Graph-Based Semi-Supervised Learning via p-Conductances

Anonymous authors
Paper under double-blind review

Abstract

We introduce *p-conductance learning*, a graph-based semi-supervised learning method in which each class is represented by a probability measure, and predictions are obtained from potentials minimizing a graph p-energy subject to an affine measure-separation constraint. From this viewpoint, our approach connects to several familiar graph optimization problems: for $p = 1$ it admits a generalized `maxflow-mincut` characterization, for $p = 2$ it admits an effective resistance characterization and a direct connection to Poisson learning, and for $p = \infty$ it is characterized by the reciprocal of a 1-Wasserstein distance on the graph. To implement the program at scale for general p, we develop a semismooth Newton augmented Lagrangian method and prove global convergence of the outer iteration. For the semismooth Newton step, we prove local superlinear convergence for $p \in \{1, 2, \dots\}$ and $p = \infty$. Experiments on citation and image benchmarks, as well as specialized settings with corrupted and partial label information, show that the method is consistently competitive overall and, for $p = 2$ with class-size-aware decoding or MBO refinement, often attains the best or tied-best average accuracy among the compared graph-based baselines.

1 Introduction

Many well-known data classification methods perform poorly when only a small fraction of the training data is labeled. In this setting, *semi-supervised learning* (SSL) aims to leverage the geometry of unlabeled data to improve classification performance. In graph-based SSL, given a weighted graph $G = (V, E, w)$ whose nodes correspond to the training data under consideration (or having constructed one in a data-driven manner) and initial labels $u_0 : V_0 \to \mathbb{R}^k$ defined on a subset of the nodes, the goal is to extend u_0 to the unlabeled nodes by leveraging the structure of the graph. This can be formulated, for example, as an energy minimization problem on the nodes. Possibly the first method within this framework, *Laplace learning*, uses Dirichlet energy and thereby identifies a harmonic extension of u_0 (see Zhu et al. (2003); Zhu (2005)).

Over the ensuing twenty years, this line of work has matured into a mathematical theory connecting discrete objectives like that of Laplace learning to their continuum limits and Euler–Lagrange equations on the underlying data manifold; in particular, one can now precisely characterize when Laplace learning succeeds or degenerates as the amount of unlabeled data grows and the labeled fraction vanishes Nadler et al. (2009); Alaoui et al. (2016); Flores et al. (2022); Calder (2019); Holtz et al. (2024).

This line of work has led to a sharpened focus on a specific regime known as the *low label-rate* setting, where only a few labels per class are available. Here, the harmonic extension approach can collapse toward (nearly) constant predictions, even when the unlabeled graph is highly informative (see, e.g., Nadler et al. (2009)). In Calder et al. (2020), the authors resolve this degeneracy by proposing a *Poisson learning* objective that replaces hard interpolation constraints with a forcing term, derived from centered labels which encode the initial label information. The resulting method is strikingly effective in the low label-rate setting, and it also interfaces naturally with discrete refinement procedures (e.g., volume-constrained thresholding and MBO iterations) that improve classification accuracy when either partial or full class cardinality information is available (see Calder et al. (2020); Jacobs et al. (2018)).

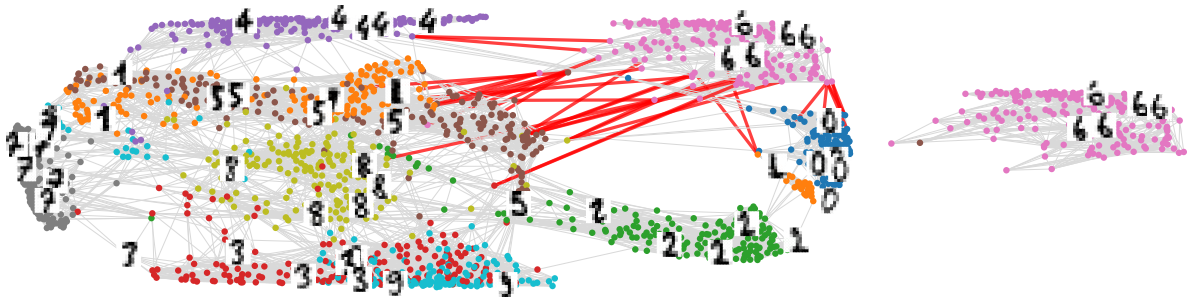


Figure 1: **(Left)** One-vs-all measure `mincut` on the Sklearn Digits dataset Dua & Graff (2017). The initially labeled nodes are shown with images overlaid. μ is given by the five images of the digit six, and ν from all of the other classes. Solving the program (3) for $p = 1$, we obtain a sparse potential φ with edge differences $|\varphi(x) - \varphi(y)|$ concentrated near either 0 (light gray) or 0.11 (red). **(Right)** Removing red edges isolates a connected component with good class separation.

In parallel, a large body of work has explored alternative regularizers and PDE limits on graphs: ℓ_p-based p-Laplacian learning and its asymptotics Alaoui et al. (2016); Flores (2018); Flores et al. (2022), sparse label propagation Jung et al. (2016), higher order regularization Zhou & Belkin (2011), Lipschitz learning and ∞-Laplacian limits Kyng et al. (2015); Calder (2019), and Hamilton–Jacobi/eikonal formulations on graphs Calder & Ettehad (2022). Taken together, these methods are powerful but somewhat fragmented: they are often presented as separate objectives with different constraints and pre- and post-processing conventions, which can obscure the shared geometric principle driving their empirical success.

Our guiding observation is that formulating SSL in the language of *measure-valued data*, rather than pointwise function extension, provides a useful organizing perspective on these methods. From this perspective, several familiar objectives can be viewed as instances of a parameterized family with related geometric structure. In this view, rather than forcing node representations to strictly interpolate training labels, the learner optimizes *measure separation*, i.e., finding optimal cuts or transport plans between class distributions. Concretely, instead of encoding the labeled set as pointwise constraints, we encode each class by a probability measure on the nodes.

With this motivation in mind, in this paper we introduce *p-conductance learning*. By encoding classes as probability measures, we pose SSL as a family of optimization programs that seek a potential φ with minimal p-Dirichlet energy subject to an affine measure separation constraint; see Equation (3) and Equation (6). At $p = 2$, this shares Poisson learning's Laplacian pseudoinverse operator but differs in how label information is encoded and read out: Poisson learning uses centered-label forcing, whereas our method imposes an affine measure-separation constraint and then decodes the resulting class potentials by argmax or constrained assignment. For classification, the learned potential acts as a score function: in the binary case, larger values favor μ over ν, and in the multiclass case the collection of one-vs-all potentials provides per-class scores for each vertex. The parameter p continuously tunes the inductive bias from sparse, cut-like potentials ($p \approx 1$) to smooth, resistance-like potentials ($p = 2$), and then to bounded gradient potentials ($p \to \infty$). Figure 2 visualizes representative potentials across this range.

1.1 Our contributions.

Theoretical contributions. Our results aim to address the question: what is the driving geometric structure behind the p-conductance program, and how does it connect to existing graph-based optimization problems? Our framework reveals that several disparate graph optimization problems are actually instances of gauge duality for linearly constrained norm minimization problems Freund (1987); Friedlander et al. (2014).

We explicitly characterize the driving geometric structure behind the p-conductance program for specific regimes:

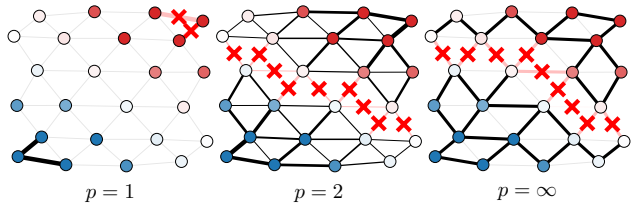


Figure 2: Representative binary p-conductance potentials φ solving $\mathcal{C}_p(\mu, \nu)$ for $p \in [1, \infty]$. The blue and red node masses indicate the class measures μ and ν, respectively, with opacity proportional to magnitude. Edge width is proportional to $|\varphi(x) - \varphi(y)|$. "X" symbols mark edges crossing the mean-threshold partition $\{\varphi \leq T\}$ and $\{\varphi > T\}$, where T is the mean of φ.

- $p = 1$: We show the $p = 1$ objective aligns with a generalized measure minimum cut program, admitting a randomized thresholding representation; see Theorem 3.4 and Theorem 3.5. This encourages sparse decision boundaries.
- $p = 2$: The optimal value is the reciprocal of a natural notion of effective resistance between class measures. We highlight the link between our 2-conductance learning to Poisson learning through the shared pseudoinverse characterization in Theorem 3.2, Corollary 3.9, and Equation (13).
- $p = \infty$: The objective connects directly to the 1-Wasserstein distance on graphs; see Theorem 3.3 and Corollary 3.10.

Algorithmic contributions. In the case of $p = 2$, the resulting program admits a closed form characterization through the Laplacian pseudoinverse, but for general p, the program does not admit a closed form solution. In particular the cases $p = 1$ and $p = \infty$ are nonsmooth and not strictly convex. To this end, we develop an algorithmic method that remains valid for generic choices of p. We develop a highly scalable inexact augmented Lagrangian method (SSNAL, Algorithm 2). By applying a semismooth Newton-CG iteration requiring only sparse incidence operations, our method scales linearly with the number of edges, making arbitrary p-conductance learning practical for real-world datasets.

Experimental contributions. We demonstrate our approach on citation networks (Cora, Citeseer, Pubmed) and image datasets (MNIST, CIFAR-10). Empirically, our method is consistently competitive in the low-label regime and attains the best or tied-best accuracies in many reported settings, particularly for $p = 2$ and when class cardinality information is available. These experiments also motivate a clear distinction among unconstrained decoding, constrained assignment, and constrained assignment with MBO refinement.

Organization. In Section 3 we analyze the theoretical structure of the p-conductance program in the binary setting. In Section 3.1 we characterize $\mathcal{C}_p(\mu, \nu)$ through cut, flow, resistance, and transport formulations for $p \in \{1, 2, \infty\}$ and state Theorems 3.1 to 3.3. In Section 3.2 we derive stability bounds for the $p = 2$ solution under measure perturbations and diffusion preprocessing. In Section 4 we present the optimization framework used to solve the p-conductance program for general p. In Section 4.1 we develop the SSNAL and SSNCG method, including reduced subproblems, generalized Newton systems, and convergence guarantees. In Section 4.2 we present multiclass decoding with assignment constraints and the conductance MBO refinement. In Section 5 we report empirical evaluation across citation, vision, noisy label, and weak supervision regimes. In Section 6 we remark on some future directions for the work presented herein.

2 Background and Problem Formulation

2.1 Preliminaries

Let $G = (V, E, w)$ be a weighted graph whose vertices represent entries in the dataset and whose edge weights encode affinities between their features. Let $n := |V|$ and $m := |E|$. We denote by $\mathscr{P}(V)$ the simplex of probability measures on V.

In *graph-based SSL*, one begins by constructing a weighted graph G, then one encodes training labels on a small subset $V_0 \subseteq V$ by a one-hot label vector $u_0 : V_0 \to \mathbb{R}^k$, where k denotes the number of classes, and then aims to extend this function to the remaining vertices by solving an energy minimization problem.

Laplace learning Zhu et al. (2003); Zhu (2005) consists of solving the optimization program

$$\min_{\substack{u:V\to\mathbb{R}^k \\ u|_{V_0}=u_0}} \sum_{\{x,y\}\in E} w_{xy}\|u(x)-u(y)\|_2^2. \tag{1}$$

If $u_\star$ achieves the infimum in Equation (1), then the predicted label for vertex $x \in V \setminus V_0$ is usually taken to be $\arg\max_{i\in[k]} u_\star^{(i)}(x)$, where $u_\star^{(i)}(x)$ is the i-th coordinate of $u_\star(x) \in \mathbb{R}^k$. Poisson learning, which aimed to resolve the degeneracy of Laplace learning in the low-label rate regime, replaced hard constraints with a zero-mean forcing term. Given an initial label matrix $u_0 : V_0 \to \mathbb{R}^k$, where $\ell := |V_0|$ is the number of labeled vertices, one writes $\overline{u_0} := \frac{1}{\ell}\sum_{x\in V_0} u_0(x) \in \mathbb{R}^k$. One then computes a function $u : V \to \mathbb{R}^k$ by solving the graph Poisson equation

$$Lu(x) \;=\; \sum_{y\in V_0} \big(u_0(y)-\overline{u_0}\big)\,\mathbf{1}_y(x), \qquad x \in V, \tag{2}$$

where L is the graph Laplacian operator, and $\mathbf{1}_y \in \mathbb{R}^V$ denotes the standard basis vector at vertex y. The solution to Equation (2) becomes unique once we impose a normalization, for example the mean-zero condition $\sum_{x\in V} u(x) = 0$. Under this convention, the solution $u_\star$ can be computed via the Moore–Penrose pseudoinverse of L, denoted $L^\dagger$:

$$u_\star^{(j)} = L^\dagger \sum_{y\in V_0} \big(u_0^{(j)}(y) - \overline{u_0}^{(j)}\big)\mathbf{1}_y, \quad j \in [k].$$

As in Laplace learning, class predictions may be taken as $\arg\max_{i\in[k]} u_\star^{(i)}(x)$.

2.2 The p-Conductance Framework

Our framework replaces pointwise function extension with measure separation. We begin with the binary problem, which contains the core affine measure-separation idea, and then record the multiclass one-vs-all extension used later in the paper.

In the binary setting, given initially labeled data $u_0 : V_0 \to \mathbb{R}^2$, we represent each class by a probability measure and define

$$\mu(x) = \frac{u_0^{(1)}(x)\mathbf{1}_{V_0}(x)}{\sum_{x'\in V_0} u_0^{(1)}(x')}, \text{ and } \nu(x) = \frac{u_0^{(2)}(x)\mathbf{1}_{V_0}(x)}{\sum_{x'\in V_0} u_0^{(2)}(x')}, \quad x \in V.$$

Here $\mathbf{1}_{V_0}$ denotes the indicator of the labeled subset V_0.

