Consistency, Robustness and Sparsity for Learning Algorithms

Konsistenz, Robustheit und Dünnbesetztheit von Lernalgorithmen

Der Naturwissenschaftlichen Fakultät
der
Friedrich-Alexander-Universität Erlangen-Nürnberg

zur Erlangung des Doktorgrades
Dr. rer. nat.

vorgelegt von
Tim Roith
aus
Amberg

Als Dissertation genehmigt von der Naturwissenschaftlichen Fakultät der Friedrich-Alexander-Universität Erlangen-Nürnberg

Tag der mündlichen Prüfung:	20.03.2024
Gutachter*in:	Martin Burger
	Dejan Slepčev
	Franca Hoffmann

Acknowledgement

The three years of my PhD have been a thrilling period of learning, both in terms of scientific knowledge and personal growth. This journey allowed me to explore exciting research directions, travel to various parts of the world, and, most significantly, meet a lot of people. I want to thank all of my colleagues at FAU Erlangen-Nürnberg for making the experience at the Chair of Applied Mathematics what it was. I especially want to express my gratitude to the following people:

- Martin Burger for granting me a student assistant position back in 2018, even though I hail from the Upper Palatinate. Since then, I have relished the unique blend of freedom to express myself, coupled with dependable support whenever required.

- Daniel Tenbrinck for being the heart of our former chair at FAU, serving as the driving force behind every important social event, being an inherently helpful person, setting up the GPU servers, and, most importantly, for winning the Best Teaching Award in 2021 and giving me credit for it. During my time at FAU, we worked a lot together, especially on teaching-related topics, which always turned out to be a great success. I miss the idea of preparing lectures together. I hope we can collaborate in the future, and I am sure we'll see each other whenever time allows.

- Leon Bungert for basically co-supervising my PhD and for teaching me a lot about mathematics, culture, cooking, and colorful shirts. Given the wealth of experiences we've shared during the last four years, it's not easy to express all of this within a few lines. Apart from our collaborations on an academic level, I want to thank you for being a very close friend.

- Antonio Esposito for being a really nice guy, greatly improving my fake Italian accent, and, most of all, for being part of our group. I was really sad when you left for Oxford, and I hope to meet you again soon.

- Alex Rossi for sparking my interest in patches, teaching me how to dress austere and introducing me to authentic Italian desserts. I think you're a great guy, and I've always loved hanging out with you.

- Lukas Weigand for introducing me to the green book, helping with the exercise sheets for numerics, and teaching me a lot about gradient flows. I look forward to working together with you in the future.

- Lorenz Kuger for sharing the office with me for a long time, despite being the best-organized PhD student I know, not getting too annoyed about every non-work related activity I started in the office, and for being a great colleague. Transitioning to Hamburg was especially easy thanks to your pioneering a lot of bureaucratic ground.

- All the other colleagues I spent time with in Erlangen, or met around the world, namely:
  - Philipp Wacker for making me aware of the field of statistics,
  - Jannik Hausladen for helping me a lot with our MRI project,
  - Alexander Prechtel for always bringing the servers back to life (together with Daniel), when I crashed them,
  - Jeff Calder for collaborating with Leon and me on Lipschitz leaning and writing that very nice graph learning package, which made everything a lot easier,
  - Nicolás García Trillos for organizing a nice mini-symposium together with me in Tokyo,
  - and Konstantin Riedl for being yet another great mathematician from the Upper Palatinate.
- All my other colleagues and short-time office mates that just recently joined the Burger group, with whom I am now starting a new chapter in Hamburg.
- Astrid Bigott for always being understanding of my difficult relationship with bureaucracy and for finding all the mistakes I ever made in any kind of form.
- My proofreaders Leon, Norman, Martin, and Samira.
- My parents for always showing me affection and supporting me unconditionally. Although I am not always physically there, I will never lose my connection to home. Also, I want to extend my thanks to Miezer for always letting me use his pictures for imaging purposes.
- Samira Kabri for teaching me about FNOs.

Contents

List of Figures

Inhalt und Struktur

Diese Arbeit ist in zwei Hauptteile strukturiert, Part I, welche die Themen und Resultate der Publikation präsentiert und erklärt, welche in Part II erneut abgedruckt sind.

Part I: Exposition	Part II: Prints
Chapter 2: Learning Paradigms	—
Chapter 3: Consistent Semi-Supervised Learning on Sparse Graphs	Chapters P1 and P2
Chapter 4: Robust and Sparse Supervised Learning	Chapters P3 to P5

Einleitung und Motivation

Das Gebiet des maschinellen Lernens entstand in den 1950er Jahren, motiviert durch die Idee, einen Computer Algorithmen und Muster entdecken zu lassen, ohne sie explizit von Hand finden zu müssen. Nach der Anfangsphase und mehreren “AI-winters” [SG96] haben zahlreiche wichtige Entwicklungen – z. B. die Wiederentdeckung der “Backpropagation”, welche ursprünglich auf [Kel60; Ros62] zurückgeht und dann in [RHW86] popularisiert wurde, siehe z. B. [Sch22] – zur Relevanz der Lernmethoden beigetragen. Die Fortschritte im Bereich von Computer-Hardware, zusammen mit der Verfügbarkeit großer Datenmengen, haben schließlich den Enthusiasmus für maschinelles Lernen der letzten Jahre entfacht. Während “deep” Learning Methoden, d. h. Techniken, die mehrere neuronale Layer verwenden, wie sie ursprünglich in [Ros58] vorgeschlagen wurden, die prominentesten Beispiele sind, gibt es eine ganze Familie von lernbasierten Strategien, welche aktiv in Bereichen wie Computer Vision [Cha+21], Sprachverarbeitung [Khu+23] oder für medizinische Zwecke [She+22] angewendet werden. In dieser Arbeit konzentrieren wir uns hauptsächlich auf datenbasierte Ansätze, angewendet auf Klassifizierungsaufgaben, wobei die konkrete Modalität der gegebenen Daten unsere Strategie bestimmt. Wir konzentrieren uns auf überwachtes Lernen – der Datensatz besteht nur aus Eingabe-Ausgabe-Paaren, d. h., er ist vollständig gelabelt – und halbüberwachtes Lernen – die Daten sind nur teilweise gelabelt.

Beide datenbasierten Methoden waren vor allem in den letzten 20 Jahren sehr erfolgreich. Allerdings weisen die manchmal rein heuristischen Lernstrategien auch gravierende Nachteile auf. Beim überwachten Lernen ist man oft an der Generalisierung eines Klassifizierers interessiert, d. h. wie akkurat ist das Ergebnis auf ungesehenen Eingaben, die nicht Teil der Trainingsdaten sind. In [GSS14] wurde entdeckt, dass die Ausgaben des Klassifizierers durch kleine, scheinbar unsichtbare Störungen, die als *adversarial attacks* bekannt sind, vollständig verfälscht werden können. Allgemeiner führt uns dieses Phänomen zum Thema *Robustheit* unter Eingabestörungen. Nehmen wir an, dass ein Mensch und eine Maschine eine Eingabe x als vom Typ c einstufen würden. In einer eher vagen, aber anschaulichen Formulierung lautet die wichtigste Implikation, die wir für eine Eingabe $\bar{x}$ erhalten wollen

$$\left.\begin{array}{c}\bar{x} \text{ liegt nahe an } x,\\ \bar{x} \text{ wird von einem Menschen noch als } c \text{ eingestuft}\end{array}\right\} \Rightarrow \text{die Maschine stuft } \bar{x} \text{ als } c \text{ ein.}$$

Neben adversarial Examples gehört dazu auch das Ändern der Auflösung von Bildern, welche die Klassifizierung durch einen Menschen nicht verändern, sofern sie hinreichend klein sind. In jedem Fall zeigt das Vorhandensein dieser Störungen kritische Schwächen der Lernmethoden auf und erfordert ein besseres theoretisches Verständnis der verwendeten Modelle. An dieser Stelle wird die mathematische Grundlage des Fachgebiets relevanter und es kommen Eigenschaften ins Spiel, die über die Klassifizierungsleistung hinausgehen und die in dieser Arbeit diskutiert werden.

Im halb-überwachten Setting betrachten wir graphbasierte Algorithmen, wie sie ursprünglich in [ZGL03] mit dem Graph-Laplace vorgeschlagen wurden. Das Hauptproblem, das wir in dieser Arbeit hervorheben, wurde zuerst in [NSZ09] beobachtet, nämlich dass die Klassifizierungsleistung paradoxerweise mit steigender Dimension der Daten deutlich abnimmt. Es stellte sich heraus, dass die mit dem Graph-Laplace erhaltenen Lösungen über den gesamten Datensatz hinweg konstant sind, wenn die Dimension größer als zwei ist, was mit dem Sobolev Einbettungssatz [AF03] in Verbindung gebracht werden kann. Dieses Problem zeigt sich vor allem, wenn die Zahl der unbeschrifteten Datenpunkte gegen unendlich geht, was uns zu der Frage der *Konsistenz* für halb-überwachte Algorithmen führt.

Ein Problem, das für überwachte und halb-überwachte Algorithmen gleichermaßen gilt, ist der hohe Bedarf an Rechenressourcen. Das Training eines neuronalen Netzes erfordert in der Regel den Einsatz von GPUs über einen langen Zeitraum. Dies macht den Prozess einerseits für weniger leistungsfähige Maschinen oder sogar mobile Geräte undurchführbar und erzeugt andererseits große Mengen an CO_2-Emissionen [Hoe+21]. Für graphbasiertes, halb-überwachtes Lernen müssen zunächst Entfernungen zwischen vielen Datenpunkten berechnet werden, um Kantengewichte zu erhalten, was eine kostspielige Aufgabe ist. Außerdem skaliert die Rechenkomplexität verschiedener Probleme auf einem gegebenen Graphen mit der Anzahl der Kanten. Beispielsweise skaliert die Laufzeit von Dijkstras Algorithmus zur Berechnung kürzester Pfade in einem Graphen bereits linear mit der Anzahl der Kanten. In dieser Arbeit ist das Schlüsselwort zur Reduzierung der Rechenlast in beiden Fällen *Dünnbesetztheit*. Das Konzept von dünnbesetzten Matrizen ist tief in der numerischen linearen Algebra verwurzelt [Lan52; GV13] und besteht im Wesentlichen darin, Nullen in einer Matrix auszunutzen, um die Berechnungszeit zu beschleunigen. Bei neuronalen Netzen kann dies dadurch erreicht werden, dass die Gewichtsmatrizen der Layer dünnbesetzt sein müssen. Bei Graphen bedeutet eine dünnbesetzte Konnektivitätsmatrix einfach, dass nur eine kleine Anzahl an Kanten aktiv ist, was ebenfalls die Rechenkosten reduziert.

Beiträge in dieser Arbeit Anknüpfend an die zuvor genannten Themen befasst sich diese Arbeit mit *Konsistenz, Robustheit* und *Dünnbesetztheit* von überwachten und halb-überwachten Lernalgorithmen.

Für letztere betrachten wir hauptsächlich das sogenannte Lipschitz-Learning [NSZ09], für die wir Konvergenz und Konvergenzraten für diskrete Lösungen zu Lösungen im Kontinuum zeigen, wenn die Anzahl der Datenpunkte gegen unendlich geht. Dabei arbeiten wir mit Annahmen, welche sehr dünnbesetzte und daher rechnerisch attraktive Graphen zulässt.

Bei überwachtem Lernen befassen wir uns mit der Robustheit gegen adversarial Attacks und Auflösungsänderungen. Im ersten Fall schlagen wir einen effizienten Algorithmus vor, der die Lipschitz-Konstante [Lip77] eines neuronalen Netzes bestraft und ein damit robustes Netz trainiert. Im Multiresolution-Setting analysieren wir die Rolle von neuronalen Fourier-Operatoren, wie sie in [Li+21] vorgeschlagen wurden, und ihre Verbindung zu normalen Faltungsoperatoren [Fuk80]. Im Hinblick auf die Rechenkomplexität des Trainings neuronaler Netze schlagen wir einen auf Bregman Iterationen basierenden Algorithmus [Osh+05] vor, der dünnbesetzte Gewichtsmatrizen während des gesamten Trainings ermöglicht. Zusätzliche analysieren wir die Konvergenz der stochastische Adaption der ursprünglichen Bregman Iterationen.

Struktur der Exposition In Chapter 2 stellen wir die Lernparadigmen und Grundbegriffe vor, die in dieser Arbeit verwendet werden. Anschließend stellen wir in Chapter 3 die Themen zur Konsistenz beim halb-überwachten Lernen auf Graphen vor. Nach einer erläuternden Einführung heben wir die Hauptbeiträge von [LIP-I; LIP-II] hervor. Dabei versuchen wir Redundanz zu den Publikationen in Part II zu vermeiden und dennoch einen verständlichen Kontext zu ermöglichen. In Chapter 4 kommentieren wir die Themen zum überwachten Lernen. Nach einer zusätzlichen Einleitung enthält das Kapitel drei Abschnitte, in denen die Arbeiten [FNO; CLIP; BREG-I] einzeln vorgestellt werden. Schließlich werden in Chapter 5 die Inhalte der gesamten Arbeit zusammengefasst und mögliche zukünftige Richtungen aufgezeigt.

Publikationen und Beitragsauflistung

Die folgenden Arbeiten sind Teil dieser Dissertation und werden in Part II erneut abgedruckt.

[LIP-II] L. Bungert, J. Calder, and T. Roith. “Uniform convergence rates for Lipschitz learning on graphs.” In: *IMA Journal of Numerical Analysis* 43.4 (2022), pp. 2445–2495. DOI: 10.1093/imanum/drac048.

[LIP-I] T. Roith and L. Bungert. “Continuum limit of Lipschitz learning on graphs.” In: *Foundations of Computational Mathematics* 23.2 (2023), pp. 1–39. DOI: 10.1007/s10208-022-09557-9.

[CLIP] L. Bungert, R. Raab, T. Roith, L. Schwinn, and D. Tenbrinck. “CLIP: Cheap Lipschitz training of neural networks.” In: *Scale Space and Variational Methods in Computer Vision: 8th International Conference, SSVM 2021, Proceedings*. Springer. 2021, pp. 307–319. DOI: 10.1007/978-3-030-75549-2_25.

[BREG-I] L. Bungert, T. Roith, D. Tenbrinck, and M. Burger. “A Bregman learning framework for sparse neural networks.” In: *Journal of Machine Learning Research* 23.192 (2022), pp. 1–43.

[FNO] S. Kabri, T. Roith, D. Tenbrinck, and M. Burger. “Resolution-Invariant Image Classification based on Fourier Neural Operators.” In: *Scale Space and Variational Methods in Computer Vision: 9th International Conference, SSVM 2023, Proceedings*. Springer. 2023, pp. 307–319. DOI: 10.1007/978-3-031-31975-4_18.

Die folgenden Preprints sind kein Teil dieser Arbeit, geben aber zusätzliche Einsichten in die behandelten Themen.

[LIP-III] L. Bungert, J. Calder, and T. Roith. *Ratio convergence rates for Euclidean first-passage percolation: Applications to the graph infinity Laplacian*. 2022. arXiv: 2210.09023.

[BREG-II] L. Bungert, T. Roith, D. Tenbrinck, and M. Burger. *Neural Architecture Search via Bregman Iterations*. 2021. arXiv: 2106.02479.

Im Folgenden führen wir TRs Beiträge zu den oben genannten Publikationen auf.

[LIP-I]: Diese Arbeit baut auf den Erkenntnissen von TRs Masterarbeit auf. Es ist allerdings wichtig anzumerken, dass die Resultate signifikant erweitert wurden und konzeptionell stärker als die der Masterarbeit sind, siehe dazu Abschnitt 3.3 in der Dissertation. TR adaptierte die Kontinuum-Limit-Theorie für den L^∞-Fall, erarbeitet die meisten Beweise und schrieb einen großen Teil des Papers. In Zusammenarbeit mit LB, identifizierte er entscheidende Gebiets-Annahmen, welche es erlauben auch mit nichtkonvexen Gebieten zu arbeiten und bewies Konvergenz für angenäherte Randbedingungen.

[LIP-II]: In Zusammenarbeit mit LB, arbeitete TR an den Konvergenzbeweisen, basierenden auf den Ideen von JC. Zusammen mit LB und JC bewies er das Hauptresultat und die verschiedenen Lemmata, die darauf hinführen. Hierbei beschäftigte er sich vor allem mit der Adaption der Theorie für AMLEs auf den Graph-Fall, was das entscheidende Element für die ganze Arbeit ist. Weiterhin, trug er zur Gestaltung und Implementierung der numerischen Experimente, die im Paper durchgeführt wurden bei.

[**CLIP**]**:** TR erarbeitete den Algorithmus, der im Paper vorgeschlagen wird, zusammen mit LB, basierend auf dessen Idee. Zusammen mit LS, RR und DT führte er die numerischen Beispiele durch und schrieb große Teile des Quellcodes. Weiterhin schrieb er entscheidende Teile des Papers, wobei DT das Dokument Korrektur lies und klarer formulierte.

[**BREG-I**]**:** TR erweiterte LBs Idee, Bregman Iterationen für dünnbesetztes Training einzusetzen, konzipiert durch DT. Zusammen mit MB und LB erarbeitete er die Konvergenzbeweise der stochastischen Bregman Iteration. Hier schlug er auch eine fundierte Initialisierungsstrategie vor. Weiterhin führte er die numerischen Beispiele durch und schrieb den größten Teil des Quellcodes.

[**FNO**]**:** Diese Arbeit beruht auf SKs Masterarbeit und verwendet die ursprünglichen Ideen MBs, zu Auflösungsinvarianz mithilfe von FNOs. Im Paper erarbeitete TR die Beweise zur Wohldefiniertheit und Fréchet-Differenzierbarkeit, zusammen mit SK. Er schrieb große Teile des Papers und des Source-Codes, wobei DT bei der Korrektur der publizierten Version mitgeholfen hat. Hierbei führte er die numerischen Studien zusammen mit SK durch.

Preface

This work is structured into two main parts, Part I the presentation and explanation of the topics and results presented in Part II, the peer-reviewed articles.

Part I: Exposition	Part II: Prints
Chapter 2: Learning Paradigms	—
Chapter 3: Consistent Semi-Supervised Learning on Sparse Graphs	Chapters P1 and P2
Chapter 4: Robust and Sparse Supervised Learning	Chapters P3 to P5

Part I consists of five chapters, of which the first two give an introduction and explain the paradigms of *unsupervised*, *semi-supervised* and *supervised* learning. The next two chapters are split up thematically, concerning the topics of semi-supervised and supervised learning, respectively. Here, a short overview provides the necessary framework, allowing us to explain the main contributions. The last chapter presents the conclusion. In Part II the following publications are reprinted:

[LIP-II] L. Bungert, J. Calder, and T. Roith. "Uniform convergence rates for Lipschitz learning on graphs." In: *IMA Journal of Numerical Analysis* 43.4 (2022), pp. 2445–2495. DOI: 10.1093/imanum/drac048.

[LIP-I] T. Roith and L. Bungert. "Continuum limit of Lipschitz learning on graphs." In: *Foundations of Computational Mathematics* 23.2 (2023), pp. 1–39. DOI: 10.1007/s10208-022-09557-9.

[CLIP] L. Bungert, R. Raab, T. Roith, L. Schwinn, and D. Tenbrinck. "CLIP: Cheap Lipschitz training of neural networks." In: *Scale Space and Variational Methods in Computer Vision: 8th International Conference, SSVM 2021, Proceedings*. Springer. 2021, pp. 307–319. DOI: 10.1007/978-3-030-75549-2_25.

[BREG-I] L. Bungert, T. Roith, D. Tenbrinck, and M. Burger. "A Bregman learning framework for sparse neural networks." In: *Journal of Machine Learning Research* 23.192 (2022), pp. 1–43.

[FNO] S. Kabri, T. Roith, D. Tenbrinck, and M. Burger. "Resolution-Invariant Image Classification based on Fourier Neural Operators." In: *Scale Space and Variational Methods in Computer Vision: 9th International Conference, SSVM 2023, Proceedings*. Springer. 2023, pp. 307–319. DOI: 10.1007/978-3-031-31975-4_18.

The following two works that are not part of this thesis but provide an additional insight.

[LIP-III] L. Bungert, J. Calder, and T. Roith. *Ratio convergence rates for Euclidean first-passage percolation: Applications to the graph infinity Laplacian*. 2022. arXiv: 2210.09023.

[BREG-II] L. Bungert, T. Roith, D. Tenbrinck, and M. Burger. *Neural Architecture Search via Bregman Iterations*. 2021. arXiv: 2106.02479.

TR's Contribution

Here we list TR's contribution to the publications included in the thesis.

[LIP-I]: This work builds upon the findings in TR's master's thesis [Roi21]. It is however important to note that the results constitute a significant extension and are conceptually stronger than the ones in [Roi21], see Section 3.3. TR adapted the continuum limit framework to the L^∞ case, worked out most of the proofs and wrote a significant part of the paper. In collaboration with LB, he identified the crucial domain assumptions that allow to work on non-convex domains and proved convergence for approximate boundary conditions.

[LIP-II]: In collaboration with LB, TR worked on the convergence proofs building upon the ideas of JC. Together with LB and JC he proved the main convergence result and the various lemmas leading up to it. Here, he was especially concerned with the adaptation of the theory of AMLEs to the graph case, with is a crucial element for the whole work. Furthermore, he contributed to the design and implementation of the numerical examples conducted in the paper.

[CLIP]: TR worked out the main algorithm proposed in the paper together with LB, based on LB's idea. Together with LS, RR and DT he conducted the numerical examples and also wrote large parts of the source code. Furthermore, he wrote significant parts of the paper, where DT proofread and clarified the final document.

[BREG-I]: TR expanded LB's ideas of employing Bregman iteration for sparse training, conceptualized by DT. Together with MB and LB, he worked out the convergence analysis of stochastic Bregman iterations. Here, he also proposed a profound sparse initialization strategy. Furthermore, he conducted the numerical examples and wrote most of the source code.

[**FNO**]**:** This work is based on SK's master's thesis, employing the initial ideas of MB for resolution invariance with FNOs. In the paper, TR worked out the proofs for well-definedness and Fréchet-differentiability, together with SK. He wrote large parts of the paper and the source code, where DT helped with proofreading of the published version. Here, he conducted the numerical studies in collaboration with SK.

Part I.

Exposition

Chapter 1

Introduction

The field of *machine learning* emerged in the 1950s [Sam59; Ros58], motivated by the idea of letting a computer discover algorithms and patterns without having to explicitly arrange them by hand. After the initial phase and multiple “AI-winters” [SG96], numerous important developments—e.g., the rediscovery of the backpropagation algorithm, originally due to [Kel60; Ros62] and then popularized in [RHW86], see, e.g., [Sch22]—contributed to the relevance of learning methods. The advances in computer hardware, together with the availability of large amounts of data, finally allowed the machine learning enthusiasm of recent years to spark. While “deep” learning methods—i.e., techniques involving many stacked neural layers as originally proposed in [Ros58]—are the most prominent examples, there is a whole zoo of learning-based strategies that are actively applied in fields like computer vision [Cha+21], natural language processing [Khu+23] or healthcare [She+22]. In this work, we mainly focus on data-driven approaches, applied to classification tasks, where the concrete modality of the given data determines our approach. Namely, we focus on supervised—the dataset consists only of input-output pairs, i.e., is fully labeled—and semi-supervised—the data is only partially labeled—learning tasks.

For both regimes especially the last 20 years have seen great success of these data-driven methods. However, the sometimes purely heuristic learning strategies also exhibit serious drawbacks. In the supervised setting, one is usually interested in the generalization behavior of a learned classifier, i.e., how good is the performance on unseen inputs which are not part of the given training data. Unfortunately, in [GSS14] it was discovered, that this performance can be completely corrupted, by small, seemingly invisible perturbations known as *adversarial attacks*. More generally, this phenomenon leads us to the issue of *input robustness*. Given some input x, suppose that a human and some machine would classify this input to be of type c. In a rather vague but demonstrative formulation, the key implication we want to obtain for an input $\bar{x}$ is

$$\left.\begin{array}{r}\bar{x} \text{ is close to } x,\\ \bar{x} \text{ is still classified as } c \text{ by a human}\end{array}\right\} \Rightarrow \text{the machine classifies } \bar{x} \text{ as } c.$$

Next to adversarial examples this also includes resolution changes of images, which do not change the classification by a human, if they are reasonably small. In any case, the existence of these perturbations exhibits critical flaws of learning methods and calls for a better theoretical understanding of the employed models. This is where the mathematical foundation of the field becomes more relevant and properties apart from the classification performance come into play, which are discussed within this thesis.

For the semi-supervised setting, we consider graph-based algorithms as originally proposed in [ZGL03] with the graph Laplacian. The main problem we highlight in this thesis was first observed in [NSZ09], namely that the classification performance deters significantly with increasing dimensionality of the data. In fact, it turned out that solutions obtained by the standard graph Laplacian tend to be constant over the whole dataset, whenever the dimension is larger than two, which can be related to the Sobolev embedding theorem [AF03]. This issue is prevalent in the infinite data limit, where a priori we consider the case, when the amount of unlabeled data points goes to infinity, which leads us to the question of *consistency* for semi-supervised algorithms.

An issue that is shared across the supervised and semi-supervised settings is the high demand for computational resources. Training a neural network usually involves the use of GPUs for long amounts of time. On the one hand, this makes the process infeasible for less powerful machines or even mobile devices and on the other hand, generates questionable amounts of CO_2 emissions [Hoe+21]. For graph-based semi-supervised learning one first needs to compute distances between many data points, to obtain edge weights, which itself is a costly task. Furthermore, the computational complexity of various tasks on a given graph scales with the number of edges. For example, the run time for Dijkstra's algorithm to compute shortest paths on a graph, already scales linearly with the amount of edges [Dij22]. In this thesis, the keyword to reduce the computational load in both cases, is *sparsity*. The concept of sparse matrices routes deeply into the field of numerical linear algebra [Lan52; GV13] and basically consists of exploiting zeros in a matrix to speed up the computation time. For neural networks, this can be incorporated by enforcing the weight matrices of the layers to be sparse. For graphs, sparsity of the connectivity matrix simply means that we have only a small amount of active edges, which also reduces the computational cost.

Contributions in This Work Taking up the previously mentioned subjects, this thesis is concerned with *consistency, robustness* and *sparsity* of supervised and semi-supervised learning algorithms.

For the latter, we mainly consider the so-called Lipschitz learning task [NSZ09] for which we prove convergence and convergence rates for discrete solutions to their continuum counterpart in the infinite data limit. Here, we always work in a framework that allows for very sparse and therefore computationally feasible graphs.

In the supervised regime, we deal with input-robustness w.r.t. adversarial attacks and resolution changes. In the first case, we propose an efficient algorithm, penalizing the Lipschitz constant [Lip77] of a neural network, which trains an adversarially robust network. For the multi-resolution setting, we analyze the role of Fourier neural operators as proposed in [Li+21] and their connection to standard convolutional neural layers [Fuk80]. Concerning the computational complexity of neural network training, we propose an algorithm based on Bregman iterations [Osh+05] that allows for sparse weight matrices throughout the training. We also provide the convergence analysis for the stochastic adaption of the original Bregman iterations.

Structure of the Exposition In Chapter 2 we introduce the learning paradigms and basic notions used throughout this thesis. We then present the topics on consistency for semi-supervised learning on graphs in Chapter 3. After an explanatory introduction, we highlight the main contributions of [LIP-I; LIP-II]. Here, we try to have as little redundancy to the prints in Part II as possible, while still allowing for an understandable context. In Chapter 4 we comment on the supervised part of this thesis. After an additional introduction, the chapter contains three sections presenting the works [FNO; CLIP; BREG-I] individually. Finally, in Chapter 5 we summarize the contents of the whole thesis and provide possible future directions.

Chapter 2

Learning Paradigms

Throughout this thesis, we assume to be given data $\mathcal{X}_n \subset \mathcal{X} \subset \mathbb{R}^d$ consisting of n data points. We consider the task of *learning* a function $f : \tilde{\mathcal{X}} \to \mathcal{Y}$ from the given data, where $\mathcal{Y}$ denotes the output space. In our case, the set $\tilde{\mathcal{X}} \subset \mathcal{X}$ is usually chosen either as the set of data points $\mathcal{X}_n$ or as the whole space $\mathcal{X}$. The two most important cases for us are listed below.

- **Classification**: The function f assigns a label to each $x \in \tilde{\mathcal{X}}$ out of $C \in \mathbb{N}$ possible classes, i.e. $\mathcal{Y} = \{1, \dots, C\}$. In some architectures, the last layer of the neural network is given as a vector $y \in \mathbb{R}^C$. Typically, this vector is a probability vector, i.e.,

  $$y \in \Delta^C := \left\{ z \in [0,1]^C : \sum_{i=1}^{C} z_i = 1 \right\}.$$

  This can be enforced via the softmax function [Bri90] softmax : $\mathbb{R}^C \to \mathbb{R}^C$

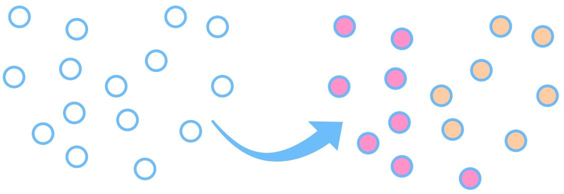
  $$\operatorname{softmax}(z)_i := \frac{\exp(z_i)}{\sum_{j=1}^{C} \exp(z_j)}$$

  which was actually introduced by Boltzmann in [Bol68]. This allows the interpretation that the ith entry of $f_\theta(x) \in \Delta^C$ models the probability that x belongs to class i. In order to obtain a label, one can simply choose the maximum entry, i.e. $\operatorname{argmax}_{i=1,\dots,C} f_\theta(x)_i$.

- **Image denoising**: The function f outputs a denoised version of an input image. Here we have $\mathcal{X} = \mathcal{Y} = \mathbb{R}^{K \times N \times M}$, where
  - $K \in \mathbb{N}$ is the number of color channels,
  - N, M denote the width and height of the image.

The learning paradigms, we consider in this thesis, differ by their usage of labeled data. We review the concepts in the following.

2.1. Unsupervised Learning

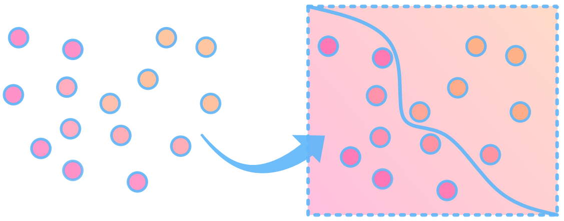


In the case of unsupervised learning, we are not given any labeled data. In our context, the most important application is data clustering. Other tasks involve dimensionality reduction or density estimation, see [ST14]. The clustering task consists of grouping data based on some similarity criterion. In this sense, clustering can also be interpreted as classification, i.e., the desired function is a mapping $f : \tilde{\mathcal{X}} \to \{1, \dots, C\}$ where $C \in \mathbb{N}$ denotes the number of clusters. Typically, one wants to obtain a clustering of the given data set, i.e., $\tilde{\mathcal{X}} = \mathcal{X}_n$. We list some typical clustering methods below:

- K-means algorithm [Ste+56],
- expectation maximization [DLR77],
- Cheeger cuts [GS15; SB09; Gar+16; GMT22],
- spectral clustering [GS18; THH21; Hof+22].

Unsupervised learning is not the main focus of this present work. However, we note that especially the concepts developed in [GS15] for Cheeger cuts are crucial for the continuum limit framework in Section 3.3.

2.2. Supervised Learning



In this setting, each data point $x \in \mathcal{X}_n$ is labeled, via a given function $g : \mathcal{X}_n \to \mathcal{Y}$ such that we have a finite training set $\mathcal{T} = \{(x, g(x)) : x \in \mathcal{X}_n\}$. The task is to infer a function defined on the underlying space, $f : \mathcal{X} \to \mathcal{Y}$, i.e., we want to assign a label to unseen inputs $x \in \mathcal{X}$ that are not necessarily part of the given data. Often, one models the problem via a joint probability function $P_{\mathcal{X},\mathcal{Y}}$ and assumes that the training data are independent and identically distributed (i.i.d) with respect to $P_{\mathcal{X},\mathcal{Y}}$. In this interpretation, the aim is to approximate the conditional $P_{\mathcal{X},\mathcal{Y}}(y|x)$ for an input $x \in \mathcal{X}$ and output $y \in \mathcal{Y}$.

In order to *learn* the function f from the given data, one needs to choose a parameterized class of functions, where typically each element can be described by a finite number of parameters. Among others, common methods or parametrizations include

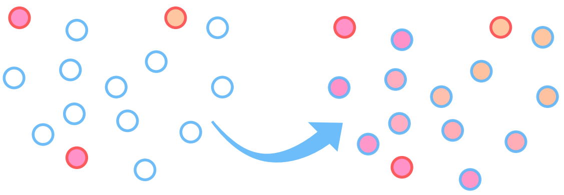
- support vector machines [CV95; SS05],

- decision trees [MS63; Bre+84],
- neural networks [Tur04; Ros58; MP69].

We refer to [Sch15] for an exhaustive historical overview.

In Chapter 4 we exclusively focus on supervised learning algorithms employing neural networks, where the concrete setting is given at the start of the chapter.

2.3. Semi-Supervised Learning



In the semi-supervised setting we assume that only a fraction of the data $\mathcal{X}_n$ is labeled, i.e., we are given a function $g : \mathcal{O}_n \to \mathcal{Y}$ where $\mathcal{O}_n \subset \mathcal{X}_n$ is the non-empty set of labeled data. Typically, this constitutes only a small fraction of all available points, i.e., $|\mathcal{O}_n| \ll |\mathcal{X}_n|$. In this thesis we restrict ourselves to the *transductive setting*, i.e., we want to infer a function acting only on the data $f : \mathcal{X}_n \to \mathcal{Y}$. This is opposed to the inductive setting, where f also classifies unseen points $x \in \mathcal{X}$, [Zhu05]. Common algorithms and methods include

- expectation maximization and mixture models [DLR77; CCC+03],
- self-training and co-training [BM98],
- graph-based learning [Zhu05].

Mostly, we consider the extension task with $\mathcal{Y}$ being chosen as $\mathbb{R}$. This can be seen as a binary classification task, where for $o \in \mathcal{O}_n$ we have $g(o) = 1$ if o belongs to a some class and $g(o) = 0$ otherwise. The function $f : \mathcal{X}_n \to \mathbb{R}$ then determines the probability that any vertex $x \in \mathcal{X}_n$ belongs to this class, where we can binarize the output via some thresholding, e.g.,

$$x \text{ belongs to the class } \Leftrightarrow f(x) > 1/2.$$

This methodology can be extended to classification tasks beyond the binary case, via the so-called one-vs-all technique [ZGL03]. Given a classification problem with $C \in \mathbb{N}$ possible classes, we assume that the labeling function $g : \mathcal{O}_n \to \Delta^C$ outputs one-hot vectors, i.e., $g(o)_c = 1$ if o belongs to class c and $g(o)_c = 0$ otherwise, for every $c = 1, \dots, C$. We then perform the binary classification problem "x belongs to class c" for every $c = 1, \dots, C$, by considering the extension task of

$$g_c : \mathcal{O}_n \to \mathbb{R} \qquad g_c(o) = g(o)_c,$$

which yield functions $f_c : \mathcal{X}_n \to [0, 1], c = 1, \dots, C$. The final output can either be obtained by employing a Bayes classifier [DGL13], i.e., $f : \mathcal{X}_n \to \{1, \dots, C\}$

$$f(x) := \operatorname{argmax}_{c=1,\dots,C} f_c(x)$$

or by applying a softmax to obtain a probability vector, i.e., $f : \mathcal{X}_n \to \Delta^C$

$$f(x) := \operatorname{softmax}\left(f_1(x), \dots, f_c(x)\right).$$

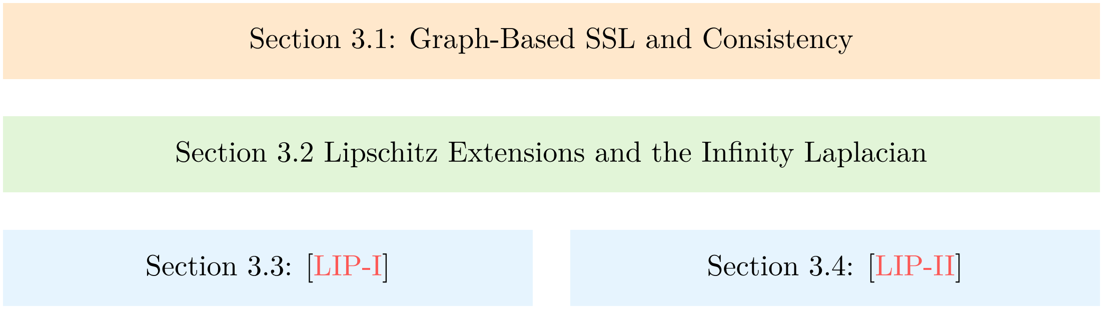
In Chapter 3 we focus on graph-based learning algorithms, however we refer to [Zhu05] for an overview of semi-supervised learning algorithms.

Chapter 3

Consistent Semi-Supervised Learning on Sparse Graphs

This chapter considers the consistency of graph-based semi-supervised learning methods and contextualizes the topics provided in the prints [LIP-I; LIP-II]. We are given partially labeled data, where the task is to obtain a labeling on the whole graph. Considering learning algorithms, we are especially interested in the following aspects:

- **Consistency:** What is the behavior in the infinite data limit?
- **Sparsity:** How does sparsity of the graph weights affect this behavior?



Section 3.1: Graph-Based SSL and Consistency

Section 3.2 Lipschitz Extensions and the Infinity Laplacian

Section 3.3: [LIP-I]

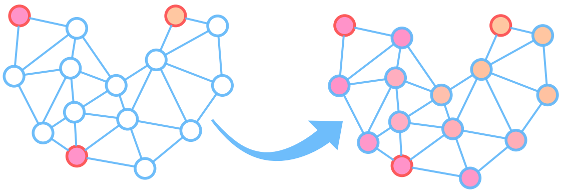
Section 3.4: [LIP-II]

In [LIP-I] we first consider Γ-convergence of discrete L^∞ functionals to their continuum counterpart, which also yields convergence of minimizers. This notion of convergence shapes our understanding of asymptotic consistency in the first part of this chapter. The proofs allow for very sparse graphs, however they only directly apply to the Lipschitz extension task, which does not admit unique solutions. We refer to Section 3.3 for more details.

The issue of non-uniqueness is addressed in [LIP-II], which considers uniform convergence of graph infinity harmonic functions to their continuum counterpart. These functions are in fact special solutions of the extension task in [LIP-I], namely so-called absolutely minimizing Lipschitz extensions, which are—under certain assumptions—unique. Again, we are able to work with a very mild scaling assumption that allows for sparse graphs. Furthermore, we are able to prove the first quantitative convergence rates, where we provide the key insight, that rates for graph distance functions transfers to rates for infinity harmonic functions. The main contributions are presented in Section 3.4.

Before we present these works, we first introduce the setting of consistency for graph-based SSL in Section 3.1, where we also discuss the p-Laplacian both in the continuum and the graph setting. Similarly, we then introduce the problem we are actually concerned with, namely the ∞-Laplacian and Lipschitz extensions in Section 3.2.

3.1. Graph-Based SSL and Consistency



In the semi-supervised learning setting of Section 2.3, we are given a finite set $\Omega_n \subset \tilde{\Omega}$ consisting of n points, where $\tilde{\Omega}$ denotes the input space. We assume that a non-empty subset $\mathcal{O}_n \subset \Omega_n$ is labeled, i.e., we are given a labeling function $\mathbf{g} : \mathcal{O}_n \to \mathbb{R}$. From now on, we denote functions acting on a discrete set Ω_n using bold symbols to distinguish them from functions acting on the continuum $\tilde{\Omega}$. In particular, the space of functions on Ω_n can simply be identified with $\mathbb{R}^n$, i.e., $\{\mathbf{u} : \Omega_n \to \mathbb{R}\} \sim \mathbb{R}^n$. The semi-supervised learning problem consists in finding an extension of $\mathbf{g}$ from $\mathcal{O}_n$ to the whole set Ω_n, i.e.,

$$\begin{gathered}\text{find } \mathbf{u} : \Omega_n \to \mathbb{R}, \\ \text{such that } \mathbf{u}(x) = \mathbf{g}(x) \text{ for all } x \in \mathcal{O}_n.\end{gathered} \tag{SSL}$$

In order to obtain meaningful solutions, one usually incorporates the *smoothness assumption* [ST14] which can be informally stated as follows:

“*Points that are close together are more likely to share a similar label.*”

Remark 3.1 (Notational Remark). In Chapter 2 we employ the symbol $\mathcal{X}$ to denote the input set, which is now replaced by Ω. This is due to the fact, that [LIP-I; LIP-II; LIP-III] employ this notation, for which we choose to use it here. △

Often, one assumes that the points $x \in \Omega_n$ are i.i.d. w.r.t. a distribution μ with density $\rho : \Omega \to \mathbb{R}$. However, as observed in [Roi21], the data distribution is not directly relevant for the works in [LIP-I; LIP-II] and we can instead consider a more general class of point clouds. The appropriate generalization to study the limit of Ω_n for $n \to \infty$ is discussed in Section 3.3.

3.1.1. Weighted Graphs

In order to employ the smoothness assumption, we need a notion of “closeness” on the set Ω_n, for which we introduce weighted graphs. Namely, we define a weighting function w that allows to compare points in Ω_n and therefore induces the desired notion.

Definition 3.2 (Weighted Graphs). For a finite set Ω_n and a weight function $w_n : \Omega_n \times \Omega_n \to \mathbb{R}$, the tuple (Ω_n, w_n) is called a *weighted graph.*

Other Notions of Graphs Typically, a graph is defined as a pair (Ω_n, E) where E denotes the set of edges. Here, one has two cases:

- *Undirected graph:* $E = \{\{x, y\} : \text{there is an edge between } x \in \Omega_n \text{ and } y \in \Omega_n\}$, i.e., s $E \subset 2^{\Omega_n}$ and each edge is undirected, since $\{x, y\} = \{y, x\}$.
- *Directed graph:* $E = \{(x, y) : \text{there is an edge from } x \text{ to } y\}$, i.e., $E \subset \Omega_n \times \Omega_n$ and each edge is directed, i.e., in general $(x, y) \in E \not\Rightarrow (y, x) \in E$.

Additionally, one then considers a weight function $W : E \to \mathbb{R}$ assigning a weight to each edge, and defines the triple (Ω_n, E, W) as a weighted graph. However, we note that all this information can be represented much more elegantly by a weight function $w : \Omega_n \times \Omega_n \to \mathbb{R}$. A directed edge set $E \subset \Omega_n \times \Omega_n$ can be equivalently expressed by a weight function $w : \Omega_n \times \Omega_n \to \mathbb{R}$ where $w(x, y) \neq 0$ if and only if $(x, y) \in E$, a weighting $W : E \to \mathbb{R}$ naturally transfers to w. In the case of an undirected graph, one simply requires the weight function to be symmetric, i.e., $w(x, y) = w(y, x)$ for all $x, y \in \Omega_n$. Furthermore, in the above definition, the set Ω_n does not enter the definition up to the ordering of its $n \in \mathbb{N}$ elements. Therefore, a graph could be entirely represented by a weight function $w : \{1, \dots, n\} \times \{1, \dots, n\} \to \mathbb{R}$. However, since the definition of w will incorporate information about points $\Omega_n \subset \mathbb{R}^d$ we use the notation (Ω_n, w_n) for graphs in the following.

