

ANALYSIS AND ALGORITHMS FOR ℓ_p -BASED SEMI-SUPERVISED LEARNING ON GRAPHS

MAURICIO FLORES, JEFF CALDER, GILAD LERMAN

Department of Mathematics, University of Minnesota

ABSTRACT. This paper addresses theory and applications of ℓ_p -based Laplacian regularization in semi-supervised learning. The graph p -Laplacian for $p > 2$ has been proposed recently as a replacement for the standard ($p = 2$) graph Laplacian in semi-supervised learning problems with very few labels, where Laplacian learning is degenerate.

In the first part of the paper we prove new discrete to continuum convergence results for p -Laplace problems on k -nearest neighbor (k -NN) graphs, which are more commonly used in practice than random geometric graphs. Our analysis shows that, on k -NN graphs, the p -Laplacian retains information about the data distribution as $p \rightarrow \infty$ and Lipschitz learning ($p = \infty$) is sensitive to the data distribution. This situation can be contrasted with random geometric graphs, where the p -Laplacian *forgets* the data distribution as $p \rightarrow \infty$. We also present a general framework for proving discrete to continuum convergence results in graph-based learning that only requires pointwise consistency and monotonicity.

In the second part of the paper, we develop fast algorithms for solving the variational and game-theoretic p -Laplace equations on weighted graphs for $p > 2$. We present several efficient and scalable algorithms for both formulations, and present numerical results on synthetic data indicating their convergence properties. Finally, we conduct extensive numerical experiments on the MNIST, FashionMNIST and EMNIST datasets that illustrate the effectiveness of the p -Laplacian formulation for semi-supervised learning with few labels. In particular, we find that Lipschitz learning ($p = \infty$) performs well with very few labels on k -NN graphs, which experimentally validates our theoretical findings that Lipschitz learning retains information about the data distribution (the unlabeled data) on k -NN graphs.

1. INTRODUCTION

Data science problems, such as regression and classification, are pervasive in today's world, and the size of datasets is growing rapidly. In the supervised setting, data needs to be labeled, requiring substantial effort (e.g. writing a transcript for speech recognition), or may require expert input (deciding whether a brain scan is healthy or not). In contrast, unlabeled data can often be acquired in large quantities with substantially less effort. Semi-supervised learning harnesses the additional information present in unlabeled data to improve learning tasks. This can include geometric or topological properties of unlabeled data, which can provide valuable information about where to place decision boundaries, for instance. This can be contrasted with fully supervised algorithms, which only make use of labeled data. Fully supervised learning algorithms typically learn parameterized functions and require an abundant amount of labeled data.

A common setting within semi-supervised learning is *graph-based* semi-supervised learning, which is concerned with propagating label information on graphs. Here, we are given an undirected weighted graph $G = (\mathcal{X}, \mathcal{W})$, where \mathcal{X} are the vertices and $\mathcal{W} = \{w_{xy}\}_{x,y \in \mathcal{X}}$ are nonnegative edge weights, which are chosen so that $w_{xy} \approx 1$ when x is similar to y , and $w_{xy} \approx 0$ when x and y are dissimilar. Each vertex x in the *observation set* $\mathcal{O} \subset \mathcal{X}$ is assigned a label $g(x)$, where

E-mail address: mauricio.a.flores.math@gmail.com, jcalder@umn.edu, lerman@umn.edu.

Source Code: <https://github.com/mauriciofloresML/Laplacian-Lp-Graph-SSL.git>

Funding: The authors gratefully acknowledge National Science Foundation grants 1713691, 1821266, 1830418, and a University of Minnesota Grant in Aid Award.

$g : \mathcal{O} \rightarrow \mathbb{R}^k$. In a classification problem with k classes, the i^{th} class is usually assigned the label vector $g(x) = e_i$, where e_i is the i^{th} standard basis vector in \mathbb{R}^k , that is, the vector with all zeros and a one in the i^{th} coordinate (called a “one-hot” vector in machine learning). The task of graph-based semi-supervised learning is to extend the labels from the observation set \mathcal{O} to a label function $u : \mathcal{X} \rightarrow \mathbb{R}^k$ on the whole graph in some meaningful way. In practice, the equations that are solved for propagating labels are separable among the coordinates of \mathbb{R}^k , and the problem reduces to solving for k functions $u_i : \mathcal{X} \rightarrow \mathbb{R}$, one for each class $i = 1, \dots, k$, and setting $u = (u_1, \dots, u_k)$ (this is referred to as the “one-vs-rest” approach in machine learning). Thus, we can, without loss of generality, focus on the scalar case $k = 1$ for algorithms and analysis in this paper.

Since the problem of extending labels is *a priori* an ill-posed problem (there are infinitely many solutions), one usually makes the semi-supervised *smoothness assumption*, which asks that the learned labeling function $u : \mathcal{X} \rightarrow \mathbb{R}$ should be smooth in dense regions of the graph [18]. The smoothness assumption is often enforced by defining a functional (or regularizer) $J(u)$ that measures the *smoothness* of a labeling $u : \mathcal{X} \rightarrow \mathbb{R}$, and then minimizing $J(u)$ subject to either hard label constraints $u(x) = g(x)$ for $x \in \mathcal{O}$, or a soft penalty constraint like the mean squared error in the labels. Soft penalties are useful when the labels are corrupted by noise. In this paper, we are concerned with learning problems with very few labels, in the range of one label per class, so a basic assumption we make is that the labels are clean and are not corrupted by noise. Thus, the hard constraint is natural to impose and nothing is gained by considering a soft constraint. In fact, the soft-constraint is more likely to be ignored, when there are very few labels, unless the penalty parameter is chosen sufficiently large so that the constraint is essentially a hard one. The soft-constrained problem would be interesting to consider in the context of more moderate label rates, where noisy labels can be better tolerated. All of the techniques we discuss in this paper extend directly, with minor modifications, to problems with soft constraints.

One of the most widely used methods in semi-supervised learning is Laplacian regularization [62], which uses the smoothness functional

$$(1.1) \quad J_2(u) := \frac{1}{4} \sum_{x,y \in \mathcal{X}} w_{xy} (u(x) - u(y))^2.$$

Minimizing J_2 attempts to force similar data points in dense regions of the graph to have similar labels. Minimizers of J_2 are graph harmonic, and solve the graph 2-Laplace equation $\Delta_2^G u = 0$, where

$$(1.2) \quad \Delta_2^G u(x) := \sum_{y \in \mathcal{X}} w_{xy} (u(y) - u(x)).$$

In classification, the values of u are rounded to the nearest label. Laplacian regularization, and ideas based upon it, are very widely used in machine learning [3, 31, 32, 54, 56–59, 61], and have achieved great successes. However, it has been noted first in [43] and later in [21], that Laplacian regularization becomes ill-posed (degenerate) in problems with very few labels. We say a graph-based learning problem is ill-posed in the limit of infinite unlabeled data and finite labeled data if the sequence of learned functions does not continuously attain the labeled (e.g., boundary) data in the continuum limit. In this case, the learned function becomes nearly constant on the whole graph, with sharp spikes near the labeled data. Thus, even with a hard constraint the labels are almost entirely ignored. See Figure 1a for a depiction of this degeneracy. In the continuum, this is merely reflecting the fact that the capacity of a point is zero in dimension $d \geq 2$ [38].

To address this issue, El Alaoui et al. [21] proposed a class of ℓ_p -based Laplacian regularizers, which use the smoothness functional

$$(1.3) \quad J_p(u) := \frac{1}{2p} \sum_{x,y \in \mathcal{X}} w_{xy} |u(x) - u(y)|^p.$$

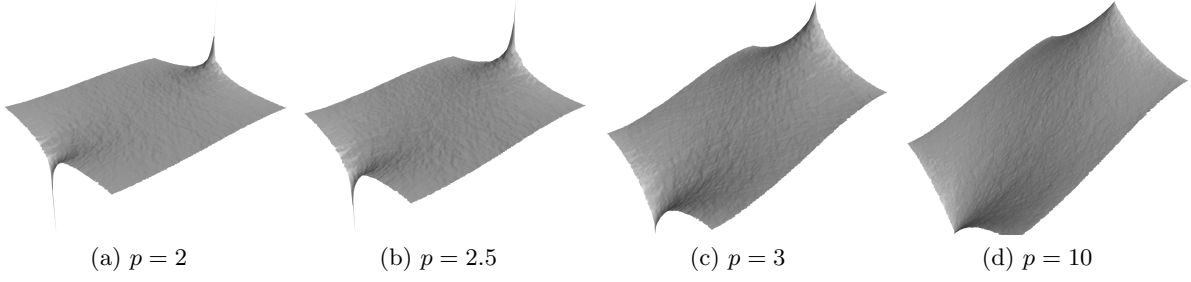


FIGURE 1. Numerical results for a toy learning problem with two labeled and 10^5 unlabeled data points on $[0, 1]^2$. For $p = 2$ the surface is nearly constant, with spikes near the labeled points, while as p becomes larger the surface becomes smoother. The result for $p = \infty$ looks very similar to $p = 10$, and the result does not change much for $10 \leq p < \infty$.

Choosing $p > 2$ places a heavier penalty on large gradients $|u(x) - u(y)|$, which discourages the solution from developing sharp spikes. We also note that choosing $1 \leq p < 2$, often $p = 1$, encourages the gradient $|u(x) - u(y)|$ to be sparse, and can be viewed as a relaxation of graph-cut energies. This can yield good results for classification at moderate label rates (see, e.g., [35]), but at very low label rates the issue with spikes is even more pronounced when $p < 2$, and the results are similar to Laplace learning.

We note that minimizers of J_p satisfy the graph p -Laplace equation $\Delta_p^G u = 0$, where

$$(1.4) \quad \Delta_p^G u(x) := \sum_{y \in \mathcal{X}} w_{xy} |u(x) - u(y)|^{p-2} (u(y) - u(x)).$$

We call (1.4) the *variational graph p -Laplacian*. Figure 1 depicts ℓ_p -regularization for different values of p . As p increases the learned function transitions more smoothly between labeled and unlabeled points. From a continuum perspective, the energy J_p is related to the p -Dirichlet energy $\int_{\Omega} |\nabla u|^p dx$, and the Sobolev embedding $W^{1,p}(\Omega) \hookrightarrow C^{0,1-d/p}(\Omega)$ allows isolated boundary points when $p > d$, where d is the dimension.¹ Indeed, by Morrey's inequality [25] we have

$$(1.5) \quad |u(x) - u(y)| \leq C \left(\int_{\Omega} |\nabla u|^p dx \right)^{1/p} |x - y|^{1-d/p}$$

whenever $p > d$ and $|x - y| \leq \frac{1}{2} \text{dist}(x, \partial\Omega)$. Morrey's inequality implies that u is Hölder continuous, and prevents spikes in Figure 1a from occurring. In particular, the continuum p -Dirichlet problem (see (1.10)) with constraints at isolated points is well-posed (e.g., admits a unique solution attaining the boundary data continuously) if and only if $p > d$.

The variational graph p -Laplacian (1.4) has appeared previously in machine learning [2, 6, 60], but was first suggested for problems with few labels in [21] with $p \geq d + 1$. Recently, it was rigorously proven that ℓ_p -based regularization is ill-posed (its minimizer is degenerate) for $p \leq d$, and well-posed for $p > d$ in the continuum limit of infinite unlabeled and finite labeled data [50]. This justifies the continuum heuristics described above.

Formally sending $p \rightarrow \infty$ in (1.3) one obtains *Lipschitz learning* [36, 41], which corresponds to the smoothness functional

$$(1.6) \quad J_{\infty}(u) = \max_{x,y \in \mathcal{X}} w_{xy} |u(x) - u(y)|.$$

¹Here, $\Omega \subset \mathbb{R}^d$ is an open, bounded domain, and $W^{1,p}(\Omega)$ is the Sobolev space of functions $u : \Omega \rightarrow \mathbb{R}$ such that $\int_{\Omega} u^p + |\nabla u|^p dx < \infty$.

We note that minimizers of (1.6) are not unique. To see why this is the case, note that if $w_{xy}|u(x) - u(y)| < J_\infty(u)$, then we can change the values of $u(x)$ and $u(y)$ slightly, without changing the value of $J_\infty(u)$. So minimizers cannot be unique unless the maximum gradient is attained everywhere. Among the non-unique minimizers, one generally looks for one whose gradient cannot be locally improved (i.e., made smaller by adjusting some values of $u(x)$). More precisely, in [36] the authors show that there is a unique minimizer whose gradient is smallest in the lexicographical order, called the *lex-minimizer*. This turns out to be equivalent to the notion of *absolutely minimal*, which has been used in the partial differential equation (PDE) and analysis community to select the unique Lipschitz extension in the continuum [4].

Lex-minimizers of (1.6) satisfy the graph ∞ -Laplace equation $\Delta_\infty^G u = 0$ where

$$(1.7) \quad \Delta_\infty^G u(x) := \min_{y \in \mathcal{X}} w_{xy}(u(y) - u(x)) + \max_{y \in \mathcal{X}} w_{xy}(u(y) - u(x)).$$

To see why (1.7) is the correct form for the graph ∞ -Laplacian, consider a graph p -harmonic function u , which satisfies $\Delta_p^G u(x) = 0$, where Δ_p^G is defined in (1.4). We split the terms in the sum defining $\Delta_p^G u(x)$ by their sign to obtain

$$\sum_{u(y) > u(x)} w_{xy}(u(y) - u(x))^{p-1} = \sum_{u(x) > u(y)} w_{xy}(u(x) - u(y))^{p-1}.$$

Taking the p^{th} root of both sides and sending $p \rightarrow \infty$ yields

$$\max_{y \sim x} (u(y) - u(x)) = \max_{y \sim x} (u(x) - u(y)),$$

where we write $y \sim x$ if $w_{xy} > 0$. This can be simplified to

$$\max_{y \sim x} (u(y) - u(x)) + \min_{y \sim x} (u(y) - u(x)) = 0,$$

which is the graph ∞ -Laplacian defined in (1.7) for an unweighted graph. To obtain the weighted graph ∞ -Laplacian (1.7) in the limit as $p \rightarrow \infty$, we simply replace w_{xy} in (1.4) with w_{xy}^p .

It was proven in [10] that Lipschitz learning is well-posed with arbitrarily few labels, and the continuum limit on random geometric graphs is the continuum ∞ -Laplace equation (see (1.11)). Several papers (see, e.g., [10, 21]) have noted that the continuum ∞ -Laplacian (1.11) does not involve the data distribution, making it presumably unsuitable for semi-supervised learning, which is supposed to use properties of the unlabeled data, often through its distribution. In [10], it was shown how to re-weight the graph to introduce varying degrees of sensitivity to the data distribution in Lipschitz learning.

In order to combine the well-posedness of Lipschitz learning with the distributional sensitivity of Laplacian regularization, it is natural to augment the 2-Laplacian with a small ∞ -Laplace term and solve an equation of the form $\Delta_2^G u + \varepsilon \Delta_\infty^G u = 0$. To this end, we define the *game theoretic* p -Laplacian on the graph (the name will be explained shortly)

$$(1.8) \quad \mathcal{L}_p^G u(x) = \frac{1}{d_x p} \Delta_2^G u(x) + \lambda \left(1 - \frac{2}{p}\right) \Delta_\infty^G u(x),$$

where $d_x = \sum_{y \in \mathcal{X}} w_{xy}$ is the degree of vertex x , and $\lambda > 0$ is a constant. For semi-supervised learning with the game-theoretic p -Laplacian we solve $\mathcal{L}_p^G u = 0$ subject to $u = g$ on \mathcal{O} (see Section 1.1 for precise definitions). The second author proved in [8] that the game theoretic p -Laplacian is well-posed with very few labels for $p > d$, and argued for the use of this formulation as an alternative regularization for semi-supervised learning on graphs. In the context of these results, λ is chosen (explicitly) depending on the kernel used to define the weights w_{xy} , in order to produce the consistency results described below. Since having two parameters, p and λ , is redundant in practice, we take $\lambda = 1$ in the numerical sections of the paper. Compared to the variational p -Laplacian, the game-theoretic p -Laplacian appears better conditioned numerically when p is large, since it

does not require computing large powers of p . Another main difference is that the game-theoretic p -Laplacian does not arise through an optimization problem, and so the methods for solving the equation are somewhat different. We note that the game-theoretic graph p -Laplacian (and similar models) have been used very recently for data clustering and learning problems [22–24, 30], though not in the context of very few labeled data points. A related definition of the game-theoretic p -Laplacian on graphs was also studied in [42]. We also mention recent work [15, 49] that approaches the semi-supervised learning problem with few labels by re-weighting the graph so that the weights w_{xy} are large near labels.

Both the variational (1.4) and the game-theoretic (1.8) graph p -Laplace equations are consistent in the continuum, on random geometric graphs, with the p -Laplace equation

$$(1.9) \quad \Delta_p u := \operatorname{div}(|\nabla u|^{p-2} \nabla u) = 0.$$

It is important to point out that the weights $1/p$ and $\lambda(1 - 2/p)$ in (1.8) are chosen precisely so that \mathcal{L}_p^G is consistent with Δ_p . The operator Δ_p is called the *p-Laplacian*, and solutions of (1.9) are called *p-harmonic functions* [40]. The p -Laplace equation arises as the necessary conditions (Euler-Lagrange equation) for the p -Dirichlet problem

$$(1.10) \quad \min_u \int_{\Omega} |\nabla u|^p dx.$$

Note that we can expand the p -Laplacian to obtain

$$\Delta_p u = |\nabla u|^{p-2} (\Delta u + (p-2) \Delta_{\infty} u),$$

where Δ_{∞} is the ∞ -Laplacian, given by

$$(1.11) \quad \Delta_{\infty} u := \frac{1}{|\nabla u|^2} \sum_{i,j=1}^d u_{x_i x_j} u_{x_i} u_{x_j}.$$

Thus, any solution of $\Delta_p u = 0$ also satisfies

$$(1.12) \quad \frac{1}{p} \Delta u + \left(1 - \frac{2}{p}\right) \Delta_{\infty} u = 0.$$

The left hand side in (1.12) is often called the *game-theoretic* or *homogeneous p-Laplacian* since it arises in two player stochastic tug-of-war games [39, 47]. This justifies the definition (1.8) of the game-theoretic graph p -Laplacian. We note that while the p -Laplace equation (1.9) is equivalent to the game theoretic version (1.12) at the continuum level, these are different formulations at the discrete level.

Given the recent interest in graph p -Laplacian models in machine learning, it is important to have both strong theoretical results that are relevant in practice, and efficient and scalable algorithms for solving the equations in real-world settings. Most of the literature on discrete to continuum convergence in graph-based learning, such as the recent work on the p -Laplacian [8, 50], assumes the graph is a random geometric graph. However, such graphs have poor sparsity properties², and practitioners almost always use some form of a k -nearest neighbor (k -NN) graph instead. Thus, it is important to develop theory for the graph p -Laplacian on k -NN graphs in order to study the graphs that are used in practice. It turns out, looking forward to Section 2, that the discrete to continuum theory for the p -Laplacian on k -NN graphs is fundamentally different from the existing theory on random geometric graphs, and this discrepancy accounts for the results of our numerical experiments conducted later in the paper.

²As an example of the poor sparsity properties of random geometric graphs, consider the MNIST dataset, which has 70,000 images of handwritten digits. A k -nearest neighbor graph on MNIST with $k = 3$ neighbors is connected, while the sparsest random geometric graph that is connected requires a bandwidth of $\varepsilon = 8.5$ and has on average 294 neighbors per image, which is roughly 100 times less sparse than the k -nearest neighbor graph. Graph connectivity is one of the basic properties required by most graph-based learning algorithms.

From a computational perspective, there are relatively few works on fast algorithms for graph p -Laplacians. Kyng et al., [36] developed an efficient algorithm for Lipschitz learning ($p = \infty$). Their algorithm has a poor worst case complexity analysis (roughly quadratic complexity in the number of data points), but seems to run very fast in practice. Oberman [45] considers the game-theoretic formulation on regular grids in dimensions $d = 2, 3$, and developed a fast semi-implicit solution method, as well as gradient-descent methods. One contribution of this paper is an adaptation of Oberman's semi-implicit method to the graph setting. Other works [22–24] use slow iterative methods, such as Jacobi iteration or gradient descent. It was suggested in [21] to use Newton's method for the variational p -Laplacian, but the method was not investigated in any depth. The energy J_p is smooth and convex, but not strongly convex when $p > 2$. Other works, such as [36], suggest to use convex programming to solve the variational p -Laplacian.

1.1. Main results and contributions. Before summarizing the main contributions of our paper, let us give precise definitions of the problems we study. We let $G = (\mathcal{X}, \mathcal{W})$ be a connected graph with vertices \mathcal{X} and edge weights $\mathcal{W} = \{w_{xy}\}_{x,y \in \mathcal{X}}$. The subset of labeled vertices is denoted $\mathcal{O} \subset \mathcal{X}$, and the label function is $g : \mathcal{O} \rightarrow \mathbb{R}$. We denote the number of data points in \mathcal{O} by m , and the number of data points in \mathcal{X} by n . The ℓ_p -based Laplacian regularized learning problem [21] is given by

$$(1.13) \quad \min_{u: \mathcal{X} \rightarrow \mathbb{R}} J_p(u) \quad \text{subject to } u(x) = g(x) \text{ for all } x \in \mathcal{O},$$

where we extend the definition of J_p to be

$$(1.14) \quad J_p(u) := \frac{1}{2p} \sum_{x,y \in \mathcal{X}} w_{xy} |u(x) - u(y)|^p + \sum_{x \in \mathcal{X}} f(x)u(x),$$

where f is a source function. The unique minimizer $u : \mathcal{X} \rightarrow \mathbb{R}$ of (1.13) satisfies the optimality conditions

$$(1.15) \quad \begin{cases} -\Delta_p^G u(x) = f(x) & \text{if } x \in \mathcal{X} \setminus \mathcal{O} \\ u(x) = g(x) & \text{if } x \in \mathcal{O}, \end{cases}$$

where Δ_p^G is defined in (1.4). In machine learning applications we always take $f \equiv 0$ in (1.14) and (1.15). However, it is useful to formulate the equation in more generality so that we can construct exact solutions by choosing $u(x)$ and computing $f(x) = -\Delta_p^G u(x)$ accordingly. This allows us to evaluate and compare the convergence rates of different methods. We will refer to (1.13) as the **variational problem**, and to (1.15) as the **variational p -Laplace equation**.

The game-theoretic graph p -Laplacian equation is given by

$$(1.16) \quad \begin{cases} -\mathcal{L}_p^G u(x) = f(x) & \text{if } x \in \mathcal{X} \setminus \mathcal{O} \\ u(x) = g(x) & \text{if } x \in \mathcal{O}, \end{cases}$$

where \mathcal{L}_p^G is the game-theoretic graph p -Laplacian defined in (1.8). We will refer to (1.16) as the **game-theoretic problem**. As before, we are only concerned with $f \equiv 0$ in machine learning, but it is convenient to proceed in generality. The **Lipschitz learning problem** corresponds to the game-theoretic problem with $p = \infty$.

We now summarize the contributions of the paper.