We enforce an affine separation constraint, $\varphi^\top(\mu-\nu) = 1$, which imposes a normalized separation between the two labeled measures (in rough terms, we might take $\varphi \approx 1$ near the support of μ and $\varphi \approx 0$ near the support of ν). We seek to minimize the p-Dirichlet energy subject to this constraint:

$$\mathcal{C}_p(\mu,\nu) = \begin{cases} \min\limits_{\substack{\varphi:V\to\mathbb{R} \\ \varphi^\top(\mu-\nu)=1}} \Big(\sum_{\{x,y\}\in E} w_{xy}|\varphi(x)-\varphi(y)|^p\Big)^{1/p}, & \text{if } 1 \le p < \infty, \\ \min\limits_{\substack{\varphi:V\to\mathbb{R} \\ \varphi^\top(\mu-\nu)=1}} \max_{\{x,y\}\in E} w_{xy}|\varphi(x)-\varphi(y)|, & \text{if } p = \infty. \end{cases} \tag{3}$$

The binary p-conductance program in Equation (3) is convex and already captures the main geometric mechanism studied throughout the paper. For multiclass prediction, we use a one-vs-all reduction built from class measures. Given an initial k-class label vector $u_0 : V_0 \to \mathbb{R}^k$ with $\sum_{x\in V_0} u_0^{(j)}(x) > 0$ for each $j \in [k]$, we

define for each $1 \leq j \leq k$ the probability measure

$$\mu_j(x) = \frac{u_0^{(j)}(x)\mathbf{1}_{V_0}(x)}{\sum_{x' \in V_0} u_0^{(j)}(x')}, \quad x \in V. \tag{4}$$

Similarly, we define the "one-vs-all" probability measure

$$\nu_j(x) = \frac{1}{k-1} \sum_{\ell \in [k] \setminus \{j\}} \mu_\ell(x), \quad x \in V. \tag{5}$$

Solving the binary problem once for each pair (μ_j, ν_j) yields class potentials that are later assembled into a score matrix and decoded as described in Section 4.2. Equivalently, for $1 \leq p < \infty$ one may write the separable multiclass one-vs-all objective as

$$\min_{\varphi: V \to \mathbb{R}^k} \sum_{j=1}^{k} \left(\sum_{\{x,y\} \in E} w_{xy} |\varphi^{(j)}(x) - \varphi^{(j)}(y)|^p \right)^{1/p} \quad \text{subject to} \quad (\varphi^{(j)})^\top (\mu_j - \nu_j) = 1,\ j = 1, \ldots, k. \tag{6}$$

which is equivalent to solving the k binary programs $\mathcal{C}_p(\mu_j, \nu_j)$ independently and stacking their minimizers as columns. In the case $p = \infty$, we likewise sum the coordinatewise binary seminorms:

$$\min_{\varphi: V \to \mathbb{R}^k} \sum_{\ell=1}^{k} \max_{\{x,y\} \in E} w_{xy} |\varphi^{(\ell)}(x) - \varphi^{(\ell)}(y)| \quad \text{subject to} \quad (\varphi^{(j)})^\top (\mu_j - \nu_j) = 1,\ j = 1, \ldots, k. \tag{7}$$

The binary problem Equation (3) and the multiclass one-vs-all extensions Equations (6) and (7) are convex programs that seek potentials with minimal p-Dirichlet energy subject to affine measure separation constraints. The parameter p continuously tunes the inductive bias from sparse, cut-like potentials ($p \approx 1$) to smooth, resistance-like potentials ($p = 2$), and then to bounded gradient potentials ($p \to \infty$). The term "p-conductance," which underlies the notation in Equation (3), is motivated by the fact that the optimal value of the binary program is the reciprocal of a natural p-resistance between μ and ν (see Theorem 3.2).

This shift in viewpoint has three immediate consequences.

First, Poisson-style label forcing is subsumed by an affine measure separation constraint. Laplace learning imposes hard interpolation constraints, whereas Poisson learning replaces those constraints with a forcing term derived from centered labels Calder et al. (2020). In our measure-based approach, the label information is instead encoded through an affine constraint coupling a potential $\varphi \in \mathbb{R}^V$ to the signed label measure. In particular, the constraint $\varphi^\top(\mu - \nu) = 1$ imposes a normalized separation between the two labeled measures (in rough terms, we might take $\varphi \approx 1$ near the support of μ and $\varphi \approx 0$ near the support of ν). In this sense, the framework most directly generalizes Poisson learning in the quadratic case: Theorem 3.2 shows that when $p = 2$ the same pseudoinverse operator $L^\dagger$ appears, but centered-label forcing is replaced by the affine measure-separation constraint.

Second, measure separation exposes the geometry of learning cleanly. Once the data are represented by class measures, the choice of cost function for measure separation leads to interpretable geometric biases that connect to a wide variety of graph optimization problems. For $p = 1$, the resulting objective aligns with cut sparsity and connects directly to the extensive literature on minimum and randomized edge cuts Shi & Malik (2000); Boykov & Veksler (2006). For $p = 2$, the optimizer can be expressed using the Laplacian pseudoinverse, which is then related to effective resistance Ben-Israel & Charnes (1963); Robertson et al. (2024); see also the body of work concerning the more general family of p-resistances and their phase transitions Alamgir & Luxburg (2011). For $p = \infty$, the same framework connects to 1-Wasserstein distance on graphs Beckmann (1952); Peyré et al. (2019); Essid & Solomon (2018); Robertson et al. (2024). The unifying mechanism behind these equivalences is gauge duality for linearly constrained norm problems Freund (1987); Friedlander et al. (2014).

Third, working with measures offers principled design of pre- and post-processing pipelines. Practical implementations of Poisson learning and related methods often include: (a) label preprocessing/reweighting

before solving the learning problem, and (b) *rounding/refining* continuous predictions using class-size priors (often via MBO-like iterations) Calder et al. (2020); Jacobs et al. (2018). In our setting, (a) can naturally include the application of any measure-preserving operator to class measures (e.g., heat kernel diffusion); (b) is modeled as optimization over a transportation polytope, where the row constraints assign one unit of mass from each vertex and the column constraints encode class-size marginals Loera & Kim (2013). From this transportation viewpoint, taking ϵ large in Equation (25) makes the class-size bounds vacuous, so the decoding LP decouples across rows and each vertex is assigned to its largest score, while p-conductance-MBO alternates Laplacian smoothing with projection onto the same feasible set.

3 Theoretical properties of the p-conductance program

We begin with the case $p = 1$.

Theorem 3.1 (Generalized `maxflow`/`mincut` characterization (Informal statement))**.** *Let $G = (V, E, w)$ be connected, and let $\mu, \nu \in \mathscr{P}(V)$ be fixed with $\mu \neq \nu$. Then the optimal value of the program $\mathcal{C}_1(\mu, \nu)$ can be recast as the optimal value of a measure minimum cut program, whose dual is a capacitated maximum flow problem with divergence proportional to $\mu - \nu$.*

Theorem 3.1 is stated in full in Section 3.1 (see Theorem 3.4). In Theorem 3.5, we also show that the case $p = 1$ admits a randomized thresholding representation as well.

Theorem 3.2 (2-Conductance and effective resistance)**.** *Let $G = (V, E, w)$ be connected, and let $\mu, \nu \in \mathscr{P}(V)$ be fixed with $\mu \neq \nu$. Then the optimal value of the program $\mathcal{C}_2(\mu, \nu)$ is the reciprocal of the effective resistance between μ and ν:*

$$\mathcal{C}_2(\mu, \nu)^2 = \frac{1}{(\mu - \nu)^\top L^\dagger (\mu - \nu)},$$

and the mean-zero minimizing potential is given by

$$\varphi_\star = \frac{L^\dagger(\mu - \nu)}{(\mu - \nu)^\top L^\dagger (\mu - \nu)}.$$

Theorem 3.2 is stated in full in Section 3.1 (see Corollary 3.9). We remark that Theorem 3.2 implicitly highlights the link between 2-conductance learning and Poisson learning, since both methods involve the same pseudoinverse operator $L^\dagger$ and differ only in the choice of label preprocessing and choice of decision rule. Our third main result considers the case $p = \infty$, as follows.

Theorem 3.3 (Transportation and $p = \infty$ (Informal statement))**.** *Let $G = (V, E, w)$ be connected, and let $\mu, \nu \in \mathscr{P}(V)$ be fixed with $\mu \neq \nu$. Then the optimal value of the program $\mathcal{C}_\infty(\mu, \nu)$ is the reciprocal of the 1-Wasserstein distance between μ and ν on G:*

$$\mathcal{C}_\infty(\mu, \nu) = \frac{1}{\mathcal{W}_{1,w^{-1}}(\mu, \nu)}.$$

Here, $\mathcal{W}_{1,w^{-1}}(\cdot, \cdot)$ denotes the 1-Wasserstein distance on $\mathscr{P}(V)$ with respect to the shortest path metric induced by the edge weights w^{-1} (see Equation (14)). Theorem 3.3 is stated in full in Section 3.1 (see Corollary 3.10). The equivalences described in Theorems 3.1 to 3.3 are all instances of gauge duality for linearly constrained norm minimization problems Freund (1987); Friedlander et al. (2014) (see Theorem 3.8).

For the sake of simplicity, in this section we analyze the program $\mathcal{C}_p(\mu, \nu)$ setting of binary classification. Our analysis then admits natural extensions and interpretations pertinent to the multiclass setting.

First we introduce some working notation. We denote by $A \in \mathbb{R}^{n \times n}$ the weighted adjacency matrix of G, and we write $D = \operatorname{diag}(d(x_1), \ldots, d(x_n)) \in \mathbb{R}^{n \times n}$ for the diagonal degree matrix, where $d(x) = \sum_{y \in V} w_{xy}$, and we write $L = D - A$ for the combinatorial graph Laplacian. Note that L is positive semidefinite and if G is connected, L has nullspace spanned by $\mathbf{1}$. We write $L^\dagger$ for the Moore–Penrose pseudoinverse of L.

We denote by E' any distinguished orientation of the edges E of G, and accordingly we define the *oriented incidence matrix* $B \in \mathbb{R}^{n\times m}$ of G according to

$$B_{x,e} = \begin{cases} 1, & \text{if } e = (x,y) \text{ for some } y \in V \\ -1, & \text{if } e = (y,x) \text{ for some } y \in V \\ 0, & \text{otherwise,} \end{cases}, \quad x \in V,\ e \in E'.$$

We define the edge weight matrix $W := \operatorname{diag}(w_e)_{e\in E'} \in \mathbb{R}^{m\times m}$, so $L = BWB^\top$. We also write E'' for the set of both orientations of each edge, and write $\widetilde{B} = (B, -B)$. If $A_1, A_2 \subseteq V$, we write $w(A_1, A_2)$ for the total weight of edges with one endpoint in A_1 and the other in A_2.

3.1 Cuts, transportation, and effective resistance

In this subsection we state Theorems 3.1 to 3.3, which identify the geometric structure of the program $\mathcal{C}_p(\mu,\nu)$ in the cases $p = 1$, $p = 2$, and $p = \infty$. In particular, we show that the program $\mathcal{C}_1(\mu,\nu)$ is equivalent to a measure minimum cut problem, whose dual is a capacitated maximum flow problem with divergence proportional to $\mu - \nu$; that $\mathcal{C}_2(\mu,\nu)$ is the reciprocal of an effective resistance between μ and ν; and that $\mathcal{C}_\infty(\mu,\nu)$ is the reciprocal of a 1-Wasserstein distance between μ and ν on G.

Theorem 3.4 (Generalized `maxflow`/`mincut`)**.** *Let $G = (V, E, w)$ be connected, and let $\mu, \nu \in \mathscr{P}(V)$ be fixed with $\mu \neq \nu$. Then $\mathcal{C}_1(\mu,\nu)$ equals the optimal value of the linear program*

$$\texttt{mincut}(\mu,\nu) = \begin{cases} \textit{minimize} & \sum_{(x,y)\in E''} w_{xy} K(x,y) \\ \textit{subject to} & K \in \mathbb{R}^{2m}, \\ & \varphi \in \mathbb{R}^n, \\ & K(x,y) \geq 0, \ \textit{for each } (x,y) \in E'', \\ & K(x,y) \geq \varphi(x) - \varphi(y), \ \textit{for each } (x,y) \in E'', \\ & (\mu-\nu)^\top \varphi \geq 1, \end{cases} \tag{8}$$

whose dual is

$$\texttt{maxflow}(\mu,\nu) = \begin{cases} \textit{maximize} & f \\ \textit{subject to} & J \in \mathbb{R}^{2m}, \\ & f \in \mathbb{R}, \\ & f \geq 0, \\ & 0 \leq J(x,y) \leq w_{xy}, \ \textit{for each } (x,y) \in E'', \\ & \widetilde{B} J = f(\mu - \nu), \end{cases} \tag{9}$$

In particular,

$$\mathcal{C}_1(\mu,\nu) = \texttt{mincut}(\mu,\nu) = \texttt{maxflow}(\mu,\nu). \tag{10}$$

Theorem 3.4 is proved in Appendix A. Here J is indexed by the oriented edge set E'', and the constraint $\widetilde{B}J = f(\mu-\nu)$ means that the net outflow minus inflow at each vertex equals $f(\mu-\nu)$, relative to the chosen reference orientation used to define B. We also remark that the programs in Equation (8) and Equation (9) generalize the classical vertex `mincut` and `maxflow` programs, since if $\mu = \mathbf{1}_s$ and $\nu = \mathbf{1}_t$ for some $s, t \in V$, then the constraint $(\mu-\nu)^\top\varphi \geq 1$ reduces to $\varphi(s) - \varphi(t) \geq 1$, and the resulting programs are exactly the classical `mincut` and `maxflow` linear programs for the s-t cut problem.

