Functional Lifting, Direct Approaches and Applications of Nonconvex Optimization in Computer Vision

DISSERTATION
zur Erlangung des Grades eines Doktors
der Naturwissenschaften

vorgelegt von
M.Sc. Hartmut Bauermeister

eingereicht bei der Naturwissenschaftlich-Technischen Fakultät
der Universität Siegen
Siegen 2022

Betreuer und erster Gutachter
Prof. Dr. Michel Möller
Universität Siegen

Zweiter Gutachter
Prof. Dr. Daniel Cremers
Technische Universität München

Tag der mündlichen Prüfung
07.09.2023

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer, nicht angegebener Hilfsmittel angefertigt habe. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt. Es wurden keine Dienste eines Promotionsvermittlungsinstituts oder einer ähnlichen Organisation in Anspruch genommen.

Unterschrift

Abstract

Optimization problems are ubiquitous in computer vision, machine learning, economics and basically any field in the domain of natural and engineering sciences. Developments in optimization theory and algorithms have hence always been tightly interconnected to the problems arising from practical applications. This work follows the same path studying and developing various optimization techniques and models depending on the problem at hand. Fruitful optimization theory has been developed around convex problems and applied in computer vision tasks. In this work we extensively study theoretical properties of *functional lifting* techniques which make even nonconvex problems amenable to the tools from convex optimization by increasing the dimensionality of the original problem formulations. Furthermore we propose enhancements making existing approaches either more faithful or more efficient. Following the idea of increasing the dimensionality we study the effects of reparameterizations and neural network models with adaptive expressivity in the domain of machine learning. For linear inverse problems we devise an optimization approach combining data driven priors with provable convergence guarantees. Finally, we propose an optimization based approach for solving high-dimensional problems arising from attacks on data security in the application of federated machine learning.

Zusammenfassung

Optimierungsprobleme sind allgegenwärtig in den Bereichen Bildverarbeitung, maschinelles Lernen, Wirtschaftswissenschaften und nahezu allen Feldern der Natur- und Ingenieurswissenschaften. Entwicklungen in der Optimierungstheorie und daraus resultierende Algorithmen sind von daher stets eng mit den Problemen und Fragestellungen aus praktischen Anwendungen verbunden. Diese Arbeit folgt diesem Pfad, indem sie verschiedene Optimierungstechniken und Modelle im Bezug auf die jeweiligen Anwendungen untersucht und entwickelt. Insbesondere auf dem Gebiet der konvexen Optimierung wurden dabei ergiebige Theorien entwickelt und unter anderem in der Bildverarbeitung angewandt. In dieser Arbeit beschäftigen wir uns ausführlich mit Techniken des *functional liftings*, welche Konzepte der konvexen Optimierung auch auf nicht-konvexe Probleme anwendbar machen, indem die Dimensionalität der ursprünglichen Problemstellung erhöht wird. Des Weiteren werden Ansätze entwickelt, welche bestehende Methoden entweder genauer oder effizienter machen. Basierend auf der Idee, die Dimensionalität zu erhöhen, untersuchen wir den Effekt von Reparameterisierungen und neuronale Netze mit adaptiver Ausdrucksstärke im Bereich des maschinellen Lernens. Weiterhin entwickeln wir einen Optimierungsansatz für lineare inverse Probleme, welcher datengetriebene Methoden mit beweisbaren Konvergenzgarantien verbindet. Zuletzt entwickeln wir einen Optimierungsansatz für hochdimensionale Probleme, welche sich aus Attacken auf Datensicherheit in der Anwendung des verbündeten maschinellen Lernens ergeben.

Danksagung

An dieser Stelle möchte ich mich bei allen Leuten bedanken, die meine Zeit in Siegen zu einer wundervollen Erfahrung gemacht haben. Zunächst gilt mein Dank natürlich meinen Eltern und meinem Bruder, auf deren Unterstützung ich jederzeit zählen konnte.

Zurückblickend auf meine Schulzeit möchte ich auch meinen damaligen Lehrer Daniel Mayer hervorheben, der eine stete Inspiration war und mich selbst zu einer akademischen Laufbahn motiviert hat. Ihm gebührt großer Dank!

Wegweisend für meine weitere Laufbahn waren Daniel Cremers, welcher mir die Arbeit in seiner Gruppe ermöglicht hat, und insbesondere Thomas, der mich zu meiner Masterarbeit motiviert und mich dabei in hohem Maße unterstützt hat. Vielen Dank!

Mein herzlicher Dank gilt dann natürlich Michael, der mir die Möglichkeit eröffnet hat, Teil seiner Gruppe zu werden. Der Austausch mit ihm sowohl auf akademischer, als auch auf persönlicher Ebene, sowie seine stete und uneingeschränkte Unterstützung waren überaus bereichernd und keinesfalls selbstverständlich!

Ganz besonders möchte ich mich bei all meinen Kolleginnen und Kollegen bedanken, die jederzeit eine heitere und familiäre Atmosphäre kreiert haben. Von großer persönlicher Bedeutung für mich ist dabei Hannah, mit der mich eine enge Freundschaft verbindet. Du hast mir die Zeit in Siegen zu einer großen Freude gemacht, vielen Dank! :)

Zuletzt möchte ich mich sehr herzlich bei Emanuel bedanken, dessen Unterstützung für mich maßgeblich und rückblickend unverzichtbar war. Mindestens genau so wichtig waren für mich jedoch die zahlreichen und ausgiebigen Gespräche, die wir, oft zusammen mit Thomas, geführt haben. Abseits der akademischen Bereicherung, waren die humorvollen und teils philosophischen Diskussionen eine wahre Erweiterung meines Horizonts, deren Wert ich nicht hoch genug einschätzen kann.

Contents

Notation

$\mathcal{X}$	Compact nonempty set $\mathcal{X} \subset \mathbb{R}^m$.
$\mathcal{C}(\mathcal{X})$	The space of continuous functions mapping from $\mathcal{X}$ to $\mathbb{R}$, see [42, Ch. 4.2].
$\lVert f \rVert_\infty$	The infinity norm $\lVert f \rVert_\infty := \max_{x \in \mathcal{X}} \lvert f(x) \rvert$ of a continuous function $f \in \mathcal{C}(\mathcal{X})$.
$\mathcal{M}(\mathcal{X})$	The space of signed Radon measures on $\mathcal{X}$, see [42, Ch. 7].
$\mathcal{M}_+(\mathcal{X})$	The convex cone of Radon measures on $\mathcal{X}$, see [42, Lem. 7.2].
$\mathcal{P}(\mathcal{X})$	The space of probability Radon measures, see [42, Ch. 9].
δ_x	The Dirac measure centered at $x \in \mathbb{R}^m$.
$\lVert \mu \rVert_{\mathrm{TV}}$	The total variation norm $\lVert \mu \rVert_{\mathrm{TV}} := \sup_{\substack{f \in \mathcal{C}(\Gamma) \\ \lVert f \rVert_\infty \leq 1}} \lvert \langle \mu, f \rangle \rvert$ of a signed Radon measure $\mu \in \mathcal{M}(\mathcal{X})$.
$T \sharp \mu$	The pushforward of μ w.r.t. T for a signed Radon measure $\mu \in \mathcal{M}(\mathcal{X}_1)$ and a measurable mapping $T : \mathcal{X}_1 \to \mathcal{X}_2$ is defined by $(T \sharp \mu)(A) := \mu(T^{-1}(A))$ for all $A \subset \mathcal{X}_2$ in the corresponding σ-algebra
$\langle \mu, f \rangle$	The integral $\langle \mu, f \rangle := \int_{\mathcal{X}} f(x) \, \mathrm{d}\mu(x)$ for $f \in \mathcal{C}(\mathcal{X})$ and $\mu \in \mathcal{M}(\mathcal{X})$, see [42, Ch. 2].
ι_C	The indicator function $\iota_C : \mathbb{R}^m \to \overline{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\}$ with $\iota_C(x) = 0$ if $x \in C$ and $\iota_C(x) = \infty$ if $x \notin C$ for a set $C \subset \mathbb{R}^m$
σ_C	The support function $\sigma_C(x) := \sup_{y \in C} \langle x, y \rangle$ of $C \subset \mathbb{R}^m$ at $x \in \mathbb{R}^m$
C^*	The dual cone of C. These notions are defined analogously for the topologically paired spaces $\mathcal{M}(\mathcal{X})$ and $\mathcal{C}(\mathcal{X})$, see [115, Ch. 4].
$\operatorname{con} C$	The convex hull $\operatorname{con} C$ of a set $C \subset \mathbb{R}^m$ is the smallest convex set that contains C. Equivalently, $\operatorname{con} C$ is the set of all finite convex combinations of points in C, see [111, Ch. 2.F].
$\operatorname{con} f$	The largest convex function below f for some function f (with slight abuse of notation), see [111, Ch. 2.F].
lsc	We write lsc for lower semicontinuous, see [111, Ch. 1.B].
$\operatorname{epi} f$	The epigraph $\operatorname{epi} f := \{ (x, \alpha) \in \mathbb{R}^m \times \mathbb{R} \mid f(x) \leq \alpha \}$ of some extended real-valued function $f : \mathbb{R}^m \to \overline{\mathbb{R}}$.
$\operatorname{cl} f$	The epigraphical closure of some function f, see [111, Ch. 1.D].
$\llbracket \cdot \rrbracket$	The Iverson bracket, where $\llbracket P \rrbracket = 1$ if P is true and $\llbracket P \rrbracket = 0$, otherwise.
f^*	The Fenchel conjugate $f^*(y) := \sup_{x \in \mathbb{R}^m} \langle y, x \rangle - f(x)$ of some extended real-valued function $f : \mathbb{R}^m \to \overline{\mathbb{R}}$ at y, see [111, Ch. 11.A].
f^{**}	The Fenchel biconjugate $f^{**} := (f^*)^*$ of some extended real-valued function $f : \mathbb{R}^m \to \overline{\mathbb{R}}$, see [111, Ch. 11.A].
$\operatorname{prox}_f(x)$	The proximal operator $\operatorname{prox}_f(x) := \arg\min_{\bar{x}} f(\bar{x}) + \frac{1}{2} \lVert \bar{x} - x \rVert^2$ of a proper, lsc, convex function $f : \mathbb{R}^m \to \overline{\mathbb{R}}$.

Table 1: Table of notations.

Chapter 1

Introduction

Problems in computer vision arise from formulating models of the real world, the complexity of whose highly depends on the application at hand, resources and the desired solution quality. Naturally, also the resulting optimization procedures vary in several aspects and feature manifold different characteristics. This thesis presents multiple optimization approaches ranging from purely model based approaches to problems including learnable priors and analyzes them in terms of provable guarantees and efficiency.

The first chapter focuses mainly on model based approaches and investigates approaches for tackling nonconvex problems with tools from convex optimization. These approaches are subsumed under the notion of *functional lifting*. First, the theoretical aspects of functional lifting are investigated in the context of increasing the dimensionality of the original problem formulation. More specifically, connections to hierarchies of dual function spaces are developed and studied for a continuous MRF problem. This approach utilizes methods from optimal transport and semialgebraic geometry in order to interpret functional lifting as a lifting to the space of probability measures and discretizations thereof. The second part focuses on a method for improving performance of existing approaches by efficiently discretizing the spatial domain. Finally, the last part investigates if ideas from functional lifting can be transferred to machine learning problems by applying learnable reparameterizations.

The second chapter considers machine learning related problems and analyzes optimization approaches with respect to provable guarantees. At first, ideas from functional lifting are incorporated into neural network architectures. This can be interpreted as a replacement of common activation functions like ReLUs with more general linear spline based functions. The second approach studies linear inverse problems and mixes ideas from spectral regularizations with learnable models. Provable convergence guarantees are derived for dedicated choices of neural networks. Finally, we also study a highly nonconvex inverse problem originating from the application of federated learning. There our goal is to reconstruct images for given parameter updates obtained during the training of neural networks. In contrast to the previous setting with a linear forward operator, we now lack provable recovery guarantees. Nevertheless, we empirically show that reasonable reconstructions can be achieved by a thoroughly chosen optimization strategy thus breaking privacy for federated learning.

The results presented in this thesis are based on the publications [Pub2, Pub5, Pub6, Pub3, Pub1, Pub4], which also structure the sections within the two chapters below. For each section, I highlight and focus on my specific contribution within each of the published papers.