1.1.1. Discrete to continuum on k NN graphs. In Section 2, we give a detailed analysis of discrete to continuum convergence of graph-based algorithms on k -nearest neighbor graphs. We prove that the continuum limit of the game-theoretic p -Laplacian on *symmetrized* k -NN graphs has a significantly different form compared to random geometric graphs. In particular, we show that the continuum operator has an additional drift term along the gradient of the data distribution that *does not* vanish as $p \rightarrow \infty$. The drift term arises through the symmetrization process in the k -NN graph

construction. This means the conventional wisdom that Lipschitz learning is not sensitive to the data distribution (see, e.g., [10, 21, 50]) is a phenomenon specific to random geometric graphs, and does not hold true for k -NN graphs, which highlights the need to perform analysis on different graph constructions. This result is also borne out in our experimental results in Section 4, which show that the $p = \infty$ graph Laplacian gives good results for semi-supervised learning with very few labels, indicating sensitivity to the data distribution as predicted by our theoretical results.

We also prove a discrete to continuum convergence result for general elliptic learning algorithms on graphs. The result is very general and only requires the equation to have a certain monotonicity property (essentially ellipticity) that allows for the comparison principle to be used. The results show that all elliptic semi-supervised learning algorithms are well-posed at label rates as low as $O(\varepsilon)$, where $\varepsilon > 0$ is the bandwidth of the graph (or average distance to the k^{th} nearest neighbor in a k -NN graph). This gives a baseline for comparing algorithms at low label rates, and indicates that algorithms should only be claimed to be superior at low label rates if they are well-posed at label rates significantly less than $O(\varepsilon)$. For example, recent work [16] by the second author shows that Laplacian regularization ($p = 2$) is well-posed at label rates as low as $O(\varepsilon^2)$, and previous work [8, 50] showed that the p -Laplacian is well-posed for arbitrarily low label rates when $p > d$. It is an open problem to determine the lowest label rates for which p -Laplacian regularization is well-posed for $2 < p < d$.

1.1.2. Efficient algorithms. We develop and study a range of algorithms for solving both the variational and game-theoretic p -Laplace equations, and identify the algorithms that are efficient and scalable in each setting. In each case, we conduct numerical experiments on synthetic data to measure execution time and its dependence on the intrinsic dimensionality of the graph,

For the variational p -Laplacian, we propose to use Newton’s method with homotopy on p . Other solvers, such as iteratively reweighted least squares (IRLS) and a primal-dual method, were considered in [27]. The primal dual method is slower than Newton with homotopy, and IRLS converges only for $p < 3$, though recent work has found ways to bypass this restriction [1].

For the game theoretic p -Laplacian, we study a gradient descent-like algorithm, a new Newton-like method, and a semi-implicit method, which is an extension of Oberman’s method [45] to graphs. The Newton-like method with homotopy on p , and the semi-implicit method converge faster than gradient descent, but do not have provable convergence guarantees. The gradient descent-like method is slower, though provably convergent (see Theorem 3.7).

All algorithms are presented and analyzed in Section 3, while the numerical experiments on synthetic data are presented in Section 4

1.1.3. Experimental study. We conduct a thorough experimental study of p -Laplacian semi-supervised learning on real data, including MNIST [37], Fashion MNIST [55], and Extended MNIST [19]. In particular, we study classification problems with very few labels, and show that graph p -Laplacian learning with $p > 2$ is superior to Laplacian regularization. Our results show that p -Laplacian learning becomes more accurate when provided with more unlabeled data, which confirms the semi-supervised learning paradigm. The experiments on real data are presented in Section 5.

1.1.4. Source code. The code for all numerical experiments is available online³, and the p -Laplacian semi-supervised learning algorithm is implemented in the GraphLearning Python package [12].

2. CONTINUUM LIMITS ON K -NEAREST NEIGHBOR GRAPHS

Since random geometric graphs (also called ε -ball graphs) generally have poor sparsity, it is common in practice to use k -nearest neighbor (k -NN) graphs, where each point is connected to its k -nearest neighbors. However, there are very few discrete to continuum or consistency results for graph Laplacians on k -NN graphs. The only results on k -NN graphs that we are aware of are

³https://github.com/mauriciofloresML/Laplacian_Lp_Graph_SSL.git.

pointwise consistency results (without rates) [51], Γ -convergence results for variational problems [28], and spectral convergence rates [14].

We give here a detailed analysis of pointwise consistency and discrete to continuum convergence for graph p -Laplace equations on various k -NN and ε -ball graphs (Sections 2.2, 2.3, and 2.4). After a careful study, we find that previous observations about p -Laplacian regularization on ε -graphs do not hold on k -NN graphs. In particular, previous work [8, 21, 50] has shown that p -Laplacian regularization *forgets* the distribution of the unlabeled data as $p \rightarrow \infty$, which renders the algorithm unsuitable for semi-supervised learning. We show in the following sections (see Remarks 2.6, 2.11, and 2.15) that this observation is true *only* for ε -ball graphs, and the situation is completely different for symmetrized k -NN graphs. For symmetrized k -NN graphs, even Lipschitz learning ($p = \infty$) is sensitive to the distribution of unlabeled data.

We also give, in Section 2.5, a general framework for proving discrete to continuum convergence results for elliptic equations on graphs. The main result, Theorem 2.23, gives a general discrete to continuum convergence result for a wide class of elliptic equations on graphs, and shows that all suitable semi-supervised learning algorithms are well-posed at label rates of $O(\varepsilon)$ and higher. This result indicates that algorithms for semi-supervised learning at low label rates should be judged by their ability to operate below the $O(\varepsilon)$ threshold, and theoretical results should aim to establish this.

Let us remark that the pointwise consistency results in this section share some similarities, in terms of proof techniques, to the results in [8] and [14]. In [8], the second author established pointwise consistency, without any rate, for the game theoretic p -Laplacian on ε -ball graphs. In Theorem 2.3 we extend this result to an $O(\varepsilon)$ convergence rate. This is already well-known for $p = 2$ (see [33]), and since the p -Laplacian is a convex combination of the $p = 2$ and $p = \infty$ Laplacians, we quote existing results for $p = 2$ (actually, we quote the $p = 2$ result for ε -ball graphs from [14] in the proof of Theorem 2.3, since the form of the result from [14] is simpler to use).

For k -NN graphs, the only existing pointwise consistency results with rates are the $p = 2$ results in the manifold setting established by the second author in [14]. The only other work we are aware of on pointwise consistency for k -NN graphs is [51], which considers $p = 2$ and proves consistency without any rates. In this section, we prove pointwise consistency with linear convergence rates for the game-theoretic p -Laplacian on k -NN graphs, both symmetrized (Theorem 2.14) and non-symmetrized (Theorem 2.10). The analogous results for $p = 2$ were established by the second author in [14], and these results can be viewed as an extension of those to the p -Laplacian with $p > 2$. Naturally, we make use of the $p = 2$ results from [16], since the game-theoretic p -Laplacian is a convex combination of the $p = 2$ and $p = \infty$ graph Laplacians. For both k -NN and ε -ball graphs, we prove linear rates of $O(\varepsilon)$ (where $\varepsilon = (\frac{k}{n})^{1/d}$ for k -NN graphs). For ε -ball graphs and $p = 2$, it is well-known that sharper $O(\varepsilon^2)$ pointwise consistency rates are available (see, e.g., [11, Remark 5.26]). It would be interesting to investigate whether these sharper rates can be extended to k -NN graphs.

For reference, we include here a table of notation used in this section.

Notation.

- n : Number of vertices in the graph.
- ε : Graph connectivity length scale for ε -ball graphs.
- k : Number of neighbors in a k -NN graph.
- η_ε : Rescaled weight kernel $\eta_\varepsilon(t) = \eta(\frac{t}{\varepsilon})$.
- \mathcal{X}_n : Vertices of our graphs—an *i.i.d.* sample of size n with density ρ on $\Omega \subset \mathbb{R}^d$.
- β : Bound on the density $0 < \beta \leq \rho \leq \beta^{-1}$.
- r_0 : Unique maximum of $r \mapsto r\eta(r)$.

$d^{n,\varepsilon}$: Degrees on an ε -ball graph.

$N_\varepsilon(x)$: The number of points in the set $\mathcal{X}_n \cap B(x, \varepsilon)$.

$\varepsilon_k(x)$: The distance from x to its k^{th} nearest neighbor in \mathcal{X}_n .

$s_k(x)$: Typical distance from x to its k^{th} nearest neighbor (satisfies $\alpha(d)ns_k(x)^d\rho(x) = k$).

$\varepsilon_k(x, y)$: Symmetrization of $\varepsilon_k(x)$, given by $\varepsilon_k(x, y) = \max\{\varepsilon_k(x), \varepsilon_k(y)\}$.

$s_k(x, y)$: Symmetrization of $s_k(x)$, given by $s_k(x, y) = \max\{s_k(x), s_k(y)\}$.

$\mathcal{L}^{n,\varepsilon}$: Unnormalized graph Laplacian on an ε -ball graph.

$\mathcal{L}_{rw}^{n,\varepsilon}$: Random walk graph Laplacian on an ε -ball graph.

$\mathcal{L}_{a,rw}^{n,k}$: Random walk graph Laplacian on a nonsymmetric k -NN graph.

$\mathcal{L}_{s,rw}^{n,k}$: Random walk graph Laplacian on a symmetric k -NN graph.

$\mathcal{L}_p^{n,\varepsilon}$: Game-theoretic graph p -Laplacian on an ε -ball graph.

$\mathcal{L}_{a,p}^{n,k}$: Game-theoretic graph p -Laplacian on a nonsymmetric k -NN graph.

$\mathcal{L}_{s,p}^{n,k}$: Game-theoretic graph p -Laplacian on a symmetric k -NN graph.

2.1. Graph construction. Some parts of the construction of the random graphs are common to ε -ball graphs and k -NN graphs, and we review this now. Let X_1, X_2, \dots, X_n be a sequence of *i.i.d.* random variables on an open connected domain $\Omega \subset \mathbb{R}^d$ with a probability density $\rho : \Omega \rightarrow [0, \infty)$. We assume the boundary $\partial\Omega$ is smooth, $\rho \in C^2(\bar{\Omega})$ and there exists $\beta > 0$ such that

$$(2.1) \quad \beta \leq \rho(x) \leq \beta^{-1} \quad \text{for all } x \in \Omega.$$

Let $\partial_r\Omega = \{x \in \Omega : \text{dist}(x, \partial\Omega) \leq r\}$ and $\Omega_r = \Omega \setminus \partial_r\Omega$. The vertices of our graph are

$$(2.2) \quad \mathcal{X}_n := \{X_1, X_2, \dots, X_n\}.$$

Let $\eta : [0, \infty) \rightarrow [0, \infty)$ be smooth and nonincreasing such that $\eta(t) \geq 1$ for $0 \leq t \leq \frac{1}{2}$, and $\eta(t) = 0$ for $t > 1$. For $\varepsilon > 0$ define $\eta_\varepsilon(t) = \eta(\frac{t}{\varepsilon})$ and set $\sigma_\eta = \int_{\mathbb{R}^n} |z_1|^2 \eta(|z|) dz$. Since the graph is unchanged under scaling the weights by a constant, we may as well assume that

$$(2.3) \quad \int_{B(0,1)} \eta(z) dz = 1.$$

Then, in particular, $\int_{B(0,\varepsilon)} \eta_\varepsilon(z) dz = 1$ as well. As in [8], we assume there exists $r_0 \in (0, 1)$ and $\theta > 0$ so that

$$(2.4) \quad r\eta(r) + \theta(r - r_0)^2 \leq r_0\eta(r_0) \quad \text{for all } 0 \leq r \leq 1.$$

Remark 2.1. The condition (2.4) says that $r \mapsto r\eta(r)$ has a unique maximum, at $r = r_0$, and is strongly concave near the maximum. Looking forward to the definition of the graph ∞ -Laplacian (e.g., (2.8)) and Lemma 2.4, the max and min over $y \in \mathcal{X}_n$ in (2.8) turn out to occur at a distance of $|x - y| \sim \varepsilon r_0$ from x , at least asymptotically as $n \rightarrow \infty$ and $\varepsilon \rightarrow 0$, and the quantitative assumption (2.4) allows us to control this approximation error. If the maximum of $r \mapsto r\eta(r)$ is not unique, then the graph ∞ -Laplacian may not be asymptotically consistent with the ∞ -Laplace operator Δ_∞ .

It is possible to construct nonincreasing η where (2.4) is not satisfied. For example, consider the kernel

$$\eta(r) = \begin{cases} 2, & \text{if } 0 \leq r \leq \frac{1}{2} \\ \frac{1}{r}, & \text{if } \frac{1}{2} \leq r \leq \frac{3}{4} \\ 0, & \text{if } r \geq \frac{3}{4}. \end{cases}$$

The maximum of $r \mapsto r\eta(r)$ is attained on the interval $[\frac{1}{2}, \frac{3}{4}]$ where $r\eta(r) = 1$. We can mollify η to obtain a smooth kernel with the same property on the slightly smaller interval $[\frac{1}{2} + \delta, \frac{3}{4} - \delta]$ for any

$\delta > 0$. We remark that the commonly used Gaussian kernel $\eta(r) = e^{-\frac{r^2}{2\sigma^2}}$ satisfies (2.4) (i.e., the maximum of $f(r) := r\eta(r)$ occurs at $r = \sigma$ and $f''(\sigma) = -\frac{2f(\sigma)}{\sigma} < 0$).

Remark 2.2. We note it is also common to make the *manifold assumption*, where X_1, X_2, \dots, X_n are a sequence of *i.i.d.* random variables sampled from an m -dimensional manifold \mathcal{M} embedded in \mathbb{R}^d , where $m \ll d$. We expect that the analysis here will carry over to the manifold setting with additional technical details.

2.2. ε -ball graphs. The graph constructed with vertices \mathcal{X}_n and edge weights $w_{xy} = \eta_\varepsilon(|x - y|)$ is called a *random geometric graph*, or sometimes an *ε -ball graph*, and is the most widely used graph construction in theoretical analysis of graph-based learning algorithms. Here, we review pointwise consistency for the p -Laplacian on ε -ball graphs. Much of the consistency theory was established previously in [8], so this section is mostly review, with the additional observation that the arguments in [8] establish pointwise consistency *rates* for the ∞ -Laplacian.

We define the graph Laplacian on a random geometric graph by

$$(2.5) \quad \mathcal{L}^{n,\varepsilon} u(x) = \frac{2}{\sigma_\eta n \varepsilon^{d+2}} \sum_{y \in \mathcal{X}_n} \eta_\varepsilon(|x - y|) (u(y) - u(x)).$$

We also define the degree

$$(2.6) \quad d^{n,\varepsilon}(x) = \sum_{y \in \mathcal{X}_n} \eta_\varepsilon(|x - y|).$$

The *random walk* graph Laplacian is defined by

$$(2.7) \quad \mathcal{L}_{rw}^{n,\varepsilon} u(x) = \frac{2}{\sigma_\eta \varepsilon^2 d^{n,\varepsilon}(x)} \sum_{y \in \mathcal{X}_n} \eta_\varepsilon(|x - y|) (u(y) - u(x)).$$

The random walk Laplacian is the generator for a random walk on the graph with transition probabilities $\eta_\varepsilon(|x - y|)/d^{n,\varepsilon}(x)$ of transitioning from x to y . The graph ∞ -Laplacian is defined by

$$(2.8) \quad \mathcal{L}_\infty^{n,\varepsilon} u(x) = \frac{1}{r_0^2 \eta(r_0) \varepsilon^2} \left(\min_{y \in \mathcal{X}_n} \{ \eta_\varepsilon(|x - y|) (u(y) - u(x)) \} + \max_{y \in \mathcal{X}_n} \{ \eta_\varepsilon(|x - y|) (u(y) - u(x)) \} \right),$$

and the game-theoretic graph p -Laplacian is defined by

$$(2.9) \quad \mathcal{L}_p^{n,\varepsilon} u(x) = \frac{1}{p} \mathcal{L}_{rw}^{n,\varepsilon} u(x) + \left(1 - \frac{2}{p}\right) \mathcal{L}_\infty^{n,\varepsilon} u(x).$$

In the continuum, $\mathcal{L}_p^{n,\varepsilon}$ is consistent with the weighted p -Laplace operator

$$(2.10) \quad \Delta_p u = \frac{1}{p} \rho^{-2} \operatorname{div}(\rho^2 \nabla u) + \left(1 - \frac{2}{p}\right) \Delta_\infty u,$$

as is shown in the following theorem.

Theorem 2.3 (Consistency on ε -graphs). *There exists $C_1, C_2, C_3 > 0$ such that for any $\varepsilon > 0$ with $n\varepsilon^d \geq 1$ and any $0 < \lambda \leq 1$, the event that*

$$(2.11) \quad \max_{x \in \mathcal{X}_n \setminus \partial_\varepsilon \Omega} |\mathcal{L}_p^{n,\varepsilon} u(x) - \Delta_p u(x)| \leq C_1 \left(\|u\|_{C^2(B(x,\varepsilon))}^2 |\nabla u(x)|^{-1} \theta^{-1} \varepsilon + \|u\|_{C^3(B(x,\varepsilon))} (\lambda + \varepsilon) \right)$$

holds for all $u \in C^3(\overline{\Omega})$ has probability at least $1 - C_2 n \exp(-C_3 n \varepsilon^{3d/2} \lambda^{d/2}) - C_2 n \exp(-C_3 n \varepsilon^{d+2} \lambda^2)$.

Proof. It was shown in [8, Theorem 5] that the event that

$$(2.12) \quad \max_{x \in \mathcal{X}_n \setminus \partial_\varepsilon \Omega} |\mathcal{L}_{rw}^{n,\varepsilon} u(x) - \rho^{-2} \operatorname{div}(\rho^2 \nabla u)(x)| \leq C \|u\|_{C^3(B(x,\varepsilon))} (\lambda + \varepsilon)$$

holds for all $u \in C^3(\overline{\Omega})$ has probability at least $1 - C n \exp(-c n \varepsilon^{d+2} \lambda^2)$ for constants $C, c > 0$ and any $0 < \lambda \leq 1$. This is a uniform (over functions u) version of pointwise consistency for the graph

Laplacian. The weaker nonuniform version dates back to results in [33, 34]. The proof follows by combining (2.12) with Lemma 2.4 below. \square

Lemma 2.4. *There exists $C_1, C_2, C_3 > 0$ such that for any $\varepsilon > 0$ with $n\varepsilon^d \geq 1$, the event that*

$$(2.13) \quad \max_{x \in \mathcal{X}_n \setminus \partial_\varepsilon \Omega} |\mathcal{L}_\infty^{n,\varepsilon} u(x) - \Delta_\infty u(x)| \leq C_1 \left(\|u\|_{C^2(B(x,\varepsilon))}^2 |\nabla u(x)|^{-1} \theta^{-1} \varepsilon + \|u\|_{C^3(B(x,\varepsilon))} (\lambda + \varepsilon) \right)$$

holds for all $u \in C^3(\overline{\Omega})$ has probability at least $1 - C_2 n \exp(-C_3 n \varepsilon^{3d/2} \lambda^{d/2})$.

Before proving Lemma 2.4, we make a few remarks.

Remark 2.5. In fact, pointwise consistency of the graph ∞ -Laplacian (Lemma 2.4) requires $\nabla u(x) \neq 0$, since the ∞ -Laplacian is discontinuous (as a function of ∇u) at $\nabla u(x) = 0$. We interpret the right hand side of (2.13) to be ∞ if $\nabla u(x) = 0$. The viscosity solution framework for proving discrete to continuum convergence does not require consistency at points where $\nabla u(x) = 0$, since the viscosity sub- and supersolution conditions are not required to hold at such points (see, e.g., [8]).

Remark 2.6. We note that the continuum operator Δ_p forgets about the distribution ρ as $p \rightarrow \infty$. This was first observed in [21], and proved rigorously in [8, 50]. On the other hand, for semi-supervised learning problems with very few labels, the theory in [8, 21, 50] suggests one should use $d < p < p + \delta$ for some small $\delta > 0$, so that the algorithm is well-posed with very few labels, and still has maximal dependence on the data distribution ρ , which is essential for semi-supervised learning. We show in Section 2.4 that this observation is completely dependent on the ε -ball graph construction, and in particular, the story is very much different on k -NN graphs.

We now turn to the proof of Lemma 2.4. By [8, Theorem 6], we know that

$$(2.14) \quad \lim_{\substack{n \rightarrow \infty \\ \varepsilon_n \rightarrow 0}} \mathcal{L}_\infty^{n,\varepsilon_n} u(x) = \Delta_\infty u(x)$$

holds for all $u \in C^3(\overline{\Omega})$ and $x \in \Omega$ with $\nabla u(x) \neq 0$ with probability one, provided $\varepsilon_n \rightarrow 0$ so that

$$(2.15) \quad \lim_{n \rightarrow \infty} \frac{n \varepsilon_n^{3d/2}}{\log n} = \infty.$$

Lemma 2.4 strengthens this to pointwise consistency with a convergence rate. For this, we need some additional notation. We define

$$(2.16) \quad \delta_n = \sup_{x \in \Omega} \text{dist}(x, \mathcal{X}_n).$$

We need a couple of preliminary results before proving Lemma 2.4. The first concerns the approximation of the maximum value of a function over Ω with the maximum over the point cloud \mathcal{X}_n .

Proposition 2.7. *Let $\psi \in C^2(\overline{\Omega})$ and assume ψ attains its maximum value over $\overline{\Omega}$ at some $x_0 \in \Omega_{\delta_n}$. Then we have*

$$(2.17) \quad \left| \max_{x \in \Omega} \psi(x) - \max_{x \in \mathcal{X}_n} \psi(x) \right| \leq \|\nabla^2 \psi\|_{L^\infty(\Omega)} \delta_n^2.$$

Proof. Since $\nabla \psi(x_0) = 0$ and ψ is C^2 , for any $x \in \mathcal{X}_n$ we have

$$\psi(x_0) - \psi(x) \leq \|\nabla^2 \psi\|_{L^\infty(\Omega)} |x_0 - x|^2,$$

and therefore

$$\|\nabla^2 \psi\|_{L^\infty(\Omega)} \delta_n^2 \geq \|\nabla^2 \psi\|_{L^\infty(\Omega)} \min_{x \in \mathcal{X}_n} |x_0 - x|^2 \geq \psi(x_0) - \max_{x \in \mathcal{X}_n} \psi(x). \quad \square$$

The second preliminary result we quote directly from [10, Proposition 4.2].

Proposition 2.8. *For any $p \in \mathbb{R}^d$ with $p \neq 0$ and $A \in \mathbb{R}^{d \times d}$, we have*

$$\left| \max_{|z|=r} \left\{ p \cdot z + \frac{1}{2} z^T A z \right\} - r|p| - \frac{1}{2} r^2 |p|^{-2} p^T A p \right| \leq 2r^3 \|A\|^2 |p|^{-1}.$$

We note that we prove a generalization of Proposition 2.8 in Section 2.4 when studying symmetrized k -NN graphs (see Proposition 2.20).

We now give the proof of Lemma 2.4.