The next theorem identifies $\mathcal{C}_1(\mu,\nu)$ as the optimal expected cut-to-mass ratio induced by thresholding a feasible potential φ at a random level.

Theorem 3.5 (Randomized cut representation)**.** *Let $G = (V, E, w)$ be connected, and let $\mu, \nu \in \mathscr{P}(V)$ be fixed with $\mu \neq \nu$. Let $\varphi \in \mathbb{R}^n$ be fixed satisfying $\varphi \geq 0$ and $\varphi^\top(\mu-\nu) = 1$. Let $T \sim \operatorname{Unif}([0, \|\varphi\|_\infty])$ and write*

$$A_T = \{x \in V : \varphi(x) \geq T\}, \qquad B_T = V \setminus A_T.$$

Then

$$\mathbb{E}_T\Big(\sum_{x\in A_T}(\mu(x)-\nu(x))\Big)=\frac{1}{\|\varphi\|_\infty}>0,$$

and hence

$$\mathcal{C}_1(\mu,\nu)=\min_{\substack{\varphi\geq 0\\ \varphi^\top(\mu-\nu)=1}}\frac{\mathbb{E}_T[w(A_T,B_T)]}{\mathbb{E}_T\big(\sum_{x\in A_T}(\mu(x)-\nu(x))\big)}.$$

Theorem 3.5 is proved in Appendix A.

Remark 3.6 (Two-level reduction for $p=2$). The quadratic program $\mathcal{C}_2(\mu,\nu)$ may also be partially formulated in terms of a cut-like objective. Recall that for the binary program,

$$\mathcal{C}_2(\mu,\nu)=\min_{\substack{\varphi\in\mathbb{R}^n\\ \varphi^\top(\mu-\nu)=1}}\left(\sum_{\{x,y\}\in E}w_{xy}|\varphi(x)-\varphi(y)|^2\right)^{1/2}.$$

Suppose we fix $A\subseteq V$ and *restrict* to potentials $\varphi=a\mathbf{1}_A+b\mathbf{1}_{A^c}$. Define

$$s_A:=\sum_{x\in A}(\mu(x)-\nu(x)).$$

Only edges crossing (A,A^c) contribute to the Dirichlet term, hence

$$\sum_{\{x,y\}\in E}w_{xy}|\varphi(x)-\varphi(y)|^2=(a-b)^2\,w(A,A^c).$$

Since $\sum_{x\in V}(\mu(x)-\nu(x))=0$, we also have

$$\varphi^\top(\mu-\nu)=a\sum_{x\in A}(\mu(x)-\nu(x))+b\sum_{x\in A^c}(\mu(x)-\nu(x))=(a-b)s_A,$$

so feasibility is equivalent to $(a-b)s_A=1$. In particular, if $s_A=0$ then no two-level potential on this partition is feasible; if $s_A\neq 0$, then $a-b=1/s_A$, and the restricted objective becomes

$$\frac{\sqrt{w(A,A^c)}}{|s_A|},$$

equivalently, after squaring and recalling that this is only a comparison over the restricted class of two-level potentials, we have

$$\mathcal{C}_2(\mu,\nu)^2\leq\min_{\substack{\varnothing\neq A\neq V\\ (\mu-\nu)^\top\mathbf{1}_A\neq 0}}\frac{w(A,A^c)}{\big(\sum_{x\in A}(\mu(x)-\nu(x))\big)^2}. \tag{11}$$

Moving on to the cases $p=2,\infty$, we first observe that the conductance programs $\mathcal{C}_p$ are dual to a certain family of optimal transportation problems on graphs. For any positive coefficient vector $a=(a_e)_{e\in E}$, for $\mu,\nu\in\mathscr{P}(V)$, and for $p\in[1,\infty]$, we define the *Beckmann transportation program*

$$\mathcal{B}_{a,p}(\mu,\nu):=\begin{cases}\displaystyle\min_{\substack{J\in\mathbb{R}^m\\ BJ=\mu-\nu}}\left(\sum_{e\in E}a_e|J(e)|^p\right)^{1/p}, & 1\leq p<\infty,\\ \displaystyle\min_{\substack{J\in\mathbb{R}^m\\ BJ=\mu-\nu}}\max_{e\in E}a_e|J(e)|, & p=\infty.\end{cases} \tag{$\mathcal{B}_{a,p}$}$$

Here the subscript records the coefficients appearing directly in the transportation objective. Thus $\mathcal{B}_{w^{-1},1}$ means that the Beckmann cost uses coefficients w_e^{-1}, not the conductances w_e. The following basic lemma, stated without proof, is used to derive the duality between $\mathcal{C}_p$ and this family of Beckmann programs.

Lemma 3.7 (Weighted norm duality)**.** *Let $w_1, \dots, w_m > 0$ and define*

$$\|x\|_{w,p} = \begin{cases} \left(\sum_{i=1}^m w_i |x_i|^p\right)^{1/p}, & 1 \le p < \infty, \\ \max_{1 \le i \le m} w_i |x_i|, & p = \infty, \end{cases}, \quad x \in \mathbb{R}^m.$$

Then the dual norm $\|\cdot\|_{w,p}^*$ to $\|\cdot\|_{w,p}$ is given by

$$\|y\|_{w,p}^* = \begin{cases} \max_{1 \le i \le m} \dfrac{|y_i|}{w_i}, & p = 1, \\ \|y\|_{w^{1-q},q}, & 1 < p < \infty,\ \frac{1}{p} + \frac{1}{q} = 1, \\ \|y\|_{w^{-1},1}, & p = \infty. \end{cases}$$

Theorem 3.8 (Beckmann duality for p-conductance)**.** *Let $G = (V, E, w)$ be a connected graph and let $\mu, \nu \in \mathscr{P}(V)$ with $\mu \neq \nu$. Then*

$$\mathcal{C}_p(\mu, \nu) = \begin{cases} \dfrac{1}{\mathcal{B}_{w^{-1},\infty}(\mu, \nu)}, & p = 1, \\ \dfrac{1}{\mathcal{B}_{w^{1-q},q}(\mu, \nu)}, & 1 < p < \infty,\ \frac{1}{p} + \frac{1}{q} = 1, \\ \dfrac{1}{\mathcal{B}_{w^{-1},1}(\mu, \nu)}, & p = \infty. \end{cases}$$

Theorem 3.8 is proved in Appendix A. Theorem 3.8 offers two immediate corollaries. First we recall the *effective resistance* between vertices $x, y \in V$:

$$\mathcal{R}_{\text{eff}}(x, y) = (\mathbf{1}_x - \mathbf{1}_y)^\top L^\dagger (\mathbf{1}_x - \mathbf{1}_y).$$

For a more general mean zero source term, the same quadratic form extends this notion. For $\mu, \nu \in \mathscr{P}(V)$, set

$$\mathcal{R}_{\text{eff}}(\mu, \nu) := (\mu - \nu)^\top L^\dagger (\mu - \nu),$$

which recovers the vertex case when $\mu = \mathbf{1}_x$ and $\nu = \mathbf{1}_y$. The next corollary identifies $\mathcal{C}_2(\mu, \nu)$ exactly in terms of this effective resistance quantity and gives the mean-zero minimizing potential in closed form.

Corollary 3.9. *Let $G = (V, E, w)$ be connected, and let $\mu, \nu \in \mathscr{P}(V)$ be fixed with $\mu \neq \nu$. Then*

$$\mathcal{C}_2(\mu, \nu)^2 = \frac{1}{\mathcal{R}_{\text{eff}}(\mu, \nu)} = \frac{1}{(\mu - \nu)^\top L^\dagger (\mu - \nu)}. \tag{12}$$

Moreover, the mean-zero minimizing potential for the program $\mathcal{C}_2(\mu, \nu)$ is given by

$$\varphi_\star = \frac{L^\dagger (\mu - \nu)}{(\mu - \nu)^\top L^\dagger (\mu - \nu)}. \tag{13}$$

We offer a short proof of Corollary 3.9 in Appendix A. For the second corollary, let $d_{w^{-1}}(\cdot, \cdot)$ denote the weighted shortest path distance on V induced by reciprocal edge weights w_e^{-1}. For $\mu, \nu \in \mathscr{P}(V)$, recall the graph 1-*Wasserstein distance* given by the linear program

$$\mathcal{W}_{1,w^{-1}}(\mu, \nu) := \min_{\pi \in \Pi(\mu,\nu)} \sum_{x,y \in V} d_{w^{-1}}(x, y)\, \pi(x, y), \tag{14}$$

where the minimum is over couplings π of μ and ν. Concretely,

$$\Pi(\mu, \nu) := \left\{ \pi \in \mathbb{R}_{\ge 0}^{n \times n} : \sum_{y \in V} \pi(x, y) = \mu(x),\ \sum_{x \in V} \pi(x, y) = \nu(y) \text{ for all } x \in V,\ y \in V \right\}.$$

Because $(V, d_{w^{-1}})$ is a connected geodesic metric space, Lagrangian duality (equivalently Kantorovich–Rubinstein duality) then yields that the static Kantorovich formulation Equation (14) and the edge flow formulation $\mathcal{B}_{w^{-1},1}$ have the same value;

$$\mathcal{W}_{1,w^{-1}}(\mu,\nu) = \mathcal{B}_{w^{-1},1}(\mu,\nu).$$

See, for example, Peyré et al. (2019). This leads to the following corollary to Theorem 3.8.

Corollary 3.10. *Let $G = (V, E, w)$ be a connected graph and let $\mu, \nu \in \mathscr{P}(V)$ with $\mu \neq \nu$. Then*

$$\mathcal{C}_\infty(\mu,\nu) = \frac{1}{\mathcal{W}_{1,w^{-1}}(\mu,\nu)}.$$

Corollary 3.10 follows immediately from the preceding observations and Theorem 3.8.

3.2 Diffusion and stability for $p = 2$

In this subsection we investigate the stability of solutions to the quadratic program $\mathcal{C}_2(\mu,\nu)$ under perturbations to the label measures μ and ν, and under diffusion-based preprocessing. Specifically, if the class measures are stored in the columns of a matrix $Y \in \mathbb{R}^{n\times k}$, we can consider preprocessing via heat kernel diffusion before solving the program $\mathcal{C}_p$:

$$Y_{\text{diff}} = e^{-tL}Y, \qquad t \geq 0. \tag{15}$$

This is exactly the preprocessing step used in Algorithm 1, where each one-vs-all class measure is diffused before the conductance subproblems are solved. Since $L\mathbf{1} = 0$, the operator e^{-tL} preserves the mass in each column of Y. For a single binary component with source term $b = \mu - \nu$, the unnormalized $p = 2$ potential is $\varphi = L^\dagger b$. If the labels are perturbed by a mean-zero vector η, then after diffusion we obtain

$$\widetilde{\varphi}_t = L^\dagger e^{-tL}(b + \eta).$$

There exists a natural tradeoff between blurring the clean signal b and damping the perturbation η. The theorem below makes this tradeoff precise by giving an explicit upper bound on the perturbation error after diffusion.

Theorem 3.11 (Diffusion improves the worst-case perturbation bound)**.** *Assume that G is connected, let $\lambda_2 > 0$ be the smallest positive eigenvalue of L, and let $b = \mu - \nu$. Define*

$$\varphi = L^\dagger b, \qquad \widetilde{\varphi}_t = L^\dagger e^{-tL}(b + \eta),$$

where $\eta \in \mathbb{R}^n$ satisfies $\mathbf{1}^\top \eta = 0$. Then, for every $t \geq 0$,

$$\|\varphi - \widetilde{\varphi}_t\|_2 \leq t\|b\|_2 + \lambda_2^{-1} e^{-t\lambda_2}\|\eta\|_2. \tag{16}$$

In particular, if $\|\eta\|_2 > \|b\|_2$ and

$$0 < t < \frac{1}{\lambda_2}\left(\frac{\|\eta\|_2}{\|b\|_2} - 1\right),$$

then

$$t\|b\|_2 + \lambda_2^{-1} e^{-t\lambda_2}\|\eta\|_2 < \lambda_2^{-1}\|\eta\|_2,$$

which strictly improves the naive bound $\|L^\dagger \eta\|_2 \leq \lambda_2^{-1}\|\eta\|_2$.

Theorem 3.11 is proved in Appendix A.

Remark 3.12. The condition $\|\eta\|_2 > \|b\|_2$ is interpretable in a model of data perturbation via label flips. If $\mu = \frac{1}{r}\mathbf{1}_A$ and $\nu = \frac{1}{r}\mathbf{1}_B$ for disjoint labeled sets A, B of equal size r, and if r_1 labels from A and r_2 labels from B are flipped, then

$$\eta = \frac{2}{r}\big(\mathbf{1}_{B'} - \mathbf{1}_{A'}\big), \qquad \frac{\|\eta\|_2}{\|\mu - \nu\|_2} > 1 \iff r_1 + r_2 > \frac{r}{2}.$$

Thus once a nontrivial fraction of the labels are corrupted, a small diffusion time provably improves the worst-case ℓ_2 error bound.