Chapter 2

Convexification Via Lifting

Many problems arising in the fields of computer vision and machine learning can be formulated as the minimization of dedicated energy functionals. These functionals typically operate on a certain class of functions $u\colon \Omega \to \Gamma$, where the choices of Ω and Γ depend on the application at hand. This chapter will focus on problems which can be written as a sum of a data dependent fidelity term $f\colon \Omega \times \Gamma \to \mathbb{R}$ and a regularizer $R\colon \Gamma^{\Omega} \to \mathbb{R}$ based on some assumed data priors. More specifically, the problem structure considered is given as functional of the form

$$E(u) = \int_{\Omega} f(x, u(x))\,\mathrm{d}x + R(u). \tag{2.1}$$

In most cases, either f or R or both are, however, nonconvex which in general implies also nonconvexity of E making the optimization problem notoriously difficult to solve. A very fundamental class expressible by Eq. (2.1) are multilabel problems. There Γ is a finite discrete set of labels and hence not even Γ is convex. In this case u is mapping from points in Ω to their corresponding labels and Γ is referred to as *label space*. For segmentation and minimal partition tasks a convex relaxation of the data term can now be achieved by interpreting the set of discrete labels as unit vectors in a $|\Gamma|$-dimensional vector space and replacing Γ by the convex envelope of those unit vectors, the $|\Gamma|$-dimensional unit simplex [74]. For total variation regulated minimal partition problems tight relaxations in a local sense have been studied relating the dual problem to paired calibrations [98, 26]. Multilabel problems also arise from discretizations of continuous data terms resulting from discrete sampling. This includes for instance stereo estimation [54, 104] and optical flow estimation [52, 19]. Convex relaxation approaches require reformulations in a higher-dimensional space, which usually has as many dimension as there are discrete labels [100, 99]. Hence those methods are subsumed under the notion of *functional lifting*. Various relaxation strategies have been studied in the context of stereo estimation [100, 99, 149, 104, 105], optical flow [129, 127], segmentation [156, 158, 74, 26], and optimization on manifolds [75], with algorithmic improvements such as [125]. Despite of the algorithmic advantages of functional lifting arising from convex relaxations those methods also suffer from the curse of dimensionality as the computational cost soars with the increasing the number of dimensions. In order to remediate this inherent drawback efficient discretization techniques have been proposed allowing for faithful relaxations even in a sublabel accurate sense and hence reducing the need for a large number of labels [92, 73, 91].

In Section 2.1 we study and extend existing theory of functional lifting in the context of *Markov Random Fields* (MRF). More specifically, Eq. (2.1) is analyzed for Ω being a discrete graph and Γ a continuous state space instead of finite sets which are used for classical multilabel problems. We interpret functional lifting as a discrete version of lifting the original problem to the infinite-dimensional space of probability measures. Finite-dimensional relaxations are then derived using subspaces of dual functions in the dual problem. For numerical implementations we use piecewise polynomial dual functions as strict generalization of existing sublabel accurate approaches [92, 73, 91]. We numerically show a reduction of the primal-dual gap for finer discretizations of the dual space in large scale stereo matching experiments.

Lifting approaches as described in Section 2.1 provide faithful relaxations of the nonconvex energy. However, those methods also come with the cost of high computational time and memory usage. In contrast to improvements in the discretizations of the label space in Section 2.1 we will discuss a more efficient spatial discretization of the domain Ω in Section 2.2. More specifically, we consider total variation (TV) regularized problems and motivate the proposed discretization by the piece-wise constant nature of TV regularized solutions. Using the constructed super-pixel graph we can apply lifting based approaches with a substantial decrease in computational effort in comparison to applying the lifting directly on the full Cartesian pixel grid. In stereo matching experiments we show significant speedups and at the same time obtain a near-globally optimal solution.

In contrast to the previous concepts of attaining more faithful solutions of Eq. (2.1) by increasing the dimensionality of Γ or improving efficiency by discretizing Ω, we will study a different way of increasing the dimensionality of our problem in Section 2.3. More precisely, we will introduce a parameterized model $\mathcal{N} \colon \Omega \times \Theta \to \Gamma$ and we optimize over the parameters $\theta \in \Theta$ for some high-dimensional parameter space Θ. We will discuss theoretical implications of this reparameterization and interpret empirical results obtained from numerical experiments.

2.1 Polynomial Lifting

This section is based on [Pub2] and studies problem (2.1) for discrete Ω in the context of MRFs. MRF problems and functional lifting are closely intertwined and early works on functional lifting [100, 99] have been inspired by the MRF community [53]. We will turn towards a theoretical analysis of the idea behind functional lifting and derive a generalized view on existing approaches. More specifically, we consider Eq. (2.1) for a MRF problem where Ω is a discrete grid and Γ is a continuous state space. We start our analysis by applying a dual decomposition to the original nonconvex problem resulting in duality gaps. To eliminate such gaps, we considers a reformulation of the original nonconvex task in the space of measures where the problems is convex and the duality gap hence vanishes. This infinite-dimensional reformulation is then approximated by a semi-infinite one, which is obtained via a piecewise polynomial discretization in the dual. We provide a geometric intuition behind the primal problem induced by the dual discretization and draw connections to optimization over moment spaces. This viewpoint allows us to regard existing functional lifting methods as the primal counterpart of certain discretizations of the space of dual functions in the dual problem. In contrast to existing discretizations which suffer from a grid bias, we show that a piecewise polynomial discretization better preserves the continuous nature of our problem. Invoking results from optimal transport theory and convex algebraic geometry we reduce the semi-infinite program to a finite one and provide a practical implementation based on semidefinite programming. We show, experimentally and in theory, that the approach successfully reduces the duality gap. To showcase the scalability of our approach, we apply it to the stereo matching problem between two disparate images.

2.1.1 Problem Description

This section considers the MAP-inference problem in a continuous *Markov Random Field* (MRF). Continuous MRFs are versatile and therefore widely used as a model in image processing, computer vision and machine learning [132, 153, 157, 30, 36, 146, 9, 147, 35, 10, 94, 117, 143]. Those continuous MRFs can be regarded as a spatially discrete case of Eq. (2.1) where $\Omega = \mathcal{V}$ for a finite set of nodes $\mathcal{V}$ in an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Furnished with the counting measure, the integral in Eq. (2.1) becomes a sum and we can write the inference problem as energy minimization of the form

$$\min_{u \in \Gamma^{\mathcal{V}}} \left\{ E(u) = \sum_{v \in \mathcal{V}} f_v(u_v) + \sum_{vw \in \mathcal{E}} f_{vw}(u_v, u_w) \right\}, \tag{P}$$

where we use the simplified notation $f_v(\,\cdot\,) := f(v, \cdot\,)$. The objective function E is an additive composition of a separable part, $\sum_{v \in \mathcal{V}} f_v(u_v)$, which represents the data fidelity term in Eq. (2.1), and a coupling part $\sum_{vw \in \mathcal{E}} f_{vw}(u_v, u_w)$ with pairwise functions $f_{vw} : \Gamma \times \Gamma \to \mathbb{R}$ corresponding to the regularizer in Eq. (2.1). We assume f_v and f_{vw} to be lower semicontinuous (lsc) and $\emptyset \neq \Gamma \subset \mathbb{R}^m$ compact. Our goal is to develop a convex optimization framework à la dual decomposition for the MAP inference problem.

The coupling term $\sum_{vw \in \mathcal{E}} f_{vw}(u_v, u_w)$ introduces a major challenge for efficient optimization in particular when $\mathcal{V}$ is large and therefore the problem is high-dimensional. For tractability, one therefore seeks to find a decomposable reformulation of the problem. In continuous optimization a viable approach is to derive a *Lagrangian relaxation* of the problem: The key idea is to introduce auxiliary variables $u_{vw} \in \mathbb{R}^m \times \mathbb{R}^m$ for each edge $vw \in \mathcal{E}$ and linear constraints $u_{vw} = (u_v, u_w)$. Dualizing the linear constraints with Lagrange multipliers $\lambda_{vw} \in \mathbb{R}^m \times \mathbb{R}^m$ one arrives at the Lagrangian dual problem which falls within the regime of convex optimization algorithms that can exploit its separable dual structure. Examples include subgradient ascent or the *primal-dual hybrid gradient* (PDHG) method [27].

However, the approach suffers from potentially large duality gaps since problem Eq. (P) is nonconvex in general. Indeed, it can be shown that a direct Lagrangian relaxation of Eq. (P) leads to the following "naive" convexification of the original problem:

$$\min_{u \in \Gamma^{\mathcal{V}}} \sum_{v \in \mathcal{V}} f_v^{**}(u_v) + \sum_{vw \in \mathcal{E}} f_{vw}^{**}(u_v, u_w), \tag{2.2}$$

where $f_v^*(\,\cdot\,) = \sup_{x \in \Gamma} \langle \,\cdot\,, x \rangle - f_v(x)$ is the convex conjugate of f_v, $f_v^{**} = (f_v^*)^*$ and f_{vw}^*, f_{vw}^{**} are defined likewise. Then f_v^{**} and f_{vw}^{**} are the convex biconjugates which correspond to the largest convex lsc under-approximations of f_v and f_{vw} respectively. With some abuse of notation, here, f_v and f_{vw} attain the value $+\infty$ whenever $u_v \notin \Gamma$ or $(u_v, u_w) \notin \Gamma^2$. Such component-wise convex envelopes can produce inaccurate or even trivial convex under-approximations to the global objective E.

2.1.2 Overview

To remedy duality gaps, in Section 2.1.3, we consider a reformulation of the problem in terms of a linear one over probability measures and perform the Lagrangian relaxation afterwards. To this end, let $\mathcal{P}(\Gamma^{\mathcal{V}})$ denote the set of probability Radon measures on the set $\Gamma^{\mathcal{V}}$. Instead of a naive formulation $\min_{\mu \in \mathcal{P}(\Gamma^{\mathcal{V}})} \int E \, \mathrm{d}\mu$ over $\mathcal{P}(\Gamma^{\mathcal{V}})$, which is intractable for large $\mathcal{V}$, we consider a formulation over $\mathcal{P}(\Gamma)^{\mathcal{V}}$ which exploits the partially separable structure of our problem. This is called the *local marginal polytope relaxation* (the former being called *marginal polytope relaxation*). For univariate Γ and submodular pairwise terms f_{vw} we prove tightness of the relaxation regardless of nonconvexity of f_v.

Since the Lagrange multipliers will be continuous functions, in Section 2.1.4, we consider hierarchies of dual programs Eq. (2.19) obtained by subspace approximations for which we show in Section 2.1.6 that the duality gap vanishes in the limit for a piecewise polynomial discretization.

In Section 2.1.5 we derive and study the primal optimization problem corresponding to the dual discretization. In particular we draw connections to optimization over moment spaces and aspects from variational analysis. We show that under a certain extremality condition, satisfied by a polynomial discretization with degree at least 2, the generalized biconjugate of a potentially discontinuous, nonconvex function equals (up to closure) the original function. As a consequence, in contrast to a piecewise linear approximation, our piecewise polynomial approximation conserves the original nonconvex cost including concavities when restricted to moment vectors of Diracs.

In Section 2.1.6, based on the above developments, we derive a piecewise polynomial discretization of the infinite-dimensional local marginal polytope relaxation and show that for univariate Γ and submodular pairwise terms f_{vw} the duality gap vanishes in the limit at rate $\mathcal{O}(1/(K \cdot \sqrt{\deg}))$ where K is the number of pieces and deg the degree of the pieces. After discretization the problem amounts to a semi-infinite program which can be transformed into a separable finite-dimensional semidefinite program applying concepts from algebraic geometry, such as nonnegativity certificates of polynomials. This allows us to derive an efficient first-order primal-dual algorithm which, due to the separable problem structure, can be parallelized on a GPU to handle large problems.

In Section 2.1.7 we provide numerical evidence for the strict reduction of the duality-gap and demonstrate the advantages of the nonlinear approximation over existing liftings in the literature. We implement our algorithm on a GPU and apply it to the nonconvex problem of large-scale stereo matching. The results show that increasing the dual subspace improves both, the dual energy, and the quality of a recovered primal solution from a discretized measure.

Our work tightly connected to a multitude of research areas in mathematics and computer science. In the following we give an overview of related work for the various tools used in this section.