Proof of Lemma 2.4. First, we claim that

$$(2.18) \quad r_0^2 \eta(r_0) \mathcal{L}_\infty^{n,\varepsilon} u(x) = B^+(x) - B^-(x) + O(\|u\|_{C^3(B(x,\varepsilon))} (\delta_n^2 \varepsilon^{-3} + \varepsilon)),$$

where

$$B^\pm(x) = \max_{0 \leq r \leq 1} \left\{ \frac{1}{\varepsilon} r \eta(r) |\nabla u(x)| \pm \frac{1}{2} r^2 \eta(r) \Delta_\infty u(x) \right\}.$$

To see this, consider the max term in $\mathcal{L}_\infty^{n,\varepsilon} u(x)$,

$$M := \frac{1}{\varepsilon^2} \min_{y \in \mathcal{X}_n} \{ \eta_\varepsilon(|x - y|) (u(y) - u(x)) \}.$$

We use Proposition 2.7 to obtain

$$\begin{aligned} M &= \frac{1}{\varepsilon^2} \max_{y \in B(x,\varepsilon)} \{ \eta_\varepsilon(|x - y|) (u(y) - u(x)) \} + O(\delta_n^2 \varepsilon^{-3}) \\ &= \frac{1}{\varepsilon^2} \max_{z \in B(0,1)} \{ \eta(|z|) (u(x + \varepsilon z) - u(x)) \} + O(\delta_n^2 \varepsilon^{-3}) \\ &= \frac{1}{\varepsilon^2} \max_{z \in B(0,1)} \left\{ \eta(|z|) \left(\varepsilon \nabla u(x) \cdot z + \frac{\varepsilon^2}{2} z^T \nabla^2 u(x) z \right) \right\} + O(\delta_n^2 \varepsilon^{-3} + \varepsilon) \\ &= \max_{0 \leq r \leq 1} \left\{ \eta(r) \max_{|z|=r} \left\{ \frac{1}{\varepsilon} \nabla u(x) \cdot z + \frac{1}{2} z^T \nabla^2 u(x) z \right\} \right\} + O(\delta_n^2 \varepsilon^{-3} + \varepsilon). \end{aligned}$$

Applying Proposition 2.8 we have

$$\begin{aligned} M &= \max_{0 \leq r \leq 1} \left\{ \frac{1}{\varepsilon} r \eta(r) |\nabla u(x)| + \frac{1}{2} r^2 \eta(r) |\nabla u(x)|^{-2} \nabla u(x)^T \nabla^2 u(x) \nabla u(x) \right\} + O(\delta_n^2 \varepsilon^{-3} + \varepsilon) \\ &= B^+(x) + O(\delta_n^2 \varepsilon^{-3} + \varepsilon), \end{aligned}$$

since $\Delta_\infty u(x) = |\nabla u(x)|^{-2} \nabla u(x)^T \nabla^2 u(x) \nabla u(x)$. We can apply a similar argument to the min term from $\mathcal{L}_\infty^{n,\varepsilon} u(x)$, and this establishes the claim (2.18).

Now, let $r^\pm \in [0, 1]$ such that

$$B^\pm(x) = \frac{1}{\varepsilon} r^\pm \eta(r^\pm) |\nabla u(x)| \pm \frac{1}{2} (r^\pm)^2 \eta(r^\pm) \Delta_\infty u(x).$$

By (2.4) we have

$$\begin{aligned} \frac{1}{\varepsilon} r_0 \eta(r_0) |\nabla u(x)| \pm \frac{1}{2} r_0^2 \eta(r_0) \Delta_\infty u(x) &\leq B^\pm(x) \\ &= \frac{1}{\varepsilon} r^\pm \eta(r^\pm) |\nabla u(x)| \pm \frac{1}{2} (r^\pm)^2 \eta(r^\pm) \Delta_\infty u(x) \\ &\leq \frac{1}{\varepsilon} |\nabla u(x)| (r_0 \eta(r_0) - \theta(r^\pm - r_0)^2) \pm \frac{1}{2} (r^\pm)^2 \eta(r^\pm) \Delta_\infty u(x). \end{aligned}$$

Therefore

$$\frac{\theta}{\varepsilon} (r^\pm - r_0)^2 |\nabla u(x)| \leq \frac{1}{2} |r_0^2 \eta(r_0) - (r^\pm)^2 \eta(r^\pm)| |\Delta_\infty u(x)|.$$

Since $r \mapsto r^2\eta(r)$ is Lipschitz continuous we have

$$\frac{\theta}{\varepsilon}(r^\pm - r_0)^2|\nabla u(x)| \leq C|\Delta_\infty u(x)||r^\pm - r_0|,$$

and we deduce that

$$|r^\pm - r_0| \leq C\|u\|_{C^2(B(x,\varepsilon))}\theta^{-1}|\nabla u(x)|^{-1}\varepsilon.$$

Since

$$B^-(x) \geq \frac{1}{\varepsilon}r^+\eta(r^+)|\nabla u(x)| - \frac{1}{2}(r^+)^2\eta(r^+)\Delta_\infty u(x)$$

we have

$$B^+(x) - B^-(x) \leq (r^+)^2\eta(r^+)\Delta_\infty u(x) \leq r_0^2\eta(r_0)\Delta_\infty u(x) + C\|u\|_{C^2(B(x,\varepsilon))}^2\theta^{-1}|\nabla u(x)|^{-1}\varepsilon.$$

Similarly, since

$$B^+(x) \geq \frac{1}{\varepsilon}r^-\eta(r^-)|\nabla u(x)| + \frac{1}{2}(r^-)^2\eta(r^-)\Delta_\infty u(x)$$

we have

$$B^+(x) - B^-(x) \geq (r^-)^2\eta(r^-)\Delta_\infty u(x) \geq r_0^2\eta(r_0)\Delta_\infty u(x) - C\|u\|_{C^2(B(x,\varepsilon))}^2\theta^{-1}|\nabla u(x)|^{-1}\varepsilon.$$

Combining this with (2.18) and Proposition 2.9 below completes the proof. \square

Proposition 2.9. *There exists $C_1, C_2 > 0$ such that for every $t > 0$ with $nt^d \geq 1$ we have $\mathbb{P}(\delta_n > t) \leq C_1 n \exp(-C_2 \beta n t^d)$.*

Proof. Let $t > 0$ with $nt^d \geq 1$. Since Ω has a smooth (and hence Lipschitz) boundary, for any $h > 0$ we can find a covering of Ω by balls $B(x_1, h), B(x_2, h), \dots, B(x_M, h)$ so that $|B(x_i, h) \cap \Omega| \geq C_1 h^d$ and $M \leq C_2 h^{-d}$ for some universal constants $C_1, C_2 > 0$. Since $\rho \geq \beta > 0$, the probability that $B(x_i, h) \cap \Omega \cap \mathcal{X}_n$ is empty is bounded by $(1 - C_1 \beta h^d)^n \leq \exp(-C_1 \beta n h^d)$. Hence, the event that $B(x_i, h) \cap \Omega \cap \mathcal{X}_n$ has at least one point for all $i = 1, \dots, M$ has probability at least $1 - C_2 h^{-d} \exp(-C_1 \beta n h^d)$. Since the balls $B(x_i, h)$ cover Ω , for each $x \in \Omega$ there exists x_i such that $|x - x_i| \leq h$. If $B(x_i, h) \cap \Omega \cap \mathcal{X}_n$ is nonempty, then there exists $y \in \mathcal{X}_n$ such that $|x - y| \leq 2h$. Thus, if $\delta_n > 2h$ then at least one of $B(x_i, h) \cap \Omega \cap \mathcal{X}_n$ is empty. Therefore

$$\mathbb{P}(\delta_n > 2h) \leq C_2 h^{-d} \exp(-C_1 \beta n h^d).$$

The proof is completed by setting $h = t/2$ and recalling $h^{-d} = 2^d t^{-d} \leq 2^d n$. \square

2.3. Nonsymmetric k-nearest neighbor graphs. We now consider the simplest k -nearest neighbor graph. Let

$$(2.19) \quad N_\varepsilon(x) = \sum_{y \in \mathcal{X}_n} \mathbb{1}_{B(x,\varepsilon)}(y)$$

be the number of samples in an ε -neighborhood of $x \in \Omega$. Here, $B(x, \varepsilon)$ is the closed ball of radius $\varepsilon > 0$ centered about x . For $k \leq n$ we define

$$(2.20) \quad \varepsilon_k(x) = \min\{\varepsilon > 0 : N_\varepsilon(x) \geq k\}.$$

The value of $\varepsilon_k(x)$ for $x \in \mathcal{X}_n$ is the distance from x to its k^{th} nearest neighbor in \mathcal{X}_n . The *nonsymmetric* k -nearest neighbor random walk graph Laplacian is then given by

$$(2.21) \quad \mathcal{L}_{a,rw}^{n,k} u(x) = \frac{2}{\sigma_\eta d^{n,\varepsilon_k(x)}(x)} \left(\frac{n\alpha(d)}{k} \right)^{2/d} \sum_{y \in \mathcal{X}_n} \eta_{\varepsilon_k(x)}(|x - y|)(u(y) - u(x)).$$

We note that since $\varepsilon_k(x)$ satisfies, on average, $\alpha(d)n\varepsilon_k(x)^d \rho(x) \approx k$, the normalization in (2.21) is equivalent, up to the factor $\rho(x)$, with the normalization in the case of the ε -ball graph given in (2.7).

The graph ∞ -Laplacian is defined by

$$(2.22) \quad \mathcal{L}_{a,\infty}^{n,k} u(x) = \frac{1}{r_0^2 \eta(r_0)} \left(\frac{n\alpha(d)}{k} \right)^{2/d} \left(\min_{y \in \mathcal{X}_n} \{ \eta_{\varepsilon_k(x)}(|x-y|)(u(y) - u(x)) \} \right. \\ \left. + \max_{y \in \mathcal{X}_n} \{ \eta_{\varepsilon_k(x)}(|x-y|)(u(y) - u(x)) \} \right).$$

We now define the game-theoretic k -nearest neighbor graph p -Laplacian to be

$$(2.23) \quad \mathcal{L}_{a,p}^{n,k} u(x) = \frac{1}{p} \mathcal{L}_{a,rw}^{n,k} u(x) + \left(1 - \frac{2}{p} \right) \mathcal{L}_{a,\infty}^{n,k} u(x).$$

The main result in this section is consistency for the nonsymmetric k -nearest neighbor graph Laplacian.

Theorem 2.10 (Consistency on nonsymmetric k -NN graphs). *There exists $C_1, C_2, C_3, c_1, c_2 > 0$ such that for $k \leq c_1 n$ and $0 < \lambda \leq 1/4$, the event that*

$$|\mathcal{L}_{a,p}^{n,k} u(x) - \rho(x)^{-2/d} \Delta_p u(x)| \leq C_1 \left(\|u\|_{C^2(B_k)}^2 |\nabla u(x)|^{-1} \theta^{-1} \left(\frac{k}{n} \right)^{1/d} + \|u\|_{C^3(B_k)} \left(\lambda + \left(\frac{k}{n} \right)^{1/d} \right) \right),$$

holds for all $u \in C^3(\bar{\Omega})$ and $x \in \Omega_{c_2(k/n)^{1/d}} \cap \mathcal{X}_n$, where $B_k = B(x, c_2(k/n)^{1/d})$, has probability at least

$$1 - C_2 n^3 \exp \left(-C_3 \left(\frac{k}{n} \right)^{1/2} k \lambda^{d/2} \right) - C_2 n^3 \exp \left(-C_3 \left(\frac{k}{n} \right)^{2/d} k \lambda^2 \right).$$

Remark 2.11. Notice that, up to the factor $\rho^{-2/d}$, the nonsymmetric k -NN graph p -Laplacian is consistent with the same operator Δ_p as the ε -ball graph Laplacian. This is due to the ability to treat the k -NN Laplacian as an ε -ball graph Laplacian with spatially varying ε , since there is no symmetrization to be concerned with. The same observation as in Remark 2.6 holds here; that is, the continuum operator Δ_p forgets the distribution ρ as $p \rightarrow \infty$. We also remark that the rates in Theorems 2.10 and 2.3 are essentially the same, since the quantity $\left(\frac{k}{n} \right)^{1/d}$ is the average distance to the k^{th} nearest neighbor (obtained by setting $n\varepsilon^d \sim k$), and plays the same role as ε in Theorem 2.3.

We now turn to the proof of Theorem 2.10. To connect $\mathcal{L}_{a,p}^{n,k}$ to $\mathcal{L}_p^{n,\varepsilon}$, we define

$$(2.24) \quad s_k(x) = \left(\frac{k}{\alpha(d)\rho(x)n} \right)^{1/d}.$$

The radius $s_k(x)$ satisfies $\alpha(d)ns_k(x)^d \rho(x) = k$, so that the ball $B(x, s_k(x))$ contains on average k points from \mathcal{X}_n . Then by algebraic manipulations we have

$$(2.25) \quad \mathcal{L}_{a,p}^{n,k} u(x) = \frac{\varepsilon_k(x)^2}{s_k(x)^2 \rho(x)^{2/d}} \mathcal{L}_p^{n,\varepsilon_k(x)} u(x).$$

This identity allows us to view the nonsymmetric k -nearest neighbor graph Laplacian $\mathcal{L}_{a,p}^{n,k}$ as an ε -ball graph Laplacian for spatially varying $\varepsilon = \varepsilon(x)$, and consistency will follow by applying Theorem 2.3 and a covering argument.

Before proving consistency, we recall some facts about $N_\varepsilon(x)$ and $\varepsilon_k(x)$. These facts can be found in [14], in the manifold setting, but we include the proofs here for the reader's convenience, as they are simpler in the Euclidean setting.

Proposition 2.12. *For any $0 < t \leq 1$ and $x \in \Omega_\varepsilon$ we have*

$$(2.26) \quad \mathbb{P}(|N_\varepsilon(x) - np(x, \varepsilon)| \geq np(x, \varepsilon)t) \leq 2 \exp \left(-\frac{3}{8} \alpha(d) \beta n \varepsilon^d t^2 \right),$$

where $p(x, \varepsilon) = \int_{B(x, \varepsilon)} \rho(y) dy$.

Proof. Note that $N_\varepsilon(x)$ is the sum of Bernoulli zero/one random variables with parameter $p(x, \varepsilon) = \int_{B(x, \varepsilon)} \rho(x) dx$. By the Chernoff bounds we have

$$\mathbb{P}(|N_\varepsilon(x) - np(x, \varepsilon)| \geq np(x, \varepsilon)t) \leq 2 \exp\left(-\frac{3}{8}np(x, \varepsilon)t^2\right)$$

for any $0 < t \leq 1$. The proof is completed by noting that $p(x, \varepsilon) \geq \alpha(d)\beta\varepsilon^d$. \square

Lemma 2.13. *There exists $C_1, C_2, c_1, c_2 > 0$ such that for any $0 < t \leq 1/4$, $k \leq c_1 n$, and $x \in \Omega_{c_2(k/n)^{1/d}}$, we have*

$$(2.27) \quad \mathbb{P}\left(\left|\frac{\varepsilon_k(x)}{s_k(x)} - 1\right| \geq t + C_1 \left(\frac{k}{n}\right)^{2/d}\right) \leq 2 \exp(-C_2 kt^2).$$

Proof. For any $\varepsilon > 0$ we have

$$\mathbb{P}(\varepsilon_k(x) \geq \varepsilon) \leq \mathbb{P}(N_\varepsilon(x) \leq k).$$

Choose $\varepsilon > 0$ so that $np(x, \varepsilon) = k(1+t)$ for $t > 0$. Then we have

$$\begin{aligned} \mathbb{P}(\varepsilon_k(x) \geq \varepsilon) &\leq \mathbb{P}(N_\varepsilon(x) \leq k) \\ &= \mathbb{P}(N_\varepsilon(x) - np(x, \varepsilon) \leq -kt) \\ &\leq \mathbb{P}(N_\varepsilon(x) - np(x, \varepsilon) \leq -\frac{1}{2}np(x, \varepsilon)t), \end{aligned}$$

provided $t \leq 1$. By Proposition 2.12 we have

$$\mathbb{P}(\varepsilon_k(x) \geq \varepsilon) \leq 2 \exp\left(-\frac{3}{32}\alpha(d)\beta n \varepsilon^d t^2\right),$$

for all $0 < t \leq 1$. We now note that

$$2 \geq 1+t = \frac{n}{k}p(x, \varepsilon) \geq \frac{n}{k}\alpha(d)\beta\varepsilon^d,$$

and

$$1 \leq 1+t = \frac{n}{k}p(x, \varepsilon) \leq \frac{n}{k}\alpha(d)\beta^{-1}\varepsilon^d.$$

Therefore

$$(2.28) \quad \left(\frac{\beta}{\alpha(d)}\right) \frac{k}{n} \leq \varepsilon^d \leq \left(\frac{2}{\alpha(d)\beta}\right) \frac{k}{n}.$$

For a more refined estimate, we have

$$(2.29) \quad p(x, \varepsilon) = \int_{B(x, \varepsilon)} \rho(y) dy = \alpha(d)\rho(x)\varepsilon^d + O(\varepsilon^{d+2}).$$

Therefore

$$k(1+t) = np(x, \varepsilon) \geq \alpha(d)\rho(x)n\varepsilon^d(1 - C\varepsilon^2),$$

and so

$$\varepsilon^d \leq \frac{k(1+t)}{\alpha(d)\rho(x)n(1 - C\varepsilon^2)} = \frac{s_k(x)^d(1+t)}{1 - C\varepsilon^2} \leq s_k(x)^d(1+t)(1 + 4C\varepsilon^2),$$

provided $C\varepsilon^2 \leq \frac{1}{2}$. Hence, due to (2.28), there exists $c > 0$ such that $k \leq cn$ implies $C\varepsilon^2 \leq \frac{1}{2}$ and so

$$\varepsilon^d \leq s_k(x)^d(1+t + 8C\varepsilon^2).$$

It follows that there exists $C_1, C_2 > 0$ such that

$$\mathbb{P}\left(\frac{\varepsilon_k(x)}{s_k(x)} \geq 1+t + C_1 \left(\frac{k}{n}\right)^{2/d}\right) \leq 2 \exp(-C_2 kt^2),$$

for any $0 < t \leq 1$. The proof for the estimate in the other direction is similar, but requires a restriction $t \leq 1/4$. \square

Proof of Theorem 2.10. Fix $0 < a < b \leq 1$ with $na^d \geq 1$. Let $C_1, C_2, C_3 > 0$ so that Theorem 2.3 holds and let $x \in \Omega_b$. For $0 < \delta \leq a/2$ and $0 \leq \lambda \leq 1$, let $A_{\delta,\lambda}$ denote the event that

$$|\mathcal{L}_p^{n,\varepsilon}u(x) - \Delta_p u(x)| \leq C_1 \left(\|u\|_{C^2(B(x,\varepsilon))}^2 |\nabla u(x)|^{-1} \theta^{-1} \varepsilon + \|u\|_{C^3(B(x,\varepsilon))} (\lambda + \varepsilon) \right)$$

holds for all $u \in C^3(\bar{\Omega})$ and all $\varepsilon \in [a - \delta, b + \delta] \cap \mathbb{Z}_\delta$, where $\mathbb{Z}_\delta = \delta\mathbb{Z}$. By Theorem 2.3 and a union bound we have

$$\mathbb{P}(A_{\delta,\lambda}) \geq 1 - C_2 n \delta^{-1} \exp\left(-C_3 n a^{3d/2} \lambda^{d/2}\right) - C_2 n \delta^{-1} \exp\left(-C_3 n a^{d+2} \lambda^2\right).$$

For $\varepsilon > 0$, let us denote by $\lfloor \varepsilon \rfloor_\delta$ the largest number belonging to the set $\mathbb{Z}_\delta := \delta\mathbb{Z}$ that is less than or equal to ε , and write $\lceil \varepsilon \rceil_\delta = \lfloor \varepsilon \rfloor_\delta + \delta$. Noting that

$$|\partial_\varepsilon \eta_\varepsilon(|x - y|)| = |x - y| \varepsilon^{-2} \eta_\varepsilon(|x - y|) \leq \frac{C}{\varepsilon},$$

provided $|x - y| \leq \varepsilon$, we can compute that

$$|\partial_\varepsilon \mathcal{L}_p^{n,\varepsilon}u(x)| \leq \frac{C \|u\|_{C^1(B(x,\varepsilon))}}{\varepsilon^2}$$

holds for all $u \in C^3(\bar{\Omega})$ and $\varepsilon \in [a - \delta, b + \delta]$ with probability at least $1 - 2 \exp(-cna^d)$ for $c > 0$, due to Proposition 2.12. Therefore, when $A_{\delta,\lambda}$ occurs and $\varepsilon_k(x) \in [a, b]$ we have

$$|\mathcal{L}_p^{n,\varepsilon_k(x)}u(x) - \Delta_p u(x)| \leq C_1 \left(\|u\|_{C^2(B(x,b+\delta))}^2 |\nabla u(x)|^{-1} \theta^{-1} \varepsilon_k(x) + \|u\|_{C^3(B(x,b+\delta))} (\lambda + \varepsilon_k(x)) \right),$$

provided we choose $\delta = a^3$. Using (2.25) we can write

$$|\rho(x)^{2/d} \mathcal{L}_{a,p}^{n,k}u(x) - \Delta_p u(x)| \leq \frac{\varepsilon_k(x)^2}{s_k(x)^2} |\mathcal{L}_p^{n,\varepsilon_k(x)}u(x) - \Delta_p u(x)| + \left| \frac{\varepsilon_k(x)^2}{s_k(x)^2} - 1 \right| |\Delta_p u(x)|.$$

We now invoke Lemma 2.13, set $a = c_1(k/n)^{1/d}$ and $b = c_2(k/n)^{1/d}$ for constants $0 < c_1 < c_2$, and union bound over $x \in \Omega_{c_2(k/n)^{1/d}} \cap \mathcal{X}_n$. The proof is completed by noting that $\delta^{-1} \leq Cn^{3/d} \leq Cn^2$. \square

2.4. Symmetric k -nearest neighbor graphs. We now consider symmetrized k -nearest neighbor graphs. Define

$$(2.30) \quad \varepsilon_k(x, y) = \max\{\varepsilon_k(x), \varepsilon_k(y)\}.$$

The symmetric k -nearest neighbor random walk graph Laplacian is then defined by

$$(2.31) \quad \mathcal{L}_{s,rw}^{n,k}u(x) = \frac{2}{\sigma_\eta d_s^{m,k}(x)} \left(\frac{n\alpha(d)}{k} \right)^{2/d} \sum_{y \in \mathcal{X}_n} \eta_{\varepsilon_k(x,y)}(|x - y|)(u(y) - u(x)),$$

where the degree $d_s^{m,k}(x)$ is defined by

$$(2.32) \quad d_s^{m,k}(x) = \sum_{y \in \mathcal{X}_n} \eta_{\varepsilon_k(x,y)}(|x - y|).$$

We can also define a symmetrized graph using $\varepsilon_k(x, y) = \min\{\varepsilon_k(x), \varepsilon_k(y)\}$, which would produce the *mutual* k -NN graph, or any other suitable symmetric combination of $\varepsilon_k(x)$ and $\varepsilon_k(y)$. We expect the consistency results in this section to hold, with minor modifications, for many other types of symmetrization.