Algorithm 1 p-conductance learning

Input: weighted graph $G = (V, E, w)$, initial labels $u_0 : V_0 \to \mathbb{R}^k$, parameter $p \in [1, \infty]$, diffusion time $t \geq 0$
Input (optional): class-size prior $\beta \in \mathbb{Z}^k_{\geq 0}$, tolerance $\epsilon \geq 0$
for $j = 1, \dots, k$ **do**
  Build one-vs-all class measures (μ_j, ν_j) as in Equations (4) and (5)
  $\mu_j \leftarrow e^{-tL}\mu_j$
  $\nu_j \leftarrow e^{-tL}\nu_j$
  Set source term $b_j \leftarrow \mu_j - \nu_j$
  Solve binary conductance subproblem $\mathcal{C}_p(\mu_j, \nu_j)$ using SSNAL (Algorithm 2), output potential $\varphi_j \in \mathbb{R}^V$
end for
Form score matrix $\Phi = [\varphi_1, \dots, \varphi_k] \in \mathbb{R}^{n \times k}$
if no class size prior is used **then**
  Predict labels via argmax decision rule Equation (26)
else
  Decode by solving the constrained assignment LP Equation (25) with (β, ϵ)
  if refinement is enabled **then**
    Run p-conductance-MBO updates (Algorithm 4) initialized from decoded assignments
  end if
end if
Output: final class predictions (and potentials Φ)

4 Algorithms for solving the p-conductance program

In this section we describe an efficient algorithm for solving the conductance program $\mathcal{C}_p$ for general $p \in [1, \infty]$. In the case of $p = 2$, the program $\mathcal{C}_2(\mu, \nu)$ admits a closed form characterization through $L^\dagger$, but for general p, the program does not admit a closed form solution; in particular the cases $p = 1$ and $p = \infty$ are nonsmooth and not strictly convex. To this end, we develop an algorithmic method that remains valid for generic choices of p. For clarity, we summarize the full training/inference pipeline in Algorithm 1, including the preprocessing and postprocessing stages used throughout the experiments. In Section 4.1 we give a detailed description of the semismooth Newton augmented Lagrangian method for $\mathcal{C}_p$. In Section 4.2 we describe multiclass assignment, class-size-aware postprocessing, and an MBO-type refinement step.

For k-class prediction, for each $1 \leq j \leq k$, we solve the one-vs-all problem $\mathcal{C}_p(\mu_j, \nu_j)$ as defined in Equations (4) and (5), yielding class potentials φ_j, and assemble these into the score matrix $\Phi = [\varphi_1, \dots, \varphi_k] \in \mathbb{R}^{n \times k}$. From there we model class decision as a constrained assignment problem rather than as an ad hoc thresholding rule. Without class-size information, decoding reduces to the rowwise argmax rule Equation (26). With cardinality priors, we solve an assignment linear program Equation (25), and optionally apply p-conductance-MBO (Algorithm 4), which alternates a Laplacian smoothing step with projection onto the same polytope to minimize $\alpha \operatorname{tr}(P^\top L P) - \langle \Phi, P \rangle$ under assignment constraints.

4.1 Semismooth Newton augmented Lagrangian method for $\mathcal{C}_p$

By reformulating the program on a certain subspace and introducing an auxiliary edge variable, we build an inexact augmented Lagrangian method (Algorithm 2) based on an augmented objective (see Equation (18)). The outer updates enforce both primal feasibility conditions, while separating the nonsmooth edge penalty from linear graph constraints. This design gives a decomposition in which each outer iteration reduces to a convex inner subproblem with explicit primal and dual residuals.

The inner subproblem is solved by first reducing to a smooth equation on the potential variable via a proximal elimination step (Lemma 4.1), and then applying a semismooth Newton–CG iteration (Algorithm 3). Matrix–vector products of the generalized Newton matrix use only sparse incidence operations and can be carried out in time linear in the number of edges (Remark 4.3). We prove local superlinear convergence of the semismooth Newton step for $p \in \{1, 2, \dots\}$ and $p = \infty$ (Theorem 4.5). Moreover, under summable

inexactness conditions, we prove global convergence of the outer augmented Lagrangian sequence to primal solutions (Theorem 4.6).

For $p = 2$, Corollary 3.9 gives the mean-zero minimizing potential in closed form. For general p, however, the conductance program is convex but neither smooth nor strictly convex at the endpoints $p = 1, \infty$, and no comparably explicit formula is available. We therefore solve the program $\mathcal{C}_p$ by employing an augmented Lagrangian method, with each inner subproblem handled by a semismooth Newton–conjugate gradient step. The multiclass one-vs-all extensions Equations (6) and (7) are separable across class coordinates, so we treat them one column at a time, with source terms $b_j = \mu_j - \nu_j$, $j = 1, \dots, k$.

Fix $\mu, \nu \in \mathscr{P}(V)$ with $\mu \neq \nu$, and write $b := \mu - \nu$. Since $b^\top \mathbf{1} = 0$ and $B^\top(\varphi + c\mathbf{1}) = B^\top \varphi$ for every $c \in \mathbb{R}$, both the objective and the affine constraint are invariant under additive constants. It is therefore natural to work in the centered subspace

$$\mathcal{H} := \left\{ \varphi \in \mathbb{R}^n : \mathbf{1}^\top \varphi = 0 \right\}.$$

Every feasible potential has a unique representative in $\mathcal{H}$, so minimizing $\mathcal{C}_p(\mu, \nu)$ is equivalent to minimizing over $\mathcal{H}$. For $1 \leq p < \infty$, the map $r \mapsto r^{1/p}$ is increasing on $[0, \infty)$, so the minimizers of Equation (3) agree with those of the unrooted energy. Accordingly, we define

$$s_p(u) := \begin{cases} \displaystyle\sum_{e \in E} w_e |u_e|^p, & 1 \leq p < \infty, \\ \displaystyle\max_{e \in E} w_e |u_e|, & p = \infty, \end{cases}$$

and consider the equivalent problem

$$\begin{cases} \text{minimize} & s_p(u), \\ \text{subject to} & \varphi \in \mathcal{H}, \\ & u \in \mathbb{R}^m, \\ & b^\top \varphi = 1, \\ & B^\top \varphi - u = 0. \end{cases} \tag{17}$$

Given $\sigma > 0$, $y \in \mathbb{R}$, and $z \in \mathbb{R}^m$, we define the augmented Lagrangian

$$\mathcal{L}_\sigma(\varphi, u; y, z) = s_p(u) + y(b^\top \varphi - 1) + z^\top (B^\top \varphi - u) + \frac{\sigma}{2}(b^\top \varphi - 1)^2 + \frac{\sigma}{2}\|B^\top \varphi - u\|_2^2. \tag{18}$$

The outer iteration is an inexact augmented Lagrangian method:

$$\begin{aligned} (\varphi_{k+1}, u_{k+1}) &\approx \operatorname*{arg\,min}_{\varphi \in \mathcal{H},\ u \in \mathbb{R}^m} \mathcal{L}_{\sigma_k}(\varphi, u; y_k, z_k), \\ z_{k+1} &= z_k + \sigma_k (B^\top \varphi_{k+1} - u_{k+1}), \\ y_{k+1} &= y_k + \sigma_k (b^\top \varphi_{k+1} - 1). \end{aligned} \tag{19}$$

In Algorithm 2 we outline the resulting scheme. In practice, each primal subproblem is solved through the reduced semismooth Newton iteration described below.

Lemma 4.1 (Reduced inner problem)**.** *Fix $\sigma > 0$, $\widetilde{y} \in \mathbb{R}$, and $\widetilde{z} \in \mathbb{R}^m$. Let*

$$u_\sigma(\varphi) := \operatorname{prox}_{s_p/\sigma}\left(B^\top \varphi + \sigma^{-1} \widetilde{z}\right),$$

and define

$$f_\sigma(\varphi) := \inf_{u \in \mathbb{R}^m} \mathcal{L}_\sigma(\varphi, u; \widetilde{y}, \widetilde{z}), \qquad \varphi \in \mathcal{H}.$$

Then f_σ is convex and continuously differentiable on $\mathcal{H}$, and

$$\nabla f_\sigma(\varphi) = B\big(\widetilde{z} + \sigma(B^\top \varphi - u_\sigma(\varphi))\big) + \big(\widetilde{y} + \sigma(b^\top \varphi - 1)\big)\, b. \tag{20}$$

Moreover, $(\varphi_\star, u_\star)$ minimizes $\mathcal{L}_\sigma(\cdot, \cdot; \widetilde{y}, \widetilde{z})$ over $\mathcal{H} \times \mathbb{R}^m$ if and only if

$$u_\star = u_\sigma(\varphi_\star) \qquad \text{and} \qquad \nabla f_\sigma(\varphi_\star) = 0.$$

Algorithm 2 Semismooth Newton augmented Lagrangian (SSNAL)

Choose $(\varphi^0, u^0) \in \mathcal{H} \times \mathbb{R}^m$, $y_0 \in \mathbb{R}$, $z_0 \in \mathbb{R}^m$, and penalties $\sigma_k > 0$.
for $k = 0, 1, 2, \ldots$ until convergence **do**
  Approximately solve the primal subproblem

$$\min_{\varphi \in \mathcal{H},\ u \in \mathbb{R}^m} \mathcal{L}_{\sigma_k}(\varphi, u; y_k, z_k)$$

  to obtain (φ_{k+1}, u_{k+1}), with stopping criterion (23).
  Set $z_{k+1} \leftarrow z_k + \sigma_k(B^\top \varphi_{k+1} - u_{k+1})$.
  Set $y_{k+1} \leftarrow y_k + \sigma_k(b^\top \varphi_{k+1} - 1)$.
  Update $\sigma_{k+1} \geq \sigma_k$ if desired.
end for

Algorithm 3 Semismooth Newton–CG (SSNCG)

Fix $\sigma > 0$, $\widetilde{y} \in \mathbb{R}$, $\widetilde{z} \in \mathbb{R}^m$, and $\varphi^0 \in \mathcal{H}$.
for $t = 0, 1, 2, \ldots$ until $\|\nabla f_\sigma(\varphi^t)\|_2$ is below tolerance **do**
  Set $u^t \leftarrow \operatorname{prox}_{s_p/\sigma}(B^\top \varphi^t + \sigma^{-1}\widetilde{z})$ and $g^t \leftarrow \nabla f_\sigma(\varphi^t)$.
  Choose $Q_t \in \partial_C \operatorname{prox}_{s_p/\sigma}(B^\top \varphi^t + \sigma^{-1}\widetilde{z})$ and set $V_t \leftarrow \sigma B(I - Q_t)B^\top + \sigma bb^\top$.
  Approximately solve $(V_t + \eta_t I)d^t = -g^t$ on $\mathcal{H}$ by conjugate gradients.
  Choose $\alpha_t \in (0, 1]$ by Armijo backtracking for f_σ.
  Set $\varphi^{t+1} \leftarrow \varphi^t + \alpha_t d^t$.
end for
Return $\varphi_\star \leftarrow \varphi^t$ and $u_\star \leftarrow \operatorname{prox}_{s_p/\sigma}(B^\top \varphi_\star + \sigma^{-1}\widetilde{z})$.

Lemma 4.1 is proved in Appendix B. The inner ALM subproblem is therefore reduced to the nonsmooth equation $\nabla f_\sigma(\varphi) = 0$ on $\mathcal{H}$. Given a current iterate $\varphi^t \in \mathcal{H}$, we choose an element

$$Q_t \in \partial_C \operatorname{prox}_{s_p/\sigma}\big(B^\top \varphi^t + \sigma^{-1}\widetilde{z}\big)$$

from the Clarke generalized Jacobian and define the generalized Newton matrix

$$V_t := \sigma B(I - Q_t)B^\top + \sigma bb^\top. \tag{21}$$

We then compute a search direction $d^t \in \mathcal{H}$ from the regularized linear system

$$(V_t + \eta_t I)d^t = -\nabla f_\sigma(\varphi^t), \qquad \eta_t > 0, \tag{22}$$

approximately by conjugate gradients, and perform an Armijo line search on f_σ. The corresponding inner iteration is summarized in Algorithm 3. Here the Armijo step size is chosen as the largest $\alpha_t = \rho^{\ell_t}$, with $\rho \in (0, 1)$ and $\ell_t \in \mathbb{Z}_{\geq 0}$, such that

$$f_\sigma(\varphi^t + \alpha_t d^t) \leq f_\sigma(\varphi^t) + c\alpha_t \langle \nabla f_\sigma(\varphi^t), d^t \rangle$$

for some fixed $c \in (0, 1)$.

Lemma 4.2 (Generalized Hessian structure)**.** *Let $\sigma > 0$, $\widetilde{y} \in \mathbb{R}$, and $\widetilde{z} \in \mathbb{R}^m$ be fixed. For every $\varphi \in \mathcal{H}$ and every*

$$Q \in \partial_C \operatorname{prox}_{s_p/\sigma}\big(B^\top \varphi + \sigma^{-1}\widetilde{z}\big)\,,$$

the matrix

$$V_Q := \sigma B(I - Q)B^\top + \sigma bb^\top$$

belongs to $\partial_C \nabla f_\sigma(\varphi)$. Moreover, every such V_Q is symmetric and positive semidefinite on $\mathcal{H}$.

Lemma 4.2 is proved in Appendix B.

Remark 4.3. Every matrix–vector product with V_Q has cost $O(m)$ arithmetic operations:

$$V_Q x = \sigma B\big((I-Q)(B^\top x)\big) + \sigma b(b^\top x).$$

Since both B and $B^\top$ are sparse incidence operators, the dominant work is linear in the number of edges.

Proposition 4.4 (Strong semismoothness of the proximal map)**.** *Let $t > 0$. If $p = 1$, $p = \infty$, or $p \in \{2, 3, \ldots\}$, then the proximal map prox_{ts_p} is strongly semismooth.*

Proposition 4.4 is proved in Appendix B.

