2022] and entropic [Cuturi, 2013] regularizations .

7 Conclusion

In this paper we proposed a new graph cut algorithm for partitioning with arbitrary size constraints through optimal transport. This approach generalizes the concept of the normalized and ratio cut to arbitrary size distributions and this for any notion of size. The proposed algorithm works well when used in conjunction with a classical spectral graph cut algorithm as a post-processing step to obtain some desired distribution. Experiments on balanced and imbalanced datasets showed the superiority of our approach both in terms of clustering performance and empirical speed compared to spectral clustering, as well as its ability to recover partitions that almost perfectly match the desired ones which is valuable for practical problems where we wish to obtain constrained and balanced clusterings.

References

[Agrawal et al., 1998] Agrawal, R., Gehrke, J., Gunopulos, D., and Raghavan, P. (1998). Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pages 94–105.

[Ahalt et al., 1990] Ahalt, S. C., Krishnamurthy, A. K., Chen, P., and Melton, D. E. (1990). Competitive learning algorithms for vector quantization. *Neural networks*, 3(3):277–290.

[Bhunia et al., 2021] Bhunia, A. K., Chowdhury, P. N., Sain, A., Yang, Y., Xiang, T., and Song, Y.-Z. (2021). More photos are all you need: Semi-supervised learning for fine-grained sketch based image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4247–4256.

[Bhunia et al., 2020] Bhunia, A. K., Yang, Y., Hospedales, T. M., Xiang, T., and Song, Y.-Z. (2020). Sketch less for more: On-the-fly fine-grained sketch-based image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9779–9788.

[Cai et al., 2013] Cai, X., Nie, F., Cai, W., and Huang, H. (2013). New graph structured sparsity model for multi-label image annotations. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 801–808.

[Cao et al., 2013] Cao, J., Wu, Z., Wu, J., and Liu, W. (2013). Towards information-theoretic k-means clustering for image indexing. *Signal Processing*, 93(7):2026–2037.

[Cao et al., 2019] Cao, K., Wei, C., Gaidon, A., Arechiga, N., and Ma, T. (2019). Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32.

[Chang et al., 2017] Chang, J., Wang, L., Meng, G., Xiang, S., and Pan, C. (2017). Deep adaptive image clustering. In *Proceedings of the IEEE international conference on computer vision*, pages 5879–5887.

[Chen et al., 2019] Chen, X., Chen, R., Wu, Q., Fang, Y., Nie, F., and Huang, J. Z. (2019). Labin: balanced min cut for large-scale data. *IEEE transactions on neural networks and learning systems*, 31(3):725–736.

[Chen et al., 2017] Chen, X., Zhexue Haung, J., Nie, F., Chen, R., and Wu, Q. (2017). A self-balanced min-cut algorithm for image clustering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2061–2069.

[Cheng et al., 2018] Cheng, Q., Zhang, Q., Fu, P., Tu, C., and Li, S. (2018). A survey and analysis on automatic image annotation. *Pattern Recognition*, 79:242–259.

[Chowdhury and Needham, 2021] Chowdhury, S. and Needham, T. (2021). Generalized spectral clustering via gromov-wasserstein learning. In *International Conference on Artificial Intelligence and Statistics*, pages 712–720. PMLR.

[Clanuwat et al., 2018] Clanuwat, T., Bober-Irizar, M., Kitamoto, A., Lamb, A., Yamamoto, K., and Ha, D. (2018). Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*.

[Crouse, 2016] Crouse, D. F. (2016). On implementing 2d rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696.

[Cuturi, 2013] Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26.

[Davies and Bouldin, 1979] Davies, D. L. and Bouldin, D. W. (1979). A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227.

[Deng, 2012] Deng, L. (2012). The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142.

[DeSieno, 1988] DeSieno, D. (1988). Adding a conscience to competitive learning. In *ICNN*, volume 1.

[Dhillon et al., 2004] Dhillon, I. S., Guan, Y., and Kulis, B. (2004). Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–556.

[Elhamifar and Vidal, 2013] Elhamifar, E. and Vidal, R. (2013). Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2765–2781.