The graph ∞ -Laplacian on the symmetric k -nearest neighbor graph is defined by

$$(2.33) \quad \mathcal{L}_{s,\infty}^{n,k} u(x) = \frac{1}{r_0^2 \eta(r_0)} \left(\frac{n\alpha(d)}{k} \right)^{2/d} \left(\min_{y \in \mathcal{X}_n} \{ \eta_{\varepsilon_k(x,y)} (|x-y|)(u(y) - u(x)) \} \right. \\ \left. + \max_{y \in \mathcal{X}_n} \{ \eta_{\varepsilon_k(x,y)} (|x-y|)(u(y) - u(x)) \} \right),$$

and the graph p -Laplacian is defined by

$$(2.34) \quad \mathcal{L}_{s,p}^{n,k} u(x) = \frac{1}{p} \mathcal{L}_{s,rw}^{n,k} u(x) + \left(1 - \frac{2}{p} \right) \mathcal{L}_{s,\infty}^{n,k} u(x).$$

The graph p -Laplacian on a symmetric k -NN graph is consistent in the continuum with the operator

$$(2.35) \quad \Delta_p^s u = \rho^{-2/d} \left[\frac{1}{p} \rho^{-(1+2/d)} \operatorname{div} \left(\rho^{1-2/d} \nabla u \right) + \left(1 - \frac{2}{p} \right) \left(\Delta_\infty u - \frac{1}{d} \nabla \log \rho \cdot \nabla u \right) \right],$$

as is shown in the following theorem.

Theorem 2.14 (Consistency on symmetric k -NN graphs). *There exists $C_1, C_2, C_3, c_1, c_2 > 0$ such that for $0 < t \leq 1/4$ and $k \leq c_1 n$, the event that*

$$\left| \mathcal{L}_{s,p}^{n,k} u(x) - \Delta_p^s u(x) \right| \leq C_1 (1 + |\nabla u(x)|^{-1} \|u\|_{C^2(B(x,\varepsilon))}^2 + \|u\|_{C^3(B(x,\varepsilon))}) (\delta_n \varepsilon^{-2} + t + \varepsilon)$$

holds for all $x \in \mathcal{X}_n \cap \Omega_\varepsilon$, where $\varepsilon = c_2(k/n)^{1/d}$, and for all $u \in C^3(\overline{\Omega})$, has probability at least $1 - C_3 n \exp \left(-C_2 k \left(\frac{k}{n} \right)^{2/d} t^2 \right)$.

Proof. The proof simply combines Lemma 2.18 and Lemma 2.21, both of which are proved below. \square

Remark 2.15. It is important to point out the additional drift term $-\frac{1}{d} \nabla \log \rho \cdot \nabla u$ in Δ_p^s , compared to Δ_p . This term is due to the symmetrization of the k -nearest neighbor relation, and is a result of the Taylor expansion

$$\left(\frac{\rho(y)}{\rho(x)} \right)^{\frac{1}{d}} = 1 + \frac{1}{d} \nabla \log \rho(x) \cdot (y - x) + O(|x - y|^2),$$

which makes an appearance in the proof of Lemma 2.16 below. Since this term persists in the limit as $p \rightarrow \infty$, this shows that p -Laplacian learning for large p , and in particular, Lipschitz learning ($p = \infty$) are sensitive to the distribution ρ of the unlabeled data *provided* the graph is constructed as a symmetrized k -nearest neighbor graph, as is often done in practice. This should be contrasted with the case of ε -ball graphs and nonsymmetric k -NN graphs, where the continuum operator Δ_p *forgets* the distribution ρ as $p \rightarrow \infty$.

We first record a result about $\varepsilon_k(x, y)$, which was proved in [14], but not explicitly stated as a lemma. We define $a_- = \min\{a, 0\}$.

Lemma 2.16. *There exists $C_1, C_2, c_1, c_2 > 0$ such that for any $0 < t \leq 1/4$, $k \leq c_1 n$ and $x \in \Omega_\varepsilon$, the event that*

$$\left| \frac{\varepsilon_k(x, y)}{s_k(x)} \left(1 + \frac{1}{d} [\nabla \log \rho(x) \cdot (y - x)]_- \right) - 1 \right| \leq t + C_1 \varepsilon^2,$$

holds for all $y \in \mathcal{X}_n \cap B(x, \varepsilon)$, with $\varepsilon = c_2(k/n)^{1/d}$, has probability at least $1 - 2n \exp(-C_2 k t^2)$.

Proof. Let $x \in \Omega_\varepsilon$. By Lemma 2.13 and a union bound, there exists $C_1, C_2, c > 0$ such that for any $0 < t \leq 1/4$ and $k \leq cn$ we have

$$(2.36) \quad \max_{y \in \{x\} \cup \mathcal{X}_n \cap \Omega_\varepsilon} \left| \frac{\varepsilon_k(y)}{s_k(y)} - 1 \right| \leq t + C_1 \varepsilon^2$$

holds with probability at least $1 - 2n \exp(-C_2 k t^2)$, where $\varepsilon = c_2(k/n)^{1/d}$. We assume for the rest of the proof that (2.36) holds. For $y \in \mathcal{X}_n \cap B(x, \varepsilon)$ it follows from (2.36) that

$$\left| \frac{\varepsilon_k(x, y)}{s_k(x, y)} - 1 \right| \leq t + C_1 \varepsilon^2,$$

where $s_k(x, y) = \max\{s_k(x), s_k(y)\}$. Since $|x - y| \leq \varepsilon$, we compute

$$\begin{aligned} s_k(x, y) &= s_k(x) \max \left\{ 1, \frac{s_k(y)}{s_k(x)} \right\} \\ &= \frac{s_k(x)}{\min \left\{ 1, \frac{\rho(y)^{1/d}}{\rho(x)^{1/d}} \right\}} \\ &= \frac{s_k(x)}{\min \left\{ 1, 1 + \frac{1}{d} \nabla \log \rho(x) \cdot (y - x) + O(\varepsilon^2) \right\}} \\ &= \frac{s_k(x)}{1 + \frac{1}{d} [\nabla \log \rho(x) \cdot (y - x)]_- + O(\varepsilon^2)}. \end{aligned}$$

Therefore

$$\left| \frac{\varepsilon_k(x, y)}{s_k(x)} \left(1 + \frac{1}{d} [\nabla \log \rho(x) \cdot (y - x)]_- \right) - 1 \right| \leq t + C_1 \varepsilon^2,$$

which completes the proof. \square

Lemma 2.17. *There exists $C_1, C_2, C_3, c > 0$ such that for each $x \in \Omega_\varepsilon$ with $\varepsilon = c(k/n)^{1/d}$*

$$(2.37) \quad \left| \frac{\alpha(d)}{k} d_s^{n,k}(x) - 1 \right| \leq C_1(t + \varepsilon)$$

holds with probability at least $1 - C_2 \exp(-C_3 k t^2)$ for any $0 < t \leq 1/4$.

Proof. By Lemma 2.16 there exists $c > 0$ so that for every $x \in \Omega_\varepsilon$, where $\varepsilon = c(k/n)^{1/d}$, we have

$$(2.38) \quad \left| \frac{s_k(x)}{\varepsilon_k(x, y)} - 1 \right| \leq C(t + \varepsilon)$$

for all $y \in \mathcal{X}_n \cap B(x, \varepsilon)$ with probability at least $1 - 2n \exp(-C_2 k t^2)$. Notice the drift term involving $\nabla \log \rho(x) \cdot (y - x)_-$ from Lemma 2.16 is absorbed into the $O(\varepsilon)$ error term on the right hand side, since $|x - y| \leq \varepsilon$. Assuming (2.38) holds we have

$$\begin{aligned} d_s^{n,k}(x) &= \sum_{y \in \mathcal{X}_n} \eta \left(\frac{|x - y|}{\varepsilon_k(x, y)} \right) \\ &= \sum_{y \in \mathcal{X}_n} \eta \left(\frac{|x - y|}{s_k(x)} (1 + O(t + \varepsilon)) \right) \\ &= \sum_{y \in \mathcal{X}_n} \eta \left(\frac{|x - y|}{s_k(x)} \right) + O(N_{C s_k(x)}(x)(t + \varepsilon)), \end{aligned}$$

for $C > 0$, provided $t \leq 1$ and $\varepsilon \leq 1$. By Proposition 2.12 we have $N_{C s_k(x)}(x) \leq \bar{C}k$ with probability at least $1 - \exp(-ck)$. Thus,

$$d_s^{n,k}(x) = d^{n, s_k(x)}(x) + O(k(t + \varepsilon)).$$

By the Bernstein inequality (see, e.g., [8, Theorem 5]) we have that

$$d^{n, s_k(x)}(x) = \rho(x) n s_k(x)^d + O(n s_k(x)^d t) = \alpha(d)^{-1} k + O(kt),$$

with probability at least $1 - 2 \exp(-ckt^2)$. This completes the proof. \square

We now prove pointwise consistency for the random walk graph Laplacian $\mathcal{L}_{s,rw}^{n,k}$ on a symmetric k -nearest neighbor graph. The analogous result for the unnormalized graph Laplacian was established in [14].

Lemma 2.18. *There exists $C_1, C_2, C_3, c_1, c_2 > 0$ such that for $0 < t \leq 1/4$ and $k \leq c_1 n$, the event that*

$$\left| \mathcal{L}_{s,rw}^{n,k} u(x) - \rho(x)^{-1} \operatorname{div} \left(\rho^{1-2/d} \nabla u \right) (x) \right| \leq C_1 (1 + \|u\|_{C^3(B_k)}) \left(t + \left(\frac{k}{n} \right)^{1/d} \right)$$

holds for all $u \in C^3(\bar{\Omega})$ and $x \in \Omega_{c_2(k/n)^{1/d}} \cap \mathcal{X}_n$, where $B_k = B(x, c_2(k/n)^{1/d})$, has probability at least $1 - C_2 n \exp \left(-C_3 \left(\frac{k}{n} \right)^{2/d} k t^2 \right)$.

Proof. Let $u \in C^3(\bar{\Omega})$ and $x \in \Omega$. Let us define the unnormalized k -nearest neighbor graph Laplacian

$$Lu(x) = \frac{2}{\sigma_\eta n} \left(\frac{n\alpha(d)}{k} \right)^{1+2/d} \sum_{y \in \mathcal{X}_n} \eta_{\varepsilon_k(x,y)} (|x-y|) (u(y) - u(x)).$$

Then by [14, Theorem 3.6] we have that

$$\left| Lu(x) - \rho(x)^{-1} \operatorname{div} \left(\rho^{1-2/d} \nabla u \right) (x) \right| \leq C (1 + \|u\|_{C^3(B(x,\varepsilon))}) (t + \varepsilon)$$

holds for all $x \in \Omega_\varepsilon \cap \mathcal{X}_n$ with probability at least $1 - Cn \exp \left(-c \left(\frac{k}{n} \right)^{2/d} k t^2 \right)$ for $0 \leq t \leq 1/4$, $k \leq c_2 n$, and $\varepsilon = c_2(k/n)^{1/d}$ with $c_1, c_2 > 0$. Noting that

$$\mathcal{L}_{s,rw}^{n,k} u(x) = \frac{k}{\alpha(d) d_s^{n,k}(x)} Lu(x)$$

and invoking Lemma 2.17 we have

$$\left| \mathcal{L}_{s,rw}^{n,k} u(x) - \rho(x)^{-1} \operatorname{div} \left(\rho^{1-2/d} \nabla u \right) (x) \right| \leq C (1 + \|u\|_{C^3(B(x,\varepsilon))}) (t + \varepsilon)$$

holds for all $x \in \Omega_\varepsilon \cap \mathcal{X}_n$ with probability at least $1 - Cn \exp \left(-c \left(\frac{k}{n} \right)^{2/d} k t^2 \right)$.

To prove uniformity over $u \in C^3(\bar{\Omega})$, we fix $x \in \Omega_\varepsilon$ and $u \in C^3(\bar{\Omega})$ and we first Taylor expand u to find

$$u(y) = u(x) + \nabla u(x) \cdot (y-x) + \frac{1}{2} (y-x)^T \nabla^2 u(x) (y-x) + O(\|u\|_{C^3(B(x,\varepsilon))} \varepsilon^3)$$

for $|x-y| \leq \varepsilon$. Therefore

$$(2.39) \quad \mathcal{L}_{s,rw}^{n,k} u(x) = \sum_{i=1}^d u_{x_i}(x) \mathcal{L}_{s,rw}^{n,k} p_i(x) + \frac{1}{2} \sum_{i,j=1}^d u_{x_i x_j}(x) \mathcal{L}_{s,rw}^{n,k} (p_i p_j)(x) + O(\|u\|_{C^3(B(x,\varepsilon))} \varepsilon),$$

where $p_i(y) = y_i - x_i$ and the error term is controlled with probability at least $1 - C \exp(-ck)$, due to Lemma 2.17. By the argument above

$$|\mathcal{L}_{s,rw}^{n,k} p_i(x) - (1 - \frac{2}{d}) \rho(x)^{-(1+2/d)} \rho_{x_i}(x)| \leq C(t + \varepsilon),$$

and

$$|\mathcal{L}_{s,rw}^{n,k} (p_i p_j)(x) - 2\rho(x)^{-2/d} \delta_{ij}| \leq C(t + \varepsilon)$$

hold for all $x \in \Omega_\varepsilon \cap \mathcal{X}_n$ with probability at least $1 - Cn \exp\left(-c\left(\frac{k}{n}\right)^{2/d} kt^2\right)$, where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ otherwise. Substituting this into (2.39) we have

$$\begin{aligned} \mathcal{L}_{s,rw}^{n,k} u(x) &= (1 - \frac{2}{d}) \rho^{-(1+2/d)} \sum_{i=1}^d u_{x_i}(x) \rho_{x_i}(x) + \rho(x)^{-2/d} \sum_{i,j=1}^d u_{x_i x_j}(x) \delta_{ij} + O(\|u\|_{C^3(B(x,\varepsilon))} \varepsilon), \\ &= \rho(x)^{-2/d} \left((1 - \frac{2}{d}) \rho(x)^{-1} \nabla \rho(x) \cdot \nabla u(x) + \Delta u(x) \right) + O(\|u\|_{C^3(B(x,\varepsilon))} \varepsilon) \\ &= \rho(x)^{-1} \operatorname{div} \left(\rho^{1-2/d} \nabla u \right) (x) + O(\|u\|_{C^3(B(x,\varepsilon))} \varepsilon), \end{aligned}$$

which completes the proof. \square

We require the following simple proposition.

Proposition 2.19. *Let $\psi \in C^1(\bar{\Omega})$ and assume ψ attains its maximum value over $\bar{\Omega}$ at some $x_0 \in \Omega$. Then we have*

$$(2.40) \quad \left| \max_{x \in \Omega} \psi(x) - \max_{x \in \mathcal{X}_n} \psi(x) \right| \leq \|\nabla \psi\|_{L^\infty(\Omega)} \delta_n$$

Proof. For any $x \in \mathcal{X}_n$ we have

$$\psi(x_0) - \psi(x) \leq \|\nabla \psi\|_{L^\infty(\Omega)} |x_0 - x|,$$

and therefore

$$\delta_n \geq \min_{x \in \mathcal{X}_n} |x_0 - x| \geq \|\nabla \psi\|_{L^\infty(\Omega)}^{-1} (\psi(x_0) - \max_{x \in \mathcal{X}_n} \psi(x)),$$

where we recall δ_n was defined in (2.16). \square

We define

$$(2.41) \quad H_{r,\varepsilon}(v, p, X) = \max_{|z|=r} \left\{ p \cdot (z + [v \cdot z]_+ \varepsilon z) + \frac{\varepsilon}{2} z^T X z \right\}.$$

The following proposition can be viewed as an extension of Proposition 2.8, which applies when $v = 0$. It is used to evaluate the max and min in the graph ∞ -Laplacian, asymptotically after Taylor expansions, in Lemma 2.21 below.

Proposition 2.20. *For any $\varepsilon, r > 0$, $p \in \mathbb{R}^d$ with $p \neq 0$ and $X \in \mathbb{R}^{d \times d}$ we have*

$$\left| H_{r,\varepsilon}(v, p, X) - r|p| - [v \cdot p]_+ r^2 \varepsilon - \frac{r^2 \varepsilon}{2|p|^2} p^T X p \right| \leq 2|p|^{-1} (|p||v| + \|X\|)^2 r^3 \varepsilon^2.$$

Proof. Let z_r such that $|z_r| = r$ and

$$H_{r,\varepsilon}(v, p, X) = (p \cdot z_r)(1 + [v \cdot z_r]_+ \varepsilon) + \frac{\varepsilon}{2} z_r^T X z_r.$$

Set $w_r = rp/|p|$. Choosing $z = w_r$ yields

$$H_{r,\varepsilon}(v, p, X) \geq r|p|(1 + [v \cdot w_r]_+ \varepsilon) + \frac{\varepsilon}{2} w_r^T X w_r.$$

Note that for any unit vectors a, b we have

$$1 - a \cdot b = \frac{1}{2}(2 - 2a \cdot b) = \frac{1}{2}|a - b|^2.$$

Therefore

$$r|p| - p \cdot z_r = r|p| \left(1 - \frac{p}{|p|} \cdot \frac{z_r}{r} \right) = \frac{r|p|}{2} \left| \frac{p}{|p|} - \frac{z_r}{r} \right|^2 = \frac{|p|}{2r} |w_r - z_r|^2.$$

Combining this with the observations above we have

$$\begin{aligned}
\frac{|p|}{2r} |w_r - z_r|^2 &= r|p| - p \cdot z_r \\
&\leq \left((p \cdot z_r)[v \cdot z_r]_+ + \frac{1}{2} z_r^T X z_r - r|p|[v \cdot w_r]_+ - \frac{1}{2} w_r^T X w_r \right) \varepsilon \\
&\leq \left(r|p|([v \cdot z_r]_+ - [v \cdot w_r]_+) + \frac{1}{2} (z_r^T X z_r - w_r^T X w_r) \right) \varepsilon \\
&\leq (r|p||v||w_r - z_r| + r\|X\||w_r - z_r|) \varepsilon \\
&= (|p||v| + \|X\|)|w_r - z_r|r\varepsilon.
\end{aligned}$$

Therefore

$$|w_r - z_r| \leq 2(|v| + |p|^{-1}\|X\|) r^2 \varepsilon = 2|p|^{-1}(|p||v| + \|X\|) r^2 \varepsilon,$$

and so

$$\begin{aligned}
H_{r,\varepsilon}(v, p, X) - p \cdot (w_r + [v \cdot w_r]_+ \varepsilon w_r) - \frac{\varepsilon}{2} w_r^T X w_r \\
&= p \cdot (z_r + [v \cdot z_r]_+ \varepsilon z_r) + \frac{\varepsilon}{2} z_r^T X z_r - p \cdot (w_r + [v \cdot w_r]_+ \varepsilon w_r) - \frac{\varepsilon}{2} w_r^T X w_r \\
&= p \cdot z_r - p \cdot w_r + [v \cdot z_r]_+ (p \cdot z_r) \varepsilon - [v \cdot w_r]_+ (p \cdot w_r) \varepsilon + \frac{1}{2} (z_r^T X z_r - w_r^T X w_r) \varepsilon \\
&\leq p \cdot z_r - r|p| + r|p|([v \cdot z_r]_+ - [v \cdot w_r]_+) \varepsilon + \|X\||w_r - z_r|r\varepsilon \\
&\leq |p||v||w_r - z_r|r\varepsilon + \|X\||w_r - z_r|r\varepsilon \\
&= (|p||v| + \|X\|)|w_r - z_r|r\varepsilon \\
&\leq 2|p|^{-1}(|p||v| + \|X\|)^2 r^3 \varepsilon^2,
\end{aligned}$$

which completes the proof. \square

Lemma 2.21. *There exists constants $C_1, C_2, c_1, c_2 > 0$ such that for $0 < t \leq 1$ and $k \leq c_1 n$, the event that*

$$\begin{aligned}
\left| \mathcal{L}_{s,\infty}^{n,k} u(x) - \rho(x)^{-2/d} (\Delta_\infty u(x) - \frac{1}{d} \nabla \log \rho(x) \cdot \nabla u(x)) \right| \\
\leq C_1 \left| (1 + |\nabla u(x)|)^{-1} \|u\|_{C^2(B(x,\varepsilon))}^2 + \|u\|_{C^3(B(x,\varepsilon))} (\delta_n \varepsilon^{-2} + t + \varepsilon) \right|
\end{aligned}$$

holds for all $x \in \mathcal{X}_n \cap \Omega_\varepsilon$, where $\varepsilon = c_2(k/n)^{1/d}$, and for all $u \in C^3(\bar{\Omega})$, has probability at least $1 - 2n \exp\left(-C_2 k \left(\frac{k}{n}\right)^{2/d} t^2\right)$.