Theorem 4.5 (Local convergence of the semismooth Newton step)**.** *Fix $\sigma > 0$, $\widetilde{y} \in \mathbb{R}$, and $\widetilde{z} \in \mathbb{R}^m$. Assume that $p = 1$, $p = \infty$, or $p \in \{2, 3, \ldots\}$, and let $\varphi_\star \in \mathcal{H}$ satisfy $\nabla f_\sigma(\varphi_\star) = 0$. If every element of $\partial_C \nabla f_\sigma(\varphi_\star)$ is nonsingular on $\mathcal{H}$, then the iteration defined by Equation (22), together with sufficiently accurate conjugate gradient solves and an Armijo line search, is well defined in a neighborhood of $\varphi_\star$ and converges superlinearly to $\varphi_\star$. More precisely, if the linear systems are solved with residual of order $O(\|\nabla f_\sigma(\varphi^t)\|_2^{1+\tau})$ for some $\tau \in (0, 1]$, then*

$$\|\varphi^{t+1} - \varphi_\star\|_2 = O\big(\|\varphi^t - \varphi_\star\|_2^{1+\tau}\big)\,.$$

Theorem 4.5 is proved in Appendix B. To pass from the inner solver to the full ALM, we impose the summable inexactness condition: for some nonnegative sequence $(\varepsilon_k)_{k\geq 0}$ with $\sum_{k=0}^{\infty} \varepsilon_k < \infty$,

$$\mathrm{dist}\,\big(0, \partial_{(\varphi,u)} \mathcal{L}_{\sigma_k}(\varphi_{k+1}, u_{k+1}; y_k, z_k)\big) \leq \frac{\varepsilon_k}{\max\{1, \sqrt{\sigma_k}\}}. \tag{23}$$

Theorem 4.6 (Global convergence of the augmented Lagrangian method)**.** *Assume that Equation (17) is solved by the iteration Equation (19), and that each inner solve satisfies Equation (23). Then the primal feasibility residuals satisfy*

$$\|B^\top \varphi_k - u_k\|_2 \to 0 \qquad \textit{and} \qquad |b^\top \varphi_k - 1| \to 0.$$

Every accumulation point of (φ_k, u_k) is a primal solution of Equation (17). If the centered primal solution is unique, then the whole sequence (φ_k, u_k) converges to it.

Theorem 4.6 is proved in Appendix B.

4.2 Refinements and class assignments

The binary solver extends to the multiclass one-vs-all formulation by solving the k column problems independently. If φ_j denotes the centered potential associated with source term $b_j := \mu_j - \nu_j$, we collect the results in the matrix

$$\Phi := \begin{bmatrix} \varphi_1 & \cdots & \varphi_k \end{bmatrix}.$$

The remaining algorithmic step is to convert Φ into a discrete partition.

Multiclass assignment and refinement. Suppose an estimate of the class cardinalities is available in the form of a vector $\beta \in \mathbb{Z}_{\geq 0}^k$ with $\mathbf{1}_k^\top \beta = n$. Here $\mathbf{1}_n \in \mathbb{R}^n$ and $\mathbf{1}_k \in \mathbb{R}^k$ denote all ones vectors. We encode a hard k-way assignment by a matrix $P \in \{0,1\}^{n\times k}$ satisfying

$$P\mathbf{1}_k = \mathbf{1}_n, \qquad P^\top \mathbf{1}_n = \beta.$$

Here $P_{ir} = 1$ means that vertex x_i is assigned to class r. We therefore consider the following decoding problem:

$$\max_{P\in\{0,1\}^{n\times k}} \langle \Phi, P\rangle \quad \text{subject to} \quad P\mathbf{1}_k = \mathbf{1}_n, \qquad P^\top \mathbf{1}_n = \beta, \tag{24}$$

Algorithm 4 p-conductance-MBO refinement

1: Choose $\alpha > 0$, $\eta > 0$, and an initial $P^0 \in \mathcal{P}(\beta)$.
2: **for** $t = 0, 1, 2, \ldots$ until convergence **do**
3: Set $Q^{t+\frac{1}{2}} \leftarrow P^t - \eta(2\alpha L P^t - \Phi)$.
4: Set $P^{t+1} \leftarrow \arg\max_{P \in \mathcal{P}(\beta)} \langle P, Q^{t+\frac{1}{2}} \rangle$.
5: **end for**

where $\langle \Phi, P \rangle = \sum_{i=1}^{n} \sum_{r=1}^{k} \Phi_{ir} P_{ir}$ is the Frobenius inner product. If only approximate class sizes are available, we relax the column-sum constraint and solve

$$\max_{P \in \mathbb{R}^{n \times k}} \langle \Phi, P \rangle \quad \text{subject to} \quad P \geq 0, \quad P\mathbf{1}_k = \mathbf{1}_n, \quad \beta - \epsilon \mathbf{1}_k \leq P^\top \mathbf{1}_n \leq \beta + \epsilon \mathbf{1}_k \tag{25}$$

for some tolerance $\epsilon \geq 0$.

Remark 4.7. When $\epsilon = 0$, the feasible set in Equation (25) is a transportation polytope. Since the constraint matrix is totally unimodular, every extreme point is integral whenever $\beta \in \mathbb{Z}^k_{\geq 0}$, and an optimal solution may therefore be taken in $\{0, 1\}^{n \times k}$; see Loera & Kim (2013). At the opposite extreme, when ϵ is large enough that the bounds on $P^\top \mathbf{1}_n$ impose no restriction beyond $P \geq 0$ and $P\mathbf{1}_k = \mathbf{1}_n$, the maximization decouples rowwise and reduces to the argmax rule

$$\widehat{y}(x_i) = \arg\max_{r \in [k]} \Phi_{ir}. \tag{26}$$

The linear decoding step above is useful when the potentials already localize classes well. To further refine this step, given $\alpha > 0$, we consider the following program:

$$\min_{P \in \{0,1\}^{n \times k}} \alpha \operatorname{tr}(P^\top L P) - \langle \Phi, P \rangle \quad \text{subject to} \quad P\mathbf{1}_k = \mathbf{1}_n, \qquad P^\top \mathbf{1}_n = \beta. \tag{27}$$

The first term penalizes rough class boundaries, while the second keeps the assignment aligned with the conductance potentials. Let

$$\mathcal{P}(\beta) := \left\{ P \in \mathbb{R}^{n \times k}_{+} : P\mathbf{1}_k = \mathbf{1}_n, \quad P^\top \mathbf{1}_n = \beta \right\}$$

be the corresponding transportation polytope. Since L is symmetric, the smooth part of Equation (27) has gradient $2\alpha L P - \Phi$, so the resulting MBO-type iteration takes the form shown in Algorithm 4.

The thresholding step is a linear program over $\mathcal{P}(\beta)$, and therefore returns an extreme point when the class-size constraint is exact. If approximate class sizes are preferred, one may replace $\mathcal{P}(\beta)$ by the relaxed feasible set from Equation (25).

5 Experimental Results

We evaluate p-conductance learning across six different regimes. We first study citation networks, where the graph is given and the effect of the conductance objective can be isolated from feature learning. We then turn to image classification, where the graph must be built from learned latent representations. Next we test robustness to random label corruption through the diffusion step Equation (15), examine weak supervision in the partially labeled learning setting, and finally report optimization behavior together with qualitative structure of the learned cuts. Unless stated otherwise, all accuracies are averaged over 100 random trials, with standard deviations reported in parentheses.

5.1 Experimental setup

Across all multiclass experiments we solve the one-vs-all conductance problems associated with Equation (6). When no class size information is used, class predictions are obtained from the argmax rule Equation (26). When class cardinalities are available, we use the linear assignment step Equation (25) and, when indicated,

Table 1: Citation datasets. Average accuracy (standard deviation) over 100 trials. Best is bolded.

Cora	# Labels per class	**1**	**3**	**5**	**10**	**100**
Laplace/LP Zhu et al. (2003)		21.8 (14.3)	37.6 (12.3)	51.3 (11.9)	66.9 (6.8)	81.8 (1.1)
Sparse LP Jung et al. (2016)		16.0 (1.8)	19.4 (1.8)	23.1 (2.3)	28.7 (2.2)	47.0 (2.2)
p-laplace Flores et al. (2022)		41.9 (8.9)	57.6 (7.1)	61.9 (6.2)	67.9 (3.6)	79.2 (1.3)
p-eikonal Calder & Ettehad (2022)		40.4 (8.6)	51.8 (7.2)	56.9 (6.8)	64.5 (4.2)	80.2 (1.1)
Poisson Calder et al. (2020)		57.4 (9.2)	67.0 (4.9)	69.3 (3.6)	71.6 (3.0)	76.0 (1.0)
p-conductance ($p = 1$, $\epsilon = n$)		22.9 (6.8)	22.7 (6.8)	21.4 (6.6)	21.3 (6.6)	35.6 (13.2)
p-conductance ($p = 2$, $\epsilon = n$)		58.9 (7.2)	67.9 (3.7)	70.2 (2.3)	72.2 (1.9)	75.2 (1.3)
p-conductance ($p = \infty$, $\epsilon = n$)		44.3 (6.8)	53.7 (5.0)	58.9 (3.6)	63.7 (3.1)	73.7 (1.4)
PoissonMBO Calder et al. (2020)		58.5 (9.4)	68.5 (4.1)	70.7 (3.0)	73.3 (2.3)	80.1 (0.9)
p-conductance ($p = 2$, $\epsilon = 0$)		**63.1 (8.0)**	**72.9 (3.5)**	**75.5 (1.8)**	**77.9 (1.1)**	**82.9 (0.9)**
Pubmed	# Labels per class	**1**	**3**	**5**	**10**	**100**
Laplace/LP Zhu et al. (2003)		34.6 (8.8)	35.7 (8.2)	36.9 (8.1)	39.6 (9.1)	74.9 (3.6)
Sparse LP Jung et al. (2016)		32.4 (4.7)	33.0 (4.8)	33.6 (4.8)	33.9 (4.8)	43.2 (4.1)
p-laplace Flores et al. (2022)		44.8 (11.2)	58.3 (9.1)	61.6 (7.7)	66.2 (4.7)	74.3 (1.1)
p-eikonal Calder & Ettehad (2022)		44.3 (11.8)	55.6 (10.0)	58.4 (9.1)	65.1 (5.8)	74.9 (1.5)
Poisson Calder et al. (2020)		55.1 (11.3)	66.6 (7.4)	68.8 (5.6)	71.3 (2.2)	75.7 (0.8)
p-conductance ($p = 1$, $\epsilon = n$)		39.6 (0.3)	39.6 (0.3)	39.6 (0.3)	40.3 (0.3)	41.2 (0.3)
p-conductance ($p = 2$, $\epsilon = n$)		58.0 (12.1)	67.5 (7.5)	70.8 (4.9)	**72.4 (2.5)**	77.6 (0.6)
p-conductance ($p = \infty$, $\epsilon = n$)		48.0 (8.1)	56.5 (5.2)	57.0 (8.2)	62.6 (3.0)	72.9 (1.3)
PoissonMBO Calder et al. (2020)		54.9 (11.4)	65.3 (7.8)	68.2 (5.3)	69.9 (3.0)	74.8 (1.0)
p-conductance ($p = 2$, $\epsilon = 0$)		**58.7 (11.5)**	**67.5 (7.5)**	**70.8 (4.8)**	**72.4 (2.6)**	**77.8 (0.6)**
Citeseer	# Labels per class	**1**	**3**	**5**	**10**	**100**
Laplace/LP Zhu et al. (2003)		27.9 (10.4)	47.6 (8.1)	56.0 (5.9)	63.7 (3.5)	71.6 (1.2)
Sparse LP Jung et al. (2016)		19.3 (2.4)	21.9 (2.3)	24.5 (2.3)	28.9 (2.0)	45.6 (2.2)
p-laplace Flores et al. (2022)		39.7 (10.6)	51.4 (6.7)	56.1 (5.8)	60.9 (3.4)	69.5 (1.1)
p-eikonal Calder & Ettehad (2022)		39.5 (10.6)	49.7 (7.4)	54.9 (6.3)	60.0 (3.5)	71.0 (1.0)
Poisson Calder et al. (2020)		46.7 (12.1)	57.8 (7.4)	62.6 (4.4)	66.7 (2.5)	**74.1 (0.8)**
p-conductance ($p = 1$, $\epsilon = n$)		24.8 (2.9)	24.7 (2.6)	25.9 (2.9)	28.7 (3.4)	40.3 (9.3)
p-conductance ($p = 2$, $\epsilon = n$)		49.9 (11.4)	60.9 (4.5)	64.0 (3.0)	**67.2 (1.8)**	72.5 (0.8)
p-conductance ($p = \infty$, $\epsilon = n$)		37.9 (10.2)	47.6 (7.0)	53.4 (4.8)	58.5 (3.4)	69.6 (1.0)
PoissonMBO Calder et al. (2020)		46.2 (10.6)	55.8 (5.4)	60.0 (3.0)	63.5 (2.5)	73.2 (1.0)
p-conductance ($p = 2$, $\epsilon = 0$)		**50.7 (11.2)**	**61.0 (4.2)**	**64.0 (2.2)**	66.7 (1.7)	**74.1 (0.8)**

the refinement scheme of Algorithm 4. We compare against Laplace learning / label propagation Zhu et al. (2003), sparse label propagation Jung et al. (2016), p-Laplacian learning Flores et al. (2022), the p-eikonal method Calder & Ettehad (2022), Poisson learning Calder et al. (2020), and the MBO-based baselines VolumeMBO Jacobs et al. (2018) and PoissonMBO Calder et al. (2020) whenever those methods are applicable. All experiments were implemented with the `graphlearning` package Calder (2022).

5.2 Citation networks

We begin with the Planetoid citation datasets Yang et al. (2016), where the graph is part of the dataset rather than being constructed from ambient features. Vertices represent publications, edges represent citations, and class labels encode document subject. We evaluate on the largest connected component of each graph. VolumeMBO was omitted on these datasets because the archived runs produced runtime failures. Results for Cora, Pubmed, and Citeseer appear in Table 1. At one label per class on Cora, the unconstrained $p = 2$ decoder improves on PoissonMBO by 0.4 points (58.9 versus 58.5), while constrained assignment with $\epsilon = 0$ improves by 4.6 points (63.1 versus 58.5). Similar improvements from assignment constraints appear on the other citation datasets and label rates.