MRFs Fix and Agarwal [41] are the first to propose a dual subspace approximation of the infinite-dimensional local marginal polytope relaxation. This allows for a generalization and unified treatment of convex piecewise linear formulations for MRFs with continuous state spaces proposed in [157, 155]. However, for the polynomial case, due to the semi-infinite problem structure, the existence of an implementable algorithm beyond subgradient ascent is posed as an open question by [41]. As a consequence, no numerical results are presented. Dual subspace approximations for related models in a spatially continuous setting have been considered by [87] and further employed for the global optimization of vector- or manifold-valued optimization problems in imaging and vision [88, 138, 139]. In these papers, piecewise linear dual approximations are used and the techniques developed in the present work allow one to go beyond the piecewise linear case in a tractable way. Besides applications in imaging and vision, MRFs with continuous state spaces have for example been used in protein folding [94].

Lifting to measures Lifting to measures for global constrained polynomial optimization is used in [69, 70] to reformulate the infinite-dimensional *linear program* (LP) in terms of a linear objective over the *semidefinite programming* (SDP) characterization of the finite-dimensional space of moments. Its

SDP-dual is connected to nonnegativity certificates of polynomials based on sum-of-squares (SOS) and the Positivstellensatz [64, 126, 121]. However, for high-dimensional problems the relaxations used in [69, 70] turn out intractable. As a remedy, [141, 148] consider sparse SOS-approaches and in particular sparse versions of the Positivstellensatz [148]. This is closely related to the local marginal polytope relaxation considered in this work. As a key difference to [69, 70, 141, 148] we consider possibly nonpolynomial objective functions which results in a generally nonlinear formulation over the space of moments. In contrast to [148], we apply optimal transport duality theory to further reduce the formulation using a sum-of-squares characterization of Lipschitz continuity of piecewise polynomials. Similar techniques have been considered recently in [72, 31] to estimate Lipschitz constants in neural networks.

Lagrangian relaxation and decoupling by lower relaxations A component-wise lifting in problems with a partially separable structure results in a generalized dual decomposition approach. Dual decomposition and Lagrangian relaxation are general principles in optimization and appear as a useful tool across many disciplines, see [11, Ch. 5–6], [76] and the references therein for an overview. Traditionally, decomposition methods are applied directly in the nonconvex setting without lifting, unlike the generalized scheme we present here which is based on a lifted reformulation. Without lifting this typically results in a component-wise convex lower envelope. More recently, [124] consider a homotopy method based on component-wise Lasry–Lions envelopes which specializes to a component-wise convex envelope in a certain limit case.

Generalized conjugacy and duality Our notion of a lifted convex conjugate is closely related to generalized conjugate functions originally due to [90]. It was utilized by [7] to study nonconvex dualities, expanding upon the work of [110] on augmented Lagrangians for nonconvex optimization. The specific case of quadratic conjugate functions was developed by [101] as an analytic tool in a seminal proof that the proximal subgradient map of a lower semicontinuous extended real-valued function is monotone if and only if the function is convex. Typically, generalized conjugate functions appear in the context of eliminating duality gaps [20, 110] in nonconvex and nonsmooth optimization. In that sense our work shares the goal with the aforementioned works. We offer a somewhat complementary approach through a primal-type lifting view which allows us to establish a connection to optimization in spaces of measures.

2.1.3 The Local Marginal Polytope Relaxation

To overcome the duality gap in Eq. (2.2) for nonconvex f_v, we propose to reformulate the original nonlinear problem in terms of an infinite-dimensional linear program over the space of signed Radon measures and apply the Lagrangian relaxation afterwards.

Optimizing a lsc function f can be related to a corresponding problem involving probability measures. For a probability measure $\mu \in \mathcal{P}(\mathcal{X})$ on a compact, non-empty set $\mathcal{X}$ we define the scalar product $\langle \mu, f\rangle := \int f \,\mathrm{d}\mu$. Function evaluation of f at some $x \in \mathcal{X}$ can now equally be written as $f(x) = \langle \delta_x, f\rangle$ where δ_x is the Dirac measure at x, i.e. $\delta_x(A) = 1$ if $x \in A$ and $\delta_x(A) = 0$ if $x \notin A$ and $A \subset \mathcal{X}$ measurable. The following key lemma reveals, that every minimization problem can be equivalently formulated in terms of an infinite-dimensional linear program.

Lemma 2.1.1. *Let $f : \mathcal{X} \to \mathbb{R}$ be lsc with $\emptyset \neq \mathcal{X} \subset \mathbb{R}^m$ compact. Then we have*

$$\min_{x\in\mathcal{X}} f(x) = \min_{\mu\in\mathcal{P}(\mathcal{X})} \langle \mu, f\rangle, \tag{2.3}$$

and $\hat{x} \in \arg\min_{x\in\mathcal{X}} f(x)$ is a solution to $\min_{x\in\mathcal{X}} f(x)$ if and only if $\delta_{\hat{x}}$ is a minimizer of $\min_{\mu\in\mathcal{P}(\mathcal{X})}\langle \mu, f\rangle$.

Proof. The result follows immediately via compactness of $\mathcal{X}$ and the properties of probability measures: By compactness of $\mathcal{X}$ and since f is lsc relative to $\mathcal{X}$ we know $\hat{x}$ exists. Let $f_{\min} = f(\hat{x})$. By the properties of the Lebesgue integral we have:

$$\int f \,\mathrm{d}\mu \geq \int f_{\min} \,\mathrm{d}\mu = f_{\min} \int 1 \,\mathrm{d}\mu = \min_{x\in\mathcal{X}} f(x),$$

for all $\mu \in \mathcal{P}(\mathcal{X})$ and $\min_{x\in\mathcal{X}} f(x) = \langle \delta_{\hat{x}}, f\rangle = f(\hat{x})$. □

Let $\pi_v : \Gamma^{\mathcal{V}} \to \Gamma$, $\pi_{vw} : \Gamma^{\mathcal{E}} \to \Gamma^2$ denote the canonical projections onto the v^{th}, respectively onto the v^{th} and w^{th} components. Furthermore denote by $\pi_v\sharp\mu$ the pushforward of $\mu \in \mathcal{P}(\Gamma^{\mathcal{V}})$ w.r.t. π_v, i.e. $\pi_v\sharp\mu \in \mathcal{P}(\Gamma)$ and $\pi_v\sharp\mu(A) = \mu(\pi_v^{-1}(A))$ for $A \subset \Gamma$ measurable. The pushforward $\pi_{vw}\sharp$ is defined analogously. Then, in our case, applying the reformulation from Lemma 2.1.1 directly to the cost function E with $\mathcal{X} = \Gamma^{\mathcal{V}}$ we obtain by linearity:

$$\begin{aligned}\min_{\mu \in \mathcal{P}(\Gamma^{\mathcal{V}})} \langle \mu, E \rangle &= \min_{\mu \in \mathcal{P}(\Gamma^{\mathcal{V}})} \left\langle \mu, \sum_{v \in \mathcal{V}} f_v \circ \pi_v + \sum_{vw \in \mathcal{E}} f_{vw} \circ \pi_{vw} \right\rangle \\ &= \min_{\mu \in \mathcal{P}(\Gamma^{\mathcal{V}})} \sum_{v \in \mathcal{V}} \langle \pi_v\sharp\mu, f_v \rangle + \sum_{vw \in \mathcal{E}} \langle \pi_{vw}\sharp\mu, f_{vw} \rangle.\end{aligned}$$

This relaxation is known as the *full marginal polytope relaxation* which is, however, intractable if $\mathcal{V}$ is large as one minimizes over probability measures on the product space $\mathcal{P}(\Gamma^{\mathcal{V}})$. Instead, we consider the following linear programming relaxation of Eq. (P) which is also referred to as the *local marginal polytope relaxation* [94, 41, 142, 116] which is more tractable as the optimization variable lies in the product space of probability measures $\mathcal{P}(\Gamma)^{\mathcal{V}}$:

$$\inf_{\mu \in \mathcal{P}(\Gamma)^{\mathcal{V}}} \left\{ \mathbf{E}(\mu) := \sum_{v \in \mathcal{V}} \langle \mu_v, f_v \rangle + \sum_{vw \in \mathcal{E}} \mathrm{OT}_{f_{vw}}(\mu_v, \mu_w) \right\}. \tag{R-P}$$

Here, $\mathrm{OT}_{f_{vw}}$ denotes the optimal transportation [58] with marginals (μ_v, μ_w) and cost f_{vw} defined by

$$\mathrm{OT}_{f_{vw}}(\mu_v, \mu_w) = \inf_{\mu_{vw} \in \Pi(\mu_v, \mu_w)} \langle \mu_{vw}, f_{vw} \rangle. \tag{2.4}$$

The constraint set $\Pi(\mu_v, \mu_w)$ consists of all Radon probability measures on Γ^2 with specified marginals μ_v and μ_w:

$$\Pi(\mu_v, \mu_w) = \left\{ \mu_{vw} \in \mathcal{P}(\Gamma^2) \mid \pi_v\sharp\mu_{vw} = \mu_v,\ \pi_v\sharp\mu_{vw} = \mu_w \right\}, \tag{2.5}$$

where with a slight abuse of notation $\pi_v : \Gamma \times \Gamma \to \Gamma$ corresponds to the canonical projection onto the v^{th} component. Note that for finite state-spaces (i.e., $|\Gamma| < \infty$), the set $\mathcal{P}(\Gamma)$ can be identified with the standard $(|\Gamma| - 1)$-dimensional probability simplex and $\Pi(\mu_v, \mu_w)$ with the set of nonnegative $|\Gamma| \times |\Gamma|$ matrices whose rows and columns sum up to μ_w and μ_v. In that case, the linear program given in (R-P) is equivalent to the well-known finite-dimensional local marginal polytope relaxation for MRFs, which is for example studied in [150]. We further remark that the case of finite Γ has been extensively studied in the literature, see, e.g., [59, 140] for recent overviews.

For the more challenging setting of continuous state-spaces, a major difficulty stems from the fact that the linear programming relaxation (R-P) is an infinite-dimensional optimization problem posed in the space of Radon probability measures. Perhaps due to this difficulty, discrete MRF approaches are still routinely applied despite the continuous nature of Γ. This is typically done by considering a finite sample approximation of Γ, so that the infinite-dimensional linear program reduces to a finite-dimensional one. This, however, may lead to discretization errors and comes with an exponential complexity in the dimension of Γ.

Due to the fact that $\Pi(\delta_x, \delta_{x'}) = \{\delta_{(x,x')}\}$ one sees that restricting μ_v (and therefore μ_{vw}) to be Dirac probability measures, the formulation (R-P) reduces to the original problem (P). As one instead considers the larger convex set of all probability measures, it is a *relaxation* which lower bounds (P), i.e., we have the following important relation:

$$\text{(R-P)} \leq \text{(P)}. \tag{2.6}$$

When considering a relaxation, the immediate question arises whether this lower bound is attained, i.e., the relaxation is tight and therefore the above inequality holds with equality. For finite and ordered Γ the situation is well-understood, see [150]. For continuous $\Gamma \subset \mathbb{R}^m$ a total order is possible if $m = 1$, i.e., Γ is an interval. Indeed, if in addition E is submodular, tightness of the local marginal polytope relaxation can be derived from [6, Thm. 2] regardless of nonconvexity of f_v. A function $E \colon \Gamma^{\mathcal{V}} \to \mathbb{R}$ is said to be submodular, if for all $x' \in \Gamma^{\mathcal{V}}$ and $x'' \in \Gamma^{\mathcal{V}}$ it holds $E(x') + E(x'') \geq E(\min\{x', x''\}) + E(\max\{x', x''\})$

where the min and max operations are to be understood componentwise on Γ, see [6, Sec. 2.1].

Proposition 2.1.2. *Let $\emptyset \neq \Gamma \subset \mathbb{R}$ be compact and $f_v : \Gamma \to \mathbb{R}$, $f_{vw} : \Gamma \times \Gamma \to \mathbb{R}$ be continuous with f_v possibly nonconvex. If f_{vw} is submodular for all $vw \in \mathcal{E}$, E is submodular as well and the relaxation is tight, i.e.,*

$$(\text{R-P}) = (\text{P}). \tag{2.7}$$

Proof. First, we show that E is submodular. The unaries f_v are submodular as functions of a single variable are submodular and hence as a nonnegative sum of submodular functions, E is submodular itself.