Proof. By Lemma 2.16 we have

$$\eta_{\varepsilon_k(x,y)}(|x - y|) = \eta\left(\frac{|x - y|}{\varepsilon_k(x,y)}\right) = \eta\left(\frac{|x - y|(1 - [v \cdot (y - x)]_+)}{s_k(x)}\right) + O(t\varepsilon + \varepsilon^2),$$

for $|x - y| \leq \varepsilon_k(x, y)$, where $v = -\frac{1}{d} \nabla \log \rho(x)$, with probability at least $1 - 2n \exp(-C_2 k \varepsilon^2 t^2)$, where $0 < t \leq 1$ and $\varepsilon = c(k/n)^{1/d}$. Therefore, by Proposition 2.19 we have

$$\begin{aligned}
&\max_{y \in \mathcal{X}_n} \left\{ \eta_{\varepsilon_k(x,y)}(|x - y|)(u(y) - u(x)) \right\} \\
&= \max_{y \in \mathcal{X}_n} \left\{ \eta\left(\frac{|x - y|(1 - [v \cdot (y - x)]_+)}{s_k(x)}\right) (u(y) - u(x)) \right\} + O(\|u\|_{C^1(B(x,\varepsilon))} \varepsilon^2(t + \varepsilon)) \\
&= \max_{y \in \Omega} \left\{ \eta\left(\frac{|x - y|(1 - [v \cdot (y - x)]_+)}{s_k(x)}\right) (u(y) - u(x)) \right\} + O((1 + \|u\|_{C^1(B(x,\varepsilon))})(\delta_n + \varepsilon^2 t + \varepsilon^3))
\end{aligned}$$

We now make a change of variables, setting

$$z = \Phi(y) := \frac{1}{s_k(x)}(y - x)(1 - [v \cdot (y - x)]_+).$$

For y sufficiently close to x , depending only on $|v|$, the mapping $y \mapsto \Phi(y)$ is invertible and

$$y = \Phi^{-1}(z) = x + s_k z + [v \cdot z]_+ s_k^2 z + O(\varepsilon^3).$$

where we write $s_k = s_k(x)$ for simplicity, and note that $s_k = O(\varepsilon)$. Therefore, using Proposition 2.20 we have

$$\begin{aligned} & \max_{y \in \mathcal{X}_n} \{ \eta_{\varepsilon_k(x,y)}(|x - y|)(u(y) - u(x)) \} \\ &= \max_{z \in B(0,1)} \{ \eta(|z|)(u(x + s_k z + [v \cdot z]_+ s_k^2 z) - u(x)) \} + O((1 + \|u\|_{C^1(B(x,\varepsilon))})(\delta_n + \varepsilon^2 t + \varepsilon^3)) \\ &= s_k \max_{0 \leq r \leq 1} \left\{ \eta(r) \max_{|z|=r} \{ \nabla u(x) \cdot (z + [v \cdot z]_+ s_k z) + \frac{s_k}{2} z^T \nabla^2 u(x) z \} \right. \\ &\quad \left. + O((1 + \|u\|_{C^3(B(x,\varepsilon))})(\delta_n + \varepsilon^2 t + \varepsilon^3)) \right\} \\ &= s_k \max_{0 \leq r \leq 1} \{ \eta(r) H_{r,s_k}(v, \nabla u(x), \nabla^2 u(x)) \} + O((1 + \|u\|_{C^3(B(x,\varepsilon))})(\delta_n + \varepsilon^2 t + \varepsilon^3)) \\ &= s_k \max_{0 \leq r \leq 1} \{ r \eta(r) |\nabla u(x)| + ([v \cdot \nabla u(x)]_+ + \frac{1}{2} \Delta_\infty u(x)) r^2 \eta(r) s_k \} \\ &\quad + O\left((1 + |\nabla u(x)|^{-1} \|u\|_{C^2(B(x,\varepsilon))}^2 + \|u\|_{C^3(B(x,\varepsilon))})(\delta_n + \varepsilon^2 t + \varepsilon^3)\right). \end{aligned}$$

Let us set

$$B = \max_{0 \leq r \leq 1} \{ r \eta(r) |\nabla u(x)| + ([v \cdot \nabla u(x)]_+ + \frac{1}{2} \Delta_\infty u(x)) r^2 \eta(r) s_k \}$$

and let $r_1 \in [0, 1]$ so that

$$B = r_1 \eta(r_1) |\nabla u(x)| + ([v \cdot \nabla u(x)]_+ + \frac{1}{2} \Delta_\infty u(x)) r_1^2 \eta(r_1) s_k.$$

By (2.4) we have

$$\begin{aligned} B &\geq r_0 \eta(r_0) |\nabla u(x)| + ([v \cdot \nabla u(x)]_+ + \frac{1}{2} \Delta_\infty u(x)) r_0^2 \eta(r_0) s_k \\ &\geq \theta(r_1 - r_0)^2 |\nabla u(x)| + r_1 \eta(r_1) |\nabla u(x)| + ([v \cdot \nabla u(x)]_+ + \frac{1}{2} \Delta_\infty u(x)) r_0^2 \eta(r_0) s_k. \end{aligned}$$

It follows that

$$\theta(r_1 - r_0)^2 |\nabla u(x)| \leq ([v \cdot \nabla u(x)]_+ + \frac{1}{2} \Delta_\infty u(x)) (r_0^2 \eta(r_0) - r_1^2 \eta(r_1)) s_k$$

and so

$$|r_1 - r_0| \leq C \theta^{-1} (1 + |\nabla u(x)|^{-1} |\Delta_\infty u(x)|) s_k.$$

This yields

$$\begin{aligned} B &= r_0 \eta(r_0) |\nabla u(x)| + ([v \cdot \nabla u(x)]_+ + \frac{1}{2} \Delta_\infty u(x)) r_0^2 \eta(r_0) s_k \\ &\quad + O\left((1 + |\nabla u(x)|^{-1}) \|u\|_{C^2(B(x,\varepsilon))}^2 \varepsilon^2\right). \end{aligned}$$

Inserting this above we have

$$\begin{aligned} & \max_{y \in \mathcal{X}_n} \{ \eta_{\varepsilon_k(x,y)}(|x - y|)(u(y) - u(x)) \} \\ &= r_0 \eta(r_0) |\nabla u(x)| s_k + \left(\frac{1}{2} \Delta_\infty u(x) + [v \cdot \nabla u(x)]_+ \right) r_0^2 \eta(r_0) s_k^2 \\ &\quad + O\left((1 + |\nabla u(x)|^{-1} \|u\|_{C^2(B(x,\varepsilon))}^2 + \|u\|_{C^3(B(x,\varepsilon))})(\delta_n + \varepsilon^2 t + \varepsilon^3)\right). \end{aligned}$$

We can apply the same argument as above to $-u$ to obtain

$$\begin{aligned} & \min_{y \in \mathcal{X}_n} \{ \eta_{\varepsilon_k(x,y)} (|x-y|)(u(y) - u(x)) \} \\ &= -r_0 \eta(r_0) |\nabla u(x)| s_k + \left(\frac{1}{2} \Delta_\infty u(x) - [-v \cdot \nabla u(x)]_+ \right) r_0^2 \eta(r_0) s_k^2 \\ & \quad + O \left((1 + |\nabla u(x)|^{-1} \|u\|_{C^2(B(x,\varepsilon))}^2 + \|u\|_{C^3(B(x,\varepsilon))}) (\delta_n + \varepsilon^2 t + \varepsilon^3) \right). \end{aligned}$$

Therefore

$$\begin{aligned} \mathcal{L}_{s,\infty}^{n,k} u(x) &= \rho(x)^{-2/d} \left(\Delta_\infty u(x) - \frac{1}{d} \nabla \log \rho(x) \cdot \nabla u(x) \right) \\ & \quad + O \left((1 + |\nabla u(x)|^{-1} \|u\|_{C^2(B(x,\varepsilon))}^2 + \|u\|_{C^3(B(x,\varepsilon))}) (\delta_n + \varepsilon^2 t + \varepsilon^3) \right). \end{aligned}$$

□

2.5. Discrete to continuum convergence. Many types of discrete to continuum convergence results have been proven recently for various learning problems on graphs, using tools like Γ -convergence [28, 29, 52], the maximum principle and viscosity solutions [8, 10], and even Martingale techniques [16]. The Γ -convergence results are variational in nature and do not apply to the game theoretic p -Laplacian. The viscosity solution approach in [8] does not require a variational structure, but used uniform equicontinuity of the sequence of learned functions to establish compactness.

We present here a very general technique for proving discrete to continuum convergence results in a general setting that applies to all the symmetric and nonsymmetric operators discussed in Sections 2.2, 2.3 and 2.4, and any others that satisfy certain monotonicity properties. The framework only requires the graph problem to have a maximum principle, and to be pointwise consistent with a well-posed equation in the continuum. It is not necessary to prove that the sequence of functions is equicontinuous, or that the solution of the continuum PDE is smooth. The framework is essentially an adaptation of the Barles-Souganidis framework [5] to convergence of discrete problems on graphs.

Let $\Omega \subset \mathbb{R}^d$ be an open and bounded domain. For each $n \geq 1$, let $\mathcal{X}_n \subset \Omega$ be a collection of n points in Ω , and let $L^2(\mathcal{X}_n)$ denote the space of functions $u : \mathcal{X}_n \rightarrow \mathbb{R}$. Let

$$F_n : L^2(\mathcal{X}_n) \times \mathbb{R} \times \mathcal{X}_n \rightarrow \mathbb{R}$$

denote our graph operator, generalizing a graph Laplacian, and suppose $u_n \in L^2(\mathcal{X}_n)$ is a solution of the boundary value problem

$$(2.42) \quad \begin{cases} F_n(u_n, u_n(x), x) = 0, & \text{if } x \in \mathcal{X}_n \cap \Omega_{\varepsilon_n} \\ u_n(x) = g(x), & \text{if } x \in \mathcal{X}_n \cap \partial_{\varepsilon_n} \Omega, \end{cases}$$

where $g : \Omega \rightarrow \mathbb{R}$ is a given continuous function, and $\varepsilon_n > 0$ represents the length scale on which the graph is connected. For convenience, we recall that $\partial_r \Omega = \{x \in \Omega : \text{dist}(x, \partial \Omega) \leq r\}$ and $\Omega_r = \Omega \setminus \partial_r \Omega$. The graph equation (2.42) represents our generalization of a semi-supervised learning problem on a graph with labels g on the set $\mathcal{X}_n \cap \partial_{\varepsilon_n} \Omega$. This is just one model for labeled data, and others are possible (see [16] for other models).

We now lay out simple and general conditions on \mathcal{X}_n , u_n and F_n that ensure u_n has a well-posed continuum limit; that is, u_n converges uniformly to the solution of a continuum PDE. We say the points \mathcal{X}_n are *space filling* if

$$(2.43) \quad \lim_{n \rightarrow \infty} \sup_{x \in \Omega} \text{dist}(x, \mathcal{X}_n) = 0.$$

We say F_n is *monotone* if for all $u, v \in L^2(\mathcal{X}_n)$, $t \in \mathbb{R}$ and $x \in \Omega$

$$(2.44) \quad u \leq v \implies F_n(u, t, x) \geq F_n(v, t, x).$$

We say F_n is *proper* if for all $u \in L^2(\mathcal{X}_n)$, $s, t \in \mathbb{R}$, and $x \in \Omega$

$$(2.45) \quad s \leq t \implies F_n(u, s, x) \leq F_n(u, t, x).$$

We say the operators are *consistent* as $n \rightarrow \infty$ with the differential operator⁴

$$F : \mathbb{S}(d) \times \mathbb{R}^d \times \mathbb{R} \times \Omega \rightarrow \mathbb{R}$$

if for every $\varphi \in C^\infty(\mathbb{R}^d)$ and every sequence of real numbers $\xi_n \rightarrow 0$ we have

$$(2.46) \quad \lim_{n \rightarrow \infty} \max_{x \in \mathcal{X}_n \cap \Omega_{\varepsilon_n}} |F_n(\varphi + \xi_n, \varphi(x) + \xi_n, x) - F(\nabla^2 \varphi(x), \nabla \varphi(x), \varphi(x), x)| = 0.$$

We note that the connectivity length scale ε_n of the graph is encoded into the consistency statement, since we do not assume consistency near the boundary, where the ball $B(x, \varepsilon_n)$ overlaps with $\partial\Omega$.

Remark 2.22. In the context of Sections 2.2, 2.3 and 2.4, we would set $F_n(u, u(x), x) = -\mathcal{L}u(x)$, where \mathcal{L} is any of the graph Laplacians in those sections. For example, if \mathcal{L} is the unnormalized graph Laplacian (1.2), then

$$F_n(u, t, x) = \sum_{y \in \mathcal{X}_n} w_{xy}(t - u(y)).$$

The first argument of F_n encodes the dependence of F_n on the neighboring values $u(y)$ for $y \in \mathcal{X}_n$ with $y \neq x$, and the second argument t encodes the dependence on $t = u(x)$. Since all graph Laplacians are increasing functions of the difference $u(y) - t$, they are both monotone and proper. The consistency results established in Sections 2.2, 2.3 and 2.4 show that (2.46) holds with probability one for any choice of \mathcal{L} from those sections, provided $\varepsilon = \varepsilon_n \rightarrow 0$ sufficiently slowly. For example, in Theorem 2.3, in order to make sure the results hold with high probability we require

$$n\varepsilon^{3d/2} \gg \log(n) \quad \text{and} \quad n\varepsilon^{d+2} \gg \log(n).$$

This can be rewritten as lower bounds on the length scale ε as follows

$$(2.47) \quad \varepsilon \gg \left(\frac{\log(n)}{n} \right)^{2/(3d)} \quad \text{and} \quad \varepsilon \gg \left(\frac{\log(n)}{n} \right)^{1/(d+2)}.$$

We now follow the Barles-Souganidis framework [5] to prove that the sequence u_n converges uniformly to the solution of the boundary value problem

$$(2.48) \quad \begin{cases} F(\nabla^2 u, \nabla u, u, x) = 0, & \text{in } \Omega \\ u = g, & \text{on } \partial\Omega, \end{cases}$$

provided the equation (2.48) is well-posed in the viscosity sense with generalized Dirichlet condition $u = g$ on $\partial\Omega$. We review the definition of viscosity solution and the generalized Dirichlet problem in the appendix in Section A.

Theorem 2.23. *Assume g is continuous, (2.48) enjoys strong uniqueness, F_n is monotone, proper, and consistent with F , \mathcal{X}_n is space filling, and $\varepsilon_n \rightarrow 0$ as $n \rightarrow \infty$. Let u_n be a sequence of solutions of (2.42) that are uniformly bounded. Then*

$$(2.49) \quad \lim_{n \rightarrow \infty} \max_{x \in \mathcal{X}_n} |u_n(x) - u(x)| = 0,$$

where $u \in C(\bar{\Omega})$ is the unique viscosity solution of (2.48).

Remark 2.24. In Theorem 2.23, by uniformly bounded we mean that there exists $C > 0$ such that $\max_{x \in \mathcal{X}_n} |u_n(x)| \leq C$ for all $n \geq 1$. Bounds of this nature are generally proved using the discrete maximum principle on the graph, and often rely on graph-connectivity. For a simple example, consider

$$F_n(u, u(x), x) = u_n(x) + \alpha_n \sum_{y \in \mathcal{X}_n \cap B(x, \varepsilon_n)} (u_n(x) - u_n(y)) - 1.$$

⁴ $\mathbb{S}(d)$ denotes the space of symmetric real-valued matrices of size $d \times d$.

For an appropriate choice of normalization parameter α_n , the equation $F_n = 0$ would be consistent with the elliptic equation $u - \Delta u = 1$ in the continuum limit. To establish the uniform bound on u_n , we use a maximum principle argument. Let $x \in \mathcal{X}_n$ be a point where u_n attains its maximum value over \mathcal{X}_n . If $x \in \partial_{\varepsilon_n} \Omega$, then $u_n(x) \leq g(x)$. If $x \in \Omega_{\varepsilon_n}$, then we use that $u_n(x) - u_n(y) \geq 0$ for any $y \in \mathcal{X}_n$ to obtain

$$0 = F_n(u, u(x), x) = u_n(x) + \alpha_n \sum_{y \in \mathcal{X}_n \cap B(x, \varepsilon_n)} (u_n(x) - u_n(y)) - 1 \geq u_n(x) - 1.$$

Therefore $u_n(x) \leq 1$ and we obtain the bound

$$\max_{x \in \mathcal{X}_n} u_n(x) \leq \max\{1, \|g\|_\infty\} =: C.$$

A bound on the minimum of u_n is obtained similarly.

Proof of Theorem 2.23. We define the weak upper and lower limits

$$\bar{u}(x) = \limsup_{\substack{n \rightarrow \infty \\ \Omega_n \ni y \rightarrow x}} u_n(y) \quad \text{and} \quad \underline{u}(x) = \liminf_{\substack{n \rightarrow \infty \\ \Omega_n \ni y \rightarrow x}} u_n(y).$$

Due to (2.43) and the assumption that u_n are uniformly bounded, we have $-\infty < \underline{u}(x) \leq \bar{u}(x) < \infty$ for all $x \in \bar{\Omega}$. Furthermore, it is straightforward to check (see, e.g., [9]) that $\bar{u} \in \text{USC}(\bar{\Omega})$ and $\underline{u} \in \text{LSC}(\bar{\Omega})$, where $\text{USC}(\bar{\Omega})$ and $\text{LSC}(\bar{\Omega})$ denote the spaces of upper semicontinuous and lower semicontinuous functions on $\bar{\Omega}$, respectively. We claim that \bar{u} is a viscosity subsolution of (2.48) and \underline{u} is a viscosity supersolution. Once we establish this, it follows from strong uniqueness that $\bar{u} \leq \underline{u}$. Therefore $\bar{u} = \underline{u}$ and (2.49) immediately follows.

We will show that \bar{u} is a viscosity subsolution; the proof that \underline{u} is a supersolution is similar. Let $x_0 \in \bar{\Omega}$ and $\varphi \in C^\infty(\mathbb{R}^d)$ such $\bar{u} - \varphi$ has a local maximum at x_0 with respect to $\bar{\Omega}$. Define

$$\psi(x) = \varphi(x) + C|x - x_0|^4 + \bar{u}(x_0) - \varphi(x_0).$$

Then $\psi(x_0) = \bar{u}(x_0)$, and we can choose $C > 0$ large enough so that $\bar{u} - \psi$ has a strict global maximum at x_0 relative to $\bar{\Omega}$. It follows that there exists $n_k \rightarrow \infty$, $\mathcal{X}_{n_k} \ni x_{n_k} \rightarrow x_0$ with $u_{n_k}(x_{n_k}) \rightarrow \bar{u}(x_0)$ such that $u_{n_k} - \psi$ attains its maximum value over \mathcal{X}_{n_k} at the point x_{n_k} for each k . Set $\xi_k = u_{n_k}(x_{n_k}) - \psi(x_{n_k})$ so that $u_{n_k} \leq \psi + \xi_k$. Then since F_n is monotone (i.e., (2.44) holds) we have

$$(2.50) \quad F_{n_k}(\psi + \xi_k, \psi(x_{n_k}) + \xi_k, x_{n_k}) \leq F_{n_k}(u_{n_k}, u_{n_k}(x_{n_k}), x_{n_k}) = 0.$$

We now have two cases.

1. If $x_0 \in \Omega$ then $x_{n_k} \in \mathcal{X}_{n_k} \cap \Omega_{\varepsilon_{n_k}}$ for k sufficiently large, and since $\xi_k \rightarrow 0$ as $k \rightarrow \infty$ we can combine consistency (i.e., (2.46)) with (2.50) to find that

$$F(\nabla^2 \psi(x_0), \nabla \psi(x_0), \psi(x_0), x_0) \leq 0.$$

Since $\nabla^2 \psi(x_0) = \nabla^2 \varphi(x_0)$, $\nabla \psi(x_0) = \nabla \varphi(x_0)$ and $\psi(x_0) = \bar{u}(x_0)$ we have, as desired, the viscosity subsolution condition

$$(2.51) \quad F(\nabla^2 \varphi(x_0), \nabla \varphi(x_0), \bar{u}(x_0), x_0) \leq 0.$$

2. If $x_0 \in \partial\Omega$, then we can pass to a further subsequence, if necessary, so that either $x_{n_k} \in \mathcal{X}_{n_k} \cap \Omega_{\varepsilon_{n_k}}$ or $x_{n_k} \in \mathcal{X}_{n_k} \cap \partial_{\varepsilon_{n_k}} \Omega$ for all k . In the former case we again find that (2.51) holds. In the latter case, we have $u_{n_k}(x_{n_k}) = g(x_{n_k})$ for all k , and since g is continuous we have that $\bar{u}(x_0) = g(x_0)$. Thus, when $x \in \partial\Omega$ we have

$$\min \{F(\nabla^2 \varphi(x_0), \nabla \varphi(x_0), \bar{u}(x_0), x_0), \bar{u}(x_0) - g(x_0)\} \leq 0,$$

which is the viscosity subsolution condition on the boundary. This completes the proof. \square

Remark 2.25. In the context of semi-supervised learning, Theorem 2.23 shows that the discrete graph problems are well-posed in the continuum with $O(n\varepsilon_n)$ labeled data points, which is the number of points in $\mathcal{X}_n \cap \partial_{\varepsilon_n}\Omega$, provided \mathcal{X}_n are roughly evenly spread (e.g., an *i.i.d.* sequence with Lebesgue density). This is a labeling rate of $O(\varepsilon_n)$, which vanishes as $n \rightarrow \infty$. However, it may vanish very slowly, since pointwise consistency requires lower bounds on ε_n , as explained in Remark 2.22 (see Eq. (2.47)). We emphasize that this is a general result that is *independent* of the structure of the learning algorithm or of the continuum equation. For the game-theoretic p -Laplacian with $p > d$, it was shown in [8] that the ε -ball game-theoretic p -Laplacian is well-posed in the continuum with $O(1)$ labels, which is a labeling rate of $O(1/n)$. This is far smaller than $O(\varepsilon_n)$ due to (2.47). A similar result was proved for the variational p -Laplacian in [50], though here there is a restriction on ε_n for well-posedness, even when $p > d$. In [16], it is shown that when $p = 2$, Laplacian regularization is well-posed at the lower label rate of $O(\varepsilon_n^2)$ using random walk techniques. For the p -Laplacian with $2 \leq p \leq d$, it is an open problem to determine the lowest labeling rate at which the algorithm has a well-posed continuum limit. One would conjecture the lowest rate is $O(\varepsilon_n^p)$.

3. ALGORITHMS FOR P-LAPLACIAN LEARNING

We now present algorithms for p -Laplacian learning with both the game-theoretic and variational p -Laplacians. Section 3.1 reviews how to apply Newton's method to the variational p -Laplace equation, and discuss how to apply homotopy on p to accelerate convergence. Section 3.2 presents three algorithms for solving the game-theoretic p -Laplacian on a graph: a gradient descent approach, a Newton-like method, and a semi-implicit algorithm.

3.1. Newton's Method for variational p-Laplacian. Since J_p is smooth and convex, it is natural to use Newton's method to minimize J_p . We give here the explicit details of the Newton iteration for minimizing J_p . It is useful to first rewrite the function $J_p(u)$ using vector notation. Let $\mathcal{X} = \{x^1, \dots, x^{n+m}\} \subset \mathbb{R}^d$, where $\mathcal{O} = \{x^{n+1}, \dots, x^{n+m}\}$ is the observation set. We define $u_i = u(x^i)$ and set $\mathbf{u} = (u_1, \dots, u_n) \in \mathbb{R}^n$. Similarly, set $w_{ij} = w_{x_i x_j}$, $f_i = f(x^i)$, $g_i = g(x^{i+n})$, $\mathbf{f} = (f_1, \dots, f_n) \in \mathbb{R}^n$, and $\mathbf{g} = (g_1, \dots, g_m) \in \mathbb{R}^m$. Then, subject to the constraints in (1.13), we can write

$$(3.1) \quad J_p(\mathbf{u}) = \frac{1}{p} \left(\sum_{i=1}^n \sum_{j=i+1}^n w_{ij} |u_i - u_j|^p + \sum_{i=1}^n \sum_{j=1}^m w_{i,j+n} |u_i - g_j|^p \right) + \sum_{i=1}^n f_i u_i.$$

Newton's method corresponds to the iteration

$$(3.2) \quad \mathbf{u}^{k+1} = \mathbf{u}^k - \left[\nabla^2 J_p(\mathbf{u}^k) \right]^{-1} \nabla J_p(\mathbf{u}^k).$$

For notational convenience, define $a_{ij}(\mathbf{u}) = w_{ij} |u_i - u_j|^{p-2}$ and $b_{ij}(\mathbf{u}) = w_{i,j+n} |u_i - g_j|^{p-2}$, and

$$(3.3) \quad d_i(\mathbf{u}) = \sum_{j=1}^n a_{ij}(\mathbf{u}) + \sum_{j=1}^m b_{ij}(\mathbf{u}).$$

Then, we write $A(\mathbf{u}) = (a_{ij}(\mathbf{u}))_{ij} \in \mathbb{R}^{n \times n}$, $B(\mathbf{u}) = (b_{ij}(\mathbf{u}))_{ij} \in \mathbb{R}^{n \times m}$ and $D(\mathbf{u}) = \text{diag}(d_i(\mathbf{u})) \in \mathbb{R}^{n \times n}$. In this notation, we compute

$$\nabla J_p(\mathbf{u}) = L(\mathbf{u})\mathbf{u} - B(\mathbf{u})\mathbf{g} + \mathbf{f} \quad \text{and} \quad \nabla^2 J_p(\mathbf{u}) = (p-1)L(\mathbf{u}),$$

where $L(\mathbf{u}) := D(\mathbf{u}) - A(\mathbf{u})$, and thus the Newton update is given by

$$(3.4) \quad \mathbf{u}^{k+1} = \frac{p-2}{p-1} \mathbf{u}^k + \frac{1}{p-1} L(\mathbf{u}^k)^{-1} [B(\mathbf{u}^k)\mathbf{g} - \mathbf{f}].$$

The inversion of $L(\mathbf{u}^k)$ is performed with an iterative method, such as the preconditioned conjugate gradient. The matrix $L(\mathbf{u}^k)$, being a graph-Laplacian, is always positive semi-definite. If the graph is connected and \mathbf{u} is nondegenerate, then it is also non-singular.