Table 2: Image datasets. Average accuracy (standard deviation) over 100 trials. Best is bolded

MNIST # Labels per class	**1**	**3**	**5**	**10**	**4000**
Label propagation Zhu et al. (2003)	16.1 (6.2)	42.0 (12.4)	69.5 (12.2)	94.8 (2.1)	98.3 (0.0)
Sparse LP Jung et al. (2016)	14.0 (5.5)	14.5 (4.0)	16.2 (4.2)	18.4 (4.1)	94.1 (0.1)
p-laplace Flores et al. (2022)	83.2 (2.4)	87.4 (1.5)	90.7 (0.9)	94.8 (0.1)	**98.1 (0.1)**
p-eikonal Calder & Ettehad (2022)	81.4 (4.5)	91.2 (1.3)	92.5 (1.1)	94.3 (0.6)	**98.1 (0.1)**
Poisson Calder et al. (2020)	95.8 (0.8)	96.3 (0.3)	96.7 (0.1)	97.0 (0.1)	97.2 (0.1)
p-conductance ($p = 1$, $\epsilon = n$)	24.3 (1.6)	31.8 (1.4)	43.1 (1.1)	53.9 (0.8)	96.3 (0.1)
p-conductance ($p = 2$, $\epsilon = n$)	95.4 (1.3)	96.4 (0.5)	96.7 (0.3)	97.2 (0.1)	97.3 (0.1)
p-conductance ($p = \infty$, $\epsilon = n$)	86.3 (0.8)	89.2 (0.6)	92.7 (0.4)	95.3 (0.1)	97.2 (0.1)
VolumeMBO Jacobs et al. (2018)	89.9 (7.3)	96.2 (1.2)	96.7 (0.6)	96.7 (0.2)	96.9 (0.1)
PoissonMBO Calder et al. (2020)	**97.5 (0.1)**	**97.5 (0.1)**	**97.5 (0.1)**	**97.6 (0.1)**	**98.1 (0.1)**
p-conductance ($p = 2$, $\epsilon = 0$)	96.6 (0.4)	96.9 (0.2)	97.1 (0.1)	97.4 (0.1)	97.5 (0.1)
p-conductance-MBO	**97.5 (0.1)**	**97.5 (0.1)**	**97.5 (0.1)**	**97.6 (0.1)**	**98.1 (0.1)**

FashionMNIST # Labels per class	**1**	**3**	**5**	**10**	**4000**
Label propagation Zhu et al. (2003)	18.4 (7.3)	44.0 (8.6)	57.9 (6.7)	70.6 (3.1)	85.8 (0.1)
Sparse LP Jung et al. (2016)	14.1 (3.8)	13.7 (3.3)	16.1 (2.5)	15.2 (2.5)	79.3 (0.1)
p-laplace Flores et al. (2022)	60.9 (3.0)	65.2 (2.5)	69.4 (1.5)	78.0 (0.4)	85.5 (0.1)
p-eikonal Calder & Ettehad (2022)	55.1 (4.4)	61.3 (3.0)	65.5 (3.4)	70.1 (1.2)	85.8 (0.1)
Poisson Calder et al. (2020)	68.6 (2.6)	71.7 (2.3)	74.3 (1.5)	79.6 (0.6)	81.1 (0.1)
p-conductance ($p = 1$, $\epsilon = n$)	18.2 (6.3)	21.3 (5.1)	29.3 (3.9)	46.4 (2.4)	80.4 (0.3)
p-conductance ($p = 2$, $\epsilon = n$)	66.5 (2.8)	69.9 (2.5)	73.3 (1.7)	80.2 (0.6)	81.4 (0.2)
p-conductance ($p = \infty$, $\epsilon = n$)	54.2 (2.6)	58.3 (1.9)	64.1 (2.1)	72.8 (0.9)	80.8 (0.1)
VolumeMBO Jacobs et al. (2018)	54.7 (5.2)	66.1 (3.3)	70.1 (2.8)	74.4 (1.5)	75.1 (0.2)
PoissonMBO Calder et al. (2020)	69.7 (2.8)	73.0 (2.3)	75.7 (1.6)	80.9 (0.5)	86.3 (0.1)
p-conductance ($p = 2$, $\epsilon = 0$)	70.2 (2.5)	**73.4 (1.9)**	**76.1 (1.4)**	81.0 (0.4)	**87.3 (0.1)**
p-conductance-MBO	**71.6 (0.7)**	72.5 (1.5)	**76.1 (0.8)**	**82.4 (0.5)**	86.5 (0.1)

CIFAR-10 # Labels per class	**1**	**3**	**5**	**10**	**4000**
Label propagation Zhu et al. (2003)	10.4 (1.3)	11.6 (2.7)	14.1 (5.0)	21.8 (7.4)	80.9 (0.0)
Sparse LP Jung et al. (2016)	11.8 (2.4)	11.1 (3.3)	14.4 (3.5)	15.6 (3.1)	75.5 (0.3)
p-laplace Flores et al. (2022)	33.4 (4.7)	42.1 (3.1)	47.2 (2.7)	52.2 (1.6)	80.8 (0.2)
p-eikonal Calder & Ettehad (2022)	32.9 (5.1)	42.8 (5.4)	49.7 (3.5)	52.8 (3.0)	80.5 (0.1)
Poisson Calder et al. (2020)	39.3 (5.4)	48.9 (3.5)	53.3 (2.8)	57.8 (1.8)	80.3 (0.2)
p-conductance ($p = 1$, $\epsilon = n$)	13.4 (6.1)	17.1 (3.7)	21.8 (2.9)	39.6 (2.1)	78.4 (0.3)
p-conductance ($p = 2$, $\epsilon = n$)	40.0 (5.1)	50.4 (3.4)	54.9 (2.6)	59.4 (1.6)	80.6 (0.2)
p-conductance ($p = \infty$, $\epsilon = n$)	30.1 (4.7)	39.7 (3.1)	43.6 (2.2)	45.9 (1.4)	79.1 (0.2)
VolumeMBO Jacobs et al. (2018)	38.0 (7.2)	50.1 (5.7)	55.3 (3.8)	59.2 (3.2)	75.8 (0.9)
PoissonMBO Calder et al. (2020)	40.8 (6.8)	53.1 (4.1)	58.1 (3.2)	62.0 (1.9)	80.5 (0.2)
p-conductance ($p = 2$, $\epsilon = 0$)	40.3 (5.6)	52.1 (3.6)	57.1 (2.3)	61.5 (1.5)	80.7 (0.2)
p-conductance-MBO	**43.2 (6.5)**	**54.0 (3.2)**	**59.2 (2.4)**	**62.6 (1.0)**	**80.8 (0.2)**

5.3 Image datasets

We next study MNIST Deng (2012), Fashion-MNIST Xiao et al. (2017), and CIFAR-10 Krizhevsky & Hinton (2009). These experiments test the full pipeline in the setting where the graph must be constructed from learned representations before solving the conductance problem. As in Calder et al. (2020), we use pretrained autoencoders as feature extractors. For MNIST and Fashion-MNIST we used variational autoencoders with three fully connected layers of sizes $(784, 400, 20)$ and $(784, 400, 30)$, respectively, followed by the symmetric decoder. Each autoencoder was trained for 100 epochs, with architecture and training choices matching Kingma & Welling (2014).

For each dataset we construct a graph on the latent feature vectors. We use all available examples, giving $n = 70{,}000$ vertices for MNIST and Fashion-MNIST and $n = 60{,}000$ vertices for CIFAR-10. The graph is a k-nearest neighbor graph with Gaussian weights

$$w_{ij} = \exp\left(-4\|z_i - z_j\|_2^2 / d_k(z_i)^2\right),$$

where z_i denotes the latent feature vector of image i and $d_k(z_i)$ is the distance from z_i to its kth nearest neighbor in latent space. We use $k = 10$ throughout and symmetrize the weight matrix. The resulting accuracies are listed in Table 2. On CIFAR-10, cardinality-aware p-conductance-MBO improves on PoissonMBO by 2.4 percentage points. On Fashion-MNIST at one label per class, p-conductance-MBO improves on PoissonMBO by 1.9 percentage points.

5.4 Corrupted labels

To isolate robustness to label noise, we randomly flip the labels of a fixed proportion of the training set to incorrect classes and then apply the diffusion step from Equation (15) before solving the multiclass conductance problem. The experiment is performed on CIFAR-10 with ten labels per class. The resulting accuracy is shown in Figure 3 as a function of the diffusion time and corruption level. At 40% label corruption, our method improves on Poisson learning by 3.8 percentage points. The figure also shows that the gain depends strongly on the diffusion time: diffusing the label measures substantially improves the conductance-based method, while Poisson learning benefits much less from the same preprocessing. Interpolating methods such as Laplace learning are not included here, since after diffusion every vertex receives nonzero label mass and the hard interpolation constraint is no longer meaningful.

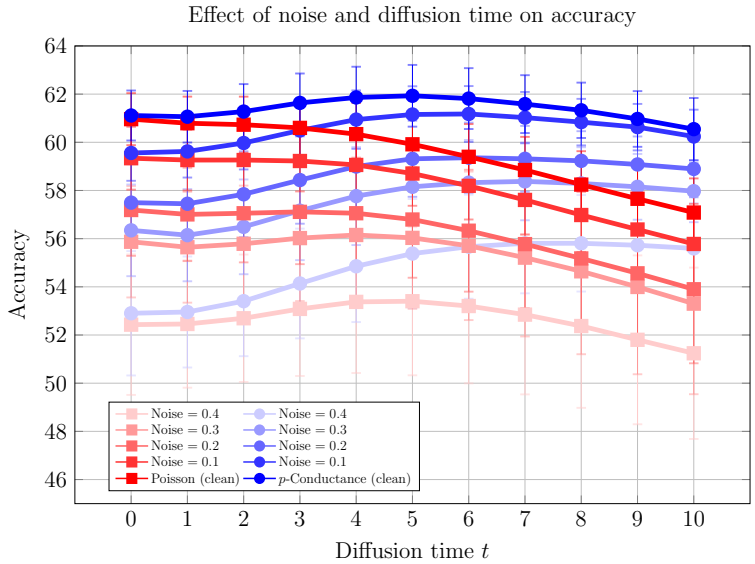


Figure 3: Accuracy on CIFAR-10 for various diffusion times and robustness. Shades denote noise level. Label rate is 10 labels per class.

5.5 Partially labeled learning

We next consider partially labeled learning (PLL), a weakly supervised setting in which each labeled training example is accompanied by a candidate set of labels rather than a single class. On CIFAR-100 Krizhevsky & Hinton (2009), we use the superclass structure to generate candidate sets: for each labeled image, the candidate classes are sampled from the five fine-grained labels contained in the superclass of the true class. The task is then to recover the fine-grained label for every test image from these candidate label training annotations. The results in Table 3 show a monotone degradation in accuracy as the candidate set grows, but the conductance-based refinements remain consistently stronger than the Poisson baselines. In particular,

p-conductance-MBO improves on PoissonMBO by between 2.4 and 4.1 percentage points across candidate-set sizes 1 through 4, while the $p = \infty$ variant is strongest when all five superclass labels are allowed.

Table 3: CIFAR-100 in the PLL setting. Average accuracy over 100 trials with standard deviation in brackets. Label rate is 10 labels per class.

CIFAR-100 # candidate labels	**1**	**2**	**3**	**4**	**5**
Label propagation (Zhu et al., '03)	39.3 (0.5)	33.4 (0.8)	26.7 (1.0)	18.6 (1.4)	11.4 (0.3)
Poisson (Calder et al., '20)	40.5 (0.5)	34.7 (0.6)	28.0 (1.1)	19.3 (1.3)	11.3 (0.2)
p-conductance ($p = 1$, $\epsilon = n$)	21.4 (1.1)	20.1 (1.3)	19.4 (1.1)	14.6 (1.9)	11.4 (0.3)
p-conductance ($p = 2$, $\epsilon = n$)	41.7 (0.4)	35.0 (0.7)	29.4 (1.1)	22.2 (1.7)	11.8 (0.1)
p-conductance ($p = \infty$, $\epsilon = n$)	30.4 (0.8)	27.1 (0.7)	26.4 (0.9)	23.1 (0.9)	**14.3 (0.1)**
PoissonMBO	42.8 (0.9)	37.8 (0.7)	31.4 (0.7)	22.3 (1.3)	11.4 (0.1)
p-conductance ($p = 2$, $\epsilon = 0$)	42.1 (0.5)	38.2 (0.5)	32.5 (0.9)	24.1 (1.8)	12.2 (0.3)
p-conductance-MBO	**45.2 (0.4)**	**40.5 (0.4)**	**34.4 (0.6)**	**26.4 (1.1)**	14.1 (0.1)

5.6 Convergence of SSNAL

Using the first-order splitting derivation in Section 4, we compare SSNAL against an ADMM-type method that uses the same two core primitives (edgewise proximal updates and constrained Laplacian solves), but replaces the semismooth Newton step with first-order splitting. This gives a clean optimization comparison in which both methods operate on the same conductance formulation but differ in update geometry and local model quality. We therefore evaluate convergence behavior directly on a 20×20 lattice with $p = 5$, and monitor the relative KKT residual

$$\max\{\eta_1, \eta_2\} \leq \varepsilon, \tag{28}$$

with

$$\begin{aligned} \eta_1 &= \frac{\|B^\top \varphi - u\|_2 + |b^\top \varphi - 1|}{1 + \|u\|_2 + \|b\|_2}, \\ \eta_2 &= \frac{\|Bz + yb\|_2 + \|u - \mathrm{prox}_{s_p/\sigma}(u + \sigma^{-1} z)\|_2}{1 + \|b\|_2 + \|u\|_2}, \end{aligned} \tag{29}$$

where $b = \mu - \nu$. In the experiments we set $\varepsilon = 10^{-4}$. The decay of the two residual components is shown in Figure 4; SSNAL reaches the stopping criterion in substantially fewer iterations than the first-order baseline.

