SparRow

A Bird's Tale

zur Erlangung des Doktorgrades

Dr. rer. nat.

im Studiengang Mathematik

am Department Mathematik der
Friedrich-Alexander-Universität Erlangen-Nürnberg

vorgelegt am **21.März 2023**

von **Tim Roith**

Prüfer: Martin Burger
Betreuer: M.Sc. C

Für eine besondere Person,
ohne sie hätte ich es nie geschafft.

Acknowledgement

I would like to thank my supervisors for helping me finish this thesis and supporting me with their profound knowledge..

Contents

List of Figures

Part I.

Intro

Chapter 1
About

This document demonstrates the appearance of a thesis build with the `fau-book` class. A full description will be available soon at my webpage. If you have any questions or found a problem/bug please don't hesitate to contact me via e-mail[1].

[1] tim.roith@fau.de

Chapter 2
Semi-Supervised Learning

2.1. Setting

Definition 2.1 (Weighted Graphs). This picture
of
a
weighted graph

2.2. Gamma Convergence

2.3. Ratio Convergence

Chapter 3
Supervised Learning

Part II.

Prints

Print P1
Continuum limit of Lipschitz learning on graphs

T. Roith and L. Bungert
Foundations of Computational Mathematics (2022)

Foundations of Computational Mathematics
https://doi.org/10.1007/s10208-022-09557-9

FOUNDATIONS OF COMPUTATIONAL MATHEMATICS
The Journal of the Society for the Foundations of Computational Mathematics

Continuum Limit of Lipschitz Learning on Graphs

Tim Roith[1] · **Leon Bungert**[2]

Received: 14 January 2021 / Revised: 29 November 2021 / Accepted: 9 December 2021

Abstract

Tackling semi-supervised learning problems with graph-based methods has become a trend in recent years since graphs can represent all kinds of data and provide a suitable framework for studying continuum limits, for example, of differential operators. A popular strategy here is p-Laplacian learning, which poses a smoothness condition on the sought inference function on the set of unlabeled data. For $p < \infty$ continuum limits of this approach were studied using tools from Γ-convergence. For the case $p = \infty$, which is referred to as Lipschitz learning, continuum limits of the related infinity Laplacian equation were studied using the concept of viscosity solutions. In this work, we prove continuum limits of Lipschitz learning using Γ-convergence. In particular, we define a sequence of functionals which approximate the largest local Lipschitz constant of a graph function and prove Γ-convergence in the L^∞-topology to the supremum norm of the gradient as the graph becomes denser. Furthermore, we show compactness of the functionals which implies convergence of minimizers. In our analysis we allow a varying set of labeled data which converges to a general closed set in the Hausdorff distance. We apply our results to nonlinear ground states, i.e., minimizers with constrained L^p-norm, and, as a by-product, prove convergence of graph distance functions to geodesic distance functions.

Keywords Lipschitz learning · Graph-based semi-supervised learning · Continuum limit · Gamma-convergence · Ground states · Distance functions

Communicated by Alan Edelman.

✉ Leon Bungert
leon.bungert@hcm.uni-bonn.de

Tim Roith
tim.roith@fau.de

[1] Department Mathematik, Friedrich-Alexander Universität Erlangen-Nürnberg, Cauerstraße 11, 91058 Erlangen, Germany

[2] Hausdorff Center for Mathematics, Rheinische Friedrich-Wilhelms-Universität Bonn, Endenicher Allee 62, Villa Maria, 53115 Bonn, Germany

Published online: 21 January 2022

Mathematics Subject Classification 35J20 · 35R02 · 65N12 · 68T05

1 Introduction

Several works in mathematical data science and machine learning have proven the importance of semi-supervised learning as an essential tool for data analysis, see [17,38–41]. Many classification tasks and problems in image analysis (see, e.g., [17] for an overview) traditionally require an expert examining the data by hand, and this so-called labeling process is often a time-consuming and expensive task. In contrast, one typically faces an abundance of unlabeled data which one would also like to equip with suitable labels. This is the key goal of the semi-supervised learning problem which mathematically can be formulated as the extension of a labeling function

$$g : \mathcal{O} \to \mathbb{R}$$

onto the whole data set $V := \mathcal{V} \cup \mathcal{O}$, where $\mathcal{O}$ denotes the set of labeled and $\mathcal{V}$ the set of unlabeled data. In most cases, the underlying data can be represented as a finite weighted graph (V, ω)—composed of vertices V and a weight function ω assigning similarity values to pairs of vertices—which provides a convenient mathematical framework. A popular method to generate a unique extension of the labeling function to the whole data set is so- called *p-Laplacian regularization*, which can be formulated as minimization task

$$\min_{\mathbf{u}:V\to\mathbb{R}} \sum_{x,y\in V} \omega(x,y)^p \, |\mathbf{u}(x) - \mathbf{u}(y)|^p \,, \quad \text{subject to } \mathbf{u} = g \text{ on } \mathcal{O}, \tag{1}$$

over all graph functions $\mathbf{u} : V \to \mathbb{R}$ subject to a constraint given by the labels on $\mathcal{O}$, see, e.g., [2,22,35,38]. This method is equivalent to solving the p-Laplacian partial differential equations on graphs [19] and therewith introduces a certain amount of smoothness of the labeling function. Furthermore, continuum limits of this model as the number of unlabeled data tends to infinity were studied using tools from Γ-convergence [22,35,36] and PDEs [14–16] (see Sect. 1.2 for more details).

Still, p-Laplacian regularization comes with the drawback that it is ill-posed if p is smaller than the ambient space dimension in the sense that the obtained solutions tend to be an average of the label values rather than properly incorporating the information. Extensive studies of this problem were carried out in [35,36]. To overcome this degeneracy, there are several options: In [16] it was investigated at which rates the number of labeled data has to grow to obtain a well-posed problem for $p = 2$ in (1). In [15] it was suggested to replace the pointwise constraint $\mathbf{u} = g$ on $\mathcal{O}$ with measure-valued source terms for the graph Laplacian equation. In contrast, in [2] the authors propose to consider the p-Laplacian regularization for large p. In order to have well-posedness for general space dimensions, one therefore considers the limit $p \to \infty$ which leads to the so-called Lipschitz learning problem

$$\min_{\mathbf{u}:V\to\mathbb{R}} \max_{x,y\in V} \omega(x,y) \, |\mathbf{u}(x) - \mathbf{u}(y)| \quad \text{subject to } \mathbf{u} = g \text{ on } \mathcal{O}. \tag{2}$$

While in the case $p < \infty$ one has the unique existence of solutions and equivalence of the p-Laplacian PDE and the energy minimization task, these properties are lost in the case $p = \infty$. One distinguished continuum model—in the sense that it admits unique solutions—connected to this problem is *absolutely minimizing Lipschitz extensions* and the associated *infinity Laplacian equation* (see, e.g., [3–5,20,27,34]). Using the concept of viscosity solutions, in [14] a convergence result on continuum limits for the infinity Laplacian equation on the flat torus was established, see again Sect. 1.2 for more details. Still, in [30] the authors suggest that other Lipschitz extensions (next to the absolutely minimizing) are indeed relevant for machine learning tasks but a rigorous continuum limit for general Lipschitz extensions has been pending.

The main goal of this paper is to derive a continuum limit for the Lipschitz learning problems (2) to which end we prove Γ-convergence and compactness of the functional in (2). We investigate novel smoothness conditions on the underlying domain which are special for this L^∞-variational problem and originate from the discrepancy between the maximum local Lipschitz constant and the global one. We apply our results to minimizers of a Rayleigh quotient involving the L^∞-norm of the gradient as first examined in [13]. The concrete outline of this paper can be found in Sect. 1.3.

1.1 Assumptions and Main Result

Let $\Omega \subset \mathbb{R}^d$, $d \in \mathbb{N}$, be an open and bounded domain, and let $\Omega_n \subset \overline{\Omega}$ for $n \in \mathbb{N}$ denote a sequence of finite subsets. For each $n \in \mathbb{N}$ we consider the *finite weighted graph* (Ω_n, ω_n), where $\omega_n : \Omega_n \times \Omega_n \to [0, \infty)$ is a weighting function which in our context is given as

$$\omega_n(x, y) := \eta_{s_n}(|x - y|) := \eta(|x - y| / s_n).$$

Here $\eta : [0, \infty) \to [0, \infty)$ denotes the *kernel* and $s_n > 0$ the scaling parameter. The edge set of the graph is implicitly characterized via the weighting function, i.e., for $x, y \in \Omega_n$ we have

$$(x, y) \text{ is an edge iff } \omega_n(x, y) > 0.$$

In the following we state standard assumptions on the kernel function η, see, e.g., [14,22,35,36],

(K1) η is positive and continuous at 0,
(K2) η is non-increasing,
(K3) $\operatorname{supp}(\eta) \subset [0, c_\eta]$ for some $c_\eta > 0$.

Similar to [22] we define the value $\sigma_\eta := \operatorname{ess\,sup}_{t \geq 0} \{\eta(t)\, t\}$ which is a positive number and appears in the Γ-limit.

To incorporate constraints, for each n we denote by $\mathcal{O}_n \subset \Omega_n$ the set of labeled vertices as shown in (2). Often this set is fixed and therefore independent of n (e.g., [14,22,35]; however, see also [15,16] for p-Laplacian learning models with varying constraint sets). As we see in Lemmas 5 and 6, making this assumption is not necessary

in our case. We only require that the sets $\mathcal{O}_n$ converge to some closed set $\mathcal{O} \subset \overline{\Omega}$ in the Hausdorff distance sufficiently fast, i.e.,

$$d_H(\mathcal{O}_n, \mathcal{O}) := \max\left(\sup_{x\in\mathcal{O}} \inf_{y\in\mathcal{O}_n} |x-y|, \sup_{y\in\mathcal{O}_n} \inf_{x\in\mathcal{O}} |x-y|\right) = o(s_n), \quad n\to\infty, \tag{3}$$

where s_n denotes the spatial scaling of the kernel, introduced above. A prototypical example for the constraint set is $\mathcal{O} = \partial\Omega$ which corresponds to the problem of extending Lipschitz continuous boundary values g from $\partial\Omega$ to Ω. Considering a labeling function $g : \overline{\Omega} \to \mathbb{R}$ which is Lipschitz continuous allows us to restrict it to the finite set of vertices $\mathcal{O}_n$. The target functional for the discrete case has the form

$$E_n(\mathbf{u}) = \frac{1}{s_n} \max_{x,y\in\Omega_n} \left\{\eta_{s_n}(|x-y|)\,|\mathbf{u}(x)-\mathbf{u}(y)|\right\} \tag{4}$$

for a function $\mathbf{u} : \Omega_n \to \mathbb{R}$.

Additionally we define the constrained version of the functional, which incorporates the labeling function, as follows

$$E_{n,\mathrm{cons}}(\mathbf{u}) = \begin{cases} E_n(\mathbf{u}) & \text{if } \mathbf{u} = g \text{ on } \mathcal{O}_n, \\ \infty & \text{else.} \end{cases}$$

A typical problem in this context of continuum limits is to find a single metric space in which the convergence of these functionals takes place. In our case we choose the normed space $L^\infty(\Omega)$ and thus need to extend the functionals E_n to $L^\infty(\Omega)$. This can be achieved by employing the well-established technique (see, e.g., [23]) of only considering piecewise constant functions. To this end we let p_n denote a closest point projection, i.e., a map $p_n : \Omega \to \Omega_n$ such that

$$p_n(x) \in \arg\min_{y\in\Omega_n} |x-y|$$

for each $x \in \Omega$. While p_n is not necessarily uniquely determined, this ambiguity is not relevant for our analysis. There, it is only important to control the value of $|p_n(x)-x|$ which is independent of the choice of p_n. This map has already been employed in [14] and it allows us to transform a graph function $\mathbf{u} : \Omega_n \to \mathbb{R}$ to a piecewise constant function, by considering $\mathbf{u} \circ p_n \in L^\infty(\Omega)$. This function is constant on each so-called Voronoi cell $\mathrm{int}(p_n^{-1}(y))$ for $y \in \Omega_n$. This procedure is similar to the technique proposed in [22], where an optimal transport map $T_n : \Omega \to \Omega_n$ is used for turning graph functions into continuum functions. Now we can extend the functional E_n and $E_{n,\mathrm{cons}}$ to arbitrary functions $u \in L^\infty(\Omega)$ by defining with a slight abuse of notation

$$E_n(u) := \begin{cases} E_n(\mathbf{u}) & \text{if } \exists \mathbf{u} : \Omega_n \to \mathbb{R} : u = \mathbf{u} \circ p_n, \\ \infty & \text{else.} \end{cases} \tag{5}$$

$$E_{n,\mathrm{cons}}(u) := \begin{cases} E_{n,\mathrm{cons}}(\mathbf{u}) & \text{if } \exists \mathbf{u} : \Omega_n \to \mathbb{R} : u = \mathbf{u} \circ p_n, \\ \infty & \text{else.} \end{cases} \tag{6}$$

In order to control how the discrete sets Ω_n fill out the domain Ω, we consider the value

$$r_n := \sup_{x \in \Omega} \min_{y \in \Omega_n} |x - y| , \tag{7}$$

and require that it tends faster to zero than the scaling s_n, namely, we assume that

$$\frac{r_n}{s_n} \longrightarrow 0, \quad n \to \infty. \tag{8}$$

We are interested in the case that $(s_n)_{n \in \mathbb{N}} \subset (0, \infty)$ is a *null sequence* meaning that $s_n \to 0$ for $n \to \infty$.

Remark 1 In the context of continuum limits one often employs random geometric graphs, where the discrete sets are obtained as a sequence of points that are i.i.d. w.r.t. a probability distribution $\mu \in \mathcal{P}(\overline{\Omega})$. Typically there is no need to use a probabilistic framework in the L^∞ context, since in contrast to the graph p-Dirichlet energy (13), which is a Monte Carlo approximation of an integral functional, the corresponding discrete Lipschitz energy (4) approximates an essential supremum. Therefore, only the support of a probability measure enters our problem. Similar observations are made in [14,31] where the value r_n is also employed to control the discrete sets Ω_n.

The Γ-limit of the discrete functionals E_n turns out to be a constant multiple of the following continuum functional

$$\mathcal{E}(u) := \begin{cases} \operatorname{ess\,sup}_{x \in \Omega} \ |\nabla u(x)| & \text{if } u \in W^{1,\infty}(\Omega), \\ \infty & \text{else.} \end{cases} \tag{9}$$

A constrained version of this functional can be defined analogously

$$\mathcal{E}_{\mathrm{cons}}(u) := \begin{cases} \mathcal{E}(u) & \text{if } u \in W^{1,\infty}(\Omega) \text{ and } u = g \text{ on } \mathcal{O}, \\ \infty & \text{else.} \end{cases} \tag{10}$$

Before stating our main results we need to introduce a final assumption on the domain Ω which is necessary because of the discrepancy of the Lipschitz constant and the supremal norm of the gradient of functions on non-convex sets. For this we introduce the geodesic distance, induced on Ω by the Euclidean distance, which is defined as

$$d_\Omega(x, y) = \inf \{\mathrm{len}(\gamma) : \gamma : [a, b] \to \Omega \text{ is a curve with } \gamma(a) = x, \gamma(b) = y\},$$

where the length of a curve is given by

$$\operatorname{len}(\gamma) := \sup\left\{\sum_{i=0}^{N-1} |\gamma(t_{i+1}) - \gamma(t_i)| \,:\, N \in \mathbb{N},\, a = t_0 \le t_1 \le \cdots \le t_N = b\right\},$$

see, e.g., [10, Prop. 3.2]. While on convex domains it holds $d_\Omega(x, y) = |x - y|$, we only need to assume the weaker condition:

$$\limsup_{\delta \downarrow 0} \left\{ \frac{d_\Omega(x, y)}{|x - y|} \,:\, x, y \in \Omega,\, |x - y| < \delta \right\} \le 1. \tag{11}$$

In Sect. 3 we explore examples of sets which satisfy this condition; however, already at this point we would like to say that it is satisfied, for instance, for convex sets, for sets with smooth boundary, or for sets which locally are diffeomorphic to a convex set. In a nutshell, condition (11) prohibits the presence of internal corners in the boundary. Furthermore, since it holds $d_\Omega(x, y) \ge |x - y|$, condition (11) requires the geodesic and the Euclidean distance to coincide locally.

Main results Our two main results state the discrete-to-continuum Γ-convergence of the functionals $E_{n,\mathrm{cons}}$ to $\sigma_\eta\, \mathcal{E}_{\mathrm{cons}}$ and that sequences of minimizers of the discrete functionals converge to a minimizer of the continuum functional.

Theorem 1 (Discrete to continuum Γ-convergence) *Let $\Omega \subset \mathbb{R}^d$ be a domain satisfying (11), let the kernel fulfill (K1)–(K3), and let the constraint sets $\mathcal{O}_n$, $\mathcal{O}$ satisfy (3), then for any null sequence $(s_n)_{n\in\mathbb{N}} \subset (0, \infty)$ which satisfies the scaling condition (8) we have*

$$E_{n,\mathrm{cons}} \xrightarrow{\Gamma} \sigma_\eta\, \mathcal{E}_{\mathrm{cons}}. \tag{12}$$

Theorem 2 (Convergence of Minimizers) *Let $\Omega \subset \mathbb{R}^d$ be a domain satisfying (11), let the kernel fulfill (K1)–(K3), let the constraint sets $\mathcal{O}_n$, $\mathcal{O}$ satisfy (3), and $(s_n)_{n\in\mathbb{N}} \subset (0, \infty)$ be a null sequence which satisfies the scaling condition (8). Then any sequence $(u_n)_{n\in\mathbb{N}} \subset L^\infty(\Omega)$ such that*

$$\lim_{n\to\infty} \left(E_{n,\mathrm{cons}}(u_n) - \inf_{u\in L^\infty(\Omega)} E_{n,\mathrm{cons}}(u) \right) = 0$$

is relatively compact in $L^\infty(\Omega)$ and

$$\lim_{n\to\infty} E_{n,\mathrm{cons}}(u_n) = \min_{u\in L^\infty(\Omega)} \sigma_\eta\, \mathcal{E}_{\mathrm{cons}}(u).$$

Furthermore, every cluster point of $(u_n)_{n\in\mathbb{N}}$ is a minimizer of $\mathcal{E}_{\mathrm{cons}}$.

1.2 Related Work

The first studies concerning the limit behavior of difference operators on random graphs were carried out in [33] and the follow-up work [32], which considers the consistency of spectral clustering on graphs. The main motivation of our paper is the works [22,35] where the Γ-convergence of the discrete functionals

$$E_n^{(p)}(\mathbf{u}) = \frac{1}{s_n^p n^2} \sum_{x,y \in \Omega_n} \eta_{s_n}(|x-y|)\, |\mathbf{u}(x) - \mathbf{u}(y)|^p \tag{13}$$

toward a continuum functional of the form

$$\mathcal{E}^{(p)}(u) = \sigma_\eta \int_\Omega |\nabla u|^p \, \rho^2(x)\, \mathrm{d}x,$$

for $1 \leq p < \infty$ and u smooth enough, is shown. Here, ρ is the density of the measure μ according to which the points x_i are distributed. The Γ-convergence is considered w.r.t. the TL^p topology, which allows to compare discrete functions $\mathbf{u} \in L^p(\nu_n)$ with continuum functions $u \in L^p(\mu)$ via an optimal transport ansatz. Here, ν_n denotes the empirical measure for the set Ω_n, see (21). In particular, the problem they study is connected to the extension task (1) by considering the constrained functional

$$E_{n,\mathrm{cons}}^{(p)}(\mathbf{u}) := \begin{cases} E_n^{(p)}(\mathbf{u}) & \text{if } \mathbf{u}(x_i) = g(x_i) \text{ for } x_i \in \mathcal{O}, \\ \infty & \text{else,} \end{cases}$$

and the associated minimization problem

$$\inf_{\mathbf{u} \in L^p(\nu_n)} E_{n,\mathrm{cons}}^{(p)}(\mathbf{u}). \tag{14}$$

Here, the constraint set $\mathcal{O}$ is assumed to be a fixed collection of finitely many points. The main result they show ensures that minimizers of the functionals $E_{n,\mathrm{cons}}^{(p)}$ converge uniformly toward minimizers of a respective constrained version of $\mathcal{E}^{(p)}$ under appropriate assumptions on the scaling s_n. The motivation for considering the limit $p \to \infty$ for above problems is given in [2], where the graph p-Laplacian for $p < d$ is noticed to have undesirable properties, namely the solution of problem (14) tends to equal a constant at the majority of the vertices with sharp spikes at the labeled data. Using a suitable scaling, this problem does not occur in the case $p > d$ (which is intuitively connected to the Sobolev embedding $W^{1,p}(\Omega) \hookrightarrow C^{0,1-d/p}(\overline{\Omega})$ for $p > d$ see [1, Ch. 4]) and, in particular, as pointed out in [30, Sec. 3] one generally hopes for better interpolation properties in the case $p \to \infty$. However, in this limit the interpretation of the measure μ and its density changes, namely only its support enters the problem. The functional $\mathcal{E}^{(p)}$ incorporates information about the distribution of the data in the form of the density ρ. Using standard properties of L^p-norms, the limit $p \to \infty$ on

the other hand reads

$$\mu\text{-ess sup}_{x\in\Omega}\,|\nabla u| = \lim_{p\to\infty}\left(\mathcal{E}^{(p)}(u)\right)^{1/p}$$

and thus only the support of the measure μ is relevant. This phenomenon was already observed in [2,14] which the authors informally described as the limit 'forgetting' the distribution of the data. Furthermore, this observation is consistent with our results, since the limit $n \to \infty$ is independent of the sequence $(x_i)_{i\in\mathbb{N}} \subset \Omega$ as long as it fills out the domain Ω sufficiently fast, see Theorem 1 for the precise condition. In the case $p < \infty$ the minimization task (1) is equivalent to the graph p-Laplace equation, for which the formal limit $p \to \infty$ leads to the graph infinity Laplace equation

$$\begin{aligned} \Delta_\infty \mathbf{u} &= 0 \text{ in } V\backslash\mathcal{O}, \\ \mathbf{u} &= g \text{ in } \mathcal{O}, \end{aligned} \tag{15}$$

where the operator Δ_∞ on Ω_n is defined as

$$\Delta_\infty \mathbf{u}(x) := \max_{y\in\Omega_n} \omega(x,y)(\mathbf{u}(y)-\mathbf{u}(x)) + \min_{y\in\Omega_n} \omega(x,y)(\mathbf{u}(y)-\mathbf{u}(x)).$$

One should note that the unique solution of (15) also solves the Lipschitz learning task (2), see, e.g., [14,30]. A first study concerning the continuum limit of this problem is carried out in [14]. The main result therein states that the solutions of the discrete problems converge uniformly to the viscosity solution of the continuum equation,

$$\begin{aligned} \Delta_\infty u &= 0 \text{ in } \Omega, \\ u &= g \text{ on } \mathcal{O}, \end{aligned} \tag{16}$$

where the continuum infinity Laplacian for a smooth function u is defined as

$$\Delta_\infty u := \langle \nabla u, D^2 u \nabla u\rangle = \sum_{i,j=1}^{d} \partial_i u\, \partial_j u\, \partial^2_{ij} u. \tag{17}$$

The considered domain is the flat torus, i.e., $\Omega = \mathbb{R}^d\backslash\mathbb{Z}^d$ and again the constraint set $\mathcal{O}$ is assumed to be fixed and finite. Furthermore, the only requirement on the sequence of points $\Omega_n \subset \Omega$ is characterized by the value r_n defined in (7).

Theorem 3 [14, Thm. 2.1] *Consider the flat torus $\Omega = \mathbb{R}^d\backslash\mathbb{Z}^d$, let the kernel $\eta \in C^2([0,\infty))$ be such that*

$$\begin{cases} \eta(s) \geq 1 & \text{for } s \leq 1, \\ \eta(s) = 0 & \text{for } s \geq 2, \end{cases}$$

then for a null sequence $(s_n)_{n\in\mathbb{N}} \subset (0,\infty)$ *such that*

$$\frac{r_n^2}{s_n^3} \longrightarrow 0 \tag{18}$$

and for the sequence $\mathbf{u}_n$ *of solutions to problem* (15) *we have that* $\mathbf{u}_n \to u$ *uniformly, where* $u \in C^{0,1}(\Omega)$ *is the unique viscosity solution of the infinity Laplace equation* (16) *on* Ω*, i.e.,*

$$\lim_{n\to\infty} \max_{x\in\Omega_n} |\mathbf{u}_n(x) - u(x)| = 0.$$

Remark 2 We state this result, since it provides the first continuum limit of the infinity Laplace equation (15) on general graphs. Solutions of this problem constitute a special subclass of minimizers in the Lipschitz learning task (2). To show that the limit of solutions of (15) solve the continuum PDE (16) the author in [14] utilizes a consistency argument which requires smoothness of the kernel η and the relatively strict scaling condition (18). In contrast, our results consider the general minimization problem (2), which allows us to work with the weakest scaling condition possible (8)—namely that the graph is asymptotically connected—and much weaker conditions on the kernel η.

Remark 3 *(Convergence Rates)* Note that [14] did not establish rates of convergence for solutions of the graph infinity Laplacian equation (15) and neither do we for general minimizers of (2). Indeed, Γ-convergence is not a good tool for proving quantitative rates since it is a very indirect notion of convergence (see also [35,36] which do not establish convergence rates either). At the same time Γ-convergence typically allows for much less restrictive conditions on the graph scaling than PDE techniques, cf. Remark 2. Still, in the recent work [12] we successfully used comparison principle techniques to show rates of convergence for solutions of the graph infinity Laplacian equation (15) in a much more general setting than the one considered in [14]. For showing this it suffices to use quantitative versions of the weak scaling assumption (8) and the domain regularity condition (11).

Since the solutions of the continuum PDE (16) only exist in the viscosity sense, the proof of this theorem involves viscosity techniques. The arguments are therefore fundamentally different to the results for the case $p < \infty$ in [36]. This is our motivation to use Γ-convergence also for $p = \infty$.

1.3 Outline

In Sect. 2 we give an overview of the concepts of Γ-convergence and the closest point projection. In particular, we derive a transformation rule for supremal functionals which is the analogue of the well-known integral transformation rule for the change of variables.

Section 3 is devoted to the proofs of our main results Theorems 1 and 2. Similar to the strategy in [22], in Sect. 3.1 we first prove Γ-convergence of the non-local

auxiliary functionals

$$\mathcal{E}_s(u) := \frac{1}{s}\ \operatorname{ess\,sup}_{x,y\in\Omega} \{\eta_s(|x-y|)\,|u(x)-u(y)|\}\,, \quad s>0 \tag{19}$$

which mimic the non-local structure of the discrete functionals E_n in (4), to the continuum functional $\mathcal{E}$ in (9). Subsequently, in Sect. 3.2 we use this result for proving our first main result, discrete to continuum Γ-convergence of the constrained discrete functionals $E_{n,\mathrm{cons}}$. In Sect. 4 we prove compactness of the discrete functionals which yields our second main result, the convergence of minimizers.

In Sect. 5 we apply our results to a nonlinear eigenvalue problem and prove convergence of discrete ground states to continuum ground states. Furthermore, generalizing the results in [13], we characterize the latter as geodesic distance function to the constraint set $\mathcal{O}$.

2 Mathematical Background

This section reviews two important mathematical tools which we use in this paper. The first one is the concept of Γ-convergence, which allows to deduce convergence of minimizers from convergence of functionals. The second concept, entirely unrelated to Γ-convergence, is the closest point projection which we employ in order to turn graph functions into continuum ones. Furthermore, we derive a supremal version of the transformation rule.

2.1 Γ-Convergence

In this section we introduce a convergence concept that is frequently employed in the theory of variational problems, namely the so-called Γ-convergence. We refer to [8] for a detailed introduction.

Definition 1 $(\Gamma-convergence)$ Let X be a metric space and let $F_n : X \to [-\infty,\infty]$ be a sequence of functionals. We say that F_n Γ-converges to the functional $F : X \to [-\infty,\infty]$ if

(i) **(liminf inequality)** for every sequence $(x_n)_{n\in\mathbb{N}} \subset X$ converging to $x \in X$ we have that

$$\liminf_{n\to\infty} F_n(x_n) \geq F(x);$$

(ii) **(limsup inequality)** for every $x \in X$ there exists a sequence $(x_n)_{n\in\mathbb{N}} \subset X$ converging to x and

$$\limsup_{n\to\infty} F_n(x_n) \leq F(x).$$

The notion of Γ-convergence is especially useful, since it implies the convergence of minimizers under additional compactness assumptions. For convenience we prove the respective result below, the proof is an adaption of a similar result in [8, Thm. 1.21].

Lemma 1 (Convergence of Minimizers) *Let X be a metric space and $F_n : X \to [0, \infty]$ a sequence of functionals Γ-converging to $F \to X : [0, \infty]$ which is not identically $+\infty$. If there exists a relatively compact sequence $(x_n)_{n\in\mathbb{N}}$ such that*

$$\lim_{n\to\infty} \left(F_n(x_n) - \inf_{x\in X} F_n(x) \right) = 0,$$

then we have that

$$\lim_{n\to\infty} \inf_{x\in X} F_n(x) = \min_{x\in X} F(x)$$

and any cluster point of $(x_n)_{n\in\mathbb{N}}$ is a minimizer of F.

Proof Using the Γ-convergence of F_n for any $y \in X$ we can find a sequence $(y_n)_{n\in\mathbb{N}} \subset X$ such that

$$\limsup_{n\to\infty} F_n(x_n) = \limsup_{n\to\infty} \inf_{x\in X} F_n(x) \leq \limsup_{n\to\infty} F_n(y_n) \leq F(y)$$

and thus

$$\limsup_{n\to\infty} F_n(x_n) \leq \inf_{x\in X} F(x) < \infty, \tag{20}$$

where for the last inequality we use the fact that F is not identically $+\infty$. By assumption the sequence $(x_n)_{n\in\mathbb{N}}$ is relatively compact, therefore we can find an element $x \in X$ and a subsequence such that $x_{n_k} \to x$, for which the liminf inequality yields

$$F(x) \leq \liminf_{n\to\infty} F_n(\tilde{x}_n) \leq \liminf_{k\to\infty} F_{n_k}(x_{n_k}) \leq \limsup_{n\to\infty} F_n(x_n),$$

where we employ the sequence

$$\tilde{x}_n := \begin{cases} x_{n_k} & \text{if } n = n_k, \\ x & \text{else.} \end{cases}$$

Together with (20) we have that x is a minimizer of F and $\lim_{n\to\infty} F_n(x_n) = F(x)$. Since the above reasoning works for any subsequence converging to some element in X we have that every cluster point is a minimizer. □

A condition that ensures the existence of a relatively compact sequence of minimizers is the so-called *compactness* property for functionals. A sequence of functionals is called compact if for any sequence $(x_n)_{n\in\mathbb{N}} \subset X$ the property

$$\sup_{n\in\mathbb{N}} F_n(x_n) < \infty$$

implies that $(x_n)_{n\in\mathbb{N}}$ is relatively compact. In Sect. 4 we show that the constrained functionals $E_{n,\mathrm{cons}}$ fulfill the compactness property. This strategy is standard in the context of continuum limits and has already been employed in [22,36].

2.2 The Closest Point Projection

In [22,36] a map $T_n : \Omega \to \Omega_n$ is employed in order to transform integrals w.r.t. the empirical measure, defined as

$$\nu_n(A) := \frac{1}{|\Omega_n|} \sum_{x\in\Omega_n} \delta_x(A), \quad A \in \mathcal{B}(\Omega), \tag{21}$$

into integrals w.r.t. a probability measure $\nu \in \mathcal{P}(\overline{\Omega})$. Here, $\mathcal{B}(\Omega)$ denotes the Borel σ-algebra and $\mathcal{P}(\overline{\Omega})$ is the set of probability measures on Ω. One assumes the push-forward condition

$$\nu_n = T_{n\#}\nu = \nu \circ T_n^{-1},$$

which yields the following transformation,

$$\sum_{x\in\Omega_n} \mathbf{u}(x) = \int_\Omega \mathbf{u}(x)\, \mathrm{d}\nu_n(x) = \int_\Omega \mathbf{u}(T_n(x))\, \mathrm{d}\nu(x),$$

for a function $\mathbf{u} : \Omega_n \to \mathbb{R}$, see, for example, [6]. Informally speaking the push-forward condition manifests the intuition that the map T_n has to preserve the weighting imposed by the empirical measure ν_n. However, the supremal functionals in our case only take into account whether a respective set has positive measure or is a null set. Therefore, the assumptions on the map T_n can be weakened for an analogous transformation rule. In fact, we only need that the push-forward measure is equivalent to the original one.

Lemma 2 *For two probability measures $\mu, \nu \in \mathcal{P}(\overline{\Omega})$, a measurable map $T : \Omega \to \Omega$ which fulfills*

(i) $\nu << T_\#\mu$,
(ii) $T_\#\mu << \nu$,

and for a measurable function $f : \Omega \to \mathbb{R}$ we have that

$$\nu\text{-ess sup}_{x\in\Omega}\, f(x) = \mu\text{-ess sup}_{y\in\Omega}\, f(T(y)).$$

Remark 4 In the case $\nu = \nu_n$ we observe that assumption (i) is equivalent to

$$\mu(T^{-1}(x)) > 0 \tag{22}$$

for all $x \in \Omega_n$. Furthermore, assumption (ii) is a generalization of the property that $T(\Omega) \subset \Omega_n$. If (i) and (ii) are fulfilled we call the measures ν and $T_\#\mu$ *equivalent*.

Additionally, the statement still holds true for a finite measure μ and a general measure ν. However, for our application it suffices to consider probability measures.

Proof First we consider a set $A \in \mathcal{B}(\Omega)$ such that $\nu(A) = 0$. For this we have that

$$\sup_{x\in\Omega\setminus A} f(x) \geq \sup_{y\in T^{-1}(\Omega\setminus A)} f(T(y)) = \sup_{y\in\Omega\setminus T^{-1}(A)} f(T(y)) = (\#)$$

and since $\nu(A) = 0$ we can use (ii) to infer that $\mu(T^{-1}(A)) = 0$. This implies that

$$(\#) \geq \inf_{\mu(B)=0} \sup_{y\in\Omega\setminus B} f(T(y)) = \mu\text{-ess sup}_{y\in\Omega} f(T(y)).$$

The null set A was arbitrary and thus taking the infimum over all ν-null sets we obtain

$$\nu\text{-ess sup}_{x\in\Omega} f(x) \geq \mu\text{-ess sup}_{x\in\Omega} f(T(y)).$$

On the other hand take $B \in \mathcal{B}(\Omega)$ such that $\mu(B) = 0$ then

$$\sup_{y\in\Omega\setminus B} f(T(y)) = \sup_{x\in T(\Omega\setminus B)} f(x)$$

and since $T^{-1}(T(\Omega\setminus B)) \supset \Omega\setminus B$ we have that

$$\begin{aligned} 1 \geq& \mu(T^{-1}(T(\Omega\setminus B))) \geq \mu(\Omega\setminus B) = 1 \\ \Rightarrow& \mu(\Omega\setminus T^{-1}(T(\Omega\setminus B))) = 0 \\ \Rightarrow& \nu(\Omega\setminus(T(\Omega\setminus B))) = 0. \end{aligned}$$

This implies that

$$\sup_{x\in T(\Omega\setminus B)} f(x) \geq \inf_{\nu(A)=0} \sup_{x\in\Omega\setminus A} f(x) = \nu\text{-ess sup}_{x\in\Omega} f(x).$$

Taking the infimum overall μ-null sets completes the proof. □

An important type of mapping T_n in our context is the so-called closest point projection.

Definition 2 (*Closest Point Projection*) For a finite set of points $\Omega_n \subset \Omega$ a map $p_n : \Omega \to \Omega_n$ is called *closest point projection* if

$$p_n(x) \in \arg\min_{y\in\Omega_n} |x - y|$$

for each $x \in \Omega$.

Remark 5 Recalling the standard definition of a Voronoi tessellation (see, e.g., [29]) one notices that the control volume associated with the vertex $x_i \in \Omega_n$ is given by $\text{int}(p_n^{-1}(x_i))$.

The use of a closest point projection is very natural for L^∞-type scenarios and has, for example, already been employed in [14] for a similar problem. In particular, we can see that $\lambda^d(p_n^{-1}(x_i)) > 0$ for every vertex $x_i \in \Omega_n$ and thus $\nu_n << p_{n\#}\lambda^d$, where λ^d denotes the d-dimensional Lebesgue measure. The second condition $p_{n\#}\lambda^d << \nu_n$ follows directly from the definition of the map p_n and thus the conditions for Lemma 2 are fulfilled. In fact, for each function $u \in L^\infty(\Omega)$ such that $u = \mathbf{u} \circ p_n$ for some $\mathbf{u} : \Omega_n \to \mathbb{R}$ we can employ Lemma 2 to reformulate the extension (5) of the discrete functional as follows,

$$\begin{aligned} E_n(u) &= E_n(\mathbf{u}) \\ &= \frac{1}{s_n} \max_{x,y\in\Omega_n} \eta_{s_n}(|x-y|)\,|\mathbf{u}(x)-\mathbf{u}(y)| \\ &= \frac{1}{s_n}\nu_n\text{-ess sup}_{x,y\in\Omega}\,\eta_{s_n}(|x-y|)\,|\mathbf{u}(x)-\mathbf{u}(y)| \\ &= \frac{1}{s_n}\,\text{ess sup}_{x,y\in\Omega}\,\eta_{s_n}(|p_n(x)-p_n(y)|)\,|u(x)-u(y)|\,. \end{aligned} \tag{23}$$

Note that the weights consider the distance between the nearest vertices to x and y, respectively, and not the Euclidean distance between x and y. This observation is important for the estimate in Sect. 3.2.

3 Γ-Convergence of Lipschitz Functionals

3.1 Non-local to Local Convergence

In this section we show the Γ-convergence of the non-local functionals (19) to the continuum functional defined in (9) with respect to the L^∞ topology. We first prove the liminf inequality.

Lemma 3 (liminf inequality) *Let $\Omega \subset \mathbb{R}^d$ be an open domain and let the kernel fulfill (K1)–(K3), then for a null sequence $(s_n)_{n\in\mathbb{N}} \subset (0,\infty)$ we have*

$$\liminf_{n\to\infty} \mathcal{E}_{s_n}(u_n) \geq \sigma_\eta\, \mathcal{E}(u) \tag{24}$$

for every sequence $(u_n)_{n\in\mathbb{N}} \subset L^\infty(\Omega)$ converging to $u \in L^\infty(\Omega)$ in $L^\infty(\Omega)$.

Proof We assume w.l.o.g. that

$$\liminf_{n\to\infty} \mathcal{E}_{s_n}(u_n) < \infty. \tag{25}$$

We choose a vector $h \in \mathbb{R}^d$ and estimate the supremum over $x, y \in \Omega$ by a supremum over a difference quotient, namely

$$\begin{aligned}\mathcal{E}_{s_n}(u_n) &= \operatorname{ess\,sup}_{x,y\in\Omega} \eta_{s_n}(|x-y|)\ \frac{|u_n(x)-u_n(y)|}{s_n}\\ &\geq \eta(|h|)\ \operatorname{ess\,sup}_{x\in\Omega} \frac{|u_n(x)-u_n(x+s_nh)|}{s_n}\ \mathbb{I}_\Omega(x+s_nh).\end{aligned}$$

In the above transformation we ensured to not enlarge the supremum by multiplying by the indicator function. Considering the function

$$v_n^h(x) := \frac{u_n(x)-u_n(x+s_nh)}{s_n}\ \mathbb{I}_\Omega(x+s_nh)$$

for $\eta(|h|) > 0$ we have that

$$\liminf_{n\to\infty} \left\| v_n^h \right\|_{L^\infty} \leq C$$

which follows directly form (25). Thus, by the sequential Banach–Alaoglu theorem, the sequence $(v_n^h)_{n\in\mathbb{N}}$ possesses convergent subsequences. For any such subsequence $(v_{n_k}^h)_{k\in\mathbb{N}}$ there exists $v^h \in L^\infty$ such that

$$v_{n_k}^h \rightharpoonup^* v^h$$

in the weak* topology of L^∞, i.e., for every $w \in L^1(\Omega)$ we have

$$\int_\Omega v_{n_k}^h\ w\ \mathrm{d}x \to \int_\Omega v^h\ w\ \mathrm{d}x. \tag{26}$$

We want to identify the function v^h, for which we use a smooth function $\phi \in C_c^\infty(\Omega)$ as the test function in (26). We shift the difference quotient of u_n to a quotient of ϕ and hope to obtain the directional derivative in the limit. Since $\operatorname{supp}(\phi) \subset\subset \Omega$, we can choose n_0 large enough such that

$$x + s_nh \in \Omega$$

for all $n \geq n_0$ and for all $x \in \operatorname{supp}(\phi)$. Therefore, we get

$$\begin{aligned}\int_\Omega v_{n_k}^h(x)\ \phi(x)\ \mathrm{d}x &= \int_\Omega \frac{u_n(x)-u_n(x+s_nh(x))}{s_n}\ \phi(x)\ \mathrm{d}x\\ &= \int_\Omega u_n(x)\ \frac{\phi(x)-\phi(x-s_nh)}{s_n}\ \mathrm{d}x.\end{aligned}$$

Furthermore, for $n \geq n_0$ we have

$$\left| u_n(x)\ \frac{\phi(x)-\phi(x-s_nh)}{s_n} - u(x)\nabla\phi(x)\cdot(-h) \right|$$

$$
\begin{aligned}
&\leq \|u_n - u\|_{L^\infty} \left| \frac{\phi(x) - \phi(x - s_n h)}{s_n} \right| \\
&\quad + \|u\|_{L^\infty} \left| \frac{\phi(x) - \phi(x - s_n h)}{s_n} - \nabla\phi(x) \cdot (-h) \right| \\
&\leq |h| \, \|u_n - u\|_{L^\infty} \, \|\nabla\phi\|_{L^\infty} \\
&\quad + \|u\|_{L^\infty} \left| \frac{\phi(x) - \phi(x - s_n h) + \nabla\phi(x) \cdot (s_n h)}{s_n} \right| \xrightarrow{n\to\infty} 0,
\end{aligned}
$$

since u_n converges to u in L^∞, $\phi \in C_c^\infty$ has a bounded gradient, and since the difference quotient converges to the directional derivative. Besides the pointwise convergence, we also easily obtain the boundedness of the function sequence, since

$$
\begin{aligned}
\left| u_n(x) \, \frac{\phi(x) - \phi(x - s_n h)}{s_n} \right| &\leq |h| \, \|u_n\|_{L^\infty} \;\; \|\nabla\phi\|_{L^\infty} \\
&\leq |h| \, (\|u_n - u\|_{L^\infty} + \|u\|_{L^\infty}) \;\; \|\nabla\phi\|_{L^\infty} ,
\end{aligned}
$$

which is uniformly bounded. Thus, we can apply Lebesgue's convergence theorem to see that

$$
\int_{\mathbb{R}^d} v^h \, \phi \, \mathrm{d}x = \lim_{k\to\infty} \int_\Omega v_{n_k}^h \, \phi \, \mathrm{d}x = - \int_{\mathbb{R}^d} u \, (\nabla\phi \cdot h) \, \mathrm{d}x .
$$

In particular, we can choose $h_i = c\, e_i$, where e_i denotes the ith unit vector and the constant $c > 0$ is small enough to ensure that $\eta(|h_i|) > 0$, to obtain

$$
\int_{\mathbb{R}^d} v^{h_i} \, \phi \, \mathrm{d}x = -c \int_{\mathbb{R}^d} u \, \partial_i \phi \, \mathrm{d}x
$$

for all $i \in \{1, \ldots, d\}$ and all $\phi \in C_c^\infty(\Omega)$. This yields that $u \in W^{1,\infty}(\Omega)$ and again for any h such that $\eta(|h|) > 0$

$$
\int_{\mathbb{R}^d} v^h \, \phi \, \mathrm{d}x = \int_{\mathbb{R}^d} (\nabla u \cdot h) \, \phi \, \mathrm{d}x .
$$

Using the density of $C_c^\infty(\Omega)$ in $L^1(\Omega)$ w.r.t. $\|\cdot\|_{L^1}$ we obtain that

$$
\int_{\mathbb{R}^d} v^h \, w \, \mathrm{d}x = \int_{\mathbb{R}^d} (\nabla u \cdot h) \, w \, \mathrm{d}x
$$

for any $w \in L^1(\Omega)$. Since the limit is independent of the subsequence $v_{n_k}^h$, we obtain that the weak* convergence holds for the whole sequence, i.e., $v_n^h \rightharpoonup^* \nabla u \cdot h$ and thus together with the lower semi-continuity of $\|\cdot\|_{L^\infty}$

$$
\liminf_{n\to\infty} \mathcal{E}_{s_n}(u_n) \geq \eta(|h|) \;\; \liminf_{n\to\infty} \left\| v_n^h \right\|_{L^\infty} \geq \eta(|h|) \;\; \|\nabla u \cdot h\|_{L^\infty} ,
$$

for every $h \in \mathbb{R}^d$ such that $\eta(|h|) > 0$. Since the inequality is trivially true for $\eta(|h|) = 0$ we obtain

$$\liminf_{n\to\infty} \mathcal{E}_{s_n}(u_n) \geq \sup_{h\in\mathbb{R}^d} \eta(|h|)\, \|\nabla u \cdot h\|_{L^\infty}\,.$$

Considering $z \in \Omega$ such that $\nabla u(z)$ exists and satisfies $|\nabla u(z)| > 0$, and taking $t \geq 0$ we have that

$$\sup_{h\in\mathbb{R}^d} \eta(|h|)\ \|\nabla u \cdot h\|_{L^\infty} \geq \eta(t) \left\|\nabla u \cdot t\,\frac{\nabla u(z)}{|\nabla u(z)|}\right\|_{L^\infty} \geq \eta(t)\, t\, |\nabla u(z)|.$$

This inequality holds for every $t \geq 0$ and almost every $z \in \Omega$, since it is again trivially fulfilled if $\nabla u(z)$ exists and is equal to zero. Hence, we obtain

$$\liminf_{n\to\infty} \mathcal{E}_{s_n}(u_n) \geq \sigma_\eta\, \mathcal{E}(u)$$

which concludes the proof. □

We proceed by proving the limsup inequality. The most important fact here is that for $u \in W^{1,\infty}(\Omega)$ and for almost every $x, y \in \Omega$ we have the inequality

$$|u(x) - u(y)| \leq \|\nabla u\|_{L^\infty}\, d_\Omega(x, y), \tag{27}$$

where $d_\Omega(\cdot,\cdot)$ denotes the geodesic distance on Ω, see [9, P. 269]. Since the non-local functional $\mathcal{E}_s$ compares points $x, y \in \Omega$ that are close together w.r.t. the Euclidean distance, we need to asymptotically bound the geodesic distance from above by the Euclidean distance. For this, we assume condition (11), which we repeat here for convenience:

$$\limsup_{\delta\downarrow 0} \left\{ \frac{d_\Omega(x, y)}{|x - y|} \ :\ x, y \in \Omega,\ |x - y| < \delta \right\} \leq 1.$$

Lemma 4 (limsup inequality) *Let $\Omega \subset \mathbb{R}^d$ be a domain satisfying (11), $(s_n)_{n\in\mathbb{N}} \subset (0, \infty)$ a null sequence and let the kernel fulfill (K1)–(K3), then for each $u \in L^\infty(\Omega)$ there exists a sequence $(u_n)_{n\in\mathbb{N}} \subset L^\infty(\Omega)$ converging to u strongly in $L^\infty(\Omega)$ such that*

$$\limsup_{n\to\infty} \mathcal{E}_{s_n}(u_n) \leq \sigma_\eta\, \mathcal{E}(u). \tag{28}$$

Proof If $u \notin W^{1,\infty}$ the inequality holds trivially. If $u \in W^{1,\infty}$ we see that

$$\mathcal{E}_{s_n}(u) = \frac{1}{s_n}\ \operatorname{ess\,sup}_{x,y\in\Omega} \left\{\eta_{s_n}(|x - y|)\, |u(x) - u(y)|\right\}$$

$$\leq \frac{1}{s_n} \operatorname{ess\,sup}_{x,y\in\Omega} \left\{\eta_{s_n}(|x-y|)\, d_\Omega(x,y)\right\} \ \|\nabla u\|_{L^\infty}$$
$$\leq \frac{1}{s_n} \operatorname{ess\,sup}_{x,y\in\Omega} \left\{\eta_{s_n}(|x-y|)\ |x-y| \frac{d_\Omega(x,y)}{|x-y|}\right\} \ \|\nabla u\|_{L^\infty}.$$

By (11), for any $\varepsilon > 0$ we can find $\delta > 0$ such that

$$\frac{d_\Omega(x,y)}{|x-y|} \leq 1+\varepsilon, \quad \forall x, y \in \Omega \ : \ |x-y| < \delta.$$

Choosing $n \in \mathbb{N}$ so large that $c_\eta s_n < \delta$, where c_η is the radius of the kernel η, we obtain

$$\begin{aligned}\mathcal{E}_{s_n}(u) &\leq (1+\varepsilon) \operatorname{ess\,sup}_{x,y\in\Omega} \left\{\eta_{s_n}(|x-y|)\frac{|x-y|}{s_n}\right\} \ \|\nabla u\|_{L^\infty} \\ &= (1+\varepsilon) \operatorname{ess\,sup}_{z\in\mathbb{R}^d} \{\eta(|z|)\,|z|\} \ \|\nabla u\|_{L^\infty} \\ &= (1+\varepsilon)\,\sigma_\eta \ \|\nabla u\|_{L^\infty} = (1+\varepsilon)\,\sigma_\eta\,\mathcal{E}(u).\end{aligned}$$

Since $\varepsilon > 0$ was arbitrary, this shows that the constant sequence $u_n := u$ fulfills the limsup inequality. □

The previous lemmata directly imply the Γ-convergence of the respective functionals, which we state below.

Theorem 4 (Non-local to local Γ-convergence) *Let $\Omega \subset \mathbb{R}^d$ be a domain satisfying (11) and let the kernel fulfill (K1)–(K3), then for any null sequence $(s_n)_{n\in\mathbb{N}} \subset (0,\infty)$ we have that*

$$\mathcal{E}_{s_n} \xrightarrow{\Gamma} \sigma_\eta\,\mathcal{E}. \tag{29}$$

Remark 6 Assumption (11) is not satisfied for general non-convex domains, whereas

$$|u(x)-u(y)| \leq \operatorname{Lip}(u)\,|x-y|$$

is. Hence, one might consider replacing the functional $\mathcal{E}(u) = \sigma_\eta \|\nabla u\|_{L^\infty}$ by $\mathcal{E}(u) = \sigma_\eta \operatorname{Lip}(u)$ which allows to prove the limsup inequality for arbitrary (in particular, non-convex) domains. However, as the following example shows, the liminf inequality is not true for this functional and one has

$$\sigma_\eta \|\nabla u\|_{L^\infty} \leq \liminf_{n\to\infty} \mathcal{E}_{s_n}(u_n) \leq \limsup_{n\to\infty} \mathcal{E}_{s_n}(u_n) \leq \sigma_\eta \operatorname{Lip}(u)$$

where each inequality can be strict.

Example 1 We consider the non-convex domain $\Omega = \{x \in \mathbb{R}^2 : \max(|x_1|, |x_2|) < 1\} \setminus ([0, 1] \times [-1, 0])$ which does not satisfy (11), the function

$$u(x) = \begin{cases} x_1^p & \text{if } x_1, x_2 \geq 0, \\ x_2^p & \text{if } x_1, x_2 \leq 0, \\ 0 & \text{else,} \end{cases}$$

for some power $p \geq 1$ and the kernel $\eta(t) = \chi_{[0,1]}(t)$ for $x \geq 0$. Then one can compute that

$$\|\nabla u\|_{L^\infty} = p, \quad \mathrm{Lip}(u) = \max(\sqrt{2}, p), \quad \lim_{n\to\infty} \mathcal{E}_{s_n}(u) = \begin{cases} \sqrt{2}, & p = 1, \\ p, & p > 1. \end{cases}$$

The case $1 < p < \sqrt{2}$ shows that the liminf inequality $\liminf_{n\to\infty} \mathcal{E}_{s_n}(u_n) \geq \mathrm{Lip}(u)$ is false, in general.

Example 2 (The domain condition (11)) In this example we will study several scenarios where condition (11) is satisfied. Let us first remark that if one fixes $x \in \Omega$ then

$$\limsup_{\delta \downarrow 0} \left\{ \frac{d_\Omega(x, y)}{|x - y|} \ : \ y \in \Omega, \ |x - y| < \delta \right\} \leq 1.$$

is always true since Ω is open. Hence, (11) is in fact a condition on the boundary of the domain.

- If Ω is convex, it holds $d_\Omega(x, y) = |x - y|$ and hence (11) is trivially true.
- If Ω is locally $C^{1,1}$-diffeomorphic to a convex set, then (11) is satisfied as well. By this we mean that for all $x \in \Omega$ there exists $\delta > 0$, a convex set $C \subset \mathbb{R}^d$, and a diffeomorphism $\Phi : C \to \mathbb{R}^d$ with inverse Ψ such that $B_\delta(x) \cap \Omega = \Phi(C)$. In particular, this includes domains with a sufficiently regular boundary. To see this let $x \in \Omega$ and $y \in B_\delta(x) \cap \Omega = \Phi(C)$. Because C is convex, we can connect $\Psi(x)$ and $\Psi(y)$ with a straight line $\tau(t) = (1 - t)\Psi(x) + t\Psi(y)$ for $t \in [0, 1]$ and consider the curve $\gamma(t) = \Phi(\tau(t)) \subset \Phi(C)$ which lies in $B_\delta(x) \cap \Omega$ since τ lies in C. Hence,

$$\begin{aligned} d_\Omega(x, y) \leq \mathrm{len}(\gamma) &= \int_0^1 |\dot\gamma(t)| \, \mathrm{d}t \leq \int_0^1 |\nabla\Phi(\tau(t))| \, |\dot\tau(t)| \, \mathrm{d}t \\ &= \int_0^1 |\nabla\Phi(\tau(t))| \, |\Psi(y) - \Psi(x)| \, \mathrm{d}t \\ &= \int_0^1 |\nabla\Phi(\tau(t))| \, |\nabla\Psi(x)| \, |x - y| \, \mathrm{d}t + o(|x - y|) \\ &\leq \int_0^1 |\nabla\Phi(\tau(t))| \, |\nabla\Psi(\gamma(t))| \, |x - y| \, \mathrm{d}t \\ &\quad + \int_0^1 |\nabla\Phi(\tau(t))| \, |\nabla\Psi(\gamma(t)) - \nabla\Psi(x)| \, |x - y| \, \mathrm{d}t + o(|x - y|) \end{aligned}$$

$$
\begin{aligned}
&\leq |x-y| + c\,|x-y| \int_0^1 |\nabla \Phi(\tau(t))|\,|\gamma(t)-x|\,\mathrm{d}t + o(|x-y|)\\
&= |x-y| + c\,|x-y| \int_0^1 |\nabla \Phi(\tau(t))|\\
&\quad \times |\Phi(\tau(t)) - \Phi(\Psi(x))|\,\mathrm{d}t + o(|x-y|)\\
&\leq |x-y| + c\,|x-y| \int_0^1 |\tau(t) - \Psi(x)|\,\mathrm{d}t + o(|x-y|)\\
&= |x-y| + c\,|x-y|\,|\Psi(y) - \Psi(x)| \int_0^1 t\,\mathrm{d}t + o(|x-y|)\\
&\leq |x-y| + c\,|x-y|^2 + o(|x-y|),
\end{aligned}
$$

where we used Lipschitz continuity of Φ, Ψ and $\nabla\Psi$ and the fact that Φ is a diffeomorphism which implies $\nabla\Phi(\tau(t)) = (\nabla\Psi(\gamma(t)))^{-1}$. Note that the constant $c > 0$ is changing with every inequality. Dividing by $|x-y|$ and letting $|x-y| \to 0$, we finally get (11).

3.2 Discrete to Continuum Convergence

We now consider the Γ-convergence of the discrete functionals. While in the previous section we employed an arbitrary null sequence $(s_n)_{n\in\mathbb{N}} \subset (0,\infty)$ for the scaling, we are now limited to certain scaling sequences depending on the sequence of sets Ω_n. In particular, we have to control how fast the scaling s_n tends to zero in comparison with how fast the points in Ω_n fill out the domain Ω. The following simple example illustrates why we have to consider the relationship between s_n and Ω_n.

Example 3 Let $(x_n)_{n\in\mathbb{N}} \subset \mathbb{R}^d$ be an arbitrary sequence of points, then we can choose s_n small enough such that $\eta_{s_n}(|x-y|) = 0$ for $x, y \in \Omega_n$ and thus we have that $E_n(u_n) = 0$ for every $n \in \mathbb{N}$. In this situation the liminf inequality does not hold true.

As illustrated in the example above, we need to take special care of points $x, y \in \Omega_n$, where $\eta_{s_n}(|x-y|) = 0$. Formulating this problem in terms of the map p_n we have to consider the case where

$$\eta_{s_n}(|p_n(x) - p_n(y)|) = 0.$$

Using that the kernel has radius $c_\eta < \infty$ it follows that $|p_n(x) - p_n(y)| > c_\eta s_n$ and thus

$$
\begin{aligned}
|x-y| &= |x - p_n(x) + p_n(x) - p_n(y) + p_n(y) - y|\\
&\geq |p_n(x) - p_n(y)| - 2\,\|\mathrm{Id} - p_n\|_{L^\infty}\\
&> c_\eta s_n - 2\,\|\mathrm{Id} - p_n\|_{L^\infty} =: c_\eta \tilde{s}_n.
\end{aligned}
$$

The idea now is to use this new scaling $\tilde{s}_n$ for the non-local functionals, where we have to impose that $\tilde{s}_n > 0$ for all n large enough. But more importantly we must

ensure that the quotient $\tilde{s}_n/s_n$ converges to 1, i.e.,

$$\frac{\tilde{s}_n}{s_n} = \frac{s_n - 2\,\|\mathrm{Id} - p_n\|_{L^\infty}/c_\eta}{s_n} = 1 - \frac{2\,\|\mathrm{Id} - p_n\|_{L^\infty}/c_\eta}{s_n} \longrightarrow 1,$$

which is equivalent to the fact that

$$\frac{\|\mathrm{Id} - p_n\|_{L^\infty}}{s_n} \longrightarrow 0.$$

This argumentation was first applied in [22], where instead of the map p_n an optimal transport map T_n was employed. For a closest point projection p_n we know that

$$\|\mathrm{Id} - p_n\|_{L^\infty} = \sup_{x\in\Omega} \operatorname{dist}(x, \Omega_n) = r_n.$$

which thus yields the scaling assumption (8).

Lemma 5 (liminf inequality) *Let $\Omega \subset \mathbb{R}^d$ be a domain, let the constraint sets satisfy (3), and let the kernel fulfill (K1)–(K3), then for any null sequence $(s_n)_{n\in\mathbb{N}} \subset (0,\infty)$ which satisfies the scaling condition (8) we have that*

$$\liminf_{n\to\infty} E_{n,\mathrm{cons}}(u_n) \geq \sigma_\eta\, \mathcal{E}_{\mathrm{cons}}(u)$$

for every sequence $(u_n)_{n\in\mathbb{N}} \subset L^\infty(\Omega)$ converging to $u \in L^\infty(\Omega)$ in $L^\infty(\Omega)$.

Proof W.l.o.g. we assume that $\liminf_{n\to\infty} E_{n,\mathrm{cons}}(u_n) < \infty$. After possibly passing to a subsequence, we can, furthermore, assume that $u_n = g$ on $\mathcal{O}_n$. We first show that the limit function u satisfies $u = g$ on $\mathcal{O}$.

Since η is continuous and positive in 0, we know that there exists $0 < t < c_\eta$ such that $\eta(s) > C$ for all $0 < s < t$ where $C > 0$. Furthermore, using (3) we infer that for all $x \in \mathcal{O}$ there exists $x_n \in \mathcal{O}_n$ with $|x - x_n| = o(s_n)$. In particular, for n large enough it holds $|x - x_n| \leq s_n t$. This allows us to estimate:

$$\begin{aligned}
|u(x) - g(x)| &\leq |u(x) - u_n(x)| + |u_n(x) - u_n(x_n)| \\
&\quad + |u_n(x_n) - g(x_n)| + |g(x_n) - g(x)| \\
&\leq \|u - u_n\|_{L^\infty(\Omega)} + \frac{1}{C}\eta(|x - x_n|)\,|u_n(x) - u_n(x_n)| \\
&\quad + 0 + \mathrm{Lip}(g)\,|x - x_n| \\
&\leq \|u - u_n\|_{L^\infty(\Omega)} + \frac{s_n}{C} E_{n,\mathrm{cons}}(u_n) + \mathrm{Lip}(g)\,|x - x_n| \\
&\leq \|u - u_n\|_{L^\infty(\Omega)} + \frac{s_n}{C} E_{n,\mathrm{cons}}(u_n) + \mathrm{Lip}(g) s_n\, t.
\end{aligned}$$

Taking $n \to \infty$, using that $E_{s_n}(u_n)$ is uniformly bounded and $s_n \to 0$, we obtain $u = g$ on $\mathcal{O}$.

The main idea for proving the liminf inequality here is to establish a discrete to non-local control estimate and then use Lemma 3.

Since we assumed $\liminf_{n\to\infty} E_{n,\mathrm{cons}}(u_n) < \infty$, we know that u_n is piecewise constant for every $n \in \mathbb{N}$, in the sense of Sect. 1.1, i.e., $u_n = \mathbf{u}_n \circ p_n$ for some $\mathbf{u}_n : \Omega_n \to \Omega$. As shown in (23) we can express $E_{n,\mathrm{cons}}$ as follows:

$$E_{n,\mathrm{cons}}(u_n) = \frac{1}{s_n} \operatorname{ess\,sup}_{x,y\in\Omega} \eta_{s_n}(|p_n(x) - p_n(y)|)\, |u_n(x) - u_n(y)| = (\#).$$

In order to apply Lemma 3, we need to transform the weighting that considers the distance between $p_n(x)$ and $p_n(y)$ into another one that measures the distance between x and y.

Case 1 There exists $t > 0$ such that η is constant on $[0, t]$.