Proposition 3.1. *If the graph on n nodes with weights $(a_{ij}(\mathbf{u}))_{i,j=1}^n$ is connected, and $b_{ij}(\mathbf{u}) > 0$ for some i, j , then $L(\mathbf{u})$ is positive definite.*

Proof. Since $L(\mathbf{u})$ is positive semi-definite, we simply have to prove that $L(\mathbf{u})$ is non-singular. The proof follows a maximum principle argument. If $L(\mathbf{u})\mathbf{x} = 0$, $\mathbf{x} = (x_1, \dots, x_n)$, then we have

$$(3.5) \quad \sum_{j=1}^n a_{ij}(\mathbf{u})(x_i - x_j) + x_i \sum_{j=1}^m b_{ij}(\mathbf{u}) = 0$$

for all $i = 1, \dots, n$.

Let i be an index for which $x_i \geq x_j$ for all j ; that is, the node where \mathbf{x} attains its maximum over the graph. We have two cases now.

Case 1. If $\sum_{j=1}^m b_{ij}(\mathbf{u}) = 0$, then it follows from (3.5), and the fact that $x_i - x_j \geq 0$ for all j , that

$$a_{ij}(\mathbf{u})(x_i - x_j) = 0 \quad \text{for all } j = 1, \dots, n.$$

Thus, at any neighbor in the graph where $a_{ij}(\mathbf{u}) > 0$ we have $x_i = x_j$.

Case 2. If $\sum_{j=1}^m b_{ij}(\mathbf{u}) > 0$, then it follows from (3.5) that

$$x_i \sum_{j=1}^m b_{ij}(\mathbf{u}) \leq 0,$$

and so $x_i \leq 0$.

Note the observations above hold for any node i where \mathbf{x} attains its maximum value. We claim that these observations imply that $\max_{1 \leq j \leq n} x_j \leq 0$. To see this, choose an index i for which \mathbf{x} attains its maximum value, and let $k \in \{1, \dots, n\}$ be a node for which

$$(3.6) \quad \sum_{j=1}^m b_{kj}(\mathbf{u}) > 0,$$

which is guaranteed to exist by assumption. Since the graph with weights $(a_{ij}(\mathbf{u}))_{i,j=1}^n$ is assumed connected, we can construct a path $i = i_0, i_1, \dots, i_q = k$ between node i and k for which $a_{i_\ell, i_{\ell+1}}(\mathbf{u}) > 0$ for $\ell = 0, \dots, q-1$. We can assume that

$$\sum_{j=1}^m b_{i_\ell, j}(\mathbf{u}) = 0 \quad \text{for } \ell = 0, \dots, q-1$$

or else we can redefine k as the earliest node along the path for which (3.6) holds. Thus, we can apply Case 1 above along the path to show that $x_{i_\ell} = x_{i_{\ell+1}}$ for $\ell = 0, \dots, q-1$. Thus, $x_i = x_k$. Then we apply Case 2 above, since (3.6) holds, to show that $x_i = x_k \leq 0$, which proves the claim.

We have proved that if $L(\mathbf{u})\mathbf{x} = 0$ then $x_i \leq 0$ for all i . Since the equation is linear, we also have $L(\mathbf{u})(-\mathbf{x}) = 0$ and so $-x_i \leq 0$ for all i . Thus, $L(\mathbf{u})\mathbf{x} = 0$ implies $\mathbf{x} \equiv 0$, and so $L(\mathbf{u})$ is nonsingular. \square

Remark 3.2. The conditions in Proposition 3.1 hold when the original graph $G = (\mathcal{X}, \mathcal{W})$ is connected, and $u_i \neq u_j$ for all $i \neq j$, and $u_i \neq g_j$ for some i, j with $w_{i,j+n} > 0$. In other words, if all the values $\{u_1, \dots, u_n, g_1, \dots, g_m\}$ are unique then $L(\mathbf{u})$. We can get into trouble when \mathbf{u} is constant, or locally constant, since the conditions in the Proposition 3.1 fail to hold. In practice we find that $L(\mathbf{u})$ remains non-singular throughout the Newton iterations.

Remark 3.3. By the Newton-Kantorovich Theorem [46], Newton's method is guaranteed to converge provided the initial guess is sufficiently close to the true solution. Since J_p is convex, but not strongly convex for $p > 2$, convergence may not be quadratic. In fact, according to [27], the number of iterations required for Newton's method to converge may scale at least linearly in some cases.

Remark 3.4 (Homotopy on p). Let us mention that one can significantly speed up Newton's method with a good starting point \mathbf{u}^0 , since most of the computational cost of Newton's method comes from approaching the solution, and once we reach the quadratic convergence region, only a few steps are required for convergence. As a result, we can solve (1.13) efficiently with a homotopy method in p . That is, we compute the solution for an increasing sequence of values of p , starting at $p = 2$, and initializing Newton's method each time from the solution from the previous value of p . If the steps in p are small enough, this initialization falls in the quadratic convergence region and only a handful of Newton iterations are required for each increment in p . This approach is known in the literature as homotopy, and has been applied to a variety of problems in order to improve performance on related problems. For example, see [26, 53]. We illustrate the effectiveness of Newton's method with homotopy in Section 4.2.

3.2. Algorithms for the game theoretic formulation . We now consider algorithms for solving the game theoretic problem (1.16). We shall discuss a gradient descent-type method in Section 3.2.1, a Newton-like algorithm in Section 3.2.2, and a semi-implicit method in Section 3.2.3.

3.2.1. Gradient Descent Approach. We first consider a gradient descent-type approach to solving (1.16), which is based on iterating

$$(3.7) \quad u^{k+1}(x) = \begin{cases} u^k(x) + \alpha(\mathcal{L}_p^G u^k(x) + f(x)), & \text{if } x \in \mathcal{X} \setminus \mathcal{O}, \\ g(x), & \text{if } x \in \mathcal{O}, \end{cases}$$

where $\alpha > 0$ is the time step. We call this a gradient descent-type iteration, where $\mathcal{L}_p^G u^k(x) + f(x)$ plays the role of a gradient, but it is important to point out it is not gradient descent, since the game theoretic p -Laplacian does not arise from a variational principle. The most straightforward stopping condition is to fix $\varepsilon > 0$ and iterate until

$$|\mathcal{L}_p^G u^k(x) + f(x)| \leq \varepsilon$$

for all $x \in \mathcal{X} \setminus \mathcal{O}$. However, this stopping condition does not guarantee that u^k is within ε of the solution u of the p -Laplacian learning problem (1.16), since the stability of the operator \mathcal{L}_p^G depends in complicated ways on the graph and choice of boundary nodes \mathcal{O} . We present here a modification of the gradient descent method that uses the comparison principle to inform the stopping condition.

Lemma 3.5 (Comparison principle). *Assume $w_{xy} \leq 1$ for all $x, y \in \mathcal{X}$. Let $u^k, v^k : \mathcal{X} \rightarrow \mathbb{R}$ satisfy*

$$(3.8) \quad u^{k+1}(x) \leq u^k(x) + \alpha(\mathcal{L}_p^G u^k(x) + f(x))$$

and

$$(3.9) \quad v^{k+1}(x) \geq v^k(x) + \alpha(\mathcal{L}_p^G v^k(x) + f(x))$$

for all $x \in \mathcal{X} \setminus \mathcal{O}$ and $0 \leq k \leq T - 1$, where $T \in \mathbb{N}$, and assume that $u^0 \leq v^0$ and $u^k(x) \leq v^k(x)$ for all $x \in \mathcal{O}$ and $1 \leq k \leq T$. If $\alpha \leq p/(2p - 3)$ then $u^k \leq v^k$ on \mathcal{X} for all $0 \leq k \leq T$.

Proof. Fix $u, v : \mathcal{X} \rightarrow \mathbb{R}$ and assume $u(x) \leq v(x)$ for all $x \in \mathcal{X}$. Fix $x \in \mathcal{X}$ and let $y_1, y_2 \in \mathcal{X}$ such that

$$\min_{y \in \mathcal{X}} w_{xy}(v(y) - v(x)) = w_{xy_1}(v(y_1) - v(x))$$

and

$$\max_{y \in \mathcal{X}} w_{xy}(u(y) - u(x)) = w_{xy_2}(u(y_2) - u(x)).$$

Since $w_{xy} \leq 1$ and $\alpha \leq p/(2p-3)$ we have that

$$1 - \frac{\alpha}{p} - \frac{\alpha}{p}(p-2)(w_{xy_1} + w_{xy_2}) \geq 0.$$

We can now compute

$$\begin{aligned} u(x) + \alpha \mathcal{L}_p^G u(x) &= u(x) + \frac{\alpha}{d_x p} \Delta_2^G u(x) + \alpha \left(1 - \frac{2}{p}\right) \Delta_\infty^G u(x) \\ &= u(x) + \frac{\alpha}{d_x p} \sum_{y \in \mathcal{X}} w_{xy} (u(y) - u(x)) \\ &\quad + \alpha \left(1 - \frac{2}{p}\right) \left\{ \min_{y \in \mathcal{X}} w_{xy} (u(y) - u(x)) + \max_{y \in \mathcal{X}} w_{xy} (u(y) - u(x)) \right\} \\ &\leq \left(1 - \frac{\alpha}{p} - \frac{\alpha}{p}(p-2)(w_{xy_1} + w_{xy_2})\right) u(x) + \frac{\alpha}{p} \frac{1}{d_x} \sum_{y \in \mathcal{X}} w_{xy} u(y) \\ &\quad + \frac{\alpha}{p} (p-2)(w_{xy_1} u(y_1) + w_{xy_2} u(y_2)) \\ &\leq \left(1 - \frac{\alpha}{p} - \frac{\alpha}{p}(p-2)(w_{xy_1} + w_{xy_2})\right) v(x) + \frac{\alpha}{p} \frac{1}{d_x} \sum_{y \in \mathcal{X}} w_{xy} v(y) \\ &\quad + \frac{\alpha}{p} (p-2)(w_{xy_1} v(y_1) + w_{xy_2} v(y_2)) \\ &\leq v(x) + \frac{\alpha}{d_x p} \sum_{y \in \mathcal{X}} w_{xy} (v(y) - v(x)) \\ &\quad + \alpha \left(1 - \frac{2}{p}\right) \left\{ \min_{y \in \mathcal{X}} w_{xy} (v(y) - v(x)) + \max_{y \in \mathcal{X}} w_{xy} (v(y) - v(x)) \right\} \\ &\leq v(x) + \alpha \mathcal{L}_p^G v(x). \end{aligned}$$

Thus, we have shown that when $\alpha \leq p/(2p-3)$ we have

$$u \leq v \implies u + \alpha \mathcal{L}_p^G u \leq v + \alpha \mathcal{L}_p^G v.$$

The proof is completed by using (3.8) and (3.9) to write

$$u^{k+1}(x) - v^{k+1}(x) \leq u^k(x) + \alpha \mathcal{L}_p^G u^k(x) - (v^k(x) + \alpha \mathcal{L}_p^G v^k(x)),$$

for $x \in \mathcal{X} \setminus \mathcal{O}$ and using induction. □

We present our method in the case that $f(x) = 0$. We define

$$(3.10) \quad \bar{u}^0(x) = \begin{cases} \max_{x \in \mathcal{O}} g(x), & \text{if } x \in \mathcal{X} \setminus \mathcal{O}, \\ g(x), & \text{if } x \in \mathcal{O}, \end{cases}$$

and

$$(3.11) \quad \underline{u}^0(x) = \begin{cases} \min_{x \in \mathcal{O}} g(x), & \text{if } x \in \mathcal{X} \setminus \mathcal{O}, \\ g(x), & \text{if } x \in \mathcal{O}, \end{cases}$$

and define $\bar{u}^k(x)$ and $\underline{u}^k(x)$ for $k \geq 1$ by

$$(3.12) \quad \bar{u}^{k+1}(x) = \bar{u}^k(x) + \alpha \mathcal{L}_p^G \bar{u}^k(x), \quad \text{and} \quad \underline{u}^{k+1}(x) = \underline{u}^k(x) + \alpha \mathcal{L}_p^G \underline{u}^k(x),$$

for $x \in \mathcal{X} \setminus \mathcal{O}$, and $\bar{u}^{k+1}(x) = \underline{u}^{k+1}(x) = g(x)$ for $x \in \mathcal{O}$.

Remark 3.6. The extension of this method to $f \neq 0$ is not immediately obvious. It is important for the convergence analysis in Theorem 3.7 below that \bar{u}^0 and \underline{u}^0 are super- and subsolutions of (1.16), respectively. When f is nonzero, it is not clear how to construct such super- and subsolutions to initialize the method. We leave the extension of the method to nonzero f to future work. We recall that in semi-supervised learning we always take $f = 0$, so this extension is not needed for machine learning applications (though it may be of interest in numerical analysis).

We prove below that this iteration scheme converges to the solution $u_* : \mathcal{X} \rightarrow \mathbb{R}$ of (1.16) with $f = 0$. We remark that the solution u_* of (1.16) is unique when the graph is connected [8, 10].

Theorem 3.7 (Convergence). *Assume $w_{xy} \leq 1$ for all $x, y \in \mathcal{X}$, and assume the graph is connected. Let u_* be the solution of (1.16) with $f = 0$, and define $u^k(x) = \frac{1}{2}(\bar{u}^k(x) + \underline{u}_k(x))$. If $\alpha \leq p/(2p-3)$ then for all $k \geq 1$ and all $x \in \mathcal{X}$ we have*

$$(3.13) \quad |u^k(x) - u_*(x)| \leq \frac{1}{2} |\bar{u}^k(x) - \underline{u}_k(x)|$$

and

$$(3.14) \quad \lim_{k \rightarrow \infty} \bar{u}^k(x) = \lim_{k \rightarrow \infty} \underline{u}^k(x) = u_*(x).$$

Remark 3.8. Theorem 3.7 proves convergence of the gradient descent-type scheme, and gives us a simple way to set the stopping condition. If we fix $\varepsilon > 0$ and iterate until

$$(3.15) \quad |\bar{u}^k(x) - \underline{u}_k(x)| \leq 2\varepsilon,$$

then Theorem 3.7 guarantees that $u^k(x) = \frac{1}{2}(\bar{u}^k(x) + \underline{u}_k(x))$ satisfies $|u^k(x) - u_*(x)| \leq \varepsilon$.

Proof. By Lemma 3.5 we have

$$\underline{u}_k(x) \leq u_*(x) \leq \bar{u}^k(x)$$

for all $k \geq 0$ and $x \in \mathcal{X}$. It follows that

$$u^k(x) - u_*(x) \leq \frac{1}{2}(\bar{u}^k(x) + \underline{u}_k(x)) - \underline{u}_k(x) \leq \frac{1}{2} |\bar{u}^k(x) - \underline{u}_k(x)|,$$

and

$$u_*(x) - u^k(x) \leq \bar{u}^k(x) - \frac{1}{2}(\bar{u}^k(x) + \underline{u}_k(x)) \leq \frac{1}{2} |\bar{u}^k(x) - \underline{u}_k(x)|.$$

This establishes (3.13).

We now prove (3.14). We claim that $\mathcal{L}_p^G \bar{u}^k(x) \leq 0$ for all $k \geq 0$ and $x \in \mathcal{X} \setminus \mathcal{O}$. The proof is by induction. Since $\bar{u}^0(y) - \bar{u}^0(x) \leq 0$ for all $x \in \mathcal{X} \setminus \mathcal{O}$ and all $y \in \mathcal{X}$, the case $k = 0$ is trivial. Now assume $\mathcal{L}_p^G \bar{u}^k(x) \leq 0$ for all $x \in \mathcal{X} \setminus \mathcal{O}$. Fix $x_0 \in \mathcal{X} \setminus \mathcal{O}$ and define

$$w(x) = \begin{cases} \bar{u}^k(x_0) + \alpha \mathcal{L}_p^G \bar{u}^k(x_0), & \text{if } x = x_0, \\ \bar{u}^k(x), & \text{if } x \neq x_0. \end{cases}$$

Then $w \leq \bar{u}^k$ and so

$$w(x_0) + \alpha \mathcal{L}_p^G w(x_0) \leq \bar{u}^k(x_0) + \alpha \mathcal{L}_p^G \bar{u}^k(x_0) = \bar{u}^{k+1}(x_0),$$

as in the proof of Lemma 3.5. Since $w(x_0) = \bar{u}^{k+1}(x_0)$ and $w \geq \bar{u}^{k+1}$, we have

$$0 \geq \mathcal{L}_p^G w(x_0) \geq \mathcal{L}_p^G \bar{u}^{k+1}(x_0),$$

which establishes the claim.

By a similar argument, we have that $\mathcal{L}_p^G \underline{u}^k(x) \geq 0$ for all $k \geq 0$ and $x \in \mathcal{X} \setminus \mathcal{O}$. It follows that $\bar{u}^{k+1} \leq \bar{u}^k$ and $\underline{u}^{k+1} \geq \underline{u}^k$ for all $k \geq 0$. Therefore, there exists $\bar{u}, \underline{u} : \mathcal{X} \rightarrow \mathbb{R}$ such that

$$\lim_{k \rightarrow \infty} \bar{u}^k(x) = \bar{u}(x) \quad \text{and} \quad \lim_{k \rightarrow \infty} \underline{u}^k(x) = \underline{u}(x).$$

By continuity of \mathcal{L}_p^G we have that \bar{u} and \underline{u} both satisfy (1.16). Since the graph is connected, solutions of (1.16) are unique, and hence $\bar{u} = \underline{u} = u_*$, which completes the proof. \square

If we make a minor modification to the equation, then we can obtain a linear convergence rate. Let $\varepsilon > 0$ and consider the iteration

$$(3.16) \quad u_\varepsilon^{k+1}(x) = u_\varepsilon^k(x) + \alpha(\mathcal{L}_p^G u_\varepsilon^k(x) - \varepsilon u_\varepsilon^k(x)),$$

for $x \in \mathcal{X} \setminus \mathcal{O}$, and $u_\varepsilon^{k+1}(x) = g(x)$ for $x \in \mathcal{O}$. We prove below that this iteration scheme converges at a *linear rate* to the solution $u_\varepsilon : \mathcal{X} \rightarrow \mathbb{R}$ of

$$(3.17) \quad \begin{cases} \varepsilon u_\varepsilon - \mathcal{L}_p^G u_\varepsilon(x) = 0 & \text{if } x \in \mathcal{X} \setminus \mathcal{O} \\ u_\varepsilon(x) = g(x) & \text{if } x \in \mathcal{O}. \end{cases}$$

When $\varepsilon > 0$, the solution u_ε of (3.17) is unique even when the graph is disconnected. We have the following convergence theorem, which is an adaptation of the contraction argument from [44].

Theorem 3.9 (Convergence rate). *Assume $w_{xy} \leq 1$ for all $x, y \in \mathcal{X}$ and $\alpha \leq p/((2+\varepsilon)p-3)$. Let $u^0 : \mathcal{X} \rightarrow \mathbb{R}$ and define u_ε^k by (3.16), and let u_ε be the solution of (3.17). Then for every $k \geq 0$ we have*

$$(3.18) \quad \max_{x \in \mathcal{X}} |u_\varepsilon^k(x) - u_\varepsilon(x)| \leq (1 - \varepsilon)^k \max_{x \in \mathcal{X}} |u^0(x) - u_\varepsilon(x)|.$$

Remark 3.10. Letting u_* solve (1.16) and u_ε solve (3.17), it follows from Theorem 3.9 that

$$\max_{x \in \mathcal{X}} |u_\varepsilon^k(x) - u_*(x)| \leq C \left[\max_{x \in \mathcal{X}} |u_\varepsilon(x) - u_*(x)| + (1 - \varepsilon)^k \right].$$

If we can quantify the error $|u_\varepsilon - u_*|$, then this would prove a convergence rate for the original problem (1.16) with $\varepsilon = 0$. However, it seems that proving error estimates between (1.16) and (3.17) would require some additional strong assumptions on properties of the graph.

Proof. Define

$$(3.19) \quad \Phi_\varepsilon[u](x) = \begin{cases} u(x) + \alpha(\mathcal{L}_p^G u(x) - \varepsilon u(x)), & \text{if } x \in \mathcal{X} \setminus \mathcal{O}, \\ g(x), & \text{if } x \in \mathcal{O}. \end{cases}$$

Then we have $u_\varepsilon^k = \Phi_\varepsilon^k[u^0]$. As in the proof of Lemma 3.5 we have

$$u \leq v \implies \Phi_0[u] \leq \Phi_0[v].$$

We also have

$$\Phi_0[u + C] = \Phi_0[u] + C$$

for any constant $C > 0$. It follows that

$$\Phi_0[u] - \Phi_0[v] = \Phi_0[u - \max_{x \in \mathcal{X}}(u - v)] - \Phi_0[v] + \max_{x \in \mathcal{X}}(u - v) \leq \max_{x \in \mathcal{X}}(u - v).$$

Since

$$\Phi_\varepsilon[u] = (1 - \alpha\varepsilon) \left(u + \frac{\alpha}{1 - \alpha\varepsilon} \mathcal{L}_p^G u \right),$$

we have

$$\Phi_\varepsilon[u] - \Phi_\varepsilon[v] \leq (1 - \varepsilon) \max_{x \in \mathcal{X}}(u - v)$$

provided

$$\frac{\alpha}{1 - \alpha\varepsilon} \leq \frac{p}{2p - 3},$$

which is equivalent to

$$\alpha \leq \frac{p}{(2 + \varepsilon)p - 3}.$$

It follows that Φ_ε is a contraction in the norm $\|u\|_\infty := \max_{x \in \mathcal{X}} |u(x)|$, and so there exists a unique fixed point $u_\varepsilon : \mathcal{X} \rightarrow \mathbb{R}$ such that $\Phi_\varepsilon[u_\varepsilon] = u_\varepsilon$ and

$$\|u_\varepsilon^k - u_\varepsilon\|_\infty \leq (1 - \varepsilon)^k \|u^0 - u_\varepsilon\|_\infty$$

for any $u : \mathcal{X} \rightarrow \mathbb{R}$. This completes the proof. \square

3.2.2. A Newton-like Algorithm. We now consider a Newton-like method for solving (1.16). Newton's method is based on iteratively solving a linearized version of the problem. In order to linearize (1.16), we define y_+^k and y_-^k by

$$(3.20) \quad \begin{aligned} y_+^k(x) &\in \operatorname{argmax}_y w_{xy}(u^k(x) - u^k(y)) \\ y_-^k(x) &\in \operatorname{argmin}_y w_{xy}(u^k(x) - u^k(y)), \end{aligned}$$

and define β_{xy}^k by

$$(3.21) \quad \beta_{xy}^k = w_{xy} \left(1 + d_x(p-2) [\delta_{y=y_+^k} + \delta_{y=y_-^k}] \right).$$

We also define

$$(3.22) \quad \mathcal{L}_{p,k}^G u(x) := \frac{1}{d_x p} \sum_y \beta_{xy}^k (u(y) - u(x)).$$

The Newton-like iteration computes u^{k+1} as the solution of

$$(3.23) \quad \begin{cases} -\mathcal{L}_{p,k}^G u^{k+1}(x) = f(x) & \text{if } x \in \mathcal{X} \setminus \mathcal{O} \\ u(x) = g(x) & \text{if } x \in \mathcal{O}. \end{cases}$$

We note that if $y_\pm^{k+1} = y_\pm^k(x)$ for all x , then $f = -\mathcal{L}_{p,k}^G u^{k+1} = -\mathcal{L}_p^G u^{k+1}$. Hence, we obtain convergence to the exact solution as soon as the locations of the min and max in (3.20) are correct. Thus, the algorithm can converge to the exact solution in a finite number of iterations. It seems rather difficult to construct a convergence proof for the Newton-like iterations, and we leave this to future work. We note that $\mathcal{L}_{p,k}^G$ is a linear operator, but may not be symmetric, which is one drawback of the method.