Kernels and the Graph Scale In most of our applications the data Ω_n is given as a subset of $\mathbb{R}^d$ and in fact we are interested in the limit $n \to \infty$, where Ω_n fills out a domain $\tilde{\Omega} \subset \mathbb{R}^d$. In the continuum, we consider *local* operators incorporating changes of functions $u : \tilde{\Omega} \to \mathbb{R}$ at an infinitesimal small scale. Since interactions on a graph are inherently non-local in the Euclidean sense, we need to localize them in the limit $n \to \infty$. A popular choice of weight function that guarantees this behavior is

$$w(x, y) = \eta_\varepsilon(|x - y|)$$

where $\eta_\varepsilon : [0, \infty) \to [0, \infty]$ is a kernel function depending on a scaling parameter $\varepsilon \in \mathbb{R}^+$. The parameter ε is also referred to as the *graph scale* and informally speaking determines the scale of the graph interactions. The smaller ε the smaller the interaction radius of points in Ω_n should be. In our specific setting of L^∞ problems, it is typical to choose

$$\eta_\varepsilon(\cdot) = \frac{1}{c_\eta\, \varepsilon}\, \eta\left(\frac{\cdot}{\varepsilon}\right)$$

where η is a non-increasing kernel and c_η is a constant depending on the kernel. Typical examples of kernels include

- (constant weights) $\eta(t) = 1_{[0,1]}(t)$,

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

In the above examples, we ensured that each weight function has a compact support, by multiplying it with $1_{[0,1]}$. This allows us to directly connect the sparsity of the graph with the scale ε.

Remark 3.3. In the continuum limit one needs to consider the value

$$\sigma_\eta = \sup_{t\in\mathbb{R}^+} t\,\eta(t),$$

which is assumed to be finite. Considering graph problems in L^p for $p < \infty$ the corresponding value is given as

$$\sigma_\eta^{(p)} = \int_{\mathbb{R}^+} \eta(t)\, t^{d+p} dt \tag{3.1}$$

which was first employed in [GS15] for $p = 1$ and then in [ST19] for general $p < \infty$. With the slight modification

$$\tilde{\sigma}_\eta^{(p)} = \int_{\mathbb{R}^+} \eta(t)^p\, t^{d+p} dt,$$

we see that $\sqrt[p]{\tilde{\sigma}_\eta^{(p)}}$ degenerates to σ_η in the limit $p \to \infty$. △

Remark 3.4. In [LIP-I] we choose $c_\eta = 1$ and therefore σ_η then appears in the limit functional. In [LIP-II] we rescale the kernel, i.e., $c_\eta = \sigma_\eta$ which allows us to work with non-rescaled operators in the limit. △

Remark 3.5. In order to obtain continuum limits in the case, $p < \infty$ one usually employs a factor of $1/\varepsilon^{p+d}$ in front of the graph weights [GS15; ST19]. In Section 3.2.2 we see that this factor degenerates to $1/\varepsilon$ in the case $p \to \infty$. The intuition here is that problems in L^∞ do not treat mass in a quantitative but rather a qualitative manner. Therefore, factors like ε^d which appear because of integrals in $\mathbb{R}^d$ do not contribute for $p = \infty$. △

Sparse Graphs An important practical aspect connected to the graph scale is the sparsity of the graph, which is given by the numbers of non-zero elements in the weight matrix $W \in \mathbb{R}^{n\times n}$

$$W_{ij} := w(x_i, x_j) \quad x_i, x_j \in \Omega_n,$$

where we assume an ordering $\Omega_n = \{x_1, \dots, x_n\}$. In order to have a computationally feasible problem, this matrix should have very few non-zero elements. Assuming that the kernel η has compact support, w.l.o.g. $\mathrm{supp}(\eta) \subset [0, 1]$ we observe that the sparsity is directly influenced by the graph scale ε. Namely, $|x_i - x_j| > \varepsilon \Rightarrow W_{ij} = 0$, see Fig. 3.1.

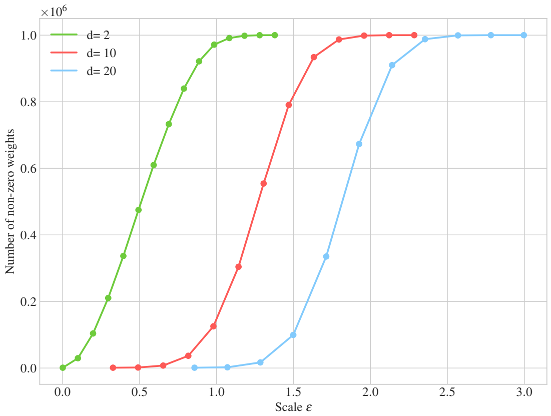


Figure 3.1.: Dependence of the number of non-zero edges on the scaling parameter ε. Here, we sample $n = 10^3$ points uniformly in the cube $[0,1]^d$ and only connect vertices $x_i, x_j \in \Omega_n$ if $|x_i - x_j| < \varepsilon$. The plots show the behavior for $d = 2, 10, 20$.

Therefore, from a practical point of view ε should be chosen relatively small. However, showing consistency of graph-based SSL algorithms often requires scaling assumptions that permit the desired length scales [GS15; ST19; Cal19]. In Section 3.3 we give concrete examples of such scaling assumptions. Note, that in the case $p < \infty$ these scaling assumptions are, however, not an artifact of the proof technique. They are provably optimal and arise through the problem formulation. One of the main goals of [LIP-I; LIP-II] was to show convergence and rates at the smallest possible length scale, which was indeed achieved.

Remark 3.6. In many practical applications, graph weights connecting all points within an ε-ball of Euclidean distance are inferior to k-nearest neighbor (knn) graphs [Cal+20; FCL22; CT22]. While most consistency results employ the ε-ball setting, recently, the authors in [CT22] were able to show convergence rates for knn graphs. In this thesis we focus on the ε-ball setting, however it is an interesting open question to transfer the results of [LIP-I; LIP-II; LIP-III] to the knn setting. △

3.1.2. The p-Laplacian: Continuum and Graph

The archetype of learning methods we consider in the following is the so-called *Laplacian learning*, which had one of its first appearances in [ZGL03]. The associated problem was given as

$$\begin{gathered}\min_{\mathbf{u}:\Omega_n\to\mathbb{R}} \sum_{x,y\in\Omega_n} w_n(x,y)^2 \left(\mathbf{u}(y)-\mathbf{u}(x)\right)^2, \\ \text{subject to } \mathbf{u}(x) = \mathbf{g}(x) \text{ for all } x \in \mathcal{O}_n.\end{gathered} \tag{3.2}$$

Here, we always assume non-negative weights $w_n : \Omega_n \times \Omega_n \to [0, \infty)$. The intuitive idea behind this method is that it minimizes a discrete approximation of the Dirichlet energy and should therefore enforce a certain kind of “smoothness” of the solution $\mathbf{u}$. In [ZS05] it was proposed to generalize the idea of the graph Dirichlet energy in Eq. (3.2) to arbitrary $1 \leq p < \infty$, which lead to the p-Laplacian on the graph. While our contributions in [LIP-I; LIP-II; LIP-III] mainly consider the case of $p = \infty$, the case $p < \infty$ serves as a motivation for the whole discussion. Therefore, we briefly review the p-Laplacian in the continuum setting and then discuss the situation on the graph.

Continuum Setting We follow the exposition in [Lin17]. Let $\Omega \subset \mathbb{R}^d$ be a bounded domain, then we consider the p-Dirichlet energy for functions $u \in W^{1,p}(\Omega)$,

$$\mathcal{E}_{p,\rho}(u) := \int_\Omega |\nabla u|^p \, \rho(x)^2 \, dx, \tag{3.3}$$

where $\rho : \Omega \to \mathbb{R}$ is some given function. In the setting of [GS15; ST19] ρ denotes the density of data distribution as in Section 3.1. For $\rho \equiv 1$ we simply write $\mathcal{E}_{p,1} = \mathcal{E}_p$. The associated variational problem is given in the following.

Problem 3.7 (Variational Formulation). For $p \in [1, \infty)$ find $u \in W^{1,p}(\Omega)$ such that

$$\mathcal{E}_{p,\rho}(u) \leq \mathcal{E}_{p,\rho}(v)$$

for all v, such that $(u - v) \in W^{1,p}_0(\Omega)$.

Assume for simplicity that $\rho \equiv 1$ and that $u \in W^{1,p}(\Omega)$ is a minimizer of the above problem, then its first variation must vanish, i.e., for all $\phi \in C_0^\infty(\Omega)$ one has

$$\int_\Omega \langle |\nabla u|^p \, \nabla u, \nabla \phi \rangle \, dx = 0. \tag{3.4}$$

A function $u \in W^{1,p}(\Omega)$ satisfying Eq. (3.4) is called a *weak solution* of the p-Laplace equation. In fact, if u is smooth enough, one can infer that

$$\Delta_p u := \operatorname{div}(|\nabla u|^{p-2} \, \nabla u) = 0 \tag{3.5}$$

where Δ_p is called the p-Laplacian. Boundary conditions on $\partial\Omega$ given by a function $g \in W^{1,p}(\Omega)$ can be incorporated by considering the set $U_g := \{u \in W^{1,p}(\Omega) : u - g \in W^{1,p}_0(\Omega)\}$. We state the following classical result, which can for example be found in [Lin17].

Theorem 3.8 (Existence and Uniqueness). For $p \in (1, \infty)$ and $g \in W^{1,p}(\Omega)$ there exists a unique minimizer $u \in U_g$ of the p-Dirichlet energy, i.e.,

$$\operatorname{argmin}_{u \in U_g} \mathcal{E}_p(u) = u.$$

Moreover, u is a weak solution of the p-Laplace equation and there exists a function $\tilde{u} \in C(\Omega)$ such that $u = \tilde{u}$ a.e. in Ω. If $g \in C(\Omega)$ and Ω is sufficiently smooth, then $\tilde{u}|_{\partial\Omega} = g|_{\partial\Omega}$.

Proof. The proof can be found in [Lin17, Thm. 2.16]. □

Local Minimization Property Let u solve Problem 3.7 on the whole domain Ω for given boundary values $g \in W^{1,p}(\Omega)$ and let $V \subset\subset \Omega$ be a sufficiently regular subset. Then we have that

$$\mathcal{E}_p(u) = \int_V |\nabla u|^p \, dx + \int_{\Omega \setminus V} |\nabla u|^p \, dx.$$

Now consider Problem 3.7 restricted on V with boundary values given by $u|_{\partial V}$ and denote by v the solution of this sub-problem on V. Since $u = v$ on ∂V we can extend v onto Ω by setting $v = u$ in $\Omega \setminus V$, which yields a function $v \in W^{1,p}(\Omega)$. Therefore, we obtain

$$\mathcal{E}_p(v) = \int_V |\nabla v|^p \, dx + \int_{\Omega \setminus V} |\nabla u|^p \, dx \leq \int_V |\nabla u|^p \, dx + \int_{\Omega \setminus V} |\nabla u|^p \, dx = \mathcal{E}_p(u).$$

Since u is unique this yields $u|_V = v|_V$, and therefore we know that any solution on Ω solves Problem 3.7 also on any “nice” subset, with the boundary values given by itself. In Section 3.2 we see that for $p = \infty$ this *local minimization property* does not hold automatically and has to enforced additionally. The underlying reason here is, that integrals are set-additive, but suprema are not, which is similarly explained in [ACJ04].

Laplacian Learning A natural extension of the problem given in Eq. (3.3) is obtained by considering the target functional

$$\mathbf{E}_p^{w_n}(\mathbf{u}) := \sum_{x,y \in \Omega_n} w_n(x,y)^p \left|\mathbf{u}(y) - \mathbf{u}(x)\right|^p,$$

which we refer to as the graph p-Dirichlet energy. Indeed, we notice structural similarities to the p-Dirichlet energy $\mathcal{E}_p$ in Eq. (3.3), replacing the integral by a finite sum and derivatives by weighted finite differences

$$w_n(x,y)^p \left|\mathbf{u}(y) - \mathbf{u}(x)\right|^p.$$

This naturally leads to the following minimization problem.

Problem 3.9 (Graph Energy Minimization). Given a weighted graph (Ω_n, w_n) and a labeling function $\mathbf{g} : \mathcal{O}_n \to \mathbb{R}$, for $\mathcal{O}_n \subset \Omega_n$ we consider the problem

$$\min_{\mathbf{u}:\Omega_n \to \mathbb{R}} \mathbf{E}_p^{w_n}(\mathbf{u}) \text{ subject to } \mathbf{u}(x) = \mathbf{g}(x) \text{ for all } x \in \mathcal{O}_n.$$

Since every function $\mathbf{u} : \Omega_n \to \mathbb{R}$ can be identified with a vector $\mathbf{u} \in \mathbb{R}^n$, the above problem is in fact an optimization problem in $\mathbb{R}^n$. The functional $\mathbf{E}_p^{w_n} : \mathbb{R}^n \to \mathbb{R}$ is bounded from below by 0, continuous, convex and in the case $p > 1$ even differentiable. However, one can prove unique existence with techniques very similar to the continuum case in [Lin17]. We give the adapted proof below for completeness. The important property to establish uniqueness is that the graph is connected to the boundary, i.e., for every $x \in \Omega_n$ there exists a path $x = \gamma_1, \dots, \gamma_m = o \in \Omega_n$ with $w(\gamma_{i+1}, \gamma_i) > 0$ connecting x to a point $o \in \mathcal{O}_n$ in the boundary. This is very intuitive, since on any connected component that does not communicate with the boundary, an arbitrary constant minimizes the "graph gradient".

Theorem 3.10 (Existence and Uniqueness). Let (Ω_n, w_n) be a graph with non-negative weights, which is connected to its boundary $\mathcal{O}_n \subset \Omega_n$. Then Problem 3.9 admits a unique solution $\mathbf{u} : \Omega_n \to \mathbb{R}$.

Proof. The proof is an adaption from the continuum case in [Lin17, Thm. 2.16]. We start by proving uniqueness. Let $E = \{(x, y) : w_n(x, y) > 0\}$ be the set of all active edges and denote by $\nabla^{w_n}\mathbf{u} \in \mathbb{R}^{|E|}$

$$(\nabla^{w_n}\mathbf{u})_{(x,y)} := w_n(x,y)\,(\mathbf{u}(y) - \mathbf{u}(x))$$

the "graph gradient" on E, where we have that

$$\mathbf{E}_p^{w_n}(\mathbf{u}) = \sum_{(x,y)\in E} w_n(x,y)^p\,|\mathbf{u}(y) - \mathbf{u}(x)|^p = \sum_{(x,y)\in E} \left|(\nabla^{w_n}\mathbf{u})_{(x,y)}\right|^p = \|\nabla^{w_n}\mathbf{u}\|_p^p\,.$$

Let $\mathbf{u}_1, \mathbf{u}_2$ be two solutions of Problem 3.9, i.e., $\mathbf{E}_p^{w_n}(\mathbf{u}_1) = \mathbf{E}_p^{w_n}(\mathbf{u}_2)$. Since $\mathbf{u}_1, \mathbf{u}_2$ fulfill the boundary conditions, this also holds for $1/2\,(\mathbf{u}_1 + \mathbf{u}_2)$ and therefore $\mathbf{E}_p^{w_n}(\mathbf{u}_1) \le \mathbf{E}_p^{w_n}(1/2(\mathbf{u}_1 + \mathbf{u}_2))$.

Assume that there exists $(\bar{x}, \bar{y}) \in E$ such that $\nabla^{w_n}\mathbf{u}_1(\bar{x}, \bar{y}) \neq \nabla^{w_n}\mathbf{u}_2(\bar{x}, \bar{y})$, then we have that

$$\begin{aligned}
\mathbf{E}_p^{w_n}(\mathbf{u}_1) \le \mathbf{E}_p^{w_n}\left(\frac{\mathbf{u}_1 + \mathbf{u}_2}{2}\right) &= \left\|\frac{\nabla^{w_n}\mathbf{u}_1(x,y) + \nabla^{w_n}\mathbf{u}_2(x,y)}{2}\right\|_p^p \\
&< \frac{1}{2} \sum_{(x,y)\in E\setminus\{(\bar{x},\bar{y})\}} \Big(|\nabla^{w_n}\mathbf{u}_1(x,y)|^p + |\nabla^{w_n}\mathbf{u}_2(x,y)|^p \Big) \\
&\quad + \frac{1}{2}\Big(|\nabla^{w_n}\mathbf{u}_1(\bar{x},\bar{y})|^p + |\nabla^{w_n}\mathbf{u}_2(\bar{x},\bar{y})|^p \Big) \\
&= \frac{1}{2}\Big(\mathbf{E}_p^{w_n}(\mathbf{u}_1) + \mathbf{E}_p^{w_n}(\mathbf{u}_2)\Big) = \mathbf{E}_p^{w_n}(\mathbf{u}_1),
\end{aligned}$$

which is a contradiction. Here, we employed the strict convexity of $t \mapsto |t|^p$, which implies $\left|t + \bar{t}\right|^p \le 2^{p-1}\left(|t|^p + \left|\bar{t}\right|^p\right)$, where the inequality is sharp if $t \neq \bar{t}$. Therefore, we obtain that $\nabla^{w_n}\mathbf{u}_1 = \nabla^{w_n}\mathbf{u}_2$, which yields

$$\mathbf{u}_1(y) - \mathbf{u}_1(x) = \mathbf{u}_2(y) - \mathbf{u}_2(x) \qquad \text{for all} \qquad (x,y) \in E. \tag{3.6}$$

Since (Ω_n, w_n) is connected to $\mathcal{O}_n$ for any $x \in \Omega_n$ we can find a path $x = \gamma_1, \dots, \gamma_m = o$ connecting x to a point $o \in \mathcal{O}_n$. By definition $w_n(\gamma_{i+1}, \gamma_i) > 0$ for all $i = 1, \dots, m-1$ and therefore using Eq. (3.6) we obtain

$$\mathbf{u}_1(\gamma_i) - \mathbf{u}_1(\gamma_{i+1}) = \mathbf{u}_2(\gamma_i) - \mathbf{u}_2(\gamma_{i+1}) \qquad \text{for all} \qquad i = 1, \dots, m-1.$$

which together with the boundary conditions $\mathbf{u}_1(o) = \mathbf{u}_2(o) = \mathbf{g}(o)$ yields

$$\begin{aligned}
\mathbf{u}_1(x) = \mathbf{u}_1(\gamma_1) &= \mathbf{u}_1(\gamma_1) - \mathbf{u}_1(\gamma_2) + \mathbf{u}_1(\gamma_2) \\
&= \mathbf{u}_1(\gamma_1) - \mathbf{u}_1(\gamma_2) + \mathbf{u}_1(\gamma_2) - \mathbf{u}_1(\gamma_3) + \mathbf{u}_1(\gamma_3) = \dots \\
&= \left[\sum_{i=1}^{m-1} \mathbf{u}_1(\gamma_i) - \mathbf{u}_1(\gamma_{i+1})\right] + \mathbf{u}_1(o) \\
&= \left[\sum_{i=1}^{m-1} \mathbf{u}_2(\gamma_i) - \mathbf{u}_2(\gamma_{i+1})\right] + \mathbf{u}_2(o) \\
&= \mathbf{u}_2(x).
\end{aligned}$$

Since $x \in \Omega_n$ was arbitrary, we get $\mathbf{u}_1 = \mathbf{u}_2$.
To show existence, we do not need to assume that the graph is connected to the boundary. On every connected component $V \subset \Omega_n$ that is not connected to $\mathcal{O}$ we can set $\mathbf{u}(x) = c$ for all $x \in V$, where $c \in \mathbb{R}$ is an arbitrary constant. We can select all edges in V, i.e., $E_V := E \cap V^{\times 2}$ and since V was a connected component—i.e., it does not have active edges to vertices outside of V—we can also split up the functional

$$\mathbf{E}_p^{w_n}(\mathbf{u}) = \sum_{(x,y)\in E\setminus E_V} |\nabla^{w_n}\mathbf{u}(x,y)|^p + \underbrace{\sum_{(x,y)\in E_V} |\nabla^{w_n}\mathbf{u}(x,y)|^p}_{=0} = \sum_{(x,y)\in E\setminus E_V} |\nabla^{w_n}\mathbf{u}(x,y)|^p$$

where the contribution on the connected component is zero, since we set $\mathbf{u}$ to be constant here. Therefore, w.l.o.g. we can assume that Ω_n is connected to the boundary, since any component not connected to the boundary does not contribute to the problem. Let now $\mathbf{u} : \Omega_n \to \mathbb{R}$ be a function and for $x \in \Omega_n$ let $\gamma \in \Omega_n^{\times m}$ be a path to a point in the boundary $o^x \in \mathcal{O}_n$, then we have that

$$|\mathbf{u}(x)|^p \leq C \left(\sum_{i=1}^{m^x-1} |\mathbf{u}(\gamma_i) - \mathbf{u}(\gamma_{i+1})|^p + |\mathbf{g}(o^x)|^p\right) \leq C \left(\mathbf{E}_p^{w_n}(\mathbf{u}) + |\mathbf{g}(o^x)|^p\right)$$

where $C > 0$ is a generic constant—not depending on $\mathbf{u}$—that also accounts for the factor $\left(\min_{(x,y)\in E} w_n(x,y)\right)^{-1}$. Summing over all $x \in \Omega_n$ yields

$$\begin{aligned}
\|\mathbf{u}\|_p^p = \sum_{x\in\Omega_n} |\mathbf{u}(x)|^p &\leq C \left(n\, \mathbf{E}_p^{w_n}(\mathbf{u}) + \sum_{x\in\Omega_n} |\mathbf{g}(o^x)|^p\right) \\
&\leq C\Big(\mathbf{E}_p^{w_n}(\mathbf{u}) + \|\mathbf{g}\|_p^p\Big).
\end{aligned}$$

Let now $\mathbf{u}_j$ denote a sequence such that

$$\mathbf{E}_p^{w_n}(\mathbf{u}_j) \leq I + 1/j, \qquad \text{where} \qquad I := \inf_{\mathbf{u}:\mathbf{u}=\mathbf{g} \text{ on } \mathcal{O}_n} \mathbf{E}_p^{w_n}(\mathbf{u}).$$

Then we have that

$$\|\mathbf{u}_j\|_p^p \leq C\ (\mathbf{E}_p^{w_n}(\mathbf{u}_j) + \|\mathbf{g}\|_p^p) \leq C\ (I + 1/j + \|\mathbf{g}\|_p^p) \leq \tilde{C}$$

$j \in \mathbb{N}$, i.e., $\mathbf{u}_j$ is a uniformly bounded sequence of vectors in the finite dimensional space $\mathbb{R}^n$, and therefore there exists a subsequence—which we do not relabel—that converges to some $\mathbf{u}^* \in \mathbb{R}^n$. Since the functional $\mathbf{E}_p^{w_n} : \mathbb{R}^n \to \mathbb{R}$ is continuous, we obtain that

$$\mathbf{E}_p^{w_n}(\mathbf{u}^*) = \lim_{j\to\infty} \mathbf{E}_p^{w_n}(\mathbf{u}_j) = I,$$

which yields existence. □

The Graph Laplacian In the continuum case one considers the Euler–Lagrange equation for the functional $\mathcal{E}_p$, which yields the p-Laplacian, see Section 3.1.2. Analogously, the optimality conditions for the graph p-Dirichlet energy $\mathbf{E}_p^{w_n} : \mathbb{R}^n \to \mathbb{R}$ for $p > 1$ yield

$$\nabla_{\mathbf{u}} \mathbf{E}_p^{w_n}(\mathbf{u}) = 0,$$

where for $x \in \Omega_n$ we have

$$\left(\nabla_{\mathbf{u}} \mathbf{E}_p^{w_n}(\mathbf{u})\right)_x =: \Delta_p^{w_n} \mathbf{u}(x) = \sum_{y\in\Omega_n} w_n(x,y)^p \left|\mathbf{u}(y) - \mathbf{u}(x)\right|^{p-2} (\mathbf{u}(y) - \mathbf{u}(x)),$$

which is referred to as the graph p-Laplacian operator. This yields the following problem.

Problem 3.11 (Graph p-Laplacian). Given a weighted graph (Ω_n, w_n) and a labeling function $\mathbf{g} : \mathcal{O}_n \to \mathbb{R}$ with $\mathcal{O}_n \subset \Omega_n$, find a function $\mathbf{u} : \Omega_n \to \mathbb{R}$ such that

$$\begin{aligned} \Delta_p^{w_n}\mathbf{u} = 0, &\text{ in } \Omega_n \setminus \mathcal{O}_n, \\ \mathbf{u} = \mathbf{g} &\text{ on } \mathcal{O}_n. \end{aligned}$$

Since the functional $\mathbf{E}_p^{w_n}$ has a unique minimizer subject to the constraints given by $\mathbf{g}$ and the graph p-Laplacian is derived via optimality conditions, one expects that Problem 3.9 and Problem 3.11 are equivalent. This is formulated in the following theorem.

Theorem 3.12 (Existence and Uniqueness). Let (Ω_n, w_n) be a graph with non-negative weights, that is connected to its boundary $\mathcal{O}_n \subset \Omega_n$. Then there exists a unique solution $\mathbf{u} : \Omega_n \to \mathbb{R}$ to Problem 3.11, which is also the unique minimizer of Problem 3.9.

Proof. This statement can be proven similarly to [ETT15, Thm. 5.3]. □

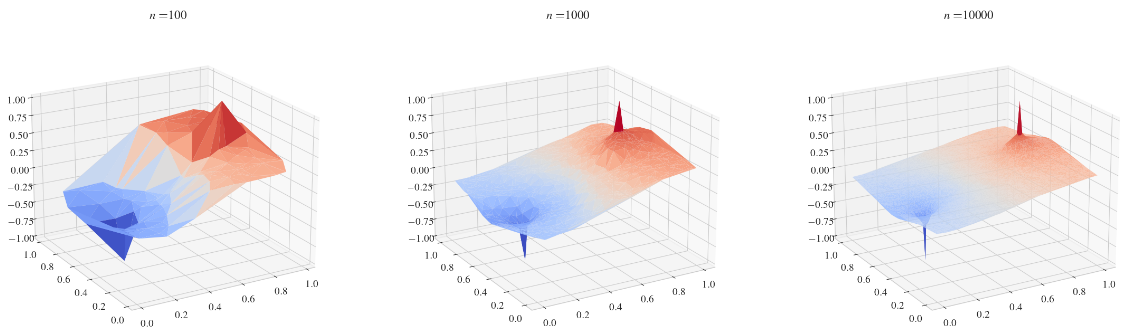


Figure 3.2.: Solution to the Laplacian Learning problem ($p = 2$) for different number of data points $n \in \{100, 1000, 10000\}$.

3.1.3. Consistency for Graph-based SSL

While the graph 2-Laplacian has been successfully employed for classification tasks, see, e.g., [ZGL03; ZS05], it has been observed that solutions tend to degenerate for large amounts of data, whenever $d \geq 2$ [NSZ09; AL11; El +16]. We illustrate this phenomenon in Example 3.13.

Example 3.13. We sample different number of points $n \in \{100, 1000, 10000\}$ in $\Omega = (0, 1)^2$ and keep a fixed constraint on $\mathcal{O}_n = \{o_1, o_2\} = \{(0.2, 0.5), (0.8, 0.5)\}$ with $\mathbf{g}(o_1) = -1, \mathbf{g}(o_2) = 1$. In Fig. 3.2 we observe that for increasing n the solution $\mathbf{u}$ of Eq. (3.2) tends to be a constant equal to the average of the given constraints and spikes at the given constraints.

This observation motivates the driving question of this chapter and the works presented:

Are semi-supervised learning algorithms consistent in the infinite data limit?

In our case, “consistency” roughly ask the discrete solutions to converge to the continuum counterparts in the data limit $n \to \infty$. Here, we distinguish between point-wise consistency as in [VBB08; GK06; HAV05], consistency on a function space level [ST19; GS15; Cal19; LIP-I] and consistency of a discretized operator [Cal19; LIP-II]. Other related works also consider the limit of clustering methods [Hof+22; GS18] or the no-noise limit [Hof+20; Dun+20]. We detail the notion of consistency in the following.

Consistency for the p-Laplacian As observed in [NSZ09; AL11; El +16; CS20] the question of consistency is connected to the relation between p and d, where one considers the three regimes $p < d$, $p = d$ and probably the most relevant case $p > d$. The p-Dirichlet problem in the continuum can be formulated as a variational problem in $W^{1,p}$, however, in our setting we most likely consider pointwise constraints on a discrete set $\mathcal{O}$ even in the continuum. Therefore, one requires that functions in $W^{1,p}$ exhibit some continuity properties. By the Sobolev embedding theorem, this is the case when $p > d$, [AF03]. This leads to a first qualitative intuition that consistency can only be achieved in the case $p > d$. We note that p-harmonic functions are actually more regular also in the case $p \leq d$, beyond the implication of the Sobolev embedding theorem. However, this regularity does not help for the continuum limits [NSZ09; AL11; El +16].

The most relevant result for our contribution in [LIP-I] is the Γ-convergence statement of [GS15], which considers a clustering task, via the TV functional. Here, the authors show

$$\frac{1}{\varepsilon_n n^2}\,\mathbf{E}_1^{w_n} \xrightarrow{\Gamma} \sigma_\eta\,\mathcal{E}_{1,\rho}$$

where the constant σ_η is defined as in Eq. (3.1). We review details on Γ-convergence in Section 3.3. The important insight here, is that the graph scaling ε_n must not tend to zero too fast, which is guided by the following intuition:

- Problems on the graph are non-local, communication between points takes place on a finite scale.
- In the limit we want to obtain a local problem, communication takes place at an infinitesimal small scale.
- If the graph scale ε_n tends to zero too fast, there is not enough communication between vertices, or even worse the graph becomes disconnected.

The optimal scaling obtained in [GS15] for $d \geq 3$ is given as

$$\lim_{n\to\infty} \left(\frac{\log n}{n}\right)^{1/d} \varepsilon_n^{-1} = 0, \tag{3.7}$$

i.e., ε_n must go to zero slower than $\left(\frac{(\log n)}{n}\right)^{1/d}$. This scaling is optimal, in the sense that the graph is disconnected with high probability if $\varepsilon_n < \lambda \left(\frac{\log n}{n}\right)^{1/d}$, see [GS15]. The main difference here, is however, that the considered clustering task does not directly involve a given labeling on the set $\mathcal{O}_n \subset \Omega_n$. This setting was considered in [ST19], where also a generalization of the Γ-convergence statement to the case $p \geq 1$ was provided. Furthermore, the authors consider the constrained model by incorporating the boundary conditions into the functional

$$\mathbf{E}_p^{w_n,\mathrm{cons}}(\mathbf{u}) := \begin{cases} \mathbf{E}_p^{w_n}(\mathbf{u}) & \text{if } \mathbf{u} = \mathbf{g} \text{ on } \mathcal{O}_n, \\ \infty & \text{else.} \end{cases}$$

In order to show Γ-convergence of the constrained functionals, an additional upper bound on the scaling has to be assumed. This means that the graph scaling must not tend to zero too slowly, namely

$$n\,\varepsilon_n^p \xrightarrow{n\to\infty} 0.$$

Together, with Eq. (3.7) this then implies

$$n^{-1/d} \ll \varepsilon_n \ll n^{-1/p}$$

which is only possible if $p > d$. Therefore, one again obtains well-posedness of the constrained problem, if p is large enough.

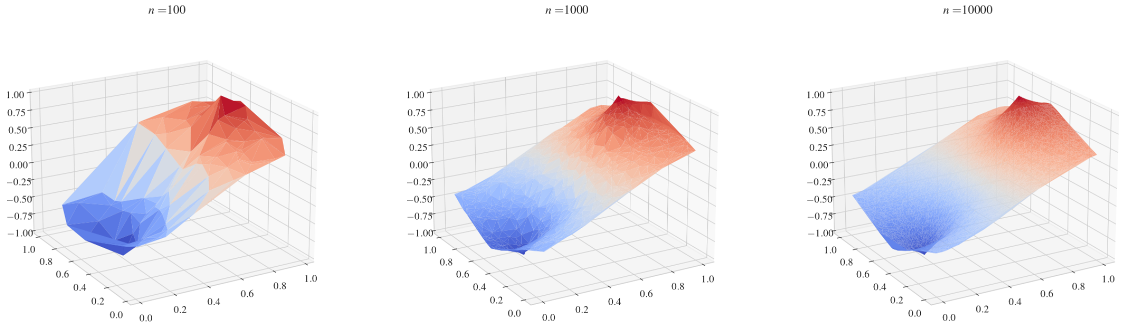


Figure 3.3.: Solution to the Laplacian Learning problem ($p = \infty$) for different number of data points $n \in \{100, 1000, 10000\}$. The setup is otherwise copied from Example 3.13.

Sending p to Infinity Since the assumption that $p > d$ is crucial for asymptotic consistency, a natural idea is to send p to ∞ and analyze the corresponding limit problem. This yields the so-called *Lipschitz learning* task [vLB04; Kyn+15], which is the main point of interest in this chapter. We formalize this problem in Section 3.2. In Fig. 3.3 we observe that for $p = \infty$ we indeed obtain smoother solutions, compared to Fig. 3.2. In this regard, Lipschitz learning seems promising to overcome the consistency issue.

However, as already noticed in [El +16] being a pure L^∞ problem, Lipschitz learning does not properly respect the density of data. This behavior is a drawback in many machine learning applications. However, by carefully rescaling the graph weights one obtains a data sensitive problem, see [Cal19] and Section 3.3. Here, we also want to mention an alternative method to overcome the spiking phenomenon for the p-Laplacian, by re-weighting the graph accordingly [CS20].

The contributions presented in this chapter are all connected to the Lipschitz learning task. Namely, we are able to show consistency and convergence rates under very mild scaling assumptions.

3.2. Lipschitz Extensions and the Infinity Laplacian: Continuum and Graph

Our main point of interest in this chapter is the Lipschitz learning task and its continuum limit. In this section we first motivate the continuum problem as considered in [LIP-I; LIP-II] and then given some details on the discrete problem.

3.2.1. The Continuum Setting

We first recall, that for $u \in W^{1,\infty}(\Omega)$ we have that

$$\lim_{p\to\infty} \mathcal{E}_{p,\rho}(u)^{1/p} = \operatorname*{ess\,sup}_{x\in\operatorname{supp}(\rho)} |\nabla u(x)| =: \mathcal{E}_{\infty,\rho}(u),$$

see [Jen93]. We observe that the weighting ρ only enters via its support, therefore we do not explicitly consider it in the following and instead assume $\Omega = \operatorname{supp}(\rho)$ and only write $\mathcal{E}_\infty$. The functional $\mathcal{E}_\infty$ is weak*-lower semi-continuous over $W^{1,\infty}(\Omega)$, (see, e.g., [BJW01, Thm. 2.6]). In the classical theory developed by Jensen in [Jen93] one considers the following problem, which tries to minimize the *sup-norm* of the gradient as described by Jensen.

Problem 3.14 (Gradient Sup-Norm Problem). For an open domain $\Omega \subset \mathbb{R}^d$, find a function $u \in W^{1,\infty}(\Omega)$ such that

$$\|\nabla u\|_\infty \leq \|\nabla v\|_\infty \qquad \text{for every } v \text{ with } (u - v) \in W_0^{1,\infty}(\Omega).$$

This problem can be seen as the limit problem for $p \to \infty$ of Problem 3.7. Additionally, imposing boundary conditions with $g : \partial\Omega \to \mathbb{R}$, [Jen93] then draws the connection to so-called *Lipschitz extensions*, which are the driving concept in this section. We introduce a more general viewpoint—that does not require the notion of a gradient—later on, but first introduce the variational problem.

The Lipschitz Constant As noticed in [Jen93] working with the Lipschitz constant and the sup-norm of the gradient requires a careful treatment of the distance function. Let $\tilde{\Omega}$ be a set and let $d(\cdot, \cdot)$ be a semi-metric on $\tilde{\Omega}$, that is $d(\cdot, \cdot)$ fulfills the requirement of a metric up to triangle inequality. Then we define the Lipschitz constant of a function $u : V \to \mathbb{R}$ on a subset $V \subset \tilde{\Omega}$ as

$$\operatorname{Lip}_d(u; V) := \sup_{x,y \in V, x \neq y} \frac{|u(x) - u(y)|}{d(x, y)}.$$

If $d(\cdot, \cdot)$ denotes the Euclidean distance we omit the subscript, i.e., $\operatorname{Lip}_{|\cdot - \cdot|} = \operatorname{Lip}$. Additionally, we can introduce the space of Lipschitz functions $\operatorname{Lip}_d(V)$ on V via $u \in \operatorname{Lip}_d(V) \Leftrightarrow \operatorname{Lip}_d(u; V) < \infty$.

Remark 3.15 (Lipschitz and Sobolev functions). If $\Omega \subset \mathbb{R}^d$ is sufficiently regular—e.g., it has Lipschitz boundary—then we have that

$$\operatorname{Lip}(\Omega) = W^{1,\infty}(\Omega),$$

where this identity is of course to be understood in the sense of equivalence classes in L^p spaces. We refer to [Eva18] for a proof of this result. △

The above remark already relates Lipschitz with $W^{1,\infty}$ functions. Often however, we need a quantitative comparison between the Lipschitz constant and the sup-norm of the gradient of a function. Here, it is essential which distance is chosen for the Lipschitz constant. For an open domain $\Omega \subset \mathbb{R}^d$ and $u \in W^{1,\infty}$ we have the inequality

$$\|\nabla u\|_\infty \leq \sup_{x \neq y} \frac{|u(x) - u(y)|}{|x - y|}$$

which can be proven via the definition of the gradient. For the reverse inequality, one has to take the geometry of the domain into account, namely for $x, y \in \Omega$ we have that

$$|u(x) - u(y)| \leq \|\nabla u\|_\infty \; d_\Omega(x, y) \tag{3.8}$$

see, e.g., [BB11, Prop9.3, Rem. 7]. Here,

$$d_\Omega(x, y) = \inf\left\{\int_0^1 |\dot{\gamma}(t)|\, dt : \gamma \in C^1([0,1], \Omega) \text{ with } \gamma(0) = x, \gamma(1) = y\right\}$$

denotes the *geodesic distance* on Ω. If Ω is convex, we have that $d_\Omega(x, y) = |x - y|$ for every $x, y \in \Omega$ and therefore Eq. (3.8) yields $\mathrm{Lip}(u) = \|\nabla u\|_\infty$. This identity generalizes to non-convex domains, namely $\mathrm{Lip}_{d_\Omega}(u) = \|\nabla u\|_\infty$, see [Bun23, Eq. 2.12] or [BDM89, P. 23]. Additionally, it is often necessary to define a distance measure on the closure of $\Omega \subset \mathbb{R}^d$. In order to have a geodesic on $\overline{\Omega}$ one can simply consider $d_{\overline{\Omega}}$, see, e.g., [LIP-II], which then yields the length space $(\overline{\Omega}, d_{\overline{\Omega}})$. In the classical theory developed in [Jen93] one alternatively considers

$$\tilde{d}_{\overline{\Omega}}(x, y) := \liminf_{(\tilde{x}, \tilde{y}) \to (x, y)} d_\Omega(\tilde{x}, \tilde{y}).$$

The differences between these notions are demonstrated in the following example. Also note, that in general $\tilde{d}_{\overline{\Omega}}$ is only a semi-metric on $\overline{\Omega}$ since it is lacking a triangle inequality.

> **Example 3.16.** For $I = [-\pi, c] \cup [c, \pi]$ with $c = \pi/6$ we consider the domain
>
> $$\bigcup_{\theta \in I} B_1\left((\cos(\theta), \sin(\theta)\right)$$
>
> which is visualized in Fig. 3.4 and the points $x = (2\,c, 1), y = (2\,c, -1)$. The line segment between x and y contains the point $z = (2\,c, 0)$, however $z \notin \Omega$. One can show that the geodesic has the length $d_\Omega(x, y) = 4\,\cos(\pi/6) + \pi \approx 6.606$ which is the length of the dotted path in Fig. 3.4. However, we observe that
>
> $$\overline{\Omega} = \bigcup_{\theta \in I} \overline{B_1\left((\cos(\theta), \sin(\theta)\right)}$$
>
> and in particular $z \in \overline{B_1\left((c, c)\right)}$, therefore $d_{\overline{\Omega}}(x, y) = 2$.

Solutions to the Gradient Sup-Norm Problem Before generalizing the theory of Lipschitz extension to arbitrary metric spaces, we first note, that one can explicitly construct solutions of Problem 3.14. Namely, for given $g \in \mathrm{Lip}(\partial\Omega)$ the functions

$$\begin{aligned} \overline{g}(x) &:= \inf_{y \in \partial\Omega} g(y) + \mathrm{Lip}_{\tilde{d}_{\overline{\Omega}}}(g; \partial\Omega) \cdot \tilde{d}_{\overline{\Omega}}(x, y) \\ \underline{g}(x) &:= \sup_{y \in \partial\Omega} g(y) - \mathrm{Lip}_{\tilde{d}_{\overline{\Omega}}}(g; \partial\Omega) \cdot \tilde{d}_{\overline{\Omega}}(x, y) \end{aligned} \tag{3.9}$$

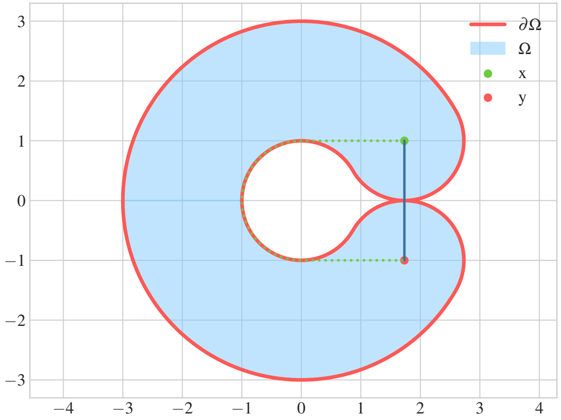


Figure 3.4.: The domain in Example 3.16.

are solutions to the gradient sup-norm problem that coincide with g on $\partial\Omega$, see [Jen93, Thm. 1.8]. The same concept of constructing solutions is applied in the following sections in a more abstract setting. These solutions are then called Whitney and McShane or respectively maximal and minimal extensions, see Lemma 3.19.

One easily observes that there are cases where $\overline{g} \neq \underline{g}$ and therefore the problem does not admit a unique solution. A concrete example, to showcase this phenomenon, is given in [Jen93, p. 53].

Lipschitz Extensions in Metric Spaces The problem considered in the last section was motivated by a variational problem for the functional $\mathcal{E}_\infty(u) = \|\nabla u\|_\infty$. However, the theory of Lipschitz extensions provides a more general framework. Namely, it is not necessary that Ω is a subset of $\mathbb{R}^d$ but we can rather consider a metric space $(\tilde{\Omega}, d)$ with $\Omega \subset \tilde{\Omega}$.

Remark 3.17. For applications within this thesis, we have that $\Omega \subset \mathbb{R}^d$ is an open and bounded domain and consider $\tilde{\Omega} := \overline{\Omega}$, i.e., the closure of Ω within the topology induced by the Euclidean distance. △

A result originally due to Kierszbraun [Kir34] states that for two Hilbert spaces $\mathcal{X}, \mathcal{Y}$, a subset $\mathcal{O} \subset \mathcal{X}$ and a function $g : \mathcal{O} \to \mathcal{Y}$ there exists a function $u : \mathcal{X} \to \mathcal{Y}$ such that

$$\begin{gathered} u = g \text{ on } \mathcal{O}, \\ \operatorname{Lip}(u; \mathcal{X}) = \operatorname{Lip}(g; \mathcal{O}). \end{gathered}$$

Here, the metrics for the respective Lipschitz constants are induced by the inner products of the Hilbert spaces. We refer to [Kir34] for the original proof and to [Sch69, Thm. 1.31] for a proof of the version as stated above. In this work, we only consider the case $\mathcal{Y} = \mathbb{R}$ which allows for more general assumptions on the space $\mathcal{X}$. Below, we formulate the Lipschitz extension problem in our setting.