We employ the observation that whenever $|p_n(x) - p_n(y)| > s_n\, t$, for the new scaling $\tilde{s}_n := s_n - 2r_n/t$ we have

$$\begin{aligned}\frac{|x-y|}{\tilde{s}_n} &\geq \frac{|p_n(x) - p_n(y)| - 2r_n}{s_n - 2r_n/t} \\ &= \frac{|p_n(x) - p_n(y)|}{s_n} \underbrace{\frac{1 - 2r_n/\,|p_n(x) - p_n(y)|}{1 - 2r_n/(t s_n)}}_{>1} \\ &> \frac{|p_n(x) - p_n(y)|}{s_n},\end{aligned}$$

where we used that $r_n = \|\mathrm{Id} - p_n\|_{L^\infty}$. Since η is non-increasing (K2) and η is constant on $[0, t)$, we get

$$\eta_{s_n}(|p_n(x) - p_n(y)|) \geq \eta_{\tilde{s}_n}(|x - y|)$$

for almost all $x, y \in \Omega$. This allows us to further estimate

$$(\#) \geq \frac{1}{s_n} \operatorname{ess\,sup}_{x,y\in\Omega} \eta_{\tilde{s}_n}(|x-y|)\, |u_n(x) - u_n(y)| = \frac{\tilde{s}_n}{s_n}\, \mathcal{E}_{\tilde{s}_n}(u_n).$$

Together with the assumption $r_n/s_n \to 0$ we obtain that $\tilde{s}_n > 0$ for n large enough and $\tilde{s}_n/s_n \to 1$ which finally justifies the application of Lemma 3, i.e.,

$$\liminf_{n\to\infty} E_{n,\mathrm{cons}}(u_n) \geq \liminf_{n\to\infty} \frac{\tilde{s}_n}{s_n}\, \mathcal{E}_{\tilde{s}_n}(u_n) \geq \sigma_\eta\, \mathcal{E}(u) = \sigma_\eta\, \mathcal{E}_{\mathrm{cons}}(u).$$

Case 2 We now assume the kernel to fulfill (K1)–(K3). The strategy is to find a $t > 0$ where one can cut off the kernel without changing the value σ_η. From the continuity at $t = 0$ (K1) we have that

$$\lim_{t\to 0} \eta(t)\, t = 0$$

and thus there exists a $t^* > 0$ such that

$$\sup_{t\in[0,t^*]} \eta(t)\, t \le \sigma_\eta.$$

We define

$$\tilde{\eta}(t) = \begin{cases} \eta(t) & \text{for } t > t^*, \\ \eta(t^*) & \text{for } t \in [0, t^*], \end{cases}$$

for which we have $\sigma_{\tilde{\eta}} = \sigma_\eta$ and thus the first case applies. Namely, using that η is non-increasing (K2) and hence $\eta \ge \tilde{\eta}$ we obtain

$$\begin{aligned} &\liminf_{n\to\infty} E_{n,\mathrm{cons}}(u_n) \\ &\quad \ge \liminf_{n\to\infty} \left(\frac{1}{s_n} \operatorname{ess\,sup}_{x,y\in\Omega} \tilde{\eta}_{s_n}(|p_n(x) - p_n(y)|)\, |u_n(x) - u_n(y)| \right) \\ &\quad \ge \sigma_\eta\, \mathcal{E}_{\mathrm{cons}}(u). \end{aligned}$$

□

We now consider the limsup inequality for the constrained functionals.

Lemma 6 (limsup inequality) *Let $\Omega \subset \mathbb{R}^d$ be a domain satisfying (11) and let the kernel fulfill (K1)–(K3), then for a null sequence $(s_n)_{n\in\mathbb{N}} \subset (0,\infty)$ and a function $u \in L^\infty(\Omega)$ there exists a sequence $(u_n)_{n\in\mathbb{N}} \subset L^\infty(\Omega)$ converging to $u \in L^\infty(\Omega)$ in $L^\infty(\Omega)$ such that*

$$\limsup_{n\to\infty} E_{n,\mathrm{cons}}(u_n) \le \sigma_\eta\, \mathcal{E}_{\mathrm{cons}}(u).$$

Proof If $\mathcal{E}_{\mathrm{cons}}(u) = \infty$ the inequality holds trivially. We thus consider $u \in W^{1,\infty}(\Omega)$ such that $u(x) = g(x)$ for every $x \in \mathcal{O}$ and define a recovery sequence as follows: Let $\mathbf{u}_n \in L^\infty(v_n)$ be defined by

$$\mathbf{u}_n(x) = \begin{cases} u(x), & x \in \Omega_n \backslash \mathcal{O}_n, \\ g(x), & x \in \mathcal{O}_n, \end{cases}$$

and define $u_n := \mathbf{u}_n \circ p_n$, where $p_n : \Omega \to \Omega_n$ denotes a closest point projection. Then $u_n \in L^\infty(\Omega)$ and by definition it holds

$$E_{n,\mathrm{cons}}(u_n) = E_{n,\mathrm{cons}}(\mathbf{u}_n) = \frac{1}{s_n} \max_{x,y\in\Omega_n} \eta_{s_n}(|x-y|)\, |\mathbf{u}_n(x) - \mathbf{u}_n(y)|\,.$$

We have to distinguish three cases:

Case 1 Let $x, y \in \Omega_n \backslash \mathcal{O}_n$, then we can compute, using (27)

$$|\mathbf{u}_n(x) - \mathbf{u}_n(y)| = |u(x) - u(y)| \leq d_\Omega(x, y) \ \|\nabla u\|_{L^\infty}$$

and therefore

$$\begin{aligned}\frac{1}{s_n}\eta_{s_n}(|x-y|)\,|\mathbf{u}_n(x) - \mathbf{u}_n(y)| &\leq \underbrace{\eta_{s_n}(|x-y|)\,\frac{|x-y|}{s_n}}_{\leq \sigma_\eta}\,\frac{d_\Omega(x,y)}{|x-y|}\ \|\nabla u\|_{L^\infty} \\ &\leq \sigma_\eta\,\frac{d_\Omega(x,y)}{|x-y|}\ \|\nabla u\|_{L^\infty}\,.\end{aligned}$$

Case 2 Let $x \in \Omega_n \backslash \mathcal{O}_n$ and $y \in \mathcal{O}_n$. Then for every $\tilde{y} \in \mathcal{O}$ it holds, using (27)

$$\begin{aligned}|\mathbf{u}_n(x) - \mathbf{u}_n(y)| &= |u(x) - g(y)| \\ &\leq |u(x) - u(\tilde{y})| + \underbrace{|u(\tilde{y}) - g(\tilde{y})|}_{=0} + |g(\tilde{y}) - g(y)| \\ &\leq \|\nabla u\|_{L^\infty}\,\mathrm{d}_\Omega(x,\tilde{y}) + \mathrm{Lip}(g)\,|\tilde{y} - y| \\ &\leq \|\nabla u\|_{L^\infty}\,d_\Omega(x,y) + \|\nabla u\|_{L^\infty}\,d_\Omega(y,\tilde{y}) + \mathrm{Lip}(g)\,|\tilde{y} - y| \\ &\leq \|\nabla u\|_{L^\infty}\,d_\Omega(x,y) + \|\nabla u\|_{L^\infty}\,\frac{d_\Omega(y,\tilde{y})}{|y-\tilde{y}|} \\ &\quad |y-\tilde{y}| + \mathrm{Lip}(g)\,|y-\tilde{y}|\,.\end{aligned}$$

From this we have, using the same arguments as in the first case, that there is a $C > 0$ such that

$$\frac{1}{s_n}\eta_{s_n}(|x-y|)\,|\mathbf{u}_n(x) - \mathbf{u}_n(y)| \leq \sigma_\eta\,\frac{d_\Omega(x,y)}{|x-y|}\ \|\nabla u\|_{L^\infty} + C\,\frac{|y-\tilde{y}|}{s_n}.$$

Case 3 Let $x, y \in \mathcal{O}_n$, then for $\tilde{x}, \tilde{y} \in \mathcal{O}$ we have

$$\begin{aligned}|\mathbf{u}_n(x) - \mathbf{u}_n(y)| &= |g(x) - g(y)| \\ &= |g(x) - u(\tilde{x})| + |u(\tilde{x}) - u(\tilde{y})| + |u(\tilde{y}) - g(y)| \\ &= |g(x) - g(\tilde{x})| + |u(\tilde{x}) - u(\tilde{y})| + |g(\tilde{y}) - g(y)| \\ &\leq \mathrm{Lip}(g)\,|x-\tilde{x}| + \|\nabla u\|_{L^\infty}\,d_\Omega(\tilde{x},\tilde{y}) + \mathrm{Lip}(g)\,|y-\tilde{y}|\end{aligned}$$

and therefore again

$$\begin{aligned}\frac{1}{s_n}\eta_{s_n}(|x-y|)\,|\mathbf{u}_n(x) - \mathbf{u}_n(y)| &\leq \sigma_\eta\,\frac{d_\Omega(x,y)}{|x-y|}\ \|\nabla u\|_{L^\infty} \\ &\quad + \mathrm{Lip}(g)\,\left(\frac{|y-\tilde{y}| + |x-\tilde{x}|}{s_n}\right).\end{aligned}$$

By (11) for every $\varepsilon > 0$ there is $n_0 \in \mathbb{N}$ sufficiently large such that for all $n \geq n_0$ it holds

$$\sigma_\eta \, \frac{d_\Omega(x, y)}{|x - y|} \, \|\nabla u\|_{L^\infty} \leq \sigma_\eta \, \|\nabla u\|_{L^\infty} + \varepsilon/2,$$

whenever $|x - y| \leq c_\eta s_n$. Additionally, thanks to (3) and the compactness of $\mathcal{O}_n$ and $\mathcal{O}$ for every $x \in \mathcal{O}_n$ we can choose $\tilde{x} \in \mathcal{O}$ such that

$$\frac{|x - \tilde{x}|}{s_n} \leq \frac{\varepsilon}{4 \max\{C, \mathrm{Lip}(g)\}}$$

and analogously for y and $\tilde{y}$. Combining the estimates from all three cases, we obtain

$$\begin{aligned} E_{n,\mathrm{cons}}(u_n) &\leq \max\left\{\sigma_\eta \, \|\nabla u\|_{L^\infty} + \frac{\varepsilon}{2}, \sigma_\eta \, \|\nabla u\|_{L^\infty} + \frac{3\varepsilon}{4}, \sigma_\eta \, \|\nabla u\|_{L^\infty} + \varepsilon\right\} \\ &= \sigma_\eta \, \|\nabla u\|_{L^\infty} + \varepsilon \end{aligned}$$

for all $n \geq n_0$. Finally, this yields

$$\limsup_{n \to \infty} E_{n,\mathrm{cons}}(u_n) \leq \sigma_\eta \, \mathcal{E}_{\mathrm{cons}}(u),$$

as desired.

For showing that $u_n \to u$ in $L^\infty(\Omega)$ one proceeds similarly: If $p_n(x) \in \Omega_n \backslash \mathcal{O}_n$ one has thanks to (27)

$$\begin{aligned} |u(x) - u_n(x)| &= |u(x) - u(p_n(x))| \leq \|\nabla u\|_{L^\infty} \, d_\Omega(x, p_n(x)) \\ &= \|\nabla u\|_{L^\infty} \, \frac{d_\Omega(x, p_n(x))}{|x - p_n(x)|} \, |x - p_n(x)| \to 0, \quad n \to \infty, \end{aligned}$$

where we also used (11) and $|x - p_n(x)| \leq r_n \to 0$. In the case $p_n(x) \in \mathcal{O}_n$ by (3) one again finds $\tilde{x} \in \mathcal{O}$ such that $|p_n(x) - \tilde{x}| = o(s_n)$. Then by (27) and (11) we have

$$\begin{aligned} |u(x) - u_n(x)| &= |u(x) - g(p_n(x))| \\ &\leq |u(x) - u(p_n(x))| + |u(p_n(x)) - \underbrace{g(\tilde{x})}_{=u(\tilde{x})}| + |g(\tilde{x}) - g(p_n(x))| \\ &\leq \|\nabla u\|_{L^\infty} \, \frac{d_\Omega(p_n(x), x)}{|p_n(x) - x|} \, |p_n(x) - x| \\ &\quad + \|\nabla u\|_{L^\infty} \, \frac{d_\Omega(p_n(x), \tilde{x})}{|p_n(x) - \tilde{x}|} \, |p_n(x) - \tilde{x}| + \mathrm{Lip}(g) \, |p_n(x) - \tilde{x}| \\ &\to 0 \end{aligned}$$

since $|p_n(x) - x| \leq r_n \to 0$ and $|p_n(x) - \tilde{x}| = o(s_n) \to 0$. Combining both cases proves $\|u - u_n\|_{L^\infty} \to 0$ as $n \to \infty$. □

Remark 7 We note that the proof of the limsup inequality does not use any specific properties of the scaling, in fact even a sequence of disconnected graphs or the situation of Example 3 allows for such an inequality.

Remark 8 (Relevance of the Hausdorff convergence) The condition that the Hausdorff distance of $\mathcal{O}_n$ and $\mathcal{O}$ converges to zero as $n \to \infty$ (cf. (3)) implies both that $\mathcal{O}_n$ well approximates $\mathcal{O}$ and vice versa. The first condition is only used in the proof of the liminf inequality Lemma 5 whereas the second one only enters for the limsup inequality Lemma 6. Furthermore, the proof of the latter is drastically simplified if one assumes that $\mathcal{O}_n \subset \mathcal{O}$ for all $n \in \mathbb{N}$, which implies that the second term in the Hausdorff distance (3) equals zero. In this case, introducing the continuum points $\tilde{x}, \tilde{y} \in \mathcal{O}$ is not necessary and many estimates in the previous proof become trivial.

Combining Lemmas 5 and 6 we immediately obtain the Γ-convergence of the discrete functionals to those defined in the continuum, which is the statement of Theorem 1.

Remark 9 (Homogeneous boundary conditions) In the case that $\mathcal{O} = \partial\Omega$ and the constraints satisfy $g = 0$ on $\mathcal{O}$ any function with $\mathcal{E}_{\text{cons}}(u) < \infty$ satisfies $u \in W_0^{1,\infty}(\Omega)$. For this it is well known that functions $u \in W_0^{1,\infty}(\Omega)$ can be extended from Ω to $\mathbb{R}^d$ by zero without changing $\|\nabla u\|_{L^\infty}$. In this case one can prove the limsup inequality Lemma 6 and hence also the Γ-convergence Theorem 1 for *general open sets* Ω without demanding (11) or even convexity. For this one simply utilizes the estimate

$$|\mathbf{u}_n(x) - \mathbf{u}_n(y)| = |u(x) - u(y)| \leq \|\nabla u\|_{L^\infty} |x - y|$$

which is true if one extends u by zero on $\mathbb{R}^d \setminus \Omega$, multiplies with the kernel, and takes the supremum.

4 Compactness

We now want to make use of Lemma 1 in order to characterize the behavior of minimizers of the discrete problems or more generally sequences of approximate minimizers, as described in the condition of the mentioned lemma. The first result is a general characterization of relatively compact sets in L^∞, the proof uses classical ideas from [18, Lem. IV.5.4].

Lemma 7 *Let (Ω, μ) be a finite measure space and $K \subset L^\infty(\Omega; \mu)$ be a bounded set w.r.t. $\|\cdot\|_{L^\infty(\Omega;\mu)}$ such that for every $\varepsilon > 0$ there exists a finite partition $\{V_i\}_{i=1}^n$ of Ω into subsets V_i with positive and finite measure such that*

$$\mu\text{-ess sup}_{x,y \in V_i} |u(x) - u(y)| < \varepsilon \ \forall u \in K, i = 1, \ldots, n, \tag{30}$$

then K is relatively compact.

Proof Let $\varepsilon > 0$ be given and let $\{V_i\}_{i=1}^n$ be a partition into sets with finite and positive measure such that

$$\mu\text{-ess sup}_{x,y \in V_i} |u(x) - u(y)| < \frac{\varepsilon}{3}, \quad \forall u \in K, \ i = 1, \ldots, n. \tag{31}$$

We define the operator $\mathcal{T} : L^\infty(\Omega;\mu) \to L^\infty(\Omega;\mu)$ as

$$(\mathcal{T}u)(x) := \frac{1}{\mu(V_i)} \int_{V_i} u(y)\, \mathrm{d}\mu(y) \quad \text{for } x \in V_i,$$

which is well defined thanks to $0 < \mu(V_i) < \infty$ for all $i = 1, \dots, n$. Using (31) we observe that for μ-almost every $x \in V_i$

$$|u(x) - (\mathcal{T}u)(x)| \le \frac{1}{\mu(V_i)} \int_{V_i} |u(x) - u(y)|\, \mathrm{d}\mu(y) < \frac{\varepsilon}{3}$$

and thus $\|u - \mathcal{T}u\|_{L^\infty(\Omega;\mu)} < \frac{\varepsilon}{3}$. Furthermore, $\mathcal{T}(K) \subset \operatorname{span}(\{\mathbb{I}_{V_1}, \dots, \mathbb{I}_{V_n}\})$, where we let $\mathbb{I}_M$ denote the indicator function of a set M, defined by $\mathbb{I}_M(x) = 0$ if $x \notin M$ and $\mathbb{I}_M(x) = 1$ if $x \in M$. Hence, $\mathcal{T}$ has finite-dimensional range and since K is bounded we have

$$\|\mathcal{T}u\|_{L^\infty(\Omega;\mu)} \le \|u\|_{L^\infty(\Omega;\mu)} \le C \quad \forall u \in K$$

and therefore $\mathcal{T}(K)$ is relatively compact. This implies that there exist finitely many functions $\{u_j\}_{j=1}^N \subset L^\infty(\Omega;\mu)$ such that

$$\mathcal{T}(K) \subset \bigcup_{j=1}^N B_{\frac{\varepsilon}{3}}(\mathcal{T}(u_j)),$$

where $B_t(u) := \{v \in L^\infty(\Omega;\mu) \,:\, \|u - v\|_{L^\infty(\Omega;\mu)} < t\}$ denotes the open ball with radius $t > 0$ around $u \in L^\infty(\Omega;\mu)$. For $u \in K$ we can thus find $j \in \{1, \dots, N\}$ such that $\mathcal{T}(u) \in B_{\frac{\varepsilon}{3}}(\mathcal{T}(u_j))$ and thus

$$\|u - u_j\|_{L^\infty} \le \|u - \mathcal{T}(u)\|_{L^\infty} + \|\mathcal{T}(u) - \mathcal{T}(u_j)\|_{L^\infty} + \|\mathcal{T}(u_j) - u_j\|_{L^\infty} < \varepsilon.$$

This implies that K is totally bounded and since $L^\infty(\Omega;\mu)$ is complete the result follows from [18, Lem. I.6.15]. □

The previous lemma allows us to prove a compactness result for the non-local functionals, where we again need the domain Ω to fulfill condition (11).

Lemma 8 *Let $\Omega \subset \mathbb{R}^d$ be a bounded domain satisfying (11) and let the kernel η fulfill (K1)–(K3), and let $(s_n)_{n\in\mathbb{N}} \subset (0,\infty)$ be a null sequence. Then every bounded sequence $(u_n)_{n\in\mathbb{N}} \subset L^\infty(\Omega)$ such that*

$$\sup_{n\in\mathbb{N}} \mathcal{E}_{s_n}(u_n) < \infty \tag{32}$$

is relatively compact.

Proof We want to apply Lemma 7 in order to see that the sequence is relatively compact. Therefore, let $\varepsilon > 0$ be given and w.l.o.g. we rescale the kernel such that

$$\eta(t) \geq 1 \text{ for } t \leq 1.$$

Using (11) we can find $\delta > 0$ such that for every $x, y \in \Omega$ with $|x - y| \leq \delta$ there is a path $\gamma : [0, 1] \to \Omega$ such that $\gamma(0) = x$, $\gamma(1) = y$ and

$$\operatorname{len}(\gamma) \leq (1 + \varepsilon)\,|x - y|\,.$$

We divide this path by points $0 = t_0 < \cdots < t_i < \cdots < t_{k_n} = 1$ such that for $z_i := \gamma(t_i)$ we have that

$$|z_i - z_{i+1}| \leq s_n$$

for $i = 0, \ldots, k_n$, where

$$k_n \leq \lfloor (1 + \varepsilon)\ |x - y|\,/s_n \rfloor.$$

Then we have that

$$\begin{aligned}
|u_n(x) - u_n(y)| &\leq \sum_{i=0}^{k_n-1} |u_n(z_i) - u_n(z_{i+1})| \\
&\leq s_n \sum_{i=0}^{k_n-1} \eta_{s_n}(|z_i - z_{i+1}|) \frac{|u_n(z_i) - u_n(z_{i+1})|}{s_n} \\
&\leq s_n \sum_{i=0}^{k_n-1} \mathcal{E}_{s_n}(u_n) \\
&\leq s_n\ k_n \underbrace{\sup_{n\in\mathbb{N}} \mathcal{E}_{s_n}(u_n)}_{=:C<\infty} \\
&\leq C\ (1 + \varepsilon)\ |x - y|\,.
\end{aligned}$$

Choosing a partition $\{V_i\}_{i=1}^N$ of Ω into sets with positive Lebesgue measure such that

$$\operatorname{diam}(V_i) < \min\left\{\delta, \frac{\varepsilon}{C\ (1 + \varepsilon)}\right\}$$

for $i = 1, \ldots, N$, yields that

$$\begin{aligned} &\operatorname{ess\,sup}_{x,y\in V_i} |u_n(x) - u_n(y)| \leq C\ (1+\varepsilon)\ \operatorname{ess\,sup}_{x,y\in V_i} |x-y| \\ &\quad \leq C\ (1+\varepsilon)\ \operatorname{diam}(V_i) < \varepsilon. \end{aligned}$$

Since $(u_n)_{n\in\mathbb{N}} \subset L^\infty$ is bounded in L^∞ we can therefore apply Lemma 7 to infer that the sequence is relatively compact. □

We will use this result in order to prove that the constrained functionals $E_{n,\mathrm{cons}}$ are compact, which then directly shows Theorem 2. The intuitive reason that these functionals are compact is the fact that for a domain Ω that fulfills (11) each point $x \in \Omega_n$ has finite geodesic distance to the set $\mathcal{O}_n$. This follows from the fact that the geodesic diameter of Ω is bounded, as we show in the following lemma.

Lemma 9 *Condition* (11) *implies that the geodesic diameter is finite, i.e.,*

$$\operatorname{diam}_g(\Omega) := \sup_{x,y\in\Omega} d_\Omega(x,y) < \infty. \tag{33}$$

Proof For $\varepsilon > 0$ we can use (11) to find $\delta > 0$ such that $d_\Omega(x,y) < |x-y|\,(1+\varepsilon)$ for all $x, y \in \Omega$ with $|x-y| < \delta$. Since we assume Ω to be bounded we know that there exists a finite collection $\{x_1, \ldots, x_N\} \subset \Omega$ such that

$$\Omega \subset \bigcup_{i=1}^{N} B_\delta(x_i).$$

If two balls at centers x_i, x_j share a common point $z \in \Omega$ we see that

$$\begin{aligned} d_\Omega(x_i,x_j) &\leq d_\Omega(x_i,z) + d_\Omega(z,x_j) \\ &\leq (1+\varepsilon)\,|x_i - z| + (1+\varepsilon)\,|z - x_j| \\ &\leq 2(1+\varepsilon)\delta. \end{aligned} \tag{34}$$

For any $x, y \in \Omega$ assume that there exists a path γ in Ω from x to y. Therefore, also the image of γ is covered by finitely many balls at centers $x_{k_1}, \ldots, x_{k_n}$ such that

$$B_\delta(x_{k_i}) \cap B_\delta(x_{k_{i+1}}) \cap \Omega \neq \emptyset$$

for $i = 1, \ldots, n-1$ with $x \in B_\delta(x_{k_1})$, $y \in B_\delta(x_{k_n})$. Using (34) this yields

$$\begin{aligned} d_\Omega(x,y) &\leq d_\Omega(x,x_{k_1}) + \sum_{i=1}^{k_{n-1}} d_\Omega(x_i,x_{i+1}) + d_\Omega(x_{k_n},y) \\ &\leq 2\ (N+1)(1+\varepsilon)\delta \end{aligned}$$

where we used $k_n \leq N$. Note that the last expression above is independent of x, y which concludes the proof. □

Lemma 10 *Let $\Omega \subset \mathbb{R}^d$ be a domain satisfying (11), let the kernel fulfill (K1)–(K3), and $(s_n)_{n\in\mathbb{N}} \subset (0,\infty)$ be a null sequence which satisfies the scaling condition (8). Let $(u_n)_{n\in\mathbb{N}} \subset L^\infty(\Omega)$ be a sequence with*

$$\sup_{n\in\mathbb{N}} E_{n,\mathrm{cons}}(u_n) < \infty$$

then it is bounded with respect to $\|\cdot\|_\infty$.

Remark 10 Similar to Remark 9, one can relax condition (11) in Lemma 10 by taking into account the specific form of the constraint set $\mathcal{O} \subset \overline{\Omega}$. Indeed, an inspection of the following proof shows that it suffices to demand that

$$\sup_{x\in\Omega} \inf_{y\in\mathcal{O}} d_\Omega(x,y) < \infty, \tag{35}$$

which means that $\mathcal{O}$ has finite geodesic distance to any point in Ω. In the case that $\mathcal{O} = \partial\Omega$ this is always satisfied since Ω is bounded. However, the condition is violated, e.g., if Ω is an open, infinite but bounded spiral and $\mathcal{O}$ a single point at its center.

Proof W.l.o.g. we assume

$$\eta(t) \geq 1 \text{ for } t \leq 1.$$

Let $n_0 \in \mathbb{N}$ be large enough such that

$$r_n < s_n/2, \ \forall n \geq n_0$$

and for $x_0 \in \Omega_n$, $n \geq n_0$ let $\gamma : [0,1] \to \Omega$ be a path in Ω such that $\gamma(0) = x_0$ and $\gamma(1) \in \mathcal{O}_n$. We divide γ by points $0 = t_0 < \cdots < t_{k_n} = 1$ such that

$$\begin{aligned} |\gamma(t_i) - \gamma(t_{i+1})| &= s_n - 2r_n, \quad i = 0,\dots,k_n-2, \\ \left|\gamma(t_{k_n}) - \gamma(t_{k_n+1})\right| &\leq s_n - 2r_n, \end{aligned}$$

and by definition of the parameter r_n we know that for each $i = 0,\dots,k_n$ there exists a vertex $x_i \in \Omega_n$ such that

$$|x_i - \gamma(t_i)| \leq r_n.$$

Applying the triangle inequality this yields

$$\begin{aligned} |x_i - x_{i+1}| &\leq |x_i - \gamma(t_i)| + |\gamma(t_i) - \gamma(t_{i+1})| + |x_{i+1} - \gamma(t_{i+1})| \\ &\leq 2r_n + s_n - 2r_n \leq s_n, \end{aligned}$$

and thus $\eta_{s_n}(|x_i - x_{i+1}|) \geq 1$ for all $i = 0, \ldots, k_n - 1$. By definition of the discrete functional (6) there exists $\mathbf{u}_n : \Omega \to \mathbb{R}$ with $u_n = \mathbf{u}_n \circ p_n$ and we can estimate

$$\begin{aligned} |\mathbf{u}_n(x_0)| &\leq \sum_{i=0}^{k_n-2} |\mathbf{u}_n(x_i) - \mathbf{u}_n(x_{i+1})| + \left|\mathbf{u}_n(x_{k_n})\right| \\ &\leq s_n \sum_{i=0}^{k_n-2} \eta_{s_n}(|x_i - x_{i+1}|)\, |\mathbf{u}_n(x_i) - \mathbf{u}_n(x_{i+1})| \,/ s_n + \left|g(x_{k_n})\right| \\ &\leq s_n\ (k_n - 1)\ E_{n,\mathrm{cons}}(u_n) + \left|g(x_{k_n})\right|, \end{aligned} \tag{36}$$

where we used $\mathbf{u}_n(x) = g(x)$ for all $x \in \mathcal{O}_n$ since $E_{n,\mathrm{cons}}(u_n) < \infty$. It remains to show that the product $s_n\ (k_n - 1)$ is uniformly bounded in n, for which we first observe that the path γ can be chosen such that

$$k_n - 1 \leq \lfloor \mathrm{diam}_g(\Omega)/(s_n - 2r_n) \rfloor$$

and thus using (8)

$$s_n\ (k_n - 1) \leq s_n \left\lfloor \frac{\mathrm{diam}_g(\Omega)}{(s_n - 2r_n)} \right\rfloor \leq C \frac{1}{(1 - 2\, r_n/s_n)} < \tilde{C},\ \forall n \in \mathbb{N},$$

where we note that $\mathrm{diam}_g(\Omega) < \infty$ according to Lemma 9. Together with (36) this yields that there exists a uniform constant $C > 0$ such that $\|u_n\|_{L^\infty} \leq C$ for all $n \in \mathbb{N}$. □

We can now prove that the constrained functionals $E_{n,\mathrm{cons}}$ are indeed compact.

Lemma 11 *Let $\Omega \subset \mathbb{R}^d$ be a domain satisfying (11), let the kernel fulfill (K1)–(K3), and $(s_n)_{n\in\mathbb{N}} \subset (0, \infty)$ be a null sequence which satisfies the scaling condition (8). Then we have that every sequence $(u_n)_{n\in\mathbb{N}} \subset L^\infty(\Omega)$ such that*

$$\sup_{n\in\mathbb{N}} E_{n,\mathrm{cons}}(u_n) < \infty$$

is relatively compact in L^∞.

Proof Using the same arguments as in the proof of Lemma 5 we can find a scaling sequence $(\tilde{s}_n)_{n\in\mathbb{N}} \subset (0, \infty)$ such that

$$E_n(u_n) \geq \frac{\tilde{s}_n}{s_n}\, \mathcal{E}_{\tilde{s}_n}(u_n)$$

and $\tilde{s}_n/s_n \longrightarrow 1$. We choose $C := \sup_{n\in\mathbb{N}} \frac{s_n}{\tilde{s}_n} < \infty$ to obtain

$$\sup_{n\in\mathbb{N}} \mathcal{E}_{\tilde{s}_n}(u_n) \leq C \sup_{n\in\mathbb{N}} E_n(u_n) = C \sup_{n\in\mathbb{N}} E_{n,\mathrm{cons}}(u_n) < \infty.$$

Thanks to Lemma 10 the sequence $(u_n)_{n\in\mathbb{N}}$ is bounded in L^∞ and thus we can apply Lemma 8 to infer that $(u_n)_{n\in\mathbb{N}} \subset L^\infty(\Omega)$ is relatively compact. □

Together with Lemma 1 this finally yields our second main statement Theorem 2.

Proof of Theorem 2 Let $v \in L^\infty(\Omega)$ with $\mathcal{E}_{\text{cons}}(v) < \infty$ be arbitrary and $(v_n)_{n\in\mathbb{N}} \subset L^\infty(\Omega)$ be a recovery sequence for v, the existence of which is guaranteed by Lemma 6. By assumption, the sequence u_n satisfies

$$\begin{aligned}\limsup_{n\to\infty} E_{n,\text{cons}}(u_n) &= \limsup_{n\to\infty} \inf_{u\in L^\infty(\Omega)} E_{n,\text{cons}}(u)\\ &\le \limsup_{n\to\infty} E_{n,\text{cons}}(v_n) \le \sigma_\eta\, \mathcal{E}_{\text{cons}}(v) < \infty.\end{aligned}$$

Hence, Lemma 11 implies that $(u_n)_{n\in\mathbb{N}} \subset L^\infty(\Omega)$ is relatively compact. Lemma 1 then concludes the proof. □

5 Application to Ground States

In this section we apply the discrete-to-continuum Γ-convergence from Theorem 1 to so-called ground states, first studied in [13]. These are restricted minimizers of the functionals $E_{n,\text{cons}}$ and $\mathcal{E}_{\text{cons}}$ on L^p-spheres, where we assume that the constraint satisfies $g = 0$ on $\overline{\Omega}$. This makes the functionals $E_{n,\text{cons}}$ and $\mathcal{E}_{\text{cons}}$ absolutely 1-homogeneous.

For absolutely p-homogeneous functionals $F : X \to \mathbb{R} \cup \{\infty\}$ on a Banach space $(X, \|\cdot\|)$ with $p \in [1, \infty)$, which per definitionem satisfy

$$F(cu) = |c|^p\, F(u), \quad \forall u \in X,\ c \in \mathbb{R},$$

ground states are defined as solutions to the minimization problem

$$\min \left\{ F(u) \ :\ \inf_{v\in \arg\min F} \|u - v\| = 1 \right\}.$$

Ground states and their relations to gradient flows and power methods are well studied in the literature, see, e.g., [11,13,21,24–26]. In particular, they constitute minimizers of the nonlinear Rayleigh quotient

$$R(u) = \frac{F(u)}{\inf_{v\in\arg\min F} \|u - v\|^p}$$

and are related to nonlinear eigenvalue problems with the prime example being $F(u) = \int_\Omega |\nabla u|^p \,\mathrm{d}x$ on $X := L^\Omega$ where ground states solve the p-Laplacian eigenvalue problem

$$\lambda\, |u|^{p-2}\, u = -\Delta_p u.$$

In [13] ground states of the functionals $E_{n,\text{cons}}$ and $\mathcal{E}_{\text{cons}}$ were characterized as distance functions. While there it was assumed that $\mathcal{O} = \partial\Omega$, we will in the following generalize these results to the case of an arbitrary closed constraint set $\mathcal{O} \subset \overline{\Omega}$. Subsequently, we will use the Γ-convergence, established in Theorem 1, to show discrete-to-continuum convergence of ground states.

5.1 Relation to Distance Functions

Here, we show that the unique L^p ground states of the limit functional $\mathcal{E}_{\text{cons}}$ coincide with multiples of the geodesic distance function to the set $\mathcal{O}$. To prove the desired statement, we need the following lemma, stating that the gradient of the geodesic distance function is bounded by one.

Lemma 12 *Let $\mathcal{O} \subset \overline{\Omega}$ be a closed set and*

$$d_{\mathcal{O}}(x) := \inf_{y \in \mathcal{O}} d_{\Omega}(x, y)$$

be the geodesic distance function of $\mathcal{O}$, where $d_{\Omega}(x, y)$ denotes the geodesic distance between $x, y \in \Omega$. Then it holds

$$\|\nabla d_{\mathcal{O}}\|_{L^\infty} \leq 1.$$

Proof Let $x, y \in \Omega$ be arbitrary. Using the triangle inequality for d_{Ω} we get

$$d_{\mathcal{O}}(y) \leq d_{\Omega}(x, y) + d_{\mathcal{O}}(x), \quad \forall x, y \in \Omega.$$

If x, y lie in a ball fully contained in Ω, then $d_{\Omega}(x, y) = |x - y|$ and we obtain that $d_{\mathcal{O}}$ is Lipschitz continuous on this ball. Rademacher's theorem then implies that $\nabla d_{\mathcal{O}}$ exists almost everywhere in the ball. Since the ball is arbitrary, $\nabla d_{\mathcal{O}}$ in fact exists almost everywhere in Ω.

Furthermore, since Ω is open, for $x \in \Omega$ and $t > 0$ small enough the ball $B_t(x) := \{y \in \mathbb{R}^d : |x - y| < t\}$ lies within Ω and it holds $d_{\Omega}(x, y) = |x - y|$ for all $y \in B_t(x)$.

Choosing $y = x + ta \in B_t(x)$ with $a \in B_1(0)$ we get

$$\frac{d_{\mathcal{O}}(x + ta) - d_{\mathcal{O}}(x)}{t} \leq \frac{d_{\Omega}(x, x + ta)}{t} = \frac{|at|}{t} \leq 1.$$

Since a was arbitrary, we can conclude $|\nabla d_{\mathcal{O}}(x)| \leq 1$ for almost all $x \in \Omega$ which implies the desired statement. □

With this lemma we now can prove that the unique ground state (up to scalar multiples) of the functional $\mathcal{E}_{\text{cons}}$ is given by the geodesic distance function to $\mathcal{O}$. The only (weak) assumption which we need here is that $d_{\mathcal{O}} \in L^p(\Omega)$ which is fulfilled, for instance, if Ω has finite geodesic diameter or even only satisfies the relaxed condition (35), in which case $d_{\mathcal{O}} \in L^\infty(\Omega)$ holds.

Theorem 5 *Let $\mathcal{O}$ be a closed set such that $\overline{\Omega}\backslash\mathcal{O}$ is connected and non-empty and $d_{\mathcal{O}} \in L^p(\Omega)$, and let the constraint function satisfy $g = 0$ on $\mathcal{O}$. For $p \in [1, \infty)$ the unique solution (up to global sign) to*

$$\min \left\{ \mathcal{E}_{\text{cons}}(u) \,:\, u \in L^\infty(\Omega),\ \|u\|_{L^p} = 1 \right\} \tag{37}$$

is given by a positive multiple of the geodesic distance function $d_{\mathcal{O}}$.

If Ω is convex or $\mathcal{O} = \partial\Omega$, the geodesic distance $d_\Omega(x, y)$ in the definition of $d_{\mathcal{O}}$ can be replaced by the Euclidean $|x - y|$ and $d_{\mathcal{O}} \in L^p(\Omega)$ is always satisfied.

Proof The case $\mathcal{O} = \partial\Omega$ was already proved in [13]. If Ω is convex it holds $d_\Omega(x, y) = |x - y|$ which is and hence $d_{\mathcal{O}}$ is bounded and, in particular, lies in $L^p(\Omega)$ for all $p \geq 1$.

We first prove that the geodesic distance function $d_{\mathcal{O}}$ is a solution of

$$\max \left\{ \|u\|_{L^p} \,:\, u \in L^\infty(\Omega),\ \mathcal{E}_{\text{cons}}(u) = 1 \right\}. \tag{38}$$

Since $\mathcal{O}$ is closed and bounded, for every $x \in \Omega$ we can choose $y_x \in \mathcal{O}$ such that $d_\Omega(x, y_x) \leq d_\Omega(x, y)$ for all $y \in \mathcal{O}$. Hence, if $u = 0$ on $\mathcal{O}$, we can choose $y = y_x$ and obtain from (27) that

$$|u(x)| \leq \|\nabla u\|_{L^\infty} d_{\mathcal{O}}(x) \tag{39}$$

for almost every $x \in \Omega$ which implies

$$\|u\|_{L^p} \leq \|\nabla u\|_{L^\infty} \|d_{\mathcal{O}}\|_{L^p}. \tag{40}$$

Hence, for all $u \in L^\infty(\Omega)$ with $\mathcal{E}_{\text{cons}}(u) = 1$ one obtains from (40) that

$$\|u\|_{L^p} \leq \|d_{\mathcal{O}}\|_{L^p}.$$

From Lemma 12 we know that $\mathcal{E}_{\text{cons}}(d_{\mathcal{O}}) = \|\nabla d_{\mathcal{O}}\|_{L^\infty} \leq 1$. At the same time choosing $u = d_{\mathcal{O}}$ in (40) shows that in fact $\mathcal{E}_{\text{cons}}(d_{\mathcal{O}}) = \|\nabla d_{\mathcal{O}}\|_{L^\infty} = 1$ and therefore $d_{\mathcal{O}}$ solves (38).

Regarding uniqueness we argue as follows: Since $p < \infty$ the inequality (40) is sharp if (39) is sharp which, using that $\|\nabla u\|_{L^\infty} = \mathcal{E}_{\text{cons}}(u) = 1$, implies that all solutions u of (38) must fulfill

$$|u(x)| = d_{\mathcal{O}}(x), \quad \forall x \in \Omega.$$

If $\Omega\backslash\mathcal{O}$ is connected, the continuity of u implies that (up to global sign) $u(x) = d_{\mathcal{O}}(x)$ for all $x \in \Omega$.

Finally, we argue that (37) and (38) are equivalent: Since $d_{\mathcal{O}} \in L^p(\Omega)$ we have $\|d_{\mathcal{O}}\|_{L^p} < \infty$. Then $d_{\mathcal{O}} / \|d_{\mathcal{O}}\|_{L^p}$ solves (37) since for any $u \in L^\infty(\Omega)$ with $\|u\|_{L^p} =$

1 it holds

$$\mathcal{E}_{\text{cons}}\left(\frac{d_{\mathcal{O}}}{\|d_{\mathcal{O}}\|_{L^p}}\right) = \frac{\mathcal{E}_{\text{cons}}(d_{\mathcal{O}})}{\|d_{\mathcal{O}}\|_{L^p}} = \frac{1}{\|d_{\mathcal{O}}\|_{L^p}} = \frac{\|u\|_{L^p}}{\|d_{\mathcal{O}}\|_{L^p}} \leq \mathcal{E}_{\text{cons}}(u),$$

where we used (40) for the inequality. Analogously, if u solves (37) then

$$\mathcal{E}_{\text{cons}}(u) \leq \mathcal{E}_{\text{cons}}(d_{\mathcal{O}}/\|d_{\mathcal{O}}\|_{L^p}) = 1/\|d_{\mathcal{O}}\|_{L^p} < \infty.$$

This follows from the fact that $d_{\mathcal{O}} \neq 0$ since $\overline{\Omega}\backslash\mathcal{O} \neq \emptyset$. Then $u/\mathcal{E}_{\text{cons}}(u)$ solves (38) since, using again (40), it holds

$$\left\|\frac{u}{\mathcal{E}_{\text{cons}}(u)}\right\|_{L^p} = \frac{1}{\mathcal{E}_{\text{cons}}(u)} \geq \frac{\|d_{\mathcal{O}}\|_{L^p}}{\|u\|_{L^p}} = \|d_{\mathcal{O}}\|_{L^p}$$

and $d_{\mathcal{O}}$ solves (38). □

Remark 11 In the case $p = \infty$ the geodesic distance function is still a ground state, however, not the unique one. In this case, other ground states are given by ∞-Laplacian eigenfunctions, see, e.g., [7,28,37].

Remark 12 If one drops the condition that $\overline{\Omega}\backslash\mathcal{O}$ is connected, ground states coincide with (positive or negative) multiples of the distance function on each connected component of $\overline{\Omega}\backslash\mathcal{O}$.

Similarly, one can also prove that ground states of the discrete functionals $E_{n,\text{cons}}$ coincide with multiples of distance functions to $\mathcal{O}_n$ with respect to the geodesic graph distance if $\Omega_n\backslash\mathcal{O}_n$ is connected in the graph-sense. The result can be found in [13]; however, since we do not need it here, we refrain from stating it.

5.2 Convergence of Ground States

In this section we first show that the Γ-convergence of the functionals $E_{n,\text{cons}}$ to $\sigma_\eta\mathcal{E}_{\text{cons}}$ implies the convergence of their respective ground states. Together with the characterization from Theorem 5 this implies that discrete ground states converge to the geodesic distance function.

Theorem 6 (Convergence of Ground States) *Under the conditions of Theorem 1 let the sequence $(u_n)_{n\in\mathbb{N}} \subset L^\infty(\Omega)$ fulfill*

$$u_n \in \arg\min\left\{E_{n,\text{cons}}(u) \,:\, u \in L^\infty(\Omega),\ \|u\|_{L^p} = 1\right\}.$$

Then (up to a subsequence) $u_n \to u$ in $L^\infty(\Omega)$ where

$$u \in \arg\min\left\{\mathcal{E}_{\text{cons}}(u) \,:\, u \in L^\infty(\Omega),\ \|u\|_{L^p} = 1\right\}$$

and it holds

$$\lim_{n\to\infty} E_{n,\mathrm{cons}}(u_n) = \sigma_\eta\, \mathcal{E}_{\mathrm{cons}}(u). \tag{41}$$

Proof Let $u \in L^\infty(\Omega)$ be a ground state of $\mathcal{E}_{\mathrm{cons}}$ and $(v_n)_{n\in\mathbb{N}} \subset L^\infty(\Omega)$ be a recovery sequence of u, whose existence is guaranteed by Theorem 1. Since u_n is a ground state and $E_{n,\mathrm{cons}}$ is absolutely 1-homogeneous, we get

$$E_{n,\mathrm{cons}}(u_n) \le E_{n,\mathrm{cons}}\left(\frac{v_n}{\|v_n\|_{L^p}}\right) = E_{n,\mathrm{cons}}(v_n)\frac{1}{\|v_n\|_{L^p}}.$$

Taking the limsup on both sides yields

$$\limsup_{n\to\infty} E_{n,\mathrm{cons}}(u_n) \le \sigma_\eta\, \mathcal{E}_{\mathrm{cons}}(u)\frac{1}{\|u\|_{L^p}} = \sigma_\eta\, \mathcal{E}_{\mathrm{cons}}(u) < \infty,$$

where we used boundedness of Ω to conclude that L^∞-convergence implies convergence of the L^p-norms. Hence, by Lemma 11 the sequence $(u_n)_{n\in\mathbb{N}}$ possesses a subsequence (which we do not relabel) which converges to some $u^* \in L^\infty(\Omega)$ with $\|u^*\|_{L^p} = 1$. Using the previous inequality, the liminf inequality from Lemma 5, and the fact that u is a ground state we conclude

$$\begin{aligned}
\sigma_\eta\, \mathcal{E}_{\mathrm{cons}}(u^*) &\le \liminf_{n\to\infty} E_{n,\mathrm{cons}}(u_n)\\
&\le \limsup_{n\to\infty} E_{n,\mathrm{cons}}(u_n)\\
&\le \sigma_\eta\, \mathcal{E}_{\mathrm{cons}}(u)\\
&\le \sigma_\eta\, \mathcal{E}_{\mathrm{cons}}(u^*).
\end{aligned}$$

Hence, u^* is also a ground state and (41) holds true. □

Using the characterization of ground states as distance functions we obtain the following

Corollary 1 *Under the conditions of Theorems 5 and 6 the sequence* $(u_n)_{n\in\mathbb{N}} \subset L^\infty(\Omega)$*, given by*

$$u_n \in \arg\min\left\{E_{n,\mathrm{cons}}(u) \,:\, u \in L^\infty(\Omega),\ \|u\|_{L^p} = 1\right\},$$

converges to a multiple of the geodesic distance function $d_{\mathcal{O}}$.

6 Conclusion and Future Work

In this work we derived continuum limits of semi-supervised Lipschitz learning on graphs. We first proved Γ-convergence of non-local functionals to the supremal norm of the gradient. This allowed us to show Γ-convergence of the discrete energies which

appear in the Lipschitz learning problem. In order to interpret graph functions as functions defined on the continuum, we employed a closest point projection. We also showed that the discrete functionals are compact which implies discrete-to-continuum convergence of minimizers. We applied our results to a nonlinear eigenvalue problem whose solutions are geodesic distance functions.

Future work will include the generalization of our results to general metric measure spaces or Riemannian manifolds, which constitute a generic domain for the data in real-world semi-supervised learning problems. Furthermore, we intend to see how the application of our results to absolutely minimizing Lipschitz extensions [5] on graphs unfolds. Namely, we want to gain insight whether it is possible to prove their convergence toward solutions of the infinity Laplacian equation under less restrictive assumptions than the ones used in [15].

Acknowledgements This work was supported by the European Unions Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 777826 (NoMADS). The work of TR was supported by the German Ministry of Science and Technology (BMBF) under grant 05M2020 -DELETO. LB acknowledges funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - GZ 2047/1, Projekt-ID 390685813.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

1. Robert A. Adams and John Fournier. Sobolev Spaces, Volume 140 (Pure and Applied Mathematics). Academic Press, New York, 2003.
2. Ahmed El Alaoui, Xiang Cheng, Aaditya Ramdas, Martin J. Wainwright, and Michael I. Jordan. Asymptotic behavior of l_p-based Laplacian regularization in semi-supervised learning. 2016. arxiv: 1603.00564.
3. Gunnar Aronsson. Minimization problems for the functional $\sup_x F(x, f(x), f'(x))$. In: Arkiv för Matematik 6.1 (1965), pp. 33–53. https://doi.org/10.1007/bf02591326.
4. Gunnar Aronsson. On the partial differential equation $u_x^2 u_{xx} + 2u_x u_y u_{xy} + u_y^2 u_{yy} = 0$. In: Arkiv för Matematik 7.5 (1968), pp. 395–425. https://doi.org/10.1007/bf02590989.
5. Gunnar Aronsson, Michael Crandall, and Petri Juutinen. A tour of the theory of absolutely minimizing functions. In: Bulletin of the American Mathematical Society 41.4 (2004), pp. 439–505.
6. Vladimir I. Bogachev. Measure Theory. Springer, Berlin, Heidelberg, 2007. https://doi.org/10.1007/978-3-540-34514-5.
7. Farid Bozorgnia, Leon Bungert, and Daniel Tenbrinck. The Infinity Laplacian eigenvalue problem: reformulation and a numerical scheme. 2020. arXiv: 2004.08127 [math.NA].
8. Andrea Braides. Gamma-convergence for Beginners. Vol. 22. Oxford University Press, Oxford, 2002.
9. Haim Brezis. Functional Analysis, Sobolev Spaces and Partial Differential Equations. Springer, New York, 2010. https://doi.org/10.1007/978-0-387-70914-7.
10. Martin R. Bridson and André Haefliger. Metric Spaces of Non-Positive Curvature. Springer, Berlin, Heidelberg, 1999. https://doi.org/10.1007/978-3-662-12494-9.

11. Leon Bungert and Martin Burger. Asymptotic profiles of nonlinear homogeneous evolution equations of gradient flow type. In: Journal of Evolution Equations 20.3 (2020), pp. 1061–1092. https://doi.org/10.1007/s00028-019-00545-1.
12. Leon Bungert, Jeff Calder, and Tim Roith. Uniform Convergence Rates for Lipschitz Learning on Graphs. 2021. arXiv: 2111.12370 [math.NA].
13. Leon Bungert, Yury Korolev, and Martin Burger. Structural analysis of an L-infinity variational problem and relations to distance functions. Pure and Applied Analysis 2.3 (2020), pp. 703–738. https://doi.org/10.2140/paa.2020.2.703.
14. Jeff Calder. Consistency of Lipschitz Learning with Infinite Unlabeled Data and Finite Labeled Data. In: SIAM Journal on Mathematics of Data Science 1.4 (2019), pp. 780–812. https://doi.org/10.1137/18m1199241.
15. Jeff Calder, Brendan Cook, Matthew Thorpe, and Dejan Slepčev. Poisson Learning: Graph Based semi-supervised learning at very low label rates. In: International Conference on Machine Learning. PMLR. 2020, pp. 1306–1316.
16. Jeff Calder, Dejan Slepčev, and Matthew Thorpe. Rates of Convergence for Laplacian Semi-Supervised Learning with Low Labeling Rates. 2020. arXiv: 2006.02765 [math.ST].
17. Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. Semi-Supervised Learning. Cambridge, MA: The MIT Press, 2010.
18. Paul Civin, Nelson Dunford, and Jacob T. Schwartz. Linear Operators. Part I: General Theory. American Mathematical Monthly 67 (1960), p. 199.
19. Abderrahim Elmoataz, Matthieu Toutain, and Daniel Tenbrinck. On the p-Laplacian and ∞-Laplacian on Graphs with Applications in Image and Data Processing. In: SIAM Journal on Imaging Sciences 8.4 (2015), pp. 2412–2451. https://doi.org/10.1137/15m1022793.
20. Lawrence C Evans and Charles K Smart. Everywhere differentiability of infinity harmonic functions. Calculus of Variations and Partial Differential Equations 42.1-2 (2011), pp. 289–299.
21. Tal Feld, Jean-François Aujol, Guy Gilboa, and Nicolas Papadakis. Rayleigh quotient minimization for absolutely one-homogeneous functionals. Inverse Problems 35.6 (2019), p. 064003.
22. Nicolás García Trillos and Dejan Slepčev. Continuum Limit of Total Variation on Point Clouds. In: Archive for Rational Mechanics and Analysis 220.1 (2015), pp. 193–241. https://doi.org/10.1007/s00205-015-0929-z.
23. Yves van Gennip and Andrea L. Bertozzi. Gamma-convergence of graph Ginzburg–Landau functionals. In: Advances in Differential Equations 17.11/12 (2012), pp. 1115–1180.
24. Ryan Hynd and Erik Lindgren. Inverse iteration for p-ground states. In: Proceedings of the American Mathematical Society 144.5 (2016), pp. 2121– 2131. https://doi.org/10.1090/proc/12860.
25. Ryan Hynd and Erik Lindgren. Approximation of the least Rayleigh quotient for degree p homogeneous functionals. In: Journal of Functional Analysis 272.12 (2017), pp. 4873–4918. https://doi.org/10.1016/j.jfa.2017.02.024.
26. Ryan Hynd and Erik Lindgren. Extremal functions for Morrey's inequality in convex domains. In: Mathematische Annalen 375.3-4 (2019), pp. 1721–1743. https://doi.org/10.1007/s00208-018-1775-8.
27. Petri Juutinen. Absolutely minimizing Lipschitz extensions on a metric space. In: Annales Academiae Scientiarum Fennicae Mathematica Volumen 27 (Jan. 2002), pp. 57–67.
28. Petri Juutinen, Peter Lindqvist, and Juan J Manfredi. The infinity Laplacian: examples and observations. Institut Mittag-Leffler, 1999.
29. Peter Knabner and Lutz Angermann. Numerical Methods for Elliptic and Parabolic Partial Differential Equations. Springer, Berlin, Heidelberg, 2003. https://doi.org/10.1007/b97419.
30. Rasmus Kyng, Anup Rao, Sushant Sachdeva, and Daniel A Spielman. Algorithms for Lipschitz learning on graphs. In: Conference on Learning Theory. PMLR. 2015, pp. 1190–1223.
31. Erwan Le Gruyer. On absolutely minimizing Lipschitz extensions and PDE $\Delta_\infty(u) = 0$. In: Nonlinear Differential Equations and Applications NoDEA 14.1 (2007), pp. 29–55.
32. Ulrike von Luxburg, Mikhail Belkin, and Olivier Bousquet. Consistency of spectral clustering. In: The Annals of Statistics 36.2 (2008), pp. 555–586. https://doi.org/10.1214/009053607000000640.
33. David Pollard. Strong Consistency of k-Means Clustering. In: The Annals of Statistics 9.1 (1981), pp. 135–140. https://doi.org/10.1214/aos/1176345339.
34. Scott Sheffield and Charles K. Smart. Vector-valued optimal Lipschitz extensions. In: Communications on Pure and Applied Mathematics 65.1 (2012), pp. 128–154.

35. Dejan Slepčev and Matthew Thorpe. Analysis of p-Laplacian Regularization in Semisupervised Learning. In: SIAM Journal on Mathematical Analysis 51.3 (2019), pp. 2085–2120. https://doi.org/10.1137/17m115222x.
36. Nicolás García Trillos, Dejan Slepčev, James von Brecht, Thomas Laurent, and Xavier Bresson. Consistency of Cheeger and Ratio Graph Cuts. In: J. Mach. Learn. Res. 17.1 (2016), pp. 6268–6313.
37. Yifeng Yu. Some properties of the ground states of the infinity Laplacian. In: Indiana University Mathematics Journal (2007), pp. 947–964.
38. Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. In: ICML'03 (2003), pp. 912–919.
39. Xiaojin Zhu and Andrew B. Goldberg. Introduction to semi-supervised learning. In: Synthesis lectures on artificial intelligence and machine learning 3.1 (2009), pp. 1–130.
40. Xiaojin Zhu, John Lafferty, and Ronald Rosenfeld. Semi-supervised learning with graphs. PhD thesis. Carnegie Mellon University, language technologies institute, school of computer science, 2005.
41. Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Tech. rep. University of Wisconsin-Madison Department of Computer Sciences, 2005.

Print P2
Uniform convergence rates for Lipschitz learning on graphs

L. Bungert, J. Calder, and T. Roith
arXiv preprint arXiv:2111.12370 (2021)

Uniform Convergence Rates for Lipschitz Learning on Graphs

Leon Bungert * Jeff Calder † Tim Roith ‡

March 21, 2023

Abstract

Lipschitz learning is a graph-based semi-supervised learning method where one extends labels from a labeled to an unlabeled data set by solving the infinity Laplace equation on a weighted graph. In this work we prove uniform convergence rates for solutions of the graph infinity Laplace equation as the number of vertices grows to infinity. Their continuum limits are absolutely minimizing Lipschitz extensions with respect to the geodesic metric of the domain where the graph vertices are sampled from. We work under very general assumptions on the graph weights, the set of labeled vertices, and the continuum domain. Our main contribution is that we obtain quantitative convergence rates even for very sparsely connected graphs, as they typically appear in applications like semi-supervised learning. In particular, our framework allows for graph bandwidths down to the connectivity radius. For proving this we first show a quantitative convergence statement for graph distance functions to geodesic distance functions in the continuum. Using the "comparison with distance functions" principle, we can pass these convergence statements to infinity harmonic functions and absolutely minimizing Lipschitz extensions.

Key words: Lipschitz learning, graph-based semi-supervised learning, continuum limit, absolutely minimizing Lipschitz extensions, infinity Laplacian
AMS subject classifications: 35J20, 35R02, 65N12, 68T05

Contents

*Hausdorff Center for Mathematics, University of Bonn, Endenicher Allee 62, Villa Maria, 53115 Bonn, Germany. leon.bungert@hcm.uni-bonn.de

†School of Mathematics, University of Minnesota, 127 Vincent Hall, 206 Church St. S.E., Minneapolis, MN 55455, USA. jwcalder@umn.edu

‡Department of Mathematics, University of Erlangen–Nürnberg, Cauerstraße 11, 91058 Erlangen, Germany. tim.roith@fau.de

1 Introduction

The last few years have seen a deluge of discrete to continuum convergence results for various problems in graph-based learning. This theory makes connections between discrete machine learning and continuum partial differential equations or variational problems, leading to new insights and better algorithms. Current works use either a Gamma-convergence framework [20, 37, 38, 41], sometimes leading to convergence rates in energy norms (i.e., L^2), or PDE techniques like the maximum principle [4, 5, 10, 19, 45], which give uniform convergence rates, albeit with more restrictive conditions on the graph bandwidth that often rule out the sparse graphs used in practice. The different problems considered can be structured into consistency of spectral clustering—e.g., the convergence of eigenvalues and eigenvectors of graph Laplacian operators [7, 8, 18, 28]—and of semi-supervised learning problems, where one is interested in convergence of solutions to boundary value problems of graph Laplacian operators, e.g., [4, 5, 6, 9, 37, 38].

In this work we focus on graph-based semi-supervised learning. Given a large data set Ω_n with $n \in \mathbb{N}$ elements, the general task consists in extending labels $g : \mathcal{O}_n \to \mathbb{R}$ from a (typically much smaller) set of labeled data points $\mathcal{O}_n \subset \Omega_n$ to the rest of the data set.[1] To include information about the unlabeled data into the problem, one builds a graph $G_n := (\Omega_n, w_n)$ where w_n denotes a weight function. The *semi-supervised smoothness assumption* (see, e.g., [40]) then demands that data points with high similarity, i.e., connected with an edge of large weight, carry similar labels. To enforce this, the method of Laplacian learning [46] has been developed which requires solving the graph Laplace equation on G_n with prescribed boundary values on $\mathcal{O}_n$. Since this method behaves poorly if the label set $\mathcal{O}_n$ is small [33], the more general framework of p-Laplacian learning [14] was suggested, which mitigates this drawback for sufficiently large values of $p \in (1, \infty)$ [17]. It consists in solving the following nonlinear graph p-Laplacian equation (see [15] for details on the graph p-Laplacian)

$$(\forall x \in \Omega_n \backslash \mathcal{O}_n) \sum_{y \in \Omega_n} w_n(x,y) \, |u(x) - u(y)|^{p-2} \, (u(x) - u(y)) = 0, \text{ subject to } u = g \text{ on } \mathcal{O}_n. \tag{1.1}$$

The underlying reason why p-Laplacian learning outperforms its special case $p = 2$ is the Sobolev embedding $W^{1,p} \hookrightarrow C^{0,1-\frac{d}{p}}$, which implies that only for $p > d$, where d is the ambient space dimension, it is meaningful to prescribe boundary values on the label set $\mathcal{O}_n$ in the limit $n \to \infty$. Since the dimension d of the data can be quite large it seems canonical to consider the limit of p-Laplacian learning as $p \to \infty$. The resulting

[1] Generally the labels are vector-valued, in $\mathbb{R}^k$ if there are k classes, but this can be reduced to the scalar case with the *one-versus-rest* approach to multi-class classification [32].

method is called *Lipschitz learning* [24] and takes the form

$$(\forall x \in \Omega_n \backslash \mathcal{O}_n) \ \min_{y \in \Omega_n} w_n(x,y)(u(y)-u(x)) + \max_{y \in \Omega_n} w_n(x,y)(u(y)-u(x)) = 0, \text{ subject to } u = g \text{ on } \mathcal{O}_n. \tag{1.2}$$

Note that solutions to this problem are *absolutely minimizing Lipschitz extensions* of g [3, 22]. This means that they extend the labels g from $\mathcal{O}_n$ to Ω_n without increasing the (graph) Lipschitz constant and, in addition, are locally minimal in the sense that their Lipschitz constant cannot be reduced on any subset of Ω_n. The operator on the left hand side in (1.2) is referred to as graph infinity Laplacian since it arises as the limit of the graph p-Laplacian operators in (1.1). It has been established experimentally that Lipschitz learning performs very similarly to p-Laplacian learning for $d < p < \infty$ [17], and so it has the added benefit over p-Laplace learning of having one less parameter, p, to tune. In addition, while Lipschitz learning *forgets* the distribution of the labeled data [5, 14, 38], which is generally undesirable for semi-supervised learning, it is possible to reweight the graph to introduce a strong dependence on the data density [5].

The importance of Lipschitz-based algorithms in semi-supervised learning has already been observed in [27] and in [24] algorithms for solving the Lipschitz learning problem have been proposed (see also [17, 34]). In [5] a continuum limit for (1.2) in the setting of geometric graphs was proved by one author of the present paper. For $w_n(x,y) := \eta(|x-y|/h_n)$ with $h_n > 0$ being a graph length scale and $\eta : [0,\infty) \to [0,\infty)$ a sufficiently well-behaved kernel profile, graph vertices contained in the flat torus, i.e., $\Omega_n \subset \mathbb{T}^d$, and a fixed label set $\mathcal{O}_n = \mathcal{O}$ it was shown in [5] that as $n \to \infty$ solutions of (1.2) uniformly converge to the viscosity solution of the infinity Laplacian equation

$$\Delta_\infty u = 0 \text{ on } \mathbb{T}^d \backslash \mathcal{O} \quad \text{subject to } u = g \text{ on } \mathcal{O}. \tag{1.3}$$

Here, for a smooth function u the infinity Laplacian is defined as

$$\Delta_\infty u := \langle \nabla u, D^2 u \nabla u \rangle = \sum_{i,j=1}^{d} \partial_i u \, \partial_j u \, \partial_{ij}^2 u. \tag{1.4}$$

The main hypothesis for the proof in [5] is that the graph length scale h_n is sufficiently large compared to the resolution δ_n (see Section 2.2 for the definition) of the graph; in particular, it is assumed that

$$\lim_{n\to\infty} \frac{\delta_n^2}{h_n^3} = 0 \tag{1.5}$$

holds. Furthermore, the viscosity solutions techniques require that the kernel function η is sufficiently smooth. In the related work [37] by the two other authors of the present paper, continuum limits for general Lipschitz extensions on geometric graphs are proved using Gamma-convergence of the graph Lipschitz constant functional $u \mapsto \max_{x,y\in\Omega_n} w_n(x,y)\,|u(x)-u(y)|$ to the L^∞-norm of the gradient $u \mapsto \|\nabla u\|_{L^\infty(\Omega)}$ in a suitable L^∞-type topology. Note that minimizers of these functionals, satisfying label constraints on Ω_n and Ω, respectively, are not unique. In particular, the Gamma-convergence result does not imply the convergence to *absolute minimizers* of the continuum problem in a straightforward way. Furthermore, establishing non-asymptotic convergence rates using Gamma-convergence, which by definition is of asymptotic flavor, is a difficult endeavour. Still, the approach in [37] opened the door to a much more general analysis of the Lipschitz learning problem. In particular, in [37] the weakest possible scaling assumption

$$\lim_{n\to\infty} \frac{\delta_n}{h_n} = 0, \tag{1.6}$$

which ensures that the graph G_n is connected, was already sufficient to prove Gamma-convergence of the Lipschitz-constant functional. Furthermore, the theory in [37] applies to more general kernels η and a large class of continuum domains. The paper [37] was also the first to work with general (non-fixed) label sets $\mathcal{O}_n$ in a Lipschitz learning context (see [10] for a corresponding approach for p-Laplace learning (1.1) with $p = 2$).

Note that the weakest scaling assumption (1.6) has already appeared in [25] where discrete to continuum convergence of absolutely minimizing Lipschitz extensions on unweighted graphs was proved.

When it comes to convergence rates of solutions to (1.2) to a suitable continuum problem much less is known. To the best of our knowledge the only contribution in this direction is [39], where convergence rates are proved under very special conditions: the graph is unweighted and assumed to be a regular grid. Furthermore, the author works with Dirichlet boundary conditions and requires the even stricter scaling assumption

$$\lim_{n\to\infty} \frac{\delta_n}{h_n^2} = 0. \tag{1.7}$$

The main motivational factors for the present paper are therefore the following:

- We would like to prove convergence rates for (1.2) which are valid down to the smallest scaling (1.6).
- The labeled set $\mathcal{O}_n$ should be completely free, in particular, it should *not* have to approximate the boundary of Ω, which is not realistic for semi-supervised learning.
- We would like to work under minimal assumptions on the weight function η and the continuum domain $\Omega \subset \mathbb{R}^d$ from which the graph vertices in Ω_n are drawn.