3.2.3. Semi-implicit Approach. Here, we extend the semi-implicit method of Oberman [45] to the graph setting. Given $\theta(x) \geq 1$, we add $-\theta(x)\Delta_2^G u(x)/(2d_x)$ to both sides of (1.16), to obtain

$$-\frac{\theta(x)}{2d_x} \Delta_2^G u(x) = -\left(\frac{\theta(x)}{2d_x} - \frac{1}{d_x p} \right) \Delta_2^G u(x) + \left(1 - \frac{2}{p} \right) \Delta_\infty^G u(x) + f(x).$$

Solving for $\Delta_2^G u(x)$ on the left hand side reduces this equation to

$$-\Delta_2^G u(x) = -\left(\frac{\theta(x)p-2}{\theta(x)p} \right) \Delta_2^G u(x) + \frac{2d_x}{\theta(x)p} (p-2) \Delta_\infty^G u(x) + \frac{2d_x}{\theta(x)} f(x),$$

which suggests the iterative scheme

$$(3.24) \quad -\Delta_2^G u^{k+1}(x) = \beta(x) \left(2\gamma(x) \Delta_\infty^G u^k(x) - \Delta_2^G u^k(x) \right) + \frac{2d_x}{\theta(x)} f(x),$$

where we have defined

$$\beta(x) = \frac{\theta(x)p-2}{\theta(x)p} \quad \text{and} \quad \gamma(x) = d_x \frac{p-2}{\theta(x)p-2}.$$

One fundamental advantage of the semi-implicit iteration (3.24) is that, unlike (3.23), the iteration (3.24) requires the solution of the same symmetric positive definite system at each iteration. This means we can dramatically speed up the solver, for large scale problems, by pre-computing a Cholesky factorization and using it at each iteration, or pre-computing an Incomplete Cholesky

factorization, to be used as a preconditioner for CG. Another favorable feature of this scheme is the fact that the conditioning of the system is independent of p , allowing the scheme to reliably solve problems for any p .

The choice of $\theta(x)$ affects stability and convergence of the scheme. We give a heuristic argument here suggesting the iteration (3.24) is a contraction when

$$(3.25) \quad \theta(x) \geq \frac{2}{p} + d_x \left(1 - \frac{2}{p}\right) =: \eta(x).$$

Define $y_{\pm}^k(x)$ as in (3.20) and write y_{\pm} in place of $y_{\pm}^k(x)$. Noting that (3.25) implies $-1 \leq 2\gamma(x) - 1 \leq 1$, we have

$$\begin{aligned} & |2\gamma\Delta_{\infty}^G u^k(x) - \Delta_2^G u^k(x)| \\ &= \left| (2\gamma(x) - 1) \left(w_{xy_-} (u^k(y_-) - u^k(x)) + w_{xy_+} (u^k(y_+) - u^k(x)) \right) - \sum_{y \neq y_{\pm}} w_{xy} (u^k(y) - u^k(x)) \right| \\ &\leq |2\gamma(x) - 1| \left(w_{xy_-} |u^k(y_-) - u^k(x)| + w_{xy_+} |u^k(y_+) - u^k(x)| \right) + \sum_{y \neq y_{\pm}} w_{xy} |u^k(y) - u^k(x)| \\ &\leq \sum_{y \in \mathcal{X}} w_{xy} |u^k(y) - u^k(x)|. \end{aligned}$$

Therefore, for $f \equiv 0$ it follows from the definition of the iteration (3.24) that

$$\left| \sum_{y \in \mathcal{X}} w_{xy} (u^{k+1}(y) - u^{k+1}(x)) \right| \leq \beta(x) \sum_{y \in \mathcal{X}} w_{xy} |u^k(y) - u^k(x)|,$$

where $0 < \beta(x) < 1$. While this is not a contraction, it is suggestive of what we observe in practice, namely that the semi-implicit iteration is a contraction when (3.25) is satisfied. It seems that a convergence proof for the semi-implicit method not straightforward. Indeed, even in the case of a uniform grid in 2-dimensions, a proof of convergence for the semi-implicit method is not available [45].

4. ALGORITHM COMPARISONS

We now give a numerical study of the performance of each algorithm on synthetic problems. This allows us, in particular, to control the intrinsic dimension of the graph—the dimension of the ambient Euclidean space or manifold from which the graph is sampled—and study how the intrinsic dimension affects convergence rates.

4.1. Experiment Design. This section specifies design choices for our synthetic data experiments.

4.1.1. Problem S (synthetic data). We sample \mathcal{X} from a uniform distribution on $[0, 1]^d$, and label 10 of these points with labels sampled from a uniform distribution on $[0, 1]$. The experiments we report on use $d = 2$, $d = 5$ and $d = 10$, as the computational cost did not change substantially for $d > 10$.

4.1.2. Graph Construction. Define the relation \sim_K on \mathcal{X} by $x \sim_K y$ if x is among the k nearest neighbors of y in Euclidean distance. We construct a symmetric K -nearest neighbor graph as follows

$$(4.1) \quad w_{xy} = \begin{cases} \exp\left(-\frac{|x-y|^2}{\sigma^2}\right), & \text{if } x \sim_K y \text{ or } y \sim_K x \\ 0, & \text{otherwise.} \end{cases}$$

	p										
N	3	4	6	8	10	15	20	25	30	40	50
1	2e-01	3e-02	1e-02	8e-04	2e-04	8e-05	4e-06	3e-07	3e-08	1e-09	2e-11
2	5e-02	3e-03	4e-04	1e-05	7e-07	1e-06	2e-08	6e-10	3e-11	5e-12	3e-14
3	7e-03	2e-04	4e-05	7e-08	2e-10	3e-10	3e-13	4e-15	1e-16	9e-17	-
4	8e-04	1e-05	4e-06	2e-09	1e-13	2e-14	-	-	-	-	-
5	5e-05	4e-07	8e-07	5e-12	-	-	-	-	-	-	-
6	4e-07	1e-09	7e-08	2e-16	-	-	-	-	-	-	-
7	2e-11	2e-14	6e-10	-	-	-	-	-	-	-	-
8	8e-15	-	1e-12	-	-	-	-	-	-	-	-
9	-	-	2e-16	-	-	-	-	-	-	-	-

TABLE 1. Newton’s method with homotopy solved problem S for $p = 50, \epsilon < 10^{-12}$ in just 56 iterations. We take increasingly large steps in p while remaining in the quadratic convergence regime, suggesting we could go much farther than $p = 50$.

The constant σ plays the role of being a *typical length scale* for the problem. In our experiments, we compute it as $\sigma = \frac{1}{2} \max \{|x - y| : w_{xy} > 0\}$. We use $K = 10$ in all experiments. Other choices of K would produce similar accuracy, but larger K would increase the cost of each iteration. The only requirement is that K needs to be large enough to ensure the graph is connected. It should be noted that, even though we prescribe $K = 10$, after we apply (4.1), most vertices will have a higher number of neighbors (so that symmetry can be enforced).

4.1.3. *Error Reporting.* For the variational formulation, we report

$$(4.2) \quad \epsilon = \frac{1}{n\sigma^{d+p-1}} \max_{x \in \mathcal{X}} |\Delta_p^G u(x)|,$$

which amounts to scaling the largest residual in equation (1.13), by the number of points, and the typical length scale σ . This scaling ensures a fair comparison across different n, p, d . For the game theoretic formulation, the scaling of the residual in equation (1.16) is different, and we report

$$(4.3) \quad \epsilon = \frac{1}{\sigma} \max_{x \in \mathcal{X}} |\mathcal{L}_p^G u(x)|.$$

4.2. Homotopy Results for Newton and Newton-like Methods. Homotopy is very effective at improving the performance and reliability of Newton’s method for higher p . We illustrate this by solving problem S with $d = 10, p = 50, m = 10$ and $n = 10^4$. We use the solution to the $p = 2$ problem as \mathbf{u}^0 for the $p = 3$ problem. Then, we apply homotopy until $p = 50$ is reached. For each p , we require $\epsilon < 10^{-12}$ according to (4.2). Table 1 reports ϵ at each iteration.

We repeat the experiment using the Newton-like algorithm for the game-theoretic formulation (now ϵ is given by (4.3)). As we can see from Table 2, homotopy is equally effective in this case.

4.3. Computational Cost vs. Dimension d and Size n . Our timing results were performed on a laptop with 16GB of RAM memory. For the gradient descent method, we used a C-language implementation. Other methods were implemented in MATLAB 2018, as the computational cost was dominated by matrix operations. For these methods, we observed the cost of direct solvers increase by as much as $100\times$, when going from $d = 2$ up to $d = 100$, hence we used iterative methods instead. For Newton’s method, we used a conjugate gradient’s method (CG), preconditioned with an Incomplete Cholesky factorization. Given that the matrix system for Newton-like’s method is no longer symmetric, we used a Generalized Minimal Residual Algorithm (GMRES) [48], preconditioned with an Incomplete LU factorization instead. The semi-implicit method performed very well with a preconditioned CG method. Since the matrix to be inverted does not change, we compute

	p										
N	3	4	6	8	10	15	20	25	30	40	50
1	4e-02	1e-02	9e-03	4e-03	2e-03	3e-03	1e-03	7e-04	5e-04	6e-04	3e-04
2	4e-03	3e-03	3e-03	2e-03	1e-03	3e-03	2e-03	7e-04	3e-04	9e-04	6e-04
3	1e-03	8e-04	2e-03	6e-04	5e-04	2e-03	9e-04	1e-04	9e-05	1e-04	3e-05
4	1e-04	1e-04	5e-04	5e-04	6e-05	3e-04	5e-04	3e-16	3e-16	3e-16	3e-16
5	6e-06	1e-05	9e-05	2e-04	2e-07	6e-05	3e-16	-	-	-	-
6	2e-16	2e-16	2e-16	8e-06	4e-16	3e-16	-	-	-	-	-
7	-	-	-	3e-16	-	-	-	-	-	-	-

TABLE 2. The Newton-like method with homotopy solved problem S for $p = 50$ and $\epsilon < 10^{-12}$ in 52 iterations. Similar to Newton’s method, increasingly large step sizes in p are possible, suggesting the ability to go even higher in p .

an Incomplete Cholesky factorization, with a drop tolerance of 10^{-1} , at the very beginning, and use it as our preconditioner in each iteration.

We explore how the Newton, Newton-like, Semi-Implicit, and Gradient Descent algorithms scale with n, d . To measure this relationship, we solved problem S for $p = 11$, until $\epsilon < 10^{-7}$, according to (4.2) and (4.3) respectively. These results are reported in Figure 2. In the figure, the complexity of Newton’s method scales roughly like $n^{0.8}$, the Newton-like method scales roughly like $n^{1.3}$, the semi-implicit method scales like n^1 for $d = 2$ and $d^{0.75}$ for $d = 5, 10$, and the gradient descent-type method scales like n^2 for $d = 2$, and $n^{1.5}$ for $d = 5, 10$. We note that Figure 2a and Figures 2(b-d) cannot be directly compared, since the parameters p, ϵ do not have the same meaning in both problems, but give a rough idea of the relative performance of the variational and game-theoretic solvers. We also illustrate our ability to solve large scale problems by solving problem S, with $m = 10$, $K = 10$, $n = 5 \cdot 10^5$ and $d = 10$. For the variational formulation, we use Newton’s method starting from the solution of the $p = 2$ problem (1-step homotopy). For the game theoretic formulation, we choose the semi-implicit method. In both cases, iterative solvers were used. We plot the results, averaged over 5 trials in Figure 3. In the figure, the CPU time for Newton’s method scales like $\epsilon^{-0.05}$ with tolerance $\epsilon > 0$, while the CPU time for the semi-implicit method scales like $\epsilon^{-0.2}$.

Remark 4.1. It is interesting to point out that the results in Figure 2 show that the cost of all four methods decreases (sometimes significantly) as the dimension d increases. This is due to the graph Laplacian matrices having better condition numbers in higher dimensions. To see why this is the case, we recall that on a uniform grid, the condition number for a discrete Laplacian is on the order of Δx^{-2} , where Δx is the grid resolution. When Δx is small, the Laplacian is poorly conditioned and iterative methods are stiff without some form of preconditioning. The same scaling for the condition number holds for graph Laplacians constructed from random geometric graphs, due to the spectral convergence results in, for example, [14], but now Δx should be replaced by the length scale $\epsilon > 0$ on which the graph is constructed. In dimension d , the length scale is at least $\epsilon \geq n^{-1/d}$, to ensure graph connectivity. This implies that as a function of n and d , the condition number of the graph Laplacian should scale like $n^{-2/d}$. Thus, the curse of dimensionality *improves* the condition number of graph Laplacians in higher dimensions, and explains why iterative methods perform better for larger d . The stiffness of Laplace equations is essentially a phenomenon of high resolution meshes, which are only possible to construct in low dimensional spaces (e.g., $d = 2, 3$). By the manifold assumption in machine learning [18], we expect our graphs to have intrinsic dimensionality significantly higher than $d = 3$, and so the graph Laplacians we encounter in practice are better conditioned than what we may expect from experience with solving PDEs in $d = 2$ or $d = 3$ dimensions.

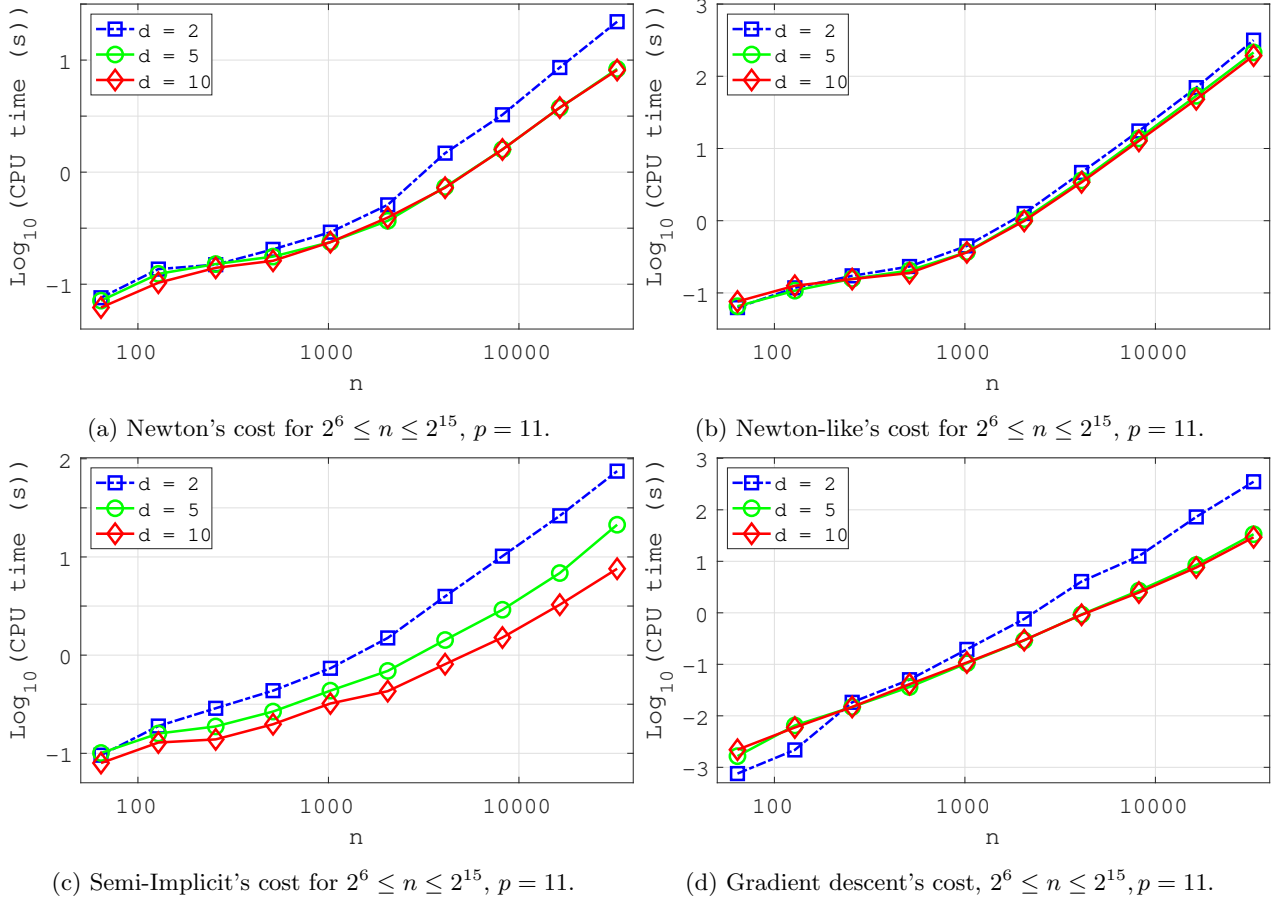


FIGURE 2. The computational cost of solving problem S , averaged over 5 trials. (a) Newton's cost in solving problem S until $\epsilon < 10^{-7}$ with preconditioned CG. (b) Newton-like's cost in solving problem S , until $\epsilon < 10^{-7}$, with preconditioned GMRES. (c) Semi-implicit method's cost for solving problem S until $\epsilon < 10^{-7}$, with preconditioned CG. (d) Gradient descent's cost for solving problem S until $\epsilon < 10^{-7}$.

4.4. Synthetic Experiment Conclusions. The experiments shown demonstrate that both variational and game-theoretic formulations can be solved with the algorithms we propose for large scale graphs with high intrinsic dimensionality, similar to what will be observed in practice. For the game-theoretic problem, the semi-implicit formulation is likely to perform the best, although it lacks the convergence guarantees enjoyed by the gradient descent method. The Newton and Newton-like algorithms show strong resemblance to one another, and both methods improve their performance and reliability substantially when homotopy is utilized.

5. EXPERIMENTS WITH REAL DATA

We now give an experimental study of p -Laplacian semi-supervised learning classification on real datasets, including MNIST, and the more complex datasets Fashion MNIST and Extended MNIST.

5.1. Description of the Experiments and Datasets. On each dataset, we shall perform two experiments. For the first, we explore the performance as the number of labels varies, with the goal of understanding how different models perform with increasing, but very small, amounts of labeled data. For the second experiment, we fix the number of labels to 1 per class, and explore

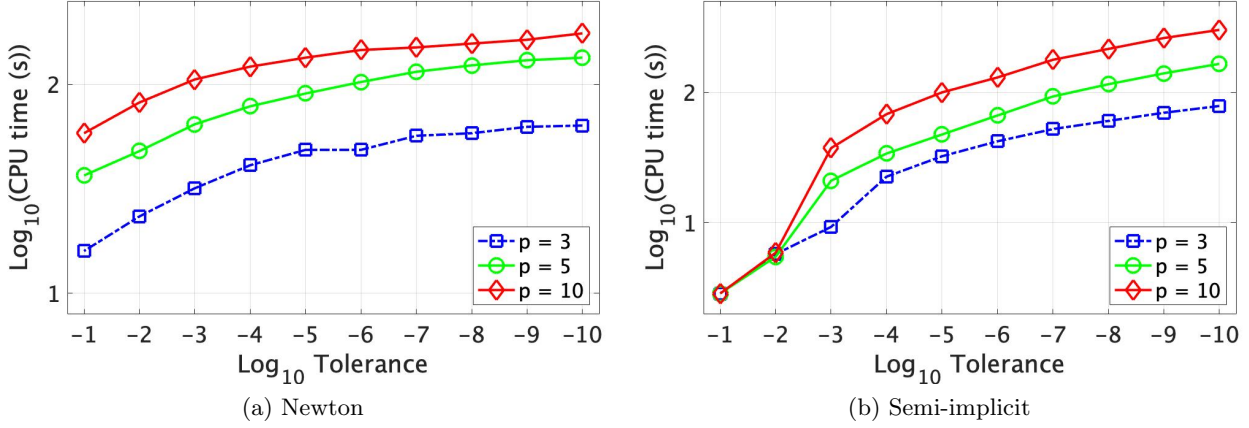


FIGURE 3. Solving problem S at large scale with $n = 5 \cdot 10^5$, $p = 5$ and $d = 10$.
(a) Newton’s cost for solving the variational formulation, using preconditioned CG.
(b) Semi-implicit’s cost for the game-theoretic problem, with preconditioned CG.

the performance as a function of the amount of unlabeled data. This experiment helps to illustrate the degeneracy of the 2-Laplacian, while also illustrating how other models can profit from an increasing amount of unlabeled data, which is the premise upon which semi-supervised learning is built. Before proceeding to the results, we provide some detail on the datasets we work with.

In all three problems, we first preprocess each image via a Scattering Transform [7] in order to extract features upon which to build the graph. The graphs are constructed as K -nearest neighbor graphs in the feature space using the weights (4.1). In all experiments we chose $K = 10$, to ensure graph connectivity and sufficient sparsity. We also experimented with $K = 25$ and $K = 50$ and found that increasing the number of neighbors in the graph does not provide any advantages, and results in significantly higher computational cost.

We solve the M -class classification problem with the standard one-vs-rest approach, whereby we solve M binary classification problems classifying each digit against the rest. This gives M scores for each unlabeled image, and the final classification is chosen as the class with the highest score.