Problem 3.18 (Lipschitz Extensions). Let $(\tilde{\Omega}, d)$ be a metric space and $\mathcal{O} \subset \tilde{\Omega}$ be a bounded subset. For a given Lipschitz function $g : \mathcal{O} \to \mathbb{R}$ find a Lipschitz function $u : \tilde{\Omega} \to \mathbb{R}$ such that

$$\operatorname{Lip}_d(u; \tilde{\Omega}) = \operatorname{Lip}_d(g; \mathcal{O}).$$

A function $u : \tilde{\Omega} \to \mathbb{R}$ with this property is called *Lipschitz extension* of g to $\tilde{\Omega}$.

In this setting, one can again explicitly construct solutions of the Lipschitz extension task. They are not unique, however, one has an upper and a lower bound. In fact, these solutions are a generalization of the concept in Eq. (3.9).

Lemma 3.19. In the setting of Problem 3.18, we have that the

- **Whitney (or maximal) extension:** $\overline{g}(x) := \inf_{y \in \mathcal{O}} g(y) + \operatorname{Lip}_d(g; \mathcal{O}) \cdot d(x, y)$ and the
- **McShane (or minimal) extension:** $\underline{g}(x) := \sup_{y \in \mathcal{O}} g(y) - \operatorname{Lip}_d(g; \mathcal{O}) \cdot d(x, y)$

defined for $x \in \tilde{\Omega}$ are Lipschitz extensions of g to $\tilde{\Omega}$. Moreover, let $u : \tilde{\Omega} \to \mathbb{R}$ be any Lipschitz extension of g, then we have that

$$\underline{g} \leq u \leq \overline{g}.$$

Proof. We refer to [Whi92] and [McS34] for the proofs of the respective result. □

As demonstrated in Example 3.20, there are cases where $\overline{g} \neq \underline{g}$ and therefore, Lipschitz extensions are not unique in general. Furthermore, [ACJ04] points out that the Whitney and McShane extension do not allow for a comparison principle, which can also be observed in Example 3.20.

Example 3.20. Consider the set $\tilde{\Omega} = [-1, 1]$ and $\mathcal{O} = \{-1, 0, 1\}$ with

$$\begin{aligned} g_1(x) &:= 0, \\ g_2(x) &:= 1/2\ (x - |x|), \\ g_3(x) &:= -g_2. \end{aligned}$$

Then we have that $g_2 \leq g_1$ on $\mathcal{O}$ but

$$\overline{g_2} > \overline{g_1} \text{ in } (0, 1),$$

see Fig. 3.5 for a visualization. Analogously, we have that $g_3 \geq g_1$ on $\mathcal{O}$ but

$$\underline{g_3} < \underline{g_1} \text{ in } (0, 1).$$

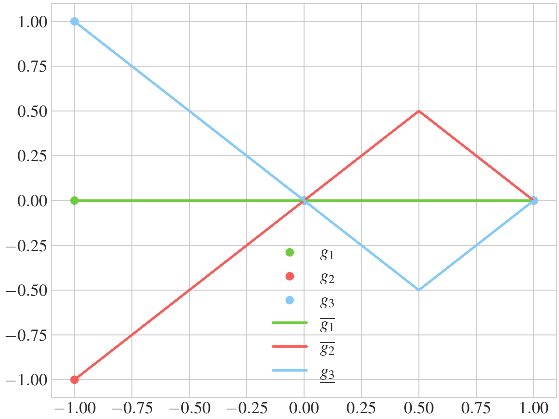


Figure 3.5.: The maximal extension does not admit a comparison principle, as demonstrated in Example 3.20.

Absolutely Minimizing Extension Sending $p \to \infty$ in the variational formulation of the p-Laplace equation yields the Lipschitz extension task, which however does not admit unique solutions. So the question arises, which property is lost in the limit case. For $p < \infty$ one has the local minimization property, as explained in Section 3.1.3. This lead Aronsson to introduce the concept of *absolutely minimizing Lipschitz extension* in [Aro67], by additionally enforcing the minimizing property on every subset. A function $u \in W^{1,\infty}$ is called absolutely minimal, iff

$$\operatorname*{ess\,sup}_{x\in V} |\nabla u| \leq \operatorname*{ess\,sup}_{x\in V} |\nabla v| \qquad \text{for every open } V \subset \Omega \tag{3.10}$$

and every function v such that $(u - v) \in W_0^{1,\infty}$. In fact in [Aro67] it is also shown, that for u_p denoting the solution of the corresponding p-Dirichlet problem, we have that $u_p \xrightarrow{p\to\infty} u_\infty$, which is absolutely minimal. In [ACJ04] it was shown, that one has an equivalent formulation involving the Lipschitz constant. For a given Lipschitz function $g : \overline{\Omega} \to \mathbb{R}$ we have that u_∞ with $(u_\infty - g) \in W_0^{1,\infty}(\Omega)$ fulfills Eq. (3.10) iff

$$\mathrm{Lip}(u_\infty; V) \leq \mathrm{Lip}(v; V) \text{ for every } V \subset \Omega$$

and every function v such that $(u - v) \in W_0^{1,\infty}(V)$, see [Aro67]. In this thesis, we work with a notion of absolute minimizers, which is equivalent to the above formulation for convex domains in $\mathbb{R}^d$. However, for our applications, it is more convenient to formulate the problem for abstract length spaces.

Problem 3.21 (AMLEs). Let $(\tilde{\Omega}, d)$ be a length space, $\mathcal{O} \subset \tilde{\Omega}$ a closed subset and $g : \mathcal{O} \to \mathbb{R}$ a Lipschitz function. Find an extension $u \in C(\tilde{\Omega})$ such that $u = g$ on $\mathcal{O}$ and

$$\mathrm{Lip}_d(u; \overline{V}) = \mathrm{Lip}_d(u, \partial V) \text{ for all open and connected sets } V \subset \tilde{\Omega} \setminus \mathcal{O}. \tag{3.11}$$

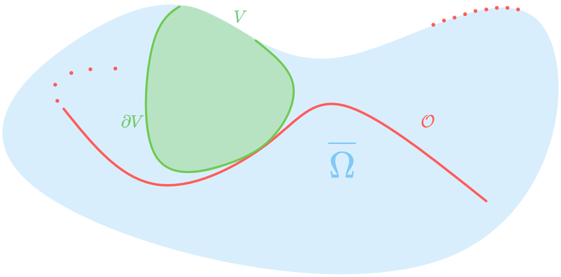


Figure 3.6.: A set $V \subset \overline{\Omega}$ can be relatively open w.r.t. the metric space $\overline{\Omega}$ although, $V \cap \partial\Omega \neq \emptyset$, where $\partial\Omega$ is the boundary within the standard topology on $\mathbb{R}^d$. The relative boundary of $\partial_{\overline{\Omega}} V$ does not include any parts of $\partial\Omega$.

> A function u fulfilling this property is called absolutely minimizing Lipschitz extension of g.

Remark 3.22. In our setting Ω is an open subset of $\mathbb{R}^d$ and we then choose $\tilde{\Omega} = \overline{\Omega}$. Here, it is important to note that the topological notions like boundary and interior are to be understood relative to $\overline{\Omega}$. A visualization of this concept can be found in Fig. 3.6. $\triangle$

Fulfilling Eq. (3.11) also makes sense without enforcing boundary conditions on some set $\mathcal{O}$. Typically, one just says u is absolutely minimizing (AM) if it Eq. (3.11) holds, see [ACJ04]. In the convex and Euclidean case, existence of solutions to Problem 3.21 follow directly from Aronssons proof that the limit $u_p \xrightarrow{p\to\infty} u_\infty$ fulfills Eq. (3.10), see [Aro67]. For general length spaces, it can be shown via an adapted Perron's method [Per23], which we state in the following result from [Juu02].

> **Theorem 3.23 ([Juu02, Thm. 4.3]).** Let $(\tilde{\Omega}, d)$ be a separable length space, and suppose $g : \mathcal{O} \to \mathbb{R}$ is Lipschitz. Then there exists an absolutely minimizing Lipschitz extension u of g to $\tilde{\Omega}$, i.e., u solves Problem 3.21.

We address the question of existence and uniqueness of solutions to Problem 3.21 in the following paragraphs.

The Infinity Laplacian In the Euclidean case, one can derive an operator equation by considering the limit of the p-Laplacian operator as $p \to \infty$. This yields the ∞-Laplacian, which is defined as

$$\Delta_\infty u = \langle \nabla u, D^2 u \, \nabla u \rangle = \sum_{i,j} \partial_{x_i} u \, \partial_{x_j} u \, \partial_{x_i, x_j} u$$

for functions of class C^2. This operator had one of its first appearances in [Aro68], but we refer to [Lin17] for a detailed overview of the topic. The brief exposition, here, also follows [Lin17] to some extent. Intuitively, the ∞-Laplace operator gives the second derivative of u into the direction of its gradient. The associated infinity Laplace equation takes the following form.

Problem 3.24. Let $\Omega \subset \mathbb{R}^d$ be an open and bounded domain and $g \in W^{1,\infty}$, then the task is to find u such that

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

At first glance, the definition of the ∞-Laplacian only makes sense for functions of class C^2. However, as shown in [Yu06; Aro68] if $u \in C^2(\Omega)$ fulfills $\Delta_\infty u = 0$ in Ω, then either $\nabla u \neq 0$ in the whole domain or it is constant. However, one can easily construct boundary values g that are not constant but force the solution u to have critical points inside the domain, see, e.g., [Lin16]. Therefore, one requires a different concept, where one typically employs *viscosity solutions*. For the ∞-Laplacian, this strategy was first used in [BDM89], where the term *viscosity* is motivated by similar concepts for the Burgers' equation [Bur48]. First we consider subsolutions, where we take any $\phi \in C^2(\Omega)$ that touches u from above at any $x \in \Omega$, i.e.,

$$u < \phi \text{ in } \Omega \setminus \{x\} \qquad \text{and} \qquad u(x) = \phi(x). \tag{3.12}$$

Since ϕ is C^2 we can shift the application of Δ_∞ onto this function and then say u is a subsolution if for any ϕ fulfilling Eq. (3.12) we have

$$-\Delta_\infty \phi(x) \leq 0.$$

Note, that the ∞-Laplacian is only evaluated at the touching point $x \in \Omega$. Analogously, we say u is a supersolution if for any ϕ touching from below we have $-\Delta_\infty \phi(x) \geq 0$. If u is both a sub- and a supersolution, we say it is a solution in the viscosity sense.

Remark 3.25. The concept of viscosity solutions can similarly be applied to the p-Laplacian or even a more general class of differential operators, see [Lin17]. We also note, that one has the consistency result, that if $u \in C^2$ is a viscosity solution, we also have that $\Delta_\infty u = 0$ in the classical sense. △

Solving the ∞-Laplace equation in the viscosity sense is in fact equivalent to being absolutely minimizing, which we state in the following theorem taken from [ACJ04]

Theorem 3.26 ([ACJ04, Thm. 4.13]). A function $u \in C(\overline{\Omega})$ fulfills $\Delta_\infty u = 0$ in Ω in the viscosity sense iff it is an absolutely minimizing function on Ω.

In Problem 3.24 we only consider the case where boundary values are given on $\partial\Omega$. However, for our application we would rather have a Dirichlet condition on the set $\mathcal{O} \subset \overline{\Omega}$, since it often does not make sense to prescribe boundary values on $\partial\Omega$. Therefore, we assume Neumann boundary conditions on this boundary and then consider the problem

$$\begin{aligned} \Delta_\infty u &= 0 \text{ in } \Omega \setminus \mathcal{O}, \\ \frac{\partial u}{\partial \nu} &= 0 \text{ on } \partial\Omega \setminus \mathcal{O}, \\ u &= g \text{ on } \mathcal{O}. \end{aligned} \tag{3.13}$$

In the case that Ω is smooth and convex, from [ASS11, Lem. 3.1] we have that $u \in C(\overline{\Omega})$ is an AMLE of $g : \mathcal{O} \to \mathbb{R}$ to $\overline{\Omega}$ iff it fulfills Eq. (3.13).

We can now also address the question of uniqueness, where we state the following famous result of Jensen [Jen93].

> **Theorem 3.27 ([Jen93]).** Let $u, v \in C(\overline{\Omega})$ be two viscosity solutions of the ∞-Laplacian, then we have
>
> $$\max_{\overline{\Omega}}(u - v) = \max_{\partial\Omega}(u - v).$$

Originally proven by Jensen in [Jen93], there exists a very simple proof by Armstrong and Smart [AS10] employing the *comparison with cones* which we detail in the following paragraph.

Comparison with Cones As shown in [ACJ04] the concept of absolutely minimizing extensions is equivalent to so-called *comparison with cones*. For some metric $d(\cdot,\cdot)$ a cone function is defined as

$$x \mapsto a\, d(x, z) + c$$

where $z \in \Omega$ denotes the origin or cone tip, $a \geq 0$ its opening angle and $c \in \mathbb{R}$ some offset. The property we consider in the following, basically asks if a function u is smaller than a cone on the boundary of some set V not including the tip z, then it should be smaller also in the interior of the domain. This means for any $a \geq 0, c \in \mathbb{R}$ and $z \notin V$ we have the implication

$$[u \leq a\, d(\cdot, z) + c \text{ on } \partial V] \quad \Rightarrow \quad [u \leq a\, d(\cdot, z) + c \text{ in } V]\,.$$

One can omit the explicit use of the offset $c \in \mathbb{R}$ and equivalently consider the property

$$\max_{\partial V} \left(u - a\, d(\cdot, z)\right) = \max_{\overline{V}} \left(u - a\, d(\cdot, z)\right). \tag{3.14}$$

In the Euclidean case, cone functions are infinity harmonic away from their cone tip, since they are constant in the direction of their gradient. We make the explicit computation below

$$\begin{aligned}\partial_{x_i}|x-z| &= \frac{x_i - z_i}{|x-z|}\\ \partial_{x_i x_j}|x-z| &= -\frac{(x_i-z_i)(x_j-z_j)}{|x-z|^3} \text{ for } i\neq j,\\ \partial^2_{x_i}|x-z| &= \frac{1}{|x-z|^3}\sum_{j\neq i}(x_j-z_j)^2,\\ \Rightarrow \Delta_\infty |x-z| &= \sum_i \frac{(x_i-z_i)^2}{|x-z|^5}\sum_{j\neq i}(x_j-z_j)^2\\ &\quad - \sum_{i\neq j}\frac{(x_i-z_i)^2(x_j-z_j)^2}{|x-z|^5} = 0.\end{aligned}$$

In the Euclidean case as in [ACJ04] one says a function fulfills *comparison with cones* from above, if it fulfills Eq. (3.14) in Ω for every subset $V\subset\subset\Omega$, every $a\geq 0$ and $z\in\mathbb{R}^n\setminus V$. We say u fulfills *comparison with cones* from below if $-u$ fulfills comparison from above. If it fulfills both comparisons from above and below, we say it fulfills comparison with cones. Here, [ACJ04, Prop. 2.1] shows the following equivalence to being absolutely minimizing, which however mainly focuses on the Euclidean case. Our setting slightly deviates from the Euclidean one, where in [LIP-II] we consider the following notion.

Definition 3.28 ([LIP-II, Def. 4.1]). We shall say that an upper semicontinuous function $u\in USC(\overline{\Omega})$ satisfies CDF from above in $\overline{\Omega}\setminus\mathcal{O}$, if for each relatively open and connected subset $V\subset\overline{\Omega}\setminus\mathcal{O}$, any $x_0\in\overline{\Omega}\setminus V$ and $a\geq 0$ we have

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

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

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

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

This definition is a special case of the notion in [JS06]. Therein, the authors also show the equivalence to the absolutely minimizing property, which we state in the following.

Theorem 3.29 ([JS06, Prop. 4.1]). A function $u\in C(\overline{\Omega})$ is absolutely minimizing iff it fulfills comparison with cones in Ω.

Therefore, we can rewrite Problem 3.21 employing the comparison with cones condition. Our metric space is given as $(\overline{\Omega}, d_{\overline{\Omega}})$, where $\Omega\subset\mathbb{R}^n$ is some open set in the Euclidean sense, therefore we have a separable length space, where Theorem 3.23 yields existence of solutions for Problem 3.21. Concerning uniqueness, we refer to [Per+09] which were the first to prove uniqueness in the length space setting. For a more general result, we refer to [NS12]. In our work, we obtain uniqueness by a careful adaption of the proof in [AS10].

> **Proposition 3.1 ([LIP-II, Prop. 4.11]).** Consider the metric space $(\overline{\Omega}, d_{\overline{\Omega}})$, then Problem 3.21 has at most one solution.

3.2.2. Graph Lipschitz Extensions

We now consider the limit $p \to \infty$ of Problem 3.9 in the graph case. Analogously to Section 3.2.1 we derive

$$\lim_{p\to\infty} \left(\mathbf{E}_p^{w_n}(\mathbf{u})\right)^{1/p} = \max_{x,y\in\Omega_n} w_n(x,y)\,|\mathbf{u}(y)-\mathbf{u}(x)| =: \mathbf{E}_\infty^{w_n}(\mathbf{u})$$

which extends the graph p-Dirichlet energy to the case $p=\infty$. Again, we notice structural similarities to the continuum version $\mathcal{E}_\infty$.

Remark 3.30. Informally speaking, the functional $\mathbf{E}_\infty^{w_n}$ combines elements of a gradient and a Lipschitz constant. Assuming that $w_n(x,y)$ relates to $1/|x-y|$ we see that the finite difference approximation resembles a Lipschitz constant. However, in the limit $n\to\infty$ the weighting $w_n(x,y)$ has a localizing property which fits the interpretation of a gradient better. △

This functional yields the graph Lipschitz extension problem.

> **Problem 3.31 (Graph Energy Minimization).** Given a weighted graph (Ω_n, w_n) and a labeling function $\mathbf{g} : \mathcal{O}_n \to \mathbb{R}$, for $\mathcal{O}_n \subset \Omega_n$ we consider the problem
>
> $$\min_{\mathbf{u}:\Omega_n\to\mathbb{R}} \mathbf{E}_\infty^{w_n}(\mathbf{u}) \text{ subject to } \mathbf{u}(x) = \mathbf{g}(x) \text{ for all } x \in \mathcal{O}_n.$$

Since the weighting function $w_n : \Omega_n \times \Omega_n \to \mathbb{R}_0^+$ does not induce a metric, Problem 3.31 does not directly fit the framework of the abstract Lipschitz extension in Problem 3.18. However, we can consider paths in (Ω_n, w_n), connecting arbitrary $x, y \in \Omega_n$, i.e., $\gamma \in \Omega_n^{\times k}, x = \gamma_1, \dots, \gamma_k = y$ such that

$$w_n(\gamma_i, \gamma_{i+1}) > 0 \quad \text{for all } i = 1, \dots, k-1,$$

for which we define the length as

$$|\gamma| = \sum_{i=1}^{k-1} w_n(\gamma_i, \gamma_{i+1})^{-1}.$$

This yields the metric space (Ω_n, d_{w_n}), where $d_{w_n} : \Omega_n \times \Omega_n \to \mathbb{R}$ is defined as

$$d_{w_n}(x,y) := \min\left\{|\gamma| : \gamma \text{ is a path in } (\Omega_n, w_n) \text{ from } x \text{ to } y\right\}. \tag{3.15}$$

Remark 3.32. We note that it is important to only consider non-negative weights, otherwise any loop with a negative "length" would decrease the length of the whole path arbitrarily. Restricting ourselves to non-negative weights, we can easily see that the minimum in Eq. (3.15) is indeed attained. △

With this definition, we can consider the Lipschitz extension task of $g : \mathcal{O}_n \to \mathbb{R}$ to Ω_n within the metric space (Ω_n, d_{w_n}), i.e., within the setting of Problem 3.18. Therefore, the question arises, whether the minimization problem in Problem 3.31 is equivalent to the metric Lipschitz extension problem, for which we have the following lemma.

Lemma 3.33. For a graph (Ω_n, w_n) with non-negative weights and a function $\mathbf{u} : \Omega_n \to \mathbb{R}$ we have that

$$\mathbf{E}_\infty^{w_n}(\mathbf{u}) = \mathrm{Lip}_{d_{w_n}}(\mathbf{u}).$$

Furthermore, for $\mathcal{O}_n \subset \Omega_n$ and a function $\mathbf{g} : \mathcal{O}_n \to \mathbb{R}$ and we have that

$$\mathbf{g} = \mathbf{u} \text{ on } \mathcal{O}_n \Rightarrow \mathrm{Lip}_{d_{w_n}}(\mathbf{g}; \mathcal{O}_n) \leq \mathrm{Lip}_{d_{w_n}}(\mathbf{u}).$$

Proof. **Step 1:** We show that $\mathrm{Lip}_{d_{w_n}}(\mathbf{u}) \leq \mathbf{E}_\infty^{w_n}(\mathbf{u})$.
We can choose a path $\gamma \in \Omega_n^{\times k}$ such that

$$\mathrm{Lip}_{d_{w_n}}(\mathbf{u}) = \frac{\mathbf{u}(\gamma_1) - \mathbf{u}(\gamma_k)}{|\gamma|}.$$

The path γ allows to compare vertices $\gamma_1, \gamma_k \in \Omega_n$ that are not necessarily neighbors in the graph. However, each consecutive vertices in the path are neighbors in the graph and therefore we have

$$w_n(\gamma_i, \gamma_{i+1})\, |\mathbf{u}(\gamma_{i+1}) - \mathbf{u}(\gamma_i)| \leq \mathbf{E}_\infty^{w_n}(\mathbf{u}) \quad \text{for all } i = 1, \dots, k-1. \tag{3.16}$$

We now employ an elementary result for numbers $a_i \in \mathbb{R}_0^+, b_i \in \mathbb{R}^+, i = 1, \dots m \in \mathbb{N}$, namely

$$[a_i \cdot b_i \leq c \in \mathbb{R} \quad \text{for } i = 1, \dots, m] \quad \Rightarrow \quad \frac{\sum_{i=1}^m a_i}{\sum_{i=1}^m b_i^{-1}} \leq c \tag{3.17}$$

which can be seen as follows

$$\begin{aligned} a_i \cdot b_i \leq c \quad & \text{for } i = 1, \dots, m, \\ \Rightarrow a_i \leq b_i^{-1} \cdot c \quad & \text{for } i = 1, \dots, m, \\ \Rightarrow \sum_{i=1}^m a_i \leq \left(\sum_{i=1}^m b_i^{-1} \right) \cdot c, & \\ \Rightarrow \frac{\sum_{i=1}^m a_i}{\sum_{i=1}^m b_i^{-1}} \leq c. & \end{aligned}$$

This then yields

$$\frac{|\mathbf{u}(\gamma_1) - \mathbf{u}(\gamma_k)|}{|\gamma|} \leq \frac{\sum_{i=1}^{k-1} |\mathbf{u}(\gamma_i) - \mathbf{u}(\gamma_{i+1})|}{|\gamma|} = \frac{\sum_{i=1}^{k-1} |\mathbf{u}(\gamma_i) - \mathbf{u}(\gamma_{i+1})|}{\sum_{i=1}^{k-1} w_n(\gamma_i, \gamma_{i+1})^{-1}} \leq \mathbf{E}_\infty^{w_n}(\mathbf{u})$$

where in the last inequality we employed Eq. (3.17) together with Eq. (3.16).
Step 2: We show that $\mathrm{Lip}_{d_{w_n}}(\mathbf{u}) \geq \mathbf{E}_\infty^{w_n}(\mathbf{u})$.
Let $x, y \in \Omega_n$, then we know that $d_w(x,y) \leq w_n(x,y)^{-1}$ and therefore

$$|\mathbf{u}(x) - \mathbf{u}(y)|\, w_n(x,y) \leq \frac{|\mathbf{u}(x) - \mathbf{u}(y)|}{d_w(x,y)} \leq \max_{\bar{x},\bar{y} \in \Omega_n} \frac{|\mathbf{u}(\bar{x}) - \mathbf{u}(\bar{y})|}{d_w(\bar{x},\bar{y})} = \mathrm{Lip}_{d_{w_n}}(\mathbf{u}).$$

Since this holds for arbitrary $x, y \in \Omega_n$ we have that

$$\mathbf{E}_\infty^{w_n}(\mathbf{u}) = \max_{x,y \in \Omega_n} |\mathbf{u}(x) - \mathbf{u}(y)|\, w_n(x,y) \leq \mathrm{Lip}_{d_{w_n}}(\mathbf{u}).$$

Step 3: We show that $\mathrm{Lip}_{d_{w_n}}(\mathbf{g}; \mathcal{O}_n) \leq \mathrm{Lip}_{d_{w_n}}(\mathbf{u})$.
If $\mathbf{g} = \mathbf{u}$ on $\mathcal{O}$ this simply follows since the maximum for the Lipschitz constant of $\mathbf{u}$ is taken over a larger set. Indeed, we have that

$$\begin{aligned}\mathrm{Lip}_{d_{w_n}}(\mathbf{u}) &= \max_{x,y \in \Omega_n} \frac{|\mathbf{u}(x) - \mathbf{u}(y)|}{d_w(x,y)} \geq \max_{x,y \in \mathcal{O}_n} \frac{|\mathbf{u}(x) - \mathbf{u}(y)|}{d_w(x,y)} = \max_{x,y \in \mathcal{O}_n} \frac{|\mathbf{g}(x) - \mathbf{g}(y)|}{d_w(x,y)} \\ &= \mathrm{Lip}_{d_{w_n}}(\mathbf{u}; \mathcal{O}_n).\end{aligned}$$

□

This lemma shows that the abstract Lipschitz extension task considered on the metric space (Ω_n, d_{w_n}) and the Graph ∞-Dirichlet minimization task are indeed equivalent. Therefore, we also have that the Whitney and McShane extensions

$$\begin{aligned}\overline{\mathbf{g}}(x) &= \inf_{y \in \mathcal{O}_n} \mathbf{g}(y) + d_{w_n}(x,y) \\ \underline{\mathbf{g}}(x) &= \sup_{y \in \mathcal{O}_n} \mathbf{g}(y) - d_{w_n}(x,y)\end{aligned}$$

are solutions on the graph. Analogously, the problem does not admit unique solutions.

Absolutely Minimizing Graph Extensions Similarly to Section 3.2.1 we can now consider absolutely minimizing extensions. However, the problem in Problem 3.21 uses a notion of a boundary and it is not directly clear how to infer this concept to the discrete set Ω_n. Therefore, we define what we mean by "boundary" on a graph.

> **Definition 3.34.** Let (Ω_n, w_n) be a weight graph and let $V \subset \Omega_n$ be a subset, then we define
>
> - the **exterior** boundary as $\partial^{\text{ext}} := \{x \in \Omega_n \setminus V : w_n(x,y) > 0 \text{ for some } y \in V\}$,

- the **interior** boundary as $\partial^{\text{int}} := \{x \in V : w_n(x, y) > 0 \text{ for some } y \in \Omega_n \setminus V\}$.

The closure of V is then defined as $\overline{V}^{\text{ext}} := V \cup \partial^{\text{ext}}$ and the interior as $\overset{\circ}{V}^{\text{int}} := V \setminus \partial^{\text{int}} V$.

We note that it is not possible to define a topology on Ω_n that would yield the above notions. Namely, the only admissible topology in our case would be the discrete topology, i.e., 2^{Ω_n}. In this topology the only closed sets are $\emptyset$ and Ω_n which is not useful for the applications in the following. Using the Kuratowski closure axioms [Kur22] we remark that the exterior closure on the graph is a so-called pre- or Čech closure, [ČFK66]. For a set $\mathcal{X}$, a mapping $\text{cl} : 2^{\mathcal{X}} \to 2^{\mathcal{X}}$ is called Čech closure, if the following conditions hold,

- $\text{cl}(\emptyset) = \emptyset$ (cl preserves the empty set),
- $V \subset \text{cl}(V)$ for all $V \subset \mathcal{X}$ (cl is extensive),
- $\text{cl}(V_1 \cup V_2) = \text{cl}(V_1) \cup \text{cl}(V_2)$ for all $V_1, V_2 \subset \mathcal{X}$ (cl preserves binary unions).

Lemma 3.35. The exterior closure on a weighted graph (Ω_n, w_n) is a preclosure or Čech closure.

Proof. We first see that $\overline{\emptyset}^{\text{ext}} = \emptyset$ and that $V \subset \overline{V}^{\text{ext}}$ for every subset $V \subset \Omega_n$, i.e., the above defined closure preserves the empty set and is extensive. Furthermore, for two sets $V_1, V_2 \subset \Omega_n$ we have that

$$\begin{gathered} x \in \partial^{\text{ext}}(V_1 \cup V_2) \\ \Leftrightarrow [x \notin V_1 \cup V_2] \wedge [\exists y \in V_1 \cup V_2 : w_n(x, y)] \neq 0 \\ \Leftrightarrow [x \notin V_1 \cup V_2] \wedge \Big([\exists y \in V_1 : w_n(x, y) \neq 0] \vee [\exists y \in V_2 : w_n(x, y) \neq 0] \Big) \\ \Leftrightarrow \Big[x \in \partial^{\text{ext}} V_1 \setminus V_2\Big] \vee \Big[x \in \partial^{\text{ext}} V_2 \setminus V_1\Big] \\ \Leftrightarrow x \in (\partial^{\text{ext}} V_1 \cup \partial^{\text{ext}} V_2) \setminus (V_1 \cup V_2). \end{gathered}$$

We have shown that $\partial^{\text{ext}}(V_1 \cup V_2) = (\partial^{\text{ext}} V_1 \cup \partial^{\text{ext}} V_2) \setminus (V_1 \cup V_2)$. Therefore, we have that

$$\begin{aligned} \overline{V_1 \cup V_2}^{\text{ext}} &= V_1 \cup V_2 \cup \partial^{\text{ext}}(V_1 \cup V_2) \\ &= V_1 \cup V_2 \cup ((\partial^{\text{ext}} V_1 \cup \partial^{\text{ext}} V_2) \setminus (V_1 \cup V_2)) \\ &= V_1 \cup \partial^{\text{ext}} V_1 \cup V_2 \cup \partial^{\text{ext}} V_2 \\ &= \overline{V_1}^{\text{ext}} \cup \overline{V_2}^{\text{ext}}. \end{aligned}$$

This shows that the closure preserves binary unions and therefore we have shown, that it is indeed a Čech closure. □

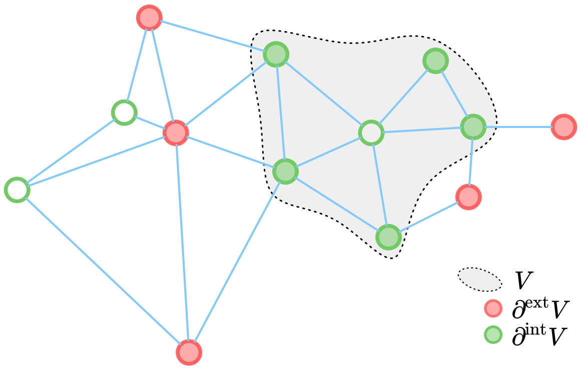


Figure 3.7.: Visualization of exterior and interior boundary on a graph.

The missing property, that inhibits the closure to induce a topology, is the so-called idempotence. Namely, there are sets $V \subset \Omega_n$ such that

$$\overline{V}^{\text{ext}} \neq \overline{\overline{V}^{\text{ext}}}^{\text{ext}}.$$

In the example visualized in Fig. 3.7 we see that $\overline{\overline{V}^{\text{ext}}}^{\text{ext}} = \Omega_n \neq \overline{V}^{\text{ext}}$. Since the closure we employ here does not induce a topology, we have a slightly modified notion of absolutely minimizers.

Problem 3.36 (Graph AMLEs). Given a connected weighted graph (Ω_n, w_n), $\mathcal{O}_n \subset \Omega_n$ and a function $\mathbf{g} : \mathcal{O}_n \to \mathbb{R}$, find a function $\mathbf{u} : \Omega_n \to \mathbb{R}$ such that

$$\begin{aligned} \operatorname{Lip}_{d_{w_n}}(\mathbf{u}; \overline{V}^{\text{ext}}) &= \operatorname{Lip}_{d_{w_n}}(\mathbf{u}; \partial^{\text{ext}} V) \text{ for all connected } V \subset \Omega_n \setminus \mathcal{O}_n, \\ \mathbf{u} &= \mathbf{g} \text{ on } \mathcal{O}_n. \end{aligned}$$

This notion of absolute minimizers is employed in [LIP-II; LIP-III]. We see, that graph AMLEs are indeed special solutions of the basic Lipschitz extension problem on the graph, by choosing $V = \Omega_n \setminus \mathcal{O}_n$ in the above problem.

Comparison with Graph Distance Functions Analogously to the continuum case Section 3.2.1, we can also consider comparison with distance functions, but now on graphs. The main ingredients here are the graph distance function d_{w_n} and the notion of closure on a graph, as developed in the last section.

Definition 3.37. For a weighted graph (Ω_n, w_n) we say a function $\mathbf{u} : \Omega_n \to \mathbb{R}$ fulfills comparison with distance function from above (CDFA) on a subset $U \subset \Omega_n$ if for every $V \subset U$ we have

$$\max_{\overline{V}^{\text{ext}}} (u - a\ d_{w_n}(\cdot, z)) = \max_{\partial^{\text{ext}} V} (u - a\ d_{w_n}(\cdot, z)) \tag{CDFA}$$

for every $z \in \Omega_n \setminus V$ and every $a \geq 0$. We say that $\mathbf{u}$ fulfills comparison with distance function from below (CDFB) on a subset $U \subset \Omega_n$ if for every $V \subset U$ we have

$$\min_{\overline{V}^{\text{ext}}} (u + a\ d_{w_n}(\cdot, z)) = \min_{\partial^{\text{ext}} V} (u + a\ d_{w_n}(\cdot, z)) \tag{CDFB}$$

for every $z \in \Omega_n \setminus V$ and every $a \geq 0$.

Analogously to the continuum case, we say that a function fulfills comparison with distance functions, if it fulfills both, (CDFA) and (CDFB).

The Graph ∞-Laplacian We can also obtain the limit of the Graph p-Laplace operator via the following calculation,

$$\begin{aligned} \Delta_p^{w_n} \mathbf{u}(x) &= 0 \\ \Leftrightarrow \sum_{y \in \Omega_n} w_n(x,y)^p \left|\mathbf{u}(y) - \mathbf{u}(x)\right|^{p-2} (\mathbf{u}(y) - \mathbf{u}(x)) &= 0 \\ \Leftrightarrow \sum_{y:\mathbf{u}(x) \leq \mathbf{u}(y)} w_n(x,y)^p (\mathbf{u}(y) - \mathbf{u}(x))^{p-1} &= \\ \sum_{y:\mathbf{u}(x) > \mathbf{u}(y)} w_n(x,y)^p (\mathbf{u}(x) - \mathbf{u}(y))^{p-1}&. \end{aligned}$$

Taking the terms on the left and right-hand side to the power of $1/p$ and then sending $p \to \infty$ yields

$$\begin{aligned} \max_{y:\mathbf{u}(x) \leq \mathbf{u}(y)} w_n(x,y)(\mathbf{u}(y) - \mathbf{u}(x)) &= \max_{y:\mathbf{u}(x) > \mathbf{u}(y)} w_n(x,y)(\mathbf{u}(x) - \mathbf{u}(y)) \\ \Leftrightarrow \max_{y \in \Omega_n} w_n(x,y)(\mathbf{u}(y) - \mathbf{u}(x)) &= -\min_{y \in \Omega_n} w_n(x,y)(\mathbf{u}(y) - \mathbf{u}(x)). \end{aligned}$$

This calculation motivates the definition of the graph infinity Laplacian

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

which then allows to formulate the associated problem as an extension of Problem 3.11.

Problem 3.38 (Graph ∞-Laplacian). Given a weighted graph (Ω_n, w_n) and a labeling function $\mathbf{g} : \mathcal{O}_n \to \mathbb{R}$ with $\mathcal{O}_n \subset \Omega_n$, find a function $\mathbf{u} : \Omega_n \to \mathbb{R}$ such that

$$\begin{aligned} \Delta_\infty^{w_n} \mathbf{u} &= 0, \text{ in } \Omega_n \setminus \mathcal{O}_n, \\ \mathbf{u} &= \mathbf{g} \text{ on } \mathcal{O}_n. \end{aligned}$$

To establish existence of solutions of the problem above, one can employ the Perron method [Per23]. Alternatively, one can establish the equivalence to so-called lex-minimizers and show existence as in [Kyn+15, Thm. 3.3]. Uniqueness is a consequence of the following theorem from [Cal19], which is similar to the uniqueness result of Jensen in Theorem 3.27.

Theorem 3.39 ([Cal19, Thm. 3.1]). Let (Ω_n, w_n) be a graph connected to the boundary $\mathcal{O}_n \subset \Omega_n$ and consider $\mathbf{u}, \mathbf{v} : \Omega_n \to \mathbb{R}$ such that

$$\Delta_\infty^{w_n} \mathbf{u} \leq 0 \leq \Delta_\infty^{w_n} \mathbf{v} \quad \text{in } \Omega_n \setminus \mathcal{O}_n.$$

Then we know that

$$\max_{\Omega_n}(\mathbf{u} - \mathbf{v}) = \max_{\mathcal{O}_n}(\mathbf{u} - \mathbf{v})$$

Relations Between the Graph Lipschitz Extensions We now establish the connection between the different notions of Lipschitz extensions on the graph. Compared to the continuum case, we do not establish the full equivalences, but only the necessary implications required for the convergence proofs in [LIP-II]. Namely, in [LIP-II] we show that if $\mathbf{u}$ solves the graph ∞-Laplace, then it is a graph AMLE and fulfills comparison with graph distance functions.

Lemma 3.40 ([LIP-II, Thm. 3.2, Prop. 3.8]). Let (Ω_n, w_n) be a weighted connected graph and $\mathbf{g} : \mathcal{O}_n \to \mathbb{R}$ be a given function for $\mathcal{O}_n \subset \Omega_n$. Furthermore, let $\mathbf{u} : \Omega_n \to \mathbb{R}$ be graph infinity harmonic on $\Omega_n \setminus \mathcal{O}_n$ with boundary conditions given by $\mathbf{g}$, i.e., $\mathbf{u}$ solves Problem 3.38 then we have that

- $\mathbf{u}$ is a graph AMLE, i.e., $\mathbf{u}$ solves Problem 3.36,
- $\mathbf{u}$ fulfills comparison with cones.

Proof. Both of the statements are proven in [LIP-II]. Let $\mathbf{u}$ be such that

$$\begin{gathered}\Delta_\infty^{w_n} \mathbf{u} = 0 \text{ in } \Omega_n \setminus \mathcal{O}_n\\ \mathbf{u} = \mathbf{g} \text{ on } \mathcal{O}_n.\end{gathered}$$

From [LIP-II, Prop. 3.8] we have that $\mathbf{u}$ is a graph AMLE. Furthermore, form [LIP-II, Thm. 3.2] we have that $\mathbf{u}$ fulfills comparison with cones. In fact, [LIP-II, Thm. 3.2], shows a more refined statement, namely that

$$\begin{gathered}-\Delta_\infty^{w_n} \mathbf{u} \leq 0 \Rightarrow \mathbf{u} \text{ fulfills (CDFA)},\\ -\Delta_\infty^{w_n} \mathbf{u} \geq 0 \Rightarrow \mathbf{u} \text{ fulfills (CDFB)}.\end{gathered}$$

□

3.3. Gamma Convergence: [LIP-I]

In this section, we present the main results of [LIP-I]. This work considers the limit of the basic Lipschitz extension from the graph Problem 3.31 to the continuum case Problem 3.18. As detailed in the previous sections, both of these problems do not admit unique solutions, but it is still meaningful to study the convergence in the infinite data limit. We start by specifying the setting and recalling the concept of Γ-convergence and then state the main results.

Main Contributions in [LIP-I] We first prove Γ-convergence of the functionals $\mathbf{E}_\infty^{w_n}$ to their continuum counterpart $\mathcal{E}_\infty$. This result can be seen as the extension of the results in [GS15; ST19] to the case $p = \infty$. Here, we employ a different type of metric as we detail in the following. Furthermore, we identify the optimal graph scaling, which in this case is derived in a deterministic manner. The key difference to [GS15; ST19] is that we can work with any point clouds and not only i.i.d. points. The inherent reason is that L^∞ problems only consider mass in a qualitative not in a quantitative way. We then establish a compactness result, that allows us to infer convergence of minimizers. Finally, we apply this theory to the ground state problem.

Parts of this work were motivated by the findings in TR's master thesis [Roi21]. The convergence result in the latter, was however not fully established and therefore [LIP-I] constitutes a non-trivial expansion and also correction. We want to highlight two major differences:

- The domain Ω in [Roi21] was assumed to be convex. This restriction was omitted in [LIP-I] by identifying a class of locally convex domains, that prevent internal sharp corners.
- We are able to work with asymptotic boundary conditions, i.e., the boundary conditions for each graph problem do need to be the same as in the continuum. They only need to approximate them in some sense.

3.3.1. Setting and Preliminaries

We detail the concrete setting of [LIP-I] and provide some background on the employed notions.

Γ-Convergence Originally, the concept of Γ-convergence dates back to De Giorgi [DF75] as a type of variational convergence. We refer to [Bra02; Dal12] for a detailed overview on this notion and related topics. While Γ-convergence was successfully employed in a pure continuum setting for a longer time (see, e.g., [Mod77]), it was more recently used to prove convergence from a discrete to a continuum functional [CGL10; BY12; VB+12].

Definition 3.41 ([DF75]: Γ-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 $(x)_{n\in\mathbb{N}} \subset X$ 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 $(x)_{n\in\mathbb{N}} \subset X$ converging to x and

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

The most relevant reference for [LIP-I] was the disruptive work presented by García Trillos and Slepčev in [GS15], which is also commented on in Section 3.1. Among other important ideas and notions, we want to highlight two ingredients that directly influenced [LIP-I]:

- Γ-convergence on a common metric space, that allows to compare graph functions with continuum functions.
- The proof strategy of first considering the discrete to non-local convergence and then non-local to continuum.

Employing Γ-convergence the authors in [ST19] were able to prove convergence of minimizers of the graph p-Dirichlet problem. Here, one can employ the convenient result that Γ-convergence implies convergence of minimizers.

Lemma 3.42 ([Bra02, Thm. 1.21] 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.

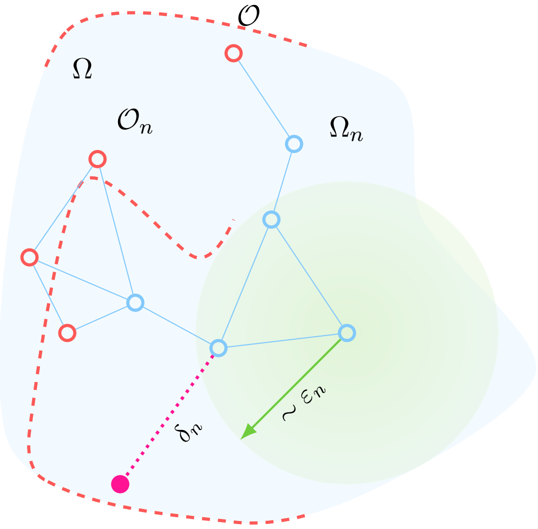
The Kernel and the Graph Scale In Section 3.1 we comment on the relevance of the kernel and graph scaling ε_n for continuum limits. We first state the concrete assumptions we make in [LIP-I] on the kernel $\eta : \mathbb{R}_0^+ \to \mathbb{R}_0^+$, namely we require

(K1) η is positive and continuous at 0,

(K2) η is non-increasing,

(K3) $\operatorname{supp}(\eta) \subset [0, t_\eta]$ for some $t_\eta > 0$.