Let us reiterate that, while the previous scaling conditions (1.5), (1.6), and (1.7) might look very similar, they in fact have strong impacts on the numerical complexity of the problem (1.2). To give a concrete example, let Ω_n coincide with a square grid of the hypercube $[0,1]^d$. In this case it holds that $\delta_n \sim 1/n^{1/d}$ and suitable graph length scales have the form $h_n = 1/n^\alpha$ where $\alpha \in (0, 1/d)$ for the weakest scaling (1.6), $\alpha \in (0, 2/(3d))$ for the scaling (1.5), and $\alpha \in (0, 1/(2d))$ for the largest scaling (1.7). Correspondly, the degree of a graph vertex, i.e., the number of neighbors which have to be considered when solving (1.2), scale like $nh_n^d = n^{1-\alpha d}$. For $n = 10,000$ vertices the size of these computational stencils would scale like 5, 100, and 500 for the different scalings. These are dramatic differences in numerical complexity, and the differences grow larger as n increases (e.g., at $n = 10^6$ points there is a 1,000-fold difference in sparsity between the weakest and strongest conditions). Furthermore, in applications of graph-based learning in practice, it is common to use very sparse graphs (typically k-nearest neighbor graphs) that operate slightly above the graph connectivity threshold [42]. These observations justify the need for analysis and convergence rates that hold in the sparsest connectivity regime.

Let us briefly discuss where the restrictive length scale conditions (1.5) and (1.7) arise in discrete to continuum convergence, and how the techniques used in this paper overcome these issues. Both of these conditions stem from the use of pointwise consistency of the graph infinity Laplacian given in (1.2) with the continuous infinity Laplacian Δ_∞, which is one of the main steps required for uniform convergence rates. Pointwise consistency results normally have two steps, one in which we pass to a nonlocal operator by controlling the randomness (or variance) in the graph construction, and a second step that uses Taylor expansion to pass from a nonlocal operator to a local PDE operator. For Lipschitz learning (1.2) with geometric weights $w_n(x,y) = \eta(|x-y|/h_n)$, the corresponding nonlocal equation (see [11] for analytical results on such a nonlocal infinity Laplacian equation) is obtained by replacing the discrete set Ω_n with the continuum domain Ω, yielding

$$\min_{y\in\Omega} \eta\left(\frac{|x-y|}{h_n}\right)(u(y)-u(x)) + \max_{y\in\Omega} \eta\left(\frac{|x-y|}{h_n}\right)(u(y)-u(x)) = 0. \tag{1.8}$$

To control the difference between the discrete and nonlocal operators, we need to prove estimates of the form

$$\max_{y\in\Omega_n} w_n(x,y)(u(y)-u(x)) = \max_{y\in\Omega} \eta\left(\frac{|x-y|}{h_n}\right)(u(y)-u(x)) + O(r_n), \tag{1.9}$$

where r_n depends on the regularity of η and u. If η and u are Lipschitz continuous, then the function

$$y \mapsto \eta\left(\frac{|x-y|}{h_n}\right)(u(y)-u(x)) \tag{1.10}$$

is Lipschitz continuous with Lipschitz constant independent of $h_n > 0$, provided η has compact support so that we can use the bound $|u(y) - u(x)| \leq Ch_n$. It then follows that $r_n = \delta_n$ is exactly the resolution of the point cloud Ω_n (see Section 2.2 for the definition). We can of course repeat the same argument for the minimum term in (1.8). When Taylor expanding $u(y) - u(x)$ to obtain consistency to Δ_∞, the second order terms are of size $O(h_n^2)$, so we have to divide both side of the nonlocal equation (1.8) by h_n^2 in order to obtain consistency with the infinity Laplace equation $\Delta_\infty u = 0$, leading to pointwise consistency truncation errors of the form $O\left(\delta_n h_n^{-2}\right)$. The condition that this term converges to zero is exactly the strongest length scale condition (1.7). When η and u are C^2, we can improve the error in (1.9) to be $r_n = \delta_n^2 h_n^{-1}$, which follows from examining the Hessian of the mapping in (1.10) (see [5, Lemma 4.1]). This leads to the pointwise consistency error term $O(\delta_n^2 h_n^{-3})$, which corresponds to the intermediate condition (1.5). This condition was suitable in [5] since the viscosity solution framework only requires pointwise consistency for smooth functions u, though no convergence rate can be established in this way. In addition to the restrictive length scale conditions on h_n required for pointwise consistency, the kernel η is also required to be Lipschitz for (1.7) and C^2 for (1.5), and in either case, the restrictive condition that $t \mapsto t\eta(t)$ has a unique maximum near which it must be strongly concave, is needed to pass from nonlocal to local (see [17, Section 2.1]).

In the present approach we overcome these limitations by introducing a nonlocal infinity Laplacian on a homogenized length scale $\varepsilon > h_n$. The homogenized operator has the form

$$\Delta_\infty^\varepsilon u(x) = \frac{1}{\varepsilon^2}\left(\inf_{y \in B_\varepsilon(x)}(u(y) - u(x)) + \sup_{y \in B_\varepsilon(x)}(u(y) - u(x))\right) \tag{1.11}$$

and notably the effect of the kernel function η has been "homogenized out". Although sometimes hidden, this operator frequently appears in the analysis of equations involving the infinity Laplacian (cf. [1, 3, 26, 29, 30, 35, 36]). Most prominently, in [1] it was used as intermediate step in a very simple and elementary proof of the maximum principle for the infinity Laplace equation. There, the key ingredient is the astonishing property that whenever $-\Delta_\infty u \leq 0$, the function $u^\varepsilon(x) := \sup_{B_\varepsilon(x)} u$ satisfies $-\Delta_\infty^\varepsilon u^\varepsilon \leq 0$ for any $\varepsilon > 0$. For proving this one only utilizes that u satisfies the celebrated "comparison with cones" property [3] which in our more general setting is rather a comparison with distance functions [12].

A key step in our approach is an approximate version of this statement. We show that a graph infinity harmonic function can be extended to a continuum function u_n^ε that satisfies

$$-\Delta_\infty^\varepsilon u_n^\varepsilon(x) \leq O\left(\frac{\delta_n}{h_n \varepsilon} + \frac{h_n}{\varepsilon^2}\right).$$

Compared to the truncation error of $O(\delta_n h_n^{-2})$ from [39] we have improved by replacing one h_n by the (asymptotically larger) nonlocal length scale ε. For this we pay the price of having an additional $h_n \varepsilon^{-2}$ term which, however, does not contribute to the error for the small length scales h_n that we are interested in. The convergence rates then follow by an application of the maximum principle for the operator $\Delta_\infty^\varepsilon$ in combination with some (approximate) Lipschitz estimates to go back from u_n^ε to the graph solution.

The main caveat in proving our results is that we are dealing with three different metrics that have to be put into relation with each other. First, there is the Euclidean metric of the ambient space which enters through the graph weights and quantifies the approximation of the continuum domain by the graph. Second, there is the geodesic metric which is inherited from the Euclidean one and constitutes the natural metric in the (possibly non-convex) domain Ω. Finally, there is also the graph metric, which describes shortest paths in the graph and is inherently connected to the Lipschitz learning problem (1.2). For proving our continuum limit we need that all three metrics behave similarly on locally scales. While it is straightforward to show that the graph metric approximates the geodesic metric, we have to assume that the latter also approximates the Euclidean metric locally. This we ensure by posing a very mild regularity condition on the continuum domain Ω which prevents internal corners, where the geodesic metric has a constant discrepancy to the Euclidean one. In particular, this allows us to work with arbitrary (non-smooth) convex domains and also with non-convex domains whose boundaries are smooth in non-convex regions.

The rest of this paper is organized as follows: Section 2 summarizes the paper by stating all assumptions, the precise settings for the discrete and continuum equations and the passage between these two worlds,

and our main results. Subsequently, we investigate both the discrete and continuum problems in much more detail. Section 3 introduces weighted graphs, proves that infinity harmonic functions on a graph satisfy comparison with graph distance functions, and are absolutely minimizing Lipschitz extensions. In Section 4 we study the continuum limit, namely absolutely minimizing Lipschitz extensions (AMLEs). Most importantly, in Section 4.3 we show that AMLEs give rise to sub- and supersolutions of a nonlocal infinity Laplace equation and collect several perturbation results for this equation which were proved in [39]. The main part of the paper is Section 5 where we first prove convergence rates of graph distance function to geodesic ones (Section 5.2) and then use these results to establish convergence rates for graph infinity harmonic functions (Section 5.3). Finally, in Section 6 we perform some numerical experiments where we compute empirical rates of convergence of AMLEs on a non-convex domain.

2 Setting, Assumptions, and Main Results

To simplify the distinction between discrete and continuum, we will from now on denote general points in the continuum domain $\overline{\Omega}$ in regular font x, y, z, etc., while we will denote points specifically belonging to the point cloud Ω_n with bold font $\mathbf{x}, \mathbf{y}, \mathbf{z}$, etc. We will use the same notation for functions on $\overline{\Omega}$, denoted, for example, as $u, v, w : \overline{\Omega} \to \mathbb{R}$, while we will again use bold notation and an index n for functions defined solely on the point cloud Ω_n, i.e., $\mathbf{u}_n, \mathbf{v}_n, \mathbf{w}_n : \Omega_n \to \mathbb{R}$.

2.1 The Discrete Problem

We let Ω_n be a point cloud and $\mathcal{O}_n \subset \Omega_n$ be a set of labelled vertices. We construct a graph with vertices Ω_n in the following way: Let $h > 0$ be a graph length scale, let $\eta : [0, \infty) \to [0, \infty)$ be a function satisfying Assumption 1 below, and define the rescaled kernel as $\eta_h(t) = \frac{1}{\sigma_\eta h}\eta\left(\frac{t}{h}\right)$, where $\sigma_\eta = \sup_{t\geq 0} t\eta(t)$.

Assumption 1. The function $\eta : (0, \infty) \to [0, \infty)$ is nonincreasing with $\operatorname{supp}\eta \subset [0, 1]$. Furthermore, $\sigma_\eta := \sup_{t>0} t\eta(t) \in (0, \infty)$ and we assume that there exists $t_0 \in (0, 1]$ (chosen as the largest number with this property) with $\sigma_\eta = t_0\eta(t_0)$.

Example 1. A couple of popular kernel functions η which satisfy Assumption 1 are given below:

- (constant weights) $\eta(t) = 1_{[0,1]}(t)$,
- (exponential weights) $\eta(t) = \exp(-t^2/(2\sigma^2))1_{[0,1]}(t)$,
- (non-integrable weights) $\eta(t) = \frac{1}{t^p}1_{[0,1]}(t)$ with $p \in (0, 1]$.

Note that the assumption that $t_0 > 0$ achieves the supremum is not redundant. Indeed, for the kernel

$$\eta(t) = \frac{1}{t^{\frac{1}{t-1}+2}}1_{[0,1]}(t), \quad t > 0,$$

the supremum of $t \mapsto t\eta(t)$ is achieved at $t_0 = 0$ although $\sigma_\eta = 1$.

Using the kernel function, the edge connecting two vertices $\mathbf{x}, \mathbf{y} \in \Omega_n$ carries the weight $\eta_h(|\mathbf{x} - \mathbf{y}|)$ whenever $\mathbf{x} \neq \mathbf{y}$ and zero otherwise. Consequently, in the whole paper we use the convention that $\eta(0)\cdot 0 := 0$ even though η is not defined in 0. Of course, the weight between two vertices $\mathbf{x}, \mathbf{y} \in \Omega_n$ is also zero whenever $|\mathbf{x} - \mathbf{y}| > h$, in which case we regard the vertices as not connected. The number $h > 0$ can be interpreted as characteristic connectivity length scale of the graph.

The main objective of this paper is to prove convergence rates for solutions to the graph infinity Laplacian equation to solutions of the corresponding continuum problem. To state the discrete problem we let $\mathbf{g}_n : \mathcal{O}_n \to \mathbb{R}$ be a labelling function and $\mathbf{u}_n : \Omega_n \to \mathbb{R}$ a solution of the graph infinity Laplacian equation

$$\begin{cases} \min_{\mathbf{y}\in\Omega_n} \eta_h(|\mathbf{x} - \mathbf{y}|)\,(\mathbf{u}_n(\mathbf{y}) - \mathbf{u}_n(\mathbf{x})) + \max_{\mathbf{y}\in\Omega_n} \eta_h(|\mathbf{x} - \mathbf{y}|)\,(\mathbf{u}_n(\mathbf{y}) - \mathbf{u}_n(\mathbf{x})) = 0, & \mathbf{x} \in \Omega_n \setminus \mathcal{O}_n, \\ \mathbf{u}_n(\mathbf{x}) = \mathbf{g}_n(\mathbf{x}), & \mathbf{x} \in \mathcal{O}_n. \end{cases} \tag{2.1}$$

The solution of this problem (see [5] for existence and uniqueness) is characterized by the fact that it is an absolutely minimizing Lipschitz extension of the labelling function $\mathbf{g}_n$ in the sense that

$$\mathrm{Lip}_n(\mathbf{u}_n; A) = \mathrm{Lip}_n(\mathbf{u}_n; \partial A), \quad \forall A \subset \Omega_n. \tag{2.2}$$

Here $\mathrm{Lip}_n(\mathbf{u}_n; A)$ is a discrete Lipschitz constant of the graph function $\mathbf{u}_n$ on a subset of vertices $A \subset \Omega_n$ with graph boundary ∂A. We refer to Section 3 for precise definitions and a proof of this statement in an abstract graph setting. In particular, not only do we have $\mathrm{Lip}_n(\mathbf{u}_n; \Omega_n) = \mathrm{Lip}_n(\mathbf{g}_n; \mathcal{O}_n)$, which would be satisfied by any Lipschitz extension, but the Lipschitz constant of $\mathbf{u}_n$ is also locally minimal.

2.2 Discrete to Continuum

To set the scene for a discrete to continuum study, we let $\Omega \subset \mathbb{R}^d$ be an open and bounded domain and assume that the point cloud Ω_n is sampled from $\overline{\Omega}$ and that the constraint set $\mathcal{O}_n$ approximates a closed set $\mathcal{O} \subset \overline{\Omega}$. Of course, we need to quantify how well Ω_n and $\mathcal{O}_n$ approximate their continuum counterparts Ω and $\mathcal{O}$. For this we introduce the Hausdorff distance of two sets $A, B \subset \mathbb{R}^d$:

$$d_H(A, B) := \sup_{x \in A} \inf_{y \in B} |x - y| \vee \sup_{x \in B} \inf_{y \in A} |x - y| \,. \tag{2.3}$$

Note that since $\Omega_n \subset \overline{\Omega}$, its Hausdorff distance simplifies to

$$d_H(\Omega_n, \overline{\Omega}) = \sup_{x \in \overline{\Omega}} \inf_{\mathbf{y} \in \Omega_n} |x - \mathbf{y}| \,. \tag{2.4}$$

To control both approximations at the same time we define the resolution of the graph as

$$\delta_n := d_H(\Omega_n, \overline{\Omega}) \vee d_H(\mathcal{O}_n, \mathcal{O}), \tag{2.5}$$

where we recall that $a \vee b = \max\{a, b\}$ and $a \wedge b = \min\{a, b\}$. For convenience we introduce a closest point projection $\pi_n : \overline{\Omega} \to \Omega_n$, meaning that

$$\pi_n(x) \in \operatorname*{arg\,min}_{\mathbf{x} \in \Omega_n} |x - \mathbf{x}| \,, \quad x \in \overline{\Omega}. \tag{2.6}$$

Note that the closest point projection has the important property that

$$|\pi_n(x) - x| \leq \delta_n, \quad \forall x \in \overline{\Omega}, \tag{2.7}$$

where δ_n is the graph resolution (2.5). For a function $\mathbf{u} : \Omega_n \to \mathbb{R}$ we define its piecewise constant extension to $\overline{\Omega}$ as

$$u_n : \overline{\Omega} \to \mathbb{R}, \quad u_n := \mathbf{u} \circ \pi_n. \tag{2.8}$$

Note that u_n is a simple function, constant on each Voronï cell [43, 44] of Ω_n.

For treating the labelling sets $\mathcal{O}_n$ and $\mathcal{O}$ we use a projection $\pi_{\mathcal{O}_n} : \mathcal{O} \to \mathcal{O}_n$, satisfying

$$\pi_{\mathcal{O}_n}(z) \in \operatorname*{arg\,min}_{\mathbf{z} \in \mathcal{O}_n} |z - \mathbf{z}| \,, \quad z \in \mathcal{O}. \tag{2.9}$$

By the definition of the graph resolution (2.5) it holds

$$|\pi_{\mathcal{O}_n}(z) - z| \leq \delta_n, \quad \forall z \in \mathcal{O}. \tag{2.10}$$

2.3 The Continuum Problem

Before we formulate the continuum problem, we need to discuss some properties of the continuum domain $\overline{\Omega}$ and the label set $\mathcal{O}$. Unlike in classical PDE theory we do not prescribe boundary values on the topological boundary $\partial\Omega$ but instead on the closed subset $\mathcal{O} \subset \overline{\Omega}$ which is equipped with a continuous labeling function $g : \mathcal{O} \to \mathbb{R}$.

Our motivation for this comes from applications like image classification where, e.g., $\Omega = (0,1)^d$ could model the space of grayscale images with d pixels and $\mathcal{O}$ is a subset of labeled images. In this case it is not meaningful to assume that $\mathcal{O} = \partial\Omega$ but instead $\mathcal{O}$ could even be a discrete subset of $\overline{\Omega}$. For notational convenience we will for the rest of the paper use the notation

$$\Omega_{\mathcal{O}} := \overline{\Omega} \setminus \mathcal{O}. \tag{2.11}$$

We define the geodesic distance $d_\Omega(x,y)$ by

$$d_\Omega(x,y) = \inf\left\{ \int_0^1 |\dot{\xi}(t)|\, dt \,:\, \xi \in C^1([0,1];\overline{\Omega}) \text{ with } \xi(0) = x \text{ and } \xi(1) = y \right\}, \quad x,y \in \overline{\Omega}. \tag{2.12}$$

By definition it holds $d_\Omega(x,y) \geq |x-y|$ and if the line segment from x to y is contained in $\overline{\Omega}$, which, in particular, is the case for convex $\overline{\Omega}$, then $d_\Omega(x,y) = |x-y|$. We pose the following assumption on the domain which relates the intrinsic geodesic distance with the extrinsic Euclidean one. The assumption is satisfied, e.g., for convex sets or sets with mildly smooth boundaries (see Section 5), and essentially demands that Ω has no inward-pointing cusps.

Assumption 2. There exists a function $\phi : [0,\infty) \to [0,\infty)$ with $\lim_{h\downarrow 0} \frac{\phi(h)}{h} = 0$ and $r_\Omega > 0$ such that

$$d_\Omega(x,y) \leq |x-y| + \phi(|x-y|), \quad \text{for all } x,y \in \Omega \,:\, |x-y| \leq r_\Omega. \tag{2.13}$$

Furthermore, for $h > 0$ we define $\sigma_\phi(h) = \sup_{0<s\leq h} \frac{\phi(s)}{s}$.

Using the geodesic distance function we can also define the distance between a point and a set and, more generally, between two sets as

$$\operatorname{dist}_\Omega(A,B) := \inf_{\substack{x\in A \\ y\in B}} d_\Omega(x,y) \tag{2.14}$$

and we abbreviate $\operatorname{dist}_\Omega(x,A) := \operatorname{dist}_\Omega(\{x\},A)$ for sets $A, B \subset \overline{\Omega}$. The corresponding distance functions with respect to the Euclidean metric are denoted by $\operatorname{dist}(\cdot,\cdot)$.

In what follows we forget that $\overline{\Omega}$ is a subset of $\mathbb{R}^d$ and simply regard $(\overline{\Omega}, d_\Omega)$ as a metric space, or more specifically, a *length space*. In particular, all topological notions like interior, closure, boundary, etc., will be understood with respect to the relative (intrinsic) topology of this length space. For example, $\Omega_{\mathcal{O}}$ is relatively open, its relative boundary $\partial^{\mathrm{rel}}\Omega_{\mathcal{O}}$ coincides with $\mathcal{O}$, and its relative closure is $\overline{\Omega}_{\mathcal{O}}^{\mathrm{rel}} = \overline{\Omega}$. Note that for a subset $A \subset \overline{\Omega}$ it holds $\overline{A}^{\mathrm{rel}} = \overline{A}$, which is why we do not distinguish those two different closures anymore. When it is clear from the context we also omit the word "relative". All closures and boundaries with respect to Ω as an open subset of $\mathbb{R}^d$ will be denoted by the usual symbols.

The continuum problem is to find an absolutely minimizing Lipschitz extension (AMLE) of the function $g : \mathcal{O} \to \mathbb{R}$ to the whole domain $\overline{\Omega}$. For this, we define the Lipschitz constant of a function $u : \overline{\Omega} \to \mathbb{R}$ on a subset $A \subset \overline{\Omega}$ as

$$\operatorname{Lip}_\Omega(u;A) := \sup_{x,y\in A} \frac{|u(x)-u(y)|}{d_\Omega(x,y)}, \tag{2.15}$$

and we abbreviate $\operatorname{Lip}_\Omega(u) := \operatorname{Lip}_\Omega(u;\overline{\Omega})$. A function $u \in C(\overline{\Omega})$ is an absolutely minimizing Lipschitz extension of g if it holds that

$$\begin{cases} \operatorname{Lip}_\Omega(u;\overline{V}) = \operatorname{Lip}_\Omega(u;\partial^{\mathrm{rel}}V) & \text{for all relatively open and connected subsets } V \subset \Omega_{\mathcal{O}}, \\ u = g & \text{on } \mathcal{O}. \end{cases} \tag{2.16}$$

2.4 Main Results

We now state our general uniform convergence result for solutions $\mathbf{u}_n : \Omega_n \to \mathbb{R}$ of the graph infinity Laplacian equation (2.1) to an AMLE $u : \overline{\Omega} \to \mathbb{R}$ satisfying (2.16). Obviously, we also have to quantify in which sense the labelling functions $\mathbf{g}_n : \mathcal{O}_n \to \mathbb{R}$ converge to $g : \mathcal{O} \to \mathbb{R}$. In the simplest case $\mathcal{O}_n = \mathcal{O}$ and $\mathbf{g}_n = g$ for all $n \in \mathbb{N}$. However, in general $\mathcal{O}_n \neq \mathcal{O}$ and we can work under the following much more general assumption.

Assumption 3. The labelling functions $\mathbf{g}_n : \mathcal{O}_n \to \mathbb{R}$ and $g : \mathcal{O} \to \mathbb{R}$ satisfy:

- $\sup_{n\in\mathbb{N}} \operatorname{Lip}_n(\mathbf{g}_n; \mathcal{O}_n) < \infty$ and $\operatorname{Lip}_\Omega(g; \mathcal{O}) < \infty$.
- There exists $C > 0$ such that for all $z \in \mathcal{O}$ it holds that $|\mathbf{g}_n(\pi_{\mathcal{O}_n}(z)) - g(z)| \leq C\delta_n$.

Example 2. In this example we illustrate some important special cases where this assumption is satisfied.

- If $\mathbf{g}_n = g$ and $\mathcal{O}_n = \mathcal{O}$ for all $n \in \mathbb{N}$ and $g : \mathcal{O} \to \mathbb{R}$ is Lipschitz with respect to the Euclidean distance, meaning $\operatorname{Lip}(g; \mathcal{O}) := \sup_{x,y\in\mathcal{O}} \frac{|g(x)-g(y)|}{|x-y|} < \infty$, then Assumption 3 is fulfilled. This is the setting of [5].
- If $\mathcal{O}_n = \mathcal{O}$ for all $n \in \mathbb{N}$ then $\pi_{\mathcal{O}_n}(z) = z$ for all $z \in \mathbb{N}$ and hence Assumption 3 reduces to sufficiently fast uniform convergence of $\mathbf{g}_n$ to g on $\mathcal{O}$ if $g : \mathcal{O} \to \mathbb{R}$ is Lipschitz.
- If $\mathbf{g}_n = g$ where $g : \overline{\Omega} \to \mathbb{R}$ is defined on the whole of $\overline{\Omega}$ and is Lipschitz, then Assumption 3 is satisfied by properties of the graph resolution δ_n, see (2.10). This is the setting of [37].

Our main result is non-asymptotic and depends on a free parameter $\varepsilon > 0$. Since this parameter arises from a nonlocal homogenized problem (see Section 4.3 for details), we refer to it as the nonlocal length scale. The relative scaling between the graph resolution δ_n, the graph length scale h, and the nonlocal length scale ε is of utmost importance for our convergence statement. Later we shall optimize over ε to obtain convergence rates depending only on δ_n and h.

Assumption 4. The parameters $\delta_n > 0$, $h > 0$, and $\varepsilon > 0$ satisfy

$$h \leq r_\Omega, \tag{2.17a}$$

$$\frac{h}{\varepsilon} < \frac{1}{2}, \tag{2.17b}$$

$$\sigma_\phi(h) + \frac{\delta_n}{h} \leq \frac{t_0}{4 + 2\sigma_\phi(h)} \left(1 - 2\frac{h}{\varepsilon}\right). \tag{2.17c}$$

Here $t_0 > 0$, $r_\Omega > 0$, and $\sigma_\phi(h)$ are defined in Assumptions 1 and 2, respectively.

Remark 2.1. By Assumption 2 $\sigma_\phi(h) \to 0$ as $h \to 0$. Hence, for very small graph bandwidths the scaling assumption on general domains approaches the one on convex domains where $\phi = 0$.

For a concise presentation we will in the following use the notation $a \lesssim b$ which is very common in the PDE community and means $a \leq C\, b$ for a universal constant $C > 0$, possibly depending on the kernel η or the label function g. Furthermore, the notation $a_n \ll b_n$ means $a_n/b_n \to 0$ as $n \to \infty$.

Theorem 2.2 (General Convergence Result)**.** *Let $\varepsilon > 0$, let Assumptions 1 to 4 hold, let $\mathbf{u}_n : \Omega_n \to \mathbb{R}$ solve (2.1), and $u : \overline{\Omega} \to \mathbb{R}$ solve (2.16).*

1. *It holds*

$$\sup_{\Omega_n} |u - \mathbf{u}_n| = \sup_{\overline{\Omega}} |u - u_n| \lesssim \varepsilon + \sqrt[3]{\frac{\delta_n}{h\varepsilon} + \frac{h}{\varepsilon^2} + \frac{\phi(h)}{h\varepsilon}}.$$

2. *If* $\inf_{\overline{\Omega}} |\nabla u| > 0$, *then it even holds*

$$\sup_{\Omega_n} |u - \mathbf{u}_n| = \sup_{\overline{\Omega}} |u - u_n| \lesssim \varepsilon + \frac{\delta_n}{h\varepsilon} + \frac{h}{\varepsilon^2} + \frac{\phi(h)}{h\varepsilon}.$$

Here, $u_n : \overline{\Omega} \to \mathbb{R}$ *denotes the piecewise constant extension of* $\mathbf{u}_n$, *defined in (2.8).*

The main advantage of our results is that they allow one to obtain uniform convergence rates for the weakest possible scaling assumption on the graph length scale h with respect to the graph resolution δ_n. Indeed, $h = h_n$ satisfying $\frac{\delta_n}{h_n} \to 0$ and $h_n \to 0$ is the smallest possible graph length scale such that the graph is asymptotically connected. For such scaling we recover known convergence results from [25, 37] and improve the result from [5].

Corollary 2.3 (Convergence under Weakest Scaling Assumption)**.** *For* $h = h_n$ *and* $\delta_n \ll h_n \ll \varepsilon$ *it holds that* $u_n \to u$ *uniformly on* $\overline{\Omega}$ *as* $n \to \infty$.

Proof. Using the scaling assumption $\delta_n \ll h_n \ll \varepsilon$ and keeping $\varepsilon > 0$ fixed, the entire root in Theorem 2.2 converges to zero and we obtain that

$$\lim_{n\to\infty} \sup_{\overline{\Omega}} |u - u_n| \lesssim \varepsilon.$$

Since $\varepsilon > 0$ was arbitrary, this shows the assertion. □

For every graph scaling h_n which is asymptotically larger than the weakest one, our general convergence theorem allows us to define a nonlocal scaling $\varepsilon = \varepsilon_n$ for which a uniform convergence rate holds true. By optimizing over ε_n we obtain convergence rates in different regimes, which we discuss in the following.

Corollary 2.4 (Small Length Scale Regime)**.** *We have the following convergence rates:*

1. *For* $\delta_n \lesssim h \lesssim \delta_n^{\frac{5}{9}}$ *it holds*

$$\sup_{\Omega_n} |u - \mathbf{u}_n| = \sup_{\overline{\Omega}} |u - u_n| \lesssim \left(\frac{\delta_n + \phi(h)}{h}\right)^{\frac{1}{4}}.$$

2. *If* $\inf_{\overline{\Omega}} |\nabla u| > 0$, *then for* $\delta_n \lesssim h \lesssim \delta_n^{\frac{3}{5}}$ *it holds*

$$\sup_{\Omega_n} |u - \mathbf{u}_n| = \sup_{\overline{\Omega}} |u - u_n| \lesssim \left(\frac{\delta_n + \phi(h)}{h}\right)^{\frac{1}{2}}.$$

Proof. For 1, we choose

$$\varepsilon = \left(\frac{\delta_n + \phi(h)}{h}\right)^{\frac{1}{4}}.$$

The condition $h \lesssim \delta_n^{\frac{5}{9}}$ implies that

$$\frac{h}{\varepsilon^2} \lesssim \frac{\delta_n + \phi(h)}{h\varepsilon},$$

and so 1 follows from Theorem 2.2.

The proof for 2 is similar, but instead we choose

$$\varepsilon = \left(\frac{\delta_n + \phi(h)}{h}\right)^{\frac{1}{2}}.$$ □

For larger length scales we make a case distinction based on the regularity of the boundary, in other words, the decay of the function ϕ in Assumption 2.

Corollary 2.5 (Large Length Scale Regime for Smooth Boundaries)**.** *Let $\phi(h) \lesssim h^{\frac{9}{5}}$ (e.g., if Ω is convex or $\partial\Omega$ is at least $C^{1,\frac{4}{5}}$). Then we have the following convergence rates:*

1. *For $h \gtrsim \delta_n^{\frac{5}{9}}$ it holds*

$$\sup_{\Omega_n} |u - \mathbf{u}_n| = \sup_{\overline{\Omega}} |u - u_n| \lesssim h^{\frac{1}{5}}.$$

2. *If $\inf_{\overline{\Omega}} |\nabla u| > 0$, then for $h \gtrsim \delta_n^{\frac{3}{5}}$ it holds*

$$\sup_{\Omega_n} |u - \mathbf{u}_n| = \sup_{\overline{\Omega}} |u - u_n| \lesssim h^{\frac{1}{3}}.$$

Proof. Since $\phi(h) \lesssim h^{\frac{9}{5}}$, we get

$$\frac{\phi(h)}{h\varepsilon} \lesssim \frac{h^{\frac{4}{5}}}{\varepsilon} = \frac{\varepsilon}{h^{\frac{1}{5}}} \frac{h}{\varepsilon^2}.$$

For both choices $\varepsilon := h^{\frac{1}{5}}$ (respectively, $\varepsilon := h^{\frac{1}{3}}$) it holds $\frac{\phi(h)}{h\varepsilon} \lesssim \frac{h}{\varepsilon^2}$ and we can can absorb the term $\frac{\phi(h)}{h\varepsilon}$ into $\frac{h}{\varepsilon^2}$. Furthermore, for these two choices it holds $\frac{h}{\varepsilon^2} \gtrsim \frac{\delta_n}{h\varepsilon}$ and moreover $\sqrt[3]{\frac{h}{\varepsilon^2}} = \varepsilon$ (respectively, $\frac{h}{\varepsilon^2} = \varepsilon$). □

In the case of less regular boundaries, where $h^{\frac{9}{5}} \ll \phi(h) \ll h$ as $h \to 0$, the third error term dominates. Note that this regime includes non-smooth domains, see Proposition 5.3 below.

Corollary 2.6 (Large Length Scale Regime for Less Smooth Boundaries)**.** *We have the following convergence rates:*

1. *For $\phi(h)^{\frac{5}{9}} \gtrsim h \gtrsim \delta_n^{\frac{5}{9}}$ it holds*

$$\sup_{\Omega_n} |u - \mathbf{u}_n| = \sup_{\overline{\Omega}} |u - u_n| \lesssim \left(\frac{\phi(h)}{h}\right)^{\frac{1}{4}}.$$

2. *If $\inf_{\overline{\Omega}} |\nabla u| > 0$, then for $\phi(h)^{\frac{3}{5}} \gtrsim h \gtrsim \delta_n^{\frac{3}{5}}$ it holds*

$$\sup_{\Omega_n} |u - \mathbf{u}_n| = \sup_{\overline{\Omega}} |u - u_n| \lesssim \left(\frac{\phi(h)}{h}\right)^{\frac{1}{2}}.$$

Proof. Assuming for a moment that $\phi(h) \gtrsim \frac{h^2}{\varepsilon}$ and $\phi(h) \gtrsim \delta_n$ the term $\frac{\phi(h)}{h\varepsilon}$ dominates $\frac{\delta_n}{h\varepsilon}$ and $\frac{h}{\varepsilon^2}$. Furthermore, for $\varepsilon := (\phi(h)/h)^{\frac{1}{4}}$ (respectively, $\varepsilon := (\phi(h)/h)^{\frac{1}{2}}$) it holds $\sqrt[3]{\frac{\phi(h)}{h\varepsilon}} = \varepsilon$ (respectively, $\frac{\phi(h)}{h\varepsilon} = \varepsilon$). Note that for these choices of ε the condition $\phi(h) \gtrsim \frac{h^2}{\varepsilon}$ is equivalent to $\phi(h)^{\frac{5}{9}} \gtrsim h$ or $\phi(h)^{\frac{3}{5}} \gtrsim h$, respectively. □

Remark 2.7 (Relation to previous results). In [39, Theorem 3.1.3] convergence rates are established for the special case where $\Omega_n = [0,1]^2_h$ is a grid, $\mathcal{O}_n = \Omega_n \cap \partial[0,1]^2$ consists of boundary vertices, and the weight function is chosen as $\eta = 1_{[0,1]}$. The condition that Ω is a grid in two dimensions is not directly used in [39, Theorem 3.1.3] and can be relaxed without any difficulty. Using the arguments in our paper, the condition that $\mathcal{O}_n = \Omega_n \cap \partial[0,1]^2$ can be relaxed as well, provided the boundary $\partial\Omega$ is $C^{2,\alpha}$, or Ω is convex. However, the assumption $\eta = 1_{[0,1]}$ seems essential to this result, since pointwise consistency for the discrete graph

∞-Laplacian is established using comparison with cones. Translating these results to our setting we would obtain

$$\sup_{\Omega_n} |u - \mathbf{u}_n| = \sup_{\overline{\Omega}} |u - u_n| \lesssim h + \left(\frac{\delta_n + \phi(h)}{h^2} \right)^{\frac{1}{3}}.$$

In the case that $\inf_{\overline{\Omega}} |\nabla u| > 0$, the result can be improved to read

$$\sup_{\Omega_n} |u - \mathbf{u}_n| = \sup_{\overline{\Omega}} |u - u_n| \lesssim h + \frac{\delta_n + \phi(h)}{h^2}.$$

Note that these rates only apply to very large stencil sizes and become meaningless as h approaches $\delta_n^{\frac{1}{2}}$. In contrast, our rates work for arbitrary small graph length scales above the connectivity threshold. In addition, for the rates to be non-vacuous, it is necesary that $\phi(h) \ll h^2$, which requires the boundary to be uniformly C^2 (or, say, $C^{2,\alpha}$ for $\alpha > 0$) when Ω is non-convex. In constrast, our results only require $\phi(h) \ll h$, which holds for non-convex domains with uniformly C^1 boundary. Finally, it does not seem clear how to relax the restrictive condition that $\eta = 1_{[0,1]}$ in [39, Theorem 3.1.3], which is important for applications to semi-supervised learning, where a very common choice is the Gaussian weight $\eta(t) = \exp(-t^2/(2\sigma^2))1_{[0,1]}$.

2.5 Extensions of our Main Results

Lastly, we mention two extensions of these results that would be natural to pursue.

1. In [5] a reweighted Lipschitz learning problem was studied of the form (1.2) with weights

$$w_n(\mathbf{x}, \mathbf{y}) = d_n(\mathbf{x})^\alpha d_n(\mathbf{y})^\alpha \eta_h(|\mathbf{x} - \mathbf{y}|), \tag{2.18}$$

   where $d_n(\mathbf{x})$ is the degree of node $\mathbf{x}$, given by

$$d_n(\mathbf{x}) = \sum_{\mathbf{y} \in \Omega_n} \eta_h(|\mathbf{x} - \mathbf{y}|).$$

   This reweighting modifes the continuum infinity Laplace equation (see [5, Theorem 2.4]) by adding a drift term along the gradient of the data density; in particular, the continuum equation becomes

$$\Delta_\infty u + 2\alpha \nabla \log \rho \cdot \nabla u = 0$$

   in the case that the point cloud Ω_n has density $\rho(x)$ locally as $n \to \infty$. The motivation for the reweighting was to introduce the density ρ into the continuum model in order to improve the semi-supervised classification results. Indeed, the whole point of semi-supervised learning is to use properties of the unlabeled data, e.g., its distribution, to improve classification results. Numerical experiments in [5] established improved performance of the reweighted Lipschitz learning classifier for both synthetic and real data.

   It would be natural to extend the results in this paper to the reweighted model from [5] with weights given in (2.18). We expect the main modifications to be to the notion of geodesic distance on the domain Ω, which will now be weighted by $\rho^{-2\alpha}$. The geodesic distance functions, which are the continuum limit of the graph versions, would be solutions of the eikonal equation $\rho^{2\alpha}|\nabla u| = 1$, and the continuum model would be modified accordingly.

2. In machine learning, it is common to take the *manifold assumption* [21], whereby the data $\Omega_n \subset \mathbb{R}^d$ is sampled from an underlying data manifold $\mathcal{M} \subset \mathbb{R}^d$ of much smaller intrinsic dimension $\dim(\mathcal{M}) = D \ll d$. It would also be natural to extend our results to this setting. Since an embedded manifold can be equipped with the geodesic metric inherited from the metric on the ambient space, and furthermore comparison with distance functions is a purely metric notion, we expect that our arguments translate to this setting in a straightforward way.

3 Discrete: Graph Infinity Laplacian Equation

In this section we investigate the discrete problem of the graph infinity Laplacian equation (2.1). In fact, we will work in an abstract weighted graph setup which is more general than the geometric graphs which we use for our final discrete to continuum results. The main result in this section is that graph infinity harmonic functions satisfy a comparison principle with graph distance functions.

3.1 Weighted Graphs

Let $G = (X, W)$ be a weighted graph with nodes X and edge weights $W = (w_{xy})_{x,y \in X}$. The edge weights are assumed to be nonnegative ($w_{xy} \geq 0$) and symmetric ($w_{xy} = w_{yx}$). For $\mathcal{O} \subset X$ and $g : \mathcal{O} \to \mathbb{R}$, we consider the graph ∞-Laplace equation

$$\begin{cases} \mathcal{L}_\infty^G u = 0, & \text{in } X \setminus \mathcal{O}, \\ u = g, & \text{on } \mathcal{O}, \end{cases} \tag{3.1}$$

where the graph ∞-Laplacian is defined by

$$\mathcal{L}_\infty^G u(x) = \min_{y \in X} w_{xy}(u(y) - u(x)) + \max_{y \in X} w_{xy}(u(y) - u(x)). \tag{3.2}$$

The well-posedness of (3.1) was established in [5] and is discussed in more detail in Section 3.2 below. In graph-based semi-supervised learning, (3.1) is referred to as Lipschitz learning [5, 24, 37], since it propagates the labels $u = g$ on $\mathcal{O}$ to the entire graph in a Lipschitz continuous manner. The discrete ∞-Laplace equation has also been used for image inpainting, which is concerned with filling in missing parts of an image (e.g., in art restoration), see [15].

In this section we study the discrete problem (3.1) and show that solutions satisfy a comparison principle with graph distance functions which is the discrete analogue of Definition 4.1.

We define the graph distance function $d_G : X \times X \to \mathbb{R} \cup \{\infty\}$ by

$$d_G(x, y) = \min \left\{ \sum_{i=1}^{N} w_{x_{i-1}x_i}^{-1} \, : \, N \in \mathbb{N}, \, \{x_i\}_{i=0}^N \subset X, \, x_0 = x, \, x_N = y \right\} \tag{3.3}$$

for $x \neq y$, and $d_G(x, x) = 0$. Here we use the convention that if $w_{x_{i-1}x_i} = 0$ for any i then $w_{x_{i-1}x_i}^{-1} = \infty$. We note that since the graph G is symmetric, we have $d_G(x, y) = d_G(y, x)$.

The notion of graph connectivity is now simple to state in terms of the graph distance function.

Definition 3.1. We say that G is *connected* if $d_G(x, y) < \infty$ for all $x, y \in X$. We say that G is *connected to* $\mathcal{O}$ if for every $x \in X$ there exists $y \in \mathcal{O}$ with $d_G(x, y) < \infty$.

We remark that a connected graph is also connected to any constraint set $\mathcal{O} \subset X$. However, a disconnected graph can still be connected to $\mathcal{O}$, provided $\mathcal{O}$ has a non-empty intersection with every connected component of G.

In order to study comparison with graph distance functions, we need a notion of graph boundary. For $X' \subset X$ we define the boundary of X', denoted $\partial X'$, as the set

$$\partial X' = \{x \in X \setminus X' \, : \, w_{xy} > 0 \text{ for some } y \in X'\}. \tag{3.4}$$

In other words, the boundary $\partial X'$ consists of all nodes in $X \setminus X'$ that are connected to some node in X'. This is often called the *exterior* boundary of X'. The *interior* boundary can be defined similarly as all nodes in X' that are connected to a node in $X \setminus X'$. While we only use the notion of exterior boundary defined in (3.4) in this paper, the results in this section hold, with minor modifications, for the interior boundary. We also define the *closure* of X', denoted $\overline{X'}$, to be the set

$$\overline{X'} = X' \cup \partial X'. \tag{3.5}$$

3.2 Comparison with Graph Distance Functions

Our main result in this section shows that graph infinity harmonic functions satisfy comparison with graph distance functions.

Theorem 3.2. *Assume G is connected. Let $X' \subsetneq X$ and suppose that $u : \overline{X'} \to \mathbb{R}$ satisfies $-\mathcal{L}_\infty^G u(x) \leq 0$ for all $x \in X'$. Then for every $a \geq 0$ and $z \in X \setminus X'$ we have*

$$\max_{\overline{X'}}(u - a\, d_G(\cdot, z)) = \max_{\partial X'}(u - a\, d_G(\cdot, z)). \tag{3.6}$$

If $u : \overline{X'} \to \mathbb{R}$ satisfies $-\mathcal{L}_\infty^G u(x) \geq 0$ for all $x \in X'$, then for every $a \geq 0$ and $z \in X \setminus X'$ we have

$$\min_{\overline{X'}}(u + a\, d_G(\cdot, z)) = \min_{\partial X'}(u + a\, d_G(\cdot, z)) \tag{3.7}$$

Remark 3.3. One could also investigate the converse statement, namely that comparison with graph distance functions implies being a solution to the graph infinity Laplacian equation (3.1). However, since this is not needed for our result we do not consider this question here. Note also that we do not have to assume that X' is a connected subset of X, as it is typically done for defining AMLEs (see (2.16)).

The proof of Theorem 3.2 relies on a discrete comparison principle for (3.1), which was established in [5]. We include the statement of the result below for reference.

Theorem 3.4 ([5, Theorem 3.1])**.** *Assume that G is connected to $\mathcal{O}$, and let $u, v : X \to \mathbb{R}$ satisfy*

$$-\mathcal{L}_\infty^G u(x) \leq 0 \leq -\mathcal{L}_\infty^G v(x) \quad \text{for all } x \in X \setminus \mathcal{O}.$$

Then

$$\max_X(u - v) = \max_{\mathcal{O}}(u - v).$$

We note that it follows from Theorem 3.4 and the Perron method that (3.1) has a unique solution u whenever G is connected to $\mathcal{O}$. We refer to [5, Theorem 3.4] and [4, Theorem 4] for the proof of this result.

We now establish that graph distance functions satisfy the graph eikonal equation.

Lemma 3.5. *Assume G is connected and let $z \in X$. Then the graph distance function $u := d_G(\cdot, z)$ is the unique solution of the graph eikonal equation*

$$\max_{y \in X} w_{xy}(u(x) - u(y)) = 1 \quad \text{for all } x \in X \setminus \{z\} \tag{3.8}$$

satisfying $u(z) = 0$.

Proof. It follows directly from the definition of the graph distance d_G, (3.3), that $u = d_G(\cdot, z)$ satisfies the dynamic programming principle

$$u(x) = \min_{y \in X} \left\{ u(y) + w_{xy}^{-1} \right\}, \quad x \in X \setminus \{z\}. \tag{3.9}$$

As above, we take $w_{xy}^{-1} = \infty$ when $w_{xy} = 0$. Since G is connected, for every $x \in X$ there exists $y \in X$ with $w_{xy} > 0$, so the minimum in (3.9) can be restricted to $y \in X$ with $w_{xy} > 0$ (i.e., graph neighbors of x). We note that (3.9) can be rearranged to show that

$$\max_{y \in X}\{u(x) - u(y) - w_{xy}^{-1}\} = 0, \quad x \in X \setminus \{z\}.$$

It follows that

$$\max_{y \in X}\{w_{xy}(u(x) - u(y)) - 1\} = 0, \quad x \in X \setminus \{z\}, \tag{3.10}$$

which is equivalent to (3.8).

To show uniqueness, let $u : X \to \mathbb{R}$ be a solution of (3.8) satisfying $u(z) = 0$. Let $\lambda > 1$ and let $x_0 \in X$ be a point at which $u - \lambda\, d_G(\cdot, z)$ attains its maximum over X. Then we have

$$u(x_0) - u(x) \geq \lambda(d_G(x_0, z) - d_G(x, z)), \quad x \in X.$$

If $x_0 \neq z$, then we have

$$1 = \max_{y \in X} w_{x_0 y}(u(x_0) - u(y)) \geq \lambda \max_{y \in X} w_{x_0 y}(d_G(x_0, z) - d_G(y, z)) = \lambda > 1,$$

which is a contradiction. Therefore $x_0 = z$ and so

$$\max_X (u - \lambda\, d_G(\cdot, z)) = u(z) - \lambda\, d_G(z, z) = 0.$$

Since $\lambda > 1$ is arbitrary, it follows that $u \leq d_G(\cdot, z)$. A similar argument, examining the minimum of $u - \lambda\, d_G(\cdot, z)$ for $0 < \lambda < 1$ shows that $u \geq d_G(\cdot, z)$, which completes the proof. □

In order to apply Theorem 3.4 to prove Theorem 3.2, we need to show that the subgraph of G with vertices $\overline{X}'$ is connected to its boundary $\partial X'$ in the sense of Definition 3.1.

Proposition 3.6. *Assume G is connected. Let $X' \subsetneq X$ and let $G' = (\overline{X'}, W')$ be the subgraph of G with weights $W' = (w_{xy})_{x,y \in \overline{X'}}$. Then G' is connected to $\partial X'$.*

Proof. If $x \in \overline{X}' \setminus X'$ then $x \in \partial X'$ and x is trivially connected to $\partial X'$. Let therefore $x \in X'$. Since G is connected and $X' \subsetneq X$, there exists $y \in X \setminus X'$ and a path $x_0 = x, x_1, x_2, \ldots, x_n = y$ with $w_{x_i x_{i+1}} > 0$ for $i = 0, \ldots, n-1$. Since $x_0 \in X'$ and $x_n \in X \setminus X'$, there exists $0 \leq j \leq n-1$ such that $x_i \in X'$ for $0 \leq i \leq j$ and $x_{j+1} \in X \setminus X'$. It follows that $x_{j+1} \in \partial X'$ and

$$d_{G'}(x, x_{j+1}) \leq \sum_{i=0}^{j} w_{x_i x_{i+1}}^{-1} < \infty.$$

This completes the proof. □

We now give the proof of Theorem 3.2.

Proof of Theorem 3.2. For notational simplicity, let us set $v = d_G(\cdot, z)$. We first show that

$$\mathcal{L}_\infty^G v(x) \leq 0 \quad \text{for all } x \in X \setminus \{z\}. \tag{3.11}$$

Let $x \in X \setminus \{z\}$. By Lemma 3.5 we have

$$\min_{y \in X} w_{xy}(v(y) - v(x)) = -\max_{y \in X} w_{xy}(v(x) - v(y)) = -1.$$

Now let $y_* \in X$ be such that

$$w_{xy_*}(v(y_*) - v(x)) = \max_{y \in X} w_{xy}(v(y) - v(x)).$$

Invoking Lemma 3.5 again yields

$$\max_{y \in X} w_{xy}(v(y) - v(x)) = w_{xy_*}(v(y_*) - v(x)) \leq \max_{z \in X} w_{zy_*}(v(y_*) - v(z)) = 1,$$

since the weights are symmetric, so that $w_{zy_*} = w_{y_* z}$. This establishes the claim (3.11).

We now consider the subgraph $G' = (\overline{X'}, W')$, as in Proposition 3.6. Since $\mathcal{L}_\infty^{G'} v(x) = \mathcal{L}_\infty^G v(x)$ for all $x \in X'$ we have

$$\mathcal{L}_\infty^{G'}(av)(x) = a\mathcal{L}_\infty^{G'} v(x) = a\mathcal{L}_\infty^G v(x) \leq 0$$

for all $x \in X'$ and $a \geq 0$. By Proposition 3.6, G' is connected to $\partial X'$, and so we can invoke the comparison principle, Theorem 3.4, to obtain that for u satisfying $-\mathcal{L}_\infty^G(u) \leq 0$ it holds

$$\max_{\overline{X'}}(u - a\,v) = \max_{\partial X'}(u - a\,v).$$

If, conversely, it holds $-\mathcal{L}_\infty^G u \geq 0$ we get $-\mathcal{L}_\infty^G(-u) \leq 0$ and hence

$$\max_{\overline{X'}}(-u - a\,v) = \max_{\partial X'}(-u - a\,v)$$

which is equivalent to

$$\min_{\overline{X'}}(u + a\,v) = \min_{\partial X'}(u + a\,v).$$

This concludes the proof. □

3.3 Relations to Absolute Minimizers

Since according to Theorem 3.2, the solution of (3.1) satisfies comparison with graph distance functions, it seems natural that it is also an absolutely minimal Lipschitz extension of g from the constraint set $\mathcal{O}$ to the entire graph X. Since, for the sake of simplifying later proofs, we are not working with connected test sets X' in Theorem 3.2, we cannot use standard results like [23, Proposition 4.1] and will prove the statement ourselves.

For a subset $X' \subset X$, and a function $u : X' \to \mathbb{R}$, we define the graph Lipschitz constant

$$\mathrm{Lip}_G(u; X') = \max_{x,y \in X'} \frac{|u(x) - u(y)|}{d_G(x, y)} \tag{3.12}$$

and we abbreviate $\mathrm{Lip}_G(u) := \mathrm{Lip}_G(u; X)$. Before we can prove the absolute minimality we need the following Lemma.

Lemma 3.7. *Let $X' \subset X$, $x \in \overline{X'}$, and $X'' := X' \setminus \{x\}$. Then it holds*

$$\partial X'' \subset \partial X' \cup \{x\}, \qquad \overline{X'} \setminus \overline{X''} \subset \partial X' \cup \{x\}.$$

Proof. If $X'' = \emptyset$ the statements are trivial so we assume that $X'' \neq \emptyset$.

For the first statement we distinguish two cases: If $z \in \partial X'' \cap X'$, then by definition of the graph boundary (3.4) it holds $z \notin X''$ which together with $z \in X'$ implies $z = x$. If $z \in \partial X'' \setminus X'$, then again by (3.4) and the fact that $X'' \neq \emptyset$ it follows that there exists $y \in X'' \subset X'$ with $w_{yz} > 0$ and hence $z \in \partial X'$. Combining both cases shows the desired statement.

For the second statement we let $z \in \overline{X'} \setminus \overline{X''}$ and argue as follows: If $z = x$ nothing needs to be shown and therefore we assume $z \neq x$. We know that in particular it holds $z \in \overline{X'} = X' \cup \partial X'$ so if $z \in \partial X'$ the proof is complete. The only interesting case is $z \in X'$ in which case we get that

$$z \in X' \setminus \overline{X''} \subset X' \setminus X'' = X' \setminus (X' \setminus \{x\}) = \{x\}.$$

This contradicts our assumption and we can conclude. □

Proposition 3.8. *Assume G is connected, let $g : \mathcal{O} \to \mathbb{R}$ be Lipschitz continuous, and let $u : X \to \mathbb{R}$ be a solution of (3.1). Then for all subsets $X' \subset X \setminus \mathcal{O}$ it holds*

$$\mathrm{Lip}_G(u; \overline{X'}) = \mathrm{Lip}_G(u; \partial X'). \tag{3.13}$$

Proof. By definition of $\mathrm{Lip}_G(u;\partial X')$ we have that

$$u(x) - u(y) \le \mathrm{Lip}_G(u;\partial X')\, d_G(x,y), \quad \forall x, y \in \partial X'. \tag{3.14}$$

Using Theorem 3.2 we get

$$\max_{\overline{X'}} \left(u - \mathrm{Lip}_G(u;\partial X')\, d_G(\cdot, y)\right) = \max_{\partial X'} \left(u - \mathrm{Lip}_G(u;\partial X')\, d_G(\cdot, y)\right) \le u(y), \quad \forall y \in \partial X',$$

and therefore

$$u(x) - u(y) \le \mathrm{Lip}_G(u;\partial X')\, d_G(x,y), \quad \forall x \in \overline{X'},\ \forall y \in \partial X'. \tag{3.15}$$

Note that (3.15) implies that

$$u(x) = \min_{y \in \partial X' \cup \{x\}} \{u(y) + \mathrm{Lip}_G(u;\partial X')\, d_G(x,y)\}, \quad \forall x \in \overline{X'}.$$

Fixing $x \in \overline{X'}$, setting $X'' = X' \setminus \{x\}$ and using comparison with cones from below from Theorem 3.2 we have

$$\begin{aligned}\min_{\overline{X''}} \left(u + \mathrm{Lip}_G(u;\partial X')\, d_G(x,\cdot)\right) &= \min_{\partial X''} \left(u + \mathrm{Lip}_G(u;\partial X')\, d_G(x,\cdot)\right) \\ &\ge \min_{\partial X' \cup \{x\}} \left(u + \mathrm{Lip}_G(u;\partial X')\, d_G(x,\cdot)\right) = u(x),\end{aligned}$$

since $\partial X'' \subset \partial X' \cup \{x\}$ according to Lemma 3.7. It follows that

$$u(x) - u(y) \le \mathrm{Lip}_G(u;\partial X')\, d_G(x,y), \quad \forall y \in \overline{X''}. \tag{3.16}$$

By Lemma 3.7 we know that $\overline{X'} \setminus \overline{X''} \subset \partial X' \cup \{x\}$. Using (3.15) we then have that (3.16) also holds for every $y \in \overline{X'} \setminus \overline{X''}$ and therefore

$$u(x) - u(y) \le \mathrm{Lip}_G(u;\partial X')\, d_G(x,y) \quad \forall y \in \overline{X'}.$$

Since $x \in \overline{X'}$ was arbitrary, this concludes the proof. □

4 Continuum: Absolutely Minimizing Lipschitz Extensions

4.1 Equivalence with Comparison with Distance Functions

In Section 2 we have already introduced the continuum problem (2.16) of finding an Absolutely Minimizing Lipschitz Extension (AMLE) of the label function $g : \mathcal{O} \to \mathbb{R}$. In fact, we will not work with AMLEs directly but rely on an intriguing property of theirs, called Comparison with Distance Functions (CDF), see [12, 23].

Definition 4.1 (CDF)**.** We shall say that a upper semicontinuous function $u \in USC(\overline{\Omega})$ satisfies CDF from above in $\Omega_{\mathcal{O}}$, if for each relatively open and connected subset $V \subset \Omega_{\mathcal{O}}$, any $x_0 \in \overline{\Omega} \setminus V$ and $a \ge 0$ we have

$$\max_{\overline{V}}(u - a\, d_\Omega(x_0,\cdot)) = \max_{\partial^{\mathrm{rel}} V}(u - a\, d_\Omega(x_0,\cdot)).$$

Similarly, we say $u \in LSC(\overline{\Omega})$ satisfies CDF from below in $\Omega_{\mathcal{O}}$, if for each relatively open and connected subset $V \subset \Omega_{\mathcal{O}}$, any $x_0 \in \overline{\Omega} \setminus V$ and $a \ge 0$ we have

$$\min_{\overline{V}}(u + a\, d_\Omega(x_0,\cdot)) = \min_{\partial^{\mathrm{rel}} V}(u + a\, d_\Omega(x_0,\cdot)).$$

We say $u \in C(\overline{\Omega})$ satisfies CDF if it satisfies CDF from above and below.

Remark 4.2. Note that this definition is a special case of [23, Definition 2.3]. For this, one regards $X := \overline{\Omega}$ equipped with d_Ω as a length space and $\Omega_\mathcal{O}$ as an open subset of X with respect to the topology of d_Ω, as explained in Section 2.3. Consequently, open subsets of $\Omega_\mathcal{O}$ with respect to this topology are relatively open subsets.

In fact, being an AMLE is equivalent to satisfying CDF, which is why we will only work with this property for the rest of the paper.

Proposition 4.3 ([23, Proposition 4.1])**.** *Let $g : \mathcal{O} \to \mathbb{R}$ be Lipschitz continuous. A function $u : C(\overline{\Omega}) \to \mathbb{R}$ with $u = g$ on $\mathcal{O}$ is an AMLE of g if and only if it satisfies CDF on $\Omega_\mathcal{O}$.*

Hence, we can equivalently reformulate the continuum problem as to find a function $u : \overline{\Omega} \to \mathbb{R}$ which attains the label values on $\mathcal{O}$ and satisfies CDF in $\Omega_\mathcal{O}$:

$$\begin{cases} u \text{ satisfies CDF} & \text{on } \Omega_\mathcal{O}, \\ u = g & \text{on } \mathcal{O}. \end{cases} \tag{4.1}$$

For convenience we also introduce the notion of sub- and super solutions to (4.1).

Definition 4.4 (Sub- and supersolutions)**.** We call $u \in USC(\overline{\Omega})$ a subsolution of (4.1) if

$$\begin{cases} u \text{ satisfies CDF from above} & \text{on } \Omega_\mathcal{O}, \\ u \leq g & \text{on } \mathcal{O}. \end{cases}$$

Furthermore, $u \in LSC(\overline{\Omega})$ is called a supersolution if $-u$ is a subsolution. Obviously, being both a sub- and a supersolution is equivalent to being a solution of (4.1) and hence also to being an AMLE.

4.2 Relations to Infinity Laplacian Equations

Under some regularity conditions on the boundary of the domain Ω (now regarded as a subset of $\mathbb{R}^d$), and if $\mathcal{O}$ is a subset of $\partial\Omega$, one can relate being an AMLE, or equivalently satisfying CDF, with solving an infinity Laplacian equation, where the infinity Laplacian operator is defined as

$$\Delta_\infty u = \langle \nabla u, D^2 u \nabla u \rangle.$$

Proposition 4.5. *Let $\partial\Omega$ denote the boundary of Ω, regarded as subset of $\mathbb{R}^d$. Then the following is true:*

1. *If $\mathcal{O} = \partial\Omega$ then $u \in C(\overline{\Omega})$ is an AMLE of $g : \partial\Omega \to \mathbb{R}$ if and only if it is a viscosity solution of*

$$\begin{cases} -\Delta_\infty u &= 0 \quad \text{in } \Omega, \\ u &= g \quad \text{on } \partial\Omega. \end{cases} \tag{4.2}$$

2. *If Ω is smooth and convex and $\mathcal{O} \subset \partial\Omega$ then $u \in C(\overline{\Omega})$ is an AMLE of $g : \mathcal{O} \to \mathbb{R}$ if and only if it is a viscosity solution of*

$$\begin{cases} -\Delta_\infty u &= 0 \quad \text{in } \Omega \setminus \mathcal{O}, \\ \frac{\partial u}{\partial \nu} &= 0 \quad \text{on } \partial\Omega \setminus \mathcal{O}, \\ u &= g \quad \text{on } \mathcal{O}, \end{cases} \tag{4.3}$$

where the Neumann boundary conditions are to be understood in the weak viscosity sense, cf. [13, 16].

Proof. The first statement is well-known and contained in the seminal paper by Aronsson [3], see Theorem 4.1 therein. The second one is a special case of the results in [2]. In Lemma 3.1 therein the authors show that, under the hypothesis that Ω is smooth and convex, the solution of the PDE admits CDF and hence is an AMLE. Furthermore, by the uniqueness of the CDF problem (see Proposition 4.11 below), which does, however, not require any convexity, this is in fact an equivalence. □

4.3 Max-Ball and Perturbation Statements

The following "max-ball" and perturbation statements will be essential when we prove discrete-to-continuum convergence rates. In a simpler setting, using Dirichlet boundary conditions $\mathcal{O} = \partial\Omega$, they were all proved in [39]. However, for our more general problem (4.1) we need to reprove them.

They main point of the max-ball statement is that sub- and supersolutions of (4.1) can be turned into sub- and supersolutions of a nonlocal difference equation which is much easier to study. In particular, as shown in [1] the nonlocal equation allows us to derive a maximum principle for (4.1) which will let us prove uniqueness and discrete-to-continuum convergence.

The perturbation statement will allow us to turn sub- or supersolutions of the nonlocal equation into strict sub- or supersolutions which will turn out useful for deriving convergence rates.

Because of the nonlocal nature of the above statements, one has to work on an inner parallel set of $\overline{\Omega}$. However, it suffices to maintain positive distance to the constraint set $\mathcal{O}$ only. In what follows we denote by

$$B_\Omega(x,\varepsilon) := \{y \in \overline{\Omega} \,:\, d_\Omega(x,y) < \varepsilon\} \tag{4.4}$$

the open geodesic ball around x with radius $\varepsilon > 0$ and its closure is denoted by $\overline{B}_\Omega(x,\varepsilon)$. Using this, we can define the set

$$\Omega^\varepsilon_{\mathcal{O}} := \{x \in \overline{\Omega} \,:\, \operatorname{dist}_\Omega(x,\mathcal{O}) > \varepsilon\}. \tag{4.5}$$

Note that this "inner parallel set" deliberately includes those parts of the topological boundary $\partial\Omega$ which are sufficiently far from the constraint set $\mathcal{O}$.

For a function $u : \overline{\Omega} \to \mathbb{R}$ and $x \in \overline{\Omega}$ we also define

$$T^\varepsilon u(x) := \sup_{\overline{B}_\Omega(x,\varepsilon)} u, \qquad T_\varepsilon u(x) := \inf_{\overline{B}_\Omega(x,\varepsilon)} u, \tag{4.6}$$

$$S^+_\varepsilon u := \frac{1}{\varepsilon}(T^\varepsilon u - u), \qquad S^-_\varepsilon u := \frac{1}{\varepsilon}(u - T_\varepsilon u), \tag{4.7}$$

$$\Delta^\varepsilon_\infty u := \frac{1}{\varepsilon}\left(S^+_\varepsilon u - S^-_\varepsilon u\right). \tag{4.8}$$

We refer to the last object as nonlocal infinity Laplacian operator. Interestingly, harmonic functions with respect to this operator coincide with AMLEs with respect to a discrete "step distance" and we refer the interested reader to [30] for more details. The following lemma, the proof of which is adapted from [1], is the desired max-ball statement. It states that functions satisfying CDF can be turned into a sub- and supersolution of the nonlocal infinity Laplacian equation.