For $\mu \in \mathcal{P}(\Gamma)$ define the cumulative distribution function $\mathcal{F}_\mu \colon \Gamma \to [0,1]$ as

$$\mathcal{F}_\mu(x) = \mu(\{\, y \in \Gamma \mid y \geq x \,\}) \tag{2.8}$$

and the "inverse" cumulative distribution function $\mathcal{F}_\mu^{-1} \colon [0,1] \to \Gamma$ as

$$\mathcal{F}_\mu^{-1}(t) = \sup\{\, x \in \Gamma \mid \mathcal{F}_\mu(x) \geq t \,\}. \tag{2.9}$$

Submodularity of E now implies by [6, Thm. 2]

$$\begin{aligned}
(\text{P}) &= \inf_{\mu \in \mathcal{P}(\Gamma)^{\mathcal{V}}} \int_0^1 \sum_{v \in \mathcal{V}} f_v(\mathcal{F}_{\mu_v}^{-1}(t)) + \sum_{vw \in \mathcal{E}} f_{vw}(\mathcal{F}_{\mu_v}^{-1}(t), \mathcal{F}_{\mu_w}^{-1}(t)) \,\mathrm{d}t && (2.10)\\
&= \inf_{\mu \in \mathcal{P}(\Gamma)^{\mathcal{V}}} \sum_{v \in \mathcal{V}} \int_0^1 f_v(\mathcal{F}_{\mu_v}^{-1}(t)) \,\mathrm{d}t + \sum_{vw \in \mathcal{E}} \int_0^1 f_{vw}(\mathcal{F}_{\mu_v}^{-1}(t), \mathcal{F}_{\mu_w}^{-1}(t)) \,\mathrm{d}t. && (2.11)
\end{aligned}$$

Using submodularity of f_v and f_{vw}, and applying [6, Prop. 2] and [6, Prop. 4] we get

$$\begin{aligned}
\int_0^1 f_v(\mathcal{F}_{\mu_v}^{-1}(t)) \,\mathrm{d}t &= \langle \mu_v, f_v \rangle, && (2.12)\\
\int_0^1 f_{vw}(\mathcal{F}_{\mu_v}^{-1}(t), \mathcal{F}_{\mu_w}^{-1}(t)) \,\mathrm{d}t &= \mathrm{OT}_{f_{vw}}(\mu_v, \mu_w) && (2.13)
\end{aligned}$$

by the definition of the optimal transportation in Eq. (2.4). Hence it holds (R-P) = (P) and the relaxation (R-P) is tight. □

Remark 2.1.3. A solution to the local marginal polytope relaxation (R-P) even allows for the reconstruction of a globally optimal solution to the unrelaxed original problem (P): More precisely, $\mu \in \mathcal{P}(\Gamma)^{\mathcal{V}}$ is a minimizer of (R-P) if and only if $v \mapsto \mathcal{F}_{\mu_v}^{-1}(t)$ is a minimizer of (P) for almost all $t \in [0,1]$, see [6, Thm. 2].

For $f_{vw}(x,y) = g(x-y)$ and g convex, f_{vw} is submodular, see [6, Sec. 2.2]. Especially the total variation-like couplings considered in the experimental sections are therefore submodular and thus the local marginal polytope relaxation is tight in this case.

2.1.4 Dual Discretization for the Continuous MRF

2.1.4.1 A Reduced Dual Formulation

A Lagrangian relaxation to the infinite-dimensional problem (R-P) is obtained by dualizing the marginalization constraints (2.5) for each $e \in \mathcal{E}$ with Lagrange multipliers $\lambda_e \in \mathcal{C}(\Gamma)^2$. This is equivalent to a substitution of the optimal transportation with its dual formulation. Adopting the approach of [41] a finite-dimensional problem is obtained by approximating the Lagrange multipliers in terms of finite linear combinations of certain basis functions. These approximations are chosen in such a way that the classical Lagrangian relaxation and the discrete approach, described above, are special cases of the considered framework.

In contrast to previous approaches [41], we restrict ourselves to metric pairwise terms $f_{vw}(x,y) = \mathbf{d}(x,y)$ which eventually leads to a different dual formulation and turns out more tractable: Then, the optimal transportation $\mathrm{OT}_{f_{vw}}(\mu_v, \mu_w)$ in problem (R-P) is the Wasserstein-1 distance $W_1^{\mathbf{d}}(\mu_v, \mu_w)$ induced by the metric $\mathbf{d}$ between μ_v and μ_w. Furthermore we assume that $f_v : \Gamma \to \mathbb{R}$ is lsc and $\Gamma \subset \mathbb{R}^m$ is a compact nonempty set. In particular this implies that f_v is proper and bounded from below.

Thanks to optimal transport duality theory [137] we are therefore able to obtain a more compact dual formulation, which is instrumental to derive a tractable implementation for a piecewise polynomial discretization later on: More precisely, we substitute $\mathrm{OT}_{f_{vw}}(\mu_v, \mu_w) = W_1^{\mathbf{d}}(\mu_v, \mu_w)$in problem (R-P) with its dual formulation

$$W_1^{\mathbf{d}}(\mu_v, \mu_w) = \sup_{\lambda \in \mathrm{Lip}_{\mathbf{d}}(\Gamma)} \int \lambda(x) \, \mathrm{d}(\mu_v - \mu_w)(x), \tag{2.14}$$

where λ is 1-Lipschitz with respect to the metric $\mathbf{d}$, i.e.,

$$\lambda \in \mathrm{Lip}_{\mathbf{d}}(\Gamma) = \{ \lambda : \Gamma \to \mathbb{R} \mid |\lambda(x) - \lambda(y)| \leq \mathbf{d}(x, y) \}. \tag{2.15}$$

In contrast to the general dual formulation of optimal transport, which involves two Lagrange multipliers per edge, interacting via the constraint set, the Wasserstein-1 dual involves only a single dual variable for each edge, that satisfies a Lipschitz constraint. We show in Section 2.1.6 that even though this formulation still involves infinitely many constraints, thanks to convex algebraic geometry [12], there exists a tractable finite representation of this constraint set in terms of SDP.

We denote by $\mathcal{C}(\Gamma)$ the set of continuous functions mapping from Γ to $\mathbb{R}$. For notational convenience, we assign an arbitrary orientation to the edges in $\mathcal{G}$. After introducing a graph divergence operator $\mathrm{Div} : \mathcal{C}(\Gamma)^{\mathcal{E}} \to \mathcal{C}(\Gamma)^{\mathcal{V}}$ defined by:

$$-(\mathrm{Div}\, \lambda)_v = \sum_{w:(v,w) \in \mathcal{E}} \lambda_{(v,w)} - \sum_{w:(w,v) \in \mathcal{E}} \lambda_{(w,v)}, \tag{2.16}$$

an interchange of min and sup yields the following reduced dual problem, which is the starting point for further discussion and a tractable implementation:

$$\sup_{\lambda \in \mathcal{C}(\Gamma)^{\mathcal{E}}} \left\{ D(\lambda) := -\sum_{v \in \mathcal{V}} \sigma_{\mathcal{P}(\Gamma)}(-f_v + (\mathrm{Div}\, \lambda)_v) - \sum_{e \in \mathcal{E}} \iota_{\mathcal{K}}(\lambda_e) \right\}, \tag{R-D}$$

where $\mathcal{K} = \mathrm{Lip}_{\mathbf{d}}(\Gamma)$ and $\sigma_{\mathcal{P}(\Gamma)}(-f_v + (\mathrm{Div}\, \lambda)_v) = \sup_{\mu \in \mathcal{P}(\Gamma)} \langle \mu, -f_v + (\mathrm{Div}\, \lambda)_v \rangle$ is the support function of $\mathcal{P}(\Gamma)$ at $-f_v + (\mathrm{Div}\, \lambda)_v$, which, thanks to Lemma 2.1.1, can be rewritten:

$$-\sigma_{\mathcal{P}(\Gamma)}(-f_v + (\mathrm{Div}\, \lambda)_v) = \min_{x \in \Gamma} (f_v - (\mathrm{Div}\, \lambda)_v)(x). \tag{2.17}$$

Remark 2.1.4 (Topological pairing). In the following sections we will study the duality of (R-P) and (R-D) and discretizations thereof. Therefore we first want to emphasize the duality of the spaces those problems live on and their topologies. Consider the vector space of continuous functions $\mathcal{C}(\Gamma)$ on Γ and the topology $\tau_{\|\cdot\|_\infty}$ induced by the infinity norm $\|f\|_\infty = \max_{x \in \Gamma} |f(x)|$. We denote the corresponding *topological vector space* as $(\mathcal{C}(\Gamma), \tau_{\|\cdot\|_\infty})$, [115, 1.6]. Due to the Riesz–Markov–Kakutani representation theorem the dual space $(\mathcal{C}(\Gamma), \tau_{\|\cdot\|_\infty})^*$ can be identified by the space of signed Radon measures $\mathcal{M}(\Gamma)$ via the bilinear mapping $\langle \mu, f \rangle := \int f \, \mathrm{d}\mu$ for $f \in \mathcal{C}(\Gamma)$ and $\mu \in \mathcal{M}(\Gamma)$, see [42, Thm. 7.17]. The infinity norm $\|\cdot\|_\infty$ on $\mathcal{C}(\Gamma)$ induces the total variation norm for measures $\|\cdot\|_{\mathrm{TV}}$ on $\mathcal{M}(\Gamma)$. Albeit every element of $\mathcal{C}(\Gamma)$ induces a continuous linear functional on $\mathcal{M}(\Gamma)$ via $\langle \cdot, \cdot \rangle$, not every element in $(\mathcal{M}(\Gamma), \tau_{\|\cdot\|_{\mathrm{TV}}})^*$ necessarily has to correspond to an element in $\mathcal{C}(\Gamma)$. However, we will heavily rely on a bijective correspondence in terms of continuous dual elements between those two vector spaces subsequently. Thus we have to furnish $\mathcal{M}(\Gamma)$ with a more suitable topology. As a matter of fact, the *weak** topology $\tau_{\mathcal{C}(\Gamma)}$, which is defined to be the weakest topology such that all elements of $\mathcal{C}(\Gamma)$ can be identified with continuous linear functions on $\mathcal{M}(\Gamma)$, does exactly what we need, [115, 3.14]. Therefore we will consider the dual system of the topological vector spaces $(\mathcal{C}(\Gamma), \tau_{\|\cdot\|_\infty})$ and $(\mathcal{M}(\Gamma), \tau_{\mathcal{C}(\Gamma)})$ henceforth.

Having specified the topologies we can now show a result on strong duality of (R-P) and (R-D):

Proposition 2.1.5. *Let $\emptyset \neq \Gamma \subset \mathbb{R}^m$ be compact, let $f_v : \Gamma \to \mathbb{R}$ be lsc, and $f_{vw} = \mathbf{d}$, where $\mathbf{d} : \Gamma^2 \to \mathbb{R}$ is lsc and a metric. Then, the following strong duality holds:*

$$(\text{R-P}) = (\text{R-D}), \tag{2.18}$$

and a minimizer of (R-P) *exists.*

Proof. We will show that strong duality holds by applying the Fenchel–Rockafellar duality Theorem [108, Thm. 3]. We first show that (R-P) and (R-D) are dual in the sense of [108, Thm. 3].

Therefore define $F : \mathcal{C}(\Gamma)^{\mathcal{E}} \to \overline{\mathbb{R}}$ as

$$F(\lambda) := \sum_{e \in \mathcal{E}} \iota_{\mathcal{K}}(\lambda_e),$$

which is proper and convex. For showing F is lsc we have to show the closedness of $\mathcal{K}$: Consider $f \in \mathcal{C}(\Gamma) \setminus \mathcal{K}$. Then there exist $x, y \in \Gamma$ such that $r := |f(x) - f(y)| - \mathbf{d}(x, y) > 0$. Denote by $B_r(f) := \{\, g \in \mathcal{C}(\Gamma) \mid \|g - f\|_\infty < r \,\}$ the open ball of radius r around f in the infinity norm. Then it follows for $g \in B_r(f)$ that

$$\begin{aligned} |g(x) - g(y)| &\geq |f(x) - f(y)| - |(g - f)(x) - (g - f)(y)| \\ &\geq |f(x) - f(y)| - \|g - f\|_\infty > \mathbf{d}(x, y). \end{aligned}$$

Hence $g \in \mathcal{C}(\Gamma) \setminus \mathcal{K}$ and therefore $\mathcal{K}$ is closed.