As seen in [GS15; ST19] and Eq. (3.1) we also need to consider the constant σ_η which in our case is given by



$$\sigma_\eta = \operatorname*{ess\,sup}_{t\in\mathbb{R}^+} \eta(t)\, t = \lim_{p\to\infty} \left(\int_{\mathbb{R}^+} \eta(t)^p\, t^{d+p} dt \right)^{1/p}.$$

A main difference to the results for $p < \infty$ is that we can work with point clouds Ω_n, that do not necessarily need to be constructed by an i.i.d. sampling process. However, this rises the question of how we can ensure that $\Omega_n \subset \overline{\Omega}$ fills out the continuum domain $\overline{\Omega}$ sufficiently fast. In [LIP-I] we consider the maximum distance of any continuum point to its next vertex

$$\sup_{x\in\overline{\Omega}} \min_{y\in\Omega_n} |x-y|, \tag{3.18}$$

as a measure of the distance between Ω_n and $\overline{\Omega}$. In fact, this is the Hausdorff distance ([Hau14]) between the two sets, since

$$d_H(\Omega_n, \overline{\Omega}) = \max\left\{ \sup_{x\in\overline{\Omega}} \inf_{y\in\Omega_n} |x-y|, \sup_{x\in\Omega_n} \inf_{y\in\overline{\Omega}} |x-y| \right\} = \sup_{x\in\Omega} \min_{y\in\Omega_n} |x-y|,$$

where the second term vanishes since $\Omega_n \subset \overline{\Omega}$.

Remark 3.43. We remark on the historical correctness of the name “Hausdorff” distance, motivated by the review in [BT06]. This distance can actually be contributed to Pompeiu, who employed a similar notion in [Pom05], which also relates to concepts developed in Fréchet's Ph.D. thesis [Fré06]. Hausdorff modified this distance into the presented form in [Hau14]. △

We also allow the constraint set $\mathcal{O}_n$ to vary with each $n \in \mathbb{N}$, therefore we also measure the distance between $\mathcal{O}_n$ and the continuum constraint set $\mathcal{O} \subset \overline{\Omega}$, via

$$d_H(\mathcal{O}_n, \mathcal{O}) = \max\left\{ \sup_{x\in\mathcal{O}} \inf_{y\in\mathcal{O}_n} |x-y|, \max_{x\in\mathcal{O}_n} \inf_{y\in\mathcal{O}} |x-y| \right\}$$

where the second term does not vanish, because in general $\mathcal{O}_n \not\subseteq \mathcal{O}$. We assume that we have a Lipschitz continuous function $g : \overline{\Omega} \to \mathbb{R}$ such that for each n the boundary conditions are given by the function $\mathbf{g}_n = g|_{\mathcal{O}_n}$. In total, we then need to control the graph resolution

$$\delta_n := \max\{d_H(\Omega_n, \overline{\Omega}), d_H(\mathcal{O}_n, \mathcal{O})\}.$$

This then allows us to formulate our scaling assumption, namely we require that the graph resolution tends to zero faster than the graph scaling

$$\frac{\delta_n}{\varepsilon_n} \longrightarrow 0, \quad n \to \infty, \tag{3.19}$$

which has similarly employed in [Pen99b; Pen99a]. This is the weakest possible assumption that ensures connectivity of the graph in the limit $n \to \infty$. However, compared to Eq. (3.7) it is more of deterministic nature, since we do not assume anything on how Ω_n is created. In case of a i.i.d. point cloud, one can show that

$$\delta_n \sim \left(\frac{\log n}{n}\right)^{1/d},$$

almost surely as $n \to \infty$, see [Pen99a], which is the setting in [GS15].

Assumptions on the Domain As we mention in the introduction, we do not need to restrict ourselves to convex domains. However, for our convergence results, it is important that the geodesic distance

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

converges to the Euclidean distance if x goes to y. We formulate this in the following assumption

$$\lim_{\delta \downarrow 0} \sup\left\{\frac{d_\Omega(x, y)}{|x - y|} : x, y \in \Omega,\ |x - y| < \delta\right\} \leq 1. \tag{3.20}$$

The inherent reason that we need to enforce this, is that edges in the graph are allowed to communicate via their direct Euclidean distance, even if this line does not lie in Ω. While it would be easier to also consider the geodesic distance on the graph, this is unrealistic in many applications since we do not have access to the concrete shape of Ω. Most importantly, Eq. (3.20) prohibits sharp internal corners in the domain, see Fig. 3.8. Furthermore, note that we consider the geodesic distance d_Ω on the open domain Ω, not on its closure. Therefore, we also exclude situations of a touching boundary, as in Fig. 3.4. In [LIP-I] we also provide examples of domains fulfilling this condition.

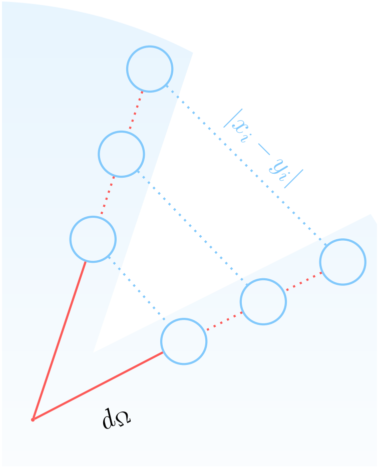


Figure 3.8.: An example of a sharp internal corner, violating Eq. (3.20). The geodesic distance between the pairs of points converging to the corner point is bigger than then Euclidean one by a constant factor.

3.3.2. Γ-Convergence of the Discrete Functionals

In order to show convergence of discrete functionals defined on Ω_n to continuum functions acting on $\overline{\Omega}$, one needs to define a common metric space. Employing results from [TS15] in [GS15] the authors introduced the space TL^p

$$TL^p := \{(\mu, u) : \mu \in \mathcal{P}(\Omega), u \in L^p(\mu)\}$$

together with the transport distance

$$d_{TL^p}((\mu,u),(\nu,v)) = \inf_{\pi \in \Gamma(\mu,\nu)} \left(\int_{\Omega\times\Omega} |x-y|^p + |u(x) - v(y)|^p \, d\pi(x,y) \right)^{1/p}$$

where $\Gamma(\mu,\nu)$ is the set of coupling between μ and ν. A sequence (μ_n, u_n) converges to (μ, u) in TL^p iff there exists a sequence of transportation maps $T_n : \Omega \to \Omega$ with $T_n \# \mu = \mu_n$ and

$$\int_\Omega |x - T_n(x)| \, d\mu(x) \to 0$$

such that $u_n \circ T_n \xrightarrow{L^p} u$, [GS15, Prop. 3.12]. Therefore, the maps T_n allow us to employ standard convergence in L^p. In order to transfer this situation to L^∞ one could try to employ a ∞-Wasserstein distance. However, as argued in [Roi21] the strategy does not transfer directly, since convergence in W^∞ does not metrize weak convergence of measures [San15, Thm. 5.10]. However, as seen in [LIP-I] one can employ a more direct argument. For problems in L^p the conservation of mass was important for the maps T_n (i.e., $T_n \# \mu = \mu_n$) such that the integrals could be transformed. This condition is irrelevant in L^∞, namely, we have the following analogous transformation rule in L^∞.

Lemma 3.44 ([LIP-I, Lem. 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 $u : \Omega \to \mathbb{R}$ we have that

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

We want to compare the discrete measure $\mu_n = \frac{1}{n}\sum_{x\in\Omega_n}\delta_x$ to the target measure μ. In this setting, a closest point projection $p_n : \Omega \to \Omega_n$

$$p_n(x) \in \operatorname{argmin}_{y\in\Omega_n} |x-y|$$

fulfills the assumption of Lemma 3.44. This allows us to extend the functional $\mathbf{E}_\infty^{w_n}$ in Problem 3.31 to L^∞ via

$$\mathbf{E}_\infty^{w_n}(u) = \begin{cases} \mathbf{E}_\infty^{w_n}(\mathbf{u}) & \text{if } u = \mathbf{u}\circ p_n, \text{ for some } \mathbf{u} : \Omega_n \to \mathbb{R}, \\ \infty & \text{else },\end{cases} \tag{3.21}$$

which was similarly done in [GS15; ST19]. Additionally, we incorporate the constraint on $\mathcal{O}_n$ in Problem 3.31 via

$$\mathbf{E}_\infty^{w_n,\mathrm{cons}}(\mathbf{u}) := \begin{cases} \mathbf{E}(\mathbf{u}) & \text{if } \mathbf{u} = \mathbf{g} \text{ on } \mathcal{O}_n, \\ \infty & \text{else,}\end{cases}$$

with the analogous extension to L^∞ as in Eq. (3.21). We can now state the first main result of [LIP-I] that provides the Γ-convergence of the graph functionals.

Theorem 3.45 (Discrete to continuum Γ-convergence). Let $\Omega \subset \mathbb{R}^d$ be a domain satisfying (3.20), let the kernel fulfill (K1)-(K3), then for any null sequence $(\varepsilon_n)_{n\in\mathbb{N}} \subset (0,\infty)$ which satisfies the scaling condition (3.19) we have

$$\mathbf{E}_\infty^{n,\mathrm{cons}} \xrightarrow{\Gamma} \sigma_\eta\, \mathcal{E}^{\mathrm{cons}}. \tag{3.22}$$

The main proof strategy here is similar to the one in [GS15; ST19]. Namely, one defines the non-local functional

$$\mathcal{E}_\infty^\varepsilon(u) := \frac{1}{\varepsilon} \operatorname*{ess\,sup}_{x,y\in\Omega} \left\{\eta_\varepsilon(|x-y|)\, |u(x)-u(y)|\right\}, \quad \varepsilon > 0$$

for which we show that for any sequence $\varepsilon_n \to 0$ we have

$$\mathcal{E}_\infty^{\varepsilon_n} \xrightarrow{\Gamma} \sigma_\eta\, \mathcal{E}_\infty,$$

see [LIP-I, Thm. 4]. For the liminf inequality of the discrete functionals, one has to take special care of points $x, y \in \Omega_n$ where $\eta_{\varepsilon_n}(|p_n(x) - p_n(y)|) = 0$. We want to bound $\mathbf{E}_\infty^{w_n}$ from below by $\mathcal{E}_\infty^{\varepsilon_n}$ for which we have to permit significant communication of x and y whenever $p_n(x)$ and $p_n(y)$ do not communicate. This can be done, (temporarily assuming η is constant on $[0, t]$) by introducing a smaller length scale $\tilde{\varepsilon}$ such that

(i) $|p_n(x) - p_n(y)| > t\varepsilon_n \Rightarrow |p_n(x) - p_n(y)| / \varepsilon_n < |x - y| / \tilde{\varepsilon}$,

(ii) $\lim_{n\to\infty} \tilde{\varepsilon}/\varepsilon = 1$.

As shown in [LIP-I] the choice $\tilde{\varepsilon} = \varepsilon - 2\delta/t$ fulfills (i). So in order to fulfill (ii) we obtain the scaling condition

$$\lim_{n\to\infty} \frac{\delta_n}{\varepsilon_n} = 0 \Rightarrow \lim_{n\to\infty} \frac{\varepsilon_n - 2\delta_n/t}{\varepsilon_n} = 1 - \frac{2}{t} \lim_{n\to\infty} \frac{\delta_n}{\varepsilon_n} = 1.$$

This then allows us to prove the liminf inequality since $\mathbf{E}_\infty^{w_n}(u_n) \geq \frac{\tilde{\varepsilon}_n}{\varepsilon_n} \mathcal{E}_\infty^{\tilde{\varepsilon}_n}(u_n)$ holds. The limsup inequality can then be shown by choosing the constant sequence, with some additional care for the changing constraint set $\mathcal{O}_n$.

3.3.3. Convergence of Minimizers

The convenient aspect of Γ-convergence is, that under additional compactness properties it directly shows convergence of minimizers, [Bra02, Thm. 8]. This yields the second main result in [LIP-I].

Theorem 3.46 ([LIP-I, Thm. 2]). Let $\Omega \subset R^d$ be a domain satisfying (3.20), let the kernel fulfil (K1)-(K3), and $(\varepsilon_n)_{n\in\mathbb{N}} \subset (0, \infty)$ be a null sequence which satisfies the scaling condition (3.19). Then any sequence $(u)_{n\in\mathbb{N}} \subset L^\infty(\Omega)$ such that

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

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

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

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

In order to show the compactness in the above theorem, one employs the following lemma.

Lemma 3.47 ([LIP-I, Lem. 4]). 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

$$\mu\text{-}\operatorname*{ess\,sup}_{x,y\in V_i} |u(x) - u(y)| < \varepsilon \ \forall u \in K, i = 1, \dots, n, \tag{3.23}$$

then K is relatively compact.

Remark 3.48. The proof of this statement employs ideas from [DS88, Lem. IV.5.4] and appeared similarly in TR's master thesis. However, therein the statement was slightly wrong, which was corrected in [LIP-I]. △

This lemma is the used to show that if a sequence u_n fulfills

$$\sup_{n\in\mathbb{N}} \mathbf{E}_\infty^{w_n,\mathrm{cons}}(u_n) < \infty,$$

then it is relatively compact.

3.3.4. Application to Ground States

The convergence results of the previous paragraphs can be applied to the ground states problem. Namely, for a p-homogeneous functional $\mathcal{E}$—i.e., $\mathcal{E}(cu) = |c|^p\,\mathcal{E}(u)$ for $c \in \mathbb{R}$—we consider the following problem

$$\min\left\{\mathcal{E}(u) \,:\, \inf_{v\in\operatorname{argmin}\mathcal{E}} \|u - v\| = 1\right\},$$

as studied in [BKB20]. In our setting, if $g \equiv 0$ then the functionals $\mathbf{E}_\infty^{w_n,\mathrm{cons}}$ and $\mathcal{E}_\infty^{\mathrm{cons}}$ are 1-homogeneous. In [LIP-I, Thm. 5] we first show that the up to a global sign unique ground state of $\mathcal{E}_\infty^{\mathrm{cons}}$ is a multiple of the distance function

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

We then employ the Γ-convergence results to show the convergence of discrete ground states to the above continuum one.

Theorem 3.49 ([LIP-I, Thm. 6]: Convergence of Ground States). Under the conditions of [LIP-I, Thm. 1] let the sequence $(u_n)_{n\in\mathbb{N}} \subset L^\infty(\Omega)$ fulfill

$$\mathbf{u}_n \in \operatorname{argmin}\left\{\mathbf{E}_\infty^{w_n,\mathrm{cons}}(\mathbf{u}) \,:\, \mathbf{u} \in L^\infty(\Omega),\ \|\mathbf{u}\|_{L^p} = 1\right\}.$$

Then (up to a subsequence) $\mathbf{u}_n \to u$ in $L^\infty(\Omega)$ where

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

and it holds

$$\lim_{n\to\infty} \mathbf{E}_\infty^{w_n,\mathrm{cons}}(\mathbf{u}_n) = \sigma_\eta\, \mathcal{E}_\infty^{\mathrm{cons}}(u). \tag{3.24}$$

3.4. Uniform Convergence of AMLEs: [LIP-II]

In the previous section, we consider the limit of the Lipschitz extension task. The major limitations of this work are:

- The considered problem does not admit unique solutions. The Γ-convergence framework does not directly allow us to select certain minimizers, e.g., AMLEs, and consider the respective limit to continuum AMLEs.
- We can only show qualitative convergence results. It is not directly possible to prove quantitative statements and convergence rates.

These drawbacks are the main motivation for the work presented in this section. We first state the main contributions and then provide details below.

Main Contributions in [LIP-II] We prove a quantitative convergence results for discrete AMLEs to their continuum counterpart. Convergence was already established in [Cal19], however employing a more restrictive scaling assumption and only allowing for smooth kernels η, which was due to the viscosity techniques employed. By using the AMLE characterization of infinity harmonic functions, we are able to show convergence rates under the weakest scaling assumption and allowing for very general kernels. This is done via a using a comparison principle on a homogenized length scale. The key insight, we obtain in this work, is that convergence rates for graph distance functions imply rates for AMLEs. Finally, we examine some numerical convergence rates.

3.4.1. Setting

Conceptually, the basic setting is very similar to the previous one in Section 3.3. We highlight some minor differences in the paragraphs below.

Assumptions on the Kernel We employ (K2) and (K3) from [LIP-I] and set $t_\eta = 1$ for simplicity. However, we do not need to assume continuity in 0, i.e. (K1). Instead, we additionally require the following assumption:

(K4) We assume that there exists $t_0 \in (0, 1]$, chosen as the largest number with this property with $\sigma_\eta = t_0 \eta(t_0)$.

Assuming (K2) to (K4) allows for the so-called singular weights

$$\eta(t) = \frac{1}{t^p} 1_{[0,1]}(t)$$

with $p \in (0, 1]$. As noted in [LIP-II] the above assumption is not redundant, since for example the function

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

violates (K4), while fulfilling (K2) and (K3).

Assumptions on the Domain Again we take the concept from [LIP-I], where we assume asymptotic convexity. We alter the Eq. (3.20) slightly to be more quantitative. We assume that 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

$$d_{\overline{\Omega}}(x,y) \leq |x-y| + \phi(|x-y|), \quad \text{for all } x,y \in \Omega : |x-y| \leq r_\Omega. \tag{3.25}$$

Furthermore, for $\varepsilon > 0$ we define $\sigma_\phi(\varepsilon) = \sup_{0<s\leq\varepsilon} \frac{\phi(s)}{s}$. Note that we now employ the geodesic distance $d_{\overline{\Omega}}$, i.e., paths can also lie on the boundary of Ω.

Assumptions on the Labeling Function The assumption on the labeling for each sub-problem is weakened even further. Namely, we do not assume the labels for each discrete problem to be given by the continuum constraint $g : \mathcal{O} \to \mathbb{R}$. Instead, we require

(C1) $\sup_{n\in\mathbb{N}} \operatorname{Lip}_n(\mathbf{g}_n; \mathcal{O}_n) < \infty$ and $\operatorname{Lip}_\Omega(g;\mathcal{O}) < \infty$,

(C2) there exists $C > 0$ such that for all $z \in \mathcal{O}$ it holds that $|\mathbf{g}_n(\pi_{\mathcal{O}_n}(z)) - g(z)| \leq C\delta_n$.

This assumption on the labeling function, is also connected to the no-noise and infinite data limit, see, e.g., [Hof+20].

3.4.2. Convergence Results

In [LIP-I] the proof strategy relied on Γ-convergence. In [LIP-II] we use the comparison with cones characterization of AMLEs together with comparison principles for the infinity Laplace operator. An important strategy is to consider the homogenized operator

$$\Delta_\infty^\tau u(x) = \frac{1}{\tau^2}\left(\inf_{y\in B_\tau(x)} (u(y)-u(x)) + \sup_{y\in B_\tau(x)} (u(y)-u(x))\right) \tag{3.26}$$

on the length scale $\tau > 0$, which is typically larger than the graph scale ε. The results in the following are non-asymptotic, for which also our scaling assumption reads as follows,

$$\begin{aligned} \varepsilon &\leq r_\Omega, \\ \frac{\varepsilon}{\tau} &< \frac{1}{2}, \\ \sigma_\phi(\varepsilon) + \frac{\delta_n}{\varepsilon} &\leq \frac{t_0}{4+2\sigma_\phi(\varepsilon)}\left(1 - 2\frac{\varepsilon}{\tau}\right), \end{aligned} \tag{3.27}$$

for some $n \in \mathbb{N}$. We detail the concrete appearance of the operator Δ_∞^τ in the following paragraphs, and first state the main general result. We employ the notation $a \lesssim b$, which means that $a \leq C\,b$ for some constant $C > 0$ depending only on g and η, but not on n, ε or δ.

Theorem 3.50 ([LIP-II, Thm. 2.2]). Let $\tau > 0$, let the kernel and data fulfill (K2) to (K4), (C1) and (C2) and let (3.25) and (3.27) hold. Let $\mathbf{u}_n : \Omega_n \to \mathbb{R}$ solve Problem 3.36, and $u : \overline{\Omega} \to \mathbb{R}$ solve Problem 3.21.

1. It holds

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

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

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

Here, $u_n : \overline{\Omega} \to \mathbb{R}$ denotes the piecewise constant extension of $\mathbf{u}_n$, as in (3.21).

We observe that in the case where $|\nabla u|$ is bounded away from zero, the rate improves, which is due to a perturbation we briefly mention in the following. The parameter τ appears due to our proof strategy and does not have a concrete meaning for the final problem. An immediate consequence is that for an arbitrary $\tau > 0$, we obtain

$$\sup_{\Omega_n} |u - \mathbf{u}_n| < \tau$$

by sending δ_n, ε_n to zero as $n \to \infty$ employing the weakest scaling assumption $\delta_n \ll \varepsilon_n$. Since $\tau > 0$ was arbitrary, this already yields convergence of discrete AMLEs under the weakest scaling assumption [LIP-II, Cor. 2.3]. In order to obtain the convergence rates involving only δ and ε we can "optimize" over τ. We first consider the small length scale regime, which in our case includes any graph scaling such that $\varepsilon \lesssim \delta_n^{5/9}$. In that case we can choose $\tau = ((\delta_n + \phi(\varepsilon))/\varepsilon)^{1/4}$ to obtain

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

If $|\nabla u|$ is bounded away from zero, this can be improved to

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

where we consider the regime $\varepsilon \lesssim \delta^{3/5}$. In the convex case, i.e., $\phi \equiv 0$ we therefore directly obtain the rates

$$\inf_{\overline{\Omega}} |\nabla u| = 0 : \left(\frac{\delta_n}{\delta_n^{5/9}}\right)^{\frac{1}{4}} = \delta_n^{\frac{1}{9}},$$

$$\inf_{\overline{\Omega}} |\nabla u| > 0 : \left(\frac{\delta_n}{\delta_n^{3/5}}\right)^{\frac{1}{2}} = \delta_n^{\frac{1}{5}},$$

see [LIP-II, Cor. 2.4]. In fact the convexity assumption is too strict, instead we just have to require that $\phi(\varepsilon) \lesssim \varepsilon^p$ for some power of $p \in \mathbb{R}^+$ that does not deteriorate the rate. This is done in [LIP-II, Cor. 2.5, 2.6], where also a better rate is obtained by allowing for larger length scales. In the following, we sketch the main ingredients of the proof. While in the Γ-convergence case we employ the hierarchy

$$\text{“discrete} \to \text{non-local} \to \text{continuum”},$$

our scheme now has the form

$$\text{“discrete} \to \text{homogenized non-local} \leftarrow \text{continuum”}.$$

We discuss the two directions below and then discuss how to “meet in the middle”.

Non-Local to Continuum In this paragraph, we comment on how to relate continuum AMLEs to the homogenized operator in Eq. (3.26). Here, it is convenient to copy the following definitions from [LIP-II],

$$T^\tau u(x) := \sup_{\overline{B}_\Omega(x,\tau)} u, \qquad T_\tau u(x) \ := \inf_{\overline{B}_\Omega(x,\tau)} u, \tag{3.28}$$

$$S^+_\tau u := \frac{1}{\tau}(T^\tau u - u), \qquad S^-_\tau u := \frac{1}{\tau}(u - T_\tau u), \tag{3.29}$$

so that we can write $\Delta^\tau_\infty u = \frac{1}{\tau}\left(S^+_\tau u - S^-_\tau u\right)$. The astonishing property of this operator is the so-called “max-ball” lemma. If $u \in USC(\overline{\Omega})$ fulfills comparison with distance functions from above, then we have that the function $T^\tau u$ fulfills

$$-\Delta^\tau_\infty T^\tau u \le 0, \quad \text{ in } \Omega^{2\tau}_{\mathcal{O}}, \tag{3.30}$$

analogously if $u \in LSC(\overline{\Omega})$ fulfills CDF form below we have

$$-\Delta^\tau_\infty T_\tau u \ge 0, \quad \text{ in } \Omega^{2\tau}_{\mathcal{O}}. \tag{3.31}$$

Here, we denote by $\Omega_{\mathcal{O}} = \overline{\Omega} \setminus \mathcal{O}$ and additional employ the inner parallel set,

$$\Omega^\tau_{\mathcal{O}} := \{x \in \overline{\Omega} \,:\, \operatorname{dist}_\Omega(x, \mathcal{O}) > \tau\}. \tag{3.32}$$

Both, Eqs. (3.30) and (3.31) are shown in [LIP-II, Lem. 4.6], again borrowing concepts from [AS10]. In fact, the proof is a simple consequence of the comparison with cones characterization. The strategy works as follows:

- Consider the ball $B(x, 2\tau)$.
- For any $y \in B(x, 2\tau)$ we have that

$$\begin{aligned} u(y) &= u(x) + u(y) - u(x) = u(x) + \frac{u(y) - u(x)}{d_{\overline{\Omega}}(x,y)} d_{\overline{\Omega}}(x,y) \\ &\le u(x) + \frac{T^{2\tau} u(x) - u(x)}{d_{\overline{\Omega}}(x,y)} d_{\overline{\Omega}}(x,y). \end{aligned}$$

- Consider the set $V := B(x, 2\tau) \setminus \{x\}$ where on its boundary we can substitute the denominator by 2τ to obtain

$$u(y) \leq u(x) + \frac{T^{2\tau}u(x) - u(x)}{2\tau} d_{\overline{\Omega}}(x, y) \quad \text{for } y \in \partial V.$$

- Employing comparison with distance functions, this inequality holds on the whole set $\overline{V}$, therefor we can take the maximum over $B(x, \tau)$ and see

$$T^\tau u(x) \leq \frac{T^{2\tau}u(x) - u(x)}{2\tau}\tau = \frac{1}{2}(T^{2\tau}u(x) + u(x)).$$

- Rearranging the terms yields

$$\begin{aligned} 0 &\geq -\left(T^{2\tau}u + u - 2T^\tau u\right) = -\left(T^\tau T^\tau u + u - 2T^\tau u\right) \\ &\geq -\left(T^\tau T^\tau u + T_\tau T^\tau u - 2T^\tau u\right) = -\tau^2 \Delta_\infty^\tau T^\tau u \end{aligned}$$

It is interesting that the inequality is exact, i.e., we do not need to introduce an error term of order τ. This observation allows us to pass from the continuum setting to the non-local one.

Discrete to Non-Local Compared to the previous paragraph, passing from the discrete problem to the non-local one is more challenging. Analogously to the continuum case, where we considered the operators T^τ, T_τ we now define the functions

$$u_n^\tau(x) := \sup_{\overline{B}_\Omega(x,\tau)\cap\Omega_n} \mathbf{u}_n, \quad (u_n)_\tau(x) := \inf_{\overline{B}_\Omega(x,\tau)\cap\Omega_n} \mathbf{u}_n, \quad x \in \overline{\Omega}. \tag{3.33}$$

Concerning the connection between the discrete problem and the non-local operator, we state the following main result from [LIP-II].

Theorem 3.51 ([LIP-II, Thm. 5.13]). Assume that (K2) to (K4) and (3.25) and (3.27) hold. Let $\mathbf{u}_n : \Omega_n \to \mathbb{R}$ solve the graph infinity Laplacian equation Problem 3.36. Then there exists a constant $C > 0$ such that for all $x_0 \in \Omega_{\mathcal{O}}^{2\tau+3\delta_n}$ it holds

$$-\Delta_\infty^\tau u_n^\tau(x_0) \leq \operatorname{Lip}_n(\mathbf{g}_n) C \left(\frac{\delta_n}{\varepsilon\tau} + \frac{\varepsilon}{\tau^2} + \frac{\phi(\varepsilon)}{\varepsilon\tau}\right), \tag{3.34a}$$

$$-\Delta_\infty^\tau (u_n)_\tau(x_0) \geq -\operatorname{Lip}_n(\mathbf{g}_n) C \left(\frac{\delta_n}{\varepsilon\tau} + \frac{\varepsilon}{\tau^2} + \frac{\phi(\varepsilon)}{\varepsilon\tau}\right). \tag{3.34b}$$

The above statement again only holds true on a inner parallel set, like [LIP-II, Lem. 4.6]. We later employ Lipschitz properties to obtain a result on the whole domain. Similarly, in the proof we first consider vertices $\mathbf{x}_0 \in \Omega_{\mathcal{O}}^{2\tau+3\delta_n} \cap \Omega_n$ together with $\mathbf{p}_n^\tau(\mathbf{x}_0), \mathbf{p}_n^{2\tau}$ such that

$$\begin{aligned} u_n^\tau(\mathbf{x}_0) &= \sup_{\overline{B}_\Omega(\mathbf{x}_0,\tau)\cap\Omega_n} \mathbf{u}_n = \mathbf{u}_n(\mathbf{p}_n^\tau(\mathbf{x}_0)),\\ u_n^{2\tau}(\mathbf{x}_0) &= \sup_{\overline{B}_\Omega(\mathbf{x}_0,2\tau)\cap\Omega_n} \mathbf{u}_n = \mathbf{u}_n(\mathbf{p}_n^{2\tau}(\mathbf{x}_0)). \end{aligned}$$

Plugging in the definition of Δ_∞^τ we then obtain the inequality

$$-\tau^2\Delta_\infty^\tau u_n^\tau(\mathbf{x}_0) \leq 2\mathbf{u}_n(\mathbf{p}_n^\tau(\mathbf{x}_0)) - \mathbf{u}_n(\mathbf{p}_n^{2\tau}(\mathbf{x}_0)) - \mathbf{u}_n(\mathbf{x}_0).$$

Here, we would now like to employ comparison with graph distance functions for a similar strategy as in the continuum case. However, the choice of the set such that we have a desirable inequality on its boundary is not trivial. This is due to the fact, that for u_n^τ we take the supremum over a Euclidean ball, but we want to apply comparison with graph distance function. As in [LIP-I] we observe that our arguments need more care, whenever the Euclidean and the graph distance meet. We remark how the situation changes from the comparison with cones application in the continuum, where we assume that $\phi \equiv 0$ for simplicity, the concrete details are given in [LIP-II, Sec. 5.2]:

- We know that $u_n^{2\tau}$ maximizes on $\overline{B}(2\tau, \mathbf{x}_0)$.
- Since $d_{w_n}(\mathbf{x}_0, \mathbf{w}) \leq |\mathbf{x}_0 - \mathbf{w}|$ we only need to shrink the graph ball by ε_n to ensure, that its closure lies in $B(2\tau, \mathbf{x}_0)$. This means we can consider

  $$B = \{\mathbf{w} \in \Omega_n : d_{w_n}(\mathbf{x}_0, \mathbf{w}) \leq 2\tau - \varepsilon\}$$

  and we have that $\overline{B} \subset B(2\tau, \mathbf{x}_0)$.
- In the continuum case, we knew that every point $y \in \partial B(2\tau, \mathbf{x}_0)$ had exactly distance 2τ from x_0. This value is not as simple in the graph case, we need to consider the value

  $$D = \inf_{\mathbf{y}\in\Omega_n\setminus\overline{B}(\mathbf{x}_0,2\tau-\varepsilon)} d_{w_n}(\mathbf{x}_0, \mathbf{y}) \geq 2\tau - \varepsilon.$$
- Employing comparison with graph distance functions we then obtain

  $$\mathbf{u}_n(\mathbf{w}) \leq \mathbf{u}_n(\mathbf{x}_0) + \frac{\mathbf{u}_n(\mathbf{p}_n^{2\tau}(\mathbf{x}_0)) - \mathbf{u}_n(\mathbf{x}_0)}{D} d_{w_n}(\mathbf{x}_0, \mathbf{w}) \tag{3.35}$$

  for all vertices $\mathbf{w} \in \Omega_n$ with $d_{w_n}(\mathbf{x}_0, \mathbf{w}) \leq 2\tau - \varepsilon$.
- We now choose $\mathbf{w} = \mathbf{p}_n^\tau(\mathbf{x}_0)$, however can not simply control the distance to $\mathbf{x}_0$ by τ instead we have the value

  $$N = \sup_{\mathbf{y}\in\overline{B}(\mathbf{x}_0,\tau)\cap\Omega_n} d_{w_n}(\mathbf{x}_0, \mathbf{y})$$

and obtain

$$\begin{aligned}2\mathbf{u}_n(\mathbf{p}_n^\tau(\mathbf{x}_0)) &\leq 2\mathbf{u}_n(\mathbf{x}_0) + (\mathbf{u}_n(\mathbf{p}_n^{2\tau}(\mathbf{x}_0)) - \mathbf{u}_n(\mathbf{x}_0))\frac{2N}{D}\\ &= 2\mathbf{u}_n(\mathbf{x}_0) + (\mathbf{u}_n(\mathbf{p}_n^{2\tau}(\mathbf{x}_0)) - \mathbf{u}_n(\mathbf{x}_0))\left(1 - 1 + \frac{2N}{D}\right)\\ &= \mathbf{u}_n(\mathbf{x}_0) + \mathbf{u}_n(\mathbf{p}_n^{2\tau}(\mathbf{x}_0)) + (\mathbf{u}_n(\mathbf{p}_n^{2\tau}(\mathbf{x}_0)) - \mathbf{u}_n(\mathbf{x}_0))2\left(\frac{N}{D} - \frac{1}{2}\right).\end{aligned}$$

- Employing the Lipschitz estimates from [LIP-II, Sec. 5.1] we obtain

$$2\mathbf{u}_n(\mathbf{p}_n^\tau(\mathbf{x}_0)) \leq \mathbf{u}_n(\mathbf{x}_0) + \mathbf{u}_n(\mathbf{p}_n^{2\tau}(\mathbf{x}_0) + C\,\mathrm{Lip}_n\,\tau(\mathbf{g}_n)\left(\frac{N}{D} - \frac{1}{2}\right).$$

In our work we employ a concrete upper bound on the graph function [LIP-II, Lem. 5.5] which tells us that for any $\mathbf{x}, \mathbf{y} \in \Omega_n$ we have

$$d_{w_n}(\mathbf{x}, \mathbf{y}) \leq \left(1 + \frac{4\delta_n}{t_0\varepsilon} + \frac{2\phi(\delta_n)}{t_0\varepsilon}\right) d_\Omega(\mathbf{x}, \mathbf{y}) + \tau_\eta\varepsilon$$

where

$$\tau_\eta := \sup_{0<t\leq t_0}\left\{\sigma_\eta\eta(t)^{-1} - t\right\}.$$

Plugging in this concrete error term, we then obtain the estimate

$$2\mathbf{u}_n(\mathbf{p}_n^\tau(\mathbf{x}_0)) \leq \mathbf{u}_n(\mathbf{x}_0) + \mathbf{u}_n(\mathbf{p}_n^{2\tau}(\mathbf{x}_0)) + \mathrm{Lip}_n(\mathbf{g}_n)C\left(\frac{\delta_n\tau}{\varepsilon} + \varepsilon + \tau\frac{\phi(\varepsilon)}{\varepsilon}\right).$$

Rearranging this inequality then gives the first statement in [LIP-II, Thm. 5.13]. The second one can be proven analogously. Quantitatively, we remark that the term

$$\frac{N}{D} - \frac{1}{2} = \frac{\sup_{\mathbf{y}\in\overline{B}(\mathbf{x}_0,\tau)\cap\Omega_n} d_{w_n}(\mathbf{x}_0, \mathbf{y})}{\inf_{\mathbf{y}\in\Omega_n\setminus\overline{B}(\mathbf{x}_0,2\tau-\varepsilon)} d_{w_n}(\mathbf{x}_0, \mathbf{y})} - \frac{1}{2} \tag{3.36}$$

determines the leading term in the estimate and therefore the rate. This coined the phrase:

"Rates for graph distance functions imply rates for graph AMLEs".

Meeting in the Middle The previous paragraphs allow us to relate the continuum and the discrete solution to the non-local operator. The question is now, how we can put these estimates together. Here, we employ the maximum principle for the operator Δ_∞^τ, from [Sma10, Thm. 2.6.5]. Namely, if for a constant $C \geq 0$ two functions $w, v : \Omega_{\mathcal{O}}^\tau \to \mathbb{R}$ satisfy

$$-\Delta_\infty^\tau w \leq C \leq -\Delta_\infty^\tau v \qquad \text{in } V \subset \Omega_{\mathcal{O}}^\tau, \tag{3.37}$$

then it holds

$$\sup_{\Omega_{\mathcal{O}}^{\tau}}(w-v) = \sup_{\Omega_{\mathcal{O}}^{\tau}\setminus V}(w-v).$$

In our case the function w is chosen as u_n^τ, $V = \Omega_{\mathcal{O}}^{2\tilde{\tau}_n}$ with $\tilde{\tau}_n := \tau + \frac{3}{2}\delta_n$ and

$$\operatorname{Lip}_n(\mathbf{g}_n) C \left| \frac{\delta_n}{\varepsilon\tau} + \frac{\varepsilon}{\tau^2} + \frac{\phi(\varepsilon)}{\varepsilon\tau} \right| =: C_{n,\tau},$$

and therefore the left inequality in (3.37) holds true. We now would like to chose $v = T_\tau u$ however, we know that $-\Delta_\infty^\tau T_\tau u \geq 0$. In order to obtain the desired inequality, we employ a perturbation trick. First we see that [Sma10, Lem. 2.6.4] tells us that for a bounded function $v : \Omega_{\mathcal{O}}^\tau \to \mathbb{R}$ with $-\Delta_\infty^\tau v \geq 0$ on $\Omega_{\mathcal{O}}^{2\tau}$, there exists $\delta_0 > 0$ such that for any $0 \leq \delta \leq \delta_0$ the function $w := v - \delta v^2$ satisfies

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

In the case that $|\nabla u|$ is bounded away from 0 we can choose $w := T_\tau u - \frac{C_{n,\tau}}{\alpha^2}(T_\tau u)^2$ with $\alpha = \inf_{\overline{\Omega}} |\nabla u| > 0$, which yields

$$-\Delta_\infty^\tau w \geq -\Delta_\infty^\tau T_\tau u + \frac{C_{n,\tau}}{\alpha^2}(S_\tau^- T_\tau u)^2 \qquad \text{in } \Omega_{\mathcal{O}}^{2\tau}.$$

Since u is lower semi-continuous we can choose y_0 with $d_\Omega(x, y_0)$ such that

$$\begin{aligned} S_\tau^- T_\tau u(x) &= \frac{1}{\tau}\left(u(y_0) - T_\tau T_\tau u(x)\right) = \frac{1}{\tau}\left(u(y_0) - T_{2\tau} u(x)\right) \\ &\geq \frac{1}{\tau}\left(u(y_0) - (u(y_0) - \tau \inf_{\overline{\Omega}} |\nabla u|)\right) \\ &= \alpha, \end{aligned}$$

and therefore we have $-\Delta_\infty^\tau w \geq C_{n,\tau}$. Furthermore, we see that $\|w - T_\tau u\|_\infty \leq \frac{c}{\alpha^2} C_{n,\tau}$ for some $c > 0$, which allows us to make the following estimate

$$\begin{aligned} \sup_{\Omega_{\mathcal{O}}^{\tilde{\tau}_n}}(u_n^\tau - T_\tau u) &\leq \sup_{\Omega_{\mathcal{O}}^{\tilde{\tau}_n}}(u_n^\tau - w) + \frac{c}{\alpha^2} C_{n,\tau} \\ &\leq \sup_{\Omega_{\mathcal{O}}^{\tilde{\tau}_n}\setminus\Omega_{\mathcal{O}}^{2\tilde{\tau}_n}}(u_n^\tau - w) + \frac{c}{\alpha^2} C_{n,\tau} \\ &\leq \sup_{\Omega_{\mathcal{O}}^{\tilde{\tau}_n}\setminus\Omega_{\mathcal{O}}^{2\tilde{\tau}_n}}(u_n^\tau - T_\tau u) + 2\frac{c}{\alpha^2} C_{n,\tau}. \end{aligned}$$

In the case that $\alpha = 0$ we need to apply the additional perturbation trick from [Sma10, Lem. 2.6.3], which traditionally introduces a root of order 3. In total, we then have the following result. Here we only include the results for the sub-solutions, the respective other inequality holds analogously and can be found in [LIP-II, Prop. 5.16].

Proposition 3.2 ([LIP-II, Prop. 5.16]). Under the previous assumptions there exists a constant $C > 0$ such that

$$\sup_{\Omega_{\mathcal{O}}^{\tilde{\tau}_n}}(u_n^\tau - T_\tau u) \leq \sup_{\Omega_{\mathcal{O}}^{\tilde{\tau}_n}\setminus\Omega_{\mathcal{O}}^{2\tilde{\tau}_n}}(u_n^\tau - T_\tau u) + \sqrt[3]{\operatorname{Lip}_n(\mathbf{g}_n) C\left(\frac{\delta_n}{\varepsilon\tau} + \frac{\varepsilon}{\tau^2} + \frac{\phi(\varepsilon)}{\varepsilon\tau}\right)}.$$

If additionally it holds that $\alpha := \inf_{\overline{\Omega}}|\nabla u| > 0$, then we even have

$$\sup_{\Omega_{\mathcal{O}}^{\tilde{\tau}_n}}(u_n^\tau - T_\tau u) \leq \sup_{\Omega_{\mathcal{O}}^{\tilde{\tau}_n}\setminus\Omega_{\mathcal{O}}^{2\tilde{\tau}_n}}(u_n^\tau - T_\tau u) + \operatorname{Lip}_n(\mathbf{g}_n) C\left(\frac{\delta_n}{\varepsilon\tau} + \frac{\varepsilon}{\tau^2} + \frac{\phi(\varepsilon)}{\varepsilon\tau}\right).$$

Now we can employ the Lipschitz estimates from [LIP-II, Sec. 5.1] to extend the above result to the whole domain $\Omega_{\mathcal{O}}$, which then finally yields [LIP-II, Thm. 2.2].

3.4.3. Numerical Examples and Extensions

Here, we briefly comment on the numerical examples conducted in [LIP-II]. In order to evaluate the convergence rates numerically, we employ a known infinity harmonic function on $\mathbb{R}^2$, namely the Aronsson function

$$u(x_1, x_2) = |x_1|^{4/3} - |x_2|^{4/3}.$$

We choose the constraint set $\mathcal{O} = \{(\pm 1, 0), (0, \pm 1), \}$ with the boundary values given by u. In order to ensure that u is the AMLE of its boundary values, we choose the domain Ω such that the Neumann boundary condition on $\partial\Omega$ is fulfilled. In [LIP-II] we therefore introduce the Aronsson star

$$\Omega := \left\{x \in [0,1]^2 \,:\, |x_1|^{2/3} + |x_2|^{2/3} \leq 1\right\},$$

which is non-convex, non-smooth but fulfills Eq. (3.20). In the discrete problems we consider all combinations of the following configurations,

The code for all the experiments is available at github.com/jwcalder/LipschitzLearningRates.

- exact boundary vertices $\mathcal{O}_n = \mathcal{O}$ and dilated boundary conditions with $\mathcal{O}_n^{\mathrm{dil}} = \{\mathbf{x} \in \Omega_n \,:\, \operatorname{dist}(\mathbf{x}, \mathcal{O}) \leq \varepsilon_n\}$,
- unit weights and singular weights,
- scalings $\varepsilon_n \sim \{\delta_n, \delta_n^{2/3}, \delta_n^{1/2}\}$.

The results are displayed in [LIP-II, Fig. 1–Fig. 4], where we want to highlight the following observations:

(i) The rates are generally better than the ones proven in [LIP-II, Thm. 2.2].