Lemma 4.6 (Max-Ball)**.** *Let $u \in USC(\overline{\Omega})$ and $v \in LSC(\overline{\Omega})$ satisfy CDF on $\Omega_{\mathcal{O}}$ from above and from below, respectively. Then for all $\varepsilon > 0$ it holds*

$$-\Delta^\varepsilon_\infty T^\varepsilon u(x_0) \le 0 \le -\Delta^\varepsilon_\infty T_\varepsilon v(x_0), \quad \forall x_0 \in \Omega^{2\varepsilon}_{\mathcal{O}}. \tag{4.9}$$

Proof. We just show the inequality for the subsolution u, the other case works analogously. Since u is upper semicontinous we can select points $y_0 \in \overline{B}_\Omega(x_0,\varepsilon)$ and $z_0 \in \overline{B}_\Omega(x_0,2\varepsilon)$ such that

$$u(y_0) = T^\varepsilon u(x_0), \quad u(z_0) = T^{2\varepsilon} u(x_0).$$

Using the definition of the nonlocal infinity Laplacian (4.8) one computes

$$-\varepsilon^2 \Delta^\varepsilon_\infty T^\varepsilon u(x_0) = 2T^\varepsilon u(x_0) - (T^\varepsilon T^\varepsilon u)(x_0) - (T_\varepsilon T^\varepsilon u)(x_0) \le 2u(y_0) - u(z_0) - u(x_0). \tag{4.10}$$

This is true since $x_0 \in \overline{B}_\Omega(y,\varepsilon)$ for all $y \in \overline{B}_\Omega(x_0,\varepsilon)$ and therefore,

$$T^\varepsilon u(y) = \sup_{B_\Omega(y,\varepsilon)} u \ge u(x_0) \implies (T_\varepsilon T^\varepsilon u)(x_0) = \inf_{y \in B_\Omega(x_0,\varepsilon)} T^\varepsilon(y) \ge u(x_0).$$

Now one observes that the following inequality is true

$$u(w) \le u(x_0) + \frac{u(z_0) - u(x_0)}{2\varepsilon} d_\Omega(w, x_0), \quad \forall w \in \partial^{\mathrm{rel}}(B_\Omega(x_0, 2\varepsilon) \setminus \{x_0\}).$$

For $w = x_0$ the inequality is obviously correct. If $d_\Omega(w, x_0) = 2\varepsilon$, then the definition of z_0 shows it.

Since u satisfies CDF from above on $\Omega_{\mathcal{O}}$ and using that $\operatorname{dist}_\Omega(x_0, \mathcal{O}) > 2\varepsilon$, we obtain

$$u(w) \le u(x_0) + \frac{u(z_0) - u(x_0)}{2\varepsilon} d_\Omega(w, x_0), \quad \forall w \in \overline{B}_\Omega(x_0, 2\varepsilon).$$

Choosing $w = y_0$ we get

$$u(y_0) \le u(x_0) + \frac{u(z_0) - u(x_0)}{2\varepsilon} d_\Omega(y_0, x_0) \le u(x_0) + \frac{u(z_0) - u(x_0)}{2}.$$

Using this inequality together with (4.10) yields the assertion. □

The next results are contained in [39] which treats a more general graph Laplacian for a graph $G = (X, E, Y)$ consisting of a vertex set X, an edge set E, and a "boundary set" Y. This graph Laplacian is defined as

$$\Delta_\infty^G u(x) = \sup_{\{x,y\} \in E} (u(y) - u(x)) - \sup_{\{x,y\} \in E} (u(x) - u(y)) \tag{4.11}$$

and for the choice

$$\begin{aligned} X &:= \Omega_{\mathcal{O}}^\varepsilon, \\ Y &:= \{x \in X \,:\, \operatorname{dist}_\Omega(x_0, \mathcal{O}) \le 2\varepsilon\}, \\ E &:= \{\{x, y\} \subset X \,:\, x \in X \setminus Y,\ 0 < d_\Omega(x, y) \le \varepsilon\}, \end{aligned}$$

it holds

$$\Delta_\infty^G u(x) = \varepsilon^2 \Delta_\infty^\varepsilon u(x), \tag{4.12}$$

where $\Delta_\infty^\varepsilon$ is defined in (4.8). Hence, we can use the results from [39] to obtain statements about the nonlocal Laplacian $\Delta_\infty^\varepsilon$.

The first result is a maximum principle for the nonlocal infinity Laplacian.

Lemma 4.7. *Assume that for a constant $C \ge 0$ the functions $u, v : \Omega_{\mathcal{O}}^\varepsilon \to \mathbb{R}$ satisfy*

$$-\Delta_\infty^\varepsilon u \le C \le -\Delta_\infty^\varepsilon v \tag{4.13}$$

in $\Omega_{\mathcal{O}}^{2\varepsilon}$. Then it holds

$$\sup_{\Omega_{\mathcal{O}}^\varepsilon} (u - v) = \sup_{\Omega_{\mathcal{O}}^\varepsilon \setminus \Omega_{\mathcal{O}}^{2\varepsilon}} (u - v). \tag{4.14}$$

Proof. See [39, Theorem 2.6.5]. □

The following two lemmas are perturbation results. The first one allows us to turn a supersolution of the nonlocal equation into one with strictly positive gradient. The second one shows how to turn a supersolution into a strict supersolution.

Lemma 4.8. *If $u : \Omega_{\mathcal{O}}^\varepsilon \to \mathbb{R}$ is bounded from below and satisfies $-\Delta_\infty^\varepsilon u \ge 0$ in $\Omega_{\mathcal{O}}^{2\varepsilon}$, then for any $\delta > 0$ there is a function $v : \Omega_{\mathcal{O}}^\varepsilon \to \mathbb{R}$ that satisfies*

$$-\Delta_\infty^\varepsilon v \ge 0, \quad S_\varepsilon^- v \ge \delta, \quad u \le v \le u + 2\delta \operatorname{dist}_\Omega(\cdot, \Omega_{\mathcal{O}}^\varepsilon \setminus \Omega_{\mathcal{O}}^{2\varepsilon}) \quad \text{on } \Omega_{\mathcal{O}}^{2\varepsilon}.$$

Proof. See [39, Lemma 2.6.3]. □

Lemma 4.9. *Suppose $v : \Omega_{\mathcal{O}}^{\varepsilon} \to \mathbb{R}$ is bounded and $-\Delta_{\infty}^{\varepsilon} v \geq 0$ on $\Omega_{\mathcal{O}}^{2\varepsilon}$. Then there exists $\delta_0 > 0$ such that for any $0 \leq \delta \leq \delta_0$ the function $w := v - \delta v^2$ satisfies*

$$-\Delta_{\infty}^{\varepsilon} w \geq -\Delta_{\infty}^{\varepsilon} v + \delta (S_{\varepsilon}^{-} v)^2 \quad \text{on } \Omega_{\mathcal{O}}^{2\varepsilon}.$$

Proof. If $v \geq 0$ the map $t \mapsto t + \delta t^2$ is monotone on the range of v for all $\delta \geq 0$. If $v \geq -c$ for some constant $c > 0$ then it is monotone for all $0 \leq \delta \leq \frac{1}{2c}$. From there on the proof works verbatim as in [39, Lemma 2.6.4]. □

Remark 4.10. Obviously, these two lemmata have analogous versions for subharmonic functions, which are obtained by replacing u with $-u$, S_{ε}^{-} with S_{ε}^{+}, etc.

4.4 Existence and Uniqueness

Existence for AMLEs (2.16) or equivalently of solutions for the CDF problem (4.1) is well understood. Indeed, since $(\overline{\Omega}, d_{\Omega})$ is a length space, existence of solutions can be obtained using Perron's method, see [22, 23, 31] for details.

Using Lemmas 4.6 and 4.7 and the same strategy as in [1] we obtain uniqueness of solutions to the continuum problem (4.1) which we state in the following proposition.

Proposition 4.11 (Uniqueness)**.** *There exists at most one solution of (4.1).*

Proof. The proof works verbatim as for the main result in [1]. Letting $u, v \in C(\overline{\Omega})$ denote two solutions, Lemma 4.6 implies that

$$-\Delta_{\infty}^{\varepsilon} T^{\varepsilon} u(x_0) \leq 0 \leq -\Delta_{\infty}^{\varepsilon} T^{\varepsilon} v(x_0), \quad \forall x_0 \in \Omega_{\mathcal{O}}^{2\varepsilon}.$$

Now Lemma 4.7 implies

$$\sup_{\Omega_{\mathcal{O}}^{\varepsilon}} (T^{\varepsilon} u - T_{\varepsilon} v) = \sup_{\Omega_{\mathcal{O}}^{\varepsilon} \setminus \Omega_{\mathcal{O}}^{2\varepsilon}} (T^{\varepsilon} u - T_{\varepsilon} v).$$

Sending $\varepsilon \searrow 0$ implies

$$\sup_{\overline{\Omega}} (u - v) = \sup_{\mathcal{O}} (u - v) = 0$$

and hence $u \leq v$ in $\overline{\Omega}$. Swapping the roles of u and v finally implies that $u = v$ on $\overline{\Omega}$. □

5 Discrete-to-Continuum Convergence

In this section we first prove convergence rates of the graph distance functions defined in Section 3 to geodesic distance functions, in the continuum limit on geometric graphs. Then we utilize this to prove convergence rates of solutions to the graph infinity Laplacian equation (3.1) to the continuum problem (4.1).

We write $B(x, r) := \{y \in \mathbb{R}^d : |x - y| < r\}$ for the Euclidean open ball of radius $r > 0$ centered at $x \in \mathbb{R}^d$ and denote its closure by $\overline{B}(x, r)$. Recall that we define the geodesic distance $d_{\Omega}(x, y)$ on $\overline{\Omega}$ by

$$d_{\Omega}(x, y) = \inf \left\{ \int_0^1 |\dot{\xi}(t)| \, \mathrm{d}t : \xi \in C^1([0, 1]; \overline{\Omega}) \text{ with } \xi(0) = x \text{ and } \xi(1) = y \right\}.$$

If the line segment from x to y is contained in $\overline{\Omega}$, then $d_{\Omega}(x, y) = |x - y|$. Remember that we pose Assumption 2 on the relation between the geodesic and Euclidean distance. In the following we list important cases where the assumption is satisfied.

Proposition 5.1. *Let $\Omega \subset \mathbb{R}^d$ be an open and bounded domain.*

1. *If Ω is convex, Assumption 2 is satisfied with $\phi = 0$.*
2. *If Ω has a $C^{1,\alpha}$ boundary for $\alpha \in (0,1]$, Assumption 2 is satisfied with $\phi(h) = Ch^{1+\alpha}$ for some constant $C > 0$.*

Proof. The proof of 1 is immediate. The proof of 2 is split into 3 steps.

Step 1 We first show that for every $x, y \in \overline{\Omega}$ we have

$$d_\Omega(x,y) \le |x-y| + \sup_{z \in \partial\Omega \cap B(x,r)} \{d_\Omega(x,z) - |x-z|\}, \tag{5.1}$$

where $r = |x-y|$. To see this, we use a maximum principle argument. Let $u(z) = d_\Omega(x,z)$, $v(z) = |x-z|$, $\lambda > 1$ and set $V = \Omega \cap B(x,r)$. Let $z_* \in \overline{V}$ be a point where $u - \lambda v$ attains its maximum value over $\overline{V}$. Since u is a viscosity solution of $|\nabla u| = 1$ on $V \setminus \{x\}$ and $\lambda > 1$, we must have $z_* \in \partial V \cap \{x\}$. Now, we claim that $z_* \notin \Omega \cap \partial B(x,r)$, since if this were the case, then we could find a small $\varepsilon > 0$ such that $B(z_*, \varepsilon) \subset \Omega$, and setting $\nu = \frac{x - z_*}{r}$ we have

$$u(z_* + \varepsilon\nu) - \lambda v(z_* + \varepsilon\nu) \ge u(z_*) - \varepsilon - \lambda v(z_*) + \lambda\varepsilon = u(z_*) - \lambda v(z_*) + (\lambda - 1)\varepsilon,$$

which contradicts the optimality of z_*, as $z_* + \varepsilon\nu \in V$. Thus $z_* \in \partial\Omega \cap B(x,r)$ or $z_* = x$. If $z_* = x$ then $u - \lambda v \le 0$ on $\overline{V}$. If $z_* \in \partial\Omega \cap B(x,r)$ then

$$u - \lambda v \le \sup_{\partial\Omega \cap B(x,r)} \{u(z) - \lambda v\}.$$

Sending $\lambda \searrow 1$ establishes (5.1).

Step 2 We now show that

$$d_\Omega(x,z) \le |x-z| + C|x-z|^{1+\alpha}, \tag{5.2}$$

whenever $x \in \overline{\Omega}$, $z \in \partial\Omega$, and $|x-z| \le r_\Omega$, where $r_\Omega > 0$ will be determined below and C depends only on Ω. Without loss of generality, we may assume that $z = 0$, and that

$$\Omega \cap B_r = \{x \in B_r \,:\, x_d < \Phi(\overline{x})\}, \tag{5.3}$$

where $\overline{x} = (x_1, \dots, x_{d-1})$, $\Phi : \mathbb{R}^{d-1} \to \mathbb{R}$ is $C^{1,\alpha}$ with $\nabla\Phi(0) = 0$, and $0 < r \le r'_\Omega$, where r'_Ω depends only on Ω. Since $\partial\Omega$ is $C^{1,\alpha}$, there exists $C > 0$ so that

$$|\Phi(\overline{x})| \le C|x|^{1+\alpha} \quad \text{and} \quad |\nabla\Phi(\overline{x})| \le C|x|^\alpha. \tag{5.4}$$

Note we can write

$$d_\Omega(x,0) = \inf_\gamma \int_0^1 |\gamma'(t)|\, dt, \tag{5.5}$$

where the infimum is over C^1 curves $\gamma : [0,1] \to \overline{\Omega}$ satisfying $\gamma(0) = 0$ and $\gamma(1) = x$. Let us set

$$\gamma(t) = tx + (\Phi(t\overline{x}) - t\Phi(\overline{x}))e_d.$$

Then $\gamma(0) = 0$ and $\gamma(1) = x$. Since $x_d \le \Phi(\overline{x})$ we have

$$\gamma_d(t) = tx_d + \Phi(t\overline{x}) - t\Phi(\overline{x}) = t(x_d - \Phi(\overline{x_d})) + \Phi(t\overline{x}) \le \Phi(t\overline{x}) = \Phi\left(\overline{\gamma(t)}\right).$$

By (5.4) we have

$$|\gamma(t)| \le t|x| + |\Phi(t\overline{x})| + t|\Phi(\overline{x})| \le |x| + C|x|^{1+\alpha}.$$

Let us set $r = (1+C)|x|$ and restrict $|x| \le r_\Omega$, where $r_\Omega \le 1$ is sufficiently small so that $|x| \le r_\Omega$ implies $r \le r'_\Omega$ (i.e., $(1+C)r_\Omega \le r'_\Omega$). Since $|x| \le 1$ we have

$$|\gamma(t)| \le (1+C)|x| = r,$$

and so $\gamma(t) \in B_r$ for all $0 \le t \le 1$. It follows that $\gamma(t) \in \overline{\Omega} \cap B_r$ for all $0 \le t \le 1$. We now compute

$$\gamma'(t) = x + (\langle \nabla\Phi(t\overline{x}), \overline{x}\rangle - \Phi(\overline{x}))\, e_d,$$

and so by (5.4) we have

$$|\gamma'(t)| \le |x| + |\nabla\Phi(\overline{x})||x| + |\Phi(\overline{x})| \le |x| + C|x|^{1+\alpha}.$$

Substituting this into (5.5) completes the proof of (5.2).

Step 3 Combining steps 1 and 2 we find that

$$d_\Omega(x,y) \le |x-y| + C \sup_{z \in \partial\Omega \cap B(x,|x-y|)} |x-z|^{1+\alpha} \le |x-y| + C|x-y|^{1+\alpha},$$

provided $|x-y| \le r_\Omega$, which completes the proof. □

Remark 5.2. Assumption 2 is slightly stronger than the one made in [37] which takes the form

$$\limsup_{h \downarrow 0} \sup_{|x-y| \le h} \frac{d_\Omega(x,y)}{|x-y|} = 1.$$

This is an asymptotic version of Assumption 2 which was used in [37] to prove a Gamma-convergence result. However, for the quantitative analysis in this paper, this does not suffice.

Note that also non-smooth domains can satisfy Assumption 2. A particular interesting example is the following star-shaped subset of $\mathbb{R}^2$:

$$\Omega := \left\{ x \in [0,1]^2 \,:\, |x_1|^{2/3} + |x_2|^{2/3} \le 1 \right\}. \tag{5.6}$$

See Section 6 for figures and numerical examples on this domain. While the boundary $\partial\Omega$ is only $C^{0,\frac{2}{3}}$ Hölder regular, we still have the following result.

Proposition 5.3. *The domain (5.6) satisfies Assumption 2 with $\phi(h) = Ch^{\frac{3}{2}}$ for some constant $C > 0$ depending on Ω. It even holds that*

$$d_\Omega(x,y) \le |x-y| + C|x-y|^{\frac{3}{2}}, \quad \forall x, y \in \Omega. \tag{5.7}$$

Proof. We sketch a proof of this here. We first assume $x, y \in \Omega$ are in the same quadrant, which we can assume, by symmetry, to be the first quadrant $\Omega_1 := \Omega \cap [0,\infty)^2$. In this case, the shortest path between x and y must also lie in the quadrant Ω_1, since if it were ever to leave and subsequently return to the quadrant, we could construct a path with strictly shorter length by projecting the portion of the path that left Ω_1 back to the quadrant. Given this observation, we have that

$$d_\Omega(x,y) = d_{\Omega'}(x,y),$$

where the domain Ω' given by

$$\Omega' = \Omega \cup (\mathbb{R}^2 \setminus [0,\infty)^2).$$

It can be easily calculated that the domain Ω' has a $C^{1,\frac{1}{2}}$ boundary, and so it follows from part 2 of Proposition 5.1 that (5.7) holds whenever $x, y \in \Omega$ are in the same quadrant.

The case where x and y lie in different quadrants is handled by symmetry. We first consider the case where x and y lie in two different quadrants that belong to a common halfspace. Without loss of generality, we take $x \in \Omega_1$ and

$$y \in \Omega_2 = \Omega \cap [0, \infty) \times (-\infty, 0].$$

These quadrants belong to the common halfspace $H := [0, \infty) \times \mathbb{R}$, and a similar argument as above can be made to show that the shortest path between x and y must remain in $H \cap \Omega$. Let $z \in H \cap \partial\Omega_1 \cap \partial\Omega_2$ be the point on the line segment between x and y. Then by the triangle inequality we have

$$d_\Omega(x, y) \leq d_\Omega(x, z) + d_\Omega(z, y). \tag{5.8}$$

Since $x, z \in \overline{\Omega_1}$, and $z, y \in \overline{\Omega_2}$, we can apply (5.7) to obtain

$$d_\Omega(x, y) \leq |x - z| + |z - y| + C(|x - z|^{\frac{3}{2}} + |z - y|^{\frac{3}{2}}).$$

Since z is on the line segment between x and y we have $|x - z| + |z - y| = |x - y|$ and so

$$d_\Omega(x, y) \leq |x - y| + 2C|x - y|^{\frac{3}{2}}.$$

The last case is where x and y belong to diagonally opposing quadrants. Without loss we can consider $x \in \Omega_1$ and

$$y \in \Omega_3 = \Omega \cap (-\infty, 0]^2.$$

Here, we let $z \in \partial\Omega_3$ be the point along the line segment between x and y. Since the points x and z lie in the same halfspace, the previous step shows that

$$d_\Omega(x, z) \leq |x - z| + 2C|x - z|^{\frac{3}{2}}.$$

Since z and y lie in the same quadrant we have

$$d_\Omega(z, y) \leq |z - y| + C|z - y|^{\frac{3}{2}}.$$

Inserting these into the triangle inequality (5.8) yields

$$d_\Omega(x, y) \leq |x - y| + 3C|x - y|^{\frac{3}{2}},$$

which completes the proof. □

5.1 Geometric Graphs

Remember that we consider a set of points $\Omega_n \subset \overline{\Omega}$ and a subset $\mathcal{O}_n \subset \Omega_n$ of labelled vertices, which acts as the discrete analog of $\mathcal{O}$. On the point cloud Ω_n we define the geometric graph $\mathbf{G}_{n,h} = (\Omega_n, \mathbf{W}_{n,h})$ with vertices Ω_n and a set of edge weights $\mathbf{W}_{n,h} = (\mathbf{w}_{n,h}(\mathbf{x}, \mathbf{y}))_{\mathbf{x},\mathbf{y} \in \Omega_n}$ given by

$$\mathbf{w}_{n,h}(\mathbf{x}, \mathbf{y}) = \begin{cases} 0, & \mathbf{x} = \mathbf{y}, \\ \sigma_\eta^{-1} \eta_h(|\mathbf{x} - \mathbf{y}|), & \mathbf{x} \neq \mathbf{y}, \end{cases} \tag{5.9}$$

where

$$\sigma_\eta = \sup_{t>0} t\eta(t). \tag{5.10}$$

By Assumption 1 we can choose $t_0 \in (0, 1]$ as a maximum of $t\eta(t)$, so that $\sigma_\eta = t_0\eta(t_0)$. When the value of t_0 is not unique, the convergence rates below are slightly better by choosing the largest such t_0.

To simplify notation we will write $\mathbf{G}_n$ in place of $\mathbf{G}_{n,h}$, and $\mathbf{w}_n$ in place of $\mathbf{w}_{n,h}$, when the value of h is clear from the context. We will denote the graph distance function $\mathbf{d}_{\mathbf{G}_n}$, defined in (3.3) by $\mathbf{d}_n : \Omega_n \times \Omega_n \to \mathbb{R}$ and the Lipschitz constant of a function $\mathbf{u}_n : \Omega_n \to \mathbb{R}$ as $\mathrm{Lip}_n(\mathbf{u}_n) := \mathrm{Lip}_{\mathbf{G}_n}(\mathbf{u}_n)$.

5.2 Convergence of Cones

The proof of convergence of the graph distance $\mathbf{d}_n$ to the distance function d_Ω is split into two parts. Lemma 5.4 gives the lower bound, while Lemma 5.5 gives the upper.

Lemma 5.4. *Let Assumptions 1 and 2 hold. Then for $h \le r_\Omega$ and all $\mathbf{x}, \mathbf{y} \in \Omega_n$ we have*

$$\mathbf{d}_n(\mathbf{x}, \mathbf{y}) \ge |\mathbf{x} - \mathbf{y}| \vee \left(1 - \frac{\phi(h)}{h}\right) d_\Omega(\mathbf{x}, \mathbf{y}). \tag{5.11}$$

Proof. Let $\mathbf{x}, \mathbf{y} \in \Omega_n$. If there is no path in $\mathbf{G}_n$ from $\mathbf{x}$ to $\mathbf{y}$ then $\mathbf{d}_n(x, y) = \infty$ and (5.11) holds trivially. Thus, we may assume there is a path

$$\mathbf{x} = \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m = \mathbf{y}$$

such that $\mathbf{w}_n(\mathbf{x}_i, \mathbf{x}_{i+1}) > 0$ for all $i = 1, \dots, m-1$. We can choose the path to be optimal for $\mathbf{x}, \mathbf{y}$, so that

$$\mathbf{d}_n(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{m-1} \mathbf{w}_n(\mathbf{x}_i, \mathbf{x}_{i+1})^{-1}.$$

It follows that

$$\mathbf{d}_n(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{m-1} \sigma_\eta \eta_h(|\mathbf{x}_i - \mathbf{x}_{i+1}|)^{-1} = \sum_{i=1}^{m-1} \sigma_\eta h \eta \left(\frac{|\mathbf{x}_i - \mathbf{x}_{i+1}|}{h}\right)^{-1}.$$

By definition of σ_η, (5.10), we have $\sigma_\eta \eta(t)^{-1} \ge t$, and so

$$\mathbf{d}_n(\mathbf{x}, \mathbf{y}) \ge \sum_{i=1}^{m-1} |\mathbf{x}_i - \mathbf{x}_{i+1}|.$$

It follows from the triangle inequality that $\mathbf{d}_n(\mathbf{x}, \mathbf{y}) \ge |\mathbf{x} - \mathbf{y}|$. Also, using Assumption 2 we have

$$d_\Omega(\mathbf{x}_i, \mathbf{x}_{i+1}) \le |\mathbf{x}_i - \mathbf{x}_{i+1}| + \phi(|\mathbf{x}_i - \mathbf{x}_{i+1}|) \le |\mathbf{x}_i - \mathbf{x}_{i+1}| + d_\Omega(\mathbf{x}_i, \mathbf{x}_{i+1}) \frac{\phi(h)}{h}.$$

Substituting this above yields

$$\mathbf{d}_n(\mathbf{x}, \mathbf{y}) \ge \left(1 - \frac{\phi(h)}{h}\right) \sum_{i=1}^{m-1} d_\Omega(\mathbf{x}_i, \mathbf{x}_{i+1}) \ge \left(1 - \frac{\phi(h)}{h}\right) d_\Omega(\mathbf{x}, \mathbf{y}),$$

where we again used the triangle inequality, this time for d_Ω. □

We now give the proof of the upper bound.

Lemma 5.5. *Let Assumptions 1 and 2 hold. Assume that $h \le r_\Omega$ and that*

$$\frac{\delta_n}{h} \le \frac{t_0}{2(2 + \frac{\phi(\delta_n)}{\delta_n})}. \tag{5.12}$$

Then for any $\mathbf{x}, \mathbf{y} \in \Omega_n$ we have

$$\mathbf{d}_n(\mathbf{x}, \mathbf{y}) \le \left(1 + \frac{4\delta_n}{t_0 h} + \frac{2\phi(\delta_n)}{t_0 h}\right) d_\Omega(\mathbf{x}, \mathbf{y}) + \tau_\eta h, \tag{5.13}$$

where

$$\tau_\eta := \sup_{0 < t \le t_0} \left\{\sigma_\eta \eta(t)^{-1} - t\right\}. \tag{5.14}$$

Remark 5.6. We note that since η is nonincreasing, for $t \le t_0$ we have

$$\sigma_\eta \eta(t)^{-1} \le \sigma_\eta \eta(t_0)^{-1} = \sigma_\eta t_0 \sigma_\eta^{-1} = t_0.$$

Therefore, $\tau_\eta \le t_0 \le 1$. In the special case of the singular kernel $\eta(t) = t^{-1}$ we have $t\eta(t) = 1$, and so $\sigma_\eta = 1$. Hence, it holds $\sigma_\eta \eta(t)^{-1} = t$ and $\tau_\eta = 0$.

In fact, we can show that if η is any kernel satisfying $\tau_\eta = 0$ then $\eta(t) = \sigma_\eta t^{-1}$ for $t \in (0, t_0]$. To see this, note that if $\tau_\eta = 0$ then $\sigma_\eta \le t\eta(t)$ for all $0 < t \le t_0$. Since $\sigma_\eta = \sup_{0<t\le 1} t\eta(t) \ge t\eta(t)$, we have $\sigma_\eta = t\eta(t)$ for all $0 < t \le t_0$, which establishes the claim.

Fig. 1 illustrates the improved approximation accuracy obtained by using the singular kernel $\eta(t) = t^{-1}$, compared to the uniform kernel $\eta = \mathbb{1}_{[0,1]}$. For the uniform kernel, the graph cone gives a roughly piecewise constant staircasing approximation of the true Euclidean cone, with steps of size $O(h)$, due to the additional $\tau_\eta h$ term in Lemma 5.5. These artifacts are removed by using the singular kernel $\eta(t) = t^{-1}$.

Remark 5.7. The condition in (5.12) is a consequence of Assumption 4. Since $\sigma_\phi(h) \ge \frac{\phi(h)}{h} \ge 0$, the assumption implies

$$\frac{\delta_n}{h} + \frac{\phi(h_n)}{h} \le \frac{t_0}{4}\left(1 - 2\frac{h}{\varepsilon}\right)$$

and in particular $\delta_n < h$. Therefore, it holds $\phi(\delta_n) \le \sigma_\phi(h)\delta_n \le \sigma_\phi(h)h$ and we can estimate, using Assumption 4 again,

$$\frac{\delta_n}{h}\left(2 + \frac{\phi(\delta_n)}{\delta_n}\right) = 2\frac{\delta_n}{h} + \frac{\phi(\delta_n)}{h} \le 2\left(\frac{\delta_n}{h} + \sigma_\phi(h)\right) \le \frac{t_0}{2}\left(1 - 2\frac{h}{\varepsilon}\right) \le \frac{t_0}{2}$$

and with this

$$\frac{\delta_n}{h} \le \frac{t_0}{2\left(2 + \frac{\phi(\delta_n)}{\delta_n}\right)}.$$

Proof. Let $\mathbf{x}, \mathbf{y} \in \Omega_n$. We construct a sequence $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \dots$ as follows. Define $\mathbf{x}_0 = \mathbf{x}$ and for $k \ge 1$ set

$$\mathbf{x}_k \in \operatorname*{arg\,min}_{\Omega_n \cap \overline{B}(x_{k-1}, t_0 h)} d_\Omega(\cdot, \mathbf{y}).$$

If $\mathbf{y} \in \overline{B}(\mathbf{x}_{k-1}, t_0 h)$ then we set $\mathbf{x}_k = \mathbf{y}$ and the procedure terminates. We claim that whenever $\mathbf{y} \notin \overline{B}(\mathbf{x}_{k-1}, t_0 h)$ we have

$$d_\Omega(\mathbf{x}_k, \mathbf{y}) \le d_\Omega(\mathbf{x}_{k-1}, \mathbf{y}) - (t_0 h - 2\delta_n - \phi(\delta_n)). \tag{5.15}$$

To see this, we start from the dynamic programming principle

$$d_\Omega(\mathbf{x}_{k-1}, \mathbf{y}) = \min_{z \in \overline{\Omega} \cap \partial B(\mathbf{x}_{k-1}, t_0 h - \delta_n)} \left\{ d_\Omega(\mathbf{x}_{k-1}, z) + d_\Omega(z, \mathbf{y}) \right\},$$

which is valid since (5.12) implies that $\delta_n < t_0 h$. Choosing an optimal z above yields

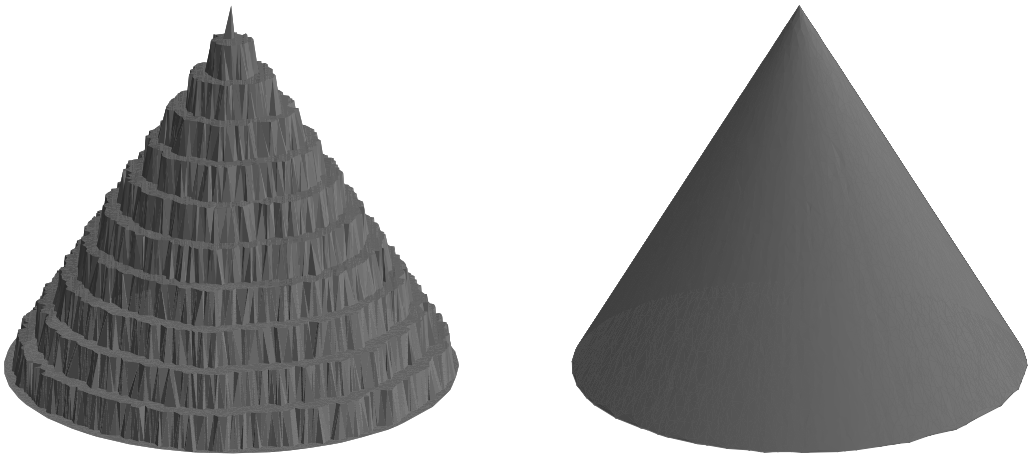
$$d_\Omega(z, \mathbf{y}) \le d_\Omega(\mathbf{x}_{k-1}, \mathbf{y}) - t_0 h + \delta_n. \tag{5.16}$$

where we used that $d_\Omega(\mathbf{x}_{k-1}, z) \ge |\mathbf{x}_{k-1} - z| = t_0 h - \delta_n$. We also have by (2.5) that

$$|\pi_n(z) - \mathbf{x}_{k-1}| \le |\pi_n(z) - z| + |z - \mathbf{x}_{k-1}| \le \delta_n + t_0 h - \delta_n = t_0 h.$$

Thus $\pi_n(z) \in \Omega_n \cap \overline{B}(x_{k-1}, t_0 h)$. Furthermore, by Assumption 2 and (5.16), $\pi_n(z)$ satisfies

$$d_\Omega(\pi_n(z), \mathbf{y}) \le d_\Omega(z, \mathbf{y}) + d_\Omega(\pi_n(z), z) \le d_\Omega(\mathbf{x}_{k-1}, \mathbf{y}) - t_0 h + \delta_n + \delta_n + \phi(\delta_n).$$



(a) $\eta = 1_{[0,1]}$ (b) $\eta(t) = t^{-1}$

Figure 1: Examples of graph cones (graph distance functions to a point) computed with (a) the nonsingular kernel $\eta = 1\!\!1_{[0,1]}$ and (b) the singular kernel $\eta(t) = t^{-1}$. The graph consists of an *i.i.d.* sample of size $n = 10^4$ uniformly distributed on the unit ball $B(0,1)$, and we set $h = 0.1$.

This establishes the claim (5.15). Let us define

$$r_0 := t_0 - 2\frac{\delta_n}{h} - \frac{\phi(\delta_n)}{h},$$

and note that (5.12) implies that $\delta_n \leq h$ and so

$$1 \geq \frac{r_0}{t_0} \geq 1 - 2\frac{\delta_n}{t_0 h} - \frac{\phi(\delta_n)}{t_0 h} = 1 - \frac{\delta_n}{t_0 h}\left(2 + \frac{\phi(\delta_n)}{\delta_n}\right) \geq \frac{1}{2}, \tag{5.17}$$

where we again used (5.12) in the last inequality. Therefore $r_0 > 0$ and so by (5.15) we have

$$d_\Omega(\mathbf{x}_k, \mathbf{y}) \leq d_\Omega(\mathbf{x}_{k-1}, \mathbf{y}) - r_0 h$$

and hence

$$d_\Omega(\mathbf{x}_k, \mathbf{y}) \leq d_\Omega(\mathbf{x}, \mathbf{y}) - r_0 h k,$$

provided $\mathbf{y} \notin \overline{B}(\mathbf{x}_j, t_0 h)$ for all $j = 0, \ldots, k-1$. Hence, the condition $\mathbf{y} \notin \overline{B}(\mathbf{x}_j, t_0 h)$ for all $j = 0, \ldots, k-1$ implies that

$$k \leq \frac{d_\Omega(\mathbf{x}, \mathbf{y}) - d_\Omega(\mathbf{x}_k, \mathbf{y})}{r_0 h}. \tag{5.18}$$

It follows that eventually $\mathbf{y} \in \overline{B}(\mathbf{x}_{k-1}, t_0 h)$ and $\mathbf{x}_k = \mathbf{y}$ for some k. Let $T \geq 1$ denote the smallest integer such that $\mathbf{x}_T = \mathbf{y}$, and note that (5.18) implies

$$T \leq \frac{d_\Omega(\mathbf{x}, \mathbf{y}) - d_\Omega(\mathbf{x}_{T-1}, \mathbf{y})}{r_0 h} + 1. \tag{5.19}$$

We now compute

$$\mathbf{d}_n(\mathbf{x}, \mathbf{y}) \leq \sum_{k=1}^{T} \mathbf{w}_n(\mathbf{x}_{k-1}, \mathbf{x}_k)^{-1} = \sum_{k=1}^{T} \sigma_\eta \eta_h(|\mathbf{x}_{k-1} - \mathbf{x}_k|)^{-1} = \sum_{k=1}^{T} h\sigma_\eta \eta\left(\frac{|\mathbf{x}_{k-1} - \mathbf{x}_k|}{h}\right)^{-1}.$$

Since $|\mathbf{x}_{k-1} - \mathbf{x}_k| \leq t_0 h$ and η is nonincreasing we have

$$\eta\left(\frac{|\mathbf{x}_{k-1} - \mathbf{x}_k|}{h}\right)^{-1} \leq \eta\left(t_0\right)^{-1}.$$

It follows that

$$\begin{aligned}
\mathbf{d}_n(\mathbf{x}, \mathbf{y}) &\leq \sum_{k=1}^{T-1} h\sigma_\eta \eta(t_0)^{-1} + h\sigma_\eta \eta\left(\frac{|\mathbf{x}_{T-1} - \mathbf{y}|}{h}\right)^{-1} \\
&= \sum_{k=1}^{T-1} t_0 h + h\sigma_\eta \eta\left(\frac{|\mathbf{x}_{T-1} - \mathbf{y}|}{h}\right)^{-1} \\
&= t_0 h(T-1) + h\sigma_\eta \eta\left(\frac{|\mathbf{x}_{T-1} - \mathbf{y}|}{h}\right)^{-1} \\
&\leq t_0 h\left(\frac{d_\Omega(\mathbf{x}, \mathbf{y}) - d_\Omega(\mathbf{x}_{T-1}, \mathbf{y})}{r_0 h}\right) + h\sigma_\eta \eta\left(\frac{|\mathbf{x}_{T-1} - \mathbf{y}|}{h}\right)^{-1} \\
&\leq \frac{t_0}{r_0} d_\Omega(\mathbf{x}, \mathbf{y}) + h\sigma_\eta \eta\left(\frac{|\mathbf{x}_{T-1} - \mathbf{y}|}{h}\right)^{-1} - |\mathbf{x}_{T-1} - \mathbf{y}| \qquad (5.20) \\
&\leq \frac{t_0}{r_0} d_\Omega(\mathbf{x}, \mathbf{y}) + \tau_\eta h, \qquad (5.21)
\end{aligned}$$

where we used that $t_0 \geq r_0$ and $d_\Omega(\mathbf{x}_{T-1}, \mathbf{y}) \geq |\mathbf{x}_{T-1} - \mathbf{y}|$ in the second to last inequality, and $\mathbf{x}_{T-1} \in \overline{B}(\mathbf{y}, t_0 h)$ in the last. By (5.17) we have

$$\frac{t_0}{r_0} = \frac{1}{1 - \left(1 - \frac{r_0}{t_0}\right)} \leq 1 + 2\left(1 - \frac{r_0}{t_0}\right) = 1 + \frac{4\delta_n}{t_0 h} + \frac{2\phi(\delta_n)}{t_0 h},$$

which completes the proof. □

5.3 Convergence of Infinity Harmonic Functions

In the previous sections we have analyzed the continuum problem (4.1), introduced the corresponding discrete problem on a graph, and proved convergence of graph cone functions to continuum Euclidean cone functions. These are all the ingredients which we need in order to finally show that the solution of (3.1) converges to the solution of (4.1) and establish convergence rates.

For this we need to introduce discrete analogies of the operators T^ε and T_ε from (4.6). For a graph function $\mathbf{u}_n : \Omega_n \to \mathbb{R}$ we define $u_n^\varepsilon, (u_n)_\varepsilon : \overline{\Omega} \to \mathbb{R}$ by

$$u_n^\varepsilon(x) := \sup_{\overline{B}_\Omega(x,\varepsilon)\cap\Omega_n} \mathbf{u}_n, \quad (u_n)_\varepsilon(x) := \inf_{\overline{B}_\Omega(x,\varepsilon)\cap\Omega_n} \mathbf{u}_n, \quad x \in \overline{\Omega}. \qquad (5.22)$$

Before we proceed with a discrete-to-nonlocal consistency statement and prove Theorem 2.2, we devote a section to collecting several technical lemmas, dealing with (approximate) Lipschitz continuity of the quantities involved. For understanding the gist of our arguments the following section can be skipped, however.

5.3.1 Approximate Lipschitz Estimates

Lemma 5.8. *Under Assumptions 1, 2 and 4 there exists a constant $C > 0$ such that for all $\mathbf{u}_n : \Omega_n \to \mathbb{R}$ and all $x, y \in \overline{\Omega}$ it holds*

$$|\Delta_\infty^\varepsilon u_n^\varepsilon(x) - \Delta_\infty^\varepsilon u_n^\varepsilon(y)| \leq \mathrm{Lip}_n(\mathbf{u}_n) C \frac{|x-y| + \phi(|x-y|) + 2\delta_n + \phi(2\delta_n) + \tau_\eta h}{\varepsilon^2}. \qquad (5.23)$$

Proof. We first note that by definition (4.8)

$$\begin{aligned}\varepsilon^2 \left|\Delta_\infty^\varepsilon u_n^\varepsilon(x) - \Delta_\infty^\varepsilon(y)\right| &\leq \left|T^\varepsilon u_n^\varepsilon(x) - T^\varepsilon u_n^\varepsilon(y)\right| + \left|T_\varepsilon u_n^\varepsilon(x) - T_\varepsilon u_n^\varepsilon(y)\right| + 2\left|u_n^\varepsilon(x) - u_n^\varepsilon(y)\right| \\ &= \left|u_n^{2\varepsilon}(x) - u_n^{2\varepsilon}(y)\right| + \left|T_\varepsilon u_n^\varepsilon(x) - T_\varepsilon u_n^\varepsilon(y)\right| + 2\left|u_n^\varepsilon(x) - u_n^\varepsilon(y)\right|.\end{aligned}$$

We first estimate the third term and then argue that the same estimate also applies to the first and second.

For $x \in \overline{\Omega}$ we know that there exists a point $\mathbf{x}^* \in \overline{B}_\Omega(x,\varepsilon) \cap \Omega_n$ such that

$$u_n^\varepsilon(x) = \mathbf{u}_n(\mathbf{x}^*).$$

For $y \in \overline{\Omega}$ we construct a point $\tilde{y} \in B_\Omega(y,\varepsilon)$ as follows. Since (Ω, d_Ω) is a length space we can find a geodesic $\gamma : [0,1] \to \Omega$ with $\gamma(0) = y$, $\gamma(1) = \mathbf{x}^*$, and length $d_\Omega(y,\mathbf{x}^*)$. We define $\tilde{y} := \gamma(t_*)$ where $t_* := \sup\{t > 0 : \gamma(t) \in B_\Omega(y,\varepsilon)\}$ as the last point which still lies in $\overline{B}_\Omega(y,\varepsilon)$. We distinguish two cases: If $\mathbf{x}^* \in \overline{B}_\Omega(y,\varepsilon)$ then $\tilde{y} = \mathbf{x}^*$ and hence $d_\Omega(\tilde{y},\mathbf{x}^*) = 0 \leq d_\Omega(x,y)$. If $\mathbf{x}^* \notin \overline{B}_\Omega(y,\varepsilon)$ then it holds $d_\Omega(y,\tilde{y}) = \varepsilon$. Furthermore, since γ is a geodesic it holds

$$d_\Omega(y,\mathbf{x}^*) = d_\Omega(y,\tilde{y}) + d_\Omega(\tilde{y},\mathbf{x}^*).$$

Hence, we obtain that also in this case it holds

$$d_\Omega(\tilde{y},\mathbf{x}^*) = d_\Omega(y,\mathbf{x}^*) - d_\Omega(y,\tilde{y}) \leq d_\Omega(x,y) + d_\Omega(x,\mathbf{x}^*) - d_\Omega(y,\tilde{y}) \leq d_\Omega(x,y).$$

By definition of the graph resolution δ_n there exists a point $\mathbf{y}^* \in \overline{B}_\Omega(y,\varepsilon) \cap \Omega_n$, such that $|\tilde{y} - \mathbf{y}^*| \leq 2\delta_n$ and by definition of u_n^ε it holds $\mathbf{u}_n(\mathbf{y}^*) \leq u_n^\varepsilon(y)$. Furthermore, by Assumption 2

$$d_\Omega(\mathbf{x}^*,\mathbf{y}^*) \leq d_\Omega(\tilde{y},\mathbf{x}^*) + d_\Omega(\tilde{y},\mathbf{y}^*) \leq d_\Omega(x,y) + 2\delta_n + \phi(2\delta_n).$$

Thus, using Lemma 5.5 and Assumptions 2 and 4 it follows that

$$\begin{aligned}u_n^\varepsilon(x) - u_n^\varepsilon(y) &\leq \mathbf{u}_n(\mathbf{x}^*) - \mathbf{u}_n(\mathbf{y}^*) \\ &\leq \mathrm{Lip}_n(\mathbf{u}_n)\, \mathbf{d}_n(\mathbf{x}^*,\mathbf{y}^*) \\ &\leq \mathrm{Lip}_n(\mathbf{u}_n)\left(C\, d_\Omega(\mathbf{x}^*,\mathbf{y}^*) + \tau_\eta h\right) \\ &\leq \mathrm{Lip}_n(\mathbf{u}_n)\left(C\left(d_\Omega(x,y) + 2\delta_n + \phi(2\delta_n)\right) + \tau_\eta h\right) \\ &\leq \mathrm{Lip}_n(\mathbf{u}_n)\, C\left(|x-y| + \phi(|x-y|) + 2\delta_n + \phi(2\delta_n) + \tau_\eta h\right).\end{aligned}$$

Exchanging the role of x and y we can estimate the difference $u_n^\varepsilon(y) - u_n^\varepsilon(x)$. The resulting estimate on $|u_n^\varepsilon(x) - u_n^\varepsilon(y)|$ does not depend on the choice of ε and is therefore also valid for $\left|u_n^{2\varepsilon}(x) - u_n^{2\varepsilon}(y)\right|$. Finally, we can estimate

$$\begin{aligned}T_\varepsilon u_n^\varepsilon(x) - T_\varepsilon u_n^\varepsilon(y) &= \inf_{B_\Omega(x,\varepsilon)} u_n^\varepsilon - \inf_{B_\Omega(y,\varepsilon)} u_n^\varepsilon \\ &= \inf_{B_\Omega(x,\varepsilon)} u_n^\varepsilon - u_n^\varepsilon(y^*),\end{aligned}$$

where $y^* \in \overline{B}_\Omega(y,\varepsilon)$ realizes the infimum. With the same trick as above we can choose a geodesic $\gamma : [0,1] \to \Omega$ from y^* to x and define the first point lying in $B_\Omega(x,\varepsilon)$ as $x^* := \gamma(t^*)$ where

$$t^* := \inf\{t > 0 : \gamma(t) \in B_\Omega(x,\varepsilon)\}.$$

Either $y^* \in B_\Omega(x,\varepsilon)$ and therefore $x^* = y^*$ or, similar as before, it holds $d_\Omega(x,x^*) = \varepsilon$ and

$$d_\Omega(y^*,x^*) = d_\Omega(y^*,x) - d_\Omega(x^*,x) \leq d_\Omega(x,y) + d_\Omega(y^*,y) - d_\Omega(x^*,x) \leq d_\Omega(x,y).$$

Hence, as before we can estimate

$$\begin{aligned} T_\varepsilon u_n^\varepsilon(x) - T_\varepsilon u_n^\varepsilon(y) &\le u_n^\varepsilon(x^*) - u_n^\varepsilon(y^*) \\ &\le \mathrm{Lip}_n(\mathbf{u}_n)\left(C\,(\underbrace{d_\Omega(x^*,y^*)}_{\le d_\Omega(x,y)}) + 2\delta_n + \phi(2\delta_n) + \tau_\eta h\right) \\ &\le \mathrm{Lip}_n(\mathbf{u}_n)\, C\left(|x-y| + \phi(|x-y|) + 2\delta_n + \phi(2\delta_n) + \tau_\eta h\right). \end{aligned}$$

□

Lemma 5.9. *For $u : \overline{\Omega} \to \mathbb{R}$ it holds*

$$\sup_{\overline{\Omega}} |T^\varepsilon u - u| \vee \sup_{\overline{\Omega}} |T_\varepsilon u - u| \le \mathrm{Lip}_\Omega(u)\,\varepsilon. \tag{5.24}$$

Proof. We only give a proof of the first inequality since the one for $T_\varepsilon u$ works analogously. For $x \in \overline{\Omega}$ one computes

$$T^\varepsilon u(x) - u(x) = \max_{y \in B_\Omega(x,\varepsilon)} u(y) - u(x) \le \mathrm{Lip}_\Omega(u) \max_{y \in B_\Omega(x,\varepsilon)} d_\Omega(x,y) \le \mathrm{Lip}_\Omega(u)\,\varepsilon, \quad \forall x \in \Omega_\mathcal{O}^\varepsilon,$$

which implies $\sup_{\Omega_\mathcal{O}^\varepsilon} |T^\varepsilon - u| \le \mathrm{Lip}_\Omega(u)\,\varepsilon$. □

Lemma 5.10. *Under Assumptions 1, 2 and 4 there exists a constant $C > 0$ such that for all $\mathbf{u}_n : \Omega_n \to \mathbb{R}$ it holds*

$$\sup_{\overline{\Omega}} |u_n^\varepsilon - u_n| \vee \sup_{\overline{\Omega}} |(u_n)_\varepsilon - u_n| \le C\,\mathrm{Lip}_n(\mathbf{u}_n)\varepsilon, \tag{5.25}$$

where $u_n : \overline{\Omega} \to \mathbb{R}$ is the piecewise constant extension of $\mathbf{u}_n$, defined in (2.8).

Proof. Again we only prove the first estimate. For every $x \in \Omega$ it holds thanks to Lemma 5.5 and Assumptions 2 and 4 that

$$\begin{aligned} u_n^\varepsilon(x) - u_n(x) &= \max_{\overline{B}_\Omega(x,\varepsilon)\cap\Omega_n} \mathbf{u}_n - \mathbf{u}_n(\pi_n(x)) \le \mathrm{Lip}_n(\mathbf{u}_n) \max_{\mathbf{y} \in \overline{B}_\Omega(x,\varepsilon)\cap\Omega_n} \mathbf{d}_n(\pi_n(x),\mathbf{y}) \\ &\le \mathrm{Lip}_n(\mathbf{u}_n) \max_{\mathbf{y} \in \overline{B}_\Omega(x,\varepsilon)\cap\Omega_n} (C d_\Omega(\pi_n(x),\mathbf{y}) + \tau_\eta h) \\ &\le \mathrm{Lip}_n(\mathbf{u}_n) \max_{\mathbf{y} \in \overline{B}_\Omega(x,\varepsilon)\cap\Omega_n} (C(d_\Omega(x,\mathbf{y}) + d_\Omega(x,\pi_n(x)) + \tau_\eta h) \\ &\le \mathrm{Lip}_n(\mathbf{u}_n)\,(C(\varepsilon + \delta_n + \phi(\delta_n)) + \tau_\eta h) \\ &\le C\,\mathrm{Lip}_n(\mathbf{u}_n)\,\varepsilon, \end{aligned}$$

where the constant $C > 0$ changed its value. □

Lemma 5.11. *Under Assumptions 1 to 4 there exists a constant $C > 0$ such that for all $\mathbf{u}_n : \Omega_n \to \mathbb{R}$ with $\mathbf{u}_n = \mathbf{g}_n$ on $\mathcal{O}_n$ and all $u : \overline{\Omega} \to \mathbb{R}$ with $u = g$ on $\mathcal{O}$ it holds*

$$\sup_{\Omega_\mathcal{O}^\varepsilon \setminus \Omega_\mathcal{O}^{2\varepsilon}} (u_n^\varepsilon - T_\varepsilon u) \le C(\mathrm{Lip}_n(\mathbf{u}_n) + \mathrm{Lip}_\Omega(u))\varepsilon.$$

Proof. Using Lemmas 5.9 and 5.10 we have that

$$\begin{aligned} \sup_{\Omega_\mathcal{O}^\varepsilon \setminus \Omega_\mathcal{O}^{2\varepsilon}} (u_n^\varepsilon - T_\varepsilon u) &\le \sup_{\Omega_\mathcal{O}^\varepsilon \setminus \Omega_\mathcal{O}^{2\varepsilon}} \left(|u_n^\varepsilon - u_n| + |u_n - u| + |u - T_\varepsilon u|\right) \\ &\le (\mathrm{Lip}_n(\mathbf{u}_n) + \mathrm{Lip}_\Omega(u))C\varepsilon + \sup_{\Omega_\mathcal{O}^\varepsilon \setminus \Omega_\mathcal{O}^{2\varepsilon}} |u_n - u|\,. \end{aligned}$$

Since $x \in \Omega_{\mathcal{O}}^{\varepsilon} \setminus \Omega_{\mathcal{O}}^{2\varepsilon}$ we know that there exists $z \in \mathcal{O}$ such that $|z - x| \leq 2\varepsilon$, for which we also find a vertex $\pi_{\mathcal{O}_n}(z) \in \mathcal{O}_n$ with $|z - \pi_{\mathcal{O}_n}(z)| \leq \delta_n$ and therefore $|x - \pi_{\mathcal{O}_n}(z)| \leq 2\varepsilon + \delta_n$. This also implies that $|\pi_n(x) - \pi_{\mathcal{O}_n}(z)| \leq 2\varepsilon + 2\delta_n$. With similar computations as before, using Assumptions 3 and 4 and Lemmas 5.5, 5.9 and 5.10, we have

$$\begin{aligned}
|u_n(x) - u(x)| &\leq |\mathbf{u}_n(\pi_n(x)) - \mathbf{g}_n(\pi_{\mathcal{O}_n}(z))| + |\mathbf{g}_n(\pi_{\mathcal{O}_n}(z)) - g(z)| + |g(z) - u(x)| \\
&= |\mathbf{u}_n(\pi_n(x)) - \mathbf{u}_n(\pi_{\mathcal{O}_n}(z))| + |\mathbf{g}_n(\pi_{\mathcal{O}_n}(z)) - g(z)| + |u(z) - u(x)| \\
&\leq \mathrm{Lip}_n(\mathbf{u}_n)\mathbf{d}_n(\pi_n(x), \pi_{\mathcal{O}_n}(z)) + |\mathbf{g}_n(\pi_{\mathcal{O}_n}(z)) - g(z)| + \mathrm{Lip}_\Omega(u) d_\Omega(z, x) \\
&\leq C(\mathrm{Lip}_n(\mathbf{u}_n) + \mathrm{Lip}_\Omega(u))\varepsilon.
\end{aligned}$$

□

Lemma 5.12. *Under Assumptions 1, 2 and 4 there exists a constant $C > 0$ such that for all $\mathbf{u} : \Omega_n \to \mathbb{R}$ it holds*

$$|u_n(x) - u_n(y)| \leq \mathrm{Lip}_n(\mathbf{u}_n) \left(C\big(d_\Omega(x, y) + 2\delta_n + 2\phi(\delta_n)\big) + \tau_\eta h \right), \quad \forall x, y \in \overline{\Omega},$$

where $u_n : \overline{\Omega} \to \mathbb{R}$ is the piecewise constant extension of $\mathbf{u}_n$, defined in (2.8).

Proof. Using Lemma 5.5 and Assumptions 2 and 4 and the definition of u_n in (2.8) it holds

$$\begin{aligned}
|u_n(x) - u_n(y)| &= |\mathbf{u}_n(\pi_n(x)) - \mathbf{u}_n(\pi_n(y))| \\
&\leq \mathrm{Lip}_n(\mathbf{u}_n)\mathbf{d}_n(\pi_n(x), \pi_n(y)) \\
&\leq \mathrm{Lip}_n(\mathbf{u}_n) \left(C\, d_\Omega(\pi_n(x), \pi_n(y)) + \tau_\eta h \right) \\
&\leq \mathrm{Lip}_n(\mathbf{u}_n) \left(C \left(d_\Omega(\pi_n(x), x) + d_\Omega(x, y) + d_\Omega(y, \pi_n(y)) \right) + \tau_\eta h \right) \\
&\leq \mathrm{Lip}_n(\mathbf{u}_n) \left(C \left(d_\Omega(x, y) + 2\delta_n + 2\phi(\delta_n) \right) + \tau_\eta h \right).
\end{aligned}$$

□

5.3.2 Discrete-to-Nonlocal Consistency

In this section we will prove that if $\mathbf{u}_n$ is a solution of the graph infinity Laplace equation (2.1) then a discrete-to-continuum version of the max-ball statement Lemma 4.6 holds true. Using only the comparison with graph distance functions, established in Theorem 3.2, we shall prove

$$-\Delta_\infty^\varepsilon u_n^\varepsilon \leq C(\delta_n, h, \varepsilon) \quad \text{and} \quad -\Delta_\infty^\varepsilon (u_n)_\varepsilon \geq -C(\delta_n, h, \varepsilon),$$

where $C(\delta_n, h, \varepsilon) > 0$ is small if δ_n and h are small. This means that the extension operators (5.22) turn discrete solutions into approximate sub- and supersolutions of the nonlocal equation $-\Delta_\infty^\varepsilon u = 0$. Compared to the continuum setting from Lemma 4.6, the inhomogeneity $C(\delta_n, h, \varepsilon)$ originates from the discretization error. Using the perturbation results from Section 4.3 we can then derive uniform convergence rates of $\mathbf{u}_n$ to an AMLE (2.16).

In the proof we will first show the desired statement for all graph vertices and then use an approximate Lipschitz property to extend it to the whole continuum domain.

Theorem 5.13. *Assume that Assumptions 1, 2 and 4 hold. Let $\mathbf{u}_n : \Omega_n \to \mathbb{R}$ solve the graph infinity Laplacian equation (2.1). Then there exists a constant $C > 0$ such that for all $x_0 \in \Omega_{\mathcal{O}}^{2\varepsilon + 3\delta_n}$ it holds*

$$-\Delta_\infty^\varepsilon u_n^\varepsilon(x_0) \leq \mathrm{Lip}_n(\mathbf{g}_n) C \left(\frac{\delta_n}{h\varepsilon} + \frac{h}{\varepsilon^2} + \frac{\phi(h)}{h\varepsilon} \right), \tag{5.26a}$$

$$-\Delta_\infty^\varepsilon (u_n)_\varepsilon(x_0) \geq -\mathrm{Lip}_n(\mathbf{g}_n) C \left(\frac{\delta_n}{h\varepsilon} + \frac{h}{\varepsilon^2} + \frac{\phi(h)}{h\varepsilon} \right). \tag{5.26b}$$

Proof. We only proof the first inequality (5.26a). The second one can be established by applying the first one to $-\mathbf{u}_n$ and $-\mathbf{g}_n$. Our proof strategy consists in proving the desired inequality for points $\mathbf{x}_0 \in \Omega_{\mathcal{O}}^{2\varepsilon+2\delta_n} \cap \Omega_n$ and then extending this to the whole domain by the help of Lemma 5.8 which gives another δ_n contribution and leads to $\Omega_{\mathcal{O}}^{2\varepsilon+3\delta_n}$.

By definition of the nonlocal infinity Laplacian (4.8) it holds

$$-\varepsilon^2 \Delta_\infty^\varepsilon u_n^\varepsilon(\mathbf{x}_0) = 2u_n^\varepsilon(\mathbf{x}_0) - \sup_{y \in \overline{B}_\Omega(\mathbf{x}_0,\varepsilon)} u_n^\varepsilon(y) - \inf_{y \in \overline{B}_\Omega(\mathbf{x}_0,\varepsilon)} u_n^\varepsilon(y).$$

For the sup term we have

$$\sup_{y \in \overline{B}_\Omega(\mathbf{x}_0,\varepsilon)} u_n^\varepsilon(y) = \sup_{y \in \overline{B}_\Omega(\mathbf{x}_0,\varepsilon)} \sup_{\overline{B}_\Omega(y,\varepsilon)\cap\Omega_n} \mathbf{u}_n = \sup_{\overline{B}_\Omega(\mathbf{x}_0,2\varepsilon)\cap\Omega_n} \mathbf{u}_n = u_n^{2\varepsilon}(\mathbf{x}_0).$$

For the inf term, using the fact that u_n^ε is a simple function, we find $y^* \in \overline{B}_\Omega(\mathbf{x}_0,\varepsilon)$ such that

$$\inf_{y \in \overline{B}_\Omega(\mathbf{x}_0,\varepsilon)} u_n^\varepsilon(y) = u_n^\varepsilon(y^*) = \sup_{\overline{B}_\Omega(y^*,\varepsilon)\cap\Omega_n} \mathbf{u}_n \geq \mathbf{u}_n(x_0)$$

and therefore

$$-\varepsilon^2 \Delta_\infty^\varepsilon u_n^\varepsilon(\mathbf{x}_0) \leq 2u_n^\varepsilon(\mathbf{x}_0) - u_n^{2\varepsilon}(\mathbf{x}_0) - \mathbf{u}_n(\mathbf{x}_0).$$

Denoting by $\mathbf{p}_n^\varepsilon(\mathbf{x}_0) \in \overline{B}_\Omega(\mathbf{x}_0,\varepsilon) \cap \Omega_n$ and $\mathbf{p}_n^{2\varepsilon}(\mathbf{x}_0) \in \overline{B}_\Omega(\mathbf{x}_0,2\varepsilon) \cap \Omega_n$ two vertices such that

$$\begin{aligned} u_n^\varepsilon(\mathbf{x}_0) &= \sup_{\overline{B}_\Omega(\mathbf{x}_0,\varepsilon)\cap\Omega_n} \mathbf{u}_n = \mathbf{u}_n(\mathbf{p}_n^\varepsilon(\mathbf{x}_0)), \\ u_n^{2\varepsilon}(\mathbf{x}_0) &= \sup_{\overline{B}_\Omega(\mathbf{x}_0,2\varepsilon)\cap\Omega_n} \mathbf{u}_n = \mathbf{u}_n(\mathbf{p}_n^{2\varepsilon}(\mathbf{x}_0)), \end{aligned}$$

we rewrite this inequality to

$$-\varepsilon^2 \Delta_\infty^\varepsilon u_n^\varepsilon(\mathbf{x}_0) \leq 2\mathbf{u}_n(\mathbf{p}_n^\varepsilon(\mathbf{x}_0)) - \mathbf{u}_n(\mathbf{p}_n^{2\varepsilon}(\mathbf{x}_0)) - \mathbf{u}_n(\mathbf{x}_0). \tag{5.27}$$

In order to estimate this term we have to use the comparison with cones property, which $\mathbf{u}_n$ satisfies thanks to Theorem 3.2. To this end we define the following subset of vertices

$$B := \left\{ \mathbf{w} \in \Omega_n \setminus \{\mathbf{x}_0\} \,:\, \mathbf{d}_n(\mathbf{x}_0,\mathbf{w}) \leq 2\varepsilon \left(1 - \frac{\phi(h)}{h}\right) - h \right\}.$$

First note that B is non-empty. To see this, we know that by definition of the graph resolution δ_n in (2.5) there exists $\mathbf{y} \in \Omega_n \setminus \{\mathbf{x}_0\}$ such that $|\mathbf{x}_0 - \mathbf{y}| \leq 2\delta_n$. Thanks to Assumption 4 it trivially holds $2\delta_n/h \leq t_0$ and also $\phi(h)/h \leq 1/2$ and $(1+t_0)\frac{h}{\varepsilon} < 1$, which is why we obtain

$$\begin{aligned} \mathbf{d}_n(\mathbf{x}_0,\mathbf{y}) = \frac{\sigma_\eta}{\eta_h(|\mathbf{x}_0-\mathbf{y}|)} &\leq \frac{\sigma_\eta h}{\eta(2\delta_n/h)} \leq t_0 h \\ &= 2\varepsilon\left(1 - \frac{\phi(h)}{h}\right) - h + t_0 h - 2\varepsilon\left(1 - \frac{\phi(h)}{h}\right) + h \\ &= 2\varepsilon\left(1 - \frac{\phi(h)}{h}\right) - h + \varepsilon\left((1+t_0)\frac{h}{\varepsilon} - 2 + 2\frac{\phi(h)}{h}\right) \\ &\leq 2\varepsilon\left(1 - \frac{\phi(h)}{h}\right) - h, \end{aligned}$$

meaning that $\mathbf{y} \in B$. We claim that the graph boundary of B satisfies

$$\partial B \subset \left\{ \mathbf{w} \in \Omega_n \,:\, 2\varepsilon\left(1 - \frac{\phi(h)}{h}\right) - h < \mathbf{d}_n(\mathbf{x}_0,\mathbf{w}) \;\text{ and }\; d_\Omega(\mathbf{x}_0,\mathbf{w}) \leq 2\varepsilon \right\} \cup \{\mathbf{x}_0\} =: B'.$$

By definition of the graph boundary (3.4) it holds $\mathbf{w} \in \partial B$ if and only if $\mathbf{w} \notin B$ and there exists $\mathbf{y} \in B$ such that $\eta_h(|\mathbf{w}-\mathbf{y}|) > 0$. Hence, the only non-trivial property to check is the inequality $d_\Omega(\mathbf{x}_0, \mathbf{w}) \leq 2\varepsilon$. For this, we observe that $\mathbf{y} \in B$ as above satisfies $|\mathbf{w}-\mathbf{y}| \leq h$ and hence, using Lemma 5.4 and Assumption 2, we get

$$\begin{aligned} d_\Omega(\mathbf{x}_0, \mathbf{w}) &\leq d_\Omega(\mathbf{x}_0, \mathbf{y}) + d_\Omega(\mathbf{y}, \mathbf{w}) \leq \frac{\mathbf{d}_n(\mathbf{x}_0, \mathbf{y})}{1 - \frac{\phi(h)}{h}} + |\mathbf{y}-\mathbf{w}| + \phi(|\mathbf{y}-\mathbf{w}|) \\ &\leq 2\varepsilon - \frac{h}{1-\frac{\phi(h)}{h}} + h + \phi(h) = 2\varepsilon + h\left(1 + \frac{\phi(h)}{h} - \frac{1}{1-\frac{\phi(h)}{h}}\right) \leq 2\varepsilon \end{aligned}$$

using that $1 + x - \frac{1}{1-x} \leq 0$ for all $0 \leq x < 1$ and $\frac{\phi(h)}{h} \leq \frac{1}{2} < 1$ by Assumption 4. With this we have established that $\partial B \subset B'$. Next we claim that any vertex $\mathbf{w} \in B'$ satisfies the inequality

$$\mathbf{u}_n(\mathbf{w}) \leq \mathbf{u}_n(\mathbf{x}_0) + \frac{\mathbf{u}_n(\mathbf{p}_n^{2\varepsilon}(\mathbf{x}_0)) - \mathbf{u}_n(\mathbf{x}_0)}{2\varepsilon\left(1 - \frac{\phi(h)}{h}\right) - h} \mathbf{d}_n(\mathbf{x}_0, \mathbf{w}). \tag{5.28}$$

On one hand, if $\mathbf{w} = \mathbf{x}_0$ then the inequality is trivially fulfilled. On the other hand, for all other $\mathbf{w} \in B'$ by definition of $\mathbf{p}_n^{2\varepsilon}(\mathbf{x}_0)$ it holds

$$\mathbf{u}_n(\mathbf{w}) \leq \sup_{\overline{B}_\Omega(\mathbf{x}_0, 2\varepsilon) \cap \Omega_n} \mathbf{u}_n = \mathbf{u}_n(\mathbf{p}_n^{2\varepsilon}(\mathbf{x}_0)), \quad \text{and} \quad \mathbf{u}_n(\mathbf{x}_0) \leq \mathbf{u}_n(\mathbf{p}_n^{2\varepsilon}(\mathbf{x}_0)).$$

Combining these two inequalities with the definition of B' shows (5.28) for all $\mathbf{w} \in B'$ and in particular for all $\mathbf{w} \in \partial B$. Before we are able to use comparison with cones we have to argue that $B' \subset \Omega_n \setminus \mathcal{O}_n$, i.e., B' does not contain any labelled points. Let therefore $\mathbf{w} \in B'$ be arbitrary. Using the fact that $\operatorname{dist}_\Omega(\mathbf{x}_0, \mathcal{O}) \geq 2\varepsilon + 2\delta_n$ it holds

$$\operatorname{dist}_\Omega(\mathbf{w}, \mathcal{O}) \geq \operatorname{dist}_\Omega(\mathbf{x}_0, \mathcal{O}) - d_\Omega(\mathbf{x}_0, \mathbf{w}) \geq 2\varepsilon + 2\delta_n - 2\varepsilon = 2\delta_n.$$

Since the graph resolution δ_n is defined with respect to the Euclidean distance, we have to transfer this statement to the Euclidean distance.