[Fettal et al., 2022] Fettal, C., Labiod, L., and Nadif, M. (2022). Efficient and effective optimal transport-based biclustering. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Advances in Neural Information Processing Systems*.

[Flamary et al., 2021] Flamary, R., Courty, N., Gramfort, A., Alaya, M. Z., Boisbunon, A., Chambon, S., Chapel, L., Corenflos, A., Fatras, K., Fournier, N., Gautheron, L., Gayraud, N. T., Janati, H., Rakotomamonjy, A., Redko, I., Rolet, A., Schutz, A., Seguy, V., Sutherland, D. J., Tavenard, R., Tong, A., and Vayer, T. (2021). Pot: Python optimal transport. *The Journal of Machine Learning Research*, 22(1).

[Ganganath et al., 2014] Ganganath, N., Cheng, C.-T., and Chi, K. T. (2014). Data clustering with cluster size constraints using a modified k-means algorithm. In *2014 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, pages 158–161. IEEE.

[Genevay et al., 2019] Genevay, A., Dulac-Arnold, G., and Vert, J.-P. (2019). Differentiable deep clustering with cluster size constraints. *arXiv preprint arXiv:1910.09036*.

[Hagen and Kahng, 1992] Hagen, L. and Kahng, A. B. (1992). New spectral methods for ratio cut partitioning and clustering. *IEEE transactions on computer-aided design of integrated circuits and systems*, 11(9):1074–1085.

[Hitchcock, 1941] Hitchcock, F. L. (1941). The distribution of a product from several sources to numerous localities. *Journal of mathematics and physics*, 20(1-4):224–230.

[Höppner and Klawonn, 2008] Höppner, F. and Klawonn, F. (2008). Clustering with size constraints. In *Computational Intelligence Paradigms*, pages 167–180. Springer.

[Hubert and Arabie, 1985] Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of classification*, 2(1):193–218.

[Ji et al., 2017] Ji, P., Zhang, T., Li, H., Salzmann, M., and Reid, I. (2017). Deep subspace clustering networks. *Advances in neural information processing systems*, 30.

[Ji et al., 2019] Ji, X., Henriques, J. F., and Vedaldi, A. (2019). Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9865–9874.

[Krizhevsky et al., 2009] Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

[Kullback and Leibler, 1951] Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

[Lee et al., 2022] Lee, S., Seong, H., Lee, S., and Kim, E. (2022). Correlation verification for image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5374–5384.

[Li et al., 2018] Li, Z., Nie, F., Chang, X., Ma, Z., and Yang, Y. (2018). Balanced clustering via exclusive lasso: A pragmatic approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

[Lin et al., 2019] Lin, W., He, Z., and Xiao, M. (2019). Balanced clustering: A uniform model and fast algorithm. In *IJCAI*, pages 2987–2993.

[Lu et al., 2012] Lu, C.-Y., Min, H., Zhao, Z.-Q., Zhu, L., Huang, D.-S., and Yan, S. (2012). Robust and efficient subspace segmentation via least squares regression. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part VII 12*, pages 347–360. Springer.

[Ng et al., 2001] Ng, A., Jordan, M., and Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14.

[Orlin, 1997] Orlin, J. B. (1997). A polynomial time primal network simplex algorithm for minimum cost flows. *Mathematical Programming*, 78(2):109–129.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

[Peyré et al., 2019] Peyré, G., Cuturi, M., et al. (2019). Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607.

[Peyré et al., 2016] Peyré, G., Cuturi, M., and Solomon, J. (2016). Gromov-wasserstein averaging of kernel and distance matrices. In *International conference on machine learning*, pages 2664–2672. PMLR.

[Rousseeuw, 1987] Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

[Scetbon and marco cuturi, 2022] Scetbon, M. and marco cuturi (2022). Low-rank optimal transport: Approximation, statistics and debiasing. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Advances in Neural Information Processing Systems*.

[Shi and Malik, 2000] Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905.

[Von Luxburg, 2007] Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416.

[Xiao et al., 2017] Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

[Xu et al., 2019] Xu, H., Luo, D., and Carin, L. (2019). Scalable gromov-wasserstein learning for graph partitioning and matching. *Advances in neural information processing systems*, 32.