Likewise, define the functional $G : \mathcal{C}(\Gamma)^{\mathcal{V}} \to \overline{\mathbb{R}}$ as

$$G(\lambda) := \sum_{v \in \mathcal{V}} \sigma_{\mathcal{P}(\Gamma)}(\lambda_v - f_v),$$

which is proper convex lsc, as it is a pointwise supremum over linear functionals. Furthermore G is continuous. To this end consider $\xi, \zeta \in \mathcal{C}(\Gamma)$. Then it follows

$$\begin{aligned} \sigma_{\mathcal{P}(\Gamma)}(\xi - f_v) - \sigma_{\mathcal{P}(\Gamma)}(\zeta - f_v) &= \max_{x \in \Gamma}(\xi - f_v)(x) - \max_{x \in \Gamma}(\zeta - f_v)(x) \\ &= \max_{x \in \Gamma} \left(\xi - f_v + \min_{y \in \Gamma}(f_v - \zeta)(y) \right)(x) \\ &\leq \max_{x \in \Gamma}((\xi - f_v) + (f_v - \zeta))(x) \\ &\leq \max_{x \in \Gamma}(\xi - \zeta)(x) \\ &= \|\xi - \zeta\|_\infty. \end{aligned}$$

Interchanging ξ and ζ yields

$$\begin{aligned} \sigma_{\mathcal{P}(\Gamma)}(\xi - f_v) - \sigma_{\mathcal{P}(\Gamma)}(\zeta - f_v) &= -(\sigma_{\mathcal{P}(\Gamma)}(\zeta - f_v) - \sigma_{\mathcal{P}(\Gamma)}(\xi - f_v)) \\ &\geq -\|\xi - \zeta\|_\infty. \end{aligned}$$

and thus $|\sigma_{\mathcal{P}(\Gamma)}(\xi - f_v) - \sigma_{\mathcal{P}(\Gamma)}(\zeta - f_v)| \leq \|\xi - \zeta\|_\infty$. Especially, G is continuous.

Hence (R-D) is equivalent to $\inf_{\lambda \in \mathcal{C}(\Gamma)^{\mathcal{E}}} F(\lambda) + G(\operatorname{Div} \lambda)$ where Div is linear and continuous as finite sums are continuous on topological vector spaces, see [115, 1.6].

Next we compute the convex conjugate $F^* : \mathcal{M}(\Gamma)^{\mathcal{E}} \to \overline{\mathbb{R}}$ of f as

$$F^*(\mu) = \sum_{e \in \mathcal{E}} \sigma_{\mathcal{K}}(\mu_e).$$

For calculating the convex conjugate $G^* : \mathcal{M}(\Gamma)^{\mathcal{V}} \to \overline{\mathbb{R}}$ of g we first note that $\mathcal{P}(\Gamma)$ is closed in the weak* topology: Probability measures are defined as the non-negative measures $\mu \in \mathcal{M}(\Gamma)$ such that $\langle \mu, 1 \rangle = 1$. Therefore $\mathcal{P}(\Gamma)$ can be written as intersection of measures μ for which $\langle \mu, 1 \rangle = 1$ and

$\langle \mu, f\rangle \geq 0$ for all $f \in \mathcal{C}(\Gamma)$ with $f \geq 1$, i.e.

$$\mathcal{P}(\Gamma) = \{\, \mu \in \mathcal{M}(\Gamma) \mid \langle \mu, 1\rangle = 1 \,\} \cap \bigcap_{\substack{f\in\mathcal{C}(\Gamma)\\ f\geq 0}} \{\, \mu \in \mathcal{M}(\Gamma) \mid \langle \mu, f\rangle \geq 0 \,\}$$

is closed as an intersection of preimages of closed sets under continuous functions by the definition of the weak* topology. It follows that $\mathcal{P}(\Gamma)$ is closed. Additionally, the mapping $\mu \to \langle \mu, f_v\rangle$ is lsc on $\mathcal{P}(\Gamma)$, [119, Lem. 1.6]. Therefore $\iota_{\mathcal{P}(\Gamma)}(\mu_v) + \langle \mu_v, f_v\rangle$ is convex proper lsc. By the definition of the support function it holds $\big(\iota_{\mathcal{P}(\Gamma)}(\,\cdot\,) + \langle\,\cdot\,, f_v\rangle\big)^* = \sigma_{\mathcal{P}(\Gamma)}(\,\cdot\, - f_v)$ and thus

$$G^*(\mu) = \sum_{v\in\mathcal{V}} \big(\iota_{\mathcal{P}(\Gamma)}(\mu_v) + \langle \mu_v, f_v\rangle\big)\,.$$

Now (R-P) is exactly $\sup_{\mu\in\mathcal{P}(\Gamma)} -G^*(\mu) - F^*(\nabla\mu)$ where $\nabla : \mathcal{M}(\Gamma)^{\mathcal{V}} \to \mathcal{M}(\Gamma)^{\mathcal{E}}$ is defined as

$$(\nabla\mu)_{(v,w)} := \mu_v - \mu_w.$$

and therefore $\nabla^* : \mathcal{C}(\Gamma)^{\mathcal{E}} \to \mathcal{C}(\Gamma)^{\mathcal{V}} = -\operatorname{Div}$. As F is finite for $\lambda \in \mathcal{K}^{\mathcal{E}}$ and G is continuous, the Fenchel–Rockafellar duality Theorem yields (R-P) = (R-D) and a minimizer $\hat{\mu} \in \mathcal{P}(\Gamma)$ of (R-P) exists. □

Proposition 2.1.6. *Let $\emptyset \neq \Gamma \subset \mathbb{R}^m$ be compact, let $f_v : \Gamma \to \mathbb{R}$ be lsc, and $f_{vw} = \mathbf{d}$, where $\mathbf{d} : \Gamma^2 \to \mathbb{R}$ is a continuous metric. Then a maximizer of* (R-D) *exists.*

Proof. Consider a sequence $\lambda^t \in \mathcal{K}^{\mathcal{E}}$ such that $F(\lambda^t)+G(\operatorname{Div}\lambda^t)$ converges to $\inf_{\lambda\in\mathcal{C}(\Gamma)^{\mathcal{E}}} F(\lambda)+G(\operatorname{Div}\lambda)$. We can replace the sequence by $\overline{\lambda}^t := \lambda^t - \min_{x\in\Gamma}\lambda^t(x)$ as $F(\overline{\lambda}^t) + G(\operatorname{Div}\overline{\lambda}^t) = F(\lambda^t) + G(\operatorname{Div}\lambda^t)$. Since $\overline{\lambda}^t$ is Lipschitz wrt the continuous metric $\mathbf{d}$, the sequence $\overline{\lambda}^t$ is equicontinuous and uniformly bounded. By the Arzelà–Ascoli Theorem and the closedness of $\mathcal{K}$ there exists a subsequence converging to a function $\hat{\lambda} \in \mathcal{K}$, [115, A5]. By the lower-semicontinuity of the objective, $\hat{\lambda}$ has to be a maximizer of (R-D). □

2.1.4.2 Discretization of the Reduced Dual Formulation for the Metric MRF

The next step in our strategy to obtain a tractable formulation is to restrict $\lambda_{(v,w)} \in \mathcal{C}(\Gamma)$ in problem (R-D) to a subspace $\Lambda = \langle \varphi_0, \ldots, \varphi_n\rangle$ spanned by basis functions $\varphi_k \in \mathcal{C}(\Gamma)$ and instead consider $\sup_{\lambda\in\Lambda^{\mathcal{E}}} D(\lambda)$. We will regard the basis functions $\{\varphi_0, \ldots, \varphi_n\}$ as components of some $\varphi \in \mathcal{C}(\Gamma, \mathbb{R}^{n+1})$, i.e. $\varphi(x) := (\varphi_0(x), \ldots, \varphi_n(x))^T$ and $\mathcal{C}(\Gamma, \mathbb{R}^{n+1})$ is the vector space of continuous functions mapping from Γ to $\mathbb{R}^{n+1}$. In the situation of Proposition 2.1.2, for any hierarchy of increasingly expressive dual subspaces $\Lambda_1 \subset \Lambda_2 \subset \cdots \subset \mathcal{C}(\Gamma)$ the induced hierarchy of dual problems

$$(2.2) = \max_{\lambda\in(\Lambda_1)^{\mathcal{E}}} D(\lambda) \leq \max_{\lambda\in(\Lambda_2)^{\mathcal{E}}} D(\lambda) \leq \cdots \leq \max_{\lambda\in\mathcal{C}(\Gamma)^{\mathcal{E}}} D(\lambda) \overset{(2.18)}{=} \text{(R-P)} \overset{(2.7)}{=} \text{(P)}, \tag{2.19}$$

leads to a reduction of the duality gap where the first equality holds due to the explanation that will be given in Section 2.1.5.1. In general, properness of the inclusions in the hierarchy of the dual subspaces need not imply strict inequalities in Eq. (2.19). However, for a piecewise polynomial hierarchy with increasing degrees and/or number of pieces we will show in Proposition 2.1.28 that the duality gap (P) $- \max_{\lambda\in\Lambda^{\mathcal{E}}} D(\lambda)$ eventually vanishes with rate $\mathcal{O}(1/(K\cdot\sqrt{\deg}))$ as the number of pieces K and/or the degree deg goes to ∞.

In the context of dual discretization it is crucial to discuss the duality between finite-dimensional subspaces of $\mathcal{C}(\Gamma)$ and certain equivalence classes of measures and in particular moment spaces. This will be particularly important in Section 2.1.5 where we derive and study the primal problem corresponding to the discretized dual problem:

Note that for any $\lambda \in \Lambda$ we obtain a linear functional on $\mathcal{M}(\Gamma)$ by mapping $\mu \in \mathcal{M}(\Gamma)$ to $\mu \mapsto \langle \mu, \lambda\rangle = \int \lambda \,\mathrm{d}\mu$. Although any $\mu \in \mathcal{M}(\Gamma)$ maps to a linear functional on Λ, this mapping is not injective. In particular there exist distinct measures $\mu \neq \nu$ such that $\langle \mu, \lambda\rangle = \langle \nu, \lambda\rangle$ for every $\lambda \in \Lambda$, i.e., μ and ν induce the same linear functional on Λ. Instead, invoking [115, Thm. 4.9] the dual space

Λ^* can be related to a quotient space of $\mathcal{M}(\Gamma)$ by an isometric isomorphism to a quotient space on $\mathcal{M}(\Gamma)$ via

$$\mathcal{M}_\Lambda := \Lambda^* \cong \mathcal{M}(\Gamma)/\Lambda^0 := \{\, \mu + \Lambda^0 \mid \mu \in \mathcal{M}(\Gamma) \,\} \cong \mathbb{R}^{n+1}, \tag{2.20}$$

where $\Lambda^0 := \{\, \mu \in \mathcal{M} \mid \langle \mu, \lambda \rangle = 0 \text{ for all } \lambda \in \Lambda \,\}$ is the annihilator of Λ. Any $\mu \in \mathcal{M}(\Gamma)$ generates an equivalence class in $\mathcal{M}(\Gamma)/\Lambda^0$ and the corresponding element in $\mathcal{M}_\Lambda$ is denoted by $[\mu]_\Lambda$. Given a measure $\mu \in \mathcal{M}(\Gamma)$, for a particular choice of basis functions $\{\varphi_0, \varphi_1, \ldots, \varphi_n\}$, we refer to $\mu_k = \langle \mu, \varphi_k \rangle = \int \varphi_k(x)\, \mathrm{d}\mu(x)$ as the k^{th} *moment* of μ. Consider the dual basis of $\{\varphi_0, \ldots, \varphi_n\}$ in $\mathcal{M}_\Lambda$. Then we know that $([\mu]_\Lambda)_k = \langle [\mu]_\Lambda, \varphi_k \rangle = \mu_k$ and hence the k^{th} component of $[\mu]_\Lambda$ in the dual basis is exactly the k^{th} moment of μ. Furthermore we define the space induced by (unsigned) Radon measures as

$$(\mathcal{M}_\Lambda)_+ = \{\, [\mu]_\Lambda \mid \mu \in \mathcal{M}_+(\Gamma) \,\}. \tag{2.21}$$

This is discussed in Section 2.1.6 in the context of implementation. Note that we can now define $\boldsymbol{\varphi}\colon \Gamma \to \mathcal{M}_\Lambda$, $x \mapsto [\delta_x]_\Lambda$ and it holds $(\boldsymbol{\varphi}(x))_k = \langle [\delta_x]_\Lambda, \varphi_k \rangle = \varphi_k(x)$. Hence $\boldsymbol{\varphi}$ is the analogous counterpart of φ in $\mathcal{M}_\Lambda$. Note that $\boldsymbol{\varphi}$ is continuous as $\mathcal{M}(\Gamma)$ is endowed with the weak* topology and $\Lambda \subset \mathcal{C}(\Gamma)$. We will denote the dual basis of $\{\varphi_0, \ldots, \varphi_n\}$ by $\{\mathbf{m}_0, \ldots, \mathbf{m}_n\}$. We will also write $y = y_0\mathbf{m}_0 + \cdots + y_n\mathbf{m}_n$ as $y = (y_0, \ldots, y_n)^T$ and identify $\mathcal{M}_\Lambda$ with $\mathbb{R}^{n+1}$ if it's clear from the context. Similarly, for $\lambda \in \Lambda$ we will identify $\lambda = \lambda_0\varphi_0 + \cdots + \lambda_n\varphi_n$ as $\lambda = (\lambda_0, \ldots, \lambda_n)^T$.