Aiming for a contradiction we assume there exists $y \in \mathcal{O}$ with $|\mathbf{w} - y| \leq \delta_n$. By Assumption 2 it holds

$$|\mathbf{w} - y| \geq d_\Omega(\mathbf{w}, y) - \phi(|\mathbf{w}-y|) \geq 2\delta_n - |\mathbf{w}-y| \frac{\phi(|\mathbf{w}-y|)}{|\mathbf{w}-y|}.$$

We observe that thanks to Assumption 4 it holds $\delta_n \leq h$ and also

$$\frac{\phi(|\mathbf{w}-y|)}{|\mathbf{w}-y|} \leq \frac{\phi(\delta_n)}{\delta_n} \leq \sigma_\phi(h) < 1.$$

Thus, we obtain

$$|\mathbf{w} - y| \geq 2\delta_n - |\mathbf{w}-y| \frac{\phi(|\mathbf{w}-y|)}{|\mathbf{w}-y|} > 2\delta_n - \delta_n = \delta_n$$

which is a contradiction. Therefore, we get that $|\mathbf{w} - y| > \delta_n$ for all $y \in \mathcal{O}$ and, since $\mathcal{O}$ is a compact set, it holds $\operatorname{dist}(\mathbf{w}, \mathcal{O}) > \delta_n$. Next, we observe that it holds

$$\operatorname{dist}(\mathbf{w}, \mathcal{O}) = \inf_{y \in \mathcal{O}} |\mathbf{w} - y| \leq \inf_{y \in \mathcal{O}} (|\mathbf{w}-\mathbf{y}| + |y-\mathbf{y}|) = |\mathbf{w}-\mathbf{y}| + \inf_{y \in \mathcal{O}} |y-\mathbf{y}|, \quad \forall \mathbf{y} \in \mathcal{O}_n.$$

Taking the infimum over $\mathbf{y}$ we get

$$\begin{aligned} \operatorname{dist}(\mathbf{w}, \mathcal{O}) &\leq \inf_{\mathbf{y} \in \mathcal{O}_n} \left(|\mathbf{w}-\mathbf{y}| + \inf_{y \in \mathcal{O}} |y-\mathbf{y}|\right) \leq \inf_{\mathbf{y} \in \mathcal{O}_n} \left(|\mathbf{w}-\mathbf{y}| + \sup_{\mathbf{y} \in \mathcal{O}_n} \inf_{y \in \mathcal{O}} |y-\mathbf{y}|\right) \\ &\leq \operatorname{dist}(\mathbf{w}, \mathcal{O}_n) + d_H(\mathcal{O}, \mathcal{O}_n). \end{aligned}$$

Hence, can we conclude

$$\operatorname{dist}(\mathbf{w}, \mathcal{O}_n) \geq \operatorname{dist}(\mathbf{w}, \mathcal{O}) - d_H(\mathcal{O}, \mathcal{O}_n) > \delta_n - \delta_n = 0,$$

which means $B' \cap \mathcal{O}_n = \emptyset$ and therefore $B' \subset \Omega_n \setminus \mathcal{O}_n$.

Therefore, we can utilize that $\mathbf{u}_n$ satisfies comparison with cones from above (see Theorem 3.2) and infer that (5.28) holds for all vertices $\mathbf{w} \in \Omega_n$ with $\mathbf{d}_n(\mathbf{x}_0, \mathbf{w}) \leq 2\varepsilon(1 - \phi(h)/h) - h$. We would like to choose $\mathbf{w} = \mathbf{p}_n^\varepsilon(\mathbf{x}_0)$ and this choice is possible because of the following estimate. Using Lemma 5.5 and abbreviating $\tilde{C}(h) := \frac{4+2\sigma_\phi(h)}{t_0}$ it holds

$$\begin{aligned}
\mathbf{d}_n(\mathbf{x}_0, \mathbf{p}_n^\varepsilon(\mathbf{x}_0)) &\leq \left(1 + \tilde{C}(h)\frac{\delta_n}{h}\right) d_\Omega(\mathbf{x}_0, \mathbf{p}^\varepsilon(\mathbf{x}_0)) + \tau_\eta h \\
&\leq \left(1 + \tilde{C}(h)\frac{\delta_n}{h}\right)\varepsilon + \tau_\eta h \\
&= \varepsilon + \tilde{C}(h)\frac{\delta_n}{h}\varepsilon + \tau_\eta h \\
&= \underbrace{2\varepsilon\left(1 - \frac{\phi(h)}{h}\right) - h - \left(2\varepsilon - 2\varepsilon\frac{\phi(h)}{h} - h\right)}_{=0} \\
&\qquad + \varepsilon + \tilde{C}(h)\frac{\delta_n}{h}\varepsilon + \tau_\eta h \\
&= 2\varepsilon\left(1 - \frac{\phi(h)}{h}\right) - h + 2\varepsilon\frac{\phi(h)}{h} - \varepsilon + (1 + \tau_\eta)h + \tilde{C}(h)\frac{\delta_n}{h}\varepsilon \\
&= 2\varepsilon\left(1 - \frac{\phi(h)}{h}\right) - h + \varepsilon\left(2\frac{\phi(h)}{h} - 1 + (1 + \tau_\eta)\frac{h}{\varepsilon} + \tilde{C}(h)\frac{\delta_n}{h}\right) \\
&\leq 2\varepsilon\left(1 - \frac{\phi(h)}{h}\right) - h, \qquad (5.29)
\end{aligned}$$

where we used $t_0 \leq 1$, $2 \leq \frac{4+2\sigma_\phi(h)}{t_0} = \tilde{C}(h)$, $\tau_\eta \leq 1$, and $\frac{\phi(h)}{h} \leq \sigma_\phi(h)$ to obtain from Assumption 4

$$2\frac{\phi(h)}{h} - 1 + (1 + \tau_\eta)\frac{h}{\varepsilon} + \tilde{C}(h)\frac{\delta_n}{h} \leq \tilde{C}(h)\left(\sigma_\phi(h) + \frac{\delta_n}{h}\right) - 1 + 2\frac{h}{\varepsilon} \leq 0.$$

Before starting with the central estimate we use Eq. (5.29), Proposition 3.8, Lemma 5.5, and Assumption 4 to compute

$$\begin{aligned}
\left|\mathbf{u}_n(\mathbf{p}_n^{2\varepsilon}(\mathbf{x}_0)) - \mathbf{u}_n(\mathbf{x}_0)\right| &\leq \operatorname{Lip}_n(\mathbf{g}_n)\mathbf{d}_n(\mathbf{p}_n^{2\varepsilon}(\mathbf{x}_0), \mathbf{x}_0)) \\
&\leq \operatorname{Lip}_n(\mathbf{g}_n)\left(\tilde{C}(h)\, d_\Omega(\mathbf{p}_n^{2\varepsilon}(\mathbf{x}_0), \mathbf{x}_0) + \tau_\eta h\right) \\
&\leq \operatorname{Lip}_n(\mathbf{g}_n)\left(2\tilde{C}(h)\varepsilon + \tau_\eta h\right) \\
&\leq \operatorname{Lip}_n(\mathbf{g}_n)\overline{C}\varepsilon \qquad (5.30)
\end{aligned}$$

for a constant $\overline{C} > 0$. Using Eq. (5.30), Lemma 5.5, and Assumption 4 we can compute

$$
\begin{aligned}
2\mathbf{u}_n(\mathbf{p}_n^\varepsilon(\mathbf{x}_0)) &\leq 2\mathbf{u}_n(\mathbf{x}_0) + \frac{\mathbf{u}_n(\mathbf{p}_n^{2\varepsilon}(\mathbf{x}_0)) - \mathbf{u}_n(\mathbf{x}_0)}{\varepsilon\left(1 - \frac{\phi(h)}{h}\right) - h/2}\mathbf{d}_n(\mathbf{x}_0, \mathbf{p}_n^\varepsilon(\mathbf{x}_0)) \\
&\leq 2\mathbf{u}_n(\mathbf{x}_0) + \frac{\mathbf{u}_n(\mathbf{p}_n^{2\varepsilon}(\mathbf{x}_0)) - \mathbf{u}_n(\mathbf{x}_0)}{\varepsilon\left(1 - \frac{\phi(h)}{h}\right) - h/2}\left(\left(1 + \tilde{C}(h)\frac{\delta_n}{h}\right)\varepsilon + \tau_\eta h\right) \\
&= 2\mathbf{u}_n(\mathbf{x}_0) + \left(\mathbf{u}_n(\mathbf{p}_n^{2\varepsilon}(\mathbf{x}_0)) - \mathbf{u}_n(\mathbf{x}_0)\right)\left(\left(1 + \tilde{C}(h)\frac{\delta_n}{h}\right)\frac{\varepsilon}{\varepsilon\left(1 - \frac{\phi(h)}{h}\right) - h/2} + \frac{\tau_\eta h}{\varepsilon\left(1 - \frac{\phi(h)}{h}\right) - h/2}\right) \\
&= \mathbf{u}_n(\mathbf{x}_0) + \mathbf{u}_n(\mathbf{p}_n^{2\varepsilon}(\mathbf{x}_0)) + \\
&\qquad \left(\mathbf{u}_n(\mathbf{p}_n^{2\varepsilon}(\mathbf{x}_0)) - \mathbf{u}_n(\mathbf{x}_0)\right)\left(\tilde{C}(h)\frac{\delta_n}{h} + \frac{\varepsilon\frac{\phi(h)}{h} + h/2}{\varepsilon(1 - \frac{\phi(h)}{h}) - h/2}\left(1 + \tilde{C}(h)\frac{\delta_n}{h}\right) + \frac{\tau_\eta h}{\varepsilon(1 - \frac{\phi(h)}{h}) - h/2}\right) \\
&\leq \mathbf{u}_n(\mathbf{x}_0) + \mathbf{u}_n(\mathbf{p}_n^{2\varepsilon}(\mathbf{x}_0)) + \mathrm{Lip}_n(\mathbf{g}_n)\overline{C}\varepsilon\left(\tilde{C}(h)\frac{\delta_n}{h} + \left(\frac{\phi(h)}{h} + \frac{h}{2\varepsilon}\right)\left(1 + 2\frac{\phi(h)}{h} + \frac{h}{\varepsilon}\right)\left(1 + \tilde{C}(h)\frac{\delta_n}{h}\right)\right. \\
&\qquad \left. + \tau_\eta\frac{h}{\varepsilon}\left(1 + 2\frac{\phi(h)}{h} + \frac{h}{\varepsilon}\right)\right) \\
&\leq \mathbf{u}_n(\mathbf{x}_0) + \mathbf{u}_n(\mathbf{p}_n^{2\varepsilon}(\mathbf{x}_0)) + \mathrm{Lip}_n(\mathbf{g}_n)C\varepsilon\left(\frac{\delta_n}{h} + \frac{h}{\varepsilon} + \frac{\phi(h)}{h}\right) \\
&= \mathbf{u}_n(\mathbf{x}_0) + \mathbf{u}_n(\mathbf{p}_n^{2\varepsilon}(\mathbf{x}_0)) + \mathrm{Lip}_n(\mathbf{g}_n)C\left(\frac{\delta_n\varepsilon}{h} + h + \varepsilon\frac{\phi(h)}{h}\right),
\end{aligned}
$$

for a constant $C > 0$. Plugging this into (5.28) with $\mathbf{w} = \mathbf{p}^\varepsilon(\mathbf{x}_0)$ and and dividing by ε^2 we get

$$
-\Delta_\infty^\varepsilon u_n^\varepsilon(\mathbf{x}_0) \leq \mathrm{Lip}_n(\mathbf{g}_n)C\left(\frac{\delta_n}{h\varepsilon} + \frac{h}{\varepsilon^2} + \frac{\phi(h)}{\varepsilon h}\right).
$$

□

Remark 5.14. Almost the whole proof of Theorem 5.13 works under the assumption that $\mathbf{u}_n$ merely satisfies comparison with cones from above / below in which case one could only prove (5.26a) / (5.26b). However, in order to have the discrete Lipschitz estimate from Proposition 3.8, we need $\mathbf{u}_n$ to satisfy both comparison properties.

Remark 5.15. We were hoping that in the case of a singular kernel, where $\tau_\eta = 0$ and hence the convergence rate of graph distance functions is better according to Lemma 5.5, the consistency error in (5.26) might be better. Closely following the proof, one observes that almost all order h contributions are multiplied with τ_η. Only in the definition of the graph set B, we have to use the lower bound which features $-h$. This then leads to the term $h\varepsilon^{-2}$ which, however, does not contribute for small graph length scales h.

5.3.3 Convergence Rates

Now we are in the position to prove an important result on our way to convergence rates which can be interpreted as an approximate maximum principle.

Proposition 5.16. *Let $u \in C(\overline{\Omega})$ satisfy CDF (cf. (4.1)) and define $\tilde{\varepsilon}_n := \varepsilon + \frac{3}{2}\delta_n$. Under the conditions*

of Theorem 5.13 there exists a constant $C > 0$ such that

$$\sup_{\Omega_{\mathcal{O}}^{\tilde{\varepsilon}_n}}(u_n^\varepsilon - T_\varepsilon u) \le \sup_{\Omega_{\mathcal{O}}^{\tilde{\varepsilon}_n}\setminus\Omega_{\mathcal{O}}^{2\tilde{\varepsilon}_n}}(u_n^\varepsilon - T_\varepsilon u) + \sqrt[3]{\mathrm{Lip}_n(\mathbf{g}_n)C\left(\frac{\delta_n}{h\varepsilon} + \frac{h}{\varepsilon^2} + \frac{\phi(h)}{h\varepsilon}\right)}, \tag{5.31a}$$

$$\sup_{\Omega_{\mathcal{O}}^{\tilde{\varepsilon}_n}}(T^\varepsilon u - (u_n)_\varepsilon) \le \sup_{\Omega_{\mathcal{O}}^{\tilde{\varepsilon}_n}\setminus\Omega_{\mathcal{O}}^{2\tilde{\varepsilon}_n}}(T^\varepsilon u - (u_n)_\varepsilon) + \sqrt[3]{\mathrm{Lip}_n(\mathbf{g}_n)C\left(\frac{\delta_n}{h\varepsilon} + \frac{h}{\varepsilon^2} + \frac{\phi(h)}{h\varepsilon}\right)}. \tag{5.31b}$$

If additionally it holds that $\alpha := \inf_{\overline{\Omega}} |\nabla u| > 0$, then we even have

$$\sup_{\Omega_{\mathcal{O}}^{\tilde{\varepsilon}_n}}(u_n^\varepsilon - T_\varepsilon u) \le \sup_{\Omega_{\mathcal{O}}^{\tilde{\varepsilon}_n}\setminus\Omega_{\mathcal{O}}^{2\tilde{\varepsilon}_n}}(u_n^\varepsilon - T_\varepsilon u) + \mathrm{Lip}_n(\mathbf{g}_n)C\left(\frac{\delta_n}{h\varepsilon} + \frac{h}{\varepsilon^2} + \frac{\phi(h)}{h\varepsilon}\right), \tag{5.32a}$$

$$\sup_{\Omega_{\mathcal{O}}^{\tilde{\varepsilon}_n}}(T^\varepsilon u - (u_n)_\varepsilon) \le \sup_{\Omega_{\mathcal{O}}^{\tilde{\varepsilon}_n}\setminus\Omega_{\mathcal{O}}^{2\tilde{\varepsilon}_n}}(T^\varepsilon u - (u_n)_\varepsilon) + \mathrm{Lip}_n(\mathbf{g}_n)C\left(\frac{\delta_n}{h\varepsilon} + \frac{h}{\varepsilon^2} + \frac{\phi(h)}{h\varepsilon}\right). \tag{5.32b}$$

Proof. We only prove (5.31a), the second inequality is obtained analogously. By Theorem 5.13 we know that there is a constant $C > 0$ such that

$$-\Delta_\infty^\varepsilon u_n^\varepsilon \le \mathrm{Lip}_n(\mathbf{g}_n)C\left|\frac{\delta_n}{h\varepsilon} + \frac{h}{\varepsilon^2} + \frac{\phi(h)}{h\varepsilon}\right| =: C_{n,\varepsilon} \quad \text{in } \Omega_{\mathcal{O}}^{2\tilde{\varepsilon}_n}.$$

By Lemma 4.6 we know that $-\Delta_\infty^\varepsilon T_\varepsilon u \ge 0$ in $\Omega_{\mathcal{O}}^{2\varepsilon}$ and hence also in $\Omega_{\mathcal{O}}^{2\tilde{\varepsilon}_n}$. Furthermore, since u is infinity harmonic, it is bounded and hence also $T_\varepsilon u$ is bounded. Applying Lemma 4.8 with $\delta := (C_{n,\varepsilon})^{\frac{1}{3}}$ we can hence choose $v : \Omega_{\mathcal{O}}^\varepsilon \to \mathbb{R}$ such that

$$-\Delta_\infty^\varepsilon v \ge 0, \quad S_\varepsilon^- v \ge (C_{n,\varepsilon})^{\frac{1}{3}}, \quad T_\varepsilon u \le v \le T_\varepsilon u + 2(C_{n,\varepsilon})^{\frac{1}{3}}\operatorname{dist}(\cdot, \Omega_{\mathcal{O}}^\varepsilon \setminus \Omega_{\mathcal{O}}^{2\varepsilon}) \quad \text{in } \Omega_{\mathcal{O}}^{2\varepsilon}.$$

Since $u \in C(\overline{\Omega})$ is a bounded function, the same holds for v. Define $w := v - (C_{n,\varepsilon})^{\frac{1}{3}}v^2$ which according to Lemma 4.9 and its definition satisfies

$$-\Delta_\infty^\varepsilon w \ge C_{n,\varepsilon} \quad \text{in } \Omega_{\mathcal{O}}^{2\varepsilon}, \quad \|w - T_\varepsilon u\|_{L^\infty(\Omega_{\mathcal{O}}^\varepsilon)} \le c(C_{n,\varepsilon})^{\frac{1}{3}}.$$

Then it holds $-\Delta_\infty^\varepsilon u_n^\varepsilon \le C_{n,\varepsilon} \le -\Delta_\infty^\varepsilon w$ in $\Omega_{\mathcal{O}}^{2\tilde{\varepsilon}_n}$ and applying Lemma 4.7 with ε replaced by $\tilde{\varepsilon}_n$ we can compute

$$\begin{aligned}\sup_{\Omega_{\mathcal{O}}^{\tilde{\varepsilon}_n}}(u_n^\varepsilon - T_\varepsilon u) &\le \sup_{\Omega_{\mathcal{O}}^{\tilde{\varepsilon}_n}}(u_n^\varepsilon - w) + c(C_{n,\varepsilon})^{\frac{1}{3}} \\ &\le \sup_{\Omega_{\mathcal{O}}^{\tilde{\varepsilon}_n}\setminus\Omega_{\mathcal{O}}^{2\tilde{\varepsilon}_n}}(u_n^\varepsilon - w) + c(C_{n,\varepsilon})^{\frac{1}{3}} \\ &\le \sup_{\Omega_{\mathcal{O}}^{\tilde{\varepsilon}_n}\setminus\Omega_{\mathcal{O}}^{2\tilde{\varepsilon}_n}}(u_n^\varepsilon - T_\varepsilon u) + 2c(C_{n,\varepsilon})^{\frac{1}{3}}.\end{aligned}$$

In the case that $\alpha = \inf_{\overline{\Omega}} |\nabla u| > 0$ we do not have to apply Lemma 4.8 to construct a perturbation v with positive slope. Instead, we define $w := u - \frac{C_{n,\varepsilon}}{\alpha^2}u^2$ which according to Lemma 4.9 satisfies

$$-\Delta_\infty^\varepsilon w \ge C_{n,\varepsilon} \quad \text{in } \Omega_{\mathcal{O}}^{2\varepsilon}, \quad \|w - T_\varepsilon u\|_{L^\infty(\Omega_{\mathcal{O}}^\varepsilon)} \le \frac{c}{\alpha^2}C_{n,\varepsilon}.$$

From here the proof continues as before. □

For getting rid of the ε-extension operators and boundary terms in Proposition 5.16, we have to use the lemmas from Section 5.3.1. Basically, one uses Lipschitzness of u (and u_n^ε in an appropriate sense) in order to transfer the statement of Proposition 5.16 to u and $\mathbf{u}_n$, estimating the boundary term, and extending the uniform estimate to the whole of $\overline{\Omega}$. All this comes with an additional additive error term of order $\varepsilon > 0$. We are now ready to prove the main theorem of this article, Theorem 2.2.

Proof of Theorem 2.2. First we notice that thanks to [37, Lemma 2.3] we have

$$\sup_{\Omega_n} |u - \mathbf{u}_n| = \sup_{\overline{\Omega}} |u - u_n| .$$

Again we abbreviate $\tilde{\varepsilon}_n := \varepsilon + \frac{3}{2}\delta_n$. Using Lemmas 5.9 to 5.11 and Proposition 5.16 we get

$$\begin{aligned}
\sup_{\Omega_{\mathcal{O}}^{\tilde{\varepsilon}_n}}(u_n - u) &\leq \sup_{\Omega_{\mathcal{O}}^{\tilde{\varepsilon}_n}}(u_n - u_n^\varepsilon) + \sup_{\Omega_{\mathcal{O}}^{\tilde{\varepsilon}_n}}(u_n^\varepsilon - T_\varepsilon u) + \sup_{\Omega_{\mathcal{O}}^{\tilde{\varepsilon}_n}}(T_\varepsilon u - u) \\
&\leq C \operatorname{Lip}_n(\mathbf{u}_n)\varepsilon + C \operatorname{Lip}_\Omega(u)\varepsilon + \sup_{\Omega_{\mathcal{O}}^{\tilde{\varepsilon}_n} \setminus \Omega_{\mathcal{O}}^{2\tilde{\varepsilon}_n}} (u_n^\varepsilon - T_\varepsilon u) + \sqrt[3]{\operatorname{Lip}_n(\mathbf{g}_n) C \left(\frac{\delta_n}{h\varepsilon} + \frac{h}{\varepsilon^2} + \frac{\phi(h)}{h\varepsilon} \right)} \\
&\leq C \operatorname{Lip}_n(\mathbf{u}_n)\varepsilon + C \operatorname{Lip}_\Omega(u)\varepsilon + C\varepsilon + \sqrt[3]{\operatorname{Lip}_n(\mathbf{g}_n) C \left(\frac{\delta_n}{h\varepsilon} + \frac{h}{\varepsilon^2} + \frac{\phi(h)}{h\varepsilon} \right)}.
\end{aligned}$$

By Proposition 3.8 we know that $\operatorname{Lip}_n(\mathbf{u}_n) = \operatorname{Lip}_n(\mathbf{g}_n)$ and by Assumption 3 it holds $\sup_{n\in\mathbb{N}} \operatorname{Lip}_n(\mathbf{g}_n) \leq C$. The term $\sup_{\Omega_{\mathcal{O}}^{\tilde{\varepsilon}_n}}(u - u_n)$ can be estimated analogously. Hence, we get the inequality

$$\sup_{\Omega_{\mathcal{O}}^{\tilde{\varepsilon}_n}} |u - u_n| \leq C \left(\varepsilon + \sqrt[3]{\frac{\delta_n}{h\varepsilon} + \frac{h}{\varepsilon^2} + \frac{\phi(h)}{h\varepsilon}} \right).$$

Note that it holds $|u(x) - u(y)| \leq \operatorname{Lip}_\Omega(u) d_\Omega(x, y)$ for all $x, y \in \overline{\Omega}$. Using this together with Lemma 5.12 and Assumption 4 we get for all $x \in \overline{\Omega} \setminus \Omega_{\mathcal{O}}^{\tilde{\varepsilon}_n}$ that

$$\begin{aligned}
|u(x) - u_n(x)| &\leq |u(x) - u(\tilde{x})| + |u(\tilde{x}) - u_n(\tilde{x})| + |u_n(\tilde{x}) - u_n(x)| \\
&\leq \operatorname{Lip}_\Omega(u)\tilde{\varepsilon}_n + C \left(\varepsilon + \sqrt[3]{\frac{\delta_n}{h\varepsilon} + \frac{h}{\varepsilon^2} + \frac{\phi(h)}{h\varepsilon}} \right) + C \operatorname{Lip}_n(\mathbf{u}_n)\tilde{\varepsilon}_n,
\end{aligned}$$

where $\tilde{x} \in \Omega_{\mathcal{O}}^{\tilde{\varepsilon}}$ is such that $d_\Omega(x, \tilde{x}) \leq \tilde{\varepsilon}_n$. Using also that thanks to Assumption 4 it holds $\tilde{\varepsilon}_n = \varepsilon(1 + \frac{3}{2}\delta_n) \leq C\varepsilon$, we can pass from $\sup_{\Omega_{\mathcal{O}}^{\tilde{\varepsilon}}} |u - u_n|$ to $\sup_{\overline{\Omega}} |u - u_n|$ at the cost of another term of order ε which proves the first assertion of the theorem. In the case that $\inf_{\overline{\Omega}} |\nabla u| > 0$ one simply uses the better estimate in Proposition 5.16 without $\sqrt[3]{\cdot}$. □

6 Numerical Results

In this section we present a brief numerical study of convergence rates for the continuum limit of Lipschitz learning.[2] In order to test convergence rates, we work with the Aronsson function

$$u(x_1, x_2) = |x_1|^{4/3} - |x_2|^{4/3},$$

which is infinity harmonic on the plane $\mathbb{R}^2$. We work with the non-convex and non-smooth domain (5.6) which satisfies Assumption 2 according to Proposition 5.3 and on which the Aronsson function is an AMLE (2.16). For convenience, we remind the reader that this domain was defined as follows

$$\Omega := \left\{ x \in [0,1]^2 \,:\, |x_1|^{2/3} + |x_2|^{2/3} \leq 1 \right\},$$

and coincides with the non-convex unit ball around zero of the $\ell_{2/3}$ metric space.

As constraint set we choose the four points $\mathcal{O} = \{(\pm 1, 0)\} \cup \{(0, \pm 1)\}$. This domain was chosen so that the Aronsson function u satisfies the homogeneous Neumann condition $\partial u / \partial \nu = 0$ on $\partial\Omega \setminus \mathcal{O}$. The point

[2] The code for our experiments is available online: https://github.com/jwcalder/LipschitzLearningRates.

cloud Ω_n is generated by drawing *i.i.d.* samples from Ω. For the discrete constraint set we consider two different choices. The first one consists of the four closest graph points to $\mathcal{O}$,

$$\mathcal{O}_n^{\mathrm{cp}} = \bigcup_{x \in \mathcal{O}} \operatorname*{arg\,min}_{\mathbf{x} \in \Omega_n} |\mathbf{x} - x| \,,$$

and the second one is given using a dilated boundary:

$$\mathcal{O}_n^{\mathrm{dil}} = \{\mathbf{x} \in \Omega_n \,:\, \mathrm{dist}(\mathbf{x}, \mathcal{O}) \leq h_n\}.$$

Fig. 2 shows the solutions of the Lipschitz learning problem for different numbers $n \in \mathbb{N}$ of vertices, using the constraint set $\mathcal{O}_n^{\mathrm{cp}}$. Furthermore, in Fig. 3 we show a convergence study where we compute empirical rates[3] of convergence for the different bandwidths and kernels. We consider two different kernels: the constant one with $\eta(t) = 1$ and the singular one with $\eta(t) = t^{-1}$. Furthermore, we consider three different scalings of the graph bandwidth h_n which we choose as $h_n \sim \{\delta_n, \delta_n^{2/3}, \delta_n^{1/2}\}$. For the first scaling our theory does not even prove convergence. This is due to the fact that the angular error in the point cloud should go to zero which manifests in $h_n \gg \delta_n$. Still, for a randomly sampled point cloud one can hope for a homogenization effect and still obtain convergence. The other two scalings are covered by our theory, whereby $h_n \sim \delta_n^{2/3}$ falls into the small length scale regime (cf. Corollary 2.4) and $h_n \sim \delta_n^{1/2}$ falls into the large one with less smooth boundary (cf. Corollary 2.6).

For the constant kernel $\eta(t) = 1$ we ran experiments with $n = 2^{12}$ up to $n = 2^{16}$ points and averaged the errors over 100 different random samples of Ω_n. For the singular kernel $\eta(t) = t^{-1}$, the numerical solver is much slower and we ran experiments only up to $n = 2^{15}$ and averaged errors over 20 trials. We observe in Fig. 3 that all scalings appear to be convergent. Interestingly, the smallest bandwidth $h_n \sim \delta_n$, where convergence cannot be expected in general, exhibits the best rates and smallest errors. Notably all empirical rates are better than the ones from Corollaries 2.4 and 2.6. This does not necessarily mean that our results are not sharp since for the Aronsson solution one can typically prove better rates, see the discussion in [39]. At the same time, it constitutes the only non-trivial (i.e., not a cone) explicit solution of the AMLE problem which is why it is a popular test case. We also observe in Fig. 3 that the rates for the singular kernel $\eta(t) = t^{-1}$ are better than the ones for unit weights which suggests that there might be a way to lift the better approximation of the distance functions (cf. Lemma 5.5) to a better approximation of the (non-local) infinity Laplacian (cf. Remark 5.15). Also the magnitudes of the errors are much smaller for the singular kernel than for the constant one.

In Figs. 4 and 5 we show the results using the dilated constraint set $\mathcal{O}_n^{\mathrm{dil}}$. For the singular kernel we see that the smallest graph length scales again yield the best rates, while for the constant kernel $\eta(t) = 1$, all length scales exhibit similar convergence rates. We can partially attribute this to the fact that the constraint set $\mathcal{O}_n^{\mathrm{cp}}$ leads to sharp cusps which is resolved better using the smallest bandwidth whereas the dilated boundary in $\mathcal{O}_n^{\mathrm{dil}}$ generates smoother domains where the larger length scales are well-suited for approximating the infinity Laplacian.

Acknowledgments

The authors thank the Institute for Mathematics and its Applications (IMA) where this collaboration started at a workshop on "Theory and Applications in Graph-Based Learning" in Fall 2020. Part of this work was also done while the authors were visiting the Simons Institute for the Theory of Computing to participate in the program "Geometric Methods in Optimization and Sampling" during the Fall of 2021. LB acknowledges funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - GZ 2047/1, Projekt-ID 390685813. JC acknowledges funding from NSF grant DMS:1944925, the Alfred P. Sloan foundation, and a McKnight Presidential Fellowship. TR acknowledges support by the German Ministry of Science and Technology (BMBF) under grant agreement No. 05M2020 (DELETO).

[3] We computed the rates by fitting a linear function to the log-log plots in Fig. 3.

References

[1] S. N. Armstrong and C. K. Smart. "An easy proof of Jensen's theorem on the uniqueness of infinity harmonic functions". In: *Calculus of Variations and Partial Differential Equations* 37.3 (2010), pp. 381–384 (cit. on pp. 5, 19, 21).

[2] S. Armstrong, C. Smart, and S. Somersille. "An infinity Laplace equation with gradient term and mixed boundary conditions". In: *Proceedings of the American Mathematical Society* 139.5 (2011), pp. 1763–1776 (cit. on p. 18).

[3] G. Aronsson, M. Crandall, and P. Juutinen. "A tour of the theory of absolutely minimizing functions". In: *Bulletin of the American mathematical society* 41.4 (2004), pp. 439–505 (cit. on pp. 3, 5, 18).

[4] J. Calder. "The game theoretic p-Laplacian and semi-supervised learning with few labels". In: *Nonlinearity* 32.1 (2018), pp. 301–330 (cit. on pp. 2, 14).

[5] J. Calder. "Consistency of Lipschitz learning with infinite unlabeled data and finite labeled data". In: *SIAM Journal on Mathematics of Data Science* 1 (2019), pp. 780–812 (cit. on pp. 2, 3, 5, 7, 9, 10, 12–14).

[6] J. Calder, B. Cook, M. Thorpe, and D. Slepcev. "Poisson Learning: Graph Based Semi-Supervised Learning At Very Low Label Rates". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by H. D. III and A. Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 13–18 Jul 2020, pp. 1306–1316 (cit. on p. 2).

[7] J. Calder and N. García Trillos. "Improved spectral convergence rates for graph Laplacians on ε-graphs and k-NN graphs". In: *Applied and Computational Harmonic Analysis* 60 (2022), pp. 123–175 (cit. on p. 2).

[8] J. Calder, N. García Trillos, and M. Lewicka. "Lipschitz regularity of graph Laplacians on random data clouds". In: *SIAM Journal on Mathematical Analysis* 54.1 (2022), pp. 1169–1222 (cit. on p. 2).

[9] J. Calder and D. Slepčev. "Properly-weighted graph Laplacian for semi-supervised learning". In: *Applied mathematics & optimization* 82.3 (2020), pp. 1111–1159 (cit. on p. 2).

[10] J. Calder, D. Slepčev, and M. Thorpe. *Rates of Convergence for Laplacian Semi-Supervised Learning with Low Labeling Rates*. 2020. arXiv: 2006.02765 [math.ST] (cit. on pp. 2, 3).

[11] A. Chambolle, E. Lindgren, and R. Monneau. "A Hölder infinity Laplacian". In: *ESAIM: Control, Optimisation and Calculus of Variations* 18.3 (2012), pp. 799–835 (cit. on p. 4).

[12] T. Champion and L. De Pascale. "Principles of comparison with distance functions for absolute minimizers". In: *Journal of Convex Analysis* 14.3 (2007), p. 515 (cit. on pp. 5, 17).

[13] M. G. Crandall, H. Ishii, and P.-L. Lions. "User's guide to viscosity solutions of second order partial differential equations". In: *Bulletin of the American mathematical society* 27.1 (1992), pp. 1–67 (cit. on p. 18).

[14] 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: *29th Annual Conference on Learning Theory*. Ed. by V. Feldman, A. Rakhlin, and O. Shamir. Vol. 49. Proceedings of Machine Learning Research. Columbia University, New York, New York, USA: PMLR, 23–26 Jun 2016, pp. 879–906 (cit. on pp. 2, 3).

[15] A. Elmoataz, M. Toutain, and D. Tenbrinck. "On the p-Laplacian and ∞-Laplacian on graphs with applications in image and data processing". In: *SIAM Journal on Imaging Sciences* 8.4 (2015), pp. 2412–2451 (cit. on pp. 2, 13).

[16] L. Esposito, B. Kawohl, C. Nitsch, and C. Trombetti. "The Neumann eigenvalue problem for the ∞-Laplacian". In: *Rendiconti Lincei-Matematica e Applicazioni* 26.2 (2015), pp. 119–134 (cit. on p. 18).

[17] M. Flores, J. Calder, and G. Lerman. "Analysis and algorithms for ℓ_p-based semi-supervised learning on graphs". In: *Applied and Computational Harmonic Analysis* 60 (2022), pp. 77–122 (cit. on pp. 2, 3, 5).

[18] N. García Trillos, M. Gerlach, M. Hein, and D. Slepčev. "Error estimates for spectral convergence of the graph Laplacian on random geometric graphs toward the Laplace–Beltrami operator". In: *Foundations of Computational Mathematics* 20.4 (2020), pp. 827–887 (cit. on p. 2).

[19] N. García Trillos and R. W. Murray. "A maximum principle argument for the uniform convergence of graph Laplacian regressors". In: *SIAM Journal on Mathematics of Data Science* 2.3 (2020), pp. 705–739 (cit. on p. 2).

[20] N. García Trillos and D. Slepčev. "Continuum Limit of Total Variation on Point Clouds". In: *Archive for Rational Mechanics and Analysis* 220.1 (2015), pp. 193–241. ISSN: 1432-0673 (cit. on p. 2).

[21] I. Goodfellow, Y. Bengio, and A. Courville. "Machine learning basics". In: *Deep learning* 1.7 (2016), pp. 98–164 (cit. on p. 12).

[22] P. Juutinen. "Absolutely minimizing Lipschitz extensions on a metric space". In: *Annales Academiae Scientiarum Fennicae Mathematica*. Vol. 27. 1. Academia Scientiarum Fennica. 2002, pp. 57–68 (cit. on pp. 3, 21).

[23] P. Juutinen and N. Shanmugalingam. "Equivalence of AMLE, strong AMLE, and comparison with cones in metric measure spaces". In: *Mathematische Nachrichten* 279.9-10 (2006), pp. 1083–1098 (cit. on pp. 16–18, 21).

[24] R. Kyng, A. Rao, S. Sachdeva, and D. A. Spielman. "Algorithms for Lipschitz Learning on Graphs". In: *Proceedings of The 28th Conference on Learning Theory*. Ed. by P. Grünwald, E. Hazan, and S. Kale. Vol. 40. Proceedings of Machine Learning Research. Paris, France: PMLR, Mar. 2015, pp. 1190–1223 (cit. on pp. 3, 13).

[25] E. Le Gruyer. "On absolutely minimizing Lipschitz extensions and PDE $\Delta_\infty(u) = 0$". In: *Nonlinear Differential Equations and Applications NoDEA* 14.1 (2007), pp. 29–55 (cit. on pp. 4, 10).

[26] M. Lewicka and J. J. Manfredi. "The obstacle problem for the p-Laplacian via optimal stopping of tug-of-war games". In: *Probability Theory and Related Fields* 167.1-2 (2017), pp. 349–378 (cit. on p. 5).

[27] U. V. Luxburg and O. Bousquet. "Distance-Based Classification with Lipschitz Functions". In: *J. Mach. Learn. Res.* 5 (2004), pp. 669–695 (cit. on p. 3).

[28] U. von Luxburg, M. Belkin, and O. Bousquet. "Consistency of spectral clustering". In: *The Annals of Statistics* 36.2 (Apr. 2008), pp. 555–586 (cit. on p. 2).

[29] J. J. Manfredi, M. Parviainen, and J. D. Rossi. "On the definition and properties of p-harmonious functions". In: *Annali della Scuola Normale Superiore di Pisa-Classe di Scienze* 11.2 (2012), pp. 215–241 (cit. on p. 5).

[30] J. Mazón, J. D. Rossi, and J. Toledo. "On the best Lipschitz extension problem for a discrete distance and the discrete ∞-Laplacian". In: *Journal de mathématiques pures et appliquées* 97.2 (2012), pp. 98–119 (cit. on pp. 5, 19).

[31] V. A. Milman. "Absolutely minimal extensions of functions on metric spaces". In: *Sbornik: Mathematics* 190.6 (1999), p. 859 (cit. on p. 21).

[32] K. P. Murphy. *Machine learning: a probabilistic perspective*. Cambridge, Massachusetts: MIT press, 2012 (cit. on p. 2).

[33] B. Nadler, N. Srebro, and X. Zhou. "Semi-supervised learning with the graph Laplacian: The limit of infinite unlabelled data". In: *Advances in neural information processing systems* 22 (2009), pp. 1330–1338 (cit. on p. 2).

[34] A. M. Oberman. "A convergent difference scheme for the infinity Laplacian: construction of absolutely minimizing Lipschitz extensions". In: *Mathematics of Computation* 74.251 (Sept. 2004), pp. 1217–1231 (cit. on p. 3).

[35] Y. Peres, O. Schramm, S. Sheffield, and D. Wilson. "Tug-of-war and the infinity Laplacian". In: *Journal of the American Mathematical Society* 22.1 (2009), pp. 167–210 (cit. on p. 5).

[36] Y. Peres and S. Sheffield. "Tug-of-war with noise: a game-theoretic view of the p-Laplacian". In: *Duke Mathematical Journal* 145.1 (2008), pp. 91–120 (cit. on p. 5).

[37] T. Roith and L. Bungert. "Continuum limit of Lipschitz learning on graphs". In: *Foundations of Computational Mathematics* (2022), pp. 1–39 (cit. on pp. 2, 3, 9, 10, 13, 23, 37).

[38] D. Slepčev and M. Thorpe. "Analysis of p-Laplacian Regularization in Semisupervised Learning". In: *SIAM Journal on Mathematical Analysis* 51.3 (2019), pp. 2085–2120 (cit. on pp. 2, 3).

[39] C. K. Smart. "On the infinity Laplacian and Hrushovski's fusion". PhD thesis. UC Berkeley, 2010 (cit. on pp. 4–6, 11, 12, 19–21, 38).

[40] J. E. Van Engelen and H. H. Hoos. "A survey on semi-supervised learning". In: *Machine Learning* 109.2 (2020), pp. 373–440 (cit. on p. 2).

[41] Y. Van Gennip, A. L. Bertozzi, et al. "Γ-convergence of graph Ginzburg-Landau functionals". In: *Advances in Differential Equations* 17.11/12 (2012), pp. 1115–1180 (cit. on p. 2).

[42] U. Von Luxburg. "A tutorial on spectral clustering". In: *Statistics and computing* 17.4 (2007), pp. 395–416 (cit. on p. 4).

[43] G. Voronoï. "Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Deuxième mémoire. Recherches sur les parallélloèdres primitifs." In: *Journal für die reine und angewandte Mathematik (Crelles Journal)* 1908.134 (1908), pp. 198–287 (cit. on p. 7).

[44] G. Voronoï. "Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Premier mémoire. Sur quelques propriétés des formes quadratiques positives parfaites." In: *Journal für die reine und angewandte Mathematik (Crelles Journal)* 1908.133 (1908), pp. 97–102 (cit. on p. 7).

[45] A. Yuan, J. Calder, and B. Osting. "A continuum limit for the PageRank algorithm". In: *European Journal of Applied Mathematics* 33.3 (2022), pp. 472–504 (cit. on p. 2).

[46] X. Zhu, Z. Ghahramani, and J. Lafferty. "Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions". In: ICML'03 (2003), pp. 912–919 (cit. on p. 2).

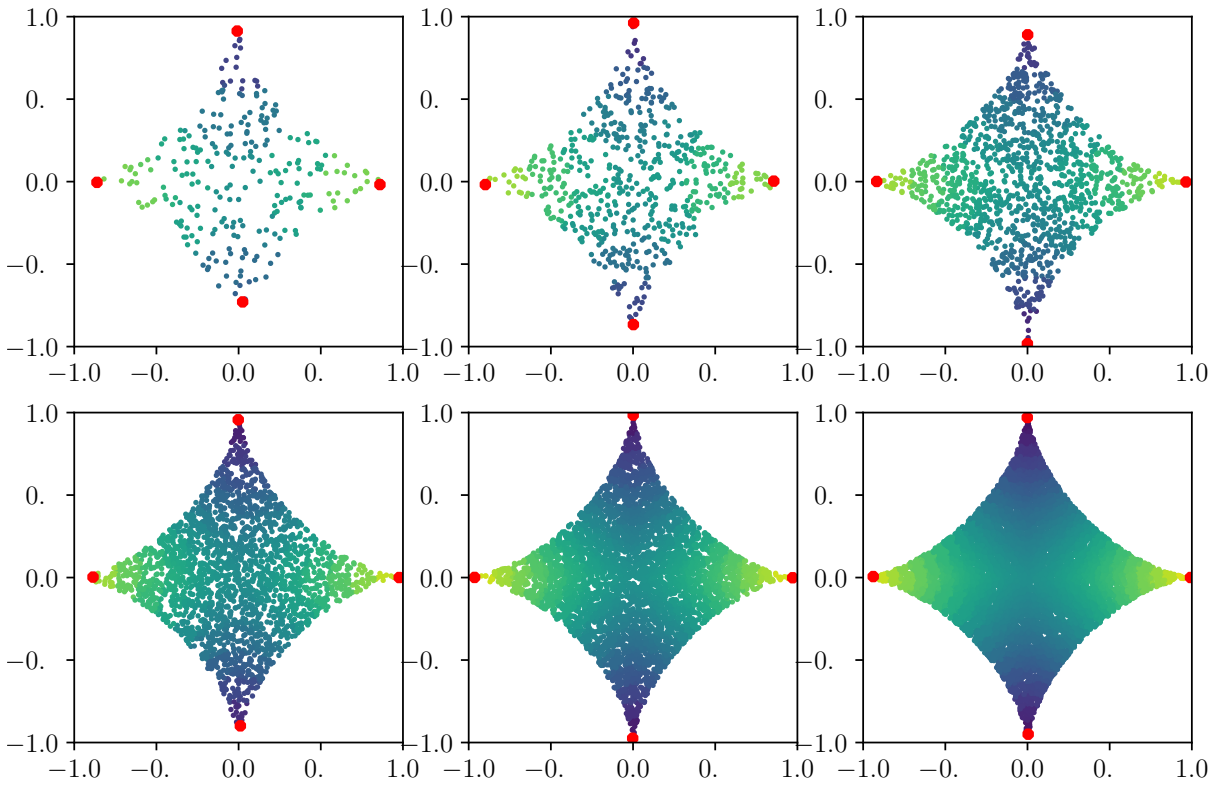


Figure 2: Solutions of the Lipschitz learning problem for different graph resolutions on a non-convex domain with mixed Neumann and Dirichlet conditions. The four labelled points in $\mathcal{O}_n^{\mathrm{cp}}$ are marked in red.

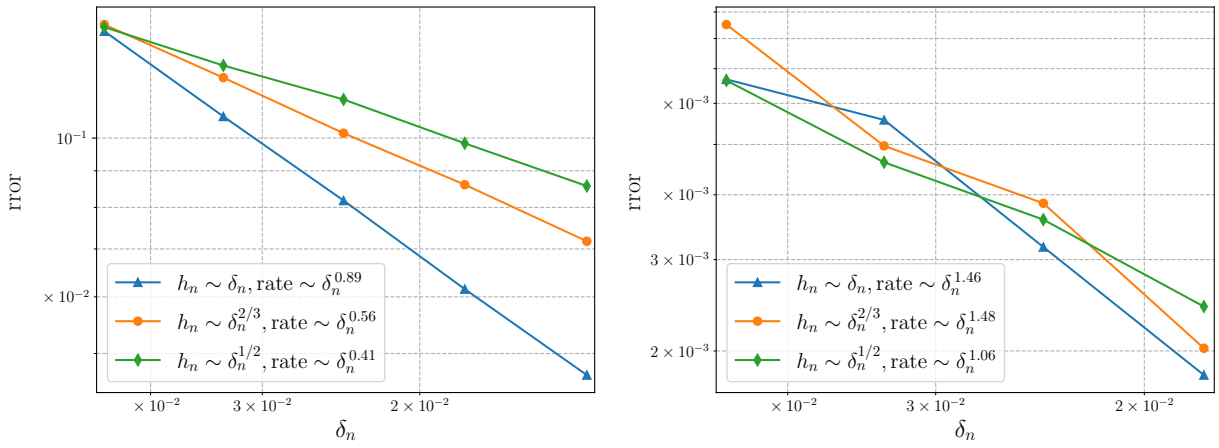


Figure 3: Empirical convergence rates of the solutions in Fig. 2 for unit weights (**left**) and singular weights (**right**) and different scalings with the constraint set $\mathcal{O}_n^{\mathrm{cp}}$. The rates are specified in the legend. Note that our theory does not prove convergence for the blue scaling.

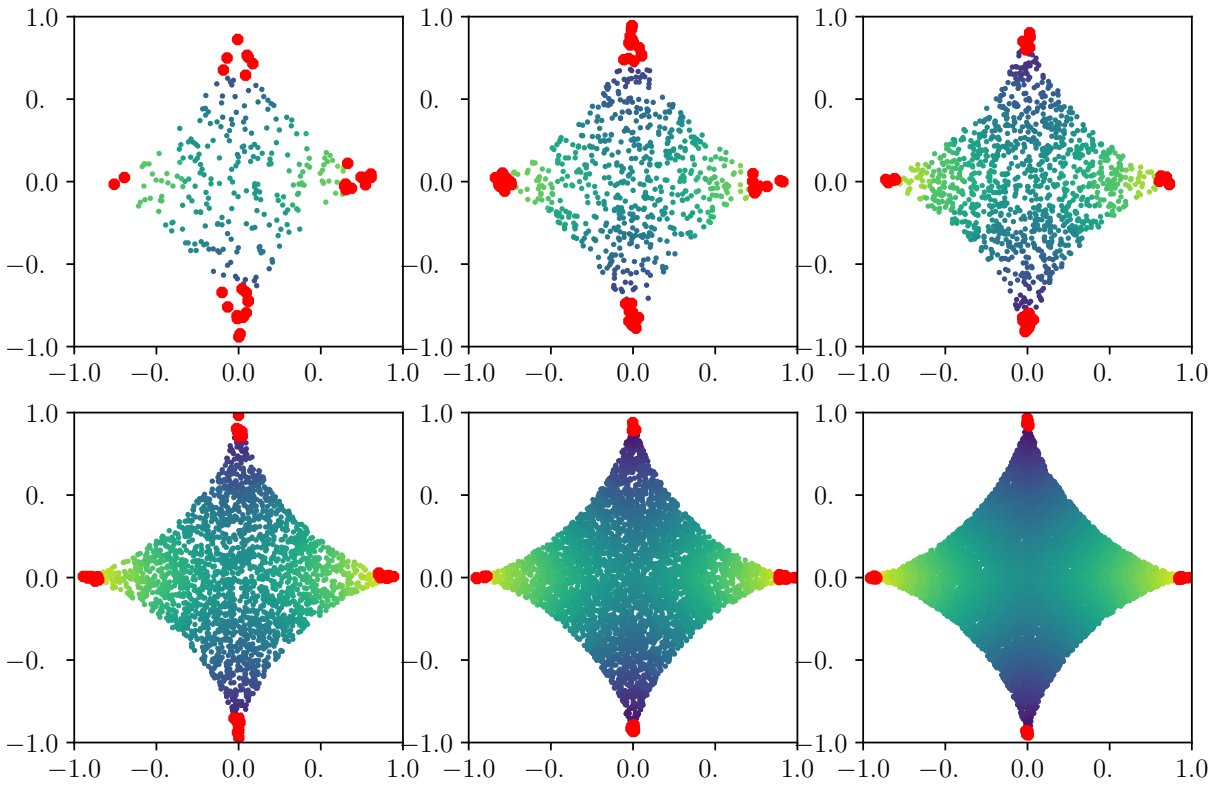


Figure 4: Solutions of the Lipschitz learning problem for different graph resolutions on a non-convex domain with mixed Neumann and Dirichlet conditions. The labelled points in $\mathcal{O}_n^{\mathrm{dil}}$ are marked in red.

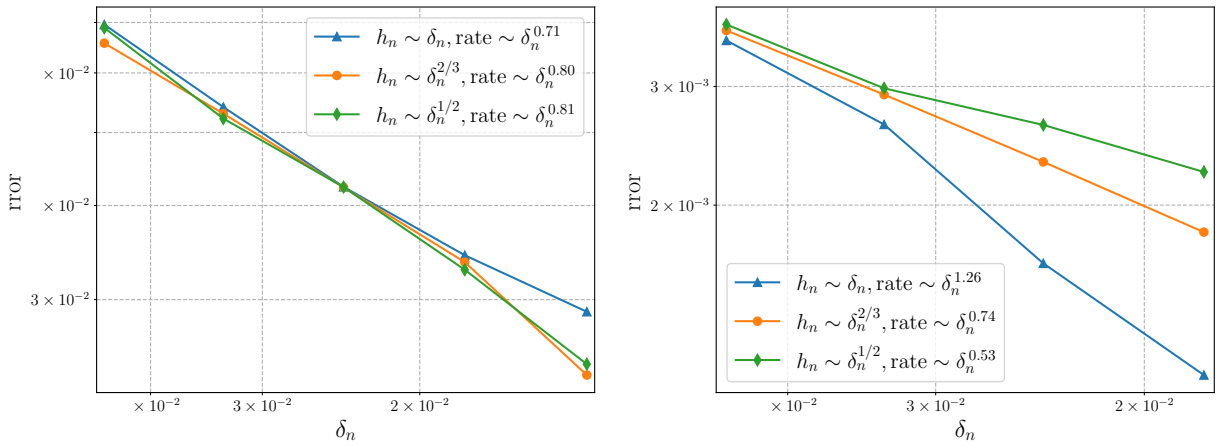


Figure 5: Empirical convergence rates of the solutions in Fig. 4 for unit weights (**left**) and singular weights (**right**) and different scalings with the constraint set $\mathcal{O}_n^{\mathrm{dil}}$. The rates are specified in the legend. Note that our theory does not prove convergence for the blue scaling.

Print P3
CLIP: Cheap Lipschitz training of neural networks

L. Bungert, R. Raab, T. Roith, L. Schwinn, and D. Tenbrinck
journaltitle (2021)

CLIP: Cheap Lipschitz Training of Neural Networks*

Leon Bungert[1], René Raab[2], Tim Roith[1], Leo Schwinn[2], and Daniel Tenbrinck[1]

[1] Department Mathematics, Friedrich-Alexander University Erlangen-Nürnberg, Cauerstraße 11, 91058 Erlangen
{leon.bungert,tim.roith,daniel.tenbrinck}@fau.de
[2] Machine Learning and Data Analytics Lab, Friedrich-Alexander University Erlangen-Nürnberg, Carl-Thiersch-Straße 2b, 91052 Erlangen
{rene.raab,leo.schwinn}@fau.de

Abstract. Despite the large success of deep neural networks (DNN) in recent years, most neural networks still lack mathematical guarantees in terms of stability. For instance, DNNs are vulnerable to small or even imperceptible input perturbations, so called adversarial examples, that can cause false predictions. This instability can have severe consequences in applications which influence the health and safety of humans, e.g., biomedical imaging or autonomous driving. While bounding the Lipschitz constant of a neural network improves stability, most methods rely on restricting the Lipschitz constants of each layer which gives a poor bound for the actual Lipschitz constant.
In this paper we investigate a variational regularization method named *CLIP* for controlling the Lipschitz constant of a neural network, which can easily be integrated into the training procedure. We mathematically analyze the proposed model, in particular discussing the impact of the chosen regularization parameter on the output of the network. Finally, we numerically evaluate our method on both a nonlinear regression problem and the MNIST and Fashion-MNIST classification databases, and compare our results with a weight regularization approach.

Keywords: Deep neural network · Machine learning · Lipschitz constant · Variational regularization · Stability · Adversarial attack.

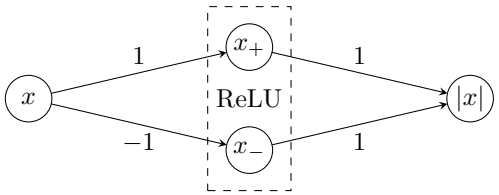
1 Introduction

Deep neural networks (DNNs) have led to astonishing results in various fields, such as computer vision [13] and language processing [18]. Despite its large success, deep learning and the resulting neural network architectures also bear certain drawbacks. On the one hand, their behaviour is mathematically not yet fully

* This work was supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 777826 (NoMADS) and by the German Ministry of Science and Technology (BMBF) under grant agreement No. 05M2020 (DELETO).

understood and there exist not many rigorous analytical results. On the other hand, most trained networks are vulnerable to adversarial attacks [9]. In image processing, adversarial examples are small, typically imperceptible perturbations to the input that cause misclassifications. In domains like autonomous driving or healthcare this can potentially have fatal consequences. To mitigate these weaknesses, many methods were proposed to make neural networks more robust and reliable. One straight-forward approach is to regularize the norms of the weight matrices [10]. Another idea is adversarial training [16,23,24] which uses adversarial examples generated from the training data to increase robustness locally around the training samples. In this context, also the Lipschitz constant of neural networks has attracted a lot of attention (e.g., [7,8,26,29]) since it constitutes a worst-case bound for their stability, and Lipschitz-regular networks are reported to have superior generalization properties [17]. In [27] Lipschitz regularization around training samples has been related to adversarial training.

Unfortunately, the Lipschitz constant of a neural network is NP-hard to compute [22], hence several methods in the literature aim to achieve more stability of neural networks by bounding the Lipschitz constant of each individual layer [1,10,12,20] or the activation functions [3]. However, this strategy is a very imprecise approximation to the real Lipschitz constant and hence the trained networks may suffer from inferior expressivity and can even be less robust [15]. The following example demonstrates a representation of the absolute value function, which clearly has Lipschitz constant 1, using a neural network whose individual layers have larger Lipschitz constants (cf. [11] for details).



(1)

In this paper we propose a new variational regularization method for training neural networks, while simultaneously controlling their Lipschitz constant. Since the additional computations can easily be embedded in the standard training process and are fully parallelizable, we name our method *Cheap Lipschitz Training of Neural Networks (CLIP)*. Instead of bounding the Lipschitz constant of each layer individually as in prior approaches, we aim to minimize the global Lipschitz constant of the neural network via an additional regularization term.

The **main contributions** of this paper are as follows. First, we motivate and introduce the proposed variational regularization method for Lipschitz training of neural networks, which leads to a min-max problem. For optimizing the proposed model we formulate a stochastic gradient descent-ascent method, the CLIP algorithm. Subsequently, we perform mathematical analysis of the proposed variational regularization method by discussing existence of solutions. Here we use techniques, developed for the analysis of variational regularization methods of inverse problems [4,6]. We analyse the two limit cases, namely the regularization parameter tending to zero and infinity, where we prove convergence to a

Lipschitz minimal fit of the data and a generalized barycenter, respectively. Finally, we evaluate the proposed approach by performing numerical experiments on a regression example and the MNIST and Fashion-MNIST classification databases [14,28]. We show that the introduced variational regularization term leads to improved stability compared to networks without additional regularization or with layerwise Lipschitz regularization. For this we investigate the impact of noise and adversarial attacks on the trained neural networks.

2 Model and Algorithms

Given a finite training set $\mathcal{T} \subset \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ denote the input and output space, we propose to determine parameters $\theta \in \Theta$ of a neural network $f_\theta : \mathcal{X} \to \mathcal{Y}$ by solving the empirical risk minimization problem with an additional Lipschitz regularization term

$$\theta_\lambda \in \operatorname*{arg\,min}_{\theta \in \Theta} \frac{1}{|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} \ell(f_\theta(x), y) + \lambda \operatorname{Lip}(f_\theta). \tag{2}$$

Here, $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a loss function, $\lambda > 0$ is a regularization parameter and the Lipschitz constant of the network $f_\theta : \mathcal{X} \to \mathcal{Y}$ is defined as

$$\operatorname{Lip}(f_\theta) := \sup_{x,x' \in \mathcal{X}} \frac{\|f_\theta(x) - f_\theta(x')\|}{\|x - x'\|}. \tag{3}$$

The norms involved in this definition can be chosen freely, however, for our CLIP algorithm we assume differentiability. The fundamental difference of the proposed model (2) compared to existing approaches in the literature [1,10,12,20] is that we use the actual Lipschitz constant (3) as regularizer and do not rely on the layer-based upper bound

$$\operatorname{Lip}(f_\theta) \;\leq\; \prod_{l=1}^{L} \operatorname{Lip}(\Phi_l) \tag{4}$$

for neural networks of the form $f_\theta = \Phi_L \circ \ldots \Phi_1$. By plugging (3) into (2) this becomes a min-max problem, which we solve numerically by using a modification of the stochastic batch gradient descent algorithm with momentum $\mathrm{SGDM}_{\eta,\gamma}$ with learning rate $\eta > 0$ and momentum $\gamma \geq 0$, see, e.g. [21].