(ii) Singular weights give better rates than unit weights.

(iii) The best rate is achieved for the scaling $\varepsilon_n \sim \delta_n$, for which our results do not even proof convergence.

Concerning (i) we note that this does not directly imply that our analysis is not sharp. Typically, rates for the Aronsson function are better than the ones one can proof in general, see, e.g., [Sma10]. Regarding point (iii) we want to mention the follow-up work [LIP-III], which deals with the case $\varepsilon_n \sim \delta_n$. Working in the setting of first passage percolation, we are able to show a rate for the quantity,

$$\left| \frac{\mathbb{E}[d_{\varepsilon_s,\mathcal{X}_s}(0, se_1)]}{\mathbb{E}[d_{\varepsilon_s,\mathcal{X}_s}(0, 2se_1)]} - \frac{1}{2} \right|,$$

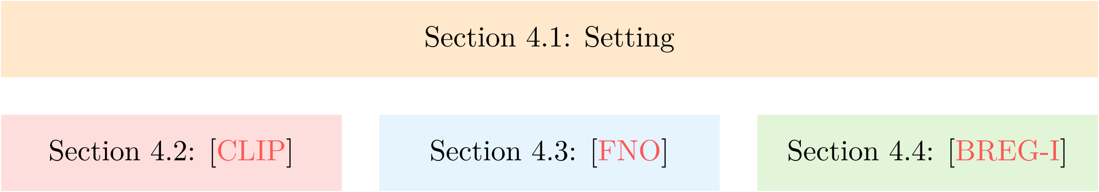
see [LIP-III, Thm. 2.1]. Here, $\mathcal{X}_s$ denote modified Poisson point processes, over which the expectation is taken. The distance $d_{\varepsilon_s,\mathcal{X}_s}(0, se_1)$ is basically a graph distance with unit weights, connecting 0 and the point se_1. We observe that this term is reminiscent of the dominating term Eq. (3.36) we have in [LIP-III]. In fact, employing the same strategy as displayed in the previous paragraphs, in [LIP-III] we are able to show a rate in the case $\varepsilon_n \sim \delta_n$.

Chapter 4

Robust and Sparse Supervised Learning

In this chapter we present the topics discussed in the works [CLIP; FNO; BREG-I] which are reprinted in Part II. Compared to the previous chapter, we are now in the setting of supervised learning, as described in Section 2.2. As already mentioned before, we focus on neural networks $f_\theta : \mathcal{X} \to \mathcal{Y}$ parameterized by $\theta \in \Theta$, where Θ is some parameter space. Here, our two main question arise, namely:

- **Input robustness**: how robust is $x \mapsto f_\theta(x)$ w.r.t. input perturbations?
- **Parameter sparsity**: how can we obtain sparse parameters $\theta \in \Theta$?


Section 4.1: Setting

Section 4.2: [CLIP]

Section 4.3: [FNO]

Section 4.4: [BREG-I]

Therefore, conceptually we again highlight the keyword **sparsity** and additionally **robustness**. In [CLIP] we consider robustness under adversarial perturbations. Here, the input is *attacked* on purpose to confuse a neural network classifier and therefore worsen its performance. In order to obtain a classifier that is less vulnerable against such attacks, we propose an optimization strategy that selects parameters yielding a more robust network. In Section 4.2 we comment on the topic and the contribution in more detail.

A different kind of input robustness is considered in [FNO]. In the setting of image classification, images are modeled as functions on a continuum domain that need to be discretized in order to represent them on a machine. However, this discretization is usually arbitrary and not inherent to the object of interest. Therefore, it is natural to assume that the output of the network should be independent of this discretization, also referred to as resolution, hence we consider robustness w.r.t. resolution changes. We remark on this and the publication in Section 4.3.

Concerning computational performance and memory storage of the neural network, we focus on sparsity of the parameters $\theta \in \Theta$. In [BREG-I] we propose a sparse optimization strategy based on Bregman iterations, which employs L^1 type penalties to promote sparsity. The conceptual difference between the existing algorithm and our proposal is the stochastic component in the gradient. Our theoretical results prove decay in loss and convergence of the iterates. Finally, we also conduct numerical experiments that demonstrate the efficiency of the method. We refer to Section 4.4 for a detailed explanation.

Before we start with the exposition on the mentioned works, we briefly review the common supervised learning framework. Building upon the basic observations in Chapter 4 we give a slightly more detailed introduction in Section 4.1.

4.1. Setting

We are given a finite training set $\mathcal{T} \subset \mathcal{X} \times \mathcal{Y}$. For a family of functions $f_\theta : \mathcal{X} \to \mathcal{Y}$ parameterized by $\theta \in \Theta$, we consider the empirical minimization

$$\min_{\theta \in \theta} \mathcal{L}(\theta),$$

where for a function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ we define

$$\mathcal{L}(\theta) := \frac{1}{|\mathcal{T}|} \sum_{(x,y) \in \mathcal{T}} \ell(f_\theta(x), y). \tag{4.1}$$

Remark 4.1. Assuming that $\mathcal{T}$ is sampled from a joint distribution $P_{\mathcal{X},\mathcal{Y}}$ on $\mathcal{X} \times \mathcal{Y}$ this approximates the computationally infeasible population risk minimization problem

$$\inf_{\theta \in \Theta} \int_{\mathcal{X} \times \mathcal{Y}} \ell(f_\theta(x), y) dP_{\mathcal{X},\mathcal{Y}}(x, y).$$

$\triangle$

In the following we provide two important choices for the function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$.

Example 4.2 (MSE). For image denoising problems, we often choose $\mathcal{X} = \mathcal{Y} = [0, 1]^{N \times M}$, assuming only one color channel for simplicity. In this context the mean squared L^2 error (MSE), defined as

$$\ell(\bar{y}, y) := \frac{1}{N \cdot M} \|\bar{y} - y\|^2$$

is commonly employed. This loss function is also used for classification problems.

Example 4.3 (Cross-Entropy). For classification problems, one often chooses the *cross-entropy* or *negative log-likelihood* loss, [Goo52]. For two discrete probability distributions, $p, q : \{1, \dots, C\} \to \mathbb{R}$ one defines

$$H(p, q) := -\sum_{c=1}^{C} p_c \cdot \log(q_c)$$

see, e.g., [COR98]. Assuming that $\mathcal{Y} = \Delta^C$ this allows to choose $\ell(\bar{y}, y) := H(y, \bar{y})$. If the network only maps to $\mathbb{R}^C$ one often additionally inserts a soft-max function $Q(y)_c := \exp(y_c)/\sum_c \exp(y_c)$ (see [Bol68]) and then sets

$$\ell(\bar{y}, y) := H(y, Q(\bar{y})).$$

In the case, where the output is given as labels $y \in \{1, \dots, C\}$ one defines

$$\ell(\bar{y}, y) := H(y_{\text{oh}}, Q(\bar{y})) = -\log(Q(\bar{y}))$$

employing the one-hot notation from Chapter 2.

4.1.1. Network Architectures

In this thesis we focus on feed-forward neural networks, i.e., we consider layers of the form

$$\Phi(w, W, b)(z) := wz + \sigma(Wz + b) \tag{4.2}$$

where $w \in \mathbb{R}$ models a residual connection, $W \in \mathbb{R}^{n \times n}$ is a weight matrix, $b \in \mathbb{R}^n$ a bias vector and $z \in \mathbb{R}^n$ is the input. We consider a concatenation of $L \in \mathbb{N}$ of such layers, which then forms a neural network

$$f_\theta = \Phi_L \circ \ldots \circ \Phi_1$$

with parameters $\theta = ((W_1, b_1, w_1) \dots, (W_L, b_L, w_L)) \in \Theta$ and layers $\Phi_i := \Phi(w_i, W_i, b_i)$. If there is no residual part, i.e., $w = 0$, then we also allow dimensional changes in each layer, i.e., $z \in \mathbb{R}^n, W \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$.

MLP In the easiest case we consider a perceptron [Ros58], which models a fully connected layer, i.e., every entry W_{ij} of the weight matrix is a parameter that is optimized in the training process.

Convolutions Especially important for visual tasks are convolutional layers. Here, we take a kernel $k \in \mathbb{R}^{M \times M}$ and define the application of $W = W(k)$ as

$$Wz = k * z$$

with $*$ denoting the convolution. We refer to Section 4.3 for the concrete definition of this operation. Typically, the input is of the form $z \in \mathbb{R}^{K_{\text{in}} \times N \times M}$, where K_{in} denotes the number of input channels. The layer is then a mapping $\Phi : \mathbb{R}^{K_{\text{in}} \times N \times M} \to \mathbb{R}^{K_{\text{out}} \times N \times M}$, where K_{out} denotes the number of output channels, which can be realized by different kernels k_{ij}. The application of W to an input z is defined as

$$(Wz)_{j,:,:} := \sum_{i=1}^{K_{\text{in}}} k_{ij} * z.$$

ResNets Often we also consider a residual component as displayed in Eq. (4.2) with the term $w\,z$. The idea of adding this residual component was first introduced in [SGS15] with a learnable parameter $w \in \mathbb{R}$ and later popularized in [He+16a] by fixing $w = 1$, see also [He+16b], which then yields the celebrated ResNet architecture. In the following applications, we both consider the case where $w = 1$ is fixed, but also the possibility of learning the parameter $w \in \mathbb{R}$ in [BREG-II].

4.1.2. Gradient Computation and Stochastic Gradient Descent

Training a neural network requires solving an optimization problem w.r.t. to the parameters $\theta \in \Theta$. In this work we only focus on first order methods, however both zero [Rie23; Pin+17; Car+21] and second order methods [Mar10] have been successfully applied in this context. Employing first order methods, requires evaluating the gradient $\nabla_\theta \mathcal{L}$, however in this scenario it is not common to compute the full gradient but rather to have a gradient estimator. This estimator is usually obtained by randomly dividing the train set $\mathcal{T}$ into disjoint mini-batches $B_1 \cup \ldots \cup B_b = \mathcal{T}$ and then successively computing the gradient of the mini-batch loss,

$$\mathcal{L}(\theta; B) := \frac{1}{|B_i|} \sum_{(x,y) \in B_i} \ell(f_\theta(x), y).$$

Iterating over all batches $i = 1, \ldots, b$ is referred to as one epoch. From a mathematical point of view, this yields stochastic optimization methods, since in each step the true gradient is replaced by an estimator. In the abstract setting we let $(\Omega, F, \mathbb{P})$ be a probability space and consider a function $g : \Theta \times \Omega \to \Theta$ as an unbiased estimator of $\nabla \mathcal{L}$, i.e.,

$$\mathbb{E}\left[g(\theta; \omega)\right] = \nabla \mathcal{L}(\theta) \text{ for all } \theta \in \Theta.$$

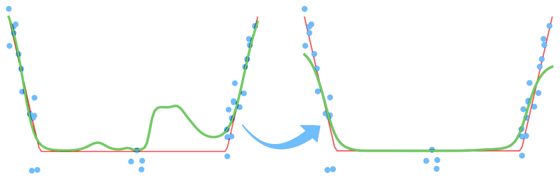
Most notably this method transforms the standard gradient descent update [Cau+47]

$$\theta^{(k+1)} = \theta^{(k)} - \tau^{(k)} \nabla \mathcal{L}(\theta^{(k)}),$$

to *stochastic* gradient descent [RM51]

$$\begin{aligned} &\text{draw } \omega^{(k)} \text{ from } \Omega \text{ using the law of } \mathbb{P}, \\ g^{(k)} &:= g(\theta^{(k)}; \omega^{(k)}), \\ \theta^{(k+1)} &:= \theta^{(k)} - \tau^{(k)} g^{(k)}. \end{aligned}$$

4.2. Adversarial Stability via Lipschitz Training: [CLIP]



In the following we consider classification problems with $C \in \mathbb{N}$ different classes and functions $f : \mathcal{X} \to \Delta^C$ for which we denote by

$$f^{\mathrm{MAP}}(x) := \mathrm{argmax}_c\, f(x)_c$$

the maximum a posteriori estimation for an input $x \in \mathcal{X}$. The capabilities of neural networks to perform classification tasks on unseen data—i.e., inputs $x \in \mathcal{X}$ that are not part of the training data—are impressive and the reason for their popularity. However, it has been noticed in [GSS14] that it is relatively easy to "fool" classification networks in the following sense:

> *Given a classification network and a human classifier $f_\theta, h : \mathcal{X} \to \Delta^C$, and an input $x \in \mathcal{X}$ such that $f_\theta^{\mathrm{MAP}}(x) = h^{\mathrm{MAP}}(x)$. An adversarial example is an input $\bar{x} \in \mathcal{X}$ that is close to x, but we have that*
>
> $$f_\theta^{\mathrm{MAP}}(\bar{x}) \neq h^{\mathrm{MAP}}(\bar{x}) = h^{\mathrm{MAP}}(x).$$

The vague concept of a *human classifier*, incorporates the intuition of the classification problem as a human would solve it. Assuming that we are given data $\mathcal{T} \subset \mathcal{X} \times \mathcal{Y}$ that is sampled i.i.d. from a joint distribution $P_{\mathcal{X},\mathcal{Y}}$, this function could also be chosen as $h(x)_c := P_{\mathcal{X},\mathcal{Y}}(c|x)$. The term "close" describes that the difference between the two images $x, \bar{x}$ should be visually imperceptible.

Remark 4.4. Some authors argue that such instabilities are inherent to classification problems solved via a data-driven approach [Sha+19a; FFF18]. From this point of view, trying to defend against these instabilities is not necessarily desirable. However, we do not consider this interpretation in the following. △

What are Adversarial Examples Formally? In order to formalize this idea, we omit any influence of the typically unavailable function h and instead consider $\bar{x}$ adversarial if $f_\theta^{\mathrm{MAP}}(x) \neq f_\theta^{\mathrm{MAP}}(\bar{x})$. This interpretation only makes sense, if we assume that the output of $f_\theta^{\mathrm{MAP}}(x)$ is *correct*, which can only be easily checked on labeled data [Bun+23]. Furthermore, in this work we assume that $\mathcal{X}$ models an image space $\mathcal{X} = \mathbb{R}^{K \times N \times M}$, with $K, N, M \in \mathbb{N}$ and choose a distance $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_0^+$ to measure the distance between points in $\mathcal{X}$.

Remark 4.5. In most of our examples, we choose $d(\cdot,\cdot) = \|\cdot - \cdot\|_p$. Employing a L^p norm in an image context—via pixel-wise comparisons—can be an unfavorable choice. Images that appear very different visually can have a smaller L^p distance than images that visually appear similar, see, e.g., [Sta+21, Fig. 16]. However, it is easy to evaluate, which is crucial for most applications. In the absence of a better criterion, being close to the original input in the L^p distance is a practical way to decide if $\bar{x}$ is an admissible adversarial example. △

Employing these simplifications, we then say that $\bar{x} \in \mathcal{X}$ is adversarial, if

$$d(x, \bar{x}) \leq \varepsilon \qquad \text{and} \qquad f_\theta^{\mathrm{MAP}}(x) \neq f_\theta^{\mathrm{MAP}}(\bar{x}),$$

assuming that $f_\theta^{\mathrm{MAP}}(x)$ is indeed correct. The parameter ε is called the *adversarial budget* and controls how far $\tilde{x}$ can be from the original point x to be still considered adversarial. Here, we also have to decide how much we trust the metric $d(\cdot, \cdot)$ as a measure of closeness.

Types of Adversarial Examples Typically, adversarial examples are created from the clean image x via some distortion $\delta \in \mathbb{R}^s$. Together with an application map $T : \mathcal{X} \times \mathbb{R}^s \to \mathcal{X}$ one then obtains $\bar{x} = T(x, \delta)$ as the adversarial example. In this formulation, one can alternatively employ the criterion $\|\delta\| \leq \varepsilon$ to decide, whether $T(x, \delta)$ is an adversarial example. We only list some approaches below:

- **Addition**: The most well-known examples are created by $T(x, \delta) := x + \delta$, i.e. $\bar{x} = x + \delta$. Here, it is important to note that typically images are assumed to have values between 0 and 1, i.e., $\mathcal{X} = [0, 1]^{K \times N \times M}$, therefore it is important to also ensure that $\bar{x} \in \mathcal{X}$.

- **Translation and Rotation**: Simple geometric transformations—that are unnoticeable for a human classifier—are quite effective to "fool" neural networks. Employing translations $T_{\mathrm{t}} : \mathcal{X} \times \mathcal{X} \to \mathcal{X}$, one has to choose the behavior on the boundary such that one obtains a valid image. The same holds true for rotations $T_{\mathrm{r}} : \mathcal{X} \times [-\pi, \pi] \to \mathcal{X}$. In this case, $\delta \in [-\pi, \pi]$ models the angle of the rotation and yields an admissible adversarial example, if $|\delta| \leq \varepsilon$. Here, we see that this formulation looses some expressivity. Consider the MNIST dataset [LC10], where the task is to classify handwritten digits from 0 to 9:

  - We see that only $\varepsilon < \pi/2$ makes sense, otherwise the number "6" could be always transformed to the number "9" and vice versa.
  - However, considering the number "0", rotations above the angle $\pi/2$ can definitely yield proper adversarial examples $\bar{x}$.

  We refer to [Eng+18] for a study on these types of adversarial examples.

- **Change of Basis**: As explored in [Guo+18] one can consider a different orthonormal basis of the image space $\mathbb{R}^{K \times N \times M}$ and then perform the attack w.r.t. this basis. The map $T : \mathcal{X} \times \mathbb{R}^{K \times N \times M} \to \mathcal{X}$ first obtains the different representation of x, then adds the coefficients of δ and then maps back to the original basis. This is only meaningful, if one restricts certain coefficients of δ to be zero in the alternative basis. For example, the discrete cosine transform ([ANR74]) has been applied successfully in this context [Guo+18].

In the following, we restrict ourselves to the additive case and only consider examples $\bar{x} = x + \delta$.

Finding Adversarial Examples Employing a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ the task of finding adversarial examples $\bar{x} = x + \delta \in \mathcal{X}$ can be relaxed via solving the problem

$$\max_{\bar{x} \in \mathcal{X}: d(x,\bar{x}) \leq \varepsilon} \ell(y, f_\theta(\bar{x})) \tag{4.3}$$

where $y = f_\theta(x)$. Solving this problem is referred to as *attacking* the network f. More precisely, this is a so-called *untargeted* attack, since we do not prescribe $\bar{x}$ to produce any special output as long as it confuses the network. Opposed to this, there are so called *targeted* attacks. Here, we pick $c^{\text{adv}} \neq f_\theta^{\text{MAP}}(x)$ and then want to find $\bar{x}$ such that $f_\theta^{\text{MAP}}(\bar{x}) = c^{\text{adv}}$, which yields the problem

$$\min_{\bar{x} \in \mathcal{X}: d(x,\bar{x}) \leq \varepsilon} \ell(c_{\text{oh}}^{\text{adv}}, f_\theta(\bar{x})).$$

Conceptually, this problem is similar to Eq. (4.3) and differs only by a change of sign and a different reference label. A popular method to solve Eq. (4.3) is the projected sign gradient ascent iteration [KGB+16],

$$\bar{x}^{(k+1)} = \text{Proj}_d(\bar{x}^{(k)} + \tau \, \text{sign}(\nabla_{\bar{x}} \ell(f_\theta(x), f_\theta(\bar{x}^{(k)})))).$$

Here, Proj_d denotes the projection onto the set $\mathcal{X} \cap B_{d,\varepsilon}(x)$, where $B_{d,\varepsilon}(x)$ is the ball with radius ε around x, w.r.t. to the distance d. Performing only one step of this iteration yields the fast gradient sign method [GSS14]. There is a wide variety of these so-called gradient-based open-box attacks [Yua+19], i.e., methods that assume that the gradient of the model is available. In more realistic scenarios, this might not be the case. Attacks that do not employ the actual gradient of the model to attack are called *closed box* attacks [Ily+18], which are not part of this thesis.

Defending against Adversarial Attacks We consider the question of finding parameters $\theta \in \Theta$ such that the corresponding model f_θ is *adversarially robust*, i.e., is less vulnerable to attacks. Therefore, we want the attack problem in Eq. (4.3) to be hard to solve. This intuition leads to the optimization problem

$$\min_{\theta \in \Theta} \sum_{(x,y) \in \mathcal{T}} \max_{\bar{x}: d(x,\bar{x}) \leq \varepsilon} \ell(f_\theta(\bar{x}), y),$$

which is known as the *adversarial training* formulation [KGB17; Mad+18]. From a distributionally robust optimization point of view, this problem relates to

$$\min_{\theta \in \Theta} \max_{\tilde{P}: D(P_{\mathcal{X},\mathcal{Y}}, \tilde{P}) \leq \varepsilon} \int_{\mathcal{X} \times \mathcal{Y}} \ell(f_\theta(x), y) \; d\tilde{P}(x,y)$$

where D denotes some distance on the space of probability distributions, see [BGM23]. Employing a batched gradient descent type iteration for the variable θ, one then has to solve a problem of the form Equation (4.3) in every step, which can again be approximated via gradient ascent on the input variable x.

Lipschitz Training of Neural Networks Adversarial robustness is closely related to the Lipschitzness of a network f_θ. Namely, for inputs $x, \bar{x} \in \mathcal{X}$ that are close, we also want the outputs of f_θ to be close. In other words, we want to find a small constant $L > 0$ such that

$$\|f_\theta(x) - f_\theta(\bar{x})\|_{\mathcal{Y}} \leq L \ \|x - \bar{x}\|_{\mathcal{X}},$$

where we typically use an L^p norm for both $\|\cdot\|_{\mathcal{X}}$ and $\|\cdot\|_{\mathcal{Y}}$. The smallest constant fulfilling this inequality is the Lipschitz constant $\operatorname{Lip}(f_\theta)$. If this constant is small, one can expect that small input perturbations do not affect the classification output significantly.

Remark 4.6. Apart from adversarial robustness, having an upper bound on the Lipschitz constant of a neural network is also important for other applications, see, e.g., [Arn+23; Has+20; ACB17]. △

In our work, we employ the Lipschitz constant as a regularizer and consider the problem

$$\min_\theta \mathcal{L}(\theta) + \lambda \ \operatorname{Lip}(f_\theta) \tag{4.4}$$

for a parameter $\lambda > 0$. Computing the Lipschitz constant of neural networks is an NP-hard problem [SV18], therefore many works rely on the estimate

$$\operatorname{Lip}(f_\theta) \ \leq \ \prod_{l=1}^{L} \operatorname{Lip}(\Phi_l) \leq C_\sigma \prod_{l=1}^{L} \|W_l\|, \tag{4.5}$$

where here $\|\cdot\|$ denotes some matrix norm and $C_\sigma > 0$ depends on the Lipschitz constants of the activation functions, see [ALG19; Gou+20; Kri+20; RKH19]. This inequality is not sharp and usually overestimates the Lipschitz constant, as we see in the following example taken from [CLIP].

Example 4.7. We consider a feed-forward neural network $f_\theta : \mathbb{R} \to \mathbb{R}$ with one hidden layer,

$$\begin{aligned}
\Phi_1(z) &:= \operatorname{ReLU}(W_1 z) := (\max\{z, 0\}, \max\{-z, 0\})^T, \qquad W_1 := (1, -1),\\
\Phi_2(z) &:= W_2 z := z_1 + z_2, \qquad W_2 := (1, 1)^T,\\
f_\theta &:= \Phi_2 \circ \Phi_1.
\end{aligned}$$

For $x \in \mathbb{R}$ we have that

$$\begin{aligned}
x \geq 0 \qquad &\Rightarrow \Phi_1(x) = (x, 0)^T \qquad \Rightarrow f_\theta(x) = x,\\
x \leq 0 \qquad &\Rightarrow \Phi_1(x) = (0, -x)^T \qquad \Rightarrow f_\theta(x) = -x,
\end{aligned}$$

and therefore $f_\theta = |\cdot|$, which yields that $\operatorname{Lip}(f_\theta) = 1$. However, employing the spectral norm for the weight matrices, we see that

$$\|W_1\| \cdot \|W_2\| = \sqrt{2}\ \sqrt{2} = 2.$$

Plugging the estimate of Eq. (4.5) into Eq. (4.4) therefore potentially over regularizes the problem. While this increases the stability of the network, it can worsen its classification performance.

Contribution in [CLIP] In [CLIP] we propose a strategy to solve Eq. (4.4) approximately, without the estimate in Eq. (4.5). The basic idea consists of approximating the Lipschitz constant on a finite set and using this approximation as a regularizer. Furthermore, we analyze the original model in Eq. (4.4) where we study existence of solutions, the influence of the parameter λ and the limits $\lambda \to 0, \lambda \to \infty$, see Section 4.2.2. Finally, we demonstrate the efficiency of the method by applying it to some simple toy problems, see Section 4.2.3.

4.2.1. Cheap Lipschitz Training

We approximate the Lipschitz constant on a finite set $\mathcal{X}_{\text{Lip}} \subset \mathcal{X} \times \mathcal{X}$ via

$$\operatorname{Lip}(f_\theta; \mathcal{X}_{\text{Lip}}) := \max_{(x,\bar{x})\in\mathcal{X}_{\text{Lip}}} \frac{\|f_\theta(x) - f_\theta(\bar{x})\|_{\mathcal{Y}}}{\|x-\bar{x}\|_{\mathcal{X}}} \approx \operatorname{Lip}(f_\theta).$$

Disregarding the non-differentiable points of $\theta \mapsto \operatorname{Lip}(f_\theta; \mathcal{X}_{\text{Lip}})$ and employing the above approximation, allows us to solve the problem in Eq. (4.4) via stochastic gradient descent on the variable θ.

Remark 4.8. Let $f, g : \mathcal{X} \to \mathcal{Y}$ be two differentiable functions. If $f(\bar{x}) > g(\bar{x})$ for some $\bar{x} \in \mathcal{X}$, then we know that there exists some $\epsilon > 0$ such that $f(x) > g(x)$ for all $x \in B_\epsilon(\bar{x})$. We have that $f \vee g = \max\{f, g\} = f$ in $B_\epsilon(\bar{x})$ and therefore $x \mapsto f(x) \vee g(x)$ is differentiable in $\bar{x}$. If $f(\bar{x}) = g(\bar{x})$ then $f \vee g$ could be non-differentiable at $\bar{x}$. However, in practice we employ automatic differentiation, where in this case one of the functions is chosen, say f, and the *gradient* at $\bar{x}$ is computed as ∇f. △

The strength of this approach is dependent on the quality of the set $\mathcal{X}_{\text{Lip}}$. In [CLIP] we propose to iteratively update the set via a gradient ascent type scheme. Namely, we initialize $\mathcal{X}_{\text{Lip}}$ as a random perturbation of a subset of the given data. In each step, we then update the points as follows:

- Consider $L(x, \bar{x}) := \|f_\theta(x) - f_\theta(\bar{x})\| \,/\, \|x - \bar{x}\|, \quad x, \bar{x} \in \mathcal{X}$.
- For each $(x, \bar{x}) \in \mathcal{X}_{\text{Lip}}$ perform the update

$$\begin{aligned} x &\leftarrow x + \tau\, L(x,\bar{x}) \nabla_x L(x,\bar{x}),\\ \bar{x} &\leftarrow \bar{x} + \tau\, L(x,\bar{x}) \nabla_{\bar{x}} L(x,\bar{x}), \end{aligned}$$

  for a parameter $\tau > 0$.

This scheme performs gradient ascent on the Lipschitz constant, see [CLIP, Alg. 1]. For each mini-batch $B \subset \mathcal{T}$ we first update the set $\mathcal{X}_{\text{Lip}}$ and then update the parameters via

$$\theta \leftarrow \theta - \eta\ \nabla_\theta \left(\mathcal{L}(\theta; B) + \lambda\ \operatorname{Lip}(f_\theta, \mathcal{X}_{\text{Lip}}) \right)$$

which yields the algorithm presented in [CLIP, Alg. 2]. Similar to [Sha+19b] one can reuse the gradients computed for ∇_x for the computation of ∇_θ. This fact yields the attribute *cheap* in the Lipschitz training algorithm.

4.2.2. Analysis of Lipschitz Regularization

In [CLIP] we analyze some basic properties of the original regularization problem in Eq. (4.4). We repeat the assumptions made therein.

Assumption 1. We assume that the loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R} \cup \{\infty\}$ satisfies:

a) $\ell(y, \bar{y}) \geq 0$ for all $y, \bar{y} \in \mathcal{Y}$,

b) $y \mapsto \ell(y, \bar{y})$ is lower semi-continuous for all $\bar{y} \in \mathcal{Y}$.

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

$$\frac{1}{|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} \ell(f_\theta(x), y) + \lambda \operatorname{Lip}(f_\theta) < \infty.$$

Employing only Assumption 2 one can show that the map $\theta \mapsto \operatorname{Lip}(f_\theta)$ is lower semi-continuous, [CLIP, Lem. 1].

Existence If Θ is a compact subset of a reflexive Banach space or finite, one can show that there exist solutions of the problem in Eq. (4.4). In the general case, one needs to add a norm term, that ensures boundedness of a minimizing sequence. The following is the main existence result, which can be proven by the direct method in the calculus of variations [Dac07].

Proposition 4.1 ([CLIP, Prop. 1]). Under Assumptions 1 to 3 the problem

$$\min_{\theta\in\Theta} \frac{1}{|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} \ell(f_\theta(x), y) + \lambda \operatorname{Lip}(f_\theta) + \mu \|\theta\|_\Theta$$

has a solution for all values $\lambda, \mu > 0$. Here, $\|\cdot\|_\Theta$ denotes a norm on Θ.

Dependency on the Regularization Parameter Assuming that for every $\lambda > 0$ we have a solution $\theta_\lambda \in \Theta$ of Eq. (4.4) one can show that

$$\begin{aligned} &\lambda \longmapsto \frac{1}{|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} \ell(f_{\theta_\lambda}(x), y) && \text{is non-decreasing,} \\ &\lambda \longmapsto \operatorname{Lip}(f_{\theta_\lambda}) && \text{is non-increasing.} \end{aligned}$$

This statement is the content of [CLIP, Prop. 2], however, the argument is exactly the same as in [BO13]. The intuition behind this result is that with increasing parameter λ, solutions of Eq. (4.4)—more precisely the corresponding networks—have smaller Lipschitz constants, i.e., tend to be more constant, which however also diminishes their expressivity. We formalize the limit cases in the following.

Assuming the realizability condition of [SB14] we know that there exists parameters $\theta \in \Theta$ such that $\mathcal{L}(\theta) = 0$. This means there are parameters that fit the data perfectly. Considering the limit $\lambda \to 0$ we obtain convergence to a solution of the unregularized problem with the smallest Lipschitz constant.

Proposition 4.2 ([CLIP, Prop. 3]). Let Assumptions 1 to 3 and the realizability assumption [SB14] be satisfied. If $\theta_\lambda \to \theta^\dagger \in \Theta$ as $\lambda \searrow 0$, then

$$\theta^\dagger \in \operatorname{argmin} \left\{ \operatorname{Lip}(f_\theta) \,:\, \theta \in \Theta,\, \frac{1}{|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} \ell(f_\theta(x), y) = 0 \right\}$$

if this problem admits a solution with $\operatorname{Lip}(f_\theta) < \infty$.

Furthermore, we study the effect of sending $\lambda \to \infty$, where we see that the network f_{θ_λ} indeed tends to be constant as expected. We can explicitly characterize this constant as the closest point to barycenter of the output data, that is realizable by a neural network.

Proposition 4.3 ([CLIP, Prop. 4]). Let Assumptions 1 to 3 be satisfied and assume that

$$\mathcal{M} := \{y \in \mathcal{Y} \,:\, \exists \theta \in \Theta,\, f_\theta(x) = y,\, \forall x \in \mathcal{X}\} \neq \emptyset.$$

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

$$\hat{y} \in \operatorname{argmin}_{y' \in \mathcal{M}} \frac{1}{|\mathcal{T}|} \sum_{y \in \mathcal{T}_\mathcal{Y}} \ell(y', y).$$

4.2.3. Numerical Results

We briefly comment on the numerical results [CLIP]. All the experiments were implemented in `Python` [VD95] employing—among others—the `PyTorch` [Pas+19] package. We conduct experiments on the MNIST [LC10] and FashionMNIST [XRV17] datasets.

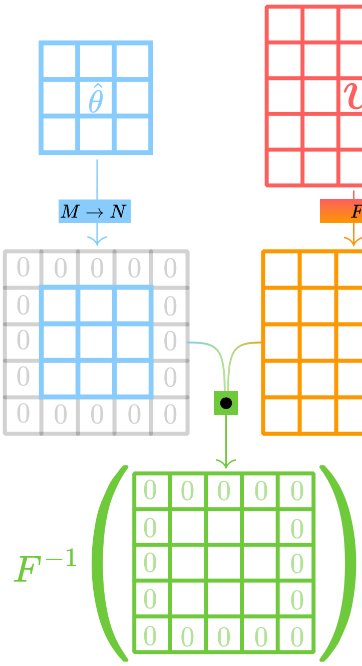
Qualitative Example We first apply the CLIP algorithm to regularize a one-dimensional regression problem. In [CLIP, Fig. 2] one observes the qualitative effect of the regularization. Namely, we are given noisy data from a ground truth function. The solution of the unregularized problem overfits the data and the resulting network has fluctuations and therefore a large Lipschitz constant in certain regions, where the ground truth function has a small Lipschitz constant. With increasing parameter λ, we observe that the networks f_{θ_λ} are smoothed out and better approximate the ground truth function.

The code for all the experiments is available at github.com/TimRoith/CLIP.

Quantitative Evaluation of Adversarial Robustness We additionally evaluate how robust a CLIP trained network is against adversarial attacks. Next to a standard learning rate scheduler, we also update the regularization parameter λ throughout the training process. Namely, we choose a target accuracy and evaluate the loss on a validation set after each epoch. If this accuracy is less than the target, we decrease λ and increase it if the accuracy is higher. In [CLIP, Tab. 1] we compare ourselves against a weight regularization scheme that is based on the estimate in Eq. (4.5). We see that CLIP outperforms this method in most cases.

4.3. Resolution Stability via FNOs: [FNO]

In this section, we comment on the results in [FNO], concerning input robustness w.r.t. resolution changes. In practice, we frequently employ the set $\mathcal{X} = [0,1]^{K\times N\times M}$ to represent images. However, from a modeling point of view it is more natural to assume that images are functions $u : \Omega \to [0,1]^K$ where $\Omega \subset \mathbb{R}^2$ is some domain. In this sense, the space $\mathcal{X}$ only constitutes a discretization of the space of all functions from Ω to $[0,1]^K$. The number of pixels, i.e., $N \cdot M$, relates to the *resolution* of the image, where a higher number of pixels yields a higher resolution. In this sense, the resolution is merely an artifact of practical restrictions and should not be relevant for the classification. Therefore, one wants to obtain a resolution-independent classifier, which we study in the following.



Images, Resolution, and Scale In the continuum setting, we consider the d-dimensional torus $\Omega = \mathbb{R}^d/\mathbb{Z}^d$. An image is a function

$$u : \Omega \to [0,1]^K$$

where $K \in \mathbb{N}$ denotes the number of color channels, see [GW87]. In order to represent images on a computer, we discretize the domain Ω, via a regular grid $\Omega_N = \{x_0, \dots, x_N\}$ indexed by the set

$$\mathcal{J}_N := \{0, \dots, N-1\}^d$$

with grid points $x_j = j/(N-1)$, see, e.g., [Kab22; KLM21]. For simplicity, we assume an equal number of discretization points in each dimension. Therefore, N^d is the resolution of an image discretized w.r.t. $\mathcal{J}_N$.

It is important to notice the differences to the concept of *scale*. A change in scale would be a change in the original image domain, for example, by zooming in on a certain area. In our scenario, we usually assume that the image u contains a certain entity to be classified. The scale roughly describes how "big" the entity is, or what percentage of the image it fills. We assume that the scale is fixed. Therefore, changing N directly influences the resolution of the discretization.

How do Neural Networks react to Resolution Changes? In many machine learning applications, one assumes a fixed input size and therefore a fixed resolution, assuming the same scale throughout the data. Formally, networks are defined as mappings $f_\theta : \mathbb{R}^{K\times N\times N} \to \Delta^C$, i.e., they only accept inputs of the fixed resolution N. In order to understand how this constraint can be weakened, we need to consider the concrete structure of our networks, which was similarly done in [KLM21; Kab22]. The architectures we consider are feed-forward networks of the form

$$f_\theta = \Phi^{\text{class}} \circ \mathcal{S} \circ \Phi^{\text{feature}}$$

where

- Φ^{feature} is the so-called feature extractor, which should be applicable independently of the input dimension,
- $\mathcal{S}$ is a function that maps inputs of any size to a fixed output dimension $\mathbb{R}^s$,
- $\Phi^{\text{class}} : \mathbb{R}^s \to \Delta^C$ denotes the classification layer.

The feature extractor usually consists of convolutional layers. As in [Kab22, Ch. 2] we consider the discrete convolution of two discretized images u_N, v_N defined as

$$(u_N * v_N)(x_j) := \sum_{k\in\mathcal{J}_N} u_N(x_k)\cdot v_N(x_{j-k}) \tag{4.6}$$

where for negative indices $j-k$ we set $x_{j-k} := x_{j-k+N}$. This assumes that both u_N and v_N live on the same discretization $\mathcal{J}_N$. For neural networks, we are interested in the convolution of an input image u and a kernel $\theta \in \mathbb{R}^{\mathcal{J}_M}$. Modeling spatial locality—and therefore a small support of the kernel—one usually chooses M much smaller than the resolution. This is for example motivated by the study in [HW62] which explores a similar methodology for the visual cortex of cats. However, if $M < N$ one can not directly employ the convolution in Eq. (4.6), since there we assumed the same discretization for both inputs. In order to account for this dimension mismatch, we consider so-called *spatial zero-padding* for kernels $\theta \in \mathbb{R}^{\mathcal{J}_M}$

$$\theta_k^{M\to N} = \begin{cases} \theta_k & \text{for } k \in \mathcal{J}_N \cap \mathcal{J}_M, \\ 0 & \text{for } k \in \mathcal{J}_N \setminus \mathcal{J}_M. \end{cases}$$

Using this method, one can define the convolution for inputs u_N of arbitrary input resolution $N \geq M$ via

$$C(\theta)(u_N) = \theta^{M\to N} * u_N.$$

In [FNO] we refer to this as the *spatial implementation* of convolution. Up to the behavior on the boundary, this is in fact the standard implementation in most libraries, especially in `PyTorch`. Therefore, a feature extractor consisting of convolutional layers can take inputs of variable resolution. In fact, ignoring possible resolution changes within the extractor—via pooling or strided convolutions—we have that $\Phi^{\text{feature}}(u_N) \in \mathbb{R}^{\mathcal{J}}_{N}$ for any $N \in \mathbb{N}$.

The mapping $\mathcal{S}$ can be realized as an adaptive pooling layer, see [Pas+19], which ensures a fixed output size. This methodology yields a *discretization invariant architecture*, see [Kab22; KLM21; Li+21]. This means that from a technical point of view, the network is able to produce outputs for inputs with arbitrary discretization. However, we are actually interested in a *discretization invariant functionality* (see [Kab22; KLM21; Li+21]), which also requires that the output similar for different resolutions.

Input Interpolation The technical possibility to handle different input sizes, as described above, usually does not perform well in practice. This is due to the fact that in the standard spatial implementation of convolution the support of the kernel changes with varying input dimension, hence the output differs, see Fig. 4.1. If a network is trained on a fixed input size, the filters are adapted to this size and therefore only create meaningful responses on this size.

A simple attempt to create a network, such that not only the architecture, but also the functionality is discretization independent—at least up to a certain degree—is input interpolation. In this case, our architecture is modified to

$$\tilde{f}_\theta = f_\theta \circ I$$

where $I : \bigcup_{M\in\mathbb{N}} \mathbb{R}^{\mathcal{J}_M} \to \mathbb{R}^{\mathcal{J}_N}$ is an interpolation function, that maps inputs of arbitrary sizes to a fixed discretization $\mathcal{I}_N$. Typical choices here include nearest neighbor, bilinear or bicubic interpolation, see, e.g., [GW87]. Especially relevant in our case, is so-called *trigonometric interpolation*, where for $v \in \mathbb{R}^{\mathcal{J}_M}$ we define

$$I^{\text{trigo}}(v) := v^{M \xrightarrow{\Delta} N} := F^{-1}\left((Fv)^{M\to N}\right).$$

Contribution in [FNO] We study the connection between FNOs and CNNs for classification problems. We identify under which assumption the architectures are equivalent (see Lemma 4.9), but also where they are not, see Fig. 4.1. Here, we are especially interested in the multi-resolution case. We show that one layer of an FNO is equivariant with respect to trigonometric interpolation, Corollary 4.11. This is also underlined by numerical experiments, where we compare the FNO implementation to interpolation methods and a simple CNN implementation. Furthermore, we show that training equivalent CNN and FNO layers leads to different results, i.e., while they have the same forward pass, the gradients w.r.t. their parameters might differ, Lemma 4.10. Moreover, we show continuity and Fréchet-differentiability of abstract neural layers as operators between L^p spaces, see Section 4.3.2. Finally, we conduct numerical experiments supporting our theoretical findings Section 4.3.3.

4.3.1. Fourier Neural Operators

We want to obtain neural networks whose output does not depend on the discretization of the image $u : \Omega \to \mathbb{R}^K$. This raises the question whether it is possible to find a formulation that allows us to work in the infinite dimensional setting. In [Kov+23] this issue was addressed in the setting of parametric PDEs, where the authors introduced the concept of *neural operators*. One layer of a neural operator is given as a mapping $\mathcal{G} : L^p(\Omega) \to L^q(\Omega)$

$$\mathcal{G}(u)(x) = \sigma\left(\Psi(u)(x)\right) \qquad \text{for a.e. } x \in \Omega, \tag{4.7}$$

with an affine linear part given by

$$\Psi(u) = Wu + \mathcal{K}u + b, \tag{4.8}$$

where

- $\mathcal{K} : u \mapsto \int_\Omega \kappa(\cdot, y)\, u(y)\, dy$ is a kernel integral operator with kernel $\kappa : \Omega \times \Omega \to \mathbb{R}$,
- $W \in \mathbb{R}$ models a residual component,
- and $b : \Omega \to \mathbb{R}$ models a bias.

By a slight abuse of notation, the activation function $\sigma : \mathbb{R} \to \mathbb{R}$ acts as a Nemytskii operator, i.e.,

$$\sigma : v \mapsto \sigma(v(\cdot)), \tag{4.9}$$

see, e.g., [Trö10]. In Section 4.3.2 we analyze continuity and differentiability of a layer in this abstract form. However, the most relevant case for us, is when $\mathcal{K}$ is a convolution operator, i.e., $\kappa(x, y) = \kappa(x - y)$ is a translation invariant kernel. In this special case $\mathcal{G}$ is then known as layer of a *Fourier Neural Operator* (FNO) as introduced in [Li+21]. We can parameterize the kernel via its Fourier coefficients $\hat{\theta}_k \in \mathbb{C}$,

$$\kappa_{\hat{\theta}}(x) = \sum_{k \in \mathcal{I}} \hat{\theta}_k\, b_k(x), \tag{4.10}$$

where $b_k(x) = \exp\left(2\pi i\, kx\right)$ denote the Fourier basis functions. In practice, we assume that $\kappa_{\hat{\theta}}$ only has a finite amount of non-zero Fourier coefficients. We choose the set $\mathcal{I}_N := \{-\lceil (N-1)/2 \rceil, \ldots, 0, \ldots, \lfloor (N-1)/2 \rfloor\}^d$ as the index set, as done in [Li+21]. Employing this set with odd $N \in \mathbb{N}$, we can easily enforce Hermitian symmetry $\hat{\theta}_k = \overline{\hat{\theta}_{-k}}$, which ensures that $\mathcal{K}_{\hat{\theta}}$ outputs real-valued functions. For now, we assume an odd number N and deal with the even case later. We note that this number of Fourier coefficients is completely independent of the spatial input discretization, which is the main advantage of FNOs.