5.1.1. MNIST. The MNIST dataset is a standard benchmark that consists of 70,000 images of handwritten digits 0 through 9 [37]. Each image is a 28×28 pixel grayscale image, meaning it can be represented as a vector in \mathbb{R}^{784} dimensions. The classes are well balanced, with roughly 7,000 examples for each digit.

5.1.2. Fashion MNIST. This dataset, recently introduced by [55], was designed as a drop-in replacement for MNIST, in order to test classification accuracy for a significantly harder problem. This dataset also consists of 70,000 grayscale images of size 28×28 pixels, except the 10 handwritten digits are replaced by 10 classes of clothing items, such as sandals, and dresses. For an example, see Figure 4.

5.1.3. Extended MNIST. The Extended MNIST (EMNIST) dataset [19] contains images of handwritten letters, of 28×28 pixels. Given that letters may be uppercase or lowercase, and there are now 26 classes, the semisupervised task on this dataset is substantially more challenging. To simplify the task slightly, we downsample the dataset for the first experiment (performance vs. number of labels) so that all classes have the same number of labels. We end up with 3419 samples for each of the 26 classes, for a total of $n = 88,894$ images. For the second task, we measure performance with subsampled datasets of $2^5, 2^6, \dots, 2^{11}$ samples per class, always having 1 labeled sample.

FIGURE 4. Sample 28×28 -pixel images from the Fashion MNIST dataset.

5.2. Experimental Results. We report the results of our experiments in Figure 5. In all experiments, we report results using 5 formulations: the standard 2-Laplacian, the Weighted Non-local Laplacian (WNLL) [49], and the game-theoretic p -Laplacian for $p = 5$, $p = 9$ and $p = \infty$ (e.g. Lipschitz learning). All models were solved to a tolerance of $\epsilon < 10^{-2}$, which provides consistent results. The results we report paint a similar picture across all three datasets. For the first type of experiment (left column), all formulations improve their accuracy as the number of labels increases, and they gradually approach a similar level as m grows. This means that the choice of a good model is of particular importance in the regime when labeled data is extremely limited. Both the p -Laplace and WNLL methods clearly outperform the 2-Laplacian at low label rates, while the WNLL and p -Laplacian give fairly similar results.

The second experiment (right column) provides further evidence of the superiority of the p -Laplacian model over the 2-Laplacian. In this case we fix 1 label per class, which means $m = 10$ for MNIST and Fashion MNIST, while $m = 26$ for EMNIST. The premise of semi-supervised is that we may achieve superior performance by including both labeled and unlabeled data. If we keep m fixed, we should expect the accuracy to improve as the number of unlabeled images increases. In all three datasets, the 2-Laplacian classification performance decreases substantially as the amount of unlabeled data n grows. This illustrates the need for alternative models when $n \gg m$. On the other hand, the p -Laplacian models (for sufficiently large p) do not degenerate as n grows, and in fact, their performance increases slightly as n grows. The Weighted Non-local Laplacian exhibits mixed results. For the MNIST and Fashion MNIST datasets, performance decreases slightly when $n = 70,000$. Meanwhile, this model outperforms others on the EMNIST dataset.

Let us make some final remarks comparing and contrasting WNLL and p -Laplace learning. The experimental results in Figure 5 show that neither method is strictly better than the other. For the most part, the methods are comparable. The largest difference is seen at 1 label per class on MNIST where p -Laplace is roughly 10% better than WNLL. The difference is much more pronounced in terms of the theoretical guarantees for each method. It was shown in [8] (see [50] for the variational p -Laplacian) that p -Laplace learning is well-posed with arbitrarily few labeled examples; in fact, we can take the number of labeled examples to be finite while sending the number of unlabeled examples to infinity, and still obtain a well-posed continuous extension of the label values. On the other hand, the WNLL was shown in [15] to be *ill-posed* in the same setting of finite labeled data and infinite unlabeled data. In particular, [15, Corollary 3.8] shows that the WNLL with finite labeled data converges to a constant labeling function as the amount of unlabeled data tends to infinity. To the best of our knowledge, the only semi-supervised learning algorithm with theoretical guarantees at arbitrarily low label rates, such as the ones given in [8, 50], is the graph p -Laplacian.

6. CONCLUSIONS

This paper is focused on theory and applications of ℓ_p -Laplacian regularized semi-supervised learning. We performed a detailed analysis of discrete to continuum theory for the p -Laplacian on k -NN graphs, which are more commonly used in practice compared to random geometric graphs, and made the surprising discovery that the p -Laplacian models retain information about the data

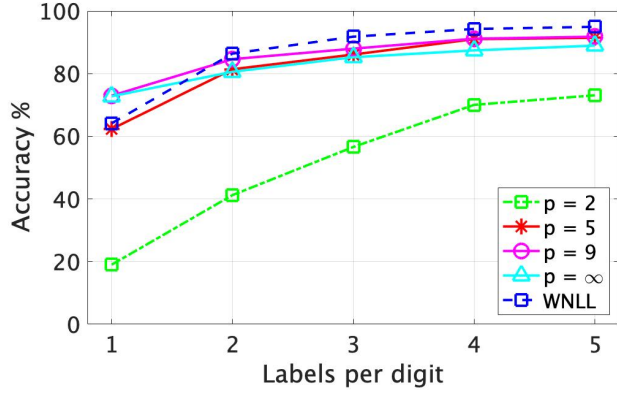
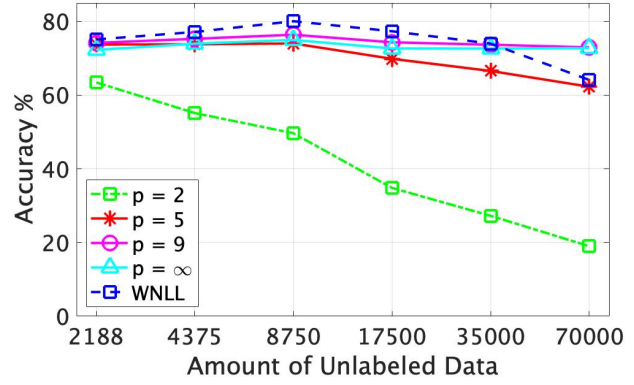
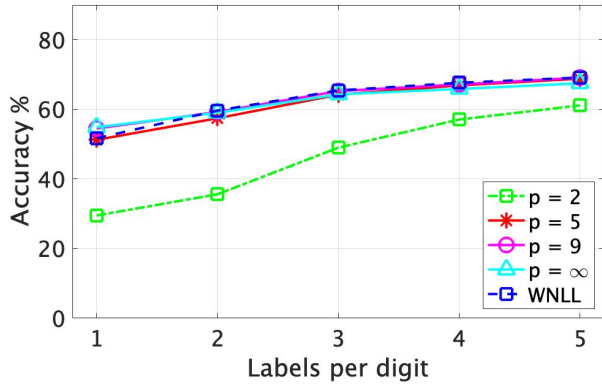
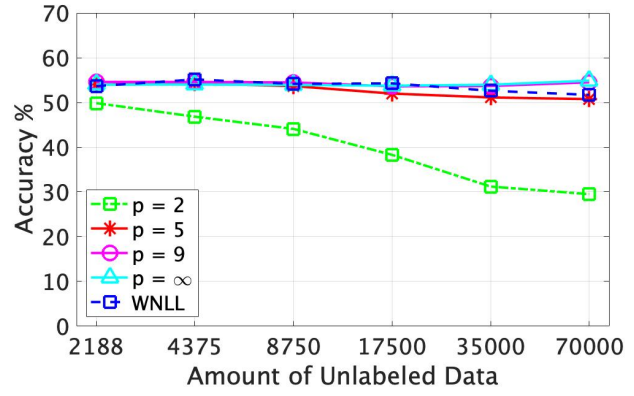
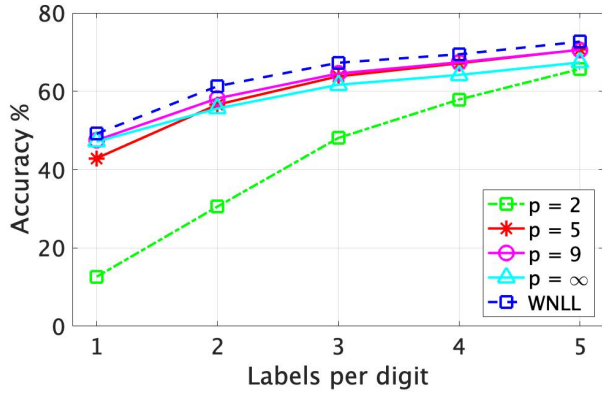
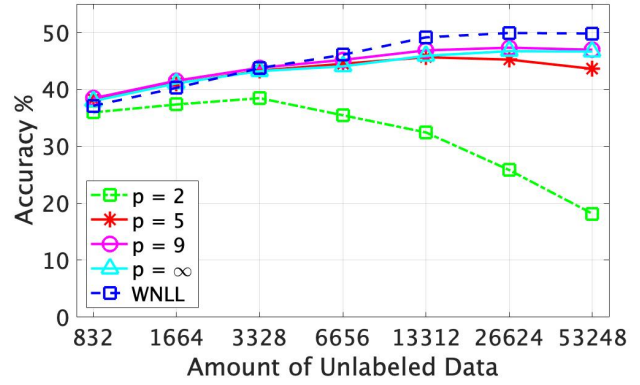
(a) MNIST, $n = 70000$, $10 \leq m \leq 50$.(b) MNIST, $m = 10$, $2188 \leq n \leq 70000$.(c) Fashion MNIST, $n = 70000$, $10 \leq m \leq 50$.(d) Fashion MNIST, $m = 10$, $2188 \leq n \leq 70000$.(e) EMNIST, $n = 88894$, $10 \leq m \leq 50$.(f) EMNIST, $m = 26$, $832 \leq n \leq 53248$.

FIGURE 5. Classification accuracy for the MNIST, Fashion MNIST and Extended MNIST datasets. Figures (a,c,e) show the performance as a function of the number of labels per class, while (b,d,f) show the performance for 1 label per class as a function of the amount of unlabeled data used.

distribution as $p \rightarrow \infty$ on k -NN graphs, contrary to conventional wisdom from the existing p -Laplace theory on random geometric graphs. We also presented a simple and very general framework for proving discrete to continuum convergence results that only requires pointwise consistency and a monotonicity property. We expect this framework to be useful in future work.

We also studied and developed algorithms for solving the variational and game-theoretic formulations of the p -Laplacian on a weighted graph. The variational formulation may be solved efficiently using Newton's method with homotopy. The semi-implicit method is the fastest method for solving the game-theoretic formulation, while the gradient-descent approach enjoys rigorous convergence guarantees. Our experiments with real data show that p -Laplacian learning is superior to Laplace learning ($p = 2$) at very low label rates on common image classification datasets including MNIST, FashionMNIST and EMNIST.

APPENDIX A. REVIEW OF VISCOSITY SOLUTIONS

Viscosity solutions are a notion of weak solution for partial differential equations that obeys the maximum principle and enjoys strong stability and uniqueness theorems. The theory is especially useful for passing from discrete to continuum limits (see, e.g., [9, 13, 17]). We review here the basic definitions. Let $USC(\bar{\Omega})$ (resp. $LSC(\bar{\Omega})$) denote the collection of functions that are upper (resp. lower) semicontinuous at all points in $\bar{\Omega}$. We make the following definitions.

Definition A.1. We say $u \in USC(\bar{\Omega})$ is a viscosity subsolution of (2.48) if for all $x \in \bar{\Omega}$ and every $\varphi \in C^\infty(\mathbb{R}^n)$ such that $u - \varphi$ has a local maximum at x with respect to $\bar{\Omega}$

$$\begin{cases} F(\nabla^2 u(x), \nabla \varphi(x), u(x), x) \leq 0, & \text{if } x \in \Omega \\ \min \{F(\nabla^2 \varphi(x), \nabla \varphi(x), u(x), x), u(x) - g(x)\} \leq 0 & \text{if } x \in \partial\Omega. \end{cases}$$

Likewise, we say that $u \in LSC(\bar{\Omega})$ is a viscosity supersolution of (2.48) if for all $x \in \bar{\Omega}$ and every $\varphi \in C^\infty(\mathbb{R}^n)$ such that $u - \varphi$ has a local minimum at x with respect to $\bar{\Omega}$

$$\begin{cases} F(\nabla^2 u(x), \nabla \varphi(x), u(x), x) \geq 0, & \text{if } x \in \Omega \\ \max \{F(\nabla^2 \varphi(x), \nabla \varphi(x), u(x), x), u(x) - g(x)\} \geq 0 & \text{if } x \in \partial\Omega. \end{cases}$$

Finally, we say that u is a viscosity solution of (2.48) if u is both a viscosity sub- and supersolution. In this case, we say that the boundary conditions in (2.48) hold in the viscosity sense, which is also known as the generalized Dirichlet sense.

Definition A.2. We say that (2.48) enjoys strong uniqueness if whenever $u \in USC(\bar{\Omega})$ is a subsolution of (2.48) and $v \in LSC(\bar{\Omega})$ is a supersolution, we have $u \leq v$ on $\bar{\Omega}$.

We refer the reader to [20] for the proof of the comparison principle for viscosity solutions with generalized Dirichlet boundary conditions, which implies strong uniqueness.

REFERENCES

- [1] D. Adil, R. Peng, and S. Sachdeva. Fast, provably convergent irls algorithm for p-norm linear regression. In *Advances in Neural Information Processing Systems*, pages 14166–14177, 2019.
- [2] M. Alamgir and U. V. Luxburg. Phase transition in the family of p-resistances. In *Advances in Neural Information Processing Systems*, pages 379–387, 2011.
- [3] R. K. Ando and T. Zhang. Learning on graph with Laplacian regularization. In *Advances in neural information processing systems*, pages 25–32, 2007.
- [4] G. Aronsson, M. Crandall, and P. Juutinen. A tour of the theory of absolutely minimizing functions. *Bulletin of the American mathematical society*, 41(4):439–505, 2004.
- [5] G. Barles and P. E. Souganidis. Convergence of approximation schemes for fully nonlinear second order equations. *Asymptotic analysis*, 4(3):271–283, 1991.
- [6] N. Bridle and X. Zhu. p-voltages: Laplacian regularization for semi-supervised learning on high-dimensional data. In *Eleventh Workshop on Mining and Learning with Graphs (MLG2013)*, 2013.

- [7] J. Bruna and S. Mallat. Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1872–1886, 2013.
- [8] J. Calder. The game theoretic p -Laplacian and semi-supervised learning with few labels. *Nonlinearity*, 32(1), 2018.
- [9] J. Calder. Lecture notes on viscosity solutions. 2018. Online Lecture Notes: http://www-users.math.umn.edu/~jwcalder/viscosity_solutions.pdf.
- [10] J. 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.
- [11] J. Calder. Calculus of variations. *Lecture notes*, 2020. <http://www-users.math.umn.edu/~jwcalder/CalculusOfVariations.pdf>.
- [12] J. Calder. GraphLearning Python Package. doi:10.5281/zenodo.5850940, 2022. <https://github.com/jwcalder/GraphLearning>.
- [13] J. Calder, S. Esedoglu, and A. O. Hero. A Hamilton–Jacobi equation for the continuum limit of nondominated sorting. *SIAM Journal on Mathematical Analysis*, 46(1):603–638, 2014.
- [14] J. Calder and N. García Trillos. Improved spectral convergence rates for graph Laplacians on ε -graphs and k -NN graphs. *arXiv:1910.13476*, 2019.
- [15] J. Calder and D. Slepčev. Properly-weighted graph laplacian for semi-supervised learning. *Applied Mathematics & Optimization*, pages 1–49, 2019.
- [16] J. Calder, D. Slepčev, and M. Thorpe. Rates of convergence for Laplacian semi-supervised learning with low labeling rates. *arXiv:2006.02765*, 2020.
- [17] J. Calder and C. K. Smart. The limit shape of convex hull peeling. *Duke Mathematical Journal*, 169(11):2079–2124, 2020.
- [18] O. Chapelle, B. Scholkopf, and A. Zien. *Semi-supervised learning*. MIT, 2006.
- [19] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926. IEEE, 2017.
- [20] M. G. Crandall, H. Ishii, and P.-L. Lions. User’s guide to viscosity solutions of second order partial differential equations. *Bulletin of the American mathematical society*, 27(1):1–67, 1992.
- [21] A. El Alaoui, X. Cheng, A. Ramdas, M. J. Wainwright, and M. I. Jordan. Asymptotic behavior of ℓ_p -based Laplacian regularization in semi-supervised learning. In *Conference on Learning Theory*, pages 879–906, 2016.
- [22] A. Elmoataz, X. Desquesnes, and M. Toutain. On the game p -Laplacian on weighted graphs with applications in image processing and data clustering. *European Journal of Applied Mathematics*, 28(6):922–948, 2017.
- [23] A. Elmoataz, F. Lozes, and M. Toutain. Nonlocal PDEs on graphs: From tug-of-war games to unified interpolation on images and point clouds. *Journal of Mathematical Imaging and Vision*, 57(3):381–401, 2017.
- [24] A. Elmoataz, M. Toutain, and D. Tenbrinck. On the p -Laplacian and ∞ -Laplacian on graphs with applications in image and data processing. *SIAM Journal on Imaging Sciences*, 8(4):2412–2451, 2015.
- [25] L. Evans. *Partial Differential Equations (Graduate Studies in Mathematics, V. 19) GSM/19*. American Mathematical Society, June 1998.
- [26] R. Fletcher, J. Grant, and M. Hebden. The calculation of linear best L_p approximations. *The Computer Journal*, 14(3):276–279, 1971.
- [27] M. Flores. Algorithms for semisupervised learning on graphs. 2018.
- [28] N. García Trillos. Variational limits of k -nn graph-based functionals on data clouds. *SIAM Journal on Mathematics of Data Science*, 1(1):93–120, 2019.
- [29] N. García Trillos and D. Slepčev. Continuum limit of Total Variation on point clouds. *Archive for Rational Mechanics and Analysis*, 220(1):193–241, 2016.
- [30] Y. Hafiene, J. Fadili, and A. Elmoataz. Nonlocal p -Laplacian variational problems on graphs. *arXiv:1810.12817*, 2018.
- [31] J. He, M. Li, H.-J. Zhang, H. Tong, and C. Zhang. Manifold-ranking based image retrieval. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 9–16. ACM, 2004.
- [32] J. He, M. Li, H.-J. Zhang, H. Tong, and C. Zhang. Generalized manifold-ranking-based image retrieval. *IEEE Transactions on image processing*, 15(10):3170–3177, 2006.
- [33] M. Hein, J.-Y. Audibert, and U. v. Luxburg. Graph Laplacians and their convergence on random neighborhood graphs. *Journal of Machine Learning Research*, 8(Jun):1325–1368, 2007.
- [34] M. Hein, J.-Y. Audibert, and U. Von Luxburg. From graphs to manifolds—weak and strong pointwise consistency of graph Laplacians. In *International Conference on Computational Learning Theory*, pages 470–485. Springer, 2005.
- [35] A. Jung, A. O. Hero III, A. Mara, and S. Jahromi. Semi-supervised learning via sparse label propagation. *arXiv preprint arXiv:1612.01414*, 2016.
- [36] R. Kyng, A. Rao, S. Sachdeva, and D. A. Spielman. Algorithms for Lipschitz learning on graphs. In *Conference on Learning Theory*, pages 1190–1223, 2015.

- [37] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [38] G. Leoni. *A first course in Sobolev spaces*. American Mathematical Soc., 2017.
- [39] M. Lewicka and J. J. Manfredi. Game theoretical methods in PDEs. *Bollettino dell’Unione Matematica Italiana*, 7(3):211–216, 2014.
- [40] P. Lindqvist. *Notes on the p -Laplace equation*. 2017.
- [41] U. v. Luxburg and O. Bousquet. Distance-based classification with Lipschitz functions. *Journal of Machine Learning Research*, 5(Jun):669–695, 2004.
- [42] J. J. Manfredi, A. M. Oberman, and A. P. Sviridov. Nonlinear elliptic partial differential equations and p -harmonic functions on graphs. *Differential Integral Equations*, 28(1-2):79–102, 2015.
- [43] B. Nadler, N. Srebro, and X. Zhou. Semi-supervised learning with the graph Laplacian: The limit of infinite unlabelled data. *Advances in neural information processing systems*, 22:1330–1338, 2009.
- [44] A. M. Oberman. Convergent difference schemes for degenerate elliptic and parabolic equations: Hamilton–jacobi equations and free boundary problems. *SIAM Journal on Numerical Analysis*, 44(2):879–895, 2006.
- [45] A. M. Oberman. Finite difference methods for the infinity laplace and p -laplace equations. *Journal of Computational and Applied Mathematics*, 254:65–80, 2013.
- [46] J. M. Ortega. The newton-kantorovich theorem. *The American Mathematical Monthly*, 75(6):658–660, 1968.
- [47] Y. Peres, O. Schramm, S. Sheffield, and D. Wilson. Tug-of-war and the infinity Laplacian. *Journal of the American Mathematical Society*, 22(1):167–210, 2009.
- [48] Y. Saad and M. H. Schultz. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869, 1986.
- [49] Z. Shi, S. Osher, and W. Zhu. Weighted nonlocal Laplacian on interpolation from sparse data. *Journal of Scientific Computing*, 73(2-3):1164–1177, 2017.
- [50] D. Slepcev and M. Thorpe. Analysis of p -laplacian regularization in semisupervised learning. *SIAM Journal on Mathematical Analysis*, 51(3):2085–2120, 2019.
- [51] D. Ting, L. Huang, and M. Jordan. An analysis of the convergence of graph laplacians. *arXiv preprint arXiv:1101.5435*, 2011.
- [52] N. G. Trillos and R. Murray. A maximum principle argument for the uniform convergence of graph laplacian regressors. *arXiv preprint arXiv:1901.10089*, 2019.
- [53] R. A. Vargas and C. S. Burrus. Iterative design of lp fir and iir digital filters. In *2009 IEEE 13th Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop*, pages 468–473, Jan 2009.
- [54] Y. Wang, M. A. Cheema, X. Lin, and Q. Zhang. Multi-manifold ranking: Using multiple features for better image retrieval. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 449–460. Springer, 2013.
- [55] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv:1708.07747*, 2017.
- [56] B. Xu, J. Bu, C. Chen, D. Cai, X. He, W. Liu, and J. Luo. Efficient manifold ranking for image retrieval. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 525–534. ACM, 2011.
- [57] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang. Saliency detection via graph-based manifold ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3166–3173, 2013.
- [58] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in neural information processing systems*, pages 321–328, 2004.
- [59] D. Zhou, J. Huang, and B. Schölkopf. Learning from labeled and unlabeled data on a directed graph. In *Proceedings of the 22nd international conference on Machine learning*, pages 1036–1043. ACM, 2005.
- [60] D. Zhou and B. Schölkopf. Regularization on discrete spaces. In *Joint Pattern Recognition Symposium*, pages 361–368. Springer, 2005.
- [61] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf. Ranking on data manifolds. In *Advances in neural information processing systems*, pages 169–176, 2004.
- [62] X. Zhu, Z. Ghahramani, and J. D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, pages 912–919, 2003.