As a subspace, Λ inherits the infinity norm from $\mathcal{C}(\Gamma)$. We will analogously introduce an adapted variant of the total variation norm to $\mathcal{M}_\Lambda$ as the dual norm of $\|\cdot\|_\infty$ via

$$\|y\|_{\mathrm{TV}} := \sup_{\substack{\|\lambda\|_\infty \le 1 \\ \lambda \in \Lambda}} |\langle y, \lambda \rangle|, \tag{2.22}$$

for $y \in \mathcal{M}_\Lambda$. This definition can be related to the total variation norm on measures:

Proposition 2.1.7. *For $y \in \mathcal{M}_\Lambda$ the total variation norm can be expressed as*

$$\|y\|_{\mathrm{TV}} = \min\{\, \|\mu\|_{\mathrm{TV}} \mid \mu \in \mathcal{M}(\Gamma),\, [\mu]_\Lambda = y \,\}. \tag{2.23}$$

Proof. First note $\|\mu\|_{\mathrm{TV}} = \sup_{\substack{\|\lambda\|_\infty \le 1 \\ \lambda \in \mathcal{C}(\Gamma)}} |\langle y, \lambda \rangle| \ge \sup_{\substack{\|\lambda\|_\infty \le 1 \\ \lambda \in \Lambda}} |\langle \mu, \lambda \rangle| = \|y\|_{\mathrm{TV}}$ for $[\mu]_\Lambda = y$. Therefore $\|y\|_{\mathrm{TV}} \le \inf\{\, \|\mu\|_{\mathrm{TV}} \mid \mu \in \mathcal{M}(\Gamma) \,\}$. By definition of $\mathcal{M}_\Lambda$ it is clear that y is a linear function on Λ and $|\langle y, \lambda \rangle| \le \|y\|_{\mathrm{TV}} \cdot \|\lambda\|_\infty$ for $\lambda \in \Lambda$. The Hahn-Banach Theorem, [115, Thm. 3.3], now implies the existence of a linear functional $\hat{y}$ on $\mathcal{C}(\Gamma)$ such that $\hat{y} \mid_\Lambda = y$ and $\|\hat{y}(\lambda)\| \le \|y\|_{\mathrm{TV}} \cdot \|\lambda\|_\infty$. Particularly this implies $\hat{y}$ is bounded, $\hat{y} \in \mathcal{M}(\Gamma)$ and $\|\hat{y}\|_{\mathrm{TV}} \le \|y\|_{\mathrm{TV}}$. Also it holds $[\hat{y}]_\Lambda = y$. Therefore $\|y\|_{\mathrm{TV}} \ge \inf\{\, \|\mu\|_{\mathrm{TV}} \mid \mu \in \mathcal{M}(\Gamma) \,\}$. Hence equality holds and $\hat{y}$ attains the minimum of $\{\, \|\mu\|_{\mathrm{TV}} \mid \mu \in \mathcal{M}(\Gamma) \,\}$. □

In view of the Wasserstein-1 dual in Eq. (2.14) constant components in $\lambda_{(v,w)} \in \Lambda$ cancel out and therefore, assuming $\varphi_0 \equiv 1$ is the constant one function, we can choose $\lambda_0 = 0$. Thanks to Eq. (2.17) we can formulate the following discretized dual problem as

$$\sup_{\lambda \in \Lambda^{\mathcal{E}}} D(\lambda) = \sup_{\lambda \in \Lambda^{\mathcal{E}}} -\sum_{v \in \mathcal{V}} \sup_{x \in \Gamma} \left(\langle \boldsymbol{\varphi}(x), (\operatorname{Div}_\Lambda \lambda)_v \rangle - f_v(x) \right) - \sum_{e \in \mathcal{E}} \iota_{\mathcal{K}}(\lambda_e). \tag{dR-D}$$

2.1.5 Lifted Convex Conjugates: A Primal View

2.1.5.1 The Discretized Primal Problem

A central goal of this section is to study the primal problem induced by the dual discretization. This allows us to show that discretizations which obey a certain extremality property conserve the original cost when restricting to discretized Diracs, see Theorem 2.1.18.

In order to derive the discretized primal problem note that for any $f_v : \Gamma \to \mathbb{R}$ the expression $\sup_{x \in \Gamma} \langle \boldsymbol{\varphi}(x), (\operatorname{Div}_\Lambda \lambda)_v \rangle - f_v(x)$ in (dR-D) (up to the presence of $\boldsymbol{\varphi}(x)$) resembles the form of a convex conjugate. Indeed, exploiting the notion of an extended real-valued function this is the convex conjugate

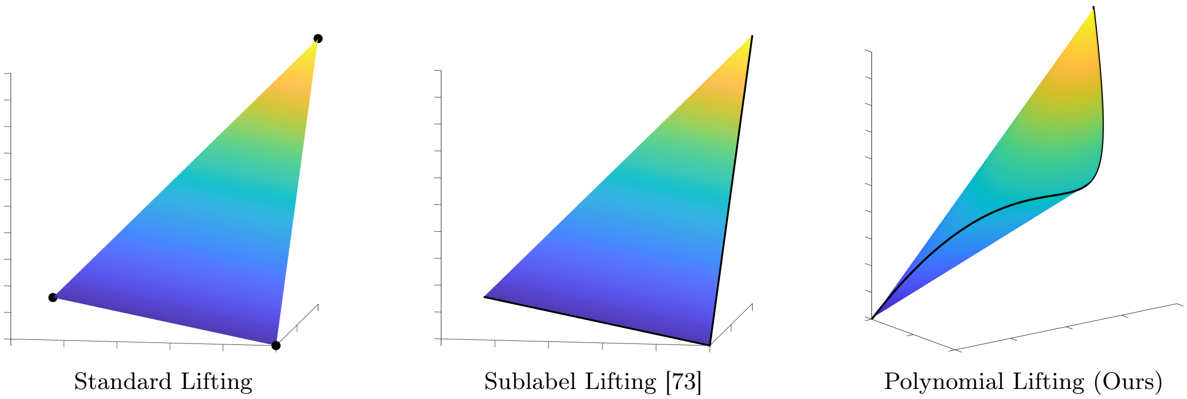


Figure 2.1: Different finite-dimensional approximations $\mathcal{P}_\Lambda$ of the infinite-dimensional space of probability measures $\mathcal{P}([-1,1])$ with $\Gamma = [-1,1]$. Left and middle: 2-dimensional probability simplex and right: Monomial moment space $\operatorname{con}\{\, (x, x^2, x^3)^T \mid x \in [-1,1] \,\}$ of degree 3. The approximations are obtained as the convex hulls of the black curves $x \mapsto \varphi(x)$ for 3 different choices of φ. From left to right cf. Example 2.1.10, Example 2.1.11 and Example 2.1.12 where $\varphi_0 \equiv 1$ is not plotted for better visual appearance. In all cases, the black curves themselves correspond to Dirac measures and the convex hulls of the curves correspond to the space of probability measures. In contrast to the simplex that only has a finite number of extreme points, the monomial moment curve comprises a continuum of extreme points so that no Dirac measure on the monomial moment curve can be expressed as a convex combination of other Diracs. Figure taken from [Pub2].

of a lifted version $f_\Lambda : \mathcal{M}_\Lambda \to \overline{\mathbb{R}}$ of $f : \Gamma \to \mathbb{R}$ which is defined by

$$f_\Lambda(y) = \begin{cases} f(x), & \text{if } y = \boldsymbol{\varphi}(x) \text{ for some } x \in \Gamma, \\ \infty, & \text{otherwise,} \end{cases} \tag{2.24}$$

and gives rise to our notion of a *lifted convex conjugate* which is studied in depth in the course of this section.

Since the "lifting" to a higher dimensional space happens through the application of the mapping $\boldsymbol{\varphi}$ it is intuitive to refer to $\boldsymbol{\varphi}$ as the lifting aka feature map in this context. Likewise we refer to the curve described by $x \mapsto \boldsymbol{\varphi}(x)$ as the moment curve.

With this construction at hand and via an application of Fenchel–Rockafellar duality we are ready to state the primal problem of the discretized dual problem in terms of the lifted biconjugates $(f_v)_\Lambda^{**}$:

$$\min_{\mathbf{u} \in \mathcal{M}_\Lambda^{\mathcal{V}}} \left\{ \mathbf{E}_\Lambda(\mathbf{u}) := \sum_{v \in \mathcal{V}} (f_v)_\Lambda^{**}(\mathbf{u}_v) + \sum_{e \in \mathcal{E}} \sigma_{\mathcal{K} \cap \Lambda}\big((\nabla_\Lambda \, \mathbf{u})_e\big) \right\}, \tag{dR-P}$$

where $\nabla_\Lambda := -\operatorname{Div}_\Lambda^*$ is the negative adjoint of the graph divergence, the graph gradient operator. A comparison of (dR-P) with the convex relaxation Eq. (2.2) shows the effect of the dual discretization, where the classical convex biconjugates are replaced with biconjugates of the "lifted" functions $(f_v)_\Lambda$. Indeed, without lifting, i.e., $\varphi(x) = x$, we recover the classical biconjugates and therefore the convex relaxation Eq. (2.2). In particular this explains the first equality in the hierarchy Eq. (2.19).

2.1.5.2 Lifting and Moments

A fundamental question in the study of the lifted biconjugate $(f_v)_\Lambda^{**}$ is the characterization of its domain $\operatorname{dom}(f_v)_\Lambda^{**} = \operatorname{con}(\boldsymbol{\varphi}(\Gamma))$, which, assuming f_v is bounded from below, equals the convex hull of the image of $\boldsymbol{\varphi}$. For that purpose we define the probability moment space as

$$\mathcal{P}_\Lambda = \{\, [\mu]_\Lambda \in \mathcal{M}_\Lambda \mid \mu \in \mathcal{P}(\Gamma) \,\}. \tag{2.25}$$

For $\varphi_0 \equiv 1$ it follows $\mathcal{P}_\Lambda = \{\, y \in (\mathcal{M}_\Lambda)_+ \mid y_0 = 1 \,\}$ as $\mu \in \mathcal{P}(\Gamma)$ is a probability measure if and only if $\mu \in \mathcal{M}_+(\Gamma)$ and $\langle \mu, \varphi_0 \rangle = 1$. Also see Lemma 2.1.24.

Proposition 2.1.8. *Let $\emptyset \neq \Gamma$ be compact and $\Lambda = \langle \varphi_0, \dots, \varphi_n \rangle \subset \mathcal{C}(\Gamma)$. Then every moment vector of a probability measure is a finite convex combination of moment vectors of Dirac measures, i.e.*

$$\operatorname{con}(\boldsymbol{\varphi}(\Gamma)) = \mathcal{P}_\Lambda, \tag{2.26}$$

and $\mathcal{P}_\Lambda$ is nonempty and compact.