Since evaluating the Lipschitz constant of a neural network is NP-hard [22], we approximate it on a finite subset $\mathcal{X}_{\mathrm{Lip}} \subset \mathcal{X} \times \mathcal{X}$ by setting

$$\operatorname{Lip}(f_\theta, \mathcal{X}_{\mathrm{Lip}}) := \max_{(x,x')\in\mathcal{X}_{\mathrm{Lip}}} \frac{\|f_\theta(x) - f_\theta(x')\|}{\|x - x'\|}. \tag{5}$$

To make sure that the tuples in $\mathcal{X}_{\mathrm{Lip}}$ correspond to points with a high Lipschitz constant during the whole training process, we update $\mathcal{X}_{\mathrm{Lip}}$ using gradient ascent of the difference quotient in (5) with adaptive step size, see Algorithm 1. We call

Algorithm 1: AdversarialUpdate$_\tau$ (with step size $\tau > 0$)

$L(x, x') := \|f_\theta(x) - f_\theta(x')\| \,/\, \|x - x'\|, \quad x, x' \in \mathcal{X}$
for $(x, x') \in \mathcal{X}_{\mathrm{Lip}}$ **do**
  $x \leftarrow x + \tau\, L(x, x') \nabla_x L(x, x')$
  $x' \leftarrow x' + \tau\, L(x, x') \nabla_{x'} L(x, x')$

Algorithm 2: CLIP (Cheap Lipschitz Training)

for *epoch* $e = 1$ **to** E **do**
  for *minibatch* $B \subset \mathcal{T}$ **do**
    $\mathcal{X}_{\mathrm{Lip}} \leftarrow \mathrm{AdversarialUpdate}_\tau(\mathcal{X}_{\mathrm{Lip}})$
    $\theta \leftarrow \mathrm{SGDM}_{\eta,\gamma}\left(\frac{1}{|B|}\sum_{(x,y)\in B} \ell(f_\theta(x), y) + \lambda\, \mathrm{Lip}(f_\theta, \mathcal{X}_{\mathrm{Lip}})\right)$
    if $\mathcal{A}(f_\theta; \mathcal{T}) > \alpha$ **then**
      $\lambda \leftarrow \lambda + d\lambda$
    else
      $\lambda \leftarrow \lambda - d\lambda$

this *adversarial update*, since the tuples in $\mathcal{X}_{\mathrm{Lip}}$ approach points which have a small distance to each other while still getting classified differently by the network.

To illustrate the effect of adversarial updates, Figure 1 depicts a pair of images from the Fashion-MNIST database before and after applying Algorithm 1, using a neural network trained with CLIP (see Section 4.2 for details). The second pair realizes a large Lipschitz constant and hence lies close to the decision boundary of the neural network.

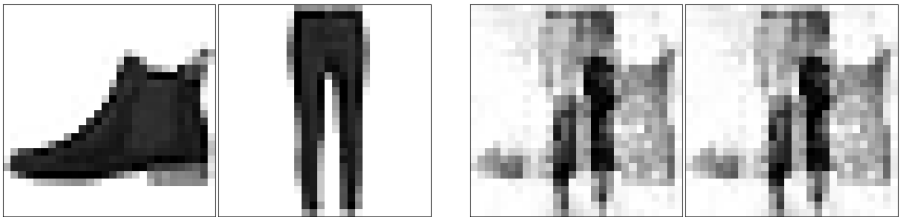


Fig. 1: Effect of Algorithm 1. The initial tuple gets classified correctly with high confidence ($> 90\%$). After adversarial updates the pair gets misclassified as the respective other class with low confidence ($< 30\%$).

The regularization parameter λ in (2) is chosen by a discrepancy principle from inverse problems [2,5]: We compare an accuracy measure $\mathcal{A}(f_\theta; \mathcal{T})$ for the neural network f_θ, evaluated on the training data, to a target accuracy α. Combining the adversarial update from Algorithm 1 with stochastic gradient descent with momentum, we propose the CLIP algorithm, which is given in Algorithm 2. Similarly to [24] one can reuse computations of the backward pass in the gradient step of the Lipschitz regularizer with respect to θ in Algorithm 2 to compute the gradient with respect to the input x, needed in Algorithm 1, which indeed makes CLIP a 'cheap' extension to conventional training. We would like to emphasize that the set $\mathcal{X}_{\text{Lip}}$, on which we approximate the Lipschitz constant of the neural network, is not fixed but is updated in an optimal way in every minibatch. Hence, it plays the role of an adaptive 'training set' for the Lipschitz constant. Future analysis will investigate the approximation quality of this approach in the framework of stochastic gradient descent methods.

3 Analysis

In this section we prove analytical results on the model (2), which are inspired by analogous statements for variational regularization methods for inverse problems (see, e.g., [4,6]). While the CLIP Algorithm 1 is stochastic in nature, our analysis focuses on the deterministic model (2), whose solutions are approximated by CLIP. For our results we have to pose some mild assumptions on the loss function ℓ and the neural network f_θ which are fulfilled for most loss functions and network architectures, such as mean squared error or cross entropy loss and architectures like feed-forward, convolutional, or residual networks with continuous activation functions. Furthermore, we assume that $\mathcal{X}, \mathcal{Y}$, and Θ are finite-dimensional spaces, which is the case in most applications and lets us state our theory more compactly than using Banach space topologies.

Assumption 1. We assume that the loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R} \cup \{\infty\}$ satisfies:

(a) $\ell(y, y') \geq 0$ for all $y, y' \in \mathcal{Y}$,

(b) $y \mapsto \ell(y, y')$ is lower semi-continuous for all $y' \in \mathcal{Y}$.

Assumption 2. We assume that the map $\theta \mapsto f_\theta(x)$ is continuous for all $x \in \mathcal{X}$.

Assumption 3. We assume that there exists $\theta \in \Theta$ such that

$$\frac{1}{|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} \ell(f_\theta(x), y) + \lambda \operatorname{Lip}(f_\theta) < \infty.$$

We will also need the following lemma, which states that the Lipschitz constant of f_θ is lower semi-continuous with respect to the parameters θ.

Lemma 1. *Under Assumption 2 the functional* $\theta \mapsto \operatorname{Lip}(f_\theta)$ *is lower semi-continuous.*

Proof. Let $(\theta_i)_{i\in\mathbb{N}} \subset \Theta$ converge to $\theta \in \Theta$. Using the continuity of $\theta \mapsto f_\theta(x)$ and the lower semi-continuity of the norm $\|\cdot\|$, one can compute

$$\begin{aligned}\mathrm{Lip}(f_\theta) &= \sup_{x,x'\in\mathcal{X}} \frac{\|f_\theta(x) - f_\theta(x')\|}{\|x - x'\|} \leq \sup_{x,x'\in\mathcal{X}} \liminf_{i\to\infty} \frac{\|f_{\theta_i}(x) - f_{\theta_i}(x')\|}{\|x - x'\|} \\ &\leq \liminf_{i\to\infty} \sup_{x,x'\in\mathcal{X}} \frac{\|f_{\theta_i}(x) - f_{\theta_i}(x')\|}{\|x - x'\|} = \liminf_{i\to\infty} \mathrm{Lip}(f_{\theta_i}).\end{aligned}$$

3.1 Existence of Solutions

We start with an existence statement for a slight modification of (2), which also regularizes the network parameters and guarantees coercivity of the objective.

Proposition 1. *Under Assumptions 1–3 the problem*

$$\min_{\theta\in\Theta} \frac{1}{|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} \ell(f_\theta(x), y) + \lambda\,\mathrm{Lip}(f_\theta) + \mu\,\|\theta\|_\Theta \tag{6}$$

has a solution for all values $\lambda, \mu > 0$. Here, $\|\cdot\|_\Theta$ denotes a norm on Θ.

Proof. We let $(\theta_i)_{i\in\mathbb{N}} \subset \Theta$ be a minimizing sequence, whose existence is assured by Assumption 3. Since $\mu > 0$ in (6), the norms $\|\theta_i\|_\Theta$ for $i \in \mathbb{N}$ are uniformly bounded and hence, up to a subsequence which we do not relabel, $\theta_i \to \theta^* \in \Theta$ as $i \to \infty$. The lower semi-continuity of ℓ, Lip, and $\|\cdot\|_\Theta$ together with the continuity of $\theta \mapsto f_\theta(x)$ then shows that θ^* solves (6).

Remark 1. If Θ is compact or even finite, existence for (2) is assured since any minimizing sequence in Θ is compact. The reason that in the general case we can only show existence for the regularized problem (6) is that it assures boundedness of minimizing sequences. Even for the unregularized empirical risk minimization problem existence is typically assumed in the *realizability assumption* [25].

3.2 Dependency on the Regularization Parameter

We start with the statement that the Lipschitz constant in (2) decreases and the empirical risk increases as the regularization parameter λ grows.

Proposition 2. *Let θ_λ solve (2) for $\lambda \geq 0$. Then it holds*

$$\lambda \longmapsto \frac{1}{|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} \ell(f_{\theta_\lambda}(x), y) \quad \textit{is non-decreasing,} \tag{7}$$

$$\lambda \longmapsto \mathrm{Lip}(f_{\theta_\lambda}) \quad \textit{is non-increasing.} \tag{8}$$

Proof. The proof works precisely as in [6].

Now we will study the limit cases $\lambda \searrow 0$ and $\lambda \to \infty$ in (2). As in Section 3.1, because of lack of coercivity, we have to assume that the corresponding sequences of optimal network parameters converge.

Our first statement deals with the behavior as the regularization parameter λ tends to zero. We show that in this case the learned neural networks converge to one which fits the training data with the smallest Lipschitz constant.

Proposition 3. *Let Assumptions 1–3 and the realizability assumption [25]*

$$\min_{\theta \in \Theta} \frac{1}{|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} \ell(f_\theta(x), y) = 0 \tag{9}$$

be satisfied. Let θ_λ be a solution of (2) for $\lambda > 0$. If $\theta_\lambda \to \theta^\dagger \in \Theta$ as $\lambda \searrow 0$, then

$$\theta^\dagger \in \arg\min \left\{ \operatorname{Lip}(f_\theta) \,:\, \theta \in \Theta,\ \frac{1}{|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} \ell(f_\theta(x), y) = 0 \right\} \tag{10}$$

if this problem admits a solution with $\operatorname{Lip}(f_\theta) < \infty$.

Proof. We fix $\theta \in \Theta$ which satisfies (9) and $\operatorname{Lip}(f_\theta) < \infty$. Using the optimality of θ_λ we infer

$$\liminf_{\lambda \searrow 0} \frac{1}{\lambda\,|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} \ell(f_{\theta_\lambda}(x), y) + \operatorname{Lip}(f_{\theta_\lambda}) \;\le\; \operatorname{Lip}(f_\theta). \tag{11}$$

Using that ℓ is non-negative we conclude that $\operatorname{Lip}(f_{\theta_\lambda}) \le \operatorname{Lip}(f_\theta)$ and using Lemma 1 we get

$$\operatorname{Lip}(f_{\theta^\dagger}) \;\le\; \liminf_{\lambda \searrow 0} \operatorname{Lip}(f_{\theta_\lambda}) \;\le\; \operatorname{Lip}(f_\theta).$$

The lower semi-continuity of the loss ℓ and the continuity of the map $\theta \mapsto f_\theta(x)$ for all $x \in \mathcal{X}$ imply

$$\ell(f_{\theta^\dagger}(x), y) \le \liminf_{\lambda \searrow 0} \ell(f_{\theta_\lambda}(x), y), \quad \forall (x, y) \in \mathcal{T}.$$

If we assume that $\ell(f_{\theta^\dagger}(x), y) > 0$ for some $(x, y) \in \mathcal{T}$, then we obtain

$$\liminf_{\lambda \searrow 0} \frac{1}{\lambda\,|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} \ell(f_{\theta_\lambda}(x), y) \ge \infty$$

which is a contradiction to (11) and implies that $\theta^\dagger$ satisfies (10).

Note that if the realizability assumption (9) is not satisfied, e.g., for noisy data or small network architectures, the previous statement has to be refined, which is subject to future work. The next proposition deals with the case in which the parameter λ approaches infinity. In this case the learned neural networks approach a constant map, coinciding with a generalized barycenter of the data. We denote by $\mathcal{T}_\mathcal{Y} = \{y : (x, y) \in \mathcal{T}\}$ the set of the data in the output space.

Proposition 4. *Let Assumptions 1–3 be satisfied and assume that*

$$\mathcal{M} := \{y \in \mathcal{Y} \,:\, \exists \theta \in \Theta,\, f_\theta(x) = y,\, \forall x \in \mathcal{X}\} \neq \emptyset. \tag{12}$$

Let θ_λ denote a solution of (2) for $\lambda > 0$. If $\theta_\lambda \to \theta_\infty \in \Theta$ as $\lambda \to \infty$, then $f_{\theta_\infty}(x) = \hat{y}$ for all $x \in \mathcal{X}$ where

$$\hat{y} \in \arg\min \left\{ \frac{1}{|\mathcal{T}|} \sum_{y \in \mathcal{T}_\mathcal{Y}} \ell(y', y) \,:\, y' \in \mathcal{M} \right\}. \tag{13}$$

Proof. From the optimality of θ_λ we deduce

$$\sum_{(x,y)\in\mathcal{T}} \ell(f_{\theta_\lambda}(x), y) + \lambda \operatorname{Lip}(f_{\theta_\lambda}) \;\leq\; \sum_{y \in \mathcal{T}_\mathcal{Y}} \ell(y', y), \quad \forall y' \in \mathcal{M}.$$

Hence, by letting $\lambda \to \infty$ we obtain $\lim_{\lambda\to\infty} \operatorname{Lip}(f_{\theta_\lambda}) = 0$. If we now assume that $\theta_\lambda \to \theta_\infty$ as $\lambda \to \infty$, we can use the lower semi-continuity of the Lipschitz constant to obtain

$$\operatorname{Lip}(f_{\theta_\infty}) \leq \liminf_{\lambda\to\infty} \operatorname{Lip}(f_{\theta_\lambda}) = 0.$$

Hence, $f_{\theta_\infty} \equiv \hat{y}$ for some element $\hat{y} \in \mathcal{Y}$. Using the lower semi-continuity of the loss function, we can conclude the proof with

$$\begin{aligned} \sum_{y\in\mathcal{T}_\mathcal{Y}} \ell(\hat{y}, y) \;&=\; \sum_{(x,y)\in\mathcal{T}} \ell(f_{\theta_\infty}(x), y) \;\leq\; \liminf_{\lambda\to\infty} \sum_{(x,y)\in\mathcal{T}} \ell(f_{\theta_\lambda}(x), y) \\ &\leq \sum_{y\in\mathcal{T}_\mathcal{Y}} \ell(y', y), \quad \forall y' \in \mathcal{M}. \end{aligned}$$

Remark 2. For a Euclidean loss function, i.e., $\ell(y', y) = \|y' - y\|^2$, the quantity (13) coincides with a projection of the barycenter of the training samples in $\mathcal{T}_\mathcal{Y}$ onto the manifold of constant networks $\mathcal{M}$, given by (12). This can be seen as follows: Let $b := \frac{1}{|\mathcal{T}|} \sum_{y\in\mathcal{T}_\mathcal{Y}} y$ be the barycenter of the training samples. Then

$$\begin{aligned} \frac{1}{|\mathcal{T}|} \sum_{y\in\mathcal{T}_\mathcal{Y}} \|y' - y\|^2 &= \|y'\|^2 - 2\,\langle y', b\rangle + \frac{1}{|\mathcal{T}|} \sum_{y\in\mathcal{T}_\mathcal{Y}} \|y\|^2 \\ &= \|y' - b\|^2 - \|b\|^2 + \frac{1}{|\mathcal{T}|} \sum_{y\in\mathcal{T}_\mathcal{Y}} \|y\|^2, \qquad \forall y' \in \mathcal{Y}. \end{aligned}$$

Using this equality for general $y' \in \mathcal{M}$ and for $y' = \hat{y}$ given by (13) we obtain

$$0 \;\leq\; \frac{1}{|\mathcal{T}|} \sum_{y\in\mathcal{T}_\mathcal{Y}} \|y' - y\|^2 - \frac{1}{|\mathcal{T}|} \sum_{y\in\mathcal{T}_\mathcal{Y}} \|\hat{y} - y\|^2 \;=\; \|y' - b\|^2 - \|\hat{y} - b\|^2.$$

This in particular implies that $\|\hat{y} - b\| \leq \|y' - b\|$ for all $y' \in \mathcal{M}$ as desired.

4 Experiments

After having studied theoretical properties of the proposed variational Lipschitz regularization method (2), we will apply the CLIP algorithm to regression and classification tasks in this section. In a first experiment we train a neural network to approximate a nonlinear function given by noisy samples, where we illustrate that CLIP prevents strong oscillations, which appear in unregularized networks. In a second experiment we use CLIP for training two different classification networks on the MNIST and Fashion-MNIST databases. Here, we quantitatively show that CLIP yields superior robustness to noise and adversarial attacks compared to unregularized and weight-regularized neural networks. In all experiments we used Euclidean norms for the Lipschitz constant (3).

Our implementation of CLIP is available on `github`.[3]

4.1 Nonlinear Regression

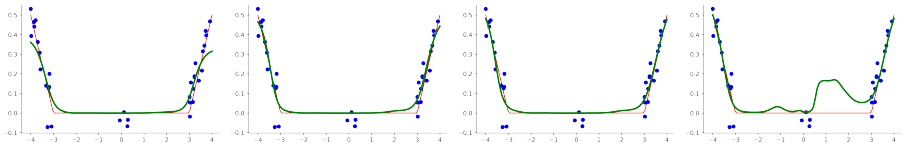
In this experiment we train a neural network to approximate the nonlinear function $f(x) = \frac{1}{2}\max(|x| - 3, 0)$ with three hidden layers consisting of 500, 200 and 100 neurons using the *sigmoid* activation function. We construct 100 noisy training pairs by sampling values of f on the set $[-4, -3] \cup [-0.3, 0.3] \cup [3, 4]$. The set of Lipschitz training samples is randomly chosen in every epoch and consists of tuples (x, x') where $x \in [-4, 4]$ and x' is a noise perturbation of x. Figure 2 shows the learned networks after applying the CLIP Algorithm 2 with different values of the regularization parameter λ. Note that for this experiment we do not use a target accuracy but choose a fixed value of λ. For $\lambda = 10$ the learned network is very smooth, which yields a poor approximation of the ground truth especially at the boundary of the domain. For decreasing values of λ the networks approach the ground truth solution steadily, showing no instabilities. In contrast, the unregularized network with $\lambda = 0$ shows strong oscillations in regions of missing data, which implies a poor generalization capability. Both for $\lambda = 10^{-10}$ and $\lambda = 0$ the trained networks do not fit the noisy data. A reason for this is that some of the data points are negative and we use non-negative sigmoid activation in the last layer. Also, using stochastic gradient descent implicitly regularizes the problem and avoids convergence to spurious critical points of the loss.

Note that we warmstart the computation for $\lambda = 10$ with the pretrained unregularized network. The computations for $\lambda = 1$ and $\lambda = 10^{-10}$ are warmstarted with the respective larger value of λ. The latter successive reduction of the regularization parameter is necessary in order to "select" the desired Lipschitz-minimal solution as $\lambda \to 0$, which resembles Bregman iterations [19].

4.2 Classification on MNIST and Fashion-MNIST

We evaluate the robustness of neural networks trained with the proposed variational CLIP algorithm on the popular MNIST [14] and Fashion-MNIST [28]

[3] https://github.com/TimRoith/CLIP



(a) $\lambda = 10$ (b) $\lambda = 1$ (c) $\lambda = 10^{-10}$ (d) $\lambda = 0$

Fig. 2: Lipschitz regularized networks for different values of λ. **Red:** ground truth. **Blue:** noisy ground truth samples. **Green:** trained neural network.

	Training Type	Train	Test	Noise	PGD	μ, λ
MNIST	Standard	**95.6**	89.8	71.8	25.3	0
	Weight Reg.$_{95}$	93.1	89.0	71.4	25.4	0.0003
	Weight Reg.$_{90}$	89.8	89.7	71.9	24.8	0.0015
	Weight Reg.$_{85}$	84.8	87.2	72.3	24.9	0.0031
	CLIP$_{95}$	95.3	90.7	70.3	24.4	0.01
	CLIP$_{90}$	91.8	**91.6**	**78.0**	36.2	3.91
	CLIP$_{85}$	87.6	87.2	74.5	**37.7**	9.82
Fashion-MNIST	Standard	**98.5**	**92.0**	25.7	1.9	0
	Weight Reg.$_{95}$	94.8	91.2	26.9	3.5	0.0011
	Weight Reg.$_{90}$	89.8	89.6	28.1	6.0	0.0023
	Weight Reg.$_{85}$	85.3	85.4	**34.2**	10.2	0.0091
	CLIP$_{95}$	94.6	**91.9**	19.8	9.1	0.18
	CLIP$_{90}$	90.6	88.9	23.3	13.0	0.74
	CLIP$_{85}$	85.9	83.1	31.9	**14.9**	4.18

Table 1: Accuracies [%] for different training setups, tested on train, test, noisy, and adversarial **(Fashion-)MNIST** data. The target accuracies are given by subscript. The regularization parameters μ and λ after training are also shown.

databases, which contain 28×28 grayscale images of handwritten digits and fashion articles, respectively. They are split into $60,000$ samples for training and $10,000$ samples for evaluation. For MNIST we train a neural network with two hidden layers with sigmoid activation function containing 64 neurons each. For Fashion-MNIST we use a simple convolutional neural network from [16]. We train non-regularized networks, CLIP regularized networks, and networks where we perform a L_2 regularization of the weights. The latter corresponds to (6) with $\lambda = 0$ and $\mu > 0$, where we update μ with the same discrepancy principle as for CLIP. Note that the weight regularization implies a layerwise Lipschitz bound of the type (4). The quantity $\mathcal{A}(f_\theta; \mathcal{T})$ in Algorithm 2 is chosen as the percentage of correctly classified training samples. We set the target accuracies for L_2 regularized and CLIP regularized networks to 85%, 90%, and 95%. The Lipschitz training set consists of 6,000 images removed from the training set and their noisy versions. We compare the robustness of all trained networks to adversarial perturbations $x + \delta$ of a sample $x \in \mathcal{X}$ in the L_2 norm, such that $\|\delta\| \leq \varepsilon$ with

$\varepsilon = 2$, calculated with the Projected Gradient Descent (PGD) attack [16] as

$$x_{\text{adv}}^{t+1} = \Pi\left(x_{\text{adv}}^{t} + \alpha \operatorname{sign}(\nabla_x \ell(f_\theta(x_{\text{adv}}^{t}), y)\right), \tag{14}$$

where $\alpha > 0$ is the step size and x_{adv}^{t} describes the adversarial example at iteration t. Here, $\Pi : \mathcal{X} \to \mathcal{X}$ is a projection operator onto the L_2-ball with radius ε, and sign is the componentwise signum operator. We set the step size $\alpha = 0.25$ and the total number of iterations to 100. Furthermore, we use Gaussian noise with zero mean and unit variance for an additional robustness evaluation. We track the performance against the PGD attack of each model during training and use the model checkpoint with the highest robustness for testing.

Table 1 shows the networks' performance for the MNIST and Fashion-MNIST databases. The CLIP algorithm increases robustness with respect to Gaussian noise and PGD attacks depending on the target accuracy. One observes an inversely proportional relationship between the target accuracy and the magnitude of the regularization parameter λ as expected from Proposition 2. The results for MNIST indicate a higher robustness of the CLIP regularized neural networks with respect to noise perturbations and adversarial attacks in comparison to unregularized or weight-regularized neural networks with similar accuracy on unperturbed test sets. Similar observations can be made for the Fashion-MNIST database. While our method is also more robust under adversarial attacks, it is slightly more prone to noise than the weight-regularized neural networks.

5 Conclusion and Outlook

We have proposed a variational regularization method to limit the Lipschitz constant of a neural network during training and derived *CLIP*, a stochastic gradient descent-ascent training method. Furthermore, we have studied the dependency of the trained neural networks on the regularization parameter and investigated the limiting behavior as the parameter tends to zero and infinity. Here, we have proved convergence to Lipschitz-minimal data fits or constant networks, respectively. We have evaluated the CLIP algorithm on regression and classification tasks and showed that our method effectively increases the stability of the learned neural networks compared to weight regularization methods and unregularized neural networks. In future work we will analyze convergence and theoretical guarantees of CLIP using techniques from stochastic analysis.

References

1. Anil, C., Lucas, J., Grosse, R.B.: Sorting Out Lipschitz Function Approximation. In: ICML. PMLR, vol. 97, pp. 291–301 (2019)
2. Anzengruber, S.W., Ramlau, R.: Morozov's discrepancy principle for Tikhonov-type functionals with nonlinear operators. Inverse Problems **26**(2), 025001 (2009)
3. Aziznejad, S., Gupta, H., Campos, J., Unser, M.: Deep neural networks with trainable activations and controlled Lipschitz constant. IEEE Transactions on Signal Processing **68**, 4688–4699 (2020)

4. Bungert, L., Burger, M.: Solution paths of variational regularization methods for inverse problems. Inverse Problems **35**(10), 105012 (2019)
5. Bungert, L., Burger, M., Korolev, Y., Schönlieb, C.B.: Variational regularisation for inverse problems with imperfect forward operators and general noise models. Inverse Problems **36**(12), 125014 (2020)
6. Burger, M., Osher, S.: A guide to the TV zoo. In: Level set and PDE based reconstruction methods in imaging, pp. 1–70. Springer (2013)
7. Combettes, P.L., Pesquet, J.C.: Lipschitz certificates for layered network structures driven by averaged activation operators. SIAM Journal on Mathematics of Data Science **2**(2), 529–557 (2020)
8. Fazlyab, M., Robey, A., Hassani, H., Morari, M., Pappas, G.: Efficient and Accurate Estimation of Lipschitz Constants for Deep Neural Networks. In: NIPS (2019)
9. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and Harnessing Adversarial Examples. In: ICLR (2015)
10. Gouk, H., Frank, E., Pfahringer, B., Cree, M.J.: Regularisation of neural networks by enforcing Lipschitz continuity. Machine Learning pp. 1–24 (2020)
11. Huster, T., Chiang, C.Y.J., Chadha, R.: Limitations of the Lipschitz constant as a defense against adversarial examples. In: ECML PKDD. pp. 16–29. Springer (2018)
12. Krishnan, V., Makdah, A.A.A., Pasqualetti, F.: Lipschitz Bounds and Provably Robust Training by Laplacian Smoothing. arXiv preprint arXiv:2006.03712 (2020)
13. Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech. rep. (2009)
14. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
15. Liang, Y., Huang, D.: Large Norms of CNN Layers Do Not Hurt Adversarial Robustness. arXiv preprint arXiv:2009.08435 (2020)
16. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards Deep Learning Models Resistant to Adversarial Attacks. In: ICLR (2018)
17. Oberman, A.M., Calder, J.: Lipschitz regularized deep neural networks converge and generalize. arXiv preprint arXiv:1808.09540 (2018)
18. van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A.W., Kavukcuoglu, K.: WaveNet: A Generative Model for Raw Audio. In: The 9th ISCA Speech Synthesis Workshop. p. 125 (2016)
19. Osher, S., Burger, M., Goldfarb, D., Xu, J., Yin, W.: An iterative regularization method for total variation-based image restoration. Multiscale Model Sim. **4**(2), 460–489 (2005)
20. Roth, K., Kilcher, Y., Hofmann, T.: Adversarial Training is a Form of Data-dependent Operator Norm Regularization. In: NeurIPS (2019)
21. Ruder, S.: An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747 (2016)
22. Scaman, K., Virmaux, A.: Lipschitz Regularity of Deep Neural Networks: Analysis and Efficient Estimation. In: NeurIPS (2018)
23. Schwinn, L., Raab, R., Eskofier, B.: Towards rapid and robust adversarial training with one-step attacks. arXiv preprint arXiv:2002.10097 (2020)
24. Shafahi, A., Najibi, M., Ghiasi, A., Xu, Z., Dickerson, J., Davis, L.S., Goldstein, T., Studer, C., Taylor, G.: Adversarial Training for Free! In: NeurIPS. pp. 3353–3364 (2019)
25. Shalev-Shwartz, S., Ben-David, S.: Understanding machine learning: From theory to algorithms. Cambridge university press (2014)

26. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. In: International Conference on Learning Representations (2014)
27. Terjék, D.: Adversarial Lipschitz regularization. arXiv preprint arXiv:1907.05681 (2019)
28. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms (2017)
29. Zou, D., Balan, R., Singh, M.: On Lipschitz bounds of general convolutional neural networks. IEEE Transactions on Information Theory **66**(3), 1738–1759 (2019)

Print P4
A bregman learning framework for sparse neural networks

L. Bungert, T. Roith, D. Tenbrinck, and M. Burger
Journal of Machine Learning Research (2022)

Journal of Machine Learning Research 23 (2022) 1-43 Submitted 5/21; Revised 2/22; Published 7/22

A Bregman Learning Framework for Sparse Neural Networks

Leon Bungert LEON.BUNGERT@HCM.UNI-BONN.DE
Hausdorff Center for Mathematics
University of Bonn
Endenicher Allee 62, Villa Maria, 53115 Bonn, Germany

Tim Roith TIM.ROITH@FAU.DE
Department of Mathematics
Friedrich–Alexander University Erlangen–Nürnberg
Cauerstraße 11, 91058 Erlangen, Germany

Daniel Tenbrinck DANIEL.TENBRINCK@FAU.DE
Department of Mathematics
Friedrich–Alexander University Erlangen–Nürnberg
Cauerstraße 11, 91058 Erlangen, Germany

Martin Burger MARTIN.BURGER@FAU.DE
Department of Mathematics
Friedrich–Alexander University Erlangen–Nürnberg
Cauerstraße 11, 91058 Erlangen, Germany

Editor: Silvia Villa

Abstract

We propose a learning framework based on stochastic Bregman iterations, also known as mirror descent, to train sparse neural networks with an inverse scale space approach. We derive a baseline algorithm called *LinBreg*, an accelerated version using momentum, and *AdaBreg*, which is a Bregmanized generalization of the *Adam* algorithm. In contrast to established methods for sparse training the proposed family of algorithms constitutes a regrowth strategy for neural networks that is solely optimization-based without additional heuristics. Our Bregman learning framework starts the training with very few initial parameters, successively adding only significant ones to obtain a sparse and expressive network. The proposed approach is extremely easy and efficient, yet supported by the rich mathematical theory of inverse scale space methods. We derive a statistically profound sparse parameter initialization strategy and provide a rigorous stochastic convergence analysis of the loss decay and additional convergence proofs in the convex regime. Using only 3.4% of the parameters of ResNet-18 we achieve 90.2% test accuracy on CIFAR-10, compared to 93.6% using the dense network. Our algorithm also unveils an autoencoder architecture for a denoising task. The proposed framework also has a huge potential for integrating sparse backpropagation and resource-friendly training. Code is available at `https://github.com/TimRoith/BregmanLearning`.

Keywords: Bregman Iterations, Mirror Descent, Sparse Neural Networks, Sparsity, Inverse Scale Space, Optimization

1. Introduction

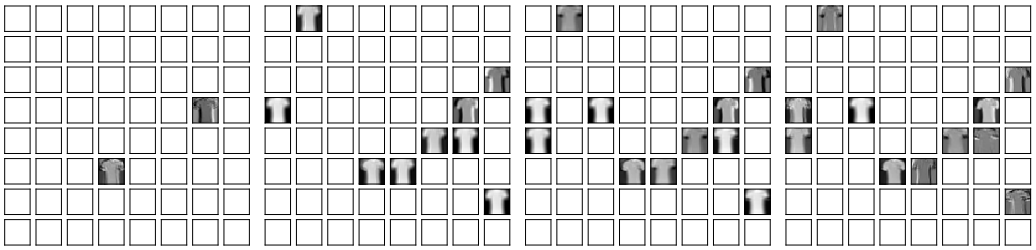
Large and deep neural networks have shown astonishing results in challenging applications, ranging from real-time image classification in autonomous driving, over assisted diagnoses in healthcare, to surpassing human intelligence in highly complex games (Amato et al., 2013; Rawat and Wang, 2017; Silver et al., 2016). The main drawback of many of these architectures is that they require huge amounts of memory and can only be employed using specialised hardware, like GPGPUs and TPUs. This makes them inaccessible to normal users with only limited computational resources on their mobile devices or computers (Hoefler et al., 2021). Moreover, the carbon footprint of training large networks has become an issue of major concern recently (Dhar, 2020), hence calling for resource-efficient methods.

The success of large and deep neural networks is not surprising as it has been predicted by universal approximation theorems (Cybenko, 1989; Lu et al., 2017), promising a smaller error with increasing number of neurons and layers. Besides the increase in computational complexity, each neuron added to the network architecture also adds to the amount of free parameters and local optima of the loss.

Consequently, a significant branch of modern research aims for training "sparse neural networks", which has lead to different strategies, based on neglecting small parameters or such with little influence on the network output, see Hoefler et al. (2021) for an extensive review. Apart from computational and resource efficiency, sparse training also sheds light on neural architecture design and might answer the question why certain architectures work better than others.

A popular approach for generating sparse neural networks are pruning techniques (LeCun et al., 1990; Han et al., 2015), which have been developed to sparsify a dense neural network during or after training by dropping dispensable neurons and connections. Another approach, which is based on the classical Lasso method from compressed sensing (Tibshirani, 1996), incorporates ℓ_1 regularization into the training problem, acting as convex relaxation of sparsity-enforcing ℓ_0 regularization. These endeavours are further supported by the recently stated "lottery ticket hypothesis" (Frankle and Carbin, 2018), which postulates that dense, feed-forward networks contain sub-networks with less neurons that, if trained in isolation, can achieve the same test accuracy as the original network.

An even more intriguing idea is "grow-and-prune" (Dai et al., 2019), which starts with a sparse network and augments it during training, while keeping it as sparse as possible. To this end new neurons are added, e.g., by splitting overloaded neurons into new specimen or using gradient-based indicators, while insignificant parameters are set to zero by thresholding.



(a) Iteration 0 (b) Iteration 5 (c) Iteration 20 (d) Iteration 100

Figure 1: Inverse scale space character of LinBreg visualized through feature maps of a convolutional neural network. Descriptive kernels are gradually added in the training process.

Many of the established methods in the literature are bound to specific architectures, e.g., fully-connected feedforward layers (Castellano et al., 1997; Liu et al., 2021). In this paper we propose a more conceptual and optimization-based approach. The idea is to mathematically follow the intuition of starting with very few parameters and adding only necessary ones in an inverse scale space manner, see Figure 1 for an illustration of our algorithm on a convolutional neural network. For this sake we propose a Bregman learning framework utilizing linearized Bregman iterations—originally introduced for compressed sensing by Yin et al. (2008)—for training sparse neural networks.

Our **main contributions** are the following:

- We derive an extremely simple and efficient algorithm for training sparse neural networks, called *LinBreg*.
- We also propose a momentum-based acceleration and *AdaBreg*, which utilizes the Adam algorithm (Kingma and Ba, 2014).
- We perform a rigorous stochastic convergence analysis of LinBreg for strongly convex losses, in infinite dimensions, and without any smoothness assumptions on the sparsity regularizer.
- We propose a sparse initialization strategy for the network parameters.
- We show that our algorithms are effective for training sparse neural networks and show their potential for architecture design by unveiling a denoising autoencoder.

The structure of this paper is as follows: In Section 1.1 we explain our baseline algorithm *LinBreg* in a nutshell and in Section 1.2 we discuss related work. Sections 1.3 and 1.4 clarify notation and collect preliminaries on neural networks and convex analysis, the latter being important for the derivation and analysis of our algorithms. In Section 2 we explain how Bregman iterations can be incorporated into the training of sparse neural networks, derive and discuss variants of the proposed

Algorithm 1: *LinBreg*, an inverse scale space algorithm for training sparse neural networks by successively adding weights whilst minimizing the loss. The functional J is sparsity promoting, e.g., the ℓ_1-norm.

default: $\delta = 1$
$\theta \leftarrow$ Section 4.1, $\quad v \leftarrow \partial J(\theta) + \frac{1}{\delta}\theta$ `// initialize`
for epoch $e = 1$ **to** E **do**
  for minibatch $B \subset \mathcal{T}$ **do**
    $g \leftarrow \nabla L(\theta; B)$ `// Backpropagation`
    $v \leftarrow v - \tau g$ `// Gradient step`
    $\theta \leftarrow \operatorname{prox}_{\delta J}(\delta v)$ `// Regularization`

Bregman learning algorithm, including accelerations using momentum and Adam. We perform a mathematical analysis for stochastic linearized Bregman iterations in Section 3 and discuss conditions for convergence of the loss function and the parameters. In Section 4 we first discuss our statistical sparse initialization strategy and then evaluate our algorithms on benchmark data sets (MNIST, Fashion-MNIST, CIFAR-10) using feedforward, convolutional, and residual neural networks.

1.1 The Bregman Training Algorithm in a Nutshell

Algorithm 1 states our baseline algorithm *LinBreg* for training sparse neural networks with an inverse scale space approach. Mathematical tools and derivations of LinBreg and its variants *LinBreg with momentum* (Algorithm 2) and *AdaBreg* (Algorithm 3), a generalization of *Adam* (Kingma and Ba, 2014), are presented in Section 2; a convergence analysis is provided in Section 3.

LinBreg can easily be applied to any neural network architecture f_θ, parametrized with parameters $\theta \in \Theta$, using a set of training data $\mathcal{T}$, and an empirical loss function $L(\theta; B)$, where $B \subset \mathcal{T}$ is a batch of training data. LinBreg's most important ingredient is a sparsity enforcing functional $J : \Theta \to (-\infty, \infty]$, which acts on groups of network parameters as, for instance, convolutional kernels, weight matrices, biases, etc. Following Scardapane et al. (2017) and denoting the collection of all parameter groups for which sparsity is desired by $\mathcal{G}$, two possible regularizers which induce sparsity or group sparsity, respectively, can be defined as

$$J(\theta) = \lambda \sum_{\mathbf{g} \in \mathcal{G}} \|\mathbf{g}\|_1, \qquad \text{the } \ell_1\text{-norm}, \tag{1.1}$$

$$J(\theta) = \lambda \sum_{\mathbf{g} \in \mathcal{G}} \sqrt{n_{\mathbf{g}}}\, \|\mathbf{g}\|_2, \qquad \text{the group } \ell_{1,2}\text{-norm}. \tag{1.2}$$

Here $\lambda > 0$ is a parameter controlling the regularization strength, $n_{\mathbf{g}}$ denotes the number of elements in $\mathbf{g}$, and the factor $\sqrt{n_{\mathbf{g}}}$ ensures a uniform weighting of all groups (Scardapane et al., 2017).

LinBreg uses two variables v and θ, coupled through the condition that $v \in \partial J_\delta(\theta)$ is a subgradient of the *elastic net regularization* $J_\delta(\theta) := J(\theta) + \frac{1}{2\delta}\|\theta\|^2$ introduced by Zou and Hastie (2005) (see Sections 1.3 and 1.4 for definitions). The algorithm successively updates v with gradients of the loss and recovers sparse parameters θ by applying a proximal operator. For instance, if $J(\theta) = \lambda\|\theta\|_1$ equals the ℓ_1-norm, the proximal operator in Algorithm 1 coincides with the soft shrinkage operator:

$$\operatorname{prox}_{\delta J}(\delta v) = \delta \operatorname{shrink}(v; \lambda) := \delta \operatorname{sign}(v) \max(|v| - \lambda, 0). \tag{1.3}$$

In this case only those parameters θ will be non-zero whose subgradients v have magnitude larger than the regularization parameter λ. Furthermore, $\delta > 0$ only steers the magnitude of the resulting weights and not their support. Furthermore, if $J(\theta) = 0$ then $\operatorname{prox}_{\delta J}(\delta v) = \delta v$ and therefore Algorithm 1 coincides with stochastic gradient descent (SGD) with learning rate $\delta\tau$. These two observations explain our default choice of $\delta = 1$.

In general, the proximal operators of the regularizers above can be efficiently evaluated since they admit similar closed form solutions based on soft thresholding. Hence, the computational complexity of LinBreg is dominated by the backpropagation and coincides with the complexity of vanilla stochastic gradient descent. However, note that our framework has great potential for complexity reduction via sparse backpropagation methods, cf. Dettmers and Zettlemoyer (2019).

The special feature which tells LinBreg apart from standard sparsity regularization (Louizos et al., 2017; Scardapane et al., 2017; Srinivas et al., 2017) or pruning (LeCun et al., 1990; Han et al., 2015) is its inverse scale space character. LinBreg is derived based on Bregman iterations, originally developed for scale space approaches in imaging (Osher et al., 2005; Burger et al., 2006; Yin et al., 2008; Cai et al., 2009b,a; Zhang et al., 2011). Instead of removing weights from a dense trained network, it starts from a very sparse initial set of parameters (see Section 4.1) and successively adds non-zero parameters whilst minimizing the loss.

1.2 Related Work

Dense-to-Sparse Training A well-established approach for training sparse neural network consists in solving the regularized empirical risk minimization

$$\min_{\theta \in \Theta} L(\theta; B) + J(\theta), \tag{1.4}$$

where J is a (sparsity-promoting) non-smooth regularization functional. If J equals the ℓ_1-norm this is referred to as *Lasso* (Tibshirani, 1996) and was extended to *Group Lasso* for neural networks by Scardapane et al. (2017) by using group norms.

We refer to de Dios and Bruna (2020) for a mean-field analysis of this approach. The regularized risk minimization (1.4) is a special case of Dense-to-Sparse training. Even if the network parameters are initialized sparsely, any optimization method for (1.4) will instantaneously generate dense weights, which are subsequently sparsified. A different strategy for Dense-to-Sparse training is *pruning* (LeCun et al., 1990; Han et al., 2015), see also Zhu and Gupta (2017), which first trains a network and then removes parameters to create sparse weights. This procedure can also be applied alternatingly, which is referred to as *iterative pruning* (Castellano et al., 1997). The weight removal can be achieved based on different criteria, e.g., their magnitude or their influence on the network output.

Sparse-to-Sparse Training In contrast, Sparse-to-Sparse training aims to grow a neural network starting from a sparse initialization until it is sufficiently accurate. This is also the paradigm of our LinBreg algorithm, generating an inverse sparsity scale space. Other approaches from literature are grow-and-prune strategies (Mocanu et al., 2018; Dettmers and Zettlemoyer, 2019; Dai et al., 2019; Liu et al., 2021; Evci et al., 2020) which, starting from sparse networks, successively add and remove neurons or connections while training the networks.

Proximal Gradient Descent A related approach to LinBreg is *proximal gradient descent* (ProxGD) for optimizing the regularized empirical risk minimization (1.4), which is an inherently non-smooth optimization problem due to the presence of the ℓ_1-norm-type functional J. Therefore, proximal gradient descent alternates between a gradient step of the loss with a proximal step of the regularization:

$$g \leftarrow \nabla L(\theta; B) \tag{1.5a}$$

$$\theta \leftarrow \theta - \tau g \tag{1.5b}$$

$$\theta \leftarrow \operatorname{prox}_{\tau J}(\theta). \tag{1.5c}$$

Applications for training neural networks and convergence analysis of this algorithm and its variants can be found, e.g., in Nitanda (2014); Rosasco et al. (2020); Reddi et al. (2016); Yang et al. (2019); Yun et al. (2020). It differs from Algorithm 1 by the lack of a subgradient variable and by using the learning rate τ within the proximal operator. These seemingly minor algorithmic differences cause major differences for the trained parameters. Indeed, the effect of J kicks in only after several iterations when the proximal operator has been applied sufficiently often to set some parameters to zero, as can be observed in Figure 2, Section 4.2. Furthermore, proximal gradient descent does not decrease the loss monotonously which we are able to prove for LinBreg.

Bregman Iterations Bregman iterations and in particular linearized Bregman iterations have been introduced and thoroughly analyzed for sparse regularization approaches in imaging and compressed sensing (see, e.g., Osher et al. (2005); Yin et al. (2008); Bachmayr and Burger (2009); Cai et al. (2009b,a); Yin (2010); Burger

et al. (2007, 2013)). More recent applications of Bregman type methods in the context of image restoration are Benfenati and Ruggiero (2013); Jia et al. (2016); Li et al. (2018). Linearized Bregman iterations for non-convex problems, which appear in machine learning and imaging applications like blind deblurring, have first been analyzed by Bachmayr and Burger (2009); Benning and Burger (2018); Benning et al. (2021). Benning and Burger (2018) also showed that linearized Bregman iterations for convex problems can be formulated as forward pass of a neural network. Benning et al. (2021) applied them for training neural networks with low-rank weight matrices, using nuclear norm regularization. Huang et al. (2016) suggested a split Bregman approach for training sparse neural networks and Fu et al. (2022) provided a deterministic convergence result along the lines of Benning et al. (2021). A recent analysis of Bregman stochastic gradient descent, which is the same as linearized Bregman iterations, however using strong regularity assumptions on the involved functions, is done by Dragomir et al. (2021).

Mirror Descent As it turns out, linearized Bregman iterations are largely known under yet another name: *mirror descent*. This method was first proposed by Nemirovskij and Yudin (1983) and related to Bregman distances by Beck and Teboulle (2003). Dragomir et al. (2021) present a literature overview of stochastic mirror descent. Some months after the release of the preprint version of the present article, D'Orazio et al. (2021) presented a convergence analysis of stochastic mirror descent a.k.a. Bregman iterations, using a weaker bounded variance condition for the stochastic gradients, albeit working in a smooth setting. In contrast, our analysis does not require any smoothness of J.

1.3 Preliminaries on Neural Networks

We denote neural networks, which map from an input space $\mathcal{X}$ to an output space $\mathcal{Y}$ and have parameters in some parameter space Θ, by

$$f_\theta : \mathcal{X} \to \mathcal{Y}, \quad \theta \in \Theta. \tag{1.6}$$

In principle $\mathcal{X}$, $\mathcal{Y}$, and Θ can be infinite-dimensional and we only assume that Θ is a Hilbert space, equipped with an inner product $\langle \tilde{\theta}, \theta \rangle$ and associated norm $\|\theta\| = \sqrt{\langle \theta, \theta \rangle}$. Given a set of training pairs $\mathcal{T} \subset \mathcal{X} \times \mathcal{Y}$ and a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ we denote the empirical loss associated to the training data by

$$\mathcal{L}(\theta) := \frac{1}{|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} \ell(f_\theta(x), y). \tag{1.7}$$

The empirical risk minimization approach to finding optimal parameters $\theta \in \Theta$ of the neural network f_θ then consists in solving

$$\min_{\theta\in\Theta} \mathcal{L}(\theta). \tag{1.8}$$

If one assumes that the training set $\mathcal{T}$ is sampled from some probability measure ρ on the product space $\mathcal{X} \times \mathcal{Y}$, the empirical risk minimization is an approximation of the infeasible population risk minimization

$$\min_{\theta \in \Theta} \int_{\mathcal{X} \times \mathcal{Y}} \ell(f_\theta(x), y) \, \mathrm{d}\rho(x, y). \tag{1.9}$$

One typically samples batches $B \subset \mathcal{T}$ from the training set and replaces $\mathcal{L}(\theta)$ by the empirical risk of the batch

$$L(\theta; B) := \frac{1}{|B|} \sum_{(x,y) \in B} \ell(f_\theta(x), y), \tag{1.10}$$

which is utilized in *stochastic* gradient descent methods.

For a feed-forward architecture with $L \in \mathbb{N}$ layers of sizes n_l we split the variable θ into weights and biases $W^l \in \mathbb{R}^{n_l, n_{l-1}}$, $b^l \in \mathbb{R}^{n_l}$ for $l \in \{1, \dots, L\}$. In this case we have

$$f_\theta(x) = \Phi^L \circ \cdots \circ \Phi^1(x), \tag{1.11}$$

where the l-th layer for $l \in \{1, \dots, L\}$ is given by

$$\Phi^l(z) := \sigma^l(W^l z + b^l). \tag{1.12}$$

Here σ^l denote activation functions, as for instance ReLU, TanH, Sigmoid, etc., (Goodfellow et al., 2016). In this case, sparsity promoting regularizers are the ℓ_1-norm or the group $\ell_{1,2}$-norm

$$J(\theta) = \lambda \sum_{l=1}^{L} \|W^l\|_{1,1}, \tag{1.13}$$

$$J(\theta) = \lambda \sum_{l=1}^{L} \sqrt{n_{l-1}} \, \|W^l\|_{1,2}, \tag{1.14}$$

which induce sparsity of the weight matrices and of the non-zero rows of weight matrices, respectively. Here the scaling $\sqrt{n_{l-1}}$ weighs the influence of the l-th layer based on the number of incoming neurons.

1.4 Preliminaries on Convex Analysis

In this section we introduce some essential concepts from convex analysis which we need to derive LinBreg and its variants and in order to make our argumentation more self-contained. For an overview of these topics we refer to Benning and Burger (2018); Rockafellar (1997); Bauschke and Combettes (2011). A functional $J : \Theta \to (-\infty, \infty]$ on a Hilbert space Θ is called convex if

$$J(\lambda \overline{\theta} + (1 - \lambda)\theta) \leq \lambda J(\overline{\theta}) + (1 - \lambda) J(\theta), \quad \forall \lambda \in [0, 1], \, \overline{\theta}, \theta \in \Theta. \tag{1.15}$$

We define the effective domain of J as $\mathrm{dom}(J) := \{\theta \in \Theta : J(\theta) \neq \infty\}$ and call J proper if $\mathrm{dom}(J) \neq \emptyset$. Furthermore, J is called lower semicontinuous if $J(u) \leq \liminf_{n\to\infty} J(u_n)$ holds for all sequences $(u_n)_{n\in\mathbb{N}} \subset \Theta$ converging to u. First, we define the subdifferential of a convex and proper functional $J : \Theta \to (-\infty, \infty]$ at a point $\theta \in \Theta$ as

$$\partial J(\theta) := \left\{p \in \Theta : J(\theta) + \langle p, \overline{\theta} - \theta\rangle \leq J(\overline{\theta}),\ \forall \overline{\theta} \in \Theta\right\}. \tag{1.16}$$

The subdifferential is a non-smooth generalization of the derivative and coincides with the classical gradient (or Fréchet derivative) if J is differentiable. We denote $\mathrm{dom}(\partial J) := \{\theta \in \Theta : \partial J(\theta) \neq \emptyset\}$ and observe that $\mathrm{dom}(\partial J) \subset \mathrm{dom}(J)$.

Next, we define the Bregman distance of two points $\theta \in \mathrm{dom}(\partial J), \overline{\theta} \in \Theta$ with respect to a convex and proper functional $J : \Theta \to (-\infty, \infty]$ as

$$D_J^p(\overline{\theta}, \theta) := J(\overline{\theta}) - J(\theta) - \langle p, \overline{\theta} - \theta\rangle, \quad p \in \partial J(\theta). \tag{1.17}$$

The Bregman distance can be interpreted as the distance between the linearization of J at θ and its graph and hence somewhat measures the degree of linearity of the functional. Note furthermore that the Bregman distance (1.17) is neither definite, symmetric nor fulfills the triangle inequality, hence it is not a metric. However, it fulfills the two distance axioms

$$D_J^p(\overline{\theta}, \theta) \geq 0, \quad D_J^p(\theta, \theta) = 0, \quad \forall \overline{\theta} \in \Theta, \theta \in \mathrm{dom}(\partial J). \tag{1.18}$$

By summing up two Bregman distances, one can also define the symmetric Bregman distance with respect to $p \in \partial J(\theta)$ and $\overline{p} \in \partial J(\overline{\theta})$ as

$$D_J^{\mathrm{sym}}(\overline{\theta}, \theta) := D_J^p(\overline{\theta}, \theta) + D_J^{\overline{p}}(\theta, \overline{\theta}). \tag{1.19}$$

Here, we suppress the dependency on p and $\overline{p}$ to simplify the notation.

Last, we define the proximal operator of a convex, proper and lower semicontinuous functional $J : \Theta \to (-\infty, \infty]$ as

$$\mathrm{prox}_J(\overline{\theta}) := \operatorname*{arg\,min}_{\theta\in\Theta} \frac{1}{2}\|\theta - \overline{\theta}\|^2 + J(\theta). \tag{1.20}$$

Proximal operators are a key concept in non-smooth optimization since they can be used to replace gradient descent steps of non-smooth functionals, as done for instance in proximal gradient descent (1.5). Obviously, given some $\overline{\theta} \in \Theta$ the proximal operator outputs a new element $\theta \in \Theta$ which has a smaller value of J whilst being close to the previous element $\overline{\theta}$.

2. Bregmanized training of Neural Networks

In this section we first give a short overview of inverse scale space flows which are the time-continuous analogue of our algorithms. Subsequently, we derive *LinBreg*

(Algorithm 1) by passing from Bregman iterations to linearized Bregman iterations, which we then reformulate in a very easy and compact form. We then derive *LinBreg with momentum* (Algorithm 2) by discretizing a second-order in time inverse scale space flow and propose *AdaBreg* (Algorithm 3) as a generalization of the popular Adam algorithm (Kingma and Ba, 2014).

2.1 Inverse Scale Space Flows (with Momentum)

In the following we discuss the inverse scale space flow, which arises as gradient flow of a loss functional $\mathcal{L}$ with respect to the Bregman distance (1.17). In particular, it couples the minimization of $\mathcal{L}$ with a simultaneous regularization through J. To give meaning to this, one considers the following implicit Euler scheme

$$\theta^{(k+1)} = \operatorname*{arg\,min}_{\theta\in\Theta} D_J^{p^{(k)}}(\theta, \theta^{(k)}) + \tau^{(k)}\mathcal{L}(\theta), \tag{2.1a}$$

$$p^{(k+1)} = p^{(k)} - \tau^{(k)}\nabla\mathcal{L}(\theta^{(k+1)}) \in \partial J(\theta^{(k+1)}) \tag{2.1b}$$

which is know as *Bregman iteration*. Here, $\theta^{(k)}$ is the previous iterate with subgradient $p^{(k)} \in \partial J(\theta^{(k)})$, and $\tau^{(k)} > 0$ is a sequence of time steps. Note that the subgradient update in the second line of (2.1) coincides with the optimality conditions of the first line.

The time-continuous limit of (2.1) as $\tau^{(k)} \to 0$ is the inverse scale space flow

$$\begin{cases} \dot{p}_t = -\nabla\mathcal{L}(\theta_t), \\ p_t \in \partial J(\theta_t), \end{cases} \tag{2.2}$$

see Burger et al. (2006, 2007) for a rigorous derivation in the context of image denoising.

If $J(\theta) = \frac{1}{2}\|\theta\|^2$ then $\partial J(\theta) = \theta$ and (2.2) coincides with the standard gradient flow

$$\dot{\theta}_t = -\nabla\mathcal{L}(\theta_t). \tag{2.3}$$

Hence, the inverse scale space is a proper generalization of the gradient flow and allows for regularizing the path along which the loss is minimized using J (see Benning et al. (2021)). For strictly convex loss functions this might seem pointless since they have a unique minimum anyways, however, for merely convex or even non-convex losses the inverse scale space allows to 'select' (local) minima with desirable properties.

In this paper, we also propose an inertial version of (2.2) which depends on an inertial parameter $\gamma \geq 0$ and takes the form

$$\begin{cases} \gamma\ddot{p}_t + \dot{p}_t = -\nabla\mathcal{L}(\theta_t), \\ p_t \in \partial J(\theta_t). \end{cases} \tag{2.4}$$

One can introduce the momentum variable $m_t := \dot{p}_t$ which solves the differential equation

$$\gamma \dot{m}_t + m_t = -\nabla \mathcal{L}(\theta_t).$$

If one assumes $m_0 = 0$, this equation has the explicit solution

$$m_t = -\int_0^t \exp\left(\frac{s-t}{\gamma}\right) \nabla \mathcal{L}(\theta_s) \mathrm{d}s$$

and hence the second-order in time equation (2.4) is equivalent to the gradient memory inverse scale space flow

$$\begin{cases} \dot{p}_t = -\int_0^t \exp\left(\frac{s-t}{\gamma}\right) \nabla \mathcal{L}(\theta_s) \mathrm{d}s, \\ p_t \in \partial J(\theta_t). \end{cases} \tag{2.5}$$

For a nice overview over the derivation of gradient flows with momentum we refer to Orvieto et al. (2020).

2.2 From Bregman to Linearized Bregman Iterations

The starting point for the derivation of Algorithm 1 is the Bregman iteration (2.1), which is the time discretization of the inverse scale space flow (2.2).

Since the iterations (2.1) require the minimization of the loss in every iteration they are not feasible for large-scale neural networks. Therefore, we consider linearized Bregman iterations (Cai et al., 2009b), which linearize the loss function by

$$\mathcal{L}(\theta) \approx \mathcal{L}(\theta^{(k)}) + \left\langle g^{(k)}, \theta - \theta^{(k)} \right\rangle, \quad g^{(k)} := \nabla \mathcal{L}(\theta^{(k)}),$$

and replace the energy J with the strongly convex elastic-net regularization

$$J_\delta(\theta) := J(\theta) + \frac{1}{2\delta}\|\theta\|^2, \quad \delta \in (0, \infty). \tag{2.6}$$

Omitting all terms which do not depend on θ, the first line of (2.1) then becomes

$$\begin{aligned} \theta^{(k+1)} &= \operatorname*{arg\,min}_{\theta \in \Theta} \langle \tau^{(k)} g^{(k)}, \theta \rangle + J_\delta(\theta) - \langle v^{(k)}, \theta \rangle \\ &= \operatorname*{arg\,min}_{\theta \in \Theta} \langle \tau^{(k)} g^{(k)}, \theta \rangle + J(\theta) + \frac{1}{2\delta}\|\theta\|^2 - \langle v^{(k)}, \theta \rangle \\ &= \operatorname*{arg\,min}_{\theta \in \Theta} \frac{1}{2\delta}\|\theta - \delta\left(v^{(k)} - \tau^{(k)} g^{(k)}\right)\|^2 + J(\theta) \\ &= \operatorname{prox}_{\delta J}\left(\delta\left(v^{(k)} - \tau^{(k)} g^{(k)}\right)\right). \end{aligned} \tag{2.7}$$

The vector $v^{(k)} \in \partial J_\delta(\theta^{(k)})$ is a subgradient of the functional J_δ in the previous iterate. Using the update

$$v^{(k+1)} := v^{(k)} - \tau^{(k)} g^{(k)}$$

and combining this with (2.7) we obtain the compact update scheme

$$g^{(k)} = \nabla \mathcal{L}(\theta^{(k)}), \tag{2.8a}$$

$$v^{(k+1)} = v^{(k)} - \tau^{(k)} g^{(k)}, \tag{2.8b}$$

$$\theta^{(k+1)} = \operatorname{prox}_{\delta J}\left(\delta v^{(k+1)}\right). \tag{2.8c}$$

This iteration is an equivalent reformulation of linearized Bregman iterations (Yin et al., 2008; Cai et al., 2009b,a; Osher et al., 2010; Yin, 2010; Benning and Burger, 2018), which are usually expressed in a more complicated way. Furthermore, it coincides with the mirror descent algorithm (Beck and Teboulle, 2003) applied to the functional J_δ.

Note that Yin (2010) showed that for quadratic loss functions the elastic-net regularization parameter $\delta > 0$ has no influence on the asymptotics of linearized Bregman iterations, if chosen larger than a certain threshold. This effect is referred to as *exact regularization* (Friedlander and Tseng, 2008). The iteration scheme simply computes a gradient descent in the subgradient variable v and recovers the weights θ by evaluating the proximal operator of v. This makes it significantly cheaper than the original Bregman iterations (2.1) which require the minimization of the loss in every iteration.

Note that the last line in (2.8) is equivalent to $v^{(k+1)}$ satisfying the optimality condition

$$v^{(k+1)} \in \partial J_\delta(\theta^{(k+1)}). \tag{2.9}$$

In particular, by letting $\tau^{(k)} \to 0$ the iteration (2.8) can be viewed as explicit Euler discretization of the inverse scale space flow (2.2) of the elastic-net regularized functional J_δ:

$$\begin{cases} \dot{v}_t = -\nabla \mathcal{L}(\theta_t), \\ v_t \in \partial J_\delta(\theta_t). \end{cases} \tag{2.10}$$

By embedding (2.8) into a stochastic batch gradient descent framework we obtain LinBreg from Algorithm 1.

2.3 Linearized Bregman Iterations with Momentum

More generally we consider an inertial version of (2.10), which as in Section 2.1 is given by

$$\begin{cases} \gamma \ddot{v}_t + \dot{v}_t = -\nabla \mathcal{L}(\theta_t), \\ v_t \in \partial J_\delta(\theta_t). \end{cases} \tag{2.11}$$

We discretize this equation in time by approximating the time derivatives as

$$\ddot{v}_t \approx \frac{v^{(k+1)} - 2v^{(k)} + v^{(k-1)}}{(\tau^{(k)})^2},$$
$$\dot{v}_t \approx \frac{v^{(k+1)} - v^{(k)}}{\tau^{(k)}},$$

such that after some reformulation we obtain the iteration

$$v^{(k+1)} = \frac{\tau^{(k)} + 2\gamma}{\tau^{(k)} + \gamma} v^{(k)} - \frac{\gamma}{\tau^{(k)} + \gamma} v^{(k-1)} - \frac{(\tau^{(k)})^2}{\tau^{(k)} + \gamma} \nabla \mathcal{L}(\theta^{(k)}), \tag{2.12a}$$
$$\theta^{(k+1)} = \operatorname{prox}_{\delta J}(\delta v^{(k+1)}). \tag{2.12b}$$

To see the relation to the gradient memory equation (2.5), derived in Section 2.1, we rewrite (2.12), using the new variables

$$m^{(k+1)} := v^{(k)} - v^{(k+1)}, \quad \beta^{(k)} := \frac{\gamma}{\tau^{(k)} + \gamma} \in [0, 1). \tag{2.13}$$

Plugging this into (2.12) yields the iteration

$$m^{(k+1)} = \beta^{(k)} m^{(k)} + (1 - \beta^{(k)}) \tau^{(k)} \nabla \mathcal{L}(\theta^{(k)}), \tag{2.14a}$$
$$v^{(k+1)} = v^{(k)} - m^{(k+1)}, \tag{2.14b}$$
$$\theta^{(k+1)} = \operatorname{prox}_{\delta J}(\delta v^{(k+1)}). \tag{2.14c}$$

Similar to before, embedding this into a stochastic batch gradient descent framework we obtain LinBreg with momentum from Algorithm 2. Note that, contrary to stochastic gradient descent with momentum (Orvieto et al., 2020), the momentum acts on the subgradients v and not on the parameters θ.

Analogously, we propose AdaBreg in Algorithm 3, which is a generalization of the Adam algorithm (Kingma and Ba, 2014). Here, we also apply the bias correction steps on the subgradient v and reconstruct the parameters θ using the proximal operator of the regularizer J.

3. Analysis of Stochastic Linearized Bregman Iterations

In this section we provide a convergence analysis of stochastic linearized Bregman iterations. They are valid in a general sense and do not rely on $\mathcal{L}$ being an empirical loss or θ being weights of a neural network. Still we keep the notation fixed for clarity. All proofs can be found in the appendix.