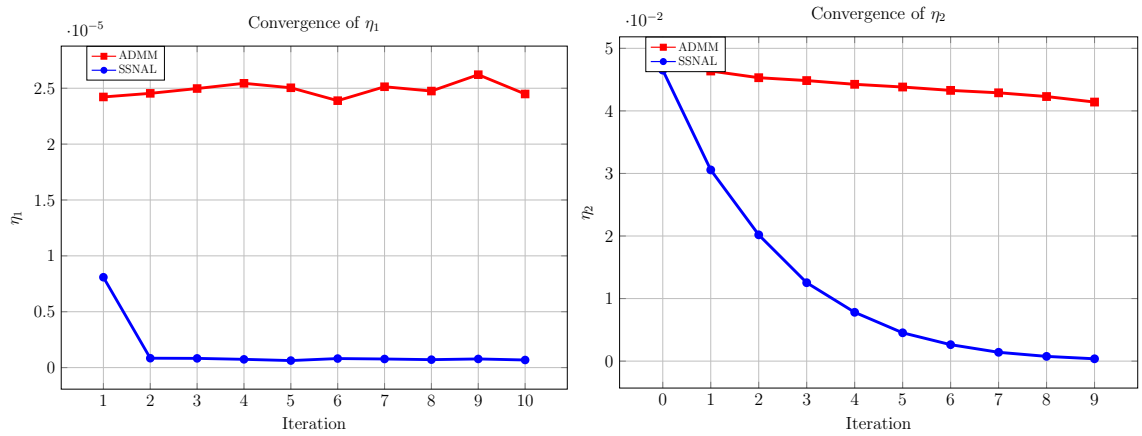


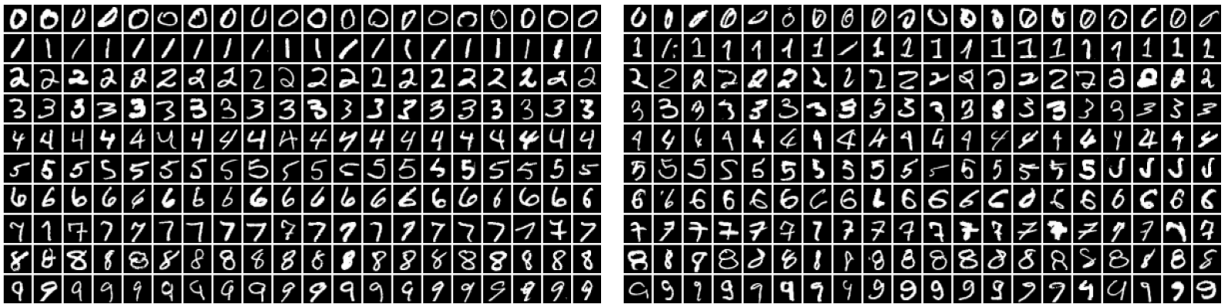
Figure 4: Convergence of KKT residuals on lattice data for $p = 5$

5.7 Qualitative structure of the learned cuts

Finally, we examine the geometry induced by the conductance potentials. For a hard bipartition $V = V_0 \cup V_1$, the node boundary of V_1 relative to V_0 is

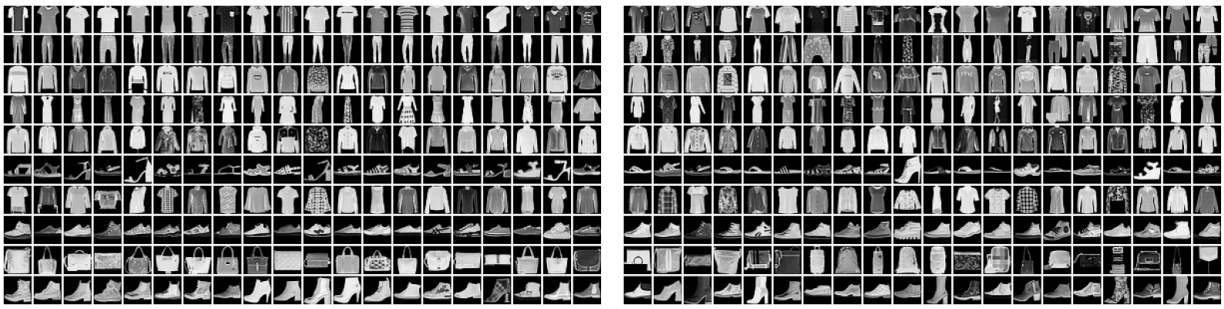
$$\partial_{V_0} V_1 := \{x \in V_1 : \text{there exists } y \in V_0 \text{ with } \{x, y\} \in E\}. \tag{30}$$

Figures 5 and 6 compare randomly sampled images with images lying on the predicted cut boundary. Boundary samples are visibly more ambiguous, which is consistent with the interpretation of the conductance potential as identifying low-margin regions of the graph. We also visualize in Figure 7 the potential $\varphi_\star$ solving Equation (3) for $p = 2$ on Cora when the positive measure is supported on randomly selected neural-network papers and the negative measure is supported on randomly selected papers from the remaining classes.



(a) Random images (b) Boundary images

Figure 5: MNIST experiments



(a) Random images (b) Boundary images

Figure 6: FashionMNIST experiments. The labels are 0: T-shirt/top 1: Trouser 2: Pullover 3: Dress 4: Coat 5: Sandal 6: Shirt 7: Sneaker 8: Bag 9: Ankle boot

6 Future directions

In this paper we focus on the theoretical properties of the p-conductance program and its connections to existing graph-based optimization problems, as well as on developing an efficient solver for the program. There remain many possible directions for future investigation, of which we highlight two:

1. As detailed in the preceding paragraphs, 2-conductance learning uses the same driving operator $L^\dagger$ as Poisson learning, and the two methods differ only in the choice of label preprocessing. However, 2-conductance learning still appears to outperform Poisson learning in many experimental settings, even when controlling for differing postprocessing conventions. This raises the question: what is the precise mechanism behind this empirical improvement? Is there a theoretical explanation for the observed performance gap?

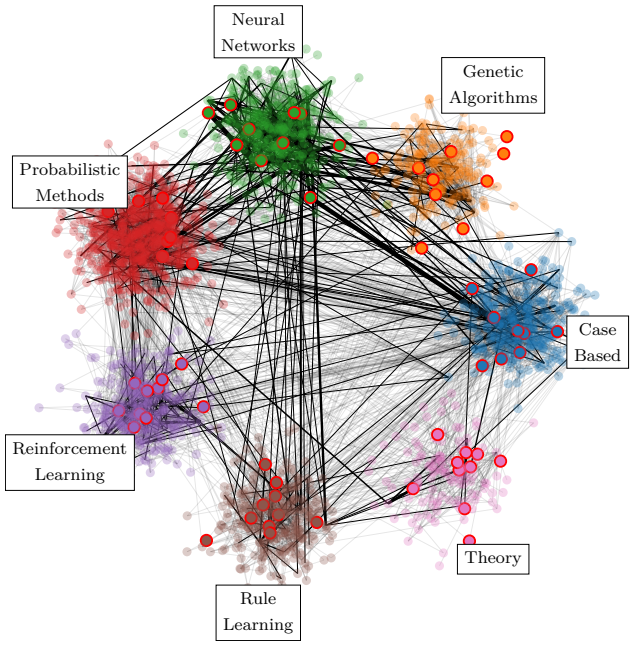


Figure 7: An illustration of the one-vs-all minimum measure cut on a graph obtained from the Cora dataset Yang et al. (2016). Here, the measure μ is obtained from five randomly selected nodes among all neural network papers and ν is obtained from five randomly selected nodes from each of the other classes. The width of each edge $\{x, y\}$ is proportional to $|\varphi_\star(x) - \varphi_\star(y)|$ when this quantity is large and edges are suppressed otherwise, where $\varphi_\star$ solves Equation (3) for $p = 2$. The training nodes used for constructing μ, ν are distinguished by size and red outlines.

2. p-conductance learning, more generally, appears to exhibit strong robustness properties in the presence of low label rates, noisy labels, and weak supervision. Can we provide a more comprehensive theoretical explanation for this robustness beyond the motivational results in Section 3.2?

Broader Impact Statement

This paper presents work whose goal is to advance the field of machine learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Morteza Alamgir and Ulrike Luxburg. Phase transition in the family of p-resistances. *Advances in neural information processing systems*, 24, 2011.

A. El Kacimi Alaoui, Xiang Cheng, Aaditya Ramdas, Martin J. Wainwright, and Michael I. Jordan. Asymptotic behavior of ℓp-based laplacian regularization in semi-supervised learning. *ArXiv*, abs/1603.00564, 2016. URL `https://api.semanticscholar.org/CorpusID:1089782`.

Martin Beckmann. A continuous model of transportation. *Econometrica: Journal of the Econometric Society*, 1952.

Adi Ben-Israel and A Charnes. Contributions to the theory of generalized inverses. *Journal of the Society for Industrial and Applied Mathematics*, 11(3), 1963.

Yuri Boykov and Olga Veksler. Graph cuts in vision and graphics: Theories and applications. In *Handbook of mathematical models in computer vision*, pp. 79–96. Springer, 2006.

Jeff Calder. Consistency of lipschitz learning with infinite unlabeled data and finite labeled data. *SIAM Journal on Mathematics of Data Science*, 1(4):780–812, 2019. doi: 10.1137/18M1199241. URL https://doi.org/10.1137/18M1199241.

Jeff Calder. Graphlearning python package, January 2022. URL https://doi.org/10.5281/zenodo.5850940.

Jeff Calder and Mahmood Ettehad. Hamilton-jacobi equations on graphs with applications to semi-supervised learning and data depth. *J. Mach. Learn. Res.*, 23(1), January 2022. ISSN 1532-4435.

Jeff Calder, Brendan Cook, Matthew Thorpe, and Dejan Slepčev. Poisson learning: Graph based semi-supervised learning at very low label rates. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

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

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Montacer Essid and Justin Solomon. Quadratically regularized optimal transport on graphs. *SIAM Journal on Scientific Computing*, 40(4), 2018.

M. Flores. Algorithms for semisupervised learning on graphs. 2018.

Mauricio Flores, Jeff Calder, and Gilad Lerman. Analysis and algorithms for ℓp-based semi-supervised learning on graphs. *Applied and Computational Harmonic Analysis*, 60:77–122, 2022. ISSN 1063-5203. doi: https://doi.org/10.1016/j.acha.2022.01.004. URL https://www.sciencedirect.com/science/article/pii/S1063520322000045.

Robert M Freund. Dual gauge programs, with applications to quadratic programming and the minimum-norm problem. *Mathematical Programming*, 38, 1987.

Michael P Friedlander, Ives Macedo, and Ting Kei Pong. Gauge optimization and duality. *SIAM Journal on Optimization*, 24(4), 2014.

Chester Holtz, Pengwen Chen, Zhengchao Wan, Chung-Kuan Cheng, and Gal Mishne. Continuous partitioning for graph-based semi-supervised learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=hCOuip5Ona.

Matt Jacobs, Ekaterina Merkurjev, and Selim Esedoḡlu. Auction dynamics: A volume constrained MBO scheme. *Journal of Computational Physics*, 354:288–310, 2018. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2017.10.036. URL https://www.sciencedirect.com/science/article/pii/S0021999117308033.

Alexander Jung, Alfred O. Hero III, Alexandru Mara, and Saeed Jahromi. Semi-supervised learning via sparse label propagation, 2016.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. (0), 2009.

Rasmus Kyng, Anup Rao, Sushant Sachdeva, and Daniel A. Spielman. Algorithms for lipschitz learning on graphs. In Peter Grünwald, Elad Hazan, and Satyen Kale (eds.), *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pp. 1190–1223, Paris, France, 03–06 Jul 2015. PMLR. URL https://proceedings.mlr.press/v40/Kyng15.html.

Jesús A. De Loera and Edward D. Kim. Combinatorics and geometry of transportation polytopes: An update, 2013. URL https://arxiv.org/abs/1307.0124.

Boaz Nadler, Nathan Srebro, and Xueyuan Zhou. Semi-supervised learning with the graph laplacian: the limit of infinite unlabelled data. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems*, NIPS'09, pp. 1330–1338, Red Hook, NY, USA, 2009. Curran Associates Inc. ISBN 9781615679119.

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6), 2019.

Sawyer Robertson, Zhengchao Wan, and Alexander Cloninger. All you need is resistance: On the equivalence of effective resistance and certain optimal transport problems on graphs. *arXiv preprint arXiv:2404.15261*, 2024.

R. T. Rockafellar. Augmented lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of Operations Research*, 1(2):97–116, 1976a. ISSN 0364765X, 15265471. URL http://www.jstor.org/stable/3689277.

R. Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976b. doi: 10.1137/0314056. URL https://doi.org/10.1137/0314056.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8), 2000.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *ArXiv*, abs/1708.07747, 2017.

Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pp. 40–48. JMLR.org, 2016.

Xin-Yuan Zhao, Defeng Sun, and Kim-Chuan Toh. A newton-cg augmented lagrangian method for semidefinite programming. *SIAM Journal on Optimization*, 20(4):1737–1765, 2010. doi: 10.1137/080718206. URL https://doi.org/10.1137/080718206.

Xueyuan Zhou and Mikhail Belkin. Semi-supervised learning by higher order regularization. In Geoffrey Gordon, David Dunson, and Miroslav Dudík (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 892–900, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.

Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, pp. 912–919. AAAI Press, 2003. ISBN 1577351894.

Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005.

A Proofs from Section 3

Proof of Theorem 3.4. Fix $\varphi \in \mathbb{R}^n$. In Equation (8), the choice of K minimizing $\sum_{(x,y)\in E''} w_{xy} K(x,y)$ for fixed φ is

$$K(x,y) = (\varphi(x) - \varphi(y))_+.$$

Since E'' contains both orientations of each undirected edge, and exactly one of $(\varphi(x) - \varphi(y))_+$ and $(\varphi(y) - \varphi(x))_+$ is nonzero,

$$\sum_{(x,y)\in E''} w_{xy}(\varphi(x) - \varphi(y))_+ = \sum_{\{x,y\}\in E} w_{xy}|\varphi(x) - \varphi(y)|.$$

Hence Equation (8) reduces exactly to Equation (3) with $p = 1$.