How to Perform Convolutions with FNOs? In [FNO] we consider the discrete Fourier transform and its inverse

$$(Fv)_k = \frac{1}{\lambda} \sum_{j \in \mathcal{J}_N} v_j\, e^{-2\pi i \left\langle k, \frac{j}{N} \right\rangle}, k \in \mathcal{I}_N,$$

$$\left(F^{-1}\hat{v}\right)_j = \frac{\lambda}{|\mathcal{J}_N|} \sum_{k \in \mathcal{I}_N} \hat{v}_k\, e^{2\pi i \left\langle k, \frac{j}{N} \right\rangle} j \in \mathcal{J}_N,$$

with a normalization constant $\lambda \in \{1, \sqrt{|\mathcal{J}_N|}, |\mathcal{J}_N|\}$. We only consider parameters $\hat{\theta} \in \mathbb{C}^{\mathcal{I}_N}_{\text{sym}} := F(\mathbb{R}^{\mathcal{J}_N})$, where we can employ the convolution theorem (see, e.g., [Gra14]) to define

$$K(\hat{\theta})(v) = F^{-1}\left(\hat{\theta} \cdot Fv\right) \qquad \text{for } v \in \mathbb{R}^{\mathcal{J}_N}, \tag{4.11}$$

which is referred to as the FNO or spectral implementation of convolution.

How do FNOs React to Resolution Changes? The number of Fourier coefficients of $\hat{\theta} \in \mathbb{C}^{\mathcal{I}_M}_{\text{sym}}$ is independent of the input resolution. Nevertheless, the point-wise multiplication in Eq. (4.11) is only defined for inputs $v \in \mathbb{R}^{\mathcal{J}_M}$, thus the question arises how FNOs can be adapted to dimension mismatch. Here, we use a conceptually similar idea, by employing zero-padding. However, the important and major difference is that this zero-padding is performed in the spectral domain. Assuming that $N > M$ is odd, we define

$$\hat{\theta}^{M\to N}_k = \begin{cases} \hat{\theta}_k & \text{for } k \in \mathcal{I}_N \cap \mathcal{I}_M, \\ 0 & \text{for } k \in \mathcal{I}_N \setminus \mathcal{I}_M, \end{cases}$$

and the spectral implementation of convolution for $v_N \in \mathbb{R}^{\mathcal{J}_N}$ is given as

$$K(\hat{\theta})(v_N) := K(\hat{\theta}^{M\to N})(v_N).$$

What Is The Difference Between Standard and FNO Implementation? The difference between the spectral and spatial implementation is best explained in [FNO, Fig. 1], which we repeat in Fig. 4.1 for convenience. We are given a kernel $\theta \in \mathbb{R}^{\mathcal{J}_M}$, its unnormalized Fourier transform $\hat{\theta} = \lambda F(\theta)$ and an input $v_N \in \mathbb{R}^{\mathcal{J}_N}$. If $M = N$, i.e., all dimensions match, we observe that spectral and spatial implementation are equivalent, see the middle row of Fig. 4.1. However, if we consider a higher resolution variant of the image with $N > M$, spatial zero-padding results in the kernel being localized in space and therefore the effect of convolving it with v_N changes. On the other hand, for the spectral implementation, we observe an equivariant behavior. The resolution of the output changes, but qualitatively the effect of the filter stays the same.

Connection to Interpolation As hinted in Fig. 4.1, when changing the resolution, the spectral implementation of convolution can be interpreted as a standard convolution with an interpolated kernel. In fact, we observe that for $\theta \in \mathbb{R}^{\mathcal{J}_M}, \hat{\theta} = \lambda F(\theta)$ and $v_N \in \mathbb{R}^{\mathcal{J}_N}$ we have that

$$\begin{aligned} K(\hat{\theta})(v_N) &= F^{-1}\left(\hat{\theta}^{M\to N} \cdot Fv_N\right) \\ &= F^{-1}\left(F\, F^{-1}\hat{\theta}^{M\to N} \cdot Fv_N\right) \\ &= F^{-1}\left(F\theta^{M\xrightarrow{\Delta}N} \cdot Fv_N\right) = \theta^{M\xrightarrow{\Delta}N} * v_N \\ &= C(\theta^{M\xrightarrow{\Delta}N})(v_N). \end{aligned}$$

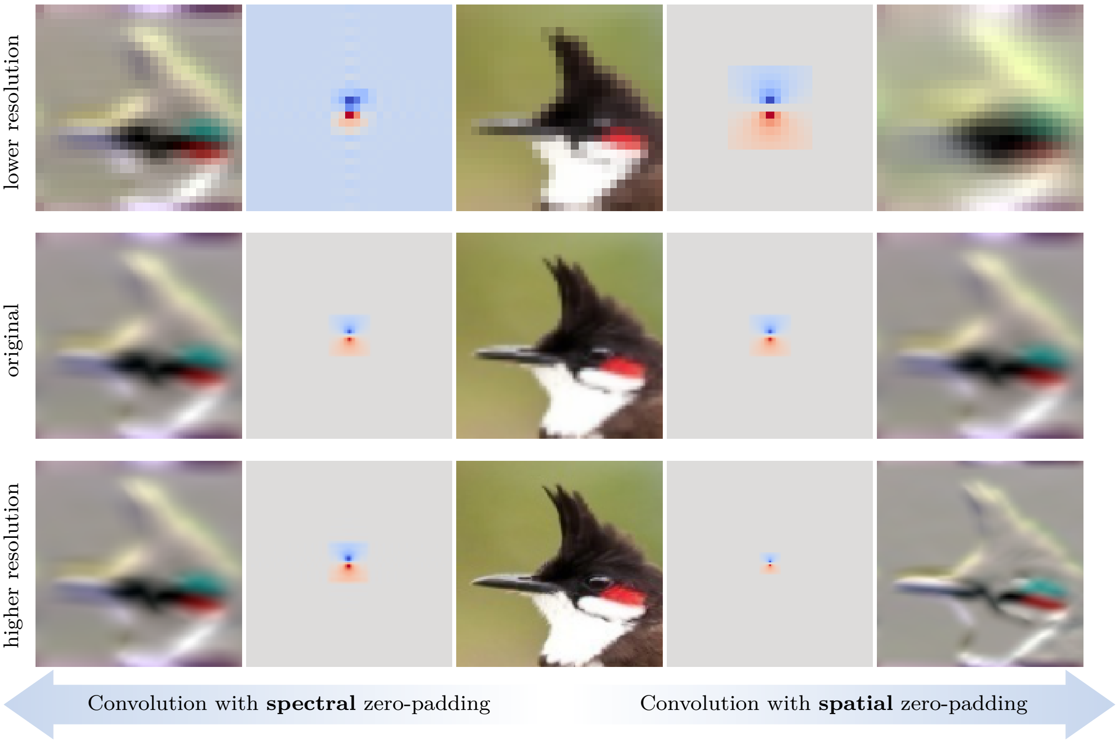

lower resolution
original
higher resolution
Convolution with **spectral** zero-padding
Convolution with **spatial** zero-padding

Figure 4.1.: The image is taken from [Kab22, Fig. 1]—depicting a red whiskered bulbul taken from the Birds500 dataset [Pio21]—and visualizes the different effects of spectral and spatial zero-padding.

Therefore, applying one FNO layer is equivalent to applying a standard CNN layer with a trigonometric interpolation of the kernel.

Adaption to Even Dimensions Zero-padding of the spectral coefficients only fulfills Hermitian symmetry in the case where M, N are odd. In order to adapt this to the even case, we employ so-called Nyquist splitting, see [Bri19]. In all our experiments, the implementation carefully employs this method, which ensures that the output of the spectral convolution is real valued. We also refer to [FNO, Sec. 3.3], where the details on this topic are given.

4.3.2. Analytical Results for FNOs

In this section, we comment on the theoretical findings in [FNO]. We first consider the abstract neural layer as in Eq. (4.7) for which we show continuity and Fréchet-differentiability.

Continuity of Neural Layers The results for neural layers in [FNO] mostly rely on the theory of Nemytskii operators, see, e.g., [Trö10; AP93]. In order to show that the layer $\mathcal{G}$ is a well-defined mapping from L^p to L^q one first needs to identify an exponent $r \in [1, \infty]$ such that the affine part is a mapping $\Psi : L^p \to L^r$. For example, if $\kappa \in L^s$ with $1/r + 1 = 1/p + 1/s$ it follows from Young's convolution inequality that $\mathcal{K}$ maps to L^r, see [Gra14, Thm. 1.2.12]. If then $W = 0$ and $b \in L^r$ we know that Ψ maps to L^r.

To ensure that the Nemytskii operator defines a mapping $\sigma : L^r \to L^q$, one needs to assume a growth condition on $\sigma : \mathbb{R} \to \mathbb{R}$, see [FNO, Eq. 4] and originally [Trö10]. Under these assumptions, we obtain [FNO, Prop. 1]. Concrete examples fulfilling these assumptions are given in [FNO] borrowing concepts from [AP93; Trö10]. The most important activation function that is valid in this setting is the ReLU function

$$\mathrm{ReLU}(x) = \max\{x, 0\}.$$

Proposition 4.4 ([FNO, Prop. 1]). For $1 \leq p, q \leq \infty$ let $\mathcal{L}$ be an operator layer given by (4.7) 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 function 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.

Differentiability of Neural Layers We furthermore consider Fréchet differentiability of a neural layer w.r.t. the input variable. This can also be transferred to differentiability w.r.t. the parameters as we show in [FNO, Ex. 4]. Conceptually, the main result we repeat here is similar to the one on continuity of the last paragraph. The major difference is that we also need to assume differentiability of the activation function, also assuming a growth condition on its derivative. The ReLU function can therefore not be chosen in this setting, however the smooth approximation called GELU ([HG16])

$$\mathrm{GELU}(x) := x\,\Phi(x)$$

where Φ is the CDF of the standard normal distribution, can be employed.

Proposition 4.5 ([FNO, Prop. 2]). For $1 \leq p, q \leq \infty$, let $\mathcal{L}$ be an operator layer given by (4.7) with affine part Ψ as in (4.8). 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., $\tilde{\Psi} = \Psi - b$.

Convertibility Between FNOs and CNNs The following lemma formalizes the intuition that FNOs and CNNs are equivalent in certain settings. Given inputs of a fixed discretization $\mathcal{J}_N$ and parameters $\theta \in \mathbb{R}^{\mathcal{J}_M}$, the standard convolution implementation is equivalent to the spectral one w.r.t. the parameters $\hat{\theta} = \lambda\, F(\theta^{M \to N})$. The subtle but important point here, is that the number of spectral parameters needs to be equal to the input size in order to achieve equivalence. In [FNO, Fig. 3] we observe numerically that the number of used spectral coefficients actually needs to match the input size in order to achieve equivalence.

Lemma 4.9 ([FNO, Lem. 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 $\hat{\theta} \in \mathbb{C}^{I_M}_{\mathrm{sym}}$ and $v \in \mathbb{R}^{J_N}$ it holds true that

$$K(\hat{\theta})(v) = C(T^{-1}(\hat{\theta}^{M \to N}))(v).$$

Together with [FNO, Fig. 3] we see that in order to convert a CNN to a FNO we need a large number of spectral parameters. This is also connected to the fact that spatial locality can only be expressed using more spectral coefficients. Therefore, one might think that FNOs are infeasible due to memory requirements. However, it turns out that directly optimizing over the spectral parameters leads to a comparable performance, already for a low number of Fourier coefficients, which is reported in the blue curve in [FNO, Fig. 3]. This hints that training a FNO via gradient descent leads to different weights, that are not simply a conversion of a similarly trained CNN. The reason for these differences is that the gradients of spectral and standard convolutional layers, that have the same forward pass, are not directly convertible via the Fourier transformation. In fact, one obtains an additional factor, which we formalize in the following lemma.

Lemma 4.10 ([FNO, Lem. 4]). For odd $N \in \mathbb{N}$ and $v, \theta \in \mathbb{R}^{J_N}$ and $\hat{\theta} = T(\theta)$ it holds true that

$$\nabla_{\hat{\theta}} K(\hat{\theta})(v) = \frac{1}{|J_N|}\, T\left(\nabla_\theta C(\theta)(v)\right).$$

Interpolation Equivariance The last result considers the main motivation of the whole section, namely, the resolution invariance of an FNO layer. However, we can not allow an arbitrary resizing operation of the input. Employing the spectral implementation of convolution layer, we can show equivariance w.r.t. trigonometric interpolation of the input.

Corollary 4.11. For $\hat{\theta} \in \mathbb{C}^{I_M}_{\text{sym}}$, $v \in \mathbb{R}^{J_N}$, $M \leq N$ it holds true for any $L \geq M$ that

$$K(\hat{\theta})(v^{N\xrightarrow{\Delta}L}) = \left(K(\hat{\theta})(v)\right)^{N\xrightarrow{\Delta}L}.$$

4.3.3. Numerical Results

In the numerical section of [FNO] we study the convertibility of CNNs to FNOs as discussed in Section 4.3.2 and the resolution invariance of the different proposed approaches. Again, we employ—among others—the `PyTorch` package [Pas+19]. The experiments are conducted on the Fashion-MNIST [XRV17] and a former version of the BIRDS500 [Pio21] dataset. We remark that our implementation carefully treats the case of even kernel or input sizes, via Nyquist splitting. This allows us to apply all the derived results for the odd and also the even case.

The code for all the experiments is available at https://github.com/samirak98/FourierImaging.

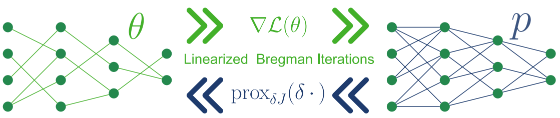
Convertibility and Training Differences As already described in Section 4.3.2, the first experiment, displayed in [FNO, Fig. 3], studies how CNNs can be converted to FNOs. Here, we employ a network with two convolutional layers for feature extraction and a linear classification layer. We train this network—in the spatial implementation—on the FashionMNIST dataset [XRV17] employing varying spatial kernel sizes. Here, we see that for $M = 5$ the spatial implementation already has the best performance and adding more parameters does not increase the performance. We then convert a set of spatial parameters to the Fourier formulation, employing again a varying number of spectral coefficients. It turns out that the requirement of [FNO, Lem. 3], that the number of coefficients must match the input dimension, is indeed relevant. We only obtain the full performance of the CNN if we use all spectral parameters. However, the example also visualizes [FNO, Lem. 4], namely that training a FNO conceptually leads to a different set of parameters. Optimizing over the spectral parameters yields a comparable performance already for a smaller number of coefficients.

Resolution Invariance In the second experiment we study the resolution invariance of the three discretization independent architectures we considered, namely the naive CNN adaptation, input interpolation and the FNO implementation. We first train the convolutional architecture as in the previous paragraph on the FashionMNIST dataset, that has an input size of 28×28. We resize the input data via trigonometric and bilinear interpolation—referred to as data sizing—to simulate multi-resolution data. One expects that the classification performance drops when the data is resized to a smaller size, since this step looses information. However, resizing the images to higher resolutions should yield the same performance.

In [FNO, Fig. 4] we see that the simple adaption does not perform well both in the lower and the higher resolution setting. Employing input interpolation improves the performance in the lower resolution setting. In the higher resolution regime, we obtain a constant performance, which is expected. Finally, we see that the FNO—which was converted from the CNN—performs as well as input interpolation, which justifies the resolution independence of this architecture.

In the second experiment, we trained a ResNet18 [He+16a] on a former version of the BIRDS500 dataset [Pio21], with an input size of 112×112. Concerning the naive adaption and the input interpolation, we observe the same behavior as in the previous example. However, the FNO variant performs slightly worse, which is surprising, especially in the higher resolution regime. In [FNO] we conclude that this is due to the dimension changes within the ResNet architectures. These changes occur due to strided convolutions or pooling operations between the layers. In fact, in order to achieve the performance as displayed in [FNO, Fig. 4 (b)] we replaced every striding by a trigonometric interpolation, which fits better into the FNO framework and vastly improves the performance. Therefore, we summarize that architecture intern dimension changes—which are very common in practice—can potentially hinder resolution invariance.

4.4. Sparsity via Bregman Iterations: [BREG-I]



In this section, we now consider parameter sparsity of neural networks. Starting from the first simple architectures (e.g., [Ros58]) the size of typical networks has grown significantly in the last years [Hoe+21]. While this development allowed to solve increasingly difficult tasks with neural networks, it also leads to immense computational requirements, both for the training and evaluation of the net. Here, the question arises, how these computational tasks can be made more efficient. We list some approaches below, and refer to [Gho+22] for a comprehensive overview.

- **Architecture Optimization**: Designing an architecture that can be trained to have the same performance as a comparable one, while having less parameters, e.g., [EMH19; How+17].
- **Quantization**: Lowering the precision of the machine numbers of the parameters, e.g., [Ban+18; CBD18].
- **Knowledge distillation**: Employing a trained larger network to learn a smaller one, in a student-teacher approach, e.g., [Sch92; HVD15].
- **Pruning**: Parameters with small saliency, i.e., parameters that contribute little to the effective output of the network are removed or set to zero, in order to obtain a sparse weight matrix or a smaller architecture, see, e.g., [LDS89; HSW93].

From the above methods, the pruning approach is most influential to our work in [BREG-I]. Namely, the concept of compressing the neural network by sparsifying the weight matrices is similarly employed. However, there are a few deviations from the classical pruning framework, which we note in the following.

Sparsity via Optimization In [LDS89; HSW93] one assumes to be given a trained neural network, which is then compressed after training based on some criterion. The authors in [CFP97] employ an iterative pruning scheme, where a similar criterion is employed, in order to throw away certain weights after each step. In our work we want to employ a L^1 type penalty (see [CM73]) on the weight matrices $W \in \mathbb{R}^{n\times m}$, i.e.,

$$\|W\|_1 = \sum_{i=1}^{n}\sum_{j=1}^{m} |W_{ij}|\,.$$

In [Tib96] this leads to the so-called Lasso problem

$$\min_\theta \mathcal{L}(\theta) + \lambda \|\theta\|_1\,, \qquad \lambda > 0,$$

where we extend the L^1 norm to the parameter space Θ as discussed in Section 4.4.5. This problem can for example be solved by a proximal gradient descent iteration

$$\theta^{(k+1)} = \text{soft shrinkage}(\theta - \tau\nabla\mathcal{L}(\theta^{(k)})), \tag{4.12}$$

where the shrinkage operator is defined in Example 4.18. This iteration was employed in [Nit14; RVV20; Red+16] to train sparse neural networks.

Sparse-to-Sparse Training All the sparsity-based methods mentioned before, start with dense weight matrices and only decrease the number of parameters during or at the end of the training process. Our approach yields an iteration, where the network is sparse throughout the iteration. In fact, we start with only very few non-zero parameters and only activate necessary weights during training. This paradigm is known as sparse-to-sparse or evolutionary training [Moc+18; DZ19; Evc+20; DYJ19; Fu+22; Hua+16; Liu+21].

Contribution in [BREG-I] Our work falls into the regime of sparse-to-sparse training. However, instead of relying on some heuristic growth strategy, we employ the concept of inverse scale flows (see Section 4.4.1) which allows us to obtain a optimization-driven framework, with a time-continuous interpretation. We propose a stochastic variant of linearized Bregman iterations (Section 4.4.3) and employ it to train a sparse neural network. We show monotonic decrease of the loss in the stochastic setting—which is not possible in the case of proximal gradient descent Eq. (4.12)—and convergence of the iterates under additional convexity assumptions, see Section 4.4.4. Finally, we demonstrate the numerical efficiency of the method (see Section 4.4.5) and provide interesting applications for neural architecture search, which was further developed in [BREG-II].

4.4.1. Preliminaries on Convex Analysis and Bregman Iterations

We first review some necessary concepts from convex analysis that allow us to introduce the framework in [BREG-I]. We refer to [BB18; Roc97; BC11] for a more exhaustive introduction to these topics. In the following Θ denotes a Hilbert space, and we focus on a lower semi-continuous regularization functional $J : \Theta \to (-\infty, \infty]$. Here, J is called lower semi-continuous 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. Furthermore, we require the functional to be convex.

Definition 4.12. Given a Hilbert space Θ and a functional $J : \Theta \to (-\infty, \infty]$.

1. The functional J is called convex, if

$$J(\lambda\overline{\theta} + (1-\lambda)\theta) \leq \lambda J(\overline{\theta}) + (1-\lambda)J(\theta), \quad \forall \lambda \in [0,1],\, \overline{\theta}, \theta \in \Theta. \tag{4.13}$$

2. The effective domain of J is defined as $\operatorname{dom}(J) := \{\theta \in \Theta : J(\theta) \neq \infty\}$ and J is called proper if $\operatorname{dom}(J) \neq \emptyset$.

We want to consider functionals J that are convex, but not necessarily differentiable. Therefore, we define the subdifferential.

Definition 4.13. The subdifferential of a convex and proper functional $J : \Theta \to (-\infty, \infty]$ at a point $\theta \in \Theta$ is given as

$$\partial J(\theta) := \left\{ p \in \Theta \ : \ J(\theta) + \langle p, \overline{\theta} - \theta \rangle \leq J(\overline{\theta}),\ \forall \overline{\theta} \in \Theta \right\}. \tag{4.14}$$

If J is differentiable, then the subdifferential coincides with the classical gradient (or Fréchet derivative). We denote by $\mathrm{dom}(\partial J) := \{\theta \in \Theta \ : \ \partial J(\theta) \neq \emptyset\}$ and observe that $\mathrm{dom}(\partial J) \subset \mathrm{dom}(J)$.

The Bregman Distance The main algorithm in this section are so-called Bregman iterations, which make use of the Bregman distance.

Definition 4.14 (Bregman Distance). Let $J : \Theta \to (-\infty, \infty]$ be a proper and convex functional. Then, for $\theta \in \mathrm{dom}(\partial J), \overline{\theta} \in \Theta$ we define

$$D_J^p(\overline{\theta}, \theta) := J(\overline{\theta}) - J(\theta) - \langle p, \overline{\theta} - \theta \rangle, \quad p \in \partial J(\theta). \tag{4.15}$$

For $p \in \partial J(\theta)$ and $\overline{p} \in \partial J(\overline{\theta})$ we define the *symmetric* Bregman distance as

$$D_J^{\mathrm{sym}}(\overline{\theta}, \theta) := D_J^p(\overline{\theta}, \theta) + D_J^{\overline{p}}(\theta, \overline{\theta}). \tag{4.16}$$

Intuitively, the Bregman distance $D_J^p(\overline{\theta}, \theta)$, measures the distance of J to its linearization around θ, see Fig. 4.2. If J is differentiable, then the subdifferential is single valued—we can suppress the super script p—and we obtain

$$D_J(\overline{\theta}, \theta) = J(\overline{\theta}) - J(\theta) - \langle \nabla J(\theta), \overline{\theta} - \theta \rangle.$$

Example 4.15. For $\Theta = \mathbb{R}^d$ and $J = \frac{1}{2} \left\| \cdot \right\|_2^2$ we see that $\partial J(\theta) = \{\theta\}$ and therefore

$$\begin{aligned} D_J^p(\overline{\theta}, \theta) &= \frac{1}{2}\langle \overline{\theta}, \overline{\theta} \rangle - \frac{1}{2}\langle \theta, \theta \rangle - \langle \theta, \overline{\theta} - \theta \rangle \\ &= \frac{1}{2}\langle \overline{\theta}, \overline{\theta} \rangle + \frac{1}{2}\langle \theta, \theta \rangle - \langle \theta, \overline{\theta} \rangle \\ &= \frac{1}{2} \left\| \overline{\theta} - \theta \right\|_2^2 = J(\overline{\theta} - \theta). \end{aligned}$$

We can easily see, that in general this "distance" is neither definite, symmetric nor fulfills the triangle inequality, hence it is not a metric. However, it fulfills the two distance axioms

$$D_J^p(\overline{\theta}, \theta) \geq 0, \quad D_J^p(\theta, \theta) = 0, \quad \forall \overline{\theta} \in \Theta, \theta \in \mathrm{dom}(\partial J). \tag{4.17}$$

The same holds for the symmetric Bregman distance, where additionally—as the name suggests—the symmetry property is fulfilled.

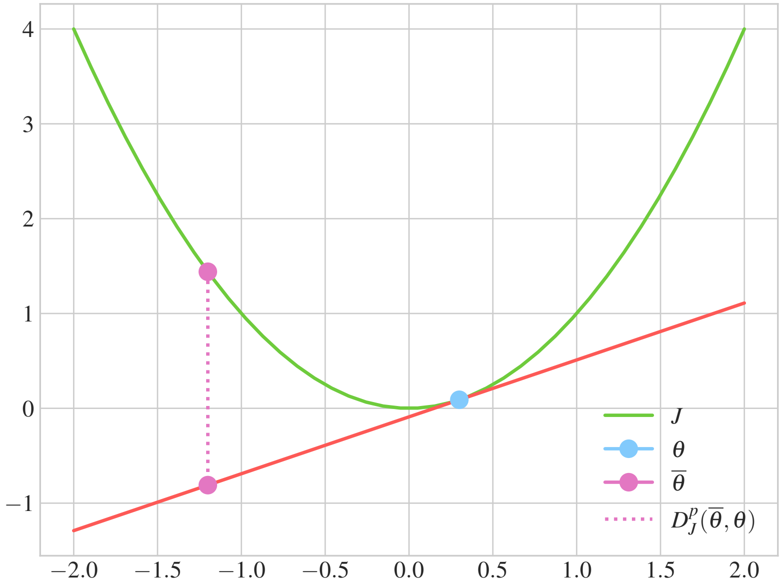


Figure 4.2.: Visualization of the Bregman distance.

The Proximal Operator Another crucial concept in this section, is the so-called proximal operator.

Definition 4.16. Let $J : \Theta \to (-\infty, \infty]$ be convex, proper and lower semicontinuous functional, then we define the *proximal operator* as

$$\operatorname{prox}_J(\overline{\theta}) := \operatorname{argmin}_{\theta\in\Theta} \frac{1}{2}\left\|\theta - \overline{\theta}\right\|^2 + J(\theta).$$

If J is a closed function, i.e., its sublevel sets

$$N_\alpha = \{\theta \in \operatorname{dom} J : J(\theta) \leq \theta\}$$

are closed for every $\alpha \in \mathbb{R}$ then we have that the function $\tilde{J} = \frac{1}{2}\|\theta - \cdot\|^2 + J(\theta)$ is closed, proper and *strongly* convex and therefore has a unique minimizer, see [Roc97, Thm. 27.1]. Additionally, one often considers a regularization parameter $\lambda > 0$ and is then interested in $\operatorname{prox}_{\lambda J}$.

Remark 4.17. The optimality conditions for $\theta = \operatorname{prox}_{\lambda J}(\overline{\theta})$ yield

$$\overline{\theta} - \theta \in \lambda\,\partial J(\theta).$$

For a proper, closed and convex function we obtain

$$\theta = (I + \lambda\partial J)^{-1}(\overline{\theta})$$

where $(I + \lambda\partial J)^{-1}$ is called the *resolvent* and is a one-to-one mapping (see [PB+14, Ch. 3.2]) which justifies the equality in the above equation. If J is differentiable, then we have $\partial J = \{\nabla J\}$ and therefore,

$$\operatorname{prox}_{\lambda J} = (I + \lambda \nabla J)^{-1}.$$

△

In the following, we list two relevant examples for the applications in [BREG-I; BREG-II].

Example 4.18. If $J = \|\cdot\|$ is a norm and $\lambda > 0$ then we have that (see, e.g., [PB+14])

$$\operatorname{prox}_{\lambda J}(\overline{\theta}) = \overline{\theta} - \operatorname{Proj}_{\|\cdot\|^*}(\overline{\theta}/\lambda)$$

where $\operatorname{Proj}_{\|\cdot\|^*}$ denotes the projection operator w.r.t. the dual norm $\|\theta\|^* = \sup\{|\langle f, \theta\rangle| : f \in \Theta^*\}$. In the case of ℓ^p norms on $\mathbb{R}^d$ we know that

$$\|\theta\|_p^* = \|\theta\|_q$$

with $1/p + 1/q = 1$ with the notational convention of $1/\infty = 0$. Especially relevant are the cases $p \in \{1, 2\}$. Here, we then have that

$$\operatorname{prox}_{\lambda\|\cdot\|_2}(\overline{\theta}) = \overline{\theta}\left(1 - \min\left\{\frac{\lambda}{\left\|\overline{\theta}\right\|_2}, 1\right\}\right) = \begin{cases} \overline{\theta}\left(1 - \lambda/\left\|\overline{\theta}\right\|_2\right) & \text{if } \left\|\overline{\theta}\right\|_2 \geq \lambda, \\ 0 & \text{else} \end{cases}$$

and for $i = 1, \dots, n$,

$$\operatorname{prox}_{\lambda\|\cdot\|_1}(\overline{\theta})_i = \operatorname{sign}(\overline{\theta}_i) \max\left\{\left|\overline{\theta}_i\right| - \lambda, 0\right\} = \begin{cases} \overline{\theta}_i - \lambda & \text{if } \overline{\theta} > \lambda, \\ 0 & \text{if } \left|\overline{\theta}_i\right| \leq \lambda, \\ \overline{\theta}_i + \lambda & \text{if } \overline{\theta} < -\lambda, \end{cases}$$

the so-called *soft shrinkage operator.*

Example 4.19 (Group Norms). Another relevant functional J is the group norm $\ell_{1,2}$ that—in the context of sparse neural networks—was first employed by [Sca+17]. Here, we assume that the parameters in $\theta \in \Theta$ can be grouped in a collection of parameters, i.e., $\theta = \{\theta_1, \dots, \theta_s\}$, and we choose

$$J(\theta) = \sum_{g \in \theta} \sqrt{s}\, \|g\|_2 .$$

In this case the proximal operator is given as

$$\operatorname{prox}_{\lambda J}(\overline{\theta})_i = \theta_i \max\left\{1 - \min\left\{\frac{\lambda\sqrt{s}}{\|\theta\|_2}, 1\right\}, 0\right\}.$$

Bregman Iterations Our goal is to minimize a function $\mathcal{L}$ while simultaneously obtaining a low value w.r.t. the functional J. A popular approach considers the regularized problem

$$\min_\theta \mathcal{L}(\theta) + \lambda J(\theta) \qquad \lambda > 0,$$

see, e.g., [Tik43; CP08; DDD04; FS06; FNW07; Cha04; CP11], which however influences the minimizers of the original problem $\min_\theta \mathcal{L}(\theta)$. In the derivation of the Bregman iterations, one can take a different viewpoint. We want to employ an iterative scheme, where in each step we minimize $\mathcal{L}$ while penalizing the distance to the previous iterate. For a stepping parameter τ and starting from some $\theta^{(0)} \in \Theta$ this yields the update

$$\theta^{(k+1)} = \operatorname{argmin}_\theta \tau\mathcal{L}(\theta) + \frac{1}{2}\left\|\theta - \theta^{(k)}\right\|^2 = \operatorname{prox}_{\tau\mathcal{L}}(\theta^{(k)}). \tag{4.18}$$

This concept is known as the proximal point algorithm [Bre67] as well as the minimizing movement scheme [De 93]. If $\mathcal{L}$ is differentiable, this update can be rewritten as

$$\theta^{(k+1)} = (I + \tau\nabla\mathcal{L})^{-1}\theta^{(k)} \Leftrightarrow \frac{1}{\tau}\left(\theta^{(k+1)} - \theta^{(k)}\right) = -\nabla\mathcal{L}(\theta^{(k+1)})$$

which is a implicit Euler discretization ([Eul24]) of the time-continuous gradient flow

$$\partial_t\theta_t = -\nabla\mathcal{L}(\theta_t).$$

We see that the penalization term in Eq. (4.18) is in fact the Bregman distance w.r.t. the functional $\frac{1}{2}\|\cdot\|^2$, see Example 4.15. In order to incorporate an arbitrary convex functional J—and therefore allow each iterate to only slightly deviate w.r.t. the Bregman distance of J to the previous iterate—we employ $D_J^{p^{(k)}}(\cdot, \theta^{(k)})$ as a penalization term. Here, we obtain a update scheme for the subgradients $p^{(k)}$, as follows,

$$\begin{align} \theta &= \operatorname{argmin}_{\theta\in\Theta} D_J^{p^{(k)}}\left(\theta, \theta^{(k)}\right) + \tau\mathcal{L}(\theta) \tag{4.19}\\ &\Leftrightarrow p^{(k)} + \tau\nabla\mathcal{L}(\theta) \in \partial J(\theta). \tag{4.20}\end{align}$$

This yields the *Bregman iteration* of [Osh+05]

$$\begin{align} \theta^{(k+1)} &= \operatorname{argmin}_{\theta\in\Theta} D_J^{p^{(k)}}\left(\theta, \theta^{(k)}\right) + \tau\mathcal{L}(\theta), \tag{4.21a}\\ p^{(k+1)} &= p^{(k)} - \tau\nabla\mathcal{L}(\theta^{(k+1)}) \in \partial J(\theta^{(k+1)}). \tag{4.21b}\end{align}$$

The nature of Bregman iterations requires starting with an iterate $\theta^{(0)}$ that has a low value in J—preferably $J(\theta^{(0)}) = 0$—and only increase $J(\theta^{(k)})$ gradually in each step.

Remark 4.20. Originally, the iterations were employed for solving inverse problems. Here, we are given a forward operator $A : \Theta \to \tilde{\Theta}$ and a noisy measurement $f = A\theta + \delta$

where $\delta \in \tilde{\Theta}$ models additive noise. The loss function is then of the form

$$\mathcal{L} = \frac{1}{2} \left\| A \cdot - f \right\|_2^2$$

for which one can show that the Bregman iterations converge to a solution of

$$\min \left\{ J(\theta) : A\theta = f \right\}, \tag{4.22}$$

see, e.g., [Osh+05]. In comparison to this, the concept of adding a regularizing term with parameter $\lambda > 0$, i.e., considering the problem

$$\min_{\theta} \mathcal{L}(\theta) + \lambda J(\theta)$$

actually modifies the minimizers. In this sense, Bregman iterations do not introduce a bias. However, for applications like image denoising, where A is the identity, this also implies that the iterations converge back to the noisy image. Therefore, one often has to employ an early stopping criterion. △

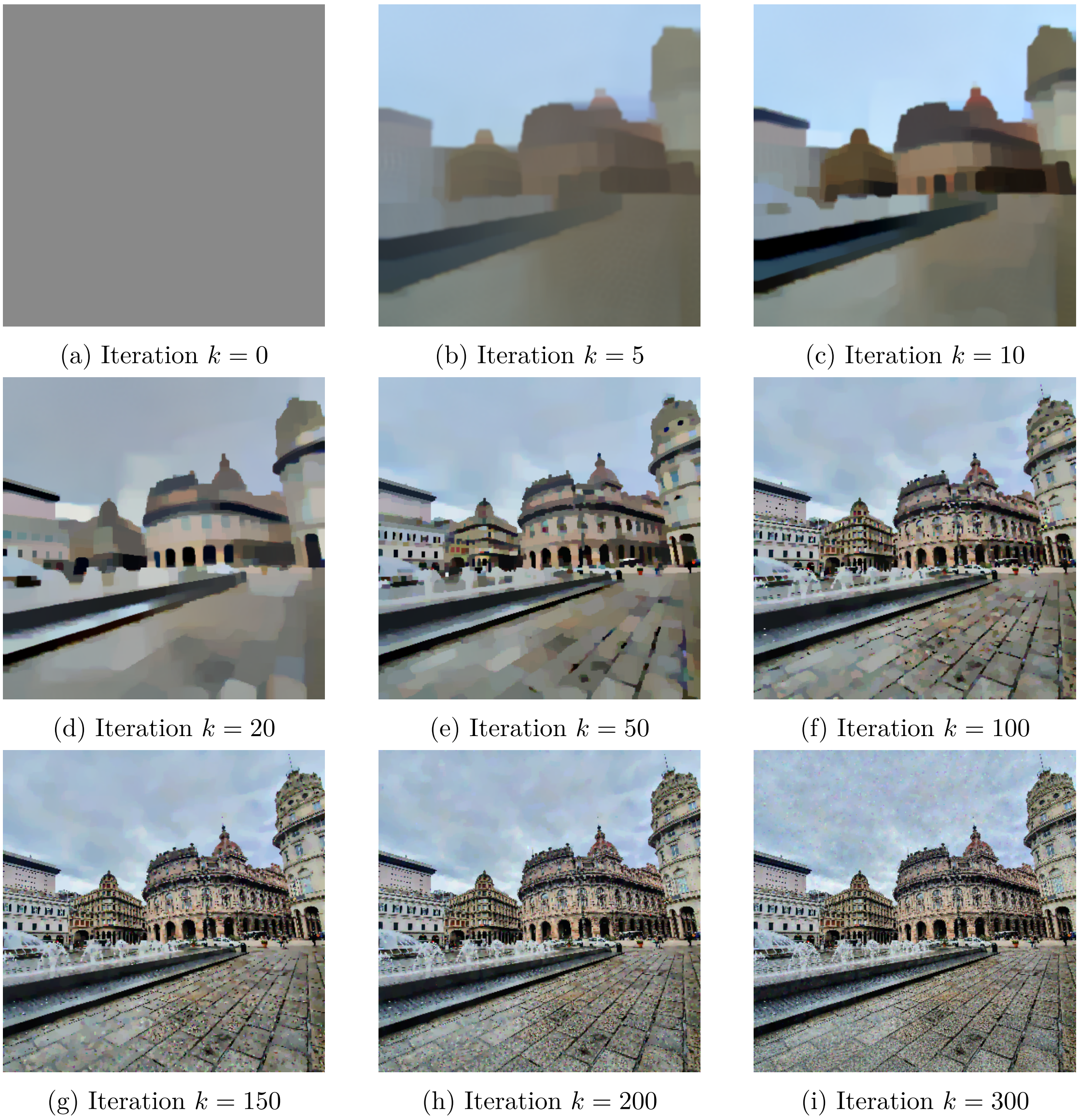
Example 4.21. In order to obtain an intuition of the behavior of Bregman iterations, we consider an image denoising task. I.e., we are given a noisy image $\mathbb{R}^{n\times m} \ni f = u+\delta$ where $\delta \in \mathbb{R}^{n\times m}$ models additive noise. In order to obtain $u \in \mathbb{R}^{n\times n}$ from f, we employ the TV functional [ROF92]

$$J(u) = TV(u) := \sum_{i,j} \sqrt{\left|u_{i+1,j} - u_{i,j}\right|^2 + \left|u_{i,j+1} - u_{i,j}\right|^2}$$

together with the loss function $\mathcal{L}(u) := \frac{1}{2} \left\| u - f \right\|_2^2$. We start with an image $u^{(0)}$ such that $TV(u^{(0)}) = 0$, i.e., a constant image. In Fig. 4.3 we visualize the iterates at for different k. At the start, the iterates only display features on a larger scale, while at the end, they converge back to the smallest possible scale, the noisy data. In order to obtain an appropriate denoising, one needs to employ a early stopping here. This fits well to the insight from Eq. (4.22), since here the forward operator is the identity, i.e.,

$$\left\{ u : \frac{1}{2} \left\| u - f \right\|^2 = 0 \right\} = \{f\}.$$

It should also be noted that this example only serves a explanatory purpose. In practice, directly applying Eq. (4.21) for $J = TV$ can become infeasible, since the first minimization problem is expensive.



(a) Iteration $k = 0$

(b) Iteration $k = 5$

(c) Iteration $k = 10$

(d) Iteration $k = 20$

(e) Iteration $k = 50$

(f) Iteration $k = 100$

(g) Iteration $k = 150$

(h) Iteration $k = 200$

(i) Iteration $k = 300$

Figure 4.3.: Bregman iterations for the image denoising in Example 4.21. We observe the inverse scale space character of the method. Furthermore, we see that the iterations converge back to the noisy data as described in Remark 4.20. In order to solve the inner problem we employed the split Bregman method of [GO09].

The time continuous flow for $\tau \to 0$ is known as the *inverse scale space* flow [Bur+06; Bur+07],

$$\begin{cases} \dot{p}_t = -\nabla\mathcal{L}(\theta_t), \\ p_t \in \partial J(\theta_t). \end{cases}$$

For $J = \frac{1}{2}\left\|\cdot\right\|_2^2$ we see that $\partial J(\theta_t) = \theta_t$ and therefore, we obtain the standard gradient flow. Hence, the inverse scale space flow is a generalization of the standard gradient flow.

4.4.2. Linearized Bregman Iterations and Mirror Descent

The minimization step in Eq. (4.21) is infeasible for large scale applications, especially in our setting for neural networks. Therefore, we employ the idea introduced in [Yin+08; COS09]. We first linearize the loss function around the previous iterate,

$$\mathcal{L}(\theta) \approx \mathcal{L}(\theta^{(k)}) + \left\langle \nabla\mathcal{L}(\theta^{(k)}), \theta - \theta^{(k)} \right\rangle.$$

The next step is to replace J with the strongly convex elastic net regularization

$$J_\delta := J + \frac{1}{2\delta}\left\|\cdot\right\|_2^2. \tag{4.23}$$

The minimization step in Eq. (4.21) then transforms to

$$\begin{aligned}
&\operatorname{argmin}_{\theta\in\Theta} D_{J_\delta}^{p^{(k)}}\left(\theta, \theta^{(k)}\right) + \tau\left\langle \nabla\mathcal{L}(\theta^{(k)}), \theta \right\rangle && (4.24)\\
&\quad= \operatorname{argmin}_{\theta\in\Theta} J(\theta) + \frac{1}{2\delta}\left\|\theta\right\|_2^2 - \langle p^{(k)}, \theta\rangle + \tau\left\langle \nabla\mathcal{L}(\theta^{(k)}), \theta \right\rangle\\
&\quad= \operatorname{argmin}_{\theta\in\Theta} J(\theta) + \frac{1}{2\delta}\left\|\theta - \delta\left(p^{(k)} - \tau\nabla\mathcal{L}(\theta^{(k)})\right)\right\|_2^2 - \underbrace{\left\|p^{(k)} - \tau\nabla\mathcal{L}(\theta^{(k)})\right\|_2^2}_{\text{constant in } \theta}\\
&\quad= \operatorname{prox}_{\delta J}\left(\delta\left(p^{(k)} - \tau\nabla\mathcal{L}(\theta^{(k)})\right)\right).
\end{aligned}$$

Note, that here $p^{(k)}$ is a subgradient of J_δ at $\theta^{(k)}$, where we derive the subgradient update rule

$$p^{(k+1)} := p^{(k)} - \tau\mathcal{L}(\theta^{(k)}).$$

This yields the linearized Bregman iterations

$$p^{(k+1)} = p^{(k)} - \tau\nabla\mathcal{L}(\theta^{(k)}), \tag{4.25a}$$
$$\theta^{(k+1)} = \operatorname{prox}_{\delta J}(\delta p^{(k+1)}). \tag{4.25b}$$

The last line is equivalent to $p^{(k+1)} \in \partial J_\delta(\theta^{(k+1)})$ for which we obtain the continuous linearized flow

$$\begin{aligned} \dot{p}_t &= -\nabla\mathcal{L}(\theta_t), \\ p_t &\in \partial J_\delta(\theta_t), \end{aligned}$$

see [Bur+06; Bur+07].