Proof. Note that $\mathcal{P}_\Lambda$ is convex and bounded. It is also closed: To this end consider a sequence $y^t \to y$ with $y^t \in \mathcal{P}_\Lambda$. This means for any t there exists $\mu^t \in \mathcal{P}(\Gamma)$ with $y^t_k = \int \varphi_k \, \mathrm{d}\mu^t$ for $k = 1, \dots, n$. Since Γ is compact, due to Prokhorov's Theorem, see [119, Sec. 1.1], there exists a weakly* convergent subsequence $\mu^{t_j} \overset{*}{\rightharpoonup} \mu$ for $\mu \in \mathcal{P}(\Gamma)$ and, hence, $\int \varphi_k \, \mathrm{d}\mu^{t_j} = y_k^{t_j} \to \int \varphi_k \, \mathrm{d}\mu = y_k$. Therefore $y \in \mathcal{P}_\Lambda$. Next we show identity of the support functions of the convex sets $\operatorname{con}(\boldsymbol{\varphi}(\Gamma))$ and $\mathcal{P}_\Lambda$ as this implies the equality of the sets. To this end let $\lambda \in \Lambda = \langle \varphi_0, \dots, \varphi_n \rangle$. We write $\lambda(x) = \langle \boldsymbol{\varphi}(x), \lambda \rangle$. We have the identities:

$$\sigma_{\boldsymbol{\varphi}(\Gamma)}(-\lambda) = -\min_{x \in \Gamma} \langle \boldsymbol{\varphi}(x), \lambda \rangle = -\min_{\mu \in \mathcal{P}(\Gamma)} \int \lambda \, \mathrm{d}\mu = -\min_{y \in \mathcal{P}_\Lambda} \langle y, \lambda \rangle = \sigma_{\mathcal{P}_\Lambda}(-\lambda),$$

where the first equality follows by the definition of the support function, the second equality by Lemma 2.1.1 and the third equality by the definition of $\mathcal{P}_\Lambda$, the identity

$$\min_{\mu \in \mathcal{P}(\Gamma)} \int \lambda \, \mathrm{d}\mu = \min_{\mu \in \mathcal{P}(\Gamma)} \sum_{k=1}^{n} \lambda_k \int \varphi_k \, \mathrm{d}\mu,$$

and the substitution $y_k = \int \varphi_k \, \mathrm{d}\mu$. Since for each such λ the support functions are equal we have equality of the support functions of $\boldsymbol{\varphi}(\Gamma)$ and $\mathcal{P}_\Lambda$. Since Γ is compact and $\boldsymbol{\varphi}$ continuous and the convex hull of a compact set stays compact, cf. [111, Cor. 2.30], we can replace $\boldsymbol{\varphi}(\Gamma)$ with its convex hull $\operatorname{con} \boldsymbol{\varphi}(\Gamma)$ and the conclusion follows. □

In other words $\mathcal{P}_\Lambda$ is the set of all "infinite" convex combinations of points $\boldsymbol{\varphi}(x) \in \mathcal{M}_\Lambda$ with $x \in \Gamma$, i.e., of points that belong to the image of $\boldsymbol{\varphi}$. Thus the probability moment space can be written equivalently as the set of all finite convex combinations of moment vectors of Dirac measures, in the same way the unit simplex is the convex hull of the unit vectors.

The proof of the above proposition also reveals that for $f \in \Lambda$, the lifted biconjugate is a linear function over the moment space,

$$f_\Lambda^{**}(y) = \langle y, f \rangle + \iota_{\mathcal{P}_\Lambda}(y), \tag{2.27}$$

whose minimization is actually equivalent to minimizing the original function.

Specializing φ_k to the monomial basis and Γ to a set defined via polynomial inequalities, this yields the formulation proposed by [69, 70] for constrained polynomial optimization.

More generally, a meaningful notion of moments y_k is induced by a lifting map $\boldsymbol{\varphi}$ which is a homeomorphism between Γ and $\boldsymbol{\varphi}(\Gamma)$.

Definition 2.1.9 (lifting map)**.** *Let Γ be compact and be nonempty. Then we say the mapping $\boldsymbol{\varphi} : \Gamma \to \mathcal{M}_\Lambda$ is a lifting map if $\boldsymbol{\varphi}$ is continuous on Γ and injective with continuous inverse $\boldsymbol{\varphi}^{-1} : \boldsymbol{\varphi}(\Gamma) \to \mathbb{R}^m$.*

It is instructive to discuss possible choices for φ including the ones that correspond to existing discretizations for the continuous MRF such as the discrete approach and the piecewise linear approach. In the latter two cases, the moment space is merely the unit simplex. In that sense, the monomial moment space can be interpreted as a "nonlinear probability simplex", which, in contrast to the unit simplex, has infinitely many extreme points, see Fig. 2.1. For the same reason, as we will see in the course of this section, it better suits the continuous nature of our optimization problem.

The discrete sampling-based approach is recovered by the following choice of φ:

Example 2.1.10. We discretize the interval $\Gamma = [a, b]$, $a < b$ and re-define $\Gamma := \{\gamma_1, \dots, \gamma_K\}$ with $\gamma_k \in [a, b]$, $\gamma_k < \gamma_{k+1}$. For any $\gamma_k \in \Gamma$ let $\varphi(\gamma_k) = e_k$ with $e_k \in \mathbb{R}^K$ being the k^{th} unit vector. As a result φ is the canonical basis and spans the space of discrete functions $f : \{\gamma_1, \dots, \gamma_K\} \to \mathbb{R}$ and the moment space $\operatorname{con}(\boldsymbol{\varphi}(\Gamma)) = \mathcal{P}_\Lambda$ is given by the unit simplex in $\mathcal{M}_\Lambda$.

The above example can be extended by assigning any points $x \in (\gamma_k, \gamma_{k+1})$ to points on the connecting line between two corresponding Diracs which results in a more continuous formulation:

Example 2.1.11. Let $\Gamma = [a, b]$, $a < b$: Let $\gamma_k < \gamma_{k+1}$ and $\gamma_1 = a$, $\gamma_{K+1} = b$ be a sequence of knots that subdivide the interval Γ into K subintervals $[\gamma_k, \gamma_{k+1}] =: \Gamma_k$. Let $\mathbf{1}_k := \sum_{i=1}^{k} e_i$. We define $\varphi(x) = \mathbf{1}_k^\alpha := \alpha e_k + (1-\alpha)e_{k-i}$, $\alpha \in [0, 1]$ such that $x = \gamma_k^\alpha := \alpha\gamma_{k+1} + (1-\alpha)\gamma_k$. This yields a 2-sparse lifting map that has been used in the related work of [73]. Its component functions φ_k are the finite element hat basis functions that span the space Λ of univariate piecewise linear functions on Γ.

For φ_k being chosen as the monomials we obtain the classical notion of moments:

Example 2.1.12. For $\varphi_0 \equiv 1$, the space of univariate polynomials $\Lambda = \mathbb{R}[x]$ with maximum degree n is spanned by the monomials:

$$\varphi(x) = (1, x, x^2, \dots, x^n)^T, \tag{2.28}$$

and $\operatorname{con}(\varphi(\Gamma)) = \mathcal{P}_\Lambda$ is the monomial moment space.

Example 2.1.13. Identifying $\Gamma = \{\, x \in \mathbb{R}^2 \mid \|x\|_2 = 1 \,\}$ with the complex unit circle $\Gamma \cong \{\, z \in \mathbb{C} \mid |z| = 1 \,\}$ and again assuming $\varphi_0 \equiv 1$, the mapping

$$\varphi(z) = (1, \operatorname{Re}(z), \operatorname{Im}(z), \dots, \operatorname{Re}(z^n), \operatorname{Im}(z^n))^T \in \mathbb{R}^{2n+1} \tag{2.29}$$

spans the space Λ of real trigonometric polynomials of maximum degree n. Parameterizing elements $z \in \Gamma$ via the bijection between $z = e^{i\omega}$ and its angle $\omega \in [0, 2\pi)$ the components of φ are the Fourier basis functions, which define the Carathéodory curve [25]. The convex hull $\operatorname{con}(\boldsymbol{\varphi}(\Gamma)) = \mathcal{P}_\Lambda$ is the trigonometric moment space.

2.1.5.3 Extremal Subspaces

In contrast to the piecewise linear lifting, the (trigonometric) polynomial lifting is extremal in the sense that no Dirac measure on the moment curve can be expressed as a convex combination of other Diracs, which sets up a certain one-to-one correspondence between Diracs δ_x and lifted points $\boldsymbol{\varphi}(x)$. This is illustrated in Fig. 2.1 monomial case.

More formally, we call a lifting map $\boldsymbol{\varphi}$ an extremal curve if each $y \in \boldsymbol{\varphi}(\Gamma)$ is an extreme point of $\operatorname{con}(\boldsymbol{\varphi}(\Gamma))$ defined according to [109, Sec. 18].

Definition 2.1.14 (extreme points)**.** *Let C be a convex set and $y \in C$. Then y is called an extreme point of C if there is no way to express y as a convex combination $y = (1-\alpha)x + \alpha z$ of $x, z \in C$ and $0 < \alpha < 1$, except by taking $y = x = z$.*

Definition 2.1.15 (extremal moment curve)**.** *Let Γ be compact and be nonempty. Then we say the mapping $\boldsymbol{\varphi} : \Gamma \to \mathcal{M}_\Lambda$ is an extremal moment curve if $\boldsymbol{\varphi}$ is a lifting map and any point $y \in \boldsymbol{\varphi}(\Gamma) \subset \mathcal{M}_\Lambda$ is an extreme point of $\operatorname{con}\boldsymbol{\varphi}(\Gamma)$.*

Via a change of basis it becomes clear that the definition of extremality is independent of a specific choice of a basis for Λ. Therefore extremality is rather a property of the subspace Λ. This also motivates the following lemma which shows that extremality is inherited along a hierarchy $\Theta \subset \Lambda \subset \mathcal{C}(\Gamma)$.

Lemma 2.1.16 (extremal subspaces)**.** *Let $\emptyset \neq \Gamma \subset \mathbb{R}^m$ be compact. Let $\Theta \subset \Lambda \subset \mathcal{C}(\Gamma)$ be a hierarchy of finite-dimensional subspaces of the space of continuous functions $\mathcal{C}(\Gamma)$. Let $\Theta = \langle \theta_1, \dots, \theta_n \rangle$ such that $\boldsymbol{\theta} : \Gamma \to \mathbb{R}^n$ is an extremal curve. Then Λ is spanned by an extremal curve as well.*

Proof. $\{\theta_1, \dots, \theta_n\}$ is a basis of $\Theta \subset \Lambda$ and therefore linearly independent. Since Λ is finite-dimensional in view of the basis extension theorem $\{\theta_1, \dots, \theta_n\}$ can be extended to a basis $\varphi = (\theta_1, \dots, \theta_n, \psi_1, \dots, \psi_k)^T$ of Λ with vectors $\psi_i \in G$, where $\operatorname{span} G = \Lambda$ and $|G| < \infty$ such that $\operatorname{span}\varphi = \Lambda$.

Now choose $x \in \Gamma$ and consider $\boldsymbol{\varphi}(x)$. Let $\alpha \in (0, 1)$ and $\boldsymbol{\varphi}(x) = \alpha y + (1-\alpha)z$ for $y, z \in \operatorname{con}\boldsymbol{\varphi}(\Gamma) \subset \mathcal{M}_\Lambda \cong \mathbb{R}^{n+k}$. Due to Carathéodory [111, Thm. 2.29] there exist coefficients α_i, $\beta_i > 0$ such that $y = \sum_{i=1}^{n+k+1} \alpha_i \boldsymbol{\varphi}(y^{(i)})$ and $z = \sum_{i=1}^{n+k+1} \beta_i \boldsymbol{\varphi}(z^{(i)})$, $y^{(i)}, z(i) \in \Gamma$ with $\sum_{i=1}^{n+k+1} \alpha_i = 1$, $\sum_{i=1}^{n+k+1} \beta_i = 1$.