We let $(\Omega, F, \mathbb{P})$ be a probability space, Θ be a Hilbert space, $\mathcal{L} : \Theta \to \mathbb{R}$ a Fréchet differentiable loss function, and $g : \Theta \times \Omega \to \Theta$ an unbiased estimator of $\nabla \mathcal{L}$, meaning $\mathbb{E}[g(\theta; \omega)] = \nabla \mathcal{L}(\theta)$ for all $\theta \in \Theta$. This and all other expected values are taken with respect to the random variable $\omega \in \Omega$ which, in our case, models

Algorithm 2: *LinBreg with Momentum*, an acceleration of LinBreg using momentum-based gradient memory.

default: $\delta = 1, \quad \beta = 0.9$

$\theta \leftarrow$ Section 4.1, $\quad v \leftarrow \partial J(\theta) + \frac{1}{\delta}\theta, \quad m \leftarrow 0$

`// initialize`

for *epoch* $e = 1$ **to** E **do**

  for *minibatch* $B \subset \mathcal{T}$ **do**

    $g \leftarrow \nabla L(\theta; B)$ `// Backpropagation`

    $m \leftarrow \beta\, m + (1-\beta)\tau\, g$ `// Momentum update`

    $v \leftarrow v - m$ `// Momentum step`

    $\theta \leftarrow \operatorname{prox}_{\delta J}(\delta v)$ `// Regularization`

Algorithm 3: *AdaBreg*, a Bregman version of the Adam algorithm which uses moment-based bias correction.

default: $\delta = 1$, $\beta_1 = 0.9, \quad \beta_2 = 0.999, \quad \epsilon = 10^{-8}$

$\theta \leftarrow$ Section 4.1, $\quad v \leftarrow \partial J(\theta) + \frac{1}{\delta}\theta, \quad m_1 \leftarrow 0, \quad m_2 \leftarrow 0$

`// initialize`

for *epoch* $e = 1$ **to** E **do**

  for *minibatch* $B \subset \mathcal{T}$ **do**

    $k \leftarrow k + 1$

    $g \leftarrow \nabla L(\theta; B)$ `// Backpropagation`

    $m_1 \leftarrow \beta_1\, m_1 + (1-\beta_1)\, g$ `// First moment estimate`

    $\hat{m}_1 \leftarrow m_1/(1-\beta_1^k)$ `// Bias correction`

    $m_2 \leftarrow \beta_2\, m_2 + (1-\beta_2)\, g^2$ `// Second raw moment estimate`

    $\hat{m}_2 \leftarrow m_2/(1-\beta_2^k)$ `// Bias correction`

    $v \leftarrow v - \tau\, \hat{m}_1/(\sqrt{\hat{m}_2} + \epsilon)$ `// Moment step`

    $\theta \leftarrow \operatorname{prox}_{\delta J}(\delta v)$ `// Regularization`

the randomly drawn batch of training in data in (1.10). We study the stochastic

linearized Bregman iterations

$$\text{draw } \omega^{(k)} \text{ from } \Omega \text{ using the law of } \mathbb{P}, \tag{3.1a}$$

$$g^{(k)} := g(\theta^{(k)}; \omega^{(k)}), \tag{3.1b}$$

$$v^{(k+1)} := v^{(k)} - \tau^{(k)} g^{(k)}, \tag{3.1c}$$

$$\theta^{(k+1)} := \operatorname{prox}_{\delta J}(\delta v^{(k+1)}). \tag{3.1d}$$

For our analysis we need some assumptions on the loss function $\mathcal{L}$ which are very common in the analysis of nonlinear optimization methods. Besides boundedness from below, we demand differentiability and Lipschitz-continuous gradients, which are standard assumptions in nonlinear optimization since they allow to prove *sufficient decrease* of the loss. We refer to Benning et al. (2021) for an example of a neural network the associated loss of which satisfies the following assumption.

Assumption 1 (Loss function) *We assume the following conditions on the loss function:*

- *The loss function $\mathcal{L}$ is bounded from below and without loss of generality we assume $\mathcal{L} \geq 0$.*
- *The function $\mathcal{L}$ is continuously differentiable.*
- *The gradient of the loss function $\theta \mapsto \nabla\mathcal{L}(\theta)$ is L-Lipschitz for $L \in (0, \infty)$:*

$$\|\nabla\mathcal{L}(\tilde{\theta}) - \nabla\mathcal{L}(\theta)\| \leq L\|\tilde{\theta} - \theta\|, \quad \forall \theta, \tilde{\theta} \in \Theta. \tag{3.2}$$

Remark 1 *Note that the Lipschitz continuity of the gradient in particular implies the classical estimate (Beck, 2017; Bauschke and Combettes, 2011)*

$$\mathcal{L}(\tilde{\theta}) \leq \mathcal{L}(\theta) + \langle \nabla\mathcal{L}(\theta), \tilde{\theta} - \theta \rangle + \frac{L}{2}\|\tilde{\theta} - \theta\|^2, \quad \forall \theta, \tilde{\theta} \in \Theta. \tag{3.3}$$

Furthermore, we need the following assumption, being of stochastic nature, which requires the gradient estimator to have uniformly bounded variance.

Assumption 2 (Bounded variance) *There exists a constant $\sigma > 0$ such that for any $\theta \in \Theta$ it holds*

$$\mathbb{E}\left[\|g(\theta; \omega) - \nabla\mathcal{L}(\theta)\|^2\right] \leq \sigma^2. \tag{3.4}$$

This is a standard assumption in the analysis of stochastic optimization methods and many authors actually demand the more restrictive condition of uniformly bounded stochastic gradients $\mathbb{E}\left[\|g(\theta; \omega)\|^2\right] \leq C$ for all $\theta \in \Theta$. Remarkably, both assumptions have been shown to be unnecessary for proving convergence of stochastic gradient descent of convex (Nguyen et al., 2018) and non-convex functions (Lei et al., 2019). Generalizing this to linearized Bregman iterations is however completely non-trivial which is why we stick to Assumption 2.

We also state our assumptions on the regularizer J, which are extremely mild.

Assumption 3 (Regularizer) *We assume that $J : \Theta \to (-\infty, \infty]$ is a convex, proper, and lower semicontinuous functional on the Hilbert space Θ.*

All other assumptions will be stated when they are needed.

3.1 Decay of the Loss Function

We first analyze how the iteration (3.1) decreases the loss $\mathcal{L}$. Such decrease properties of deterministic linearized Bregman iterations in a different formulation were already studied by Benning et al. (2021); Benning and Burger (2018). Note that for the loss decay we do not require any sort of convexity of $\mathcal{L}$ whatsoever, but merely Lipschitz continuity of the gradient, i.e., (3.3).

Theorem 2 (Loss decay) *Assume that Assumptions 1–3 hold true, let $\delta > 0$, and let the step sizes satisfy $\tau^{(k)} \leq \frac{2}{\delta L}$. Then there exist constants $c, C > 0$ such that for every $k \in \mathbb{N}$ the iterates of (3.1) satisfy*

$$\begin{aligned} \mathbb{E}\left[\mathcal{L}(\theta^{(k+1)})\right] + \frac{1}{\tau^{(k)}}\mathbb{E}\left[D_J^{\mathrm{sym}}(\theta^{(k+1)}, \theta^{(k)})\right] + \frac{C}{2\delta\tau^{(k)}}\mathbb{E}\left[\|\theta^{(k+1)} - \theta^{(k)}\|^2\right] \\ \leq \mathbb{E}\left[\mathcal{L}(\theta^{(k)})\right] + \tau^{(k)}\delta\frac{\sigma^2}{2c}, \end{aligned} \tag{3.5}$$

Corollary 3 (Summability) *Under the conditions of Theorem 2 and with the additional assumption that the step sizes are non-increasing and square-summable, meaning*

$$\tau^{(k+1)} \leq \tau^{(k)}, \quad \forall k \in \mathbb{N}, \qquad \sum_{k=0}^{\infty} (\tau^{(k)})^2 < \infty,$$

it holds

$$\sum_{k=0}^{\infty} \mathbb{E}\left[D_J^{\mathrm{sym}}(\theta^{(k+1)}, \theta^{(k)})\right] < \infty, \qquad \sum_{k=0}^{\infty} \mathbb{E}\left[\|\theta^{(k+1)} - \theta^{(k)}\|^2\right] < \infty.$$

Remark 4 (Deterministic case) *Note that in the deterministic setting with $\sigma = 0$ the statement of Theorem 2 coincides with Benning et al. (2021); Benning and Burger (2018). In particular, the loss decays monotonously and one gets stronger summability than in Corollary 3 since one does not have to multiply by $\tau^{(k)}$.*

3.2 Convergence of the Iterates

In this section we establish two convergence results for the iterates of the stochastic linearized Bregman iterations (3.1). According to common practice and for self-containedness we restrict ourselves to strongly convex losses. Obviously, our results remain true for non-convex losses if one assumes convexity around local

minima and applies our arguments locally. We note that one could also extend the deterministic convergence proof of Benning et al. (2021)—which is based on the Kurdyka–Łojasiewicz (KL) inequality and works for non-convex losses—to the stochastic setting. However, this is beyond the scope of this paper and conveys less intuition than our proofs.

For our first convergence result—asserting norm convergence of a subsequence—we need the condition on the loss function Assumption 1, the bounded variance condition from Assumption 2 and strong convexity of the loss $\mathcal{L}$:

Assumption 4 (Strong convexity) *The loss function $\theta \mapsto \mathcal{L}(\theta)$ is μ-strongly convex for $\mu \in (0, \infty)$, meaning*

$$\mathcal{L}(\tilde{\theta}) \geq \mathcal{L}(\theta) + \langle \nabla \mathcal{L}(\theta), \tilde{\theta} - \theta \rangle + \frac{\mu}{2} \|\tilde{\theta} - \theta\|^2, \quad \forall \theta, \tilde{\theta} \in \Theta. \tag{3.6}$$

Note that by virtue of (3.3) it holds $\mu \leq L$ if the loss satisfies Assumptions 1 and 4.

Our second convergence convergence result—asserting convergence in the Bregman distance of J_δ which is a stronger topology than norm convergence—requires a stricter convexity condition, tailored to the Bregman geometry.

Assumption 5 (Strong Bregman convexity) *The loss function $\theta \mapsto \mathcal{L}(\theta)$ satisfies*

$$\mathcal{L}(\tilde{\theta}) \geq \mathcal{L}(\theta) + \langle \nabla \mathcal{L}(\theta), \tilde{\theta} - \theta \rangle + \nu D^v_{J_\delta}(\tilde{\theta}, \theta), \quad \forall \theta, \tilde{\theta} \in \Theta,\ v \in \partial J_\delta(\theta), \tag{3.7}$$

where J_δ for $\delta > 0$ is defined in (2.6)*. In particular, $\mathcal{L}$ satisfies Assumption 4 with $\mu = \nu/\delta$.*

Remark 5 (The Bregman convexity assumption) *Assumption 5 seems to be quite restrictive, however, in finite dimensions it is locally equivalent to Assumption 4, as we argue in the following. Note that it suffices if the assumptions above are satisfied in a vicinity of the (local) minimum to which the algorithm converges. For proving convergence we will use Assumption 5 with $\tilde{\theta} = \theta^*$, a (local) minimum, and θ close to θ^*. Using Lemma 13 in the appendix the assumption can be rewritten as*

$$\mathcal{L}(\theta^*) \geq \mathcal{L}(\theta) + \langle \nabla \mathcal{L}(\theta), \theta^* - \theta \rangle + \frac{\nu}{2\delta} \|\theta^* - \theta\|^2 + \nu D^p_J(\theta^*, \theta), \quad p \in \partial J(\theta)$$

and we will argue that for θ close to θ^* the extra term vanishes, i.e. $D^p_J(\theta^*, \theta) = 0$. For this we have to show that p is not only a subgradient at θ but also at θ^*: If p is a subgradient of J at both points, i.e., $p \in \partial J(\theta) \cap \partial J(\theta^*)$ we obtain that their Bregman distance is zero. This can be seen using the definition of the Bregman distance (1.17)*:*

$$\left.\begin{aligned} D^p_J(\theta^*, \theta) \geq 0 \\ D^p_J(\theta, \theta^*) \geq 0 \end{aligned}\right\} \implies 0 \leq D^p_J(\theta^*, \theta) + D^p_J(\theta, \theta^*) = 0.$$

In finite dimensions and for $J(\theta) = \|\theta\|_1 = \sum_{i=1}^N |\theta_i|$ *equal to the* ℓ_1*-norm we can use that* $\langle p, \theta\rangle = J(\theta) = \sum_{i=1}^N \operatorname{sign}(\theta_i)\theta_i = \sum_{i=1}^N |\theta_i| = J(\theta)$ *and simplify the Bregman distance to*

$$D_J^p(\theta^*, \theta) = J(\theta^*) - \langle p, \theta^*\rangle = \sum_{i=1}^N |\theta_i^*| - \operatorname{sign}(\theta_i)\theta_i^* = \sum_{i=1}^N \theta_i^* \left(\operatorname{sign}(\theta_i^*) - \operatorname{sign}(\theta_i)\right).$$

Obviously, the terms in this sum where $\theta_i^* = 0$ *vanish anyways. Hence, the expression is zero whenever the non-zero entries of* θ *have the same sign as those of* θ^* *which is the case if* $\|\theta - \theta^*\|_\infty < \min\{|\theta_i^*| : i \in \{1, \dots, N\}, \theta_i^* \neq 0\}$*. Since all norms are equivalent in finite dimensions, one obtains*

$$D_J^p(\theta^*, \theta) = 0 \quad \text{for } \|\theta - \theta^*\| \text{ sufficiently small.}$$

Hence, Assumption 5 is locally implied by Assumption 4 if $J(\theta) = \|\theta\|_1$*.*

We would like to remark that—even under the weaker Assumption 4—one needs some coupling of the loss $\mathcal{L}$ and the regularization functional J in order to obtain convergence to a critical point. Benning et al. (2021) demand that a surrogate function involving both $\mathcal{L}$ and J admits the KL inequality and that the subgradients of J are bounded close to the minimum θ^* of the loss. Indeed, in our theory using Assumption 4 it suffices to demand that $J(\theta^*) < \infty$. This assumption is weaker than assuming bounded subgradients but is nevertheless necessary as the following example taken from Benning et al. (2021) shows.

Example 1 (Non-convergence to a critical point) *Let* $\mathcal{L}(\theta) = (\theta+1)^2$ *for* $\theta \in \mathbb{R}$ *and* $J(\theta) = \chi_{[0,\infty)}(\theta)$ *be the characteristic function of the positive axis. Then for any initialization the linearized Bregman iterations* (2.8) *converge to* $\theta = 0$ *which is no critical point of* $\mathcal{L}$*. On the other hand, the only critical point* $\theta^* = -1$ *clearly meets* $J(\theta^*) = \infty$*.*

Theorem 6 (Convergence in norm) *Assume that Assumptions 1–4 hold true and let* $\delta > 0$*. Furthermore, assume that the step sizes* $\tau^{(k)}$ *are such that for all* $k \in \mathbb{N}$*:*

$$\tau^{(k)} \leq \frac{\mu}{2\delta L^2}, \qquad \tau^{(k+1)} \leq \tau^{(k)}, \qquad \sum_{k=0}^{\infty} (\tau^{(k)})^2 < \infty, \qquad \sum_{k=0}^{\infty} \tau^{(k)} = \infty.$$

The function $\mathcal{L}$ *has a unique minimizer* θ^* *and if* $J(\theta^*) < \infty$ *the stochastic linearized Bregman iterations* (3.1) *satisfy the following:*

- *Letting* $d_k := \mathbb{E}\left[D_{J_\delta}^{v^{(k)}}(\theta^*, \theta^{(k)})\right]$ *it holds*

$$d_{k+1} - d_k + \frac{\mu}{4}\tau^{(k)}\mathbb{E}\left[\|\theta^* - \theta^{(k+1)}\|^2\right] \leq \frac{\sigma}{2}\left((\tau^{(k)})^2 + \mathbb{E}\left[\|\theta^{(k)} - \theta^{(k+1)}\|^2\right]\right). \tag{3.8}$$

- *The iterates possess a subsequence converging in the L^2-sense of random variables:*

$$\lim_{j\to\infty} \mathbb{E}\left[\|\theta^* - \theta^{(k_j)}\|^2\right] = 0. \tag{3.9}$$

Here, J_δ is defined as in (2.6).

Remark 7 (Choice of step sizes) *A possible step size which satisfies the conditions of Theorem 6 is given by $\tau^{(k)} = \frac{c}{(k+1)^p}$ where $0 < c < \frac{\mu}{\delta L^2}$ and $p \in (\frac{1}{2}, 1]$.*

Remark 8 (Deterministic case) *In the deterministic case $\sigma = 0$ inequality* (3.8) *even shows that the Bregman distances decrease along iterations. Furthermore, in this case it is not necessary that the step sizes are square-summable and non-increasing since the term on the right hand side does not have to be summed.*

In a finite dimensional setting and using ℓ_1-regularization one can even show convergence of the whole sequence of Bregman distances.

Remark 9 *With the help of Lemma 13 in the appendix, the quantity $D_{J_\delta}^{v^{(k)}}(\theta^*, \theta^{(k)})$ which appears in the decay estimate* (3.8) *can be simplified as follows:*

$$\begin{aligned} D_{J_\delta}^{v^{(k)}}(\theta^*, \theta^{(k)}) &= \frac{1}{2\delta}\|\theta^* - \theta^{(k)}\|^2 + D_J^{p^{(k)}}(\theta^*, \theta) \\ &= \frac{1}{2\delta}\|\theta^* - \theta^{(k)}\|^2 + J(\theta^*) - J(\theta^{(k)}) - \langle p^{(k)}, \theta^* - \theta^{(k)}\rangle, \end{aligned}$$

where $p^{(k)} := v^{(k)} - \frac{1}{\delta}\theta^{(k)} \in \partial J(\theta^{(k)})$. In the case that J is absolutely 1-homogeneous, e.g., if $J(\theta) = \|\theta\|_1$ equals the ℓ_1-norm, this simplifies to

$$D_{J_\delta}^{v^{(k)}}(\theta^*, \theta^{(k)}) = \frac{1}{2\delta}\|\theta^* - \theta^{(k)}\|^2 + J(\theta^*) - \langle p^{(k)}, \theta^*\rangle,$$

where we used that for absolutely 1-homogeneous functionals it holds $\langle p, \theta\rangle = J(\theta)$ for all $\theta \in \Theta$ and $p \in \partial J(\theta)$. Hence, it measures both the convergence of $\theta^{(k)}$ to θ^* in the norm and the convergence of the subgradients $p^{(k)} \in \partial J(\theta^{(k)})$ to a subgradient of J at θ^*.

Corollary 10 (Convergence in finite dimensions) *If the parameter space Θ is finite dimensional and J equals the ℓ_1-norm, under the conditions of Theorem 6 it even holds $\lim_{k\to\infty} d_k = 0$ which in particular implies $\lim_{k\to\infty} \mathbb{E}\left[\|\theta^* - \theta^{(k)}\|^2\right] = 0$.*

Our second convergence theorem asserts convergence in the Bregman distance and gives quantitative estimates under Assumption 5, which is a stricter assumption than Assumption 4 and relates the loss function $\mathcal{L}$ with the regularizer; cf. Dragomir et al. (2021) for a related approach working with C^2 functions. The theorem states that the Bregman distance to the minimizer of the loss can be made arbitrarily small using constant step sizes. For step sizes which go to zero and are not summable one obtains a quantitative convergence result.

Theorem 11 (Convergence in the Bregman distance) *Assume that Assumptions 1–3 and 5 hold true and let $\delta > 0$. The function $\mathcal{L}$ has a unique minimizer θ^* and if $J(\theta^*) < \infty$ the stochastic linearized Bregman iterations* (3.1) *satisfy the following:*

- *Letting $d_k := \mathbb{E}\left[D_{J_\delta}^{v^{(k)}}(\theta^*, \theta^{(k)})\right]$ it holds*

$$d_{k+1} \leq \left[1 - \tau^{(k)}\nu\left(1 - \tau^{(k)}\frac{2\delta^2 L^2}{\nu}\right)\right] d_k + \delta(\tau^{(k)})^2\sigma^2. \tag{3.10}$$

- *For any $\varepsilon > 0$ there exists $\tau > 0$ such that if $\tau^{(k)} = \tau$ for all $k \in \mathbb{N}$ then*

$$\limsup_{k\to\infty} d_k \leq \varepsilon. \tag{3.11}$$

- *If $\tau^{(k)}$ is such that*

$$\lim_{k\to\infty} \tau^{(k)} = 0 \quad \text{and} \quad \sum_{k=0}^{\infty} \tau^{(k)} = \infty \tag{3.12}$$

  then it holds

$$\lim_{k\to\infty} d_k = 0. \tag{3.13}$$

Here, J_δ is defined as in (2.6).

Corollary 12 (Convergence rate for diminishing step sizes) *The error recursion* (3.10) *coincides with the one for stochastic gradient descent. In particular, for step sizes of the form $\tau^{(k)} = \frac{c}{k}$ with a suitably small constant $c > 0$ one can prove with induction (Nemirovski et al., 2009) that $d_k = \frac{C}{k}$ for some $C > 0$.*

4. Numerical Experiments

In this section we perform an extensive evaluation of our algorithms focusing on different characteristics. First, we derive a sparse initialization strategy in Section 4.1, using similar statistical arguments as in the seminal works by Glorot and Bengio (2010); He et al. (2015). In Sections 4.2 and 4.3 we study the influence of hyperparameters and compare our family of algorithms with standard stochastic gradient descent (SGD) and the sparsity promoting proximal gradient descent method. In Sections 4.4 and 4.5 we demonstrate that our algorithms generate sparse and expressive networks for solving the classification task on Fashion-MNIST and CIFAR-10, for which we utilize state-of-the-art CNN and ResNet architectures. Finally, in Section 4.6 we show that, using row sparsity, our Bregman learning algorithm allows to discover a denoising autoencoder architecture, which shows the potential of the method for architecture design. Our code is available on `GitHub` at `https://github.com/TimRoith/BregmanLearning` and relies on `PyTorch` (Paszke et al., 2019).

4.1 Initialization

Parameter initialization for neural networks has a crucial influence on the training process, see Glorot and Bengio (2010). In order to tackle the problem of vanishing and exploding gradients, standard methods consider the variance of the weights at initialization (Glorot and Bengio, 2010; He et al., 2016), assuming that for each l the entries $W^l_{i,j}$ are i.i.d. with respect to a probability distribution. The intuition here is to preserve the variances over the forward and the backward pass similar to the variance of the respective input of the network, see Glorot and Bengio (2010, Sec. 4.2). If the distribution satisfies $\mathbb{E}\left[W^l\right] = 0$, this yields a condition of the form

$$\operatorname{Var}\left[W^l\right] = \alpha(n_l, n_{l-1}) \tag{4.1}$$

where the function α depends on the activation function. For anti-symmetric activation functions with $\sigma'(0) = 1$, as for instance a sigmoidal function, it was shown by Glorot and Bengio (2010) that

$$\alpha(n_l, n_{l-1}) = \frac{2}{n_l \cdot n_{l-1}}$$

while for ReLU He et al. (2016) suggest to use

$$\alpha(n_l, n_{l-1}) = \frac{2}{n_l} \quad \text{or} \quad \alpha(n_l, n_{l-1}) = \frac{2}{n_{l-1}}.$$

For our Bregman learning framework we have to adapt this argumentation, taking into account sparsity. For classical inverse scale space approaches and Bregman iterations of convex losses, as for instance used for image reconstruction (Osher et al., 2005) and compressed sensing (Yin et al., 2008; Burger et al., 2013; Osher et al., 2010; Cai et al., 2009b), one would initialize all parameters equal to zero. However, for neural networks this yields an unbreakable symmetry of the network parameters, which makes training impossible, see, e.g., Goodfellow et al. (2016, Ch. 6). Instead, we employ an established approach for sparse neural networks (see, e.g., Liu et al. (2021); Dettmers and Zettlemoyer (2019); Martens (2010)) which masks the initial parameters, i.e.,

$$W^l := \tilde{W}^l \odot M^l.$$

Here, the mask $M^l \in \mathbb{R}^{n_l, n_{l-1}}$ is chosen randomly such that each entry is distributed according to the Bernoulli distribution with a parameter $r \in [0, 1]$, i.e.,

$$M^l_{i,j} \sim \mathcal{B}(r).$$

The parameter r coincides with the expected percentage of non-zero weights

$$\mathrm{N}(W^l) := \frac{\|W^l\|_0}{n_l \cdot n_{l-1}} = 1 - \mathrm{S}(W^l), \tag{4.2}$$

where $\mathrm{S}(W^l)$ denotes the sparsity. In the following we derive a strategy to initialize $\tilde{W}^l$. Choosing $\tilde{W}^l$ and M^l independent and using $\mathbb{E}\left[\tilde{W}^l\right] = 0$ standard variance calculus implies

$$\begin{aligned}
\mathrm{Var}\left[W^l\right] &= \mathrm{Var}\left[\tilde{W}^l \odot M^l\right] \\
&= \mathbb{E}\left[M^l\right]^2 \ \mathrm{Var}\left[\tilde{W}^l\right] + \underbrace{\mathbb{E}\left[\tilde{W}^l\right]^2 \ \mathrm{Var}\left[M^l\right]}_{=0} + \mathrm{Var}\left[M^l\right] \ \mathrm{Var}\left[\tilde{W}^l\right] \\
&= \left(\mathbb{E}\left[M^l\right]^2 + \mathrm{Var}\left[M^l\right]\right) \mathrm{Var}\left[\tilde{W}^l\right] \\
&= \mathbb{E}\left[(M^l)^2\right] \ \mathrm{Var}\left[\tilde{W}^l\right] \\
&= r \ \mathrm{Var}\left[\tilde{W}^l\right]
\end{aligned}$$

and thus deriving from (4.1) we obtain the condition

$$\mathrm{Var}\left[\tilde{W}^l\right] = \frac{1}{r} \ \alpha(n_l, n_{l-1}). \tag{4.3}$$

Instead of having linear feedforward layers with corresponding weight matrices, the neural network architecture at hand might consist of other groups of parameters which one would like to keep sparse, e.g., using the group sparsity regularization (1.2). For example, in a convolutional neural network one might be interested in having only a few number of non-zero convolution kernels in order to obtain compact feature representations of the input. Similarly, for standard feedforward architectures sparsity of rows of the weight matrices yields compact networks with a small number of active neurons. In such cases one can apply the same arguments as above and initialize single elements $\mathbf{g} \in \mathcal{G}$ of a parameter group $\mathcal{G}$ as non-zero with probability $r \in [0, 1]$ and with respect to a variance condition similar to (4.3):

$$\mathbf{g} = \tilde{\mathbf{g}} \cdot m, \quad m \sim \mathcal{B}(r), \tag{4.4}$$

$$\mathrm{Var}\left[\tilde{\mathbf{g}}\right] = \frac{1}{r}\alpha(\mathbf{g}) \tag{4.5}$$

Note that these arguments are only valid in a linear regime around the initialization, see Glorot and Bengio (2010) for details. Hence, if one initializes with sparse parameters but optimizes with very weak regularization, e.g., by using vanilla SGD or choosing λ in (1.1) or (1.2) very small, the assumption is violated since the sparse initial parameters are rapidly filled during the first training steps.

The biases are initialized non-sparse and the precise strategy depends on the activation function. We would like to emphasize that initializing biases with zero is not a good idea in the context of sparse neural networks. In this case, the neuron activations would be equal for all "dead neurons" whose incoming weights are zero,

which would then yield an unbreakable symmetry. For ReLU we initialize biases with positive random values to ensure a flow of information and to break symmetry also for dead neurons, which is similar to the strategy proposed by Goodfellow et al. (2016, Ch. 6). For other activation functions σ which meet $\sigma(x) = 0$ if and only if $x = 0$, e.g., TanH, Sigmoid, or Leaky ReLU, biases can be initialized with random numbers of arbitrary sign.

4.2 Comparison of Algorithms

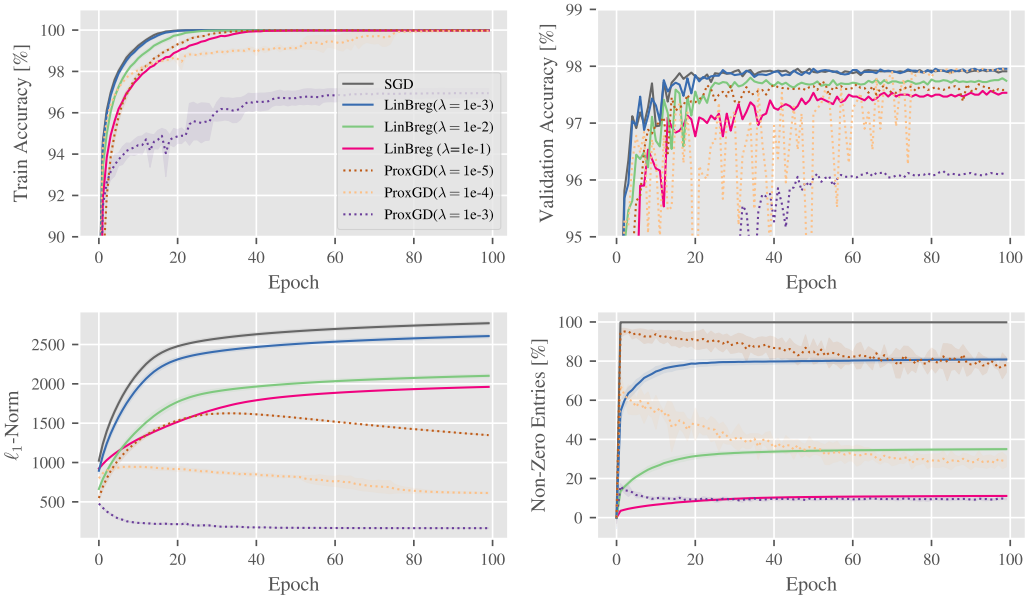


Figure 2: Comparison of vanilla SGD (black solid line), LinBreg (colored solid lines), and ProxGD (colored dotted lines) for different regularization parameters on MNIST. The curves show the averaged accuracies on train and validation sets, ℓ_1-norms, and non-zero entries over three runs. The shaded area visualizes the standard deviation.

We start by comparing the proposed LinBreg Algorithm 1 with vanilla stochastic gradient descent (SGD) without sparsity regularization and with the Lasso-based approach from Scardapane et al. (2017), for which we compute solutions to the sparsity-regularized risk minimization problem (1.4) using the proximal gradient descent algorithm (ProxGD) from (1.5).

We consider the classification task on the MNIST dataset (LeCun and Cortes, 2010) for studying the impact of the hyperparameters of these methods. The set consists of $60,000$ images of handwritten digits which we split into $55,000$ images used for the training and $5,000$ images used for a validation process during training.

We train a fully connected net with ReLU activations and two hidden layers (200 and 80 neurons), and use the ℓ_1-regularization from (1.13),

$$J(\theta) = \lambda \sum_{l=1}^{L} \|W^l\|_{1,1}$$

In Figure 2 we compare the training results of vanilla SGD, the proposed LinBreg, and the ProxGD algorithm. Following the strategy introduced in Section 4.1 we initialize the weights with 1% non-zero entries, i.e., $r = 0.01$. The learning rate is chosen as $\tau = 0.1$ and is multiplied by a factor of 0.5 whenever the validation accuracy stagnates. For a fair comparison the training is executed for three different fixed random seeds, and the plot visualizes mean and standard deviation of the three runs, respectively. We show the training and validation accuracies, the ℓ_1-norm, and the overall percentage of non-zero weights. Note that for the validation accuracies we do not show standard deviations for the sake of clarity.

While SGD without sparsity regularization instantaneously destroys sparsity, LinBreg exhibits the expected inverse scale space behaviour, where the number of non-zero weights gradually grows during training, and the train accuracy increases monotonously. This is suggested by Theorem 2, even though our experimental setup, in particular the non-smooth ReLU activation functions, is not covered by the theoretical framework which require at least L-smoothness of the loss functions.

In contrast, ProxGD shows no monotonicity of training accuracy or sparsity and the validation accuracies oscillate heavily. Instead, it adds a lot of non-zero weights in the beginning and then gradually reduces them. Obviously, the regularized empirical risk minimization (1.4) implies a trade-off between training accuracy and sparsity which depends on λ. For LinBreg this trade-off is neither predicted by theory nor observed numerically. Here, the regularization parameter only induces a trade-off between *validation* accuracy and sparsity, which is to be expected.

LinBreg (blue curves) can generate networks whose validation accuracy equals the one of a full network and use only 80% of the weights. For the largest regularization parameter (magenta curves) LinBreg uses only 10% of the weights and still does not drop more than half a percentage point in validation accuracy.

4.3 Accelerated Bregman Algorithms

We use the same setup as in the previous section to compare LinBreg with its momentum-based acceleration and AdaBreg (see Algorithms 1–3). Using the regularization parameter $\lambda = 10^{-1}$ from the previous section (see the magenta curves), Figure 3 shows the validation accuracy of the different networks trained with LinBreg, LinBreg with momentum, and AdaBreg. For comparison we visualize again the results of vanilla SGD as gray curve. It is obvious that all three proposed algorithms generate very accurate networks using approximately 10% of the weights. As expected, the accelerated versions increase both the validation accuracy and the number of non-zero parameters faster than the baseline algorithm LinBreg. While after

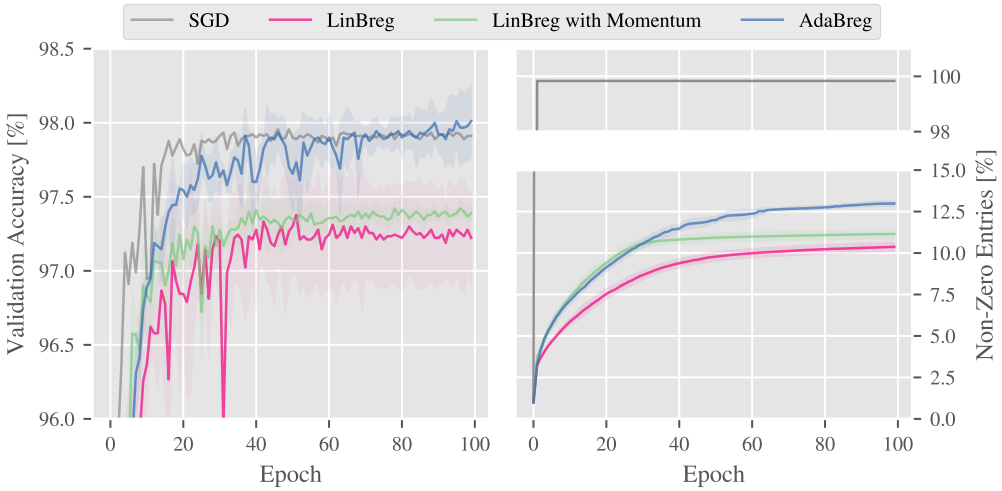


Figure 3: Comparison of LinBreg (with momentum), AdaBreg, and vanilla SGD. The networks generated by AdaBreg are sparse and generalize better.

100 epochs LinBreg and its momentum version have slightly lower validation accuracies than the non-sparse networks generated by SGD, AdaBreg outperforms the other algorithms including SGD in terms of validation accuracy while maintaining a high degree of sparsity.

4.4 Sparsity for Convolutional Neural Networks (CNNs)

In this example we apply our algorithms to a convolutional neural network of the form

$$5 \times 5 \text{ conv, } 64 \overset{\text{Maxpool}/2}{\longrightarrow} 5 \times 5 \text{ conv, } 64 \overset{\text{Maxpool}/2}{\longrightarrow} \text{fc } 1024 \longrightarrow \text{fc } 128 \longrightarrow \text{fc } 10$$

with ReLU activations to solve the classification task on Fashion-MNIST. We run experiments both for sparsity regularization utilizing the ℓ_1-norm (1.1) and for a combination of the ℓ_1-norm on the linear layers and the group $\ell_{1,2}$-norm (1.2) on the convolutional kernels. This way, we aim to obtain compressed network architectures with only few active convolutional filters.

The inverse scale space character of our algorithms is visualized in Figure 1 from the beginning of the paper which shows the 64 feature maps of an input image, generated by the first convolutional layer of the network after 0, 5, 20, and 100 epochs of LinBreg with group sparsity. One can observe that gradually more kernels are added until iteration 20, from where on the number of kernels stays fixed and the kernels themselves are optimized.

Tables 1 and 2 shows the test and training accuracies as well as sparsity levels. For plain sparsity regularization we only show the total sparsity level of all network

parameters whereas for the group sparsity regularization we show the sparsity of the linear layers and the relative number of non-zero convolutional kernels.

A convolutional layer for a input $z \in \mathbb{R}^{c_{l-1},n_{l-1},m_{l-1}}$ is given as

$$\Phi_j^l(z) = b_j + \sum_{i=1}^{c_{l-1}} K_{i,j}^l * z_{i,\bullet},$$

where $K_{i,j}^l \in \mathbb{R}^{k,k}$ denote kernel matrices with corresponding biases $b_j \in \mathbb{R}^{n_l,m_l}$ for in-channels $i \in \{1,\dots,c_{l-1}\}$ and out-channels $j \in \{1,\dots,c_l\}$. Therefore, we denote by

$$\mathrm{N}_{\mathrm{conv}} := \frac{\sum_{l\in I_{\mathrm{conv}}} \#\{K_{i,j}^l : K_{i,j}^l \neq \mathbf{0}\}}{\sum_{l\in I_{\mathrm{conv}}} c_l \cdot c_{l-1}}$$

the percentage of non-zero kernels of the whole net where I_{conv} denotes the index set of the convolutional layers. Analogously, using a similar term as in (4.2) we denote by

$$\mathrm{N}_{\mathrm{linear}} := \frac{\sum_{l\in I_{\mathrm{linear}}} \|W^l\|_0}{\sum_{l\in I_{\mathrm{linear}}} n_l \cdot n_{l-1}}$$

the percentage of weights used in the linear layers. Finally, we define $\mathrm{N}_{\mathrm{total}} := \mathrm{N}_{\mathrm{conv}} + \mathrm{N}_{\mathrm{linear}}$.

We compare our algorithms LinBreg (with momentum) and AdaBreg against vanilla training without sparsity, iterative pruning (Han et al., 2015), and the Group Lasso approach from Scardapane et al. (2017), and train all networks to a comparable sparsity level, given in brackets. The pruning scheme is taken from Han et al. (2015), where in each step a certain amount of weights is pruned, followed by a retraining step. For our experiment the amount of weights pruned in each iteration was chosen, so that a specified target sparsity is met. For the Group Lasso approach, which is based on the regularized risk minimization (1.4), we use two different optimizers. First, we apply SGD applied to the (1.4) and apply thresholding afterwards to obtain sparse weights, which is the standard approach in the community (cf. Scardapane et al. (2017)). Second, we apply proximal gradient descent (1.5) to (1.4) which yields sparse solutions without need for thresholding. Our Bregman algorithms were initialized with 1% non-zero parameters, following the strategy from Section 4.1, all other algorithms were initialized non-sparse using standard techniques (Glorot and Bengio, 2010; He et al., 2015). For all algorithms we tuned the hyperparameters (e.g., pruning rate, regularization parameter for Group Lasso and Bregman) in order to achieve comparable sparsity levels.

Note that it is non-trivial to compare different algorithms for sparse training since they optimize different objectives, and both the sparsity, the train, and the test accuracy of the resulting networks matter. Therefore, we show results whose sparsity levels and accuracies are in similar ranges.

Table 1 shows that all algorithms manage to compute very sparse networks with ca. 2% drop in test accuracy on Fasion-MNIST, compared to vanilla dense training with Adam. Note that we optimized the hyperparameters (regularization and thresholding parameters) of all algorithms for optimal performance on a validation set, subject to having comparable sparsity levels. Our algorithms LinBreg and AdaBreg yield sparser networks with the same accuracies as Pruning and Lasso.

Similar observations are true for Table 2 where we used group sparsity regularization on the convolutional kernels. Here all algorithms apart from pruning yield similar results, whereas pruning exhibits a significantly worse test accuracy despite using a larger number of non-zero parameters. The combination of SGD-optimized Group Lasso with subsequent thresholding yields the best test accuracy using a moderate sparsity level.

As mentioned above the Lasso and Group Lasso results using SGD underwent an additional thresholding step after training in order to generate sparse solutions. Obviously, one could also do this with the results of ProxGD and Bregman which would further improve their sparsity levels. However, in this experiment we refrain from doing so in order not to change the nature of the algorithms.

Strategy	Optimizer	N_{total} in [%]	Test Acc	Train Acc
Vanilla	Adam	100	92.1	100.0
Pruning (5%)	SGD	4.7	89.2	92.0
Lasso	SGD + thresh.	3.5	90.1	94.7
	ProxGD	4.8	89.4	91.4
Bregman	LinBreg	1.9	89.2	91.1
	LinBreg ($\beta = 0.9$)	2.7	89.9	93.8
	AdaBreg	2.3	90.5	93.6

Table 1: Sparsity levels and accuracies on the Fashion-MNIST data set.

Strategy	Optimizer	N_{linear} in [%]	N_{conv} in [%]	Test Acc	Train Acc
Vanilla	Adam	100	100	92.1	100.0
Pruning (7%)	SGD	7.0	6.5	86.9	89.9
GLasso	SGD + thresh.	3.6	4.3	90.3	94.8
	ProxGD	3.0	3.7	89.8	91.6
Bregman	LinBreg	3.8	4.2	89.5	93.1
	LinBreg ($\beta = 0.9$)	3.5	4.7	89.9	93.5
	AdaBreg	3.5	2.8	89.4	92.6

Table 2: Group sparsity levels and accuracies on the Fashion-MNIST data set.

4.5 Residual Neural Networks (ResNets)

In this experiment we trained a ResNet-18 architecture for classification on CIFAR-10, enforcing sparsity through the ℓ_1-norm (1.1) and comparing different strategies,

as before. Table 3 shows the resulting sparsity levels of the total number of parameters and the percentage of non-zero convolutional kernels as well as the train and test accuracies. Note that even though we used the standard ℓ_1 regularization (1.1) and no group sparsity, the trained networks exhibit large percentages of zero-kernels.

For comparison we also show the unregularized vanilla results using SGD with momentum and Adam, which both use 100% of the parameters. The LinBreg result with thresholding shows that one can train a very sparse network using only 3.4% of all parameters with 3.4% drop in test accuracy. With AdaBreg we obtain a sparsity level of 14.7%, resulting in a drop of only 1.3%. The combination of Adam-optimized Lasso with subsequent thresholding yields a 3% sparsity with a drop of 3.6% in test accuracy, which is the sparsest result in this comparison.

Strategy	Optimizer	N_{total} in [%]	N_{conv} in [%]	Test Acc	Train Acc
Vanilla	SGD with momentum	100.0	100.0	92.15	99.8%
	Adam	100.0	100.0	93.6	100.0%
Lasso	Adam	99.7	100.0	91.1	100
	Adam + thresh.	3.0	15.7	90.0	99.8
Bregman	LinBreg	5.5	24.8	90.9	99.5
	LinBreg + thresh.	3.4	16.9	90.2	99.4
	LinBreg ($\beta = 0.9$)	4.8	21.0	90.4	100.0
	LinBreg ($\beta = 0.9$) + thresh.	3.6	17.4	90.0	99.9
	AdaBreg	14.7	56.7	92.3	100.0
	AdaBreg + thresh.	9.2	42.2	90.5	99.9

Table 3: Sparsity levels and accuracies on the CIFAR-10 data set.

4.6 Towards Architecture Design: Unveiling an Autoencoder

In this final experiment we investigate the potential of our Bregman training framework for architecture design, which refers to letting the network learn its own architecture. Hoefler et al. (2021) identified this as one of the main potentials of sparse training.

The inverse scale space character of our approach turns out to be promising for starting with very sparse networks and letting the network choose its own architecture by enforcing, e.g., row sparsity of weight matrices. The simplest yet widely used non-trivial architecture one might hope to train is an autoencoder, as used, e.g., for denoising images. To this end, we utilize the MNIST data set and train a fully connected feedforward network with five hidden layers, all having the same dimension as the input layer, to denoise MNIST images. Similar to the experiments in Section 4.2 we split the dataset in 55,000 images used for training and 5,000 images to evaluate the performance. We enforce row sparsity by using the regular-

izer (1.14), which in this context is equivalent to having few active neurons in the network.

Figure 4 shows the number of active neurons in the network at different stages of the training process using LinBreg. Here darker colors indicate more iterations. We initialized around 1% of all rows non-zero, which corresponds to 8 neurons per layer being initially active. Our algorithm successively adds neurons and converges to an autoencoder-like structure, where the number of neurons decreases until the middle layer and then increases again. Note the network developed this structure "on its own" and that we *did not* artificially generate this result by using different regularization strengths for the different layers. In Figure 5 we additionally show the denoising performance of the trained network on some images from the MNIST test set. The network was trained for 100 iterations, using a regularization value of $\lambda = 0.07$ in (1.14) and employing a standard MSE loss. We evaluate the test performance using the established structural similarity index measure (SSIM) (Wang et al., 2004), which assigns values close to 1 to pairs of images that are perceptually similar. Averaging this value over the whole test set we report a value of SSIM $\approx$ 0.93. For comparison, we also trained a network with 100 iterations of standard SGD which yields no sparsity and a value of SSIM ≈ 0.89.

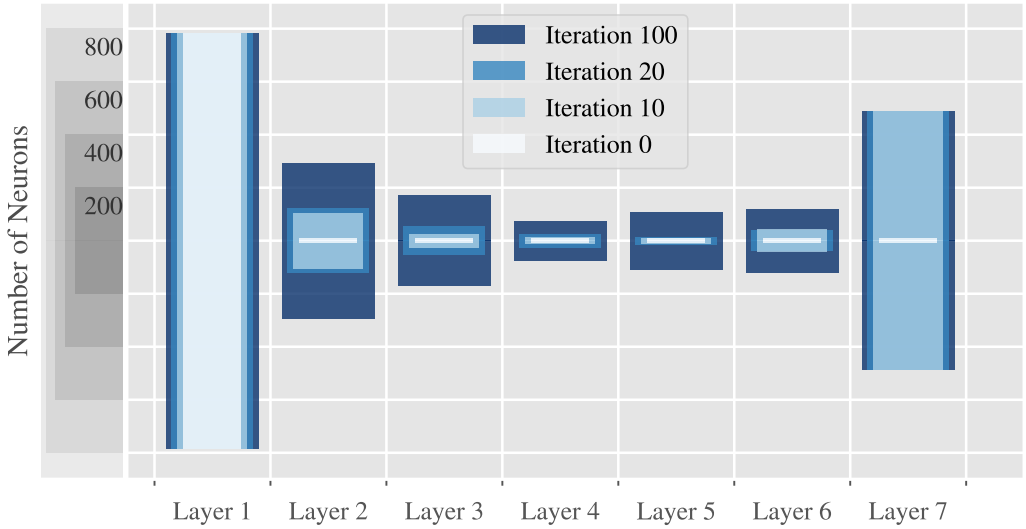


Figure 4: Architecture design for denoising: LinBreg automatically unveils an autoencoder.

5. Conclusion

In this paper we proposed an inverse scale space approach for training sparse neural networks based on linearized Bregman iterations. We introduced *LinBreg* as baseline

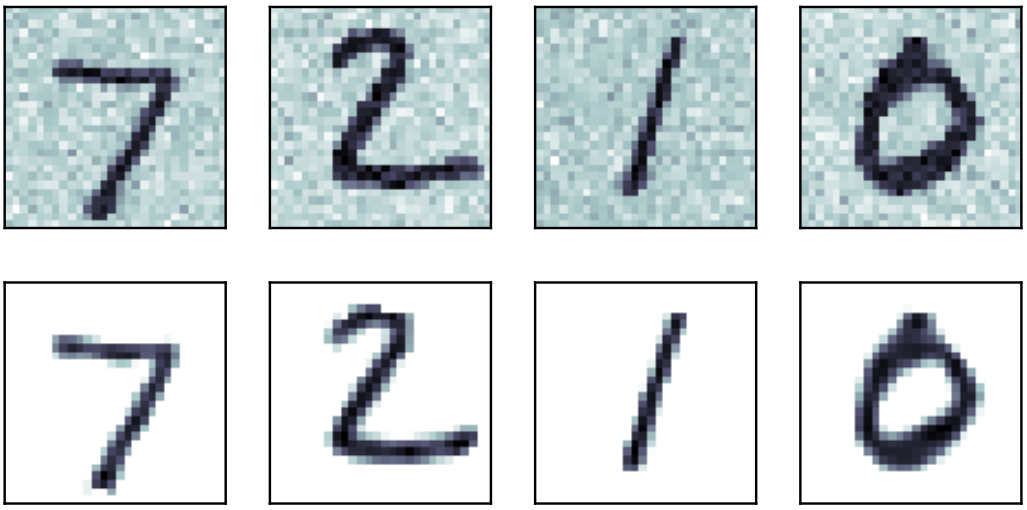


Figure 5: The denoising performance of the trained autoencoder on the test set with an average SSIM value of ≈ 0.93.

algorithm for our learning framework and also discuss two variants using momentum and Adam. The effect of incorporating Bregman iterations into neural network training was investigated in numerical experiments on benchmark data sets. Our observations showed that the proposed method is able to train very sparse and accurate neural networks in an inverse scale space manner without using additional heuristics. Furthermore, we gave a glimpse of its applicability for discovering suitable network architectures for a given application task, e.g., an autoencoder architecture for image denoising. We mathematically supported our findings by performing a stochastic convergence analysis of the loss decay, and we proved convergence of the parameters in the case of convexity.

The proposed Bregman learning framework has a lot of potential for training sparse neural networks, and there are still a few open research questions (see also Hoefler et al. (2021)) which we would like to emphasize in the following.

First, we would like to use the inverse scale space character of the proposed Bregman learning algorithms in combination with sparse backpropagation for resource-friendly training, hence improving the carbon footprint of training (Anthony et al., 2020). This is a non-trivial endeavour for the following reason: A-priori it is not clear which weights are worth updating since estimating the magnitude of the gradient with respect to these weights already requires evaluating the backpropagation. A possible way to achieve this consists in performing a Bregman step to obtain a sparse support of the weights, performing several masked backpropagation steps to optimize the weights in these positions, and alternate this procedure.

Second, our experiment from Section 4.6, where our algorithm discovered a denoising autoencoder, suggests that our method has great potential for general ar-

chitecture design tasks. Using suitable sparsity regularization, e.g., on residual connections and rows of the weight matrices, one can investigate whether networks learn to form a U-net (Ronneberger et al., 2015) structure for the solution of inverse problems.

On the analysis side, it is worth investigating the convergence of LinBreg in the fully non-convex setting based on the Kurdyka–Łojasiewicz inequality and to extend these results to our accelerated algorithms LinBreg with momentum and AdaBreg. Furthermore, it will be interesting to remove the bounded variance condition from Assumption 2, which is known to be possible for stochastic gradient descent if the batch losses satisfy (3.3), see Lei et al. (2019). Finally, the characterizing the limit point of LinBreg for non-strongly convex losses as minimizer which minimizes the Bregman distance to the initialization will be worthwhile.

Acknowledgments

This work was supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 777826 (NoMADS) and by the German Ministry of Science and Technology (BMBF) under grant agreement No. 05M2020 (DELETO). Additionally we thank for the financial support by the Cluster of Excellence "Engineering of Advanced Materials" (EAM) and the "Competence Unit for Scientific Computing" (CSC) at the University of Erlangen-Nürnberg (FAU). LB acknowledges support by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - GZ 2047/1, Projekt-ID 390685813.

References

F. Amato, A. López, E. M. Peña-Méndez, P. Vaňhara, A. Hampl, and J. Havel. Artificial neural networks in medical diagnosis, 2013.

L. F. W. Anthony, B. Kanding, and R. Selvan. Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. *arXiv preprint arXiv:2007.03051*, 2020.

M. Bachmayr and M. Burger. Iterative total variation schemes for nonlinear inverse problems. *Inverse Problems*, 25(10):105004, 2009.

H. Bauschke and P. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. Springer, New York, 2011. ISBN 978-3-319-48311-5.

A. Beck. *First-Order Methods in Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017. doi: 10.1137/1.9781611974997.

A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

A. Benfenati and V. Ruggiero. Inexact Bregman iteration with an application to poisson data reconstruction. *Inverse Problems*, 29(6):065016, 2013.

M. Benning and M. Burger. Modern regularization methods for inverse problems. *Acta Numerica*, 27:1–111, 2018.

M. Benning, M. M. Betcke, M. J. Ehrhardt, and C.-B. Schönlieb. Choose your path wisely: gradient descent in a Bregman distance framework. *SIAM Journal on Imaging Sciences*, 14(2):814–843, 2021.

M. Burger, G. Gilboa, S. Osher, J. Xu, et al. Nonlinear inverse scale space methods. *Communications in Mathematical Sciences*, 4(1):179–212, 2006.

M. Burger, K. Frick, S. Osher, and O. Scherzer. Inverse total variation flow. *Multiscale Modeling & Simulation*, 6(2):366–395, 2007.

M. Burger, M. Möller, M. Benning, and S. Osher. An adaptive inverse scale space method for compressed sensing. *Mathematics of Computation*, 82(281):269–299, 2013.

J.-F. Cai, S. Osher, and Z. Shen. Convergence of the linearized Bregman iteration for ℓ_1-norm minimization. *Mathematics of Computation*, 78(268):2127–2136, 2009a.

J.-F. Cai, S. Osher, and Z. Shen. Linearized Bregman iterations for compressed sensing. *Mathematics of computation*, 78(267):1515–1536, 2009b.

G. Castellano, A. M. Fanelli, and M. Pelillo. An iterative pruning algorithm for feedforward neural networks. *IEEE transactions on Neural networks*, 8(3):519–531, 1997.

G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

X. Dai, H. Yin, and N. K. Jha. NeST: A neural network synthesis tool based on a grow-and-prune paradigm. *IEEE Transactions on Computers*, 68(10):1487–1497, 2019.

J. de Dios and J. Bruna. On sparsity in overparametrised shallow ReLU networks. *arXiv preprint arXiv:2006.10225*, 2020.

T. Dettmers and L. Zettlemoyer. Sparse networks from scratch: Faster training without losing performance. *arXiv preprint arXiv:1907.04840*, 2019.

P. Dhar. The carbon impact of artificial intelligence. *Nat Mach Intell*, 2:423–5, 2020.

R. D'Orazio, N. Loizou, I. Laradji, and I. Mitliagkas. Stochastic mirror descent: Convergence analysis and adaptive variants via the mirror stochastic Polyak stepsize. *arXiv preprint arXiv:2110.15412*, 2021.

R. A. Dragomir, M. Even, and H. Hendrikx. Fast stochastic Bregman gradient methods: Sharp analysis and variance reduction. In *International Conference on Machine Learning*, pages 2815–2825. PMLR, 2021.

U. Evci, T. Gale, J. Menick, P. S. Castro, and E. Elsen. Rigging the lottery: Making all tickets winners. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2943–2952. PMLR, 2020.

J. Frankle and M. Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

M. P. Friedlander and P. Tseng. Exact regularization of convex programs. *SIAM Journal on Optimization*, 18(4):1326–1350, 2008.

Y. Fu, C. Liu, D. Li, Z. Zhong, X. Sun, J. Zeng, and Y. Yao. Exploring structural sparsity of deep networks via inverse scale spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

S. Han, J. Pool, J. Tran, and W. Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

T. Hoefler, D. Alistarh, T. Ben-Nun, N. Dryden, and A. Peste. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *J. Mach. Learn. Res.*, 22(241):1–124, 2021.

C. Huang, X. Sun, J. Xiong, and Y. Yao. Split LBI: An iterative regularization path with structural sparsity. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 3377–3385, 2016.

T. Jia, Y. Shi, Y. Zhu, and L. Wang. An image restoration model combining mixed L^1/L^2 fidelity terms. *Journal of Visual Communication and Image Representation*, 38:461–473, 2016.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Y. LeCun and C. Cortes. MNIST handwritten digit database. 2010. URL `http://yann.lecun.com/exdb/mnist/`.

Y. LeCun, J. S. Denker, and S. A. Solla. Optimal brain damage. In *Advances in neural information processing systems*, pages 598–605, 1990.

Y. Lei, T. Hu, G. Li, and K. Tang. Stochastic gradient descent for nonconvex learning without bounded gradient assumptions. *IEEE transactions on neural networks and learning systems*, 31(10):4394–4400, 2019.

D. Li, X. Tian, Q. Jin, and K. Hirasawa. Adaptive fractional-order total variation image restoration with split Bregman iteration. *ISA transactions*, 82:210–222, 2018.

S. Liu, D. C. Mocanu, A. R. R. Matavalam, Y. Pei, and M. Pechenizkiy. Sparse evolutionary deep learning with over one million artificial neurons on commodity hardware. *Neural Computing and Applications*, 33(7):2589–2604, 2021.

C. Louizos, M. Welling, and D. P. Kingma. Learning sparse neural networks through l_0 regularization. *arXiv preprint arXiv:1712.01312*, 2017.

Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang. The expressive power of neural networks: A view from the width. *Advances in neural information processing systems*, 30, 2017.

J. Martens. Deep learning via Hessian-free optimization. In *ICML*, volume 27, pages 735–742, 2010.

D. C. Mocanu, E. Mocanu, P. Stone, P. H. Nguyen, M. Gibescu, and A. Liotta. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature communications*, 9(1):1–12, 2018.

A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4): 1574–1609, 2009.

A. S. Nemirovskij and D. B. Yudin. Problem complexity and method efficiency in optimization. 1983.

L. Nguyen, P. H. Nguyen, M. van Dijk, P. Richtarik, K. Scheinberg, and M. Takac. SGD and hogwild! Convergence without the bounded gradients assumption. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3750–3758. PMLR, 10–15 Jul 2018.

A. Nitanda. Stochastic proximal gradient descent with acceleration techniques. *Advances in Neural Information Processing Systems*, 27:1574–1582, 2014.

A. Orvieto, J. Kohler, and A. Lucchi. The role of memory in stochastic optimization. In *Uncertainty in Artificial Intelligence*, pages 356–366. PMLR, 2020.

S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation*, 4(2):460–489, 2005.

S. Osher, Y. Mao, B. Dong, and W. Yin. Fast linearized Bregman iteration for compressive sensing and sparse denoising. *Communications in Mathematical Sciences*, 8(1):93–111, 2010.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

W. Rawat and Z. Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9):2352–2449, 2017.

S. J. Reddi, S. Sra, B. Poczos, and A. J. Smola. Proximal stochastic methods for nonsmooth nonconvex finite-sum optimization. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 1153–1161, 2016.

R. Rockafellar. *Convex analysis*. Princeton University Press, Princeton, N.J, 1997. ISBN 9781400873173.

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

L. Rosasco, S. Villa, and B. C. Vũ. Convergence of stochastic proximal gradient algorithm. *Applied Mathematics & Optimization*, 82(3):891–917, 2020.

S. Scardapane, D. Comminiello, A. Hussain, and A. Uncini. Group sparse regularization for deep neural networks. *Neurocomputing*, 241:81–89, 2017.

D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587): 484–489, 2016.

S. Srinivas, A. Subramanya, and R. Venkatesh Babu. Training sparse neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 138–145, 2017.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

G. Turinici. The convergence of the stochastic gradient descent (SGD): a self-contained proof. *arXiv preprint arXiv:2103.14350*, 2021.

Z. Wang, A. C. Bovik, H. R. Sheikh, S. Member, E. P. Simoncelli, and S. Member. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13:600–612, 2004.

Y. Yang, Y. Yuan, A. Chatzimichailidis, R. J. van Sloun, L. Lei, and S. Chatzinotas. ProxSGD: Training structured neural networks under regularization and constraints. In *International Conference on Learning Representations*, 2019.

W. Yin. Analysis and generalizations of the linearized Bregman method. *SIAM Journal on Imaging Sciences*, 3(4):856–877, 2010.

W. Yin, S. Osher, D. Goldfarb, and J. Darbon. Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing. *SIAM Journal on Imaging sciences*, 1(1):143–168, 2008.

J. Yun, A. C. Lozano, and E. Yang. A general family of stochastic proximal gradient methods for deep learning. *arXiv preprint arXiv:2007.07484*, 2020.

X. Zhang, M. Burger, and S. Osher. A unified primal-dual algorithm framework based on Bregman iteration. *Journal of Scientific Computing*, 46(1):20–46, 2011.

M. Zhu and S. Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878*, 2017.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2): 301–320, 2005.

Appendix

In all proofs we will use the abbreviation $g^{(k)} := g(\theta^{(k)}; \omega^{(k)})$ as in (3.1).

A. Proofs from Section 3.1

Proof [Proof of Theorem 2] Using (3.3) one obtains

$$\begin{aligned}
&\mathcal{L}(\theta^{(k+1)}) - \mathcal{L}(\theta^{(k)}) \\
&\leq \langle \nabla \mathcal{L}(\theta^{(k)}), \theta^{(k+1)} - \theta^{(k)} \rangle + \frac{L}{2} \|\theta^{(k+1)} - \theta^{(k)}\|^2 \\
&= \langle g^{(k)}, \theta^{(k+1)} - \theta^{(k)} \rangle + \langle \nabla \mathcal{L}(\theta^{(k)}) - g^{(k)}, \theta^{(k+1)} - \theta^{(k)} \rangle + \frac{L}{2} \|\theta^{(k+1)} - \theta^{(k)}\|^2 \\
&= -\frac{1}{\tau^{(k)}} \langle v^{(k+1)} - v^{(k)}, \theta^{(k+1)} - \theta^{(k)} \rangle \\
&\quad + \langle \nabla \mathcal{L}(\theta^{(k)}) - g^{(k)}, \theta^{(k+1)} - \theta^{(k)} \rangle + \frac{L}{2} \|\theta^{(k+1)} - \theta^{(k)}\|^2 \\
&= -\frac{1}{\tau^{(k)}} D_J^{\mathrm{sym}}(\theta^{(k+1)}, \theta^{(k)}) - \frac{1}{\delta \tau^{(k)}} \|\theta^{(k+1)} - \theta^{(k)}\|^2 \\
&\quad + \langle \nabla \mathcal{L}(\theta^{(k)}) - g^{(k)}, \theta^{(k+1)} - \theta^{(k)} \rangle + \frac{L}{2} \|\theta^{(k+1)} - \theta^{(k)}\|^2.
\end{aligned}$$

Reordering and using the Cauchy–Schwarz inequality yields

$$\begin{aligned}
\mathcal{L}(\theta^{(k+1)}) - \mathcal{L}(\theta^{(k)}) + \frac{1}{\tau^{(k)}} D_J^{\mathrm{sym}}(\theta^{(k+1)}, \theta^{(k)}) + \frac{2 - L\delta\tau^{(k)}}{2\delta\tau^{(k)}} \|\theta^{(k+1)} - \theta^{(k)}\|^2 \leq \\
\|\nabla \mathcal{L}(\theta^{(k)}) - g^{(k)}\| \|\theta^{(k+1)} - \theta^{(k)}\|.
\end{aligned}$$

Taking expectations and using Young's inequality gives for any $c > 0$

$$\begin{aligned}
&\mathbb{E}\left[\mathcal{L}(\theta^{(k+1)})\right] - \mathbb{E}\left[\mathcal{L}(\theta^{(k)})\right] + \frac{1}{\tau^{(k)}} \mathbb{E}\left[D_J^{\mathrm{sym}}(\theta^{(k+1)}, \theta^{(k)})\right] \\
&\qquad + \frac{2 - L\delta\tau^{(k)}}{2\delta\tau^{(k)}} \mathbb{E}\left[\|\theta^{(k+1)} - \theta^{(k)}\|^2\right] \leq \tau^{(k)} \delta \frac{\sigma^2}{2c} + \frac{c}{2\delta\tau^{(k)}} \mathbb{E}\left[\|\theta^{(k+1)} - \theta^{(k)}\|^2\right].
\end{aligned}$$

If c is sufficiently small and $\tau^{(k)} < \frac{2}{L\delta}$, we can absorb the last term into the left hand side and obtain

$$\begin{aligned}
\mathbb{E}\left[\mathcal{L}(\theta^{(k+1)})\right] - \mathbb{E}\left[\mathcal{L}(\theta^{(k)})\right] + \frac{1}{\tau^{(k)}} \mathbb{E}\left[D_J^{\mathrm{sym}}(\theta^{(k+1)}, \theta^{(k)})\right] + \frac{C}{2\delta\tau^{(k)}} \mathbb{E}\left[\|\theta^{(k+1)} - \theta^{(k)}\|^2\right] \\
\leq \tau^{(k)} \delta \frac{\sigma^2}{2c},
\end{aligned}$$

where $C > 0$ is a suitable constant. This shows (3.5). ■

Proof [Proof of Corollary 3] Using the assumptions on $\tau^{(k)}$, we can multiply (3.5) with $\tau^{(k)}$ and sum up the resulting inequality to obtain

$$\tau^{(K)}\mathbb{E}\left[\mathcal{L}(\theta^{(K)})\right] - \tau^{(0)}\mathbb{E}\left[\mathcal{L}(\theta^{(0)})\right] + \sum_{k=0}^{K-1}\left(\mathbb{E}\left[D_J^{\mathrm{sym}}(\theta^{(k+1)},\theta^{(k)})\right] + \frac{C}{2\delta}\mathbb{E}\left[\|\theta^{(k+1)}-\theta^{(k)}\|^2\right]\right) \leq \delta\frac{\sigma^2}{2c}\sum_{k=0}^{K-1}(\tau^{(k)})^2$$

Since $\mathcal{L} \geq 0$ we can drop the first term. Sending $K \to \infty$ and using that the step sizes are square-summable concludes the proof. ∎

B. Proof from Section 3.2

The following lemma allows to split the Bregman distance with respect to the elastic net functional J_δ into two Bregman distances with respect to the functional J and the Euclidean norm.