[Xu et al., 2022] Xu, Y., Li, Y.-L., Li, J., and Lu, C. (2022). Constructing balance from imbalance for long-tailed image recognition. In *European Conference on Computer Vision*, pages 38–56. Springer.

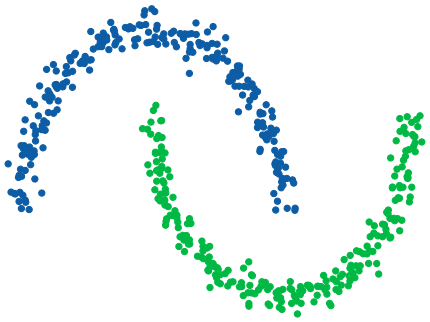
[Zhu et al., 2010] Zhu, S., Wang, D., and Li, T. (2010). Data clustering with size constraints. *Knowledge-Based Systems*, 23(8):883–889.

[Zhu et al., 2014] Zhu, X., Anguelov, D., and Ramanan, D. (2014). Capturing long-tail distributions of object subcategories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 915–922.

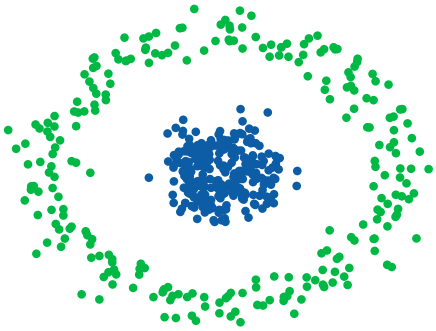
A Additional Experiments

In what follows, we report additional results on toy datasets, citation networks as well as benchmark from other OT/GW graph clustering approaches.

A.1 On Toy Datasets



(a) Partition generated by OT-ncut with spectral clustering partition as its initialization on a toy dataset (k-NN graph with $k = 10$).



(b) Partition generated by OT-rcut with random initialization on a toy dataset (RBF kernel with $\gamma = 3$).

Our algorithm deals with a graph cut-like criterion which means that it should partition a dataset according to its connectivity. This means that it should work on datasets on which metric clustering approaches such as k-means fail. Two toy examples are given in figures 3a and 3b.

A.2 On Citation Networks

We added extra experiments on three widely used citation networks [1]. Their summary statistics are reported in table 1. Results are reported in table 6. We also added S-GWL [2] as a baseline but it consistently collapsed (see its AMI and ARI). We believe that this is due to the fact that their algorithm is very sensitive to its hyperparameters as well as the fact that they used Kullback-Leibler regularization which leads to coupling matrices being dense. Considering similar regularization also lead to poor results for our approach. SpecGWL [3] was not considered due to its prohibitive computational complexity of $O(n^3 \log(n)$ in each iteration of its optimization scheme, which makes not viable in practice. Our approach outperforms all others on all three datasets.

Table 5: Summary statistics of the citation networks [1].

Dataset	#Nodes	#Classes	Balance (ρ)
Cora	2708	7	4.5
CiteSeer	3327	6	2.7
PubMed	19717	3	1.9

A.3 On the S-GWL Benchmarks

We compare our approach to S-GWL on the datasets which were used in the original paper for the single graph partition task since the authors provided the parameters for it. Our variant has the best results over EU-email while the spectral clustering while ncut has the best results over Indian-village. A variant of our approach does outperform S-GWL over both datasets. We believe that when taking all experiments into consideration (image datasets + cora, citeseer, pubmed + these two datasets) that our approach (as in one of its variants) gives the best overall results. Note that overly precise parameter setting of S-GWL makes it not very practical for unsupervised learning. In the conclusion of the S-GWL paper, it is written that "it should be noted that our S-GWL method is sensitive to its hyperparameters.".

Table 6: Results on the citation networks. We see that our OT variants consistently outperform the baselines.