To derive the dual, write the Lagrangian with multipliers $J \geq 0$ and $f \geq 0$:

$$\begin{aligned}\mathcal{L}(K, \varphi; J, f) = &\sum_{(x,y)\in E''} w_{xy}K(x, y) + \sum_{(x,y)\in E''} J(x, y)(\varphi(x) - \varphi(y) - K(x, y)) \\ &+ f\big(1 - (\mu - \nu)^\top \varphi\big).\end{aligned}$$

Minimizing over $K \geq 0$ is finite if and only if $J(x, y) \leq w_{xy}$ for all $(x, y) \in E''$, and in that case the minimum contribution from K is 0. Minimizing over $\varphi \in \mathbb{R}^n$ is finite if and only if the coefficient of φ vanishes, i.e.,

$$\widetilde{B}J = f(\mu - \nu).$$

Under these conditions the dual objective is f, which gives Equation (9).

The primal is strictly feasible, and connectivity ensures that the dual is feasible: choose any nonnegative flow on E'' with divergence $\mu - \nu$ and scale it so that all capacities $J(x, y) \leq w_{xy}$ hold. The equality Equation (10) then follows from strong duality for linear programs. □

Proof of Theorem 3.5. For each edge $\{x, y\} \in E$, the event that the threshold cut separates x and y has probability

$$\mathbb{P}\big(\min\{\varphi(x), \varphi(y)\} < T \leq \max\{\varphi(x), \varphi(y)\}\big) = \frac{|\varphi(x) - \varphi(y)|}{\|\varphi\|_\infty}.$$

Therefore

$$\mathbb{E}_T[w(A_T, B_T)] = \frac{1}{\|\varphi\|_\infty} \sum_{\{x,y\}\in E} w_{xy}|\varphi(x) - \varphi(y)|.$$

Likewise,

$$\mathbb{E}_T\Big(\sum_{x\in A_T} (\mu(x) - \nu(x))\Big) = \sum_{x\in V} (\mu(x) - \nu(x))\frac{\varphi(x)}{\|\varphi\|_\infty} = \frac{1}{\|\varphi\|_\infty},$$

because $\varphi^\top(\mu - \nu) = 1$. Taking the ratio yields the $p = 1$ objective. Finally, we impose the constraint $\varphi \geq 0$ without loss of generality: if $c \geq -\min_{x\in V} \varphi(x)$, then $\varphi + c\mathbf{1} \geq 0$, while $\sum_{x\in V}(\mu(x) - \nu(x)) = 0$ implies $(\varphi + c\mathbf{1})^\top(\mu - \nu) = \varphi^\top(\mu - \nu)$, and adding a constant also leaves all edge differences unchanged. □

Proof of Theorem 3.8. For $1 < p < \infty$, write the Beckmann problem using the notation from Lemma 3.7:

$$\min_{J\in\mathbb{R}^m} \big\{\|J\|_{w^{1-q},q} : BJ = \mu - \nu\big\}.$$

This is a linearly constrained norm minimization problem. By Lemma 3.7, the dual norm of $\|\cdot\|_{w^{1-q},q}$ is $\|\cdot\|_{w,p}$. Gauge duality for feasible linearly constrained norm problems (see Freund (1987); Friedlander et al. (2014)) therefore yields the dual program

$$\min_{\varphi\in\mathbb{R}^n} \big\{\|B^\top\varphi\|_{w,p} : \varphi^\top(\mu - \nu) \geq 1\big\}.$$

This is exactly Equation (3), since any feasible φ with strict inequality can be rescaled to satisfy equality. Strong gauge duality implies that the optimal values are reciprocal.

The endpoint cases $p = 1$ and $p = \infty$ are identical, using the first and third cases of Lemma 3.7. □

Proof of Corollary 3.9. Let $b := \mu - \nu$. By definition of the Beckmann program at $p = 2$ with weights w^{-1},

$$\mathcal{B}_{w^{-1},2}(\mu,\nu)^2 = \min_{\substack{J\in\mathbb{R}^m \\ BJ=b}} \sum_{e\in E} w_e^{-1}|J(e)|^2 = \min_{\substack{J\in\mathbb{R}^m \\ BJ=b}} J^\top W^{-1} J.$$

Since $L = BWB^\top$, the minimum equals $b^\top L^\dagger b$, i.e.

$$\mathcal{B}_{w^{-1},2}(\mu,\nu)^2 = b^\top L^\dagger b = \mathcal{R}_{\text{eff}}(\mu,\nu).$$

Hence $\mathcal{B}_{w^{-1},2}(\mu,\nu) = \sqrt{\mathcal{R}_{\text{eff}}(\mu,\nu)}$. Now apply Theorem 3.8 at $p = 2$:

$$\mathcal{C}_2(\mu,\nu) = \frac{1}{\mathcal{B}_{w^{-1},2}(\mu,\nu)}.$$

Squaring gives Equation (12).

For the minimizer, the $p = 2$ primal is

$$\min_{\varphi^\top b=1} \ \varphi^\top L\varphi.$$

Lagrange optimality gives $L\varphi = \lambda b$, so (modulo additive constants) $\varphi = \lambda L^\dagger b$. Imposing $\varphi^\top b = 1$ yields

$$\lambda = \frac{1}{b^\top L^\dagger b},$$

hence

$$\varphi_\star = \frac{L^\dagger(\mu-\nu)}{(\mu-\nu)^\top L^\dagger(\mu-\nu)}.$$

□

To prove Theorem 3.11 we make use of a calculus lemma.

Lemma A.1. *Let $0 < \alpha < 1$. If $0 < x < \alpha^{-1} - 1$, then*

$$\alpha x + e^{-x} < 1.$$

Proof of Lemma A.1. The bound $e^x \geq 1 + x$ implies $e^{-x} \leq \frac{1}{1+x}$ for $x > 0$. Therefore

$$\alpha x + e^{-x} \leq \alpha x + \frac{1}{1+x} = 1 + x\Big(\alpha - \frac{1}{1+x}\Big).$$

If $x < \alpha^{-1} - 1$, then $\alpha < \frac{1}{1+x}$, so the last term is strictly less than 1. □

Proof of Theorem 3.11. Decompose

$$\varphi - \widetilde{\varphi}_t = L^\dagger(I - e^{-tL})b - L^\dagger e^{-tL}\eta.$$

Hence

$$\|\varphi - \widetilde{\varphi}_t\|_2 \leq \|L^\dagger(I - e^{-tL})b\|_2 + \|L^\dagger e^{-tL}\eta\|_2.$$

Because $L^\dagger$ and e^{-tL} are simultaneously diagonalizable in an orthonormal basis of Laplacian eigenvectors, and both b and η are orthogonal to $\mathbf{1}$, we obtain

$$\|L^\dagger e^{-tL}\eta\|_2 \leq \max_{i\geq 2} \lambda_i^{-1} e^{-t\lambda_i}\|\eta\|_2 \leq \lambda_2^{-1} e^{-t\lambda_2}\|\eta\|_2.$$

Similarly,

$$\|L^{\dagger}(I - e^{-tL})\|_{2\to 2} = \max_{i\geq 2} \frac{1 - e^{-t\lambda_i}}{\lambda_i} \leq \max_{i\geq 2} t = t,$$

since $1 - e^{-u} \leq u$ for $u \geq 0$. This proves Equation (16).

For the improvement claim, divide both sides by $\lambda_2^{-1}\|\eta\|_2$ and set

$$\alpha = \frac{\|b\|_2}{\|\eta\|_2}, \qquad x = t\lambda_2.$$

The assumptions imply $0 < \alpha < 1$ and $0 < x < \alpha^{-1} - 1$, so Lemma A.1 yields

$$\alpha x + e^{-x} < 1,$$

which is exactly the desired strict inequality. □

B Proofs from Section 4

Proof of Lemma 4.1. Completing the square in the variable u gives

$$\mathcal{L}_\sigma(\varphi, u; \widetilde{y}, \widetilde{z}) = s_p(u) + \frac{\sigma}{2}\left\|u - \left(B^\top\varphi + \sigma^{-1}\widetilde{z}\right)\right\|_2^2 - \frac{1}{2\sigma}\|\widetilde{z}\|_2^2 + \widetilde{z}^\top B^\top\varphi + \widetilde{y}(b^\top\varphi - 1) + \frac{\sigma}{2}(b^\top\varphi - 1)^2.$$

The minimizer with respect to u is therefore unique and equal to

$$u_\sigma(\varphi) = \operatorname{prox}_{s_p/\sigma}\left(B^\top\varphi + \sigma^{-1}\widetilde{z}\right).$$

Substituting this expression back into the objective shows that f_σ is convex. Since the minimizer in u is unique for every φ, Danskin's theorem yields differentiability of f_σ and the gradient formula Equation (20). Finally, minimizing $\mathcal{L}_\sigma$ over (φ, u) is equivalent to first minimizing over u and then over φ, so $(\varphi_\star, u_\star)$ is optimal if and only if $u_\star = u_\sigma(\varphi_\star)$ and $\varphi_\star$ satisfies the first-order condition $\nabla f_\sigma(\varphi_\star) = 0$ on $\mathcal{H}$. □

Proof of Lemma 4.2. By Equation (20), ∇f_σ is the sum of a smooth affine map and the composition

$$\varphi \mapsto -\sigma B \operatorname{prox}_{s_p/\sigma}\left(B^\top\varphi + \sigma^{-1}\widetilde{z}\right).$$

The Clarke chain rule therefore implies that every matrix V_Q of the stated form lies in $\partial_C \nabla f_\sigma(\varphi)$.

It remains to prove symmetry and positive semidefiniteness. The proximal map of a proper closed convex function is the gradient of a convex C^1 Moreau envelope. Hence it is differentiable almost everywhere, and at every point of differentiability its Jacobian is symmetric with spectrum contained in $[0, 1]$. Passing to the convex hull of such limiting Jacobians shows that every $Q \in \partial_C \operatorname{prox}_{s_p/\sigma}$ is symmetric and satisfies $0 \preceq Q \preceq I$. Consequently $I - Q \succeq 0$, and for every $x \in \mathcal{H}$,

$$x^\top V_Q x = \sigma(B^\top x)^\top(I - Q)(B^\top x) + \sigma(b^\top x)^2 \geq 0.$$

Thus V_Q is symmetric positive semidefinite on $\mathcal{H}$. □

Proof of Proposition 4.4. We first consider $1 \leq p < \infty$. In this regime, $s_p(u) = \sum_{e\in E} w_e|u_e|^p$ is separable across edges, so

$$\operatorname{prox}_{ts_p}(v)_e = \operatorname{prox}_{tw_e|\cdot|^p}(v_e) \qquad \text{for each } e \in E.$$

It therefore suffices to analyze the scalar map

$$x \longmapsto \arg\min_{u\in\mathbb{R}} \left\{\frac{1}{2}(u - x)^2 + \alpha|u|^p\right\}, \qquad \alpha > 0.$$

If $p = 1$, the minimizer is the soft-threshold map

$$\operatorname{sign}(x)(|x| - \alpha)_+,$$

which is piecewise affine and hence strongly semismooth.

Now let $p \in \{2, 3, \ldots\}$. The optimality condition is

$$u + \alpha p\, u|u|^{p-2} = x.$$

Define

$$g(u) := u + \alpha p\, u|u|^{p-2}.$$

Since p is an integer at least 2, the function g is C^1 with locally Lipschitz derivative, and

$$g'(u) = 1 + \alpha p(p-1)|u|^{p-2} \geq 1.$$

Thus g is a strictly increasing C^1 diffeomorphism of $\mathbb{R}$ onto itself. On every compact interval, the inverse function theorem implies that g^{-1} has locally Lipschitz derivative, hence is strongly semismooth. Since finite Cartesian products of strongly semismooth maps remain strongly semismooth, $\operatorname{prox}_{ts_p}$ is strongly semismooth on $\mathbb{R}^m$.

Finally, consider $p = \infty$. Then $s_\infty(u) = \|u\|_{w,\infty}$, whose dual unit ball is

$$C := \left\{ q \in \mathbb{R}^m : \|q\|_{w^{-1},1} \leq 1 \right\}.$$

By Moreau decomposition,

$$\operatorname{prox}_{ts_\infty}(v) = v - \Pi_{tC}(v),$$

where Π_{tC} denotes Euclidean projection onto the weighted ℓ_1 ball tC. The set tC is polyhedral, so Π_{tC} is piecewise affine. Hence both Π_{tC} and $I - \Pi_{tC}$ are strongly semismooth, which proves the claim. □

Proof of Theorem 4.5. By Lemma 4.1, the map ∇f_σ is the sum of a smooth affine term and the composition of affine maps with $\operatorname{prox}_{s_p/\sigma}$. Hence Proposition 4.4 implies that ∇f_σ is strongly semismooth. By Lemma 4.2, the matrices V_Q furnish generalized Jacobians for ∇f_σ. The stated conclusion is therefore exactly the local convergence theorem for semismooth Newton methods with inexact linear solves and Armijo globalization; see Zhao et al. (2010). □

Proof of Theorem 4.6. Because G is connected, the restriction of $B^\top$ to $\mathcal{H}$ is injective. Hence the objective in Equation (17) is coercive on the feasible set, so the centered primal problem has a nonempty solution set. The problem is a finite-dimensional closed convex program with linear equality constraints, and Equation (23) is precisely Rockafellar's summable inexactness condition for the augmented Lagrangian method. The claimed convergence of primal residuals and the optimality of every cluster point therefore follow directly from the classical ALM theory in Rockafellar (1976a;b). If the centered primal solution is unique, the cluster set is a singleton, and thus the entire sequence converges. □