Connections To Mirror Descent As already noticed in [Vil+23] linearized Bregman iterations are equivalent to mirror descent in some situations. We show the equivalence in the following, where we employ similar arguments as in [BT03]. One assumes to be given a differentiable and strongly convex function $h : \Theta \to \mathbb{R}$, i.e.,

$$h(\overline{\theta}) - h(\theta) - \langle \nabla h(\theta), \overline{\theta} - \theta \rangle \geq \frac{1}{2} \left\| \overline{\theta} - \theta \right\|_2^2$$

for all $\theta, \overline{\theta} \in \Theta$. The mirror descent update then reads ([NY83; BT03])

$$\theta^{(k+1)} = \nabla h^* \left(\nabla h \left(\theta^{(k)} \right) - \tau \nabla \mathcal{L}(\theta^{(k)}) \right) \tag{4.26}$$

where h^* denotes the Fenchel conjugate

$$h^*(p) = \sup_\theta \langle p, \theta \rangle - h(\theta)$$

with the gradient (see [BV04])

$$\nabla h^*(p) = \operatorname{argmax}_\theta \left\{ \langle p, \theta \rangle - h(\theta) \rangle \right\}.$$

Therefore, we see that Eq. (4.26) can be written as

$$\begin{aligned}
\theta^{(k+1)} &= \operatorname{argmax}_\theta \left\{ \left\langle \nabla h \left(\theta^{(k)} \right) - \tau \nabla \mathcal{L}(\theta^{(k)}), \theta \right\rangle - h(\theta) \right\} \\
&= \operatorname{argmax}_\theta \left\{ -D_h(\theta, \theta^{(k)}) - \tau \left\langle \nabla \mathcal{L}(\theta^{(k)}), \theta \right\rangle \right\} \\
&= \operatorname{argmin}_\theta \left\{ D_h(\theta, \theta^{(k)}) + \tau \left\langle \nabla \mathcal{L}(\theta^{(k)}), \theta \right\rangle \right\}
\end{aligned}$$

which was our starting point to derive linearized Bregman iterations for $h = J_\delta$ in Eq. (4.24). In fact, since it is strongly convex, we can always find a convex functional $J : \Theta \to \mathbb{R}$ such that $h = J + \frac{1}{2} \left\| \cdot \right\|_2^2$. Therefore, we see, that Eq. (4.25) is a more general formulation of Eq. (4.26).

4.4.3. Stochastic and Momentum Variants

We want to employ linearized Bregman iterations to train a neural network. As mentioned in Section 4.1.2, we usually do not compute the full gradient of $\mathcal{L}$ but rather a minibatched variant. This yields stochastic Bregman iterations:

$$\begin{aligned}
&\text{draw } \omega^{(k)} \text{ from } \Omega \text{ using the law of } \mathbb{P}, \\
g^{(k)} &:= g(\theta^{(k)}; \omega^{(k)}), \\
v^{(k+1)} &:= v^{(k)} - \tau^{(k)} g^{(k)}, \\
\theta^{(k+1)} &:= \operatorname{prox}_{\delta J}(\delta v^{(k+1)}).
\end{aligned} \tag{4.27}$$

We also abbreviate this scheme as the *LinBreg* algorithm in the following. The presence of a stochastic gradient estimator significantly complicates the convergence analysis, as observed in Section 4.4.4. However, this algorithm can now be efficiently employed to train a neural network. For the analogous stochastic mirror descent algorithm, we refer to [Nem+09].

Momentum Variant Typically, the learning process of a neural network can be improved by introducing a momentum term (see, e.g., [Nes83; Qia99]) in the optimizer. In our case, this can be achieved by replacing the gradient update on the subgradient variable. In [BREG-I] we first consider the inertia version of the gradient flow as

$$\begin{cases} \gamma \ddot{v}_t + \dot{v}_t = -\nabla \mathcal{L}(\theta_t), \\ v_t \in \partial J_\delta(\theta_t). \end{cases}$$

The discretization then reads

$$\begin{aligned} m^{(k+1)} &= \beta^{(k)} m^{(k)} + (1-\beta^{(k)})\tau^{(k)} g^{(k)}, \\ v^{(k+1)} &= v^{(k)} - m^{(k+1)}, \\ \theta^{(k+1)} &= \operatorname{prox}_{\delta J}(\delta v^{(k+1)}). \end{aligned} \tag{4.28}$$

Adamized Bregman Iteration We shortly remark that one can replace the momentum update in Eq. (4.28) with an Adam update [KB14]. This then yields an Adamized version of linearized Bregman iterations, as employed in [BREG-I].

4.4.4. Convergence of Stochastic Bregman Iterations

While various previous works prove convergence of linearized Bregman iterations (see, e.g., [Osh+05; COS09]), the stochastic setting requires special treatment. In [BREG-I] we prove the first guarantees for the algorithm in Eq. (4.27). Other work on convergence of stochastic Bregman iterations, or mirror descent requires a differentiable functional J, see [DEH21; HR21; ZH18; DOr+21; AKL22]. Since our main motivation is a L^1 type functional, this is not applicable. Therefore, we present the novel convergence analysis of [BREG-I].

Assumptions on the Gradient Estimator In order to obtain convergence guarantees, we need to assume mainly two properties on the gradient estimator $g(\cdot,\cdot)$. First, we assume unbiasedness, which means

$$\mathbb{E}\left[g(\theta;\omega)\right] = \nabla\mathcal{L}(\theta) \text{ for all } \theta \in \Theta.$$

The second assumption we need in the following is referred to as *bounded variance* of the estimator.

Assumption 4.22 (Bounded variance). We assume that there exists a constant $\sigma > 0$ such that for any $\theta \in \Theta$ it holds

$$\mathbb{E}\left[\left\|g(\theta;\omega) - \nabla\mathcal{L}(\theta)\right\|^2\right] \leq \sigma^2. \tag{4.29}$$

Remark 4.23. We remark, that this property is weaker than the bounded gradient assumption

$$\mathbb{E}\left[\|g(\theta;\omega)\|^2\right] \leq C$$

for some constant $C > 0$. This condition would contradict a strong convexity assumption—which we employ in Theorem 4.30—as shown in [Ngu+18]. A weaker and more realistic assumption is the affine variance condition, as used in [Faw+22]

$$\mathbb{E}\left[\|g(\theta;\omega) - \nabla\mathcal{L}(\theta)\|^2\right] \leq \sigma_0^2 + \sigma_1^2\ \|\nabla\mathcal{L}(\theta)\|^2,$$

for constants σ_0, σ_1. Proving convergence of the stochastic linearized Bregman iterations under the previous assumption is an interesting open problem. △

Assumptions on the Regularizer and on the Loss Function The assumptions on the regularization functional J are mild and merely ensure the well-definedness of the proximal mapping.

> **Assumption 4.24 (Regularizer).** We assume that $J : \Theta \to (-\infty, \infty]$ is a convex, proper, and lower semicontinuous functional on the Hilbert space Θ.

Our assumptions on the loss function $\mathcal{L}$ are more restrictive. We require it to be bounded from below and differentiable, which are both standard assumptions. Additionally, we require Lipschitz continuity of the gradient, which is commonly employed in optimization literature.

> **Assumption 4.25 (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 $L \in (0, \infty)$:
>
> $$\left\|\nabla\mathcal{L}(\tilde{\theta}) - \nabla\mathcal{L}(\theta)\right\| \leq L \left\|\tilde{\theta} - \theta\right\|, \quad \forall \theta, \tilde{\theta} \in \Theta. \tag{4.30}$$

If the loss function $\mathcal{L}$ fulfills the previous assumptions, we are able to prove loss decay of the iterates, see Theorem 4.29. However, in order to show convergence of the iterates, we additionally need a convexity assumption. For a differentiable functional J, the authors in [DEH21] assumed

$$\nu\ D_J(\overline{\theta}, \theta) \leq D_{\mathcal{L}}(\overline{\theta}, \theta),$$

which for twice differentiable J and $\mathcal{L}$ yields

$$\nu\ \nabla^2 J \lesssim \nabla^2 \mathcal{L}, \qquad \forall \overline{\theta}, \theta \in \Theta.$$

Plugging in the definition of the Bregman dist $D_{\mathcal{L}}$ we obtain

$$\nu\ D_J(\overline{\theta}, \theta) \leq \mathcal{L}(\overline{\theta}) - \mathcal{L}(\theta) - \langle \nabla \mathcal{L}(\theta), \overline{\theta} - \theta \rangle.$$

In this form, one observes that this is in fact a convexity assumption on $\mathcal{L}$ w.r.t. a J dependent distance, as employed in [BREG-I].

Assumption 4.26 (Strong convexity). For a proper convex function $H : \Theta \to \mathbb{R}$ and $\nu \in (0, \infty)$, we say that the loss function $\theta \mapsto \mathcal{L}(\theta)$ is ν-strongly convex w.r.t. H, if

$$\mathcal{L}(\overline{\theta}) \geq \mathcal{L}(\theta) + \langle \nabla \mathcal{L}(\theta), \overline{\theta} - \theta \rangle + \nu D_H^p(\overline{\theta}, \theta), \quad \forall \theta, \overline{\theta} \in \Theta, p \in \partial H(\theta). \tag{4.31}$$

Remark 4.27. We have two relevant cases for the choice of H. For $H = \frac{1}{2}\left\|\cdot\right\|^2$ Assumption 4.26 reduces to standard strong ν-convexity. The other relevant case, is $H = J_\delta$, i.e., we consider convexity w.r.t. to the functional J_δ. △

Remark 4.28. In the setting of training a neural network, where we employ the empirical loss Eq. (4.1), this convexity assumption usually fails. While it is possible to enforce this conditions only locally around the minimum, this does not significantly improve the applicability. For future work, it would be desirable to enforce a Kurdyka–Łojasiewicz inequality, as in [Ben+21] for the deterministic case. △

Loss Decay The first convergence result considers the loss decay of the iterates. Here, we do not assume convexity of the loss function. Under these assumptions the authors in [Ben+21; BB18] were able to show the inequality

$$\begin{aligned}\mathbb{E}\left[\mathcal{L}(\theta^{(k+1)})\right] + \frac{1}{\tau^{(k)}}\mathbb{E}\left[D_J^{\mathrm{sym}}(\theta^{(k+1)}, \theta^{(k)})\right] + \frac{C}{2\delta\tau^{(k)}}\mathbb{E}\left[\left\|\theta^{(k+1)} - \theta^{(k)}\right\|^2\right]& \\ \leq \mathbb{E}\left[\mathcal{L}(\theta^{(k)})\right]&.\end{aligned}$$

In our setting, employing a stochastic gradient estimator, we are able to prove a similar estimate. We obtain an additional term scaled by σ, which controls the expected squared difference between the gradient estimator and the actual gradient.

Theorem 4.29 ([BREG-I, Thm. 2]: Loss decay). Assume that Assumptions 4.22, 4.24 and 4.25 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 (4.27) satisfy

$$\mathbb{E}\left[\mathcal{L}(\theta^{(k+1)})\right] + \frac{1}{\tau^{(k)}}\mathbb{E}\left[D_J^{\mathrm{sym}}(\theta^{(k+1)}, \theta^{(k)})\right] + \frac{C}{2\delta\tau^{(k)}}\mathbb{E}\left[\left\|\theta^{(k+1)} - \theta^{(k)}\right\|^2\right] \\ \leq \mathbb{E}\left[\mathcal{L}(\theta^{(k)})\right] + \tau^{(k)}\delta\frac{\sigma^2}{2c}. \tag{4.32}$$

Convergence of the Iterates Here, we have two cases respectively proving convergence w.r.t. the L^2 distance and the Bregman distance of J_δ. The first assumes strong convexity with $H = \frac{1}{2}\|\cdot\|^2$ in Assumption 4.26.

Theorem 4.30 ([BREG-I, Thm. 6]: Convergence in norm). Assume that Assumptions 4.22, 4.24 and 4.25 and Assumption 4.26 for $H = \frac{1}{2}\|\cdot\|^2$ 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 (4.27) satisfy the following:

- Letting $d_k := \mathbb{E}\left[D_{J_\delta}^{v^{(k)}}(\theta^*, \theta^{(k)})\right]$ it holds

$$d_{k+1} - d_k + \frac{\mu}{4}\tau^{(k)}\mathbb{E}\left[\left\|\theta^* - \theta^{(k+1)}\right\|^2\right] \leq \frac{\sigma}{2}\left((\tau^{(k)})^2 + \mathbb{E}\left[\left\|\theta^{(k)} - \theta^{(k+1)}\right\|^2\right]\right). \tag{4.33}$$

- The iterates possess a subsequence converging in the L^2-sense of random variables:

$$\lim_{j\to\infty}\mathbb{E}\left[\left\|\theta^* - \theta^{(k_j)}\right\|^2\right] = 0. \tag{4.34}$$

Here, J_δ is defined as in (4.23).

For the second result we assume convexity w.r.t. the Bregman distance, i.e., we choose $H = J_\delta$ in Assumption 4.26. This induces a relation between the Bregman distance of J and the loss function $\mathcal{L}$, which has been similarly employed in [DEH21].

Theorem 4.31 ([BREG-I, Thm. 11]: Convergence in the Bregman distance). Assume that Assumptions 4.22, 4.24 and 4.25 and Assumption 4.26 for $H = J_\delta$ 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 (4.27) satisfy the following:

- Letting $d_k := \mathbb{E}\left[D_{J_\delta}^{v^{(k)}}(\theta^*, \theta^{(k)})\right]$ it holds
$$d_{k+1} \leq \left[1 - \tau^{(k)}\nu\left(1 - \tau^{(k)}\frac{2\delta^2 L^2}{\nu}\right)\right] d_k + \delta(\tau^{(k)})^2\sigma^2. \tag{4.35}$$
- For any $\varepsilon > 0$ there exists $\tau > 0$ such that if $\tau^{(k)} = \tau$ for all $k \in \mathbb{N}$ then
$$\limsup_{k\to\infty} d_k \leq \varepsilon. \tag{4.36}$$
- If $\tau^{(k)}$ is such that
$$\lim_{k\to\infty} \tau^{(k)} = 0 \quad \text{and} \quad \sum_{k=0}^{\infty} \tau^{(k)} = \infty \tag{4.37}$$
then it holds
$$\lim_{k\to\infty} d_k = 0. \tag{4.38}$$

Here, J_δ is defined as in (4.23).

4.4.5. Numerical Results and Practical Considerations

Before briefly reviewing the numerical results in [BREG-I, Sec. 4], we remark on some practical considerations. In particular, we comment on the parameter initialization strategy. All the experiments were implemented in `Python` [VD95] employing—among others—the `PyTorch` package [Pas+19].

Parameter Initialization As already noticed in [GB10] parameter initialization has a significant impact on the training of the neural network. Here, in contrast to standard Bregman methods, we are not able to initialize the parameters of the neural network as $\theta = 0$. This is due to that fact, that a zero initialization induces symmetries in the network weights, for which one cannot utilize the full expressivity of the architecture [GBC16, Ch. 6]. Therefore, we rather employ the approach from [Liu+21; DZ19; Mar10] of sparsifying weight matrices $\tilde{W}^l \in \mathbb{R}^{n_{l+1}\times n_l}$ up to a certain level, by a

The code for all the experiments is available at github.com/TimRoith/BregmanLearning.

pointwise multiplication with a binary mask $M^l \in 0, 1^{n_{l+1} \times n_l}$

$$W^l := \tilde{W}^l \odot M^l.$$

Each entry in M^l is i.i.d. sampled from a Bernoulli distribution

$$M^l_{i,j} \sim \mathcal{B}(r).$$

where the parameter r determines the sparsity,

$$\mathrm{N}(W^l) := \frac{\left\|W^l\right\|_0}{n_l \cdot n_{l-1}} = 1 - \mathrm{S}(W^l)$$

with N denoting the percentage of used parameters and S the sparsity. In [GB10] the authors advise to especially control to the variance of the parameter initialization distribution, for which in [BREG-I] we derive

$$\operatorname{Var}\left[\tilde{W}^l\right] = \frac{1}{r} \operatorname{Var}\left[\tilde{W}^l \odot M^l\right] \tag{4.39}$$

and therefore scale the weights with the sparsity parameter r at initialization.

Choice of Regularizers In all our experiments we choose a L^1 type sparsity promoting regularization function functional J. We do not employ any coupling between weight matrices of different layers, and therefore for $\theta = ((W_1, b_1), \dots, (W_L, b_L))$ we have

$$J(\theta) = \sum_{l=1}^{L} J_l(W_l)$$

where J^l is chosen according to the layer type. In the easiest case of a fully connected layer, we can choose

$$J_l(W_l) := \|W_l\|_1 .$$

In the case of a convolutional layer we have that W^l is determined by convolutional kernels $K_{i,j} \in \mathbb{R}^{k \times k}$, see Section 4.1.1. Here, we typically employ a group sparsity term in the form

$$J_l(W_l) = \|W_l\|_{2,1} = \sum_{i,j} \|K_{i,j}\|_2 .$$

The outer sum acts as a L^1 regularizer on the instances $\|K_{i,j}\|$. Sparsity, in this sense, then amounts to having indices (i, j) for which $\|K_{i,j}\|_2 = 0 \Leftrightarrow K_{i,j} = 0$, i.e., we prune away whole convolutional filters. This effect is displayed in [BREG-I, Fig. 1]. We can also employ group sparsity on fully connected layers, by considering row sparsity of $W_l \in \mathbb{R}^{n_{l+1}, n_l}$s

$$J_l(W_l) = \sum_{i=1}^{n_{l+1}} \|W_{i,:}\|_2 = \sum_{i=1}^{n_{l+1}} \sqrt{\sum_{js=1}^{n_l} W_{i,j}^2}.$$

In this setting we have a L^1 penalty on the row norms $\|W_{i,:}\|_2$ which therefore enforces whole rows to be zero. This is relevant, if we employ a layer architecture with $\Psi^l(0) = 0$, e.g., using no bias vectors and the ReLU activation function. In this setting, if the ith row of W_l is zero this effectively means, that the ith neuron in layer $l+1$ is inactive. This observation allows the neural architecture search in one of the following paragraphs.

Comments on the Numerical Results We briefly remark on the numerical results as displayed in [BREG-I, Sec. 4]. In the experiments we employed feed-forward networks with simple linear, convolutional and residual layers and tested on the three datasets [Kri09; XRV17; LC10].

The basic comparison between the algorithms SGD, ProxGD and LinBreg shows the qualitative behavior of each iteration. The sparse initialization does not have any effect on SGD, since it does not preserve the sparsity in any way. ProxGD rather starts with many active parameters and reduces this number during the iteration. Only the discretization of the inverse scale space flow—via Bregman iterations—shows the desired behavior of gradually adding active parameters. Furthermore, in [BREG-I, Fig. 2] we can see, that the choice of λ in the regularizer $J = \lambda \|\cdot\|_1$ changes the results significantly. In the light of Eq. (4.22), this is not expected for the standard Bregman iterations with a convex loss. It is therefore interesting to see, that in our non-convex and stochastic situation this effect changes.

The momentum variants, as discussed in Section 4.4.3 yield the desired effect of enhancing the validation accuracy, and respectively converging faster. However, in each of the experiments, one can also observe that adding a momentum term has the effect that more parameters are added faster. On the one hand, this could mean that the network actually requires more parameters to have a higher accuracy, and a momentum variant is more likely to increase the number of needed parameters. However, the quantitative evaluation on the CIFAR10 dataset [Kri09] shows, that especially the Adamized version tends to increase the number of used parameters rather aggressively, while only slightly increasing the performance of the net. The performance here is very similar to the one of proximal gradient descent. However, the training of a residual network seems to be slightly better with a standard Lasso implementation. Neglecting the non-differentiability of the L^1 norm, one computes a derivative via automatic differentiation [Ral81; MDA15] (we employed the `autograd` library of the `PyTorch` package [Pas+19]). In order to obtain true zeros in the weight matrix one then has to employ a thresholding operation after the training. In some sense, this method is not a proper sparse training approach, but rather a regularization method with an added pruning step at the end.

Comments on Efficiency One of the major advantages of the Bregman approach, is that the network is sparse already during the training time. As with all sparse-to-sparse training approaches, this yields a very small number of active parameters over all training step. This sparsity can be easily exploited in each forward pass. However, it is not directly possible to achieve performance gains during the backward pass of the network, since in general

$$W^l_{ij} = 0 \not\Rightarrow \partial_{W^l_{ij}} \mathcal{L}(\theta) = 0.$$

In [BREG-I; BREG-II] there are no evaluations on the training time and memory consumption of the Bregman algorithm. This is an interesting open question for future work. We remark that the computational complexity of the LinBreg algorithm does not increase significantly, compared to SGD, since the evaluation of the proximal operator is very efficient for L^1 type functionals.

Neural Architecture Search An interesting aspect hinted in [BREG-I] is optimizing the architecture of a neural network via sparsity. In the example provided in [BREG-I, Fig. 4] one defines a super-architecture as a multi-layer perceptron with an equal number of neurons in each layer. Training this task with the LinBreg algorithm reveals the well-known autoencoder structure [HZ93]. This idea was developed further in [BREG-II], where also skip connection of a residual architecture were learned. Here, the super-architecture was given by a dense net [Hua+17], where each skip connection was scaled by a parameter, which was penalized with a sparsity term. We refer to [CFS23] for similar experimental result in a different setting, not directly related to sparse optimization.

The driving question for future work, is how to employ the insights from [BREG-II] to find more complex architectures. Learning an architecture similar to the U-Net as proposed in [RFB15] is of particular interest here.

Chapter 5
Conclusion

This thesis provided insights into the topics of consistency, sparsity and robustness of learning algorithms. Thematically, the thesis is split into two main chapters, dealing with semi-supervised and supervised learning. We summarize both chapters below and respectively provide an outlook and possible future work.

Consistency of SSL on Sparse Graphs We considered the infinite data limit in the semi-supervised setting, focusing on the Lipschitz learning task. For all our results we assumed mild scaling conditions which allow for very sparse graphs. We were first able to show Γ-convergence in the variational setting. We then proved convergence rates for AMLEs via a homogenization strategy, which relied on the comparison with cones principle. The key insight we obtained was, that a rate for graph distance functions implies a rate for graph AMLEs. This observation was already employed in our follow-up work in [LIP-III], where convergence rates at an even smaller scale were shown. We also conducted experiments to validate our theoretical framework in practice. Here, we observed better results than we were able to prove.

While our framework allows the graph to be very sparse, we are restricted to the case of ε-ball graphs. Here, it would be interesting to see, how our results can be transferred to the knn setting as in [CT22]. Furthermore, we only considered the standard Lipschitz learning task, which tends to forget the data distribution. In [Cal19] a modification was proposed that makes the problem sensitive to the distribution. The open problem that arises here is how to adapt the technique in [LIP-II] to show convergence for the modified problem.

Robust and Sparse SL In the supervised setting, we considered input-robustness w.r.t. adversarial perturbations and resolution changes. In the first case, we proposed a defense mechanism to train stable neural networks, based on Lipschitz regularization. We provided analytical results and numerical experiments that suggest that the strategy allows us to learn robust networks. The strength of this approach depends on how well the discrete Lipschitz constant approximates the true one. An interesting future direction would be to explore different methods to determine the discrete set of Lipschitz pairs.

E.g., one could additionally learn a generative model that outputs these pair in a faster and probably even more expressive way than the gradient descent scheme, employed right now.

For the multi-resolution setting, we analyzed the role of Fourier neural operators. We first showed certain functional analytic properties of neural layers acting on L^p spaces. We then established the relation of discretized Fourier layers to standard convolution operations, both from a theoretical and a practical side. We also highlighted the importance of the trigonometric interpolation in this context. We conducted numerical tests that suggest the resolution equivariant behavior of FNO layers. Connected to the previous topic, it would be interesting to study the adversarial robustness of FNOs. First numerical tests in [Kab22] hint that the vulnerabilities are even worse than in the standard case.

We considered the question of how to enforce sparsity in neural network weights. Here, we employed a stochastic variant of Bregman iterations, which allowed us to train networks that are very sparse throughout the whole optimization. We provided theoretic convergence guarantees, where the main novelty was the stochasticity introduced into the iteration. We also showed the numerical efficiency of the method, by training sparse neural networks performing similarly to their dense counterparts. One open problem is to weaken the convexity assumption and instead prove the convergence result, by employing a Kurdyka–Łojasiewicz type inequality as in [Ben+21]. Concerning the neural architecture search via sparsity as proposed in our work, a further interesting task would be to learn the U-Net type architecture of [RFB15].

Bibliography

Books

[AF03] R. A. Adams and J. J. Fournier. *Sobolev spaces.* Elsevier, 2003.

[AP93] A. Ambrosetti and G. Prodi. *A Primer of Nonlinear Analysis.* Cambridge University Press, 1993.

[BC11] H. Bauschke and P. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces.* New York: Springer, 2011.

[BV04] S. P. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge university press, 2004.

[Bra02] A. Braides. *Gamma-convergence for Beginners.* Vol. 22. Oxford University Press, Oxford, 2002.

[Bre+84] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classificaion and regression trees.* CRC Press, New York, 1984.

[BB11] H. Brezis and H. Brézis. *Functional analysis, Sobolev spaces and partial differential equations.* Vol. 2. 3. Springer, 2011.

[COR98] G. Cybenko, D. P. O'Leary, and J. Rissanen. *The mathematics of information coding, extraction and distribution.* Vol. 107. Springer Science & Business Media, 1998.

[Dac07] B. Dacorogna. *Direct methods in the calculus of variations.* Vol. 78. Springer Science & Business Media, 2007.

[Dal12] G. Dal Maso. *An introduction to* Γ*-convergence.* Vol. 8. Springer Science & Business Media, 2012.

[DGL13] L. Devroye, L. Györfi, and G. Lugosi. *A probabilistic theory of pattern recognition.* Vol. 31. Springer Science & Business Media, 2013.

[DS88] N. Dunford and J. T. Schwartz. *Linear operators, part 1: general theory.* Vol. 10. John Wiley & Sons, 1988.

[Eul24] L. Euler. *Institutionum calculi integralis.* Vol. 1. impensis Academiae imperialis scientiarum, 1824.

[Eva18] L. Evans. *Measure theory and fine properties of functions.* Routledge, 2018.

[GV13] G. H. Golub and C. F. Van Loan. *Matrix computations.* JHU press, 2013.

[GW87] R. C. Gonzales and P. Wintz. *Digital image processing.* Addison-Wesley Longman Publishing Co., Inc., 1987.

[GBC16] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[Gra14] L. Grafakos. *Classical Fourier Analysis*. 3rd ed. Graduate Texts in Mathematics. Springer, New York, NY, 2014.

[Hau14] F. Hausdorff. *Grundzüge der Mengenlehre*. Viet, Leipzig, 1914.

[Lin16] P. Lindqvist. *Notes on the infinity Laplace equation*. Springer, 2016.

[Lin17] P. Lindqvist. *Notes on the p-Laplace equation*. 161. University of Jyväskylä, 2017.

[Lip77] R. Lipschitz. *Lehrbuch der analysis*. Vol. 1. M. Cohen & Sohn (F. Cohen), 1877.

[Ral81] L. B. Rall. *Automatic differentiation: Techniques and applications*. Springer, 1981.

[Roc97] R. Rockafellar. *Convex analysis*. Princeton, N.J: Princeton University Press, 1997.

[Ros62] F. Rosenblatt. *Principles of neurodynamics: Perceptrons and the theory of brain mechanisms*. Vol. 55. Spartan books Washington, DC, 1962.

[Sch69] J. T. Schwartz. *Nonlinear Functional Analysis*. Boca Raton, Fla: CRC Press, 1969.

[SB14] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[Trö10] F. Tröltzsch. *Optimal Control of Partial Differential Equations: Theory, Methods, and Applications*. Vol. 112. Graduate Studies in Mathematics. American Mathematical Society, Providence, Rhode Island, 2010.

[VD95] G. Van Rossum and F. L. Drake Jr. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[Zhu05] X. Zhu. *Semi-supervised learning with graphs*. Carnegie Mellon University, 2005.

Articles and Online

[ANR74] N. Ahmed, T. Natarajan, and K. R. Rao. “Discrete cosine transform.” In: *IEEE transactions on Computers* 100.1 (1974), pp. 90–93.

[AL11] M. Alamgir and U. Luxburg. “Phase transition in the family of p-resistances.” In: *Advances in neural information processing systems* 24 (2011).

[ALG19] C. Anil, J. Lucas, and R. B. Grosse. “Sorting Out Lipschitz Function Approximation.” In: *ICML*. Vol. 97. PMLR. 2019, pp. 291–301.

[ACB17] M. Arjovsky, S. Chintala, and L. Bottou. “Wasserstein generative adversarial networks.” In: *International conference on machine learning*. PMLR. 2017, pp. 214–223.

[ASS11] 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.

[AS10] 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 (2010), pp. 381–384.

[Arn+23] C. Arndt, A. Denker, S. Dittmer, N. Heilenkötter, M. Iske, T. Kluth, P. Maass, and J. Nickel. *Invertible residual networks in the context of regularization theory for linear inverse problems*. 2023. arXiv: 2306.01335.

[Aro67] G. Aronsson. “Extension of functions satisfying Lipschitz conditions.” In: *Arkiv för Matematik* 6.6 (1967), pp. 551–561.

[Aro68] G. 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 (1968), pp. 395–425.

[ACJ04] 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.

[AKL22] P.-C. Aubin-Frankowski, A. Korba, and F. Léger. “Mirror descent with relative smoothness in measure spaces, with application to Sinkhorn and EM.” In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 17263–17275.

[Ban+18] R. Banner, I. Hubara, E. Hoffer, and D. Soudry. “Scalable methods for 8-bit training of neural networks.” In: *Advances in neural information processing systems* 31 (2018).

[BJW01] E. N. Barron, R. R. Jensen, and C. Y. Wang. “Lower semicontinuity of L^∞ functionals.” In: *Annales de l’Institut Henri Poincaré C, Analyse non linéaire*. Vol. 18. 4. Elsevier. 2001, pp. 495–517.

[BT03] A. Beck and M. Teboulle. “Mirror descent and nonlinear projected subgradient methods for convex optimization.” In: *Operations Research Letters* 31.3 (2003), pp. 167–175.

[Ben+21] M. Benning, M. M. Betcke, M. J. Ehrhardt, and C.-B. Schönlieb. “Choose your path wisely: gradient descent in a Bregman distance framework.” In: *SIAM Journal on Imaging Sciences* 14.2 (2021), pp. 814–843.

[BB18] M. Benning and M. Burger. “Modern regularization methods for inverse problems.” In: *Acta Numerica* 27 (2018), pp. 1–111.

[BDM89] T. Bhattacharya, E. DiBenedetto, and J. Manfredi. “Limits as $p \to \infty$ of $\Delta_p u_p = f$ and related extremal problems.” In: *Rend. Sem. Mat. Univ. Politec. Torino* 47 (1989), pp. 15–68.

[BT06] T. Birsan and D. Tiba. “One hundred years since the introduction of the set distance by Dimitrie Pompeiu.” In: *System Modeling and Optimization: Proceedings of the 22nd IFIP TC7 Conference held from July 18–22, 2005, in Turin, Italy 22*. Springer. 2006, pp. 35–39.

[BM98] A. Blum and T. Mitchell. “Combining labeled and unlabeled data with co-training.” In: *Proceedings of the eleventh annual conference on Computational learning theory*. 1998, pp. 92–100.

[Bol68] L. Boltzmann. “Studien über das Gleichgewicht der lebenden Kraft.” In: *Wissenschafiliche Abhandlungen* 1 (1868), pp. 49–96.

[BY12] A. Braides and N. K. Yip. “A quantitative description of mesh dependence for the discretization of singularly perturbed nonconvex problems.” In: *SIAM Journal on Numerical Analysis* 50.4 (2012), pp. 1883–1898.

[Bre67] L. M. Bregman. “The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming.” In: *USSR computational mathematics and mathematical physics* 7.3 (1967), pp. 200–217.

[Bri19] T. Briand. “Trigonometric Polynomial Interpolation of Images.” In: *Image Processing On Line* 9 (2019), pp. 291–316.

[Bri90] J. S. Bridle. “Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition.” In: *Neurocomputing: Algorithms, architectures and applications*. Springer, 1990, pp. 227–236.

[Bun23] L. Bungert. “The inhomogeneous p-Laplacian equation with Neumann boundary conditions in the limit $p \to \infty$.” In: *Advances in Continuous and Discrete Models* 2023.1 (2023), pp. 1–17.

[LIP-III] L. Bungert, J. Calder, and T. Roith. *Ratio convergence rates for Euclidean first-passage percolation: Applications to the graph infinity Laplacian*. 2022. arXiv: 2210.09023.

[LIP-II] L. Bungert, J. Calder, and T. Roith. “Uniform convergence rates for Lipschitz learning on graphs.” In: *IMA Journal of Numerical Analysis* 43.4 (2022), pp. 2445–2495. DOI: 10.1093/imanum/drac048.

[BGM23] L. Bungert, N. García Trillos, and R. Murray. “The geometry of adversarial training in binary classification.” In: *Information and Inference: A Journal of the IMA* 12.2 (2023), pp. 921–968.

[BKB20] L. Bungert, Y. Korolev, and M. Burger. “Structural analysis of an L-infinity variational problem and relations to distance functions.” In: *Pure and Applied Analysis* 2.3 (2020), pp. 703–738.

[CLIP] L. Bungert, R. Raab, T. Roith, L. Schwinn, and D. Tenbrinck. “CLIP: Cheap Lipschitz training of neural networks.” In: *Scale Space and Variational Methods in Computer Vision: 8th International Conference, SSVM 2021, Proceedings*. Springer. 2021, pp. 307–319. DOI: 10.1007/978-3-030-75549-2_25.

[BREG-I] L. Bungert, T. Roith, D. Tenbrinck, and M. Burger. “A Bregman learning framework for sparse neural networks.” In: *Journal of Machine Learning Research* 23.192 (2022), pp. 1–43.

[BREG-II] L. Bungert, T. Roith, D. Tenbrinck, and M. Burger. *Neural Architecture Search via Bregman Iterations*. 2021. arXiv: 2106.02479.

[Bun+23] L. Bungert, N. G. Trillos, M. Jacobs, D. McKenzie, Đ. Nikolić, and Q. Wang. *It begins with a boundary: A geometric view on probabilistically robust learning*. 2023. arXiv: 2305.18779.

[Bur+07] M. Burger, K. Frick, S. Osher, and O. Scherzer. “Inverse total variation flow.” In: *Multiscale Modeling & Simulation* 6.2 (2007), pp. 366–395.

[Bur+06] M. Burger, G. Gilboa, S. Osher, J. Xu, et al. “Nonlinear inverse scale space methods.” In: *Communications in Mathematical Sciences* 4.1 (2006), pp. 179–212.

[BO13] M. Burger and S. Osher. “A guide to the TV zoo.” In: *Level set and PDE based reconstruction methods in imaging*. Springer, 2013, pp. 1–70.

[Bur48] J. M. Burgers. “A mathematical model illustrating the theory of turbulence.” In: *Advances in applied mechanics* 1 (1948), pp. 171–199.

[COS09] J.-F. Cai, S. Osher, and Z. Shen. “Convergence of the linearized Bregman iteration for ℓ_1-norm minimization.” In: *Mathematics of Computation* 78.268 (2009), pp. 2127–2136.

[Cal19] J. 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.

[Cal+20] J. Calder, B. Cook, M. Thorpe, and D. Slepčev. “Poisson learning: Graph based semi-supervised learning at very low label rates.” In: *International Conference on Machine Learning*. PMLR. 2020, pp. 1306–1316.

[CS20] J. Calder and D. Slepčev. “Properly-weighted graph Laplacian for semi-supervised learning.” In: *Applied mathematics & optimization* 82 (2020), pp. 1111–1159.

[CT22] J. Calder and N. G. 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.

[Car+21] J. A. Carrillo, S. Jin, L. Li, and Y. Zhu. “A consensus-based global optimization method for high dimensional machine learning problems.” In: *ESAIM: Control, Optimisation and Calculus of Variations* 27 (2021), S5.

[CFP97] G. Castellano, A. M. Fanelli, and M. Pelillo. “An iterative pruning algorithm for feedforward neural networks.” In: *IEEE transactions on Neural networks* 8.3 (1997), pp. 519–531.

[Cau+47] A. Cauchy et al. “Méthode générale pour la résolution des systemes d’équations simultanées.” In: *Comp. Rend. Sci. Paris* 25.1847 (1847), pp. 536–538.

[ČFK66] E. Čech, Z. Frolík, and M. Katětov. “Topological spaces.” In: (1966).

[Cha+21] J. Chai, H. Zeng, A. Li, and E. W. Ngai. “Deep learning in computer vision: A critical review of emerging techniques and application scenarios.” In: *Machine Learning with Applications* 6 (2021), p. 100134.

[Cha04] A. Chambolle. “An algorithm for total variation minimization and applications.” In: *Journal of Mathematical imaging and vision* 20 (2004), pp. 89–97.

[CGL10] A. Chambolle, A. Giacomini, and L. Lussardi. “Continuous limits of discrete perimeters.” In: *ESAIM: Mathematical Modelling and Numerical Analysis* 44.2 (2010), pp. 207–230.

[CP11] A. Chambolle and T. Pock. “A first-order primal-dual algorithm for convex problems with applications to imaging.” In: *Journal of mathematical imaging and vision* 40 (2011), pp. 120–145.

[CFS23] C. Cipriani, M. Fornasier, and A. Scagliotti. *From NeurODEs to AutoencODEs: a mean-field control framework for width-varying Neural Networks.* 2023. eprint: 2307.02279.

[CM73] J. F. Claerbout and F. Muir. “Robust modeling with erratic data.” In: *Geophysics* 38.5 (1973), pp. 826–844.

[CP08] P. L. Combettes and J.-C. Pesquet. “Proximal thresholding algorithm for minimization over orthonormal bases.” In: *SIAM Journal on Optimization* 18.4 (2008), pp. 1351–1376.

[CV95] C. Cortes and V. Vapnik. “Support-vector networks.” In: *Machine learning* 20 (1995), pp. 273–297.

[CBD18] M. Courbariaux, Y. Bengio, and J.-P. David. “Training deep neural networks with low precision multiplications.” In: *Journal of Machine Learning Research* 18 (2018), pp. 1–30.

[CCC+03] F. G. Cozman, I. Cohen, M. C. Cirelo, et al. “Semi-supervised learning of mixture models.” In: *ICML.* Vol. 4. 2003, p. 24.

[DOr+21] R. D’Orazio, N. Loizou, I. Laradji, and I. Mitliagkas. *Stochastic mirror descent: Convergence analysis and adaptive variants via the mirror stochastic Polyak stepsize.* 2021. arXiv: 2110.15412.

[DYJ19] X. Dai, H. Yin, and N. K. Jha. “NeST: A neural network synthesis tool based on a grow-and-prune paradigm.” In: *IEEE Transactions on Computers* 68.10 (2019), pp. 1487–1497.

[DDD04] I. Daubechies, M. Defrise, and C. De Mol. “An iterative thresholding algorithm for linear inverse problems with a sparsity constraint.” In: *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 57.11 (2004), pp. 1413–1457.

[De 93] E. De Giorgi. “New problems on minimizing movements.” In: *Ennio de Giorgi: Selected Papers* (1993), pp. 699–713.

[DF75] E. De Giorgi and T. Franzoni. “Su un tipo di convergenza variazionale.” In: *Atti della Accademia Nazionale dei Lincei. Classe di Scienze Fisiche, Matematiche e Naturali. Rendiconti* 58.6 (1975), pp. 842–850.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. “Maximum likelihood from incomplete data via the EM algorithm.” In: *Journal of the royal statistical society: series B (methodological)* 39.1 (1977), pp. 1–22.

[DZ19] T. Dettmers and L. Zettlemoyer. *Sparse networks from scratch: Faster training without losing performance.* 2019. arXiv: 1907.04840.

[Dij22] E. W. Dijkstra. “A note on two problems in connexion with graphs.” In: *Edsger Wybe Dijkstra: His Life, Work, and Legacy.* 2022, pp. 287–290.

[DEH21] R. A. Dragomir, M. Even, and H. Hendrikx. “Fast stochastic Bregman gradient methods: Sharp analysis and variance reduction.” In: *International Conference on Machine Learning.* PMLR. 2021, pp. 2815–2825.

[Dun+20] M. M. Dunlop, D. Slepčev, A. M. Stuart, and M. Thorpe. “Large data and zero noise limits of graph-based semi-supervised learning algorithms.” In: *Applied and Computational Harmonic Analysis* 49.2 (2020), pp. 655–697.

[El +16] A. El Alaoui, X. Cheng, A. Ramdas, M. J. Wainwright, and M. I. Jordan. “Asymptotic behavior of\ell_p-based laplacian regularization in semi-supervised learning.” In: *Conference on Learning Theory.* PMLR. 2016, pp. 879–906.

[ETT15] 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.

[EMH19] T. Elsken, J. H. Metzen, and F. Hutter. “Neural architecture search: A survey.” In: *The Journal of Machine Learning Research* 20.1 (2019), pp. 1997–2017.

[Eng+18] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, and A. Madry. “A rotation and a translation suffice: Fooling CNNs with simple transformations.” In: (2018).

[Evc+20] 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.* Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 2943–2952.

[FS06] J. M. Fadili and J.-L. Starck. "Sparse representation-based image deconvolution by iterative thresholding." In: *Astronomical Data Analysis ADA'06.* 2006.

[Faw+22] M. Faw, I. Tziotis, C. Caramanis, A. Mokhtari, S. Shakkottai, and R. Ward. "The power of adaptivity in sgd: Self-tuning step sizes with unbounded gradients and affine variance." In: *Conference on Learning Theory.* PMLR. 2022, pp. 313–355.

[FFF18] A. Fawzi, H. Fawzi, and O. Fawzi. "Adversarial vulnerability for any classifier." In: *Advances in neural information processing systems* 31 (2018).

[FNW07] M. A. Figueiredo, R. D. Nowak, and S. J. Wright. "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems." In: *IEEE Journal of selected topics in signal processing* 1.4 (2007), pp. 586–597.

[FCL22] 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.

[Fré06] M. M. Fréchet. "Sur quelques points du calcul fonctionnel." In: *Rendiconti del Circolo Matematico di Palermo (1884-1940)* 22.1 (1906), pp. 1–72.

[Fu+22] 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." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[Fuk80] K. Fukushima. "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position." In: *Biological cybernetics* 36.4 (1980), pp. 193–202.

[GMT22] N. García Trillos, R. Murray, and M. Thorpe. "From graph cuts to isoperimetric inequalities: Convergence rates of Cheeger cuts on data clouds." In: *Archive for Rational Mechanics and Analysis* 244.3 (2022), pp. 541–598.

[GS18] N. García Trillos and D. Slepčev. "A variational approach to the consistency of spectral clustering." In: *Applied and Computational Harmonic Analysis* 45.2 (2018), pp. 239–281.

[GS15] 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.

[Gar+16] N. García Trillos, D. Slepčev, J. Von Brecht, T. Laurent, and X. Bresson. "Consistency of Cheeger and ratio graph cuts." In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 6268–6313.

[Gho+22] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer. "A survey of quantization methods for efficient neural network inference." In: *Low-Power Computer Vision.* Chapman and Hall/CRC, 2022, pp. 291–326.

[GK06] E. Giné and V. Koltchinskii. “Empirical graph Laplacian approximation of Laplace-Beltrami operators: large sample results.” In: *Lecture Notes-Monograph Series* (2006), pp. 238–259.