	Cora			**CiteSeer**			**PubMed**		
	AMI	ARI	CF1	AMI	ARI	CF1	AMI	ARI	CF1
S-GWL	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0
SC-ncut	3.6±0.3	0.2±0.1	10.9±0.0	1.2±0.2	0.4±0.1	11.3±0.7	0.5±0.0	0.2±0.0	19.6±0.0
SC-rcut	0.8±0.0	-0.4±0.0	7.5±0.0	1.2±0.0	0.3±0.1	10.1±0.3	0.5±0.0	0.2±0.0	19.6±0.0
OT-ncut(r)*	4.0±0.1	2.2±0.8	21.6±2.4	1.0±0.1	0.9±0.2	**21.6±0.9**	**1.3±1.0**	1.8±1.3	38.2±1.9
OT-ncut*	2.6±0.2	1.6±0.3	21.2±1.1	0.7±0.2	0.6±0.2	20.0±0.0	1.2±0.2	**1.9±0.2**	**39.9±0.3**
OT-ncut(r)	3.8±0.6	2.0±0.3	22.0±0.1	1.1±0.0	0.9±0.0	**21.6±0.2**	0.8±0.2	1.0±0.2	36.6±0.6
OT-ncut	**5.8±0.1**	**3.7±0.0**	**25.1±0.7**	**1.3±0.3**	**1.0±0.3**	21.1±1.1	0.1±0.0	0.1±0.0	34.8±0.0
OT-rcut(r)*	1.1±0.1	1.1±0.7	18.2±0.0	0.4±0.1	0.3±0.1	19.7±0.7	0.3±0.0	0.4±0.0	36.0±0.1
OT-rcut*	1.1±0.2	-0.2±0.0	16.0±0.4	0.5±0.0	0.4±0.0	20.0±0.4	0.1±0.0	0.2±0.0	35.2±0.0
OT-rcut(r)	1.1±0.2	0.5±0.1	18.1±0.0	0.5±0.0	0.4±0.0	19.7±0.2	0.1±0.0	0.2±0.0	34.7±0.1
OT-rcut	1.9±0.2	1.2±0.1	20.5±1.3	0.8±0.0	0.6±0.0	20.7±0.1	0.1±0.0	0.1±0.0	33.8±0.0

	EU-EMAIL			**Indian-Village**		
	AMI	ARI	CF1	AMI	ARI	CF1
S-GWL	44.96±0.0	24.90±0.0	35.33±0.0	72.08±0.0	53.33±0.0	67.39±0.0
SC-rcut	0.38±0.0	-0.02±0.0	5.52±0.05	75.52±7.96	50.53±14.85	57.01±11.98
SC-ncut	41.16±3.66	8.66±2.61	27.3±1.6	**91.54±1.4**	**85.55±4.04**	**95.48±2.3**
OT-rcut(r)*	40.02±0.77	26.41±2.11	25.34±1.76	60.97±5.07	45.3±6.64	54.76±7.92
OT-rcut*	39.41±1.76	25.08±1.9	25.92±1.14	64.18±3.32	52.67±3.37	60.74±7.11
OT-rcut(r)	**46.71±1.04**	**27.17±0.88**	**38.1±0.39**	62.5±3.27	46.46±4.5	60.48±4.66
OT-rcut	46.39±0.65	26.63±0.36	38.09±0.68	72.97±2.02	62.65±3.23	72.38±3.55
OT-ncut(r)*	37.93±0.73	19.69±2.17	30.6±0.96	58.55±2.68	38.0±3.38	52.41±3.96
OT-ncut*	38.37±1.95	20.01±1.93	30.17±1.76	68.18±2.95	50.2±4.38	70.75±3.3
OT-ncut(r)	37.31±0.91	11.1±0.22	31.6±0.93	62.76±4.63	42.33±5.17	60.59±2.65
OT-ncut	37.62±1.87	11.44±0.91	31.05±1.89	66.36±0.82	48.63±1.66	70.54±0.81

References

[1] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. 2008. Collective classification in network data. AI magazine 29, 3 (2008), 93–93.

[2] Xu, Hongteng, Dixin Luo, and Lawrence Carin. "Scalable Gromov-Wasserstein learning for graph partitioning and matching." Advances in neural information processing systems 32 (2019).

[3] Chowdhury, Samir, and Tom Needham. "Generalized spectral clustering via Gromov-Wasserstein learning." In International Conference on Artificial Intelligence and Statistics, pp. 712-720. PMLR, 2021.