Lemma 13 *For all $\tilde{\theta}, \theta \in \Theta$ and $v \in \partial J_\delta(\theta)$ it holds that*

$$D_{J_\delta}^v(\tilde{\theta},\theta) = D_J^p(\tilde{\theta},\theta) + \frac{1}{2\delta}\|\tilde{\theta}-\theta\|^2,$$

where $p := v - \frac{1}{\delta}\theta \in \partial J(\theta)$.

Proof Since $\partial J_\delta(\theta) = \partial J(\theta) + \partial\frac{1}{2\delta}\|\theta\|^2$, we can write $v = p + \frac{1}{\delta}\theta$ with $p \in \partial J(\theta)$. This readily yields

$$\begin{aligned} D_{J_\delta}^v(\tilde{\theta},\theta) &= J_\delta(\tilde{\theta}) - J_\delta(\theta) - \langle v, \tilde{\theta}-\theta\rangle \\ &= J(\tilde{\theta}) + \frac{1}{2\delta}\|\tilde{\theta}\|^2 - J(\theta) - \frac{1}{2\delta}\|\theta\|^2 - \langle p, \tilde{\theta}-\theta\rangle - \frac{1}{\delta}\langle\theta,\tilde{\theta}-\theta\rangle \\ &= D_J^p(\tilde{\theta},\theta) + \frac{1}{2\delta}\|\tilde{\theta}\|^2 - \frac{1}{2\delta}\|\theta\|^2 - \frac{1}{\delta}\langle\theta,\tilde{\theta}\rangle + \frac{1}{\delta}\|\theta\|^2 \\ &= D_J^p(\tilde{\theta},\theta) + \frac{1}{2\delta}\|\tilde{\theta}-\theta\|^2. \end{aligned}$$

∎

The next lemma expresses the difference of two subsequent Bregman distances along the iteration in two different ways. The first one is useful for proving convergence of (3.1) under the weaker convexity Assumption 4, whereas the second one is used for proving the stronger convergence statement Theorem 11 under Assumption 5.

Lemma 14 *Denoting* $d_k := \mathbb{E}\left[D_{J_\delta}^{v^{(k)}}(\theta^*, \theta^{(k)})\right]$ *the iteration* (3.1) *fulfills:*

$$d_{k+1} - d_k = -\mathbb{E}\left[D_{J_\delta}^{v^{(k)}}(\theta^{(k+1)}, \theta^{(k)})\right] + \tau^{(k)}\mathbb{E}\left[\langle g^{(k)}, \theta^* - \theta^{(k+1)}\rangle\right], \tag{B.1}$$

$$d_{k+1} - d_k = \mathbb{E}\left[D_{J_\delta}^{v^{(k+1)}}(\theta^{(k)}, \theta^{(k+1)})\right] + \tau^{(k)}\mathbb{E}\left[\langle \nabla\mathcal{L}(\theta^{(k)}), \theta^* - \theta^{(k)}\rangle\right]. \tag{B.2}$$

Proof We compute using the update in (3.1)

$$\begin{aligned}
&D_{J_\delta}^{v^{(k+1)}}(\theta^*, \theta^{(k+1)}) - D_{J_\delta}^{v^{(k)}}(\theta^*, \theta^{(k)}) \\
&= J_\delta(\theta^{(k)}) - J_\delta(\theta^{(k+1)}) - \langle v^{(k+1)}, \theta^* - \theta^{(k+1)}\rangle + \langle v^{(k)}, \theta^* - \theta^{(k)}\rangle \\
&= -\Big(J_\delta(\theta^{(k+1)}) - J_\delta(\theta^{(k)}) - \langle v^{(k)}, \theta^{(k+1)} - \theta^{(k)}\rangle\Big) - \langle v^{(k)}, \theta^{(k+1)} - \theta^{(k)}\rangle \\
&\qquad - \langle v^{(k+1)}, \theta^* - \theta^{(k+1)}\rangle + \langle v^{(k)}, \theta^* - \theta^{(k)}\rangle \\
&= -D_{J_\delta}^{v^{(k)}}(\theta^{(k+1)}, \theta^{(k)}) + \langle v^{(k)} - v^{(k+1)}, \theta^* - \theta^{(k+1)}\rangle \\
&= -D_{J_\delta}^{v^{(k)}}(\theta^{(k+1)}, \theta^{(k)}) + \tau^{(k)}\langle g^{(k)}, \theta^* - \theta^{(k+1)}\rangle.
\end{aligned}$$

For the second equation we similarly compute

$$\begin{aligned}
&D_{J_\delta}^{v^{(k+1)}}(\theta^*, \theta^{(k+1)}) - D_{J_\delta}^{v^{(k)}}(\theta^*, \theta^{(k)}) \\
&= J_\delta(\theta^{(k)}) - J_\delta(\theta^{(k+1)}) - \langle v^{(k+1)}, \theta^* - \theta^{(k+1)}\rangle + \langle v^{(k)}, \theta^* - \theta^{(k)}\rangle \\
&= J_\delta(\theta^{(k)}) - J_\delta(\theta^{(k+1)}) - \langle v^{(k+1)}, \theta^* - \theta^{(k+1)}\rangle \\
&\qquad + \langle v^{(k+1)}, \theta^* - \theta^{(k)}\rangle + \tau^{(k)}\langle g^{(k)}, \theta^* - \theta^{(k)}\rangle \\
&= J_\delta(\theta^{(k)}) - J_\delta(\theta^{(k+1)}) - \langle v^{(k+1)}, \theta^{(k)} - \theta^{(k+1)}\rangle + \tau^{(k)}\langle g^{(k)}, \theta^* - \theta^{(k)}\rangle \\
&= D_{J_\delta}^{v^{(k+1)}}(\theta^{(k)}, \theta^{(k+1)}) + \tau^{(k)}\langle g^{(k)}, \theta^* - \theta^{(k)}\rangle.
\end{aligned}$$

Taking expectations and using that $g^{(k)}$ and $\theta^* - \theta^{(k)}$ are stochastically independent to replace $g^{(k)}$ with $\nabla\mathcal{L}(\theta^{(k)})$ inside the expectation concludes the proof. Note that this argument does not apply to the first equality since $\theta^{(k+1)}$ is *not stochastically independent* of $\theta^{(k)}$. ■

Now we prove Theorem 6 by showing that the Bregman distance to the minimizer of the loss and that the iterates converge in norm to the minimizer.

Proof [Proof of Theorem 6] Using (3.2), Assumption 4, and Young's inequality we obtain for any $c > 0$

$$\begin{aligned}
&\langle \nabla\mathcal{L}(\theta^{(k)}), \theta^* - \theta^{(k+1)}\rangle \\
&= \langle \nabla\mathcal{L}(\theta^{(k+1)}), \theta^* - \theta^{(k+1)}\rangle + \langle \nabla\mathcal{L}(\theta^{(k)}) - \nabla\mathcal{L}(\theta^{(k+1)}), \theta^* - \theta^{(k+1)}\rangle \\
&\leq \mathcal{L}(\theta^*) - \mathcal{L}(\theta^{(k+1)}) - \frac{\mu}{2}\|\theta^* - \theta^{(k+1)}\|^2 + \frac{L^2}{2c}\|\theta^{(k)} - \theta^{(k+1)}\|^2 + \frac{c}{2}\|\theta^* - \theta^{(k+1)}\|^2.
\end{aligned}$$

Using that $\mathcal{L}(\theta^*) \leq \mathcal{L}(\theta^{(k+1)})$ we obtain

$$\langle \nabla \mathcal{L}(\theta^{(k)}), \theta^* - \theta^{(k+1)} \rangle \leq \frac{L^2}{2c} \|\theta^{(k)} - \theta^{(k+1)}\|^2 + \frac{c-\mu}{2} \|\theta^* - \theta^{(k+1)}\|^2.$$

Plugging this into the expression (B.1) for $d_{k+1} - d_k$ yields

$$\begin{aligned} d_{k+1} - d_k &= -\mathbb{E}\left[D_{J_\delta}^{v^{(k)}}(\theta^{(k+1)}, \theta^{(k)}) \right] + \tau^{(k)} \mathbb{E}\left[\langle \nabla \mathcal{L}(\theta^{(k)}), \theta^* - \theta^{(k+1)} \rangle \right] \\ &\quad + \tau^{(k)} \mathbb{E}\left[\langle g^{(k)} - \nabla \mathcal{L}(\theta^{(k)}), \theta^* - \theta^{(k+1)} \rangle \right] \\ &\leq -\mathbb{E}\left[D_{J_\delta}^{v^{(k)}}(\theta^{(k+1)}, \theta^{(k)}) \right] + \tau^{(k)} \frac{L^2}{2c} \mathbb{E}\left[\|\theta^{(k)} - \theta^{(k+1)}\|^2 \right] \\ &\quad + \frac{c-\mu}{2} \tau^{(k)} \mathbb{E}\left[\|\theta^* - \theta^{(k+1)}\|^2 \right] \\ &\quad + \tau^{(k)} \mathbb{E}\left[\langle g^{(k)} - \nabla \mathcal{L}(\theta^{(k)}), \theta^* - \theta^{(k+1)} \rangle \right]. \end{aligned}$$

Now we utilize that $\theta^* - \theta^{(k)}$ and $g^{(k)} - \nabla \mathcal{L}(\theta^{(k)})$ are stochastically independent and that the latter has zero expectation to infer

$$\begin{aligned} &\mathbb{E}\left[\langle g^{(k)} - \nabla \mathcal{L}(\theta^{(k)}), \theta^* - \theta^{(k+1)} \rangle \right] \\ &= \mathbb{E}\left[\langle g^{(k)} - \nabla \mathcal{L}(\theta^{(k)}), \theta^{(k)} - \theta^{(k+1)} \rangle \right] \\ &\qquad + \mathbb{E}\left[\langle g^{(k)} - \nabla \mathcal{L}(\theta^{(k)}), \theta^* - \theta^{(k)} \rangle \right] \\ &= \mathbb{E}\left[\langle g^{(k)} - \nabla \mathcal{L}(\theta^{(k)}), \theta^{(k)} - \theta^{(k+1)} \rangle \right]. \end{aligned}$$

Applying Young's inequality and using Assumption 2 yields

$$\begin{aligned} &\tau^{(k)} \mathbb{E}\left[\langle g^{(k)} - \nabla \mathcal{L}(\theta^{(k)}), \theta^{(k)} - \theta^{(k+1)} \rangle \right] \\ &\leq \frac{\delta (\tau^{(k)})^2}{2} \sigma + \frac{\sigma}{2\delta} \mathbb{E}\left[\|\theta^{(k)} - \theta^{(k+1)}\|^2 \right]. \end{aligned}$$

Plugging this into the expression for $d_{k+1} - d_k$ and reordering we infer that for $\tau^{(k)} \leq \frac{\mu}{2\delta L^2}$ and $c = \mu/2$ it holds

$$\begin{aligned} &d_{k+1} - d_k + \frac{\mu}{4} \tau^{(k)} \mathbb{E}\left[\|\theta^* - \theta^{(k+1)}\|^2 \right] \\ &\leq -\mathbb{E}\left[D_J^{p^{(k)}}(\theta^{(k+1)}, \theta^{(k)}) \right] + \frac{\delta (\tau^{(k)})^2}{2} \sigma + \frac{\sigma}{2\delta} \mathbb{E}\left[\|\theta^{(k)} - \theta^{(k+1)}\|^2 \right], \end{aligned}$$

which yields (3.8). Summing this inequality, using Corollary 3—which is possible since $\tau^{(k)} \leq \frac{\mu}{2\delta L^2} \leq \frac{1}{2\delta L} \leq \frac{2}{\delta L}$—and that the step sizes are square-summable then shows that

$$\sum_{k=0}^{\infty} \tau^{(k)} \mathbb{E}\left[\|\theta^* - \theta^{(k+1)}\|^2 \right] < \infty.$$

Since $\sum_{k=0}^{\infty} \tau^{(k)} = \infty$ this means that

$$\min_{k\in\{1,\dots,K\}} \mathbb{E}\left[\|\theta^* - \theta^{(k)}\|^2\right] \to 0, \quad K \to \infty.$$

Hence, for an appropriate subsequence $\theta^{(k_j)}$ it holds

$$\lim_{j\to\infty} \mathbb{E}\left[\|\theta^* - \theta^{(k_j)}\|^2\right] = 0.$$

■

Proof [Proof of Corollary 10] Since J is equal to the ℓ_1-norm, J admits the triangle inequality $J(\theta^*) - J(\theta^{(k)}) \leq J(\theta^* - \theta^{(k)})$. Furthermore, since $d := \dim \Theta$ is finite, it admits the norm inequality $J(\theta) \leq \sqrt{d}\|\theta\|$ and has bounded subgradients $\|p\| = \|\operatorname{sign}(\theta)\| \leq d$ for all $\theta \in \Theta$.

Hence, using Lemma 13 we can estimate

$$\begin{aligned} d_k &= \mathbb{E}\left[D_{J_\delta}^{v^{(k)}}(\theta^*, \theta^{(k)})\right] = \mathbb{E}\left[D_J^{p^{(k)}}(\theta^*, \theta^{(k)})\right] + \frac{1}{2\delta}\mathbb{E}\left[\|\theta^* - \theta^{(k)}\|^2\right] \\ &= \mathbb{E}\left[J(\theta^*) - J(\theta^{(k)}) - \langle p^{(k)}, \theta^* - \theta^{(k)}\rangle\right] + \frac{1}{2\delta}\mathbb{E}\left[\|\theta^* - \theta^{(k)}\|^2\right] \\ &\leq (\sqrt{d} + d)\mathbb{E}\left[\|\theta^* - \theta^{(k)}\|\right] + \frac{1}{2\delta}\mathbb{E}\left[\|\theta^* - \theta^{(k)}\|^2\right]. \end{aligned}$$

Hence, since $\mathbb{E}\left[\|\theta^* - \theta^{(k_j)}\|^2\right]$ converges to zero according to Theorem 6, the same is true for the sequence d_{k_j}. Furthermore, we have proved that $d_{k+1} - d_k \leq c_k$, where c_k is a non-negative and summable sequence. This also implies $d_m \leq d_k + \sum_{j=k}^{\infty} c_k$ for every $m > k$.

Since c_k is summable and d_{k_j} converges to zero there exists $k \in \mathbb{N}$ and $l \in \mathbb{N}$ such that

$$\sum_{j=k}^{\infty} c_j < \frac{\varepsilon}{2}, \quad d_{k_l} < \frac{\varepsilon}{2}, \quad k_l > k.$$

Hence, we obtain for any $m > k_l$

$$d_m \leq d_{k_l} + \sum_{j=k_l}^{\infty} c_k \leq d_{k_l} + \sum_{j=k}^{\infty} c_k < \varepsilon.$$

Since $\varepsilon > 0$ was arbitrary, this implies that $d_m \to 0$ as $m \to \infty$. ■

Finally we prove Theorem 11 based on Assumption 5, which asserts convergence in the Bregman distance.

Proof [Proof of Theorem 11] The proof goes along the lines of Turinici (2021), however Assumption 2 is weaker than the assumption posed there.
Item 1: Since proximal operators are 1-Lipschitz it holds

$$\|\theta^{(k+1)} - \theta^{(k)}\| = \|\operatorname{prox}_{\delta J}(\delta v^{(k+1)}) - \operatorname{prox}_{\delta J}(\delta v^{(k)})\| \leq \delta \|v^{(k+1)} - v^{(k)}\|.$$

Using the Cauchy–Schwarz inequality and this estimate yields

$$\begin{aligned}
\frac{1}{\tau^{(k)}}\mathbb{E}\left[D_{J_\delta}^{v^{(k+1)}}(\theta^{(k)}, \theta^{(k+1)})\right] &\leq \frac{1}{\tau^{(k)}}\mathbb{E}\left[D_{J_\delta}^{\mathrm{sym}}(\theta^{(k)}, \theta^{(k+1)})\right] \\
&= \frac{1}{\tau^{(k)}}\mathbb{E}\left[\langle v^{(k+1)} - v^{(k)}, \theta^{(k+1)} - \theta^{(k)}\rangle\right] \\
&= -\mathbb{E}\left[\langle g^{(k)}, \theta^{(k+1)} - \theta^{(k)}\rangle\right] \\
&\leq \sqrt{\mathbb{E}\left[\|g^{(k)}\|^2\right]}\sqrt{\mathbb{E}\left[\|\theta^{(k+1)} - \theta^{(k)}\|^2\right]} \\
&\leq \delta\sqrt{\mathbb{E}\left[\|g^{(k)}\|^2\right]}\sqrt{\mathbb{E}\left[\|v^{(k+1)} - v^{(k)}\|^2\right]} \\
&= \delta\tau^{(k)}\mathbb{E}\left[\|g^{(k)}\|^2\right].
\end{aligned}$$

It can be easily seen that Assumption 2 is equivalent to

$$\mathbb{E}\left[\|g^{(k)}\|^2\right] \leq \sigma^2 + \mathbb{E}\left[\|\nabla\mathcal{L}(\theta^{(k)})\|^2\right],$$

which together with the Lipschitz continuity of $\nabla\mathcal{L}$ from Assumption 1 implies

$$\begin{aligned}
\mathbb{E}\left[D_{J_\delta}^{v^{(k+1)}}(\theta^{(k)}, \theta^{(k+1)})\right] &\leq \delta(\tau^{(k)})^2\left(\sigma^2 + \mathbb{E}\left[\|\nabla\mathcal{L}(\theta^{(k)})\|^2\right]\right) \\
&\leq \delta(\tau^{(k)})^2\left(\sigma^2 + L^2\mathbb{E}\left[\|\theta^* - \theta^{(k)}\|^2\right]\right) \\
&\leq \delta(\tau^{(k)})^2\left(\sigma^2 + 2\delta L^2 d_k\right).
\end{aligned}$$

We plug this estimate into (B.2) and utilize Assumption 5 to obtain

$$\begin{aligned}
d_{k+1} - d_k &\leq \delta(\tau^{(k)})^2\left(\sigma^2 + 2\delta L^2 d_k\right) + \tau^{(k)}\mathbb{E}\left[\mathcal{L}(\theta^*) - \mathcal{L}(\theta^{(k)}) - \nu D_{J_\delta}^{v^{(k)}}(\theta^*, \theta^{(k)})\right] \\
&\leq \delta(\tau^{(k)})^2\left(\sigma^2 + 2\delta L^2 d_k\right) - \tau^{(k)}\nu\mathbb{E}\left[D_{J_\delta}^{v^{(k)}}(\theta^*, \theta^{(k)})\right] \\
&= \delta(\tau^{(k)})^2\left(\sigma^2 + 2\delta L^2 d_k\right) - \tau^{(k)}\nu d_k,
\end{aligned}$$

which is equivalent to (3.10).
Item 2: For $\tau^{(k)}$ sufficiently small (3.10) implies

$$d_{k+1} \leq \left(1 - \tau^{(k)}\tilde{\nu}\right)d_k + \delta(\tau^{(k)})^2\sigma^2$$

for a constant $\tilde{\nu} \in (0, \nu)$. For any $C \in \mathbb{R}$ we can reformulate this to

$$d_{k+1} - C \leq (1 - \tau^{(k)}\tilde{\nu})(d_k - C) + \tau^{(k)}\left(\tau^{(k)}\delta\sigma^2 - \tilde{\nu}C\right). \tag{B.3}$$

If $\tau^k = \tau$ for all $k \in \mathbb{N}$ is constant and we choose $C = \tau\frac{\delta\sigma^2}{\tilde{\nu}}$, we obtain

$$\left(d_{k+1} - \tau\tfrac{\delta\sigma^2}{\tilde{\nu}}\right)_+ \leq (1 - \tau\tilde{\nu})\left(d_k - \tau\tfrac{\delta\sigma^2}{\tilde{\nu}}\right)_+,$$

where we passed to the positive part $x_+ := \max(x, 0)$. Iterating this inequality yields

$$\left(d_{k+n} - \tau\tfrac{\delta\sigma^2}{\tilde{\nu}}\right)_+ \leq (1 - \tau\tilde{\nu})^n\left(d_k - \tau\tfrac{\delta\sigma^2}{\tilde{\nu}}\right)_+$$

and hence for $\tau < \frac{1}{\tilde{\nu}}$ we get

$$\limsup_{k\to\infty} d_k \leq \tau\tfrac{\delta\sigma^2}{\tilde{\nu}}.$$

Demanding $\tau < \frac{\varepsilon\tilde{\nu}}{\delta\sigma^2} \wedge \frac{1}{\tilde{\nu}}$ we finally obtain

$$\limsup_{k\to\infty} d_k \leq \varepsilon.$$

Item 3: We use the reformulation (B.3) with $C = \varepsilon > 0$. Since $\tau^{(k)}$ converges to zero the second term is non-positive for $k \in \mathbb{N}$ sufficiently large and we obtain

$$(d_{k+1} - \varepsilon)_+ \leq (1 - \tau^{(k)}\tilde{\nu})(d_k - \varepsilon)_+.$$

Iterating this inequality yields

$$(d_{k+n} - \varepsilon)_+ \leq \prod_{l=k}^{k+n-1} (1 - \tau^{(l)}\tilde{\nu})(d_k - \varepsilon)_+.$$

If $k \in \mathbb{N}$ is sufficiently large, then $\tau^{(l)}\tilde{\nu} < 1$ for all $l \geq k$ and the product satisfies

$$\prod_{l=k}^{k+n-1} (1 - \tau^{(l)}\tilde{\nu}) = \exp\left(\sum_{l=k}^{k+n-1} \log(1 - \tau^{(l)}\tilde{\nu})\right) \leq \exp\left(-\sum_{l=k}^{k+n-1} \tau^{(l)}\tilde{\nu}\right) \to 0, \quad n \to \infty,$$

where we used $\log(1-x) \leq -x$ for all $x < 1$ and $\sum_k \tau^{(k)} = \infty$. Since ε was arbitrary we obtain the assertion. ∎

Print P5
Resolution-Invariant Image Classification based on Fourier Neural Operators

S. Kabri, T. Roith, D. Tenbrinck, and M. Burger
journaltitle (2023)

Resolution-Invariant Image Classification based on Fourier Neural Operators*

Samira Kabri[1(✉)], Tim Roith[1], Daniel Tenbrinck[1], and Martin Burger[2,3]

[1] Friedrich-Alexander-Universität Erlangen-Nürnberg, 91058 Erlangen, Germany
[2] Deutsches Elektronen-Synchrotron, 22607 Hamburg, Germany
[3] Universität Hamburg, Fachbereich Mathematik, 20146 Hamburg, Germany
✉samira.kabri@fau.de

Abstract. In this paper we investigate the use of Fourier Neural Operators (FNOs) for image classification in comparison to standard Convolutional Neural Networks (CNNs). Neural operators are a discretization-invariant generalization of neural networks to approximate operators between infinite dimensional function spaces. FNOs—which are neural operators with a specific parametrization—have been applied successfully in the context of parametric PDEs. We derive the FNO architecture as an example for continuous and Fréchet-differentiable neural operators on Lebesgue spaces. We further show how CNNs can be converted into FNOs and vice versa and propose an interpolation-equivariant adaptation of the architecture.

Keywords: neural operators · trigonometric interpolation · Fourier neural operators · convolutional neural networks · resolution invariance

1 Introduction

Neural networks, in particular CNNs, are a highly effective tool for image classification tasks. Substituting fully-connected layers by convolutional layers allows for efficient extraction of local features at different levels of detail with reasonably low complexity. However, neural networks in general are not resolution-invariant, meaning that they do not generalize well to unseen input resolutions. In addition to interpolation of inputs to the training resolution, various other approaches have been proposed to address this issue, see, e.g., [3,14,17]. In this work we focus on the interpretation of digital images as discretizations of functions. This allows to model the feature extractor as a mapping between infinite dimensional spaces with the help of so-called neural operators, see [13]. In Section 2, we use established results on Nemytskii operators to derive conditions for well-definedness, continuity, and Fréchet-differentiability of neural operators

* This work was supported by the European Union's Horizon 2020 programme, Marie Skłodowska-Curie grant agreement No. 777826. TR and MB acknowledge the support of the BMBF, grant agreement No. 05M2020. SK and MB acknowledge the support of the DFG, project BU 2327/19-1. This work was carried out while MB was with the FAU Erlangen-Nürnberg.

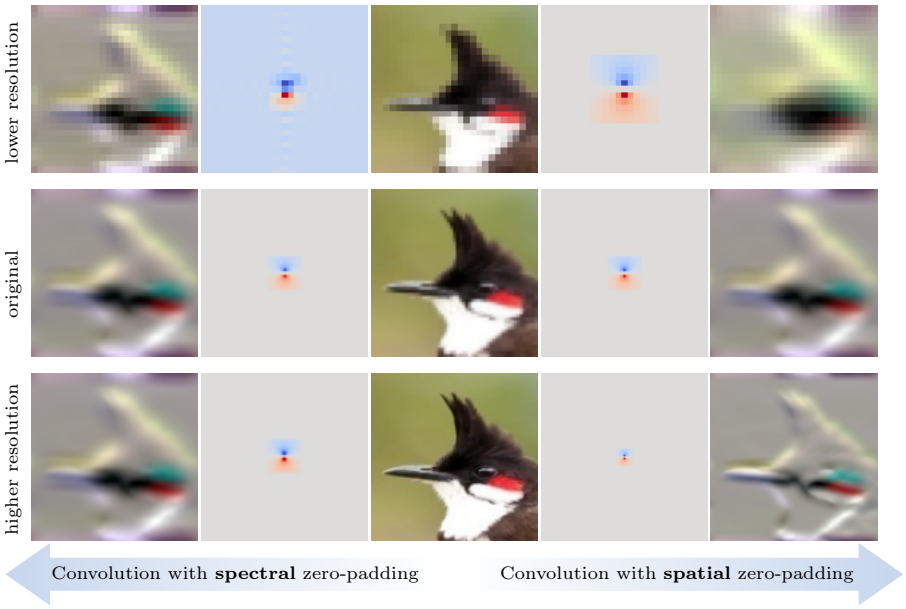


Fig. 1: Effects of applying a convolutional filter on the same image[4] with different resolutions. Spatial zero-padding (standard CNN-implementation) changes the relation of kernel support to image domain, while spectral zero-padding (FNO-implementation) captures comparable features for all resolutions.

on Lebesgue spaces. We specifically show these properties for the class of FNOs proposed in [15] as a discretization-invariant generalization of CNNs.

The key idea of FNOs is to parametrize convolutional kernels by their Fourier coefficients, i.e., in the spectral domain. Using trainable filters in the Fourier domain to represent convolution kernels in the context of image processing with neural networks has been studied with respect to performance and robustness in recent works, see e.g., [4,19,25]. In Section 3 we analyze the interchangeability of CNNs and FNOs with respect to optimization, parameter complexity, and generalization to varying input resolutions. While we restrict our theoretical derivations to real-valued functions, we note that they can be naturally extended to vector-valued functions as well. Our findings are supported by numerical experiments on the FashionMNIST [24] and Birds500 [18] data sets in Section 4.

2 Construction of Neural Operators on Lebesgue Spaces

2.1 Well-definedness and Continuity

A neural operator as defined in [13] is a composition of a finite but arbitrary number of so-called operator layers. In this section we derive conditions on the

[4] This image depicts a red whiskered bulbul taken from the Birds500 dataset [18].

components of an operator layer, such that it is a well-defined and continuous operator between two Lebesgue spaces. More precisely, for a bounded domain $\Omega \subset \mathbb{R}^d$ and $1 \leq p, q \leq +\infty$ we aim to construct a continuous operator $\mathcal{L} : L^p(\Omega) \to L^q(\Omega)$, such that an input function $u \in L^p(\Omega)$ is mapped to

$$\mathcal{L}(u)(x) = \sigma\left(\Psi(u)(x)\right) \qquad \text{for a.e. } x \in \Omega, \tag{1}$$

where we summarize all affine operations with an operator Ψ, such that

$$\Psi(u) = Wu + \mathcal{K}u + b. \tag{2}$$

Here, the weighting by $W \in \mathbb{R}$ implements a residual component and the kernel integral operator $\mathcal{K} : u \mapsto \int_\Omega \kappa(\cdot, y)\, u(y)\, dy$, determined by a kernel function $\kappa : \Omega \times \Omega \to \mathbb{R}$ generalizes the discrete weighting performed in neural networks. Analogously, the bias function $b : \Omega \to \mathbb{R}$ is the continuous counterpart of a bias vector. The (non-linear) activation function $\sigma : \mathbb{R} \to \mathbb{R}$ is applied pointwise and thus acts as a Nemytskii operator (see e.g., [6]). Thus, with a slight abuse of notation, the associated Nemytskii operator takes the form

$$\sigma : v \mapsto \sigma(v(\cdot)), \tag{3}$$

where we assume σ to be a measurable function. In order to ensure that the associated Nemytskii operator defines a mapping $\sigma : L^p(\Omega) \to L^q(\Omega)$ for $1 \leq p, q \leq \infty$ we require the following conditions to hold:

$$\begin{aligned} \underline{p, q < \infty} &: |\sigma(x)| \leq K + \beta |x|^{\frac{p}{q}} \text{ for all } x \in \mathbb{R} \text{ and constants } \beta, K \in \mathbb{R}, \\ \underline{p = \infty} &: |\sigma(x)| \leq K(c) \text{ for every } c > 0 \text{ for all } x, |x| < c \\ &\qquad \text{and a constant } K(c) \in \mathbb{R} \text{ depending on } c, \\ \underline{p < \infty, q = \infty} &: |\sigma(x)| \leq K \text{ for all } x \in \mathbb{R} \text{ and a constant } K \in \mathbb{R}, \end{aligned} \tag{4}$$

which were used in [6].

Lemma 1. *For $1 \leq p, q \leq \infty$ assume that σ fulfills (4). Then we have that the associated Nemytskii operator is a mapping $\sigma : L^p(\Omega) \to L^q(\Omega)$.*

Proof. Similar to [6, Th. 1], follows directly by employing the estimates in (4).

Since we are interested in continuity properties of the layer in (1) we consider the following continuity result for Nemytskii operators.

Lemma 2. *For $1 \leq p \leq \infty, 1 \leq q < \infty$ assume that the function $\sigma : \mathbb{R} \to \mathbb{R}$ is continuous and uniformly continuous in the case $q = \infty$. If the associated Nemytskii operator is a mapping $\sigma : L^p(\Omega) \to L^q(\Omega)$ then it is continuous.*

Proof. For $q < \infty$ the proof can be adapted from [23, p. 155-158]. For the case $q = \infty$ we refer to [6, Th. 5].

Remark 1. For $1 \leq p \leq q < \infty$ it is sufficient for σ to be p/q-Hölder continuous or locally Lipschitz continuous for $p, q = \infty$. In that case the Hölder and respectively the Lipschitz continuity transfers to the Nemytskii operator, see [22].

Example 1. The ReLU (Rectified Linear Unit, see [5]) function $\sigma(x) = \max(0, x)$ generates a continuous Nemytskii operator $\sigma : L^p(\Omega) \to L^q(\Omega)$ for any $p \geq q$. To show this, we note that the function σ is Lipschitz-continuous and with $p \geq q$ we have for all $x \in \mathbb{R}$ that $|\sigma(x)| \leq |x| \leq 1 + |x|^{\frac{p}{q}}$.

Proposition 1. *For $1 \leq p, q \leq \infty$ let $\mathcal{L}$ be an operator layer given by (1) with an activation function $\sigma : \mathbb{R} \to \mathbb{R}$. If there exists $r \geq 1$ such that*

(i) the affine part defines a mapping $\Psi : L^p(\Omega) \to L^r(\Omega)$,
(ii) the activation funtion σ generates a Nemytskii operator $\sigma : L^r(\Omega) \to L^q(\Omega)$,

then it holds that $\mathcal{L} : L^p(\Omega) \to L^q(\Omega)$. If additionally Ψ is a continuous operator on the specified spaces and the function σ is continuous, or uniformly continuous in the case $q = \infty$, the operator $\mathcal{L} : L^p(\Omega) \to L^q(\Omega)$ is also continuous.

Proof. With the assumptions on σ we directly have $\mathcal{L} = \sigma \circ \Psi : L^p(\Omega) \to L^q(\Omega)$. The continuity of $\mathcal{L}$ follows from Lemma 2.

Example 2. On the periodic domain $\Omega = \mathbb{R}/\mathbb{Z}$ consider an affine operator Ψ as defined in (2), where the integral operator is a convolution operator, i.e., $\kappa(x, y) = \kappa(x - y)$ with a slight abuse of notation. If for $1 \leq p, r, s \leq \infty$ we have that $\kappa \in L^s(\Omega)$ with $1/r + 1 = 1/p + 1/s$, it follows from Young's convolution inequality (see e.g., [8, Th. 1.2.12]) that $\mathcal{K} : L^p(\Omega) \to L^r(\Omega)$ is continuous. If further $b \in L^r(\Omega)$ and $W = 0$ in the case $r > p$, it follows directly that $\Psi : L^p(\Omega) \to L^r(\Omega)$ is continuous.

2.2 Differentiability

To analyze the differentiability of the neural operator layers we first transfer the result for general Nemytskii operators from [6, Th. 7] to our setting.

Theorem 1. *Let $1 \leq q < p < \infty$ or $q = p = \infty$ and $\sigma : \mathbb{R} \to \mathbb{R}$ a continuously differentiable function. Furthermore, let the Nemytskii operator associated to the derivative σ' be a continuous operator $\sigma' : L^p(\Omega) \to L^s(\Omega)$, with coefficient $s = pq/(p - q)$ for $q < p$ and $s = \infty$ for $q = p = \infty$. Then, the Nemytskii operator associated to σ is Fréchet-differentiable and its Fréchet-derivative $D\sigma(v) : L^p(\Omega) \to L^q(\Omega)$ in $v \in L^p(\Omega)$ is given by*

$$D\sigma(v)(h) = \sigma'(v) \cdot h, \qquad \text{for all } h \in L^p(\Omega).$$

Since the ReLU activation function from Example 1 is not differentiable, it does not fulfill the requirements of Theorem 1. An alternative is the so-called Gaussian Error Linear Unit (GELU), proposed in [10].

Example 3. The GELU function $\sigma(x) = x\,\Phi(x)$, where Φ denotes the cumulative distribution function of the standard normal distribution, generates a Fréchet-differentiable Nemytskii operator with derivative $D\sigma(v) : L^p(\Omega) \to L^q(\Omega)$ for any $p \geq q$ and $v \in L^p(\Omega)$. To show this, we compute $\sigma'(x) = \Phi(x) + x\phi(x)$, where $\phi(x) = \Phi'(x)$ is the standard normal distribution. We see that σ' is continuous and further $|\sigma'(x)| \leq 1 + |x|/\sqrt{2\pi} \leq 1 + 1/\sqrt{2\pi} + |x|^{\frac{p}{q}}/\sqrt{2\pi}$ for all $p \geq q$.

Proposition 2. *For $1 \leq p, q \leq \infty$, let $\mathcal{L}$ be an operator layer given by (1) with affine part Ψ as in (2). If there exists $r > q$, or $r = q = \infty$ such that*

(i) the affine part is a continuous operator $\Psi : L^p(\Omega) \to L^r(\Omega)$,
(ii) the activation function $\sigma : \mathbb{R} \to \mathbb{R}$ is continuously differentiable
(iii) and the derivative of the activation function generates a Nemytskii operator $\sigma' : L^r(\Omega) \to [L^r(\Omega) \to L^s(\Omega)]$ with $s = rq/(r-q)$,

then it holds that $\mathcal{L} : L^p(\Omega) \to L^q(\Omega)$ is Fréchet-differentiable in any $v \in L^p(\Omega)$ with Fréchet-derivative $D\mathcal{L}(v) : L^p(\Omega) \to L^q(\Omega)$

$$D\mathcal{L}(v)(h) = \sigma'(\Psi(v)) \cdot \tilde{\Psi}(h),$$

where $\tilde{\Psi}$ denotes the linear part of Ψ, i.e., $\tilde{\Psi} = \Psi - b$.

Proof. Theorem 1 yields that $D\sigma(v) : L^r(\Omega) \to L^q(\Omega)$ is well defined and continuous for $v \in L^r(\Omega)$. Fréchet-differentiability of linear and continuous operators on Banach spaces (see e.g., [1, Ex. 1.3]) yields the continuity of $\Psi : L^p(\Omega) \to L^r(\Omega)$ in all $v \in L^p(\Omega)$ with $D\Psi(v)(h) = \tilde{\Psi}(h)$. The claim follows from the chain-rule for Fréchet-differentiable operators, see [1, Prop. 1.4 (ii)].

For $p < q$ Fréchet-differentiability of a Nemytskii operator implies that the generating function is constant, and respectively affine linear for $p = q < \infty$, see [6, Ch 3.1]. Therefore, unless $p = \infty$, Fréchet-differentiability of neural operators with non-affine linear activation functions is only achieved at the cost of mapping the output of the affine part into a less regular space.

Example 4. For a continuous convolutional neural operator layer as constructed in Example 2, we consider a parametrization of the kernel function by a set of parameters $\hat{\theta} = \{\hat{\theta}_k\}_{k \in I} \subset \mathbb{C}$, where I is a finite set of indices, such that

$$\kappa_{\hat{\theta}}(x) = \sum_{k \in I} \hat{\theta}_k \, b_k(x), \tag{5}$$

with Fourier basis functions $b_k(x) = \exp(2\pi i \, kx)$ for $x \in \Omega$. Effectively, this amounts to parametrizing the kernel function by a finite number of Fourier coefficients. The resulting linear operator and the operator layer are denoted by $\Psi_{\hat{\theta}}$ and $\mathcal{L}_{\hat{\theta}}$. We note that FNOs proposed in [15] are neural operators that consist of such layers. It is easily seen that the kernel function defined by (5) is bounded and thus $\kappa_{\hat{\theta}} \in L^\infty(\Omega)$. Therefore, for suitable activation functions, Proposition 2 yields Fréchet-differentiability of $\mathcal{L}_{\hat{\theta}}$ with respect to its input function v, which was similarly observed in [16]. Additionally, for fixed v we consider the operator $\mathcal{L}_{(\cdot)}(v) : \hat{\theta} \mapsto \mathcal{L}_{\hat{\theta}}(v)$ which maps a set of parameters to a function. With the arguments from Proposition 2 we derive the partial Fréchet-derivatives of an FNO-layer with respect to its parameters for $h_k = (1 + i) \, e_k$ as

$$D_{\hat{\theta}_k} \mathcal{L}_{\hat{\theta}}(v) := D\mathcal{L}_{\hat{\theta}}(v)(h_k) = \sigma'(\Psi_{\hat{\theta}}(v)) \, D\Psi_{\hat{\theta}}(v)(h_k),$$

where e_k denotes the k-th canonical basis vector. Computing the Fréchet-derivative of Ψ in the sense of Wirtinger calculus ([20, Ch. 1]), this can be rewritten as $D_{\hat{\theta}_k} \mathcal{L}_{\hat{\theta}}(v) = \sigma'(\Psi_{\hat{\theta}}(v)) \, \hat{v}_k \, \bar{b}_k$, where $\hat{v}_k$ denotes the k-th Fourier coefficient of v. Here, for a complex number $z \in \mathbb{C}$, we denote by $\bar{z}$ its complex conjugate.

3 Connections to Convolutional Neural Networks

In this section we analyze the connection between FNOs and CNNs. Thus, for the remainder of this work, we set the domain to be the d-dimensional torus, i.e., $\Omega = \mathbb{R}^d/\mathbb{Z}^d$. As described in Example 4, the main idea of FNOs is to parametrize the convolution kernel by a finite number of Fourier coefficients $\hat{\theta} = \{\hat{\theta}_k\}_{k\in I} \subset \mathbb{C}$, where $I \subset \mathbb{Z}^d$ is a finite set of indices. Making use of the convolution theorem, see e.g., [8, Prop. 3.1.2 (9)], the kernel integral operator can then be written as

$$\mathcal{K}_{\hat{\theta}} v = \mathcal{F}^{-1}\left(\hat{\theta} \cdot \mathcal{F} v\right), \tag{6}$$

where $\mathcal{F}\colon [\Omega \to \mathbb{C}] \to \left[\mathbb{Z}^d \to \mathbb{C}\right]$ denotes the Fourier transform on the torus (see e.g., [8, Ch. 3]) and $\cdot$ denotes elementwise multiplication in the sense that

$$\left(\hat{\theta} \cdot \mathcal{F} v\right)_k = \begin{cases} \hat{\theta}_k \ (\mathcal{F} v)_k & \text{for } k \in I, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

We only consider parameters such that $\mathcal{K}$ maps real-valued functions to real-valued functions. This is equivalent to Hermitian symmetry, i.e., $\hat{\theta}_k = \overline{\hat{\theta}_{-k}}$ and in particular $\hat{\theta}_0 \in \mathbb{R}$. As proposed in [15], for $N \in \mathbb{N}$ we choose the set of multi-indices $I_N := \{-\lceil (N-1)/2 \rceil, \dots, 0, \dots, \lfloor (N-1)/2 \rfloor\}^d$, which corresponds to parametrizing the N lowest frequencies in each dimension. This is in accordance to the universal approximation result for FNOs derived in [12]. At this point, we assume N to be an odd number to avoid problems with the required symmetry and expand the approach to even choices of N in Section 3.3. Although an FNO is represented by a finite number of parameters, a discretization of (6) is needed to process discrete data, e.g., digital images. We therefore define the set of spatial multi-indices $J_N := \{0, \dots, N-1\}^d$ and write $v \in \mathbb{R}^{J_N}$ for mappings $v : J_N \to \mathbb{R}$. Furthermore, we discretize the Fourier transform for $v \in \mathbb{R}^{J_N}$ as

$$(Fv)_k = \frac{1}{\lambda} \sum_{j \in J_N} v_j \, e^{-2\pi i \left\langle k, \frac{j}{N} \right\rangle} \qquad \text{for all } k \in I_N$$

and its inverse for $\hat{v} \in \mathbb{C}^{I_N}$ as

$$\left(F^{-1}\hat{v}\right)_j = \frac{\lambda}{|J_N|} \sum_{k \in I_N} \hat{v}_k \, e^{2\pi i \left\langle k, \frac{j}{N} \right\rangle} \qquad \text{for all } j \in J_N,$$

where $\lambda \in \{1, \sqrt{|J_N|}, |J_N|\}$ determines the normalization factor. The discretized convolution operator parametrized by $\hat{\theta} \in \mathbb{C}^{I_N}_{\text{sym}} := F(\mathbb{R}^{J_N})$ is then defined by

$$K(\hat{\theta})(v) = F^{-1}\left(\hat{\theta} \cdot F v\right) \qquad \text{for } v \in \mathbb{R}^{J_N}.$$

For the remainder of this work, we refer to the above implementation of convolution as the FNO-implementation. In the following we compare the FNO-implementation to the standard implementation of the convolution of θ and

$v \in \mathbb{R}^{J_N}$ in a conventional CNN, which can be expressed as

$$C(\theta)(v)_j = \sum_{\tilde{\jmath} \in J_N} \theta_{j-\tilde{\jmath}}\, v_{\tilde{\jmath}} \qquad \text{for all } j \in J_N.$$

For the sake of simplicity, we handle negative indices by assuming that the values can be perpetuated periodically, although this is usually not done in practice.

3.1 Extension to Higher Input-Dimensions by Zero-Padding

So far, the presented implementations of convolution require the dimensions of the parameters θ, or $\hat{\theta}$ and the input v to coincide. In accordance to (7), the authors of [15] propose to handle dimension mismatches by zero-padding of the spectral parameters. More precisely, a low-dimensional set of parameters $\hat{\theta} \in \mathbb{C}^{I_M}$ is adapted to an input $v \in \mathbb{R}^{J_N}$ with odd $N \in \mathbb{N}$ by setting

$$\hat{\theta}_k^{M \to N} = \begin{cases} \hat{\theta}_k & \text{for } k \in I_N \cap I_M, \\ 0 & \text{for } k \in I_N \backslash I_M. \end{cases}$$

Since we choose N to be odd, the required symmetry is not hurt by the above operation. The extended FNO-implementation of the convolution is then given for $\hat{\theta} \in \mathbb{C}^{I_M}$ and $v \in \mathbb{R}^{J_N}$ by

$$K(\hat{\theta})(v) := K(\hat{\theta}^{M \to N})(v).$$

Analogously, in the conventional CNN-implementation the convolution of parameters $\theta \in \mathbb{R}^{J_M}$ and $v \in \mathbb{R}^{J_N}$ with $N \geq M$ is computed as

$$C(\theta)(v) := C(\theta^{M \to N})(v),$$

where again, $\theta^{M \to N} \in \mathbb{R}^{J_M}$ denotes the zero-padded version of θ. We stress that, although the technique to generalize the implementations to higher input dimensions is the same, the outcome differs substantially. This was already mentioned in [13, Sec. 4] and is discussed further in Section 3.4.

3.2 Convertibility and Complexity

Deriving FNOs from convolutional neural operators using the convolution theorem suggests that there is a way to convert one implementation of convolution into the other as long as the input dimension is fixed. The following Lemma shows that this is indeed possible.

Lemma 3. *Let $M \leq N$ both be odd and let $T : \mathbb{R}^{J_N} \to \mathbb{C}^{I_N}$ be defined for $\theta \in \mathbb{R}^{J_N}$ as $T(\theta) = \lambda F(\theta)$. For any $\theta \in \mathbb{R}^{J_M}$ and $v \in \mathbb{R}^{J_N}$ it holds true that*

$$C(\theta)(v) = K(T(\theta^{M \to N}))(v)$$

and for any $\hat{\theta} \in \mathbb{C}^{I_M}_{sym}$ and $v \in \mathbb{R}^{J_N}$ it holds true that

$$K(\hat{\theta})(v) = C(T^{-1}(\hat{\theta}^{M \to N}))(v).$$

Proof. By the definition of the extension to higher input dimensions we can assume $M = N$. For $\theta, v \in \mathbb{R}^{J_N}$ we derive the discrete analogon of the convolution theorem by inserting the definitions of the discrete Fourier transform as

$$F(C(\theta)(v)) = \lambda\, F(\theta) \cdot F(v).$$

Employing that $F : \mathbb{R}^{J_N} \to \mathbb{C}^{I_N}_{\text{sym}}$ is a bijection, it follows that

$$C(\theta)(v) = F^{-1}(\lambda\, F(\theta)\, F(v)) = K(T(\theta))(v).$$

The second statement can be proven analogously. We note that T^{-1} is well-defined since $\lambda \geq 1$ for odd N.

Although the above Lemma proves convertibility for a fixed set of parameters and fixed input dimensions, a conversion can increase the amount of required parameters, as in general, the dimension of the converted parameters has to match the input dimension. It becomes clear that spatial locality cannot be enforced with the proposed FNO-parametrization and spectral locality cannot be enforced with the CNN-parametrization. Therefore, different behavior during the training process is to be expected if the parameter size does not match the input size. Moreover, the following Lemma shows that even for matching dimensions, equivalent behavior for gradient-based optimization like steepest descent requires careful adaptation of the learning rate, since the computation of gradients is not equivariant with respect to the function T.

Lemma 4. *For odd $N \in \mathbb{N}$ and $v, \theta \in \mathbb{R}^{J_N}$ and $\hat{\theta} = T(\theta)$ it holds true that*

$$\nabla_{\hat{\theta}} K(\hat{\theta})(v) = \frac{1}{|J_N|}\, T\left(\nabla_\theta C(\theta)(v)\right).$$

Proof. Inserting $\hat{\theta} = T(\theta)$ it follows with the chain rule from Lemma 3 that

$$\frac{\partial K(\hat{\theta})(v)_l}{\partial \hat{\theta}_k} = \sum_{j \in J_N} \frac{\partial C(\theta)(v)_l}{\partial \theta_j} \frac{\partial T^{-1}(\hat{\theta})_j}{\partial \hat{\theta}_k} = \frac{1}{|J_N|} \sum_{j \in J_N} \frac{\partial C(\theta)(v)_l}{\partial \theta_j}\, e^{-i2\pi\langle k, \frac{j}{N}\rangle}$$

for $k \in I_N$. The claim now follows by inserting the definition of T.

3.3 Adaptation to Even Dimensions

For the remainder of this paper we consider the special case $\Omega = \mathbb{R}^2/\mathbb{Z}^2$ and adapt the FNO-implementation to even dimensions. For odd dimensions M, N, zero-padding of a set of spectral coefficients does not violate the requirement $\hat{\theta}^{M \to N} \in \mathbb{C}^{I_N}_{\text{sym}}$. This property is lost in general for even dimensions. Since for odd dimensions, zero-padding in the spectral domain is equivalent to trigonometric interpolation, we perform the adaptation of dimensions such that $\hat{\theta}^{M \to N}$ is a trigonometric interpolator of a real-valued function (see [2] for an exhaustive study on this topic). In practice, this means splitting the coefficients corresponding to the Nyquist frequencies to interpolate from an even dimension to

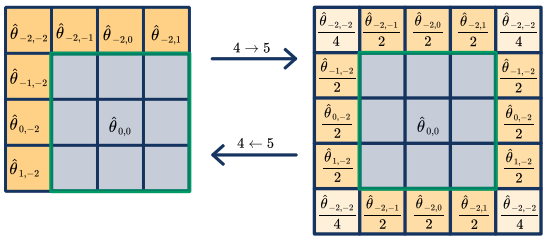


Fig. 2: Nyquist splitting for spectral parameters to extend real-valued trigonometric interpolation to even dimensions.

the next higher odd dimension, or to invert this splitting to interpolate from an odd dimension to the next lower even dimension (see Figure 2). The real-valued trigonometric interpolation of $v \in \mathbb{R}^{J_M}$ to a dimension N is then given by

$$v^{M \xrightarrow{\Delta} N} := F^{-1}\left((Fv)^{M \to N}\right).$$

We extend the FNO-implementation to parameters $\hat{\theta} \in \mathbb{C}^{I_N}_{\mathrm{sym}}$ and inputs $v \in \mathbb{R}^{J_N}$ with even $N \in \mathbb{N}$ by defining

$$K(\hat{\theta})(v) := \left(K(\hat{\theta}^{N \to \tilde{N}})(v^{N \xrightarrow{\Delta} \tilde{N}})\right)^{\tilde{N} \xrightarrow{\Delta} N}, \tag{8}$$

where $\tilde{N} = N + 1$. We note that by this choice we lose the direct convertibility to the CNN-implementation as in general for even dimensions

$$K(\hat{\theta})(v) \neq F^{-1}(T^{-1}(\hat{\theta})\, Fv),$$

as the right hand side corresponds to zero-padding of the spectral coefficients regardless of the oddity of the dimensions. However, we can still convert the FNO-implementation to the CNN-implementation and vice versa, by adapting the magnitude of coefficients to the effects of the Nyquist splitting.

3.4 Interpolation Equivariance

Our motivation to perform the adaptation to even dimension as proposed in the preceding section, is that the resulting implementation of convolution is equivariant with respect to (real-valued) trigonometric interpolation.

Corollary 1. *For $\hat{\theta} \in \mathbb{C}^{I_M}_{sym}$, $v \in \mathbb{R}^{J_N}, M \leq N$ it holds true for any $L \geq M$ that*

$$K(\hat{\theta})(v^{N \xrightarrow{\Delta} L}) = \left(K(\hat{\theta})(v)\right)^{N \xrightarrow{\Delta} L}.$$

Proof. We first note that it holds for any choice of $M \leq N, L$ that

$$K(\hat{\theta})(v^{N \xrightarrow{\Delta} L}) = \left(K(\hat{\theta}^{M \to \tilde{M} \to \tilde{L}})(v^{N \xrightarrow{\Delta} \tilde{N} \xrightarrow{\Delta} \tilde{L}}) \right)^{\tilde{L} \xrightarrow{\Delta} L},$$

where $\tilde{L} = L + (1 - L\%2)$, $\tilde{M} = M + (1 - M\%2)$, $\tilde{N} = N + (1 - N\%2)$ and $\%$ denotes the modulo operation. Therefore, we can assume L, M and N to be odd without loss of generality and thus $\tilde{L} = L, \tilde{M} = M$ and $\tilde{N} = N$. Regarding the discrete Fourier coefficients then reveals that

$$F(K(\hat{\theta})(v^{N \xrightarrow{\Delta} L}))_k = \begin{Bmatrix} \hat{\theta}_k \, F(v)_k & \text{for } k \in I_M, \\ 0 & \text{otherwise} \end{Bmatrix} = F((K(\hat{\theta})(v))^{N \xrightarrow{\Delta} L})_k.$$

Applying the inverse Fourier transform completes the proof.

4 Numerical Examples

In this section we compare the discussed implementations of convolution numerically in the context of image classification.[5] Here, the task is to assign a label from $s \in \mathbb{N}$ possible classes to a given image $v : [0,1]^2 \to \mathbb{R}^{n_c}$, with $n_c \in \mathbb{N}$ denoting the number of color channels. Solving this task numerically requires discrete input images of the form $v^N = v|_{J_N/N} \in \mathbb{R}^{J_N \times n_c}$, where $N \in \mathbb{N}$ denotes the dimension. We note that since we consider a fixed function domain the dimension is proportional to the resolution. If we assume N to be fixed the network is a function $f_\theta : \mathbb{R}^{J_N \times n_c} \to \mathbb{R}^s$. Given a finite training set $D \subset \mathbb{R}^{J_N \times nc} \times \mathbb{R}^s$ we optimize the parameters θ by minimizing the empirical loss based on the cross-entropy [7, Ch. 3]. The networks we use for our experiments consist of several convolutional layers for feature extraction followed by one fully connected classification layer. To make all architectures applicable to inputs of any resolution, we insert an adaptive average pooling layer between the feature extractor and the classifier.

4.1 Expressivity for Varying Kernel Sizes

In the first experiment (see Fig. 3) we train a CNN without any residual components on the FashionMNIST[6] dataset. The network has two convolutional layers with periodic padding and without striding, followed by an adaptive pooling layer and a linear classifier. Since we do not observe major performance changes on the test set for different kernel sizes, we conclude that on this data set the expressivity of the small kernel architectures is comparable to large kernel architectures. We then convert the convolutional layers of the CNNs with 3×3- and 28×28-kernels to FNO-layers, employing varying numbers of spectral parameters. Here, we observe decreasing performance with smaller spectral kernel sizes,

[5] Our code is available online: github.com/samirak98/FourierImaging.
[6] This dataset consists of 60,000 training and 10,000 test 28×28 images (grayscale).

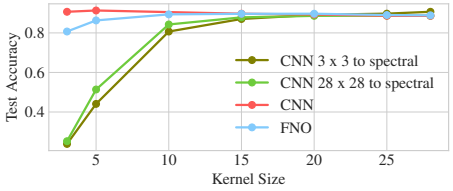


Fig. 3: Test accuracy of CNNs and FNOs for varying kernel sizes.

indicating that the learned spatial kernels cannot be expressed well by fewer frequencies. However, in this example, training an FNO with the same structure almost closes this performance gap. This implies the existence of low frequency kernels with sufficient expressivity. We refer to [11] for a study on training with a spectral parametrization.

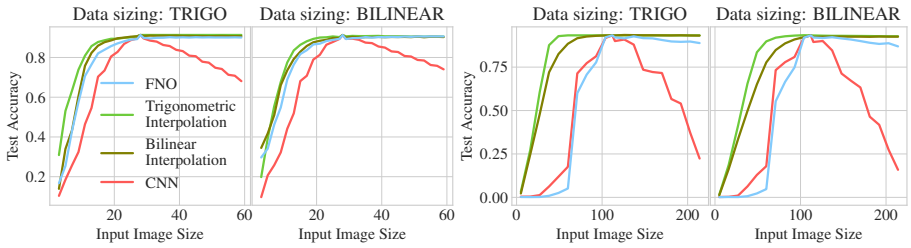
4.2 Resolution Invariance

In the second experiment, we investigate the resolution invariance of the different convolution implementations. In Fig. 4a we compare the accuracy on test data resized to different resolutions with trigonometric, or bilinear interpolation, respectively. Here, CNN refers to the conventional CNN-implementation with 5×5 kernel, where dimension mismatches are compensated for by spatial zero-padding of the kernel. FNO refers to the FNO-implementation, where the kernels are adapted to the input dimension by trigonometric interpolation. Additionally, we show the behavior of the CNN for inputs rescaled to the training resolution. Applying trigonometric interpolation before a convolutional layer can be interpreted as an FNO-layer with predetermined output dimensions.

The performance of the CNN varies drastically with the input dimension and peaks for the resolution it was trained on. This result is in accordance with the effect showcased in Fig. 1: Dimension adaption via spatial zero-padding modifies the locality of the kernel and consequently captures different features for different resolutions. While trigonometric interpolation performs best, we see that the FNO adapts very well. In particular, the performance for higher input resolutions deters only slightly, which is not the case for the standard CNN.

Additionally (see Fig. 4b), we train a ResNet18 [9] on the Birds500 data set[7] with a reduced training size of 112×112. To regularize the generalization to different resolutions, especially for the FNO-implementation, we replace the standard striding operations by trigonometric downsampling. Compared to the first experiment it stands out that the FNO performs worse for inputs with resolutions below 112×112, but only slightly diminishes for higher resolutions. We attribute this fact to the dimension reduction operations in the architecture.

[7] We employ a former version of the data set, which consists of $76,262$ RGB images for training and $2,250$ images for testing of size 224×224, where the task is to classify birds out of 450 possible classes.



(a) FMNIST (b) Birds500

Fig. 4: Performance for different interpolation methods on test data that has been resized with the interpolation method denoted on top of the plots.

5 Conclusion and Outlook

In this work, we have studied the regularity of neural operators on Lebesgue spaces and investigated the effects of implementing convolutional layers in the sense of FNOs. Based on the theoretical derivation of the convertibility from standard CNNs to FNOs, our numerical experiments show that it is possible to convert a network that was trained with the standard CNN architecture into an FNO. By this, we could combine the benefits of both approaches: Enforced spatial locality with a small number of parameters during training and an implementation that generalizes well to higher input dimensions during the evaluation. However, we have seen that the trigonometric interpolation of inputs outperforms all other considered approaches. In future work, we want to investigate how the ideas of FNOs and trigonometric interpolation can be incorporated into image-to-image architectures like U-Nets as proposed in [21]. Additionally, we want to further explore the effects of training in the spectral domain, for example with respect to adversarial robustness.

References

1. Ambrosetti, A., Prodi, G.: A Primer of Nonlinear Analysis. Cambridge University Press (1993)
2. Briand, T.: Trigonometric polynomial interpolation of images. Image Processing On Line **9**, 291–316 (10 2019)
3. Cai, D., Chen, K., Qian, Y., Kämäräinen, J.K.: Convolutional low-resolution fine-grained classification. Pattern Recognition Letters **119**, 166–171 (2019)
4. Chi, L., Jiang, B., Mu, Y.: Fast fourier convolution. Advances in Neural Information Processing Systems **33**, 4479–4488 (2020)
5. Fukushima, K.C.: Cognitron: A self-organizing multilayered neural network. Biol. Cybernetics **20**, 121–136 (1975)
6. Goldberg, H., Kampowsky, W., Tröltzsch, F.: On Nemytskij operators in lp-spaces of abstract functions. Mathematische Nachrichten **155**(1), 127–140 (1992)
7. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016)

8. Grafakos, L.: Classical Fourier Analysis. Graduate Texts in Mathematics, Springer, New York, NY, 3 edn. (2014)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE CVPR. pp. 770–778 (2016)
10. Hendrycks, D., Gimpel, K.: Gaussian error linear units (GELUs). arXiv:1606.08415 (2016)
11. Johnny, W., Brigido, H., Ladeira, M., Souza, J.C.F.: Fourier neural operator for image classification. In: 2022 17th Iberian Conference on Information Systems and Technologies (CISTI). pp. 1–6 (2022)
12. Kovachki, N.B., Lanthaler, S., Mishra, S.: On universal approximation and error bounds for fourier neural operators. Journal of Machine Learning Research (2022)
13. Kovachki, N.B., Li, Z., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A.M., Anandkumar, A.: Neural operator: Learning maps between function spaces. arXiv:2108.08481 (2021)
14. Koziarski, M., Cyganek, B.: Impact of low resolution on image recognition with deep neural networks: An experimental study. International Journal of Applied Mathematics and Computer Science **28**(4), 735–744 (2018)
15. Li, Z., Kovachki, N.B., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A.M., Anandkumar, A.: Fourier neural operator for parametric partial differential equations. In: 9th International Conference on Learning Representations (ICLR) (2021)
16. Li, Z., Zheng, H., Kovachki, N., Jin, D., Chen, H., Liu, B., Azizzadenesheli, K., Anandkumar, A.: Physics-informed neural operator for learning partial differential equations. arXiv preprint arXiv:2111.03794 (2021)
17. Peng, X., Hoffman, J., Stella, X.Y., Saenko, K.: Fine-to-coarse knowledge transfer for low-res image classification. In: 2016 IEEE International Conference on Image Processing (ICIP). pp. 3683–3687. IEEE (2016)
18. Piosenka, G.: Birds 500 - species image classification (2021), https://www.kaggle.com/datasets/gpiosenka/100-bird-species
19. Rao, Y., Zhao, W., Zhu, Z., Lu, J., Zhou, J.: Global filter networks for image classification. Advances in Neural Information Processing Systems **34**, 980–993 (2021)
20. Remmert, R.: Theory of Complex Functions. Springer New York, New York, NY (1991)
21. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. pp. 234–241. Springer International Publishing, Cham (2015)
22. Tröltzsch, F.: Optimal Control of Partial Differential Equations: Theory, Methods, and Applications, Graduate Studies in Mathematics, vol. 112. American Mathematical Society, Providence, Rhode Island (2010)
23. Vaĭnberg, M.M.: Variational method and method of monotone operators in the theory of nonlinear equations. No. 22090, John Wiley & Sons (1974)
24. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv:1708.07747 (2017)
25. Zhou, M., Yu, H., Huang, J., Zhao, F., Gu, J., Loy, C.C., Meng, D., Li, C.: Deep fourier up-sampling. arxiv:2210.05171 (2022)

Bibliography

Articles

[RB22] T. Roith and L. Bungert. "Continuum limit of Lipschitz learning on graphs." In: *Foundations of Computational Mathematics* (2022), pp. 1–39.

[BCR21] L. Bungert, J. Calder, and T. Roith. "Uniform convergence rates for Lipschitz learning on graphs." In: *arXiv preprint arXiv:2111.12370* (2021).

[Bun+22] L. Bungert et al. "A bregman learning framework for sparse neural networks." In: *Journal of Machine Learning Research* 23.192 (2022), pp. 1–43.

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, dass alle Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, als solche kenntlich gemacht sind und dass die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegt wurde.

Erlangen, den 21.März 2023 .....

Tim Roith