[GB10] 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*. JMLR Workshop and Conference Proceedings. 2010, pp. 249–256.

[GO09] T. Goldstein and S. Osher. “The split Bregman method for L1-regularized problems.” In: *SIAM journal on imaging sciences* 2.2 (2009), pp. 323–343.

[Goo52] I. J. Good. “Rational decisions.” In: *Journal of the Royal Statistical Society: Series B (Methodological)* 14.1 (1952), pp. 107–114.

[GSS14] I. J. Goodfellow, J. Shlens, and C. Szegedy. *Explaining and harnessing adversarial examples*. 2014. arXiv: 1412.6572.

[Gou+20] H. Gouk, E. Frank, B. Pfahringer, and M. J. Cree. “Regularisation of neural networks by enforcing Lipschitz continuity.” In: *Machine Learning* (2020), pp. 1–24.

[Guo+18] C. Guo, M. Rana, M. Cisse, and L. van der Maaten. “Countering Adversarial Images using Input Transformations.” In: *International Conference on Learning Representations*. 2018.

[HR21] F. Hanzely and P. Richtárik. “Fastest rates for stochastic mirror descent methods.” In: *Computational Optimization and Applications* 79 (2021), pp. 717–766.

[Has+20] M. Hasannasab, J. Hertrich, S. Neumayer, G. Plonka, S. Setzer, and G. Steidl. “Parseval proximal neural networks.” In: *Journal of Fourier Analysis and Applications* 26 (2020), pp. 1–31.

[HSW93] B. Hassibi, D. G. Stork, and G. J. Wolff. “Optimal brain surgeon and general network pruning.” In: *IEEE international conference on neural networks*. IEEE. 1993, pp. 293–299.

[He+16a] K. He, X. Zhang, S. Ren, and J. Sun. “Deep residual learning for image recognition.” In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[He+16b] K. He, X. Zhang, S. Ren, and J. Sun. “Identity mappings in deep residual networks.” In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*. Springer. 2016, pp. 630–645.

[HAV05] M. Hein, J.-Y. Audibert, and U. Von Luxburg. “From graphs to manifolds–weak and strong pointwise consistency of graph Laplacians.” In: *International Conference on Computational Learning Theory*. Springer. 2005, pp. 470–485.

[HG16] D. Hendrycks and K. Gimpel. *Gaussian Error Linear Units (GELUs)*. 2016. arXiv: 1606.08415.

[HVD15] G. Hinton, O. Vinyals, and J. Dean. *Distilling the knowledge in a neural network*. 2015. arXiv: 1503.02531.

[HZ93] G. E. Hinton and R. Zemel. "Autoencoders, minimum description length and Helmholtz free energy." In: *Advances in neural information processing systems* 6 (1993).

[Hoe+21] 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." In: *J. Mach. Learn. Res.* 22.241 (2021), pp. 1–124.

[Hof+22] F. Hoffmann, B. Hosseini, A. A. Oberai, and A. M. Stuart. "Spectral analysis of weighted Laplacians arising in data clustering." In: *Applied and Computational Harmonic Analysis* 56 (2022), pp. 189–249.

[Hof+20] F. Hoffmann, B. Hosseini, Z. Ren, and A. M. Stuart. "Consistency of semi-supervised learning algorithms on graphs: Probit and one-hot methods." In: *The Journal of Machine Learning Research* 21.1 (2020), pp. 7549–7603.

[How+17] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. *Mobilenets: Efficient convolutional neural networks for mobile vision applications*. 2017. arXiv: 1704.04861.

[Hua+16] 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*. 2016, pp. 3377–3385.

[Hua+17] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. "Densely connected convolutional networks." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.

[HW62] D. H. Hubel and T. N. Wiesel. "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex." In: *The Journal of physiology* 160.1 (1962), p. 106.

[Ily+18] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin. "Black-box adversarial attacks with limited queries and information." In: *International conference on machine learning*. PMLR. 2018, pp. 2137–2146.

[Jen93] R. Jensen. "Uniqueness of Lipschitz extensions: minimizing the sup norm of the gradient." In: *Archive for Rational Mechanics and Analysis* 123 (1993), pp. 51–74.

[Juu02] 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.

[JS06] 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.

[FNO] S. Kabri, T. Roith, D. Tenbrinck, and M. Burger. “Resolution-Invariant Image Classification based on Fourier Neural Operators.” In: *Scale Space and Variational Methods in Computer Vision: 9th International Conference, SSVM 2023, Proceedings*. Springer. 2023, pp. 307–319. DOI: 10.1007/978-3-031-31975-4_18.

[Kel60] H. J. Kelley. “Gradient theory of optimal flight paths.” In: *Ars Journal* 30.10 (1960), pp. 947–954.

[Khu+23] D. Khurana, A. Koli, K. Khatter, and S. Singh. “Natural language processing: State of the art, current trends and challenges.” In: *Multimedia tools and applications* 82.3 (2023), pp. 3713–3744.

[KB14] D. Kingma and J. Ba. *Adam: A Method for Stochastic Optimization.* 2014. arXiv: 1412.6980.

[Kir34] M. Kirszbraun. “Über die zusammenziehende und Lipschitzsche Transformationen.” In: *Fundamenta Mathematicae* 22 (1934), pp. 77–108.

[KLM21] N. Kovachki, S. Lanthaler, and S. Mishra. “On universal approximation and error bounds for Fourier neural operators.” In: *The Journal of Machine Learning Research* 22.1 (2021), pp. 13237–13312.

[Kov+23] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar. “Neural Operator: Learning Maps Between Function Spaces With Applications to PDEs.” In: *Journal of Machine Learning Research* 24.89 (2023), pp. 1–97.

[Kri+20] V. Krishnan, A. Makdah, A. AlRahman, and F. Pasqualetti. “Lipschitz bounds and provably robust training by laplacian smoothing.” In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 10924–10935.

[KGB17] A. Kurakin, I. J. Goodfellow, and S. Bengio. “Adversarial Machine Learning at Scale.” In: *International Conference on Learning Representations*. 2017.

[Kur22] C. Kuratowski. “Sur l’opération A de l’analysis situs.” In: *Fundamenta Mathematicae* 3.1 (1922), pp. 182–199.

[Kyn+15] R. Kyng, A. Rao, S. Sachdeva, and D. A. Spielman. “Algorithms for Lipschitz learning on graphs.” In: *Conference on Learning Theory*. PMLR. 2015, pp. 1190–1223.

[Lan52] C. Lanczos. “Solution of systems of linear equations by minimized iterations.” In: *J. Res. Nat. Bur. Standards* 49.1 (1952), pp. 33–53.

[LC10] Y. LeCun and C. Cortes. “MNIST handwritten digit database.” In: (2010).

[LDS89] Y. LeCun, J. Denker, and S. Solla. “Optimal brain damage.” In: *Advances in neural information processing systems* 2 (1989).

[Li+21] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. “Fourier Neural Operator for Parametric Partial Differential Equations.” In: *International Conference on Learning Representations*. 2021.

[Liu+21] 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.” In: *Neural Computing and Applications* 33.7 (2021), pp. 2589–2604.

[MDA15] D. Maclaurin, D. Duvenaud, and R. P. Adams. “Autograd: Effortless gradients in numpy.” In: *ICML 2015 AutoML workshop*. Vol. 238. 5. 2015.

[Mad+18] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. “Towards Deep Learning Models Resistant to Adversarial Attacks.” In: *International Conference on Learning Representations*. 2018.

[Mar10] J. Martens. “Deep learning via Hessian-free optimization.” In: *ICML*. Vol. 27. 2010, pp. 735–742.

[McS34] E. J. McShane. “Extension of range of functions.” In: (1934).

[MP69] M. Minsky and S. Papert. “An introduction to computational geometry.” In: *Cambridge tiass., HIT* 479 (1969), p. 480.

[Moc+18] 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.” In: *Nature communications* 9.1 (2018), pp. 1–12.

[Mod77] L. Modica. “Un esempio di Γ-convergenza.” In: *Boll. Un. Mat. Ital. B* 14 (1977), pp. 285–299.

[MS63] J. N. Morgan and J. A. Sonquist. “Problems in the analysis of survey data, and a proposal.” In: *Journal of the American statistical association* 58.302 (1963), pp. 415–434.

[NSZ09] B. Nadler, N. Srebro, and X. Zhou. “Statistical analysis of semi-supervised learning: The limit of infinite unlabelled data.” In: *Advances in neural information processing systems* 22 (2009).

[NS12] A. Naor and S. Sheffield. “Absolutely minimal Lipschitz extension of tree-valued mappings.” In: *Mathematische Annalen* 354 (2012), pp. 1049–1078.

[Nem+09] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. “Robust stochastic approximation approach to stochastic programming.” In: *SIAM Journal on optimization* 19.4 (2009), pp. 1574–1609.

[NY83] A. S. Nemirovskij and D. B. Yudin. “Problem complexity and method efficiency in optimization.” In: (1983).

[Nes83] Y. Nesterov. “A method for unconstrained convex minimization problem with the rate of convergence o (1/k2).” In: *Doklady ANSSSR* 269.3 (1983), pp. 543–547.

[Ngu+18] 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: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by J. Dy and A. Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 3750–3758.

[Nit14] A. Nitanda. "Stochastic proximal gradient descent with acceleration techniques." In: *Advances in Neural Information Processing Systems* 27 (2014), pp. 1574–1582.

[Osh+05] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. "An iterative regularization method for total variation-based image restoration." In: *Multiscale Modeling & Simulation* 4.2 (2005), pp. 460–489.

[PB+14] N. Parikh, S. Boyd, et al. "Proximal algorithms." In: *Foundations and trends® in Optimization* 1.3 (2014), pp. 127–239.

[Pas+19] A. Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library." In: *Advances in Neural Information Processing Systems*. 2019, pp. 8024–8035.

[Pen99a] M. D. Penrose. "A strong law for the largest nearest-neighbour link between random points." In: *Journal of the london mathematical society* 60.3 (1999), pp. 951–960.

[Pen99b] M. D. Penrose. "A strong law for the longest edge of the minimal spanning tree." In: *The Annals of Probability* 27.1 (1999), pp. 246–260.

[Per+09] 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.

[Per23] O. Perron. "Eine neue Behandlung der ersten Randwertaufgabe für Δu= 0." In: *Mathematische Zeitschrift* 18 (1923), pp. 42–54.

[Pin+17] R. Pinnau, C. Totzeck, O. Tse, and S. Martin. "A consensus-based model for global optimization and its mean-field limit." In: *Mathematical Models and Methods in Applied Sciences* 27.01 (2017), pp. 183–204.

[Pio21] G. Piosenka. *BIRDS 500 - SPECIES IMAGE CLASSIFICATION*. 2021. URL: https://www.kaggle.com/datasets/gpiosenka/100-bird-species.

[Pom05] D. Pompeiu. "Sur la continuité des fonctions de variables complexes." In: *Annales de la Faculté des sciences de Toulouse: Mathématiques*. Vol. 7. 3. 1905, pp. 265–315.

[Qia99] N. Qian. "On the momentum term in gradient descent learning algorithms." In: *Neural networks* 12.1 (1999), pp. 145–151.

[Red+16] 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*. 2016, pp. 1153–1161.

[Rie23] K. Riedl. "Leveraging memory effects and gradient information in consensus-based optimisation: On global convergence in mean-field law." In: *European Journal of Applied Mathematics* (2023), pp. 1–32.

[RM51] H. Robbins and S. Monro. "A stochastic approximation method." In: *The annals of mathematical statistics* (1951), pp. 400–407.

[LIP-I] T. Roith and L. Bungert. "Continuum limit of Lipschitz learning on graphs." In: *Foundations of Computational Mathematics* 23.2 (2023), pp. 1–39. DOI: 10.1007/s10208-022-09557-9.

[RFB15] 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*. Springer. 2015, pp. 234–241.

[RVV20] L. Rosasco, S. Villa, and B. C. Vũ. "Convergence of stochastic proximal gradient algorithm." In: *Applied Mathematics & Optimization* 82.3 (2020), pp. 891–917.

[Ros58] F. Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychological review* 65.6 (1958), p. 386.

[RKH19] K. Roth, Y. Kilcher, and T. Hofmann. "Adversarial Training is a Form of Data-dependent Operator Norm Regularization." In: *NeurIPS*. 2019.

[ROF92] L. I. Rudin, S. Osher, and E. Fatemi. "Nonlinear total variation based noise removal algorithms." In: *Physica D: nonlinear phenomena* 60.1-4 (1992), pp. 259–268.

[RHW86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. "Learning representations by back-propagating errors." In: *nature* 323.6088 (1986), pp. 533–536.

[Sam59] A. L. Samuel. "Some studies in machine learning using the game of checkers." In: *IBM Journal of research and development* 3.3 (1959), pp. 210–229.

[San15] F. Santambrogio. "Optimal transport for applied mathematicians." In: *Birkäuser, NY* 55.58-63 (2015), p. 94.

[SV18] K. Scaman and A. Virmaux. "Lipschitz Regularity of Deep Neural Networks: Analysis and Efficient Estimation." In: *NeurIPS*. 2018.

[Sca+17] S. Scardapane, D. Comminiello, A. Hussain, and A. Uncini. "Group sparse regularization for deep neural networks." In: *Neurocomputing* 241 (2017), pp. 81–89.

[Sch22] J. Schmidhuber. *Annotated history of modern AI and Deep learning*. 2022. arXiv: 2212.11279.

[Sch15] J. Schmidhuber. “Deep learning in neural networks: An overview.” In: *Neural Networks* 61 (2015), pp. 85–117.

[Sch92] J. Schmidhuber. “Learning complex, extended sequences using the principle of history compression.” In: *Neural Computation* 4.2 (1992), pp. 234–242.

[SS05] B. Schölkopf and A. Smola. “Support vector machines and kernel algorithms.” In: *Encyclopedia of Biostatistics*. Wiley, 2005, pp. 5328–5335.

[Sha+19a] A. Shafahi, W. R. Huang, C. Studer, S. Feizi, and T. Goldstein. “Are adversarial examples inevitable?” In: *International Conference on Learning Representations*. 2019.

[Sha+19b] A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein. “Adversarial training for free!” In: *Advances in Neural Information Processing Systems* 32 (2019).

[She+22] M. Shehab, L. Abualigah, Q. Shambour, M. A. Abu-Hashem, M. K. Y. Shambour, A. I. Alsalibi, and A. H. Gandomi. “Machine learning in medical applications: A review of state-of-the-art methods.” In: *Computers in Biology and Medicine* 145 (2022), p. 105458.

[ST19] 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.

[SGS15] R. K. Srivastava, K. Greff, and J. Schmidhuber. *Highway networks*. 2015. arXiv: 1505.00387.

[Sta+21] J. Stanczuk, C. Etmann, L. M. Kreusser, and C.-B. Schönlieb. *Wasserstein GANs work because they fail (to approximate the Wasserstein distance)*. 2021. arXiv: 2103.01678.

[SG96] G. L. Steele and R. P. Gabriel. “The evolution of Lisp.” In: *History of programming languages—II*. 1996, pp. 233–330.

[Ste+56] H. Steinhaus et al. “Sur la division des corps matériels en parties.” In: *Bull. Acad. Polon. Sci* 1.804 (1956), p. 801.

[ST14] A. Subramanya and P. P. Talukdar. “Graph-based semi-supervised learning.” In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 8.4 (2014), pp. 1–125.

[SB09] A. Szlam and X. Bresson. “A total variation-based graph clustering algorithm for cheeger ratio cuts.” In: *UCLA Cam report* (2009), pp. 09–68.

[Tib96] R. Tibshirani. “Regression shrinkage and selection via the lasso.” In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1 (1996), pp. 267–288.

[Tik43] A. N. Tikhonov. "On the stability of inverse problems." In: *Dokl. Akad. Nauk SSSR*. Vol. 39. 1943, pp. 195–198.

[TS15] N. G. Trillos and D. Slepčev. "On the rate of convergence of empirical measures in ∞-transportation distance." In: *Canadian Journal of Mathematics* 67.6 (2015), pp. 1358–1383.

[THH21] N. G. Trillos, F. Hoffmann, and B. Hosseini. "Geometric structure of graph Laplacian embeddings." In: *The Journal of Machine Learning Research* 22.1 (2021), pp. 2934–2988.

[Tur04] A. Turing. "Intelligent Machinery (1948)." In: *The Essential Turing*. Oxford University Press, 2004.

[VB+12] Y. Van Gennip, A. L. Bertozzi, et al. "Γ-convergence of graph Ginzburg-Landau functionals." In: *Adv. Differential Equations* 17.11-12 (2012), pp. 1115–1180.

[Vil+23] S. Villa, S. Matet, B. C. Vũ, and L. Rosasco. "Implicit regularization with strongly convex bias: Stability and acceleration." In: *Analysis and Applications* 21.01 (2023), pp. 165–191.

[VBB08] U. Von Luxburg, M. Belkin, and O. Bousquet. "Consistency of spectral clustering." In: *The Annals of Statistics* (2008), pp. 555–586.

[vLB04] U. von Luxburg and O. Bousquet. "Distance-Based Classification with Lipschitz Functions." In: *J. Mach. Learn. Res.* 5.Jun (2004), pp. 669–695.

[Whi92] H. Whitney. "Analytic extensions of differentiable functions defined in closed sets." In: *Hassler Whitney Collected Papers* (1992), pp. 228–254.

[XRV17] H. Xiao, K. Rasul, and R. Vollgraf. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. 2017. arXiv: 1708.07747.

[Yin+08] W. Yin, S. Osher, D. Goldfarb, and J. Darbon. "Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing." In: *SIAM Journal on Imaging sciences* 1.1 (2008), pp. 143–168.

[Yu06] Y. Yu. "A remark on C2 infinity-harmonic functions." In: (2006).

[Yua+19] X. Yuan, P. He, Q. Zhu, and X. Li. "Adversarial examples: Attacks and defenses for deep learning." In: *IEEE transactions on neural networks and learning systems* 30.9 (2019), pp. 2805–2824.

[ZH18] S. Zhang and N. He. *On the convergence rate of stochastic mirror descent for nonsmooth nonconvex optimization*. 2018. arXiv: 1806.04781.

[ZS05] D. Zhou and B. Schölkopf. "Regularization on discrete spaces." In: *Pattern Recognition: 27th DAGM Symposium, Vienna, Austria, August 31-September 2, 2005. Proceedings 27*. Springer. 2005, pp. 361–368.

[ZGL03] X. Zhu, Z. Ghahramani, and J. D. Lafferty. "Semi-supervised learning using gaussian fields and harmonic functions." In: *Proceedings of the 20th International conference on Machine learning (ICML-03)*. 2003, pp. 912–919.

Theses

[Kab22] S. Kabri. “Fourier Neural Operators for Image Classification.” MA thesis. Friedrich-Alexander-Universität Erlangen-Nürnberg, 2022.

[Roi21] T. Roith. “Master thesis: Continuum limit of Lipschitz learning on graphs.” MA thesis. Friedrich-Alexander-Universität Erlangen-Nürnberg, 2021.

[Sma10] C. K. Smart. “On the infinity Laplacian and Hrushovski’s fusion.” PhD thesis. UC Berkeley, 2010.

Part II.

Prints

Print P1

Continuum limit of Lipschitz learning on graphs

T. Roith and L. Bungert. “Continuum limit of Lipschitz learning on graphs.” In: *Foundations of Computational Mathematics* 23.2 (2023), pp. 1–39. DOI: 10.1007/s10208-022-09557-9

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

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

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

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

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

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

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 $\text{int}(p_n^{-1}(y))$ 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 $E_{n,\text{cons}}$ to arbitrary functions $u \in L^\infty(\Omega)$ by defining with a slight abuse of notation

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

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

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

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

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

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

where the length of a curve is given by

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

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:

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

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 $d_\Omega(x, y) \geq |x - y|$, 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 $1 \le p < \infty$ 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

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

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*

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

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 $E_{n,\text{cons}}$. 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 $E_{n,\text{cons}}$ 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

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

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

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

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

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

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

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

This implies that

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

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

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

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

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

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

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

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

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

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

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

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

where each inequality can be strict.

Example 1 We consider the non-convex domain $\Omega = \{x \in \mathbb{R}^2 : \max(|x_1|, |x_2|)<1\}\backslash ([0, 1] \times [-1, 0])$ 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

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

The case $1 < p < \sqrt{2}$ shows that the liminf inequality $\liminf_{n\to\infty} \mathcal{E}_{s_n}(u_n) \geq \operatorname{Lip}(u)$ 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

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

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,

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

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

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.,

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

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 $\liminf_{n\to\infty} E_{n,\text{cons}}(u_n) < \infty$, 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 $E_{n,\text{cons}}$ as follows:

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

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

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

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.,

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

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 $\mathbf{u}_n \in L^\infty(\nu_n)$ 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)

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

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

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

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

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

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

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

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 $\mathcal{E}_{\mathrm{cons}}(u) < \infty$ 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*

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

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

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

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

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

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 $i = 1, \dots, N$, yields that

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

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 $\{x_1, \dots, x_N\} \subset \Omega$ 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

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

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 $x_{k_1}, \dots, x_{k_n}$ such that

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

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

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

where we used $k_n \le N$. 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*

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

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

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

and by definition of the parameter r_n we know that for each $i = 0, \ldots, k_n$ 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 $E_{n,\mathrm{cons}}$ and $\mathcal{E}_{\mathrm{cons}}$ 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 $\mathcal{E}_{\mathrm{cons}}$ 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 $\mathcal{E}_{\mathrm{cons}}$ 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 $\overline{\Omega}\setminus\mathcal{O}$ 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*

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

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

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

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 $\mathcal{E}_{\mathrm{cons}}(u) = 1$ one obtains from (40) that

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

From Lemma 12 we know that $\mathcal{E}_{\mathrm{cons}}(d_{\mathcal{O}}) = \|\nabla d_{\mathcal{O}}\|_{L^\infty} \leq 1$. At the same time choosing $u = d_{\mathcal{O}}$ in (40) shows that in fact $\mathcal{E}_{\mathrm{cons}}(d_{\mathcal{O}}) = \|\nabla d_{\mathcal{O}}\|_{L^\infty} = 1$ 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 $\|\nabla u\|_{L^\infty} = \mathcal{E}_{\mathrm{cons}}(u) = 1$, implies that all solutions u of (38) must fulfill

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

If $\Omega\setminus\mathcal{O}$ 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

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

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

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

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

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

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. "Uniform convergence rates for Lipschitz learning on graphs." In: *IMA Journal of Numerical Analysis* 43.4 (2022), pp. 2445–2495. DOI: 10.1093/imanum/drac048

Print P3

CLIP: Cheap Lipschitz training of neural networks

L. Bungert, R. Raab, T. Roith, L. Schwinn, and D. Tenbrinck. “CLIP: Cheap Lipschitz training of neural networks.” In: *Scale Space and Variational Methods in Computer Vision: 8th International Conference, SSVM 2021, Proceedings*. Springer. 2021, pp. 307–319. DOI: 10.1007/978-3-030-75549-2_25

Print P4

Resolution-Invariant Image Classification based on Fourier Neural Operators

S. Kabri, T. Roith, D. Tenbrinck, and M. Burger. “Resolution-Invariant Image Classification based on Fourier Neural Operators.” In: *Scale Space and Variational Methods in Computer Vision: 9th International Conference, SSVM 2023, Proceedings.* Springer. 2023, pp. 307–319. DOI: 10.1007/978-3-031-31975-4_18

Print P5

A Bregman learning framework for sparse neural networks

L. Bungert, T. Roith, D. Tenbrinck, and M. Burger. “A Bregman learning framework for sparse neural networks.” In: *Journal of Machine Learning Research* 23.192 (2022), pp. 1–43

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

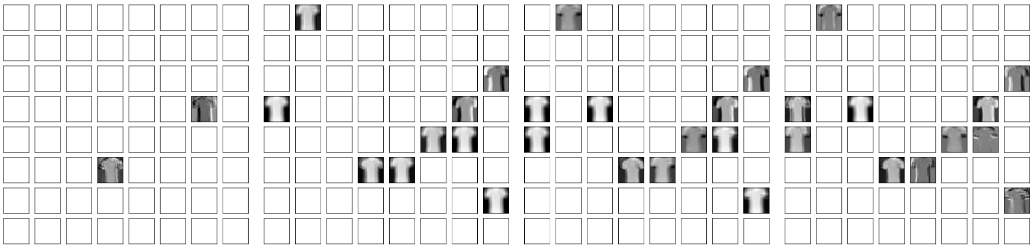
Large and deep neural networks have shown astonishing results in challenging applications, ranging from real-time image classification in autonomous driving, over assisted diagnoses in healthcare, to surpassing human intelligence in highly complex games (Amato et al., 2013; Rawat and Wang, 2017; Silver 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,   $v \leftarrow \partial J(\theta) + \frac{1}{\delta}\theta$          // 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
        $\theta \leftarrow \mathrm{prox}_{\delta J}(\delta v)$                         // 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 $l \in \{1, \ldots, L\}$. 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 $l \in \{1, \ldots, L\}$ 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

$$\begin{aligned} \theta^{(k+1)} &= \arg\min_{\theta \in \Theta} \langle \tau^{(k)} g^{(k)}, \theta \rangle + J_\delta(\theta) - \langle v^{(k)}, \theta \rangle \\ &= \arg\min_{\theta \in \Theta} \langle \tau^{(k)} g^{(k)}, \theta \rangle + J(\theta) + \frac{1}{2\delta}\|\theta\|^2 - \langle v^{(k)}, \theta \rangle \\ &= \arg\min_{\theta \in \Theta} \frac{1}{2\delta} \|\theta - \delta\left(v^{(k)} - \tau^{(k)} g^{(k)}\right)\|^2 + J(\theta) \\ &= \operatorname{prox}_{\delta J}\left(\delta\left(v^{(k)} - \tau^{(k)} g^{(k)}\right)\right). \end{aligned} \tag{2.7}$$

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

$$\begin{aligned} g^{(k)} &= \nabla \mathcal{L}(\theta^{(k)}), && \text{(2.8a)} \\ v^{(k+1)} &= v^{(k)} - \tau^{(k)} g^{(k)}, && \text{(2.8b)} \\ \theta^{(k+1)} &= \operatorname{prox}_{\delta J}\left(\delta v^{(k+1)}\right). && \text{(2.8c)} \end{aligned}$$

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

$$\begin{aligned}\ddot{v}_t &\approx \frac{v^{(k+1)} - 2v^{(k)} + v^{(k-1)}}{(\tau^{(k)})^2},\\ \dot{v}_t &\approx \frac{v^{(k+1)} - v^{(k)}}{\tau^{(k)}},\end{aligned}$$

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
        $\theta \leftarrow \mathrm{prox}_{\delta J}(\delta v)$   // 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
        $\theta \leftarrow \mathrm{prox}_{\delta J}(\delta v)$         // 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

$$D_J^p(\theta^*, \theta) = 0 \quad \textit{for } \|\theta - \theta^*\| \textit{ sufficiently small.}$$

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*

$$\lim_{k\to\infty} \tau^{(k)} = 0 \quad \textit{and} \quad \sum_{k=0}^{\infty} \tau^{(k)} = \infty \tag{3.12}$$

  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

$$\mathrm{Var}\left[W^l\right] = \alpha(n_l, n_{l-1}) \tag{4.1}$$

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

$$\begin{aligned}
\operatorname{Var}\left[W^l\right] &= \operatorname{Var}\left[\tilde{W}^l \odot M^l\right] \\
&= \mathbb{E}\left[M^l\right]^2 \operatorname{Var}\left[\tilde{W}^l\right] + \underbrace{\mathbb{E}\left[\tilde{W}^l\right]^2 \operatorname{Var}\left[M^l\right]}_{=0} + \operatorname{Var}\left[M^l\right] \operatorname{Var}\left[\tilde{W}^l\right] \\
&= \left(\mathbb{E}\left[M^l\right]^2 + \operatorname{Var}\left[M^l\right]\right) \operatorname{Var}\left[\tilde{W}^l\right] \\
&= \mathbb{E}\left[(M^l)^2\right] \operatorname{Var}\left[\tilde{W}^l\right] \\
&= r \operatorname{Var}\left[\tilde{W}^l\right]
\end{aligned}$$

and thus deriving from (4.1) we obtain the condition

$$\operatorname{Var}\left[\tilde{W}^l\right] = \frac{1}{r}\,\alpha(n_l, n_{l-1}). \tag{4.3}$$

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):

$$\mathbf{g} = \tilde{\mathbf{g}} \cdot m, \quad m \sim \mathcal{B}(r), \tag{4.4}$$

$$\operatorname{Var}\left[\tilde{\mathbf{g}}\right] = \frac{1}{r}\alpha(\mathbf{g}) \tag{4.5}$$

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

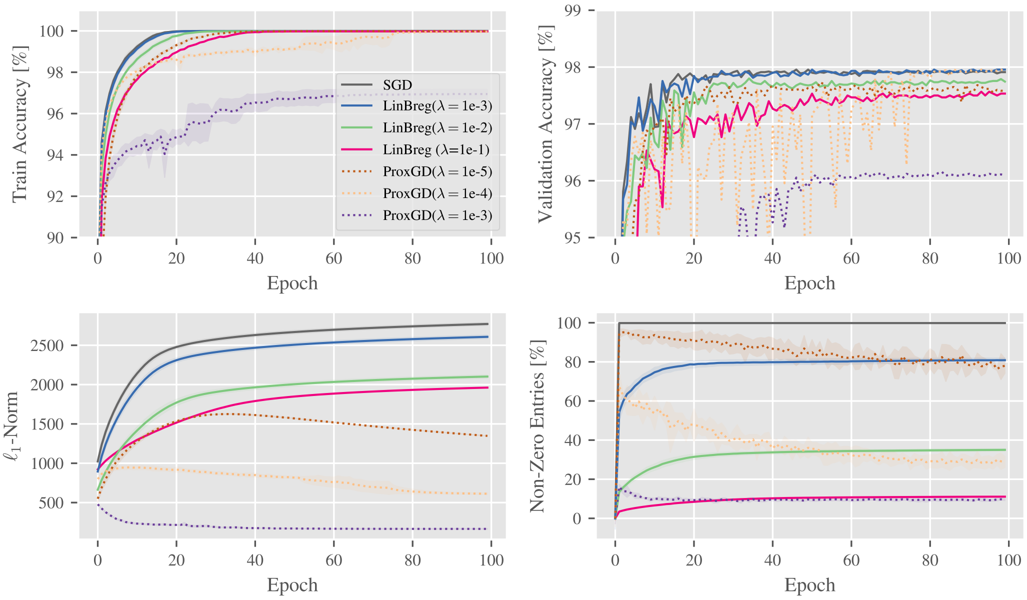


Figure 2: Comparison of vanilla SGD (black solid line), LinBreg (colored solid lines), and ProxGD (colored dotted lines) for different regularization parameters on MNIST. The curves show the averaged accuracies on train and validation sets, ℓ_1-norms, and non-zero entries over three runs. The shaded 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

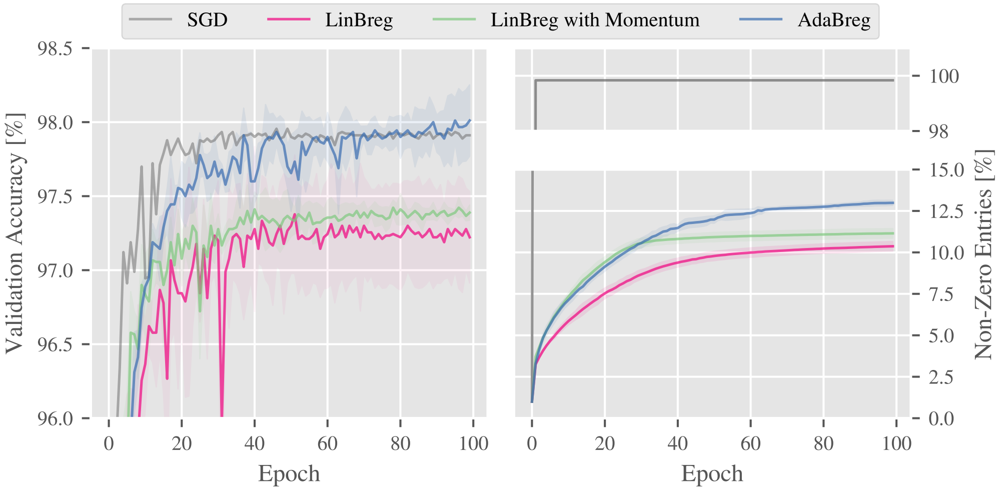


Figure 3: Comparison of LinBreg (with momentum), AdaBreg, and vanilla SGD. The networks generated by AdaBreg are sparse and generalize better.

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

$$5 \times 5 \text{ conv}, 64 \xrightarrow{\text{Maxpool}/2} 5 \times 5 \text{ conv}, 64 \xrightarrow{\text{Maxpool}/2} \text{fc } 1024 \longrightarrow \text{fc } 128 \longrightarrow \text{fc } 10$$

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	$\mathrm{N}_{\mathrm{total}}$ in [%]	$\mathrm{N}_{\mathrm{conv}}$ 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 regularizer (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 $\approx$ 0.89.

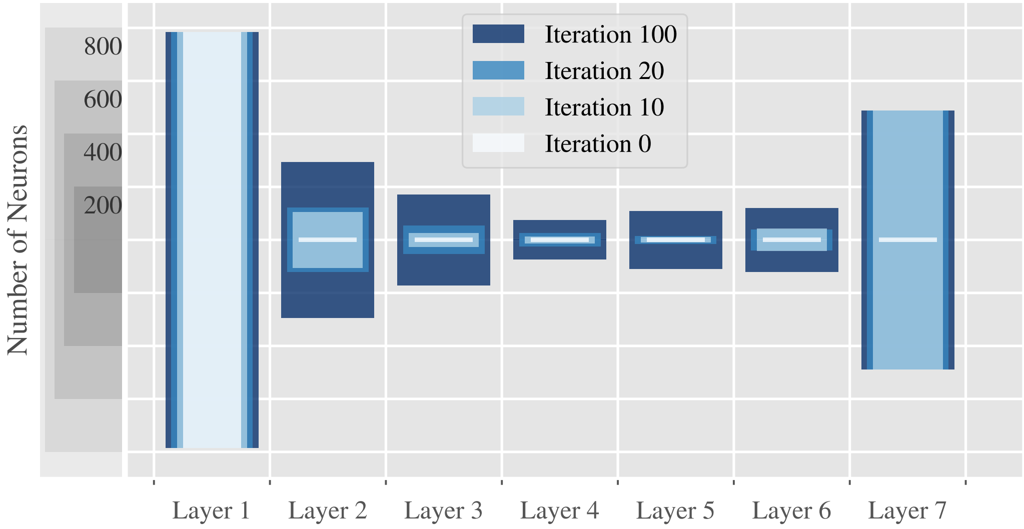


Figure 4: Architecture design for denoising: LinBreg automatically unveils an autoencoder.

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

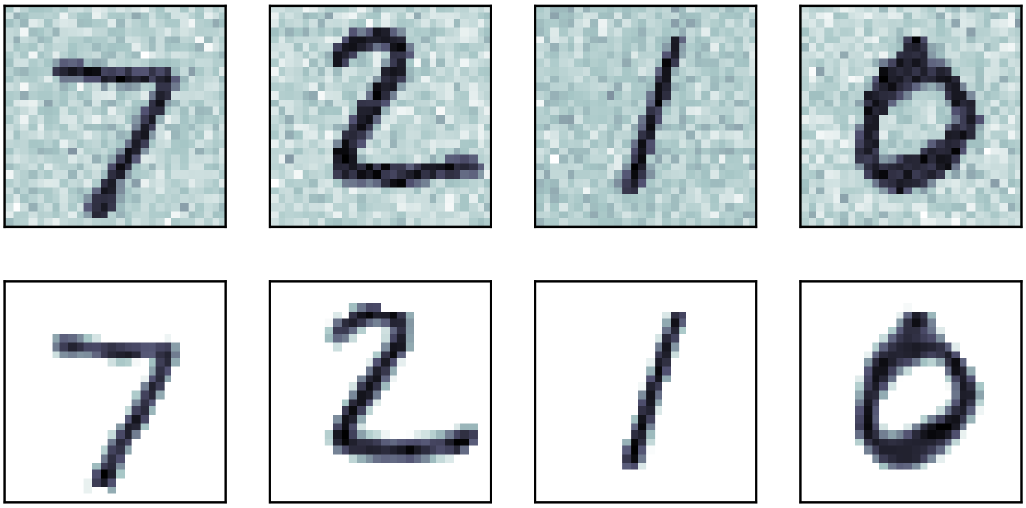


Figure 5: The denoising performance of the trained autoencoder on the test set with an average SSIM value of ≈ 0.93.

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 architecture 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

$$\begin{aligned}
&\mathcal{L}(\theta^{(k+1)}) - \mathcal{L}(\theta^{(k)}) \\
&\leq \langle \nabla\mathcal{L}(\theta^{(k)}), \theta^{(k+1)} - \theta^{(k)}\rangle + \frac{L}{2}\|\theta^{(k+1)} - \theta^{(k)}\|^2 \\
&= \langle g^{(k)}, \theta^{(k+1)} - \theta^{(k)}\rangle + \langle \nabla\mathcal{L}(\theta^{(k)}) - g^{(k)}, \theta^{(k+1)} - \theta^{(k)}\rangle + \frac{L}{2}\|\theta^{(k+1)} - \theta^{(k)}\|^2 \\
&= -\frac{1}{\tau^{(k)}}\langle v^{(k+1)} - v^{(k)}, \theta^{(k+1)} - \theta^{(k)}\rangle \\
&\qquad + \langle \nabla\mathcal{L}(\theta^{(k)}) - g^{(k)}, \theta^{(k+1)} - \theta^{(k)}\rangle + \frac{L}{2}\|\theta^{(k+1)} - \theta^{(k)}\|^2 \\
&= -\frac{1}{\tau^{(k)}} D_J^{\mathrm{sym}}(\theta^{(k+1)}, \theta^{(k)}) - \frac{1}{\delta\tau^{(k)}}\|\theta^{(k+1)} - \theta^{(k)}\|^2 \\
&\qquad + \langle \nabla\mathcal{L}(\theta^{(k)}) - g^{(k)}, \theta^{(k+1)} - \theta^{(k)}\rangle + \frac{L}{2}\|\theta^{(k+1)} - \theta^{(k)}\|^2.
\end{aligned}$$

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

$$\begin{aligned}
&\mathbb{E}\left[\mathcal{L}(\theta^{(k+1)})\right] - \mathbb{E}\left[\mathcal{L}(\theta^{(k)})\right] + \frac{1}{\tau^{(k)}}\mathbb{E}\left[D_J^{\mathrm{sym}}(\theta^{(k+1)}, \theta^{(k)})\right] + \frac{C}{2\delta\tau^{(k)}}\mathbb{E}\left[\|\theta^{(k+1)} - \theta^{(k)}\|^2\right] \\
&\qquad \leq \tau^{(k)}\delta\frac{\sigma^2}{2c},
\end{aligned}$$

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

$$\begin{aligned}&\tau^{(K)}\mathbb{E}\left[\mathcal{L}(\theta^{(K)})\right] - \tau^{(0)}\mathbb{E}\left[\mathcal{L}(\theta^{(0)})\right] + \\ &\sum_{k=0}^{K-1}\left(\mathbb{E}\left[D_J^{\mathrm{sym}}(\theta^{(k+1)},\theta^{(k)})\right] + \frac{C}{2\delta}\mathbb{E}\left[\|\theta^{(k+1)} - \theta^{(k)}\|^2\right]\right) \leq \delta\frac{\sigma^2}{2c}\sum_{k=0}^{K-1}(\tau^{(k)})^2\end{aligned}$$

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*

$$D^v_{J_\delta}(\tilde{\theta},\theta) = D^p_J(\tilde{\theta},\theta) + \frac{1}{2\delta}\|\tilde{\theta} - \theta\|^2,$$

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

$$\begin{aligned}D^v_{J_\delta}(\tilde{\theta},\theta) &= J_\delta(\tilde{\theta}) - J_\delta(\theta) - \langle v, \tilde{\theta} - \theta\rangle \\ &= J(\tilde{\theta}) + \frac{1}{2\delta}\|\tilde{\theta}\|^2 - J(\theta) - \frac{1}{2\delta}\|\theta\|^2 - \langle p, \tilde{\theta} - \theta\rangle - \frac{1}{\delta}\langle\theta, \tilde{\theta} - \theta\rangle \\ &= D^p_J(\tilde{\theta},\theta) + \frac{1}{2\delta}\|\tilde{\theta}\|^2 - \frac{1}{2\delta}\|\theta\|^2 - \frac{1}{\delta}\langle\theta,\tilde{\theta}\rangle + \frac{1}{\delta}\|\theta\|^2 \\ &= D^p_J(\tilde{\theta},\theta) + \frac{1}{2\delta}\|\tilde{\theta} - \theta\|^2.\end{aligned}$$

■

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)

$$\begin{aligned}
&D_{J_\delta}^{v^{(k+1)}}(\theta^*, \theta^{(k+1)}) - D_{J_\delta}^{v^{(k)}}(\theta^*, \theta^{(k)}) \\
&= J_\delta(\theta^{(k)}) - J_\delta(\theta^{(k+1)}) - \langle v^{(k+1)}, \theta^* - \theta^{(k+1)}\rangle + \langle v^{(k)}, \theta^* - \theta^{(k)}\rangle \\
&= -\left(J_\delta(\theta^{(k+1)}) - J_\delta(\theta^{(k)}) - \langle v^{(k)}, \theta^{(k+1)} - \theta^{(k)}\rangle\right) - \langle v^{(k)}, \theta^{(k+1)} - \theta^{(k)}\rangle \\
&\qquad - \langle v^{(k+1)}, \theta^* - \theta^{(k+1)}\rangle + \langle v^{(k)}, \theta^* - \theta^{(k)}\rangle \\
&= -D_{J_\delta}^{v^{(k)}}(\theta^{(k+1)}, \theta^{(k)}) + \langle v^{(k)} - v^{(k+1)}, \theta^* - \theta^{(k+1)}\rangle \\
&= -D_{J_\delta}^{v^{(k)}}(\theta^{(k+1)}, \theta^{(k)}) + \tau^{(k)}\langle g^{(k)}, \theta^* - \theta^{(k+1)}\rangle.
\end{aligned}$$

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

$$\begin{aligned} d_{k+1} - d_k &= -\mathbb{E}\left[D_{J_\delta}^{v^{(k)}}(\theta^{(k+1)}, \theta^{(k)})\right] + \tau^{(k)}\mathbb{E}\left[\langle \nabla\mathcal{L}(\theta^{(k)}), \theta^* - \theta^{(k+1)}\rangle\right] \\ &\qquad + \tau^{(k)}\mathbb{E}\left[\langle g^{(k)} - \nabla\mathcal{L}(\theta^{(k)}), \theta^* - \theta^{(k+1)}\rangle\right] \\ &\leq -\mathbb{E}\left[D_{J_\delta}^{v^{(k)}}(\theta^{(k+1)}, \theta^{(k)})\right] + \tau^{(k)}\frac{L^2}{2c}\mathbb{E}\left[\|\theta^{(k)} - \theta^{(k+1)}\|^2\right] \\ &\qquad + \frac{c-\mu}{2}\tau^{(k)}\mathbb{E}\left[\|\theta^* - \theta^{(k+1)}\|^2\right] \\ &\qquad + \tau^{(k)}\mathbb{E}\left[\langle g^{(k)} - \nabla\mathcal{L}(\theta^{(k)}), \theta^* - \theta^{(k+1)}\rangle\right]. \end{aligned}$$

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. ■