This implies that $\boldsymbol{\theta}(x) = \alpha \sum_{i=1}^{n+k+1} \alpha_i \boldsymbol{\theta}(y^{(i)}) + (1-\alpha) \sum_{i=1}^{n+k+1} \beta_i \boldsymbol{\theta}(z^{(i)})$. Extremality of $\boldsymbol{\theta}$ implies that $\boldsymbol{\theta}(x) = \sum_{i=1}^{n+k+1} \alpha_i \boldsymbol{\theta}(y^{(i)}) = \sum_{i=1}^{n+k+1} \beta_i \boldsymbol{\theta}(z^{(i)})$ and therefore $\boldsymbol{\theta}(x) = \boldsymbol{\theta}(y^{(i)}) = \boldsymbol{\theta}(z^{(i)})$. Since $\boldsymbol{\theta}$ is an extremal curve $\boldsymbol{\theta}$ is injective as well and therefore $x = y^{(i)} = z^{(i)}$. This implies that $\boldsymbol{\varphi}(x) = y = z$ and hence $\boldsymbol{\varphi}$ is an extremal curve. □

Lemma 2.1.17 (extremality of quadratic subspace)**.** *Let $\emptyset \neq \Gamma \subset \mathbb{R}^m$ be compact. Let $\varphi(x) = (1, x_1, x_2, \ldots, x_m, \|x\|^2)^T$. Assume that $\langle \varphi_0, \varphi_1, \ldots, \varphi_{m+2} \rangle \subset \Lambda$ for a vector space $\Lambda \subset \mathcal{C}(\Gamma)$. Then Λ is spanned by an extremal curve.*

Proof. Choose $\overline{x} \in \Gamma$. Let $\alpha \in (0,1)$ and $\boldsymbol{\varphi}(\overline{x}) = \alpha y + (1-\alpha)z$ for $y, z \in \operatorname{con} \boldsymbol{\varphi}(\Gamma) \subset \mathcal{M}_\Lambda \cong \mathbb{R}^{m+2}$. Due to Carathéodory [111, Thm. 2.29] there exist coefficients α_i, $\beta_i > 0$ such that $y = \sum_{i=1}^{m+3} \alpha_i \boldsymbol{\varphi}(y^{(i)})$ and $z = \sum_{i=1}^{m+3} \beta_i \boldsymbol{\varphi}(z^{(i)})$, $y^{(i)}, z^{(i)} \in \Gamma$ with $\sum_{i=1}^{m+3} \alpha_i = 1$, $\sum_{i=1}^{m+3} \beta_i = 1$.

Now choose $f(x) = \|x - \overline{x}\|^2$ and note $f \in \langle \varphi_0, \varphi_1, \ldots, \varphi_{m+2} \rangle$. Then we have:

$$\begin{aligned} 0 = f(\overline{x}) = \langle \boldsymbol{\varphi}(\overline{x}), f \rangle &= \left\langle \alpha \sum_{i=1}^{m+3} \alpha_i \boldsymbol{\varphi}(y^{(i)}) + (1-\alpha) \sum_{i=1}^{m+3} \beta_i \boldsymbol{\varphi}(z^{(i)}), f \right\rangle \\ &= \sum_{i=1}^{m+3} \alpha \cdot \alpha_i \cdot \langle \boldsymbol{\varphi}(y^{(i)}), f \rangle + \sum_{i=1}^{m+3} (1-\alpha) \cdot \beta_i \cdot \langle \boldsymbol{\varphi}(z^{(i)}), f \rangle \\ &= \sum_{i=1}^{m+2} \alpha \cdot \alpha_i \cdot f(y^{(i)}) + \sum_{i=1}^{m+2} (1-\alpha) \cdot \beta_i \cdot f(z^{(i)}) \end{aligned}$$

As $f(y^{(i)}) > 0$ for $y^{(i)} \neq \overline{x}$ and $\alpha > 0$ it holds that $y^{(i)} \neq \overline{x}$ implies $\alpha_i = 0$. The same is true for $z^{(i)}$ and β_i. Hence $\boldsymbol{\varphi}(\overline{x}) = y = z$, and therefore $\boldsymbol{\varphi} : \Gamma \to \mathcal{M}_\Lambda \cong \mathbb{R}^{m+2}$ is an extremal curve. In view of Lemma 2.1.16 Λ is spanned by an extremal curve. □

This shows that in a piecewise polynomial discretization with degree at least 2 the corresponding basis inherits the extreme point property from the extremality of the subspace of quadratic functions. Extremal curves are key to preserve the cost function when restricted to the set of discretized Diracs $\varphi(x)$:

Theorem 2.1.18. *Let $\Gamma \subset \mathbb{R}^m$ be nonempty and compact and let $f : \Gamma \to \mathbb{R}$ be lsc. Furthermore, let $\boldsymbol{\varphi} : \Gamma \to \mathcal{M}_\Lambda$ be an extremal curve. Then we have*

$$f_\Lambda^{**} \circ \boldsymbol{\varphi} = f \tag{2.30}$$

on Γ. In addition, we have that $\operatorname{con} f_\Lambda = f_\Lambda^{**}$.

Proof. Since f is lsc and Γ compact f is bounded from below, i.e., there is $C > -\infty$ so that $f(x) \geq C$ for all $x \in \Gamma$. We have $\operatorname{dom} f_\Lambda = \boldsymbol{\varphi}(\Gamma) \subset \mathcal{M}_\Lambda$ and in view of [111, Prop. 2.31] it holds for any $y \in \mathcal{M}_\Lambda$,

$$\begin{aligned} (\operatorname{con} f_\Lambda)(y) &= \inf \left\{ \sum_{i=1}^{n+1} \lambda_i f_\Lambda(y^{(i)}) : \lambda_i \geq 0, \sum_{i=1}^{n+1} \lambda_i = 1, \, y = \sum_{i=1}^{n+1} \lambda_i y^{(i)}, \, y^{(i)} \in \mathcal{M}_\Lambda \right\} \\ &= \inf \left\{ \sum_{i=1}^{n+1} \lambda_i f(x^{(i)}) : \lambda_i \geq 0, \sum_{i=1}^{n+1} \lambda_i = 1, \, y = \sum_{i=1}^{n+1} \lambda_i \boldsymbol{\varphi}(x^{(i)}), \, x^{(i)} \in \Gamma \right\} \geq C. \end{aligned}$$

Let $x \in \Gamma$. Since the only possible convex combination of the extreme point $\boldsymbol{\varphi}(x)$ from points $y^{(i)} \in \boldsymbol{\varphi}(\Gamma)$ is $\boldsymbol{\varphi}(x)$ itself, we have

$$\begin{aligned} (\operatorname{con} f_\Lambda)(\boldsymbol{\varphi}(x)) &= \inf \left\{ \sum_{i=1}^{n+1} \lambda_i f_\Lambda(y^{(i)}) : \lambda_i \geq 0, \sum_{i=1}^{n+1} \lambda_i = 1, \, \boldsymbol{\varphi}(x) = \sum_{i=1}^{n+1} \lambda_i y^{(i)}, \, y^{(i)} \in \boldsymbol{\varphi}(\Gamma) \right\} \\ &= f_\Lambda(\boldsymbol{\varphi}(x)) = f(x). \end{aligned}$$

This shows that $\operatorname{con} f_\Lambda \circ \boldsymbol{\varphi} = f$. Since $\operatorname{con} f_\Lambda$ is bounded from below $\operatorname{con} f_\Lambda$ is proper.

Next we show that f_Λ inherits its lower semicontinuity from f: Assume that $y \in \boldsymbol{\varphi}(\Gamma)$. Then there

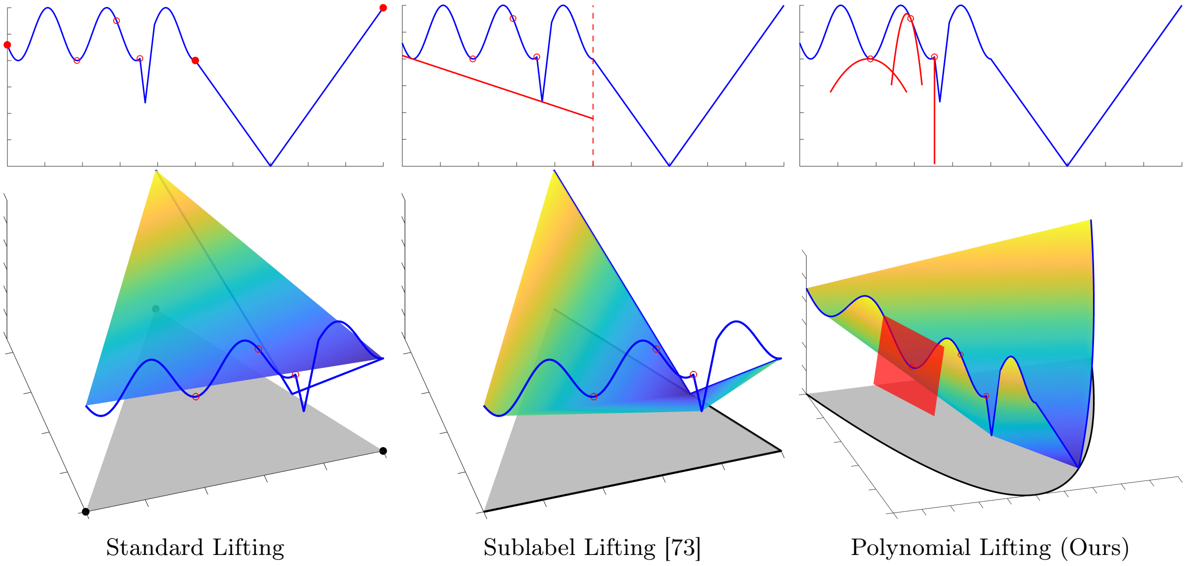


Figure 2.2: Lifted version f_Λ of the function f for 3 different choices of φ. In the top row the same nonconvex function f (blue curves) is depicted. The colored surfaces in the bottom row correspond to the lifted biconjugates f_Λ^{**}, where the gray shadow areas correspond to their domains $\operatorname{dom} f_\Lambda^{**} = \mathcal{P}_\Lambda$ and the blue curves correspond to the lifted cost f_Λ, see Eq. (2.24). The black curves in the bottom row correspond to the moment curve described by Diracs $\varphi(\Gamma)$, i.e., the domain of f_Λ. From left to right, cf. Example 2.1.10, Example 2.1.11 and Example 2.1.12. The nonlinear supporting dual functions $q_{\lambda,\beta}(x) = \langle \varphi(x), \lambda \rangle - \beta$ (red curves) to f in the top row (middle and right), are transformed into linear supporting hyperplanes $l_{\lambda,\beta}(y) = \langle y, \lambda \rangle - \beta$ (red surfaces) to f_Λ^{**} in the bottom row through the feature map φ. In the language of kernel methods, these functions can be interpreted as the nonlinear decision boundaries that separate individual points on the graph of f from the epigraph of f. Only in the most right case such a separation is possible. As a result the polynomial lifting preserves the nonconvex cost function $f_\Lambda^{**} \circ \varphi = f$ on $\varphi(\Gamma)$ whereas the 2-sparse lifting (middle) only leads to a piecewise convex under-approximation $f_\Lambda^{**} \circ \varphi \le f$. Figure taken from [Pub2].

exists $x \in \Gamma$ with $y = \boldsymbol{\varphi}(x)$ and we have due to the continuity of $\boldsymbol{\varphi}$ and $\boldsymbol{\varphi}^{-1}$:

$$\begin{aligned}
\liminf_{y' \to \boldsymbol{\varphi}(x)} \; f_\Lambda(y') &:= \lim_{\delta \to 0^+} \inf \, \{ f_\Lambda(y') : y' \in B_\delta(\boldsymbol{\varphi}(x)) \} \\
&= \lim_{\delta \to 0^+} \inf \, \{ f_\Lambda(\boldsymbol{\varphi}(x')) : \boldsymbol{\varphi}(x') \in B_\delta(\boldsymbol{\varphi}(x)) \} \\
&= \lim_{\epsilon \to 0^+} \inf \, \{ f(x') : x' \in B_\epsilon(x) \} \\
&= \liminf_{x' \to x} \; f(x') \\
&\ge f(x) = f_\Lambda(y).
\end{aligned}$$

Since Γ is compact and $\boldsymbol{\varphi}$ continuous the image $\boldsymbol{\varphi}(\Gamma) = \operatorname{dom} f_\Lambda$ is compact as well. Since f_Λ is also bounded from below it is coercive in the sense of [111, Def. 3.25] (also called super-coercive in other literature). Then we can invoke [111, Cor. 3.47] and deduce that $\operatorname{con} f_\Lambda$ is proper, lsc and convex. In view of [111, Thm. 11.1] we have $\operatorname{con} f_\Lambda = \operatorname{cl} \operatorname{con} f_\Lambda = f_\Lambda^{**}$. □

For a geometric intuition of this theorem we refer to Fig. 2.2. For a piecewise polynomial discretization with degree at least 2, due to extremality, the primal discretized energy $\mathbf{E}_\Lambda$ restricted to $\boldsymbol{\varphi}(\Gamma)^{\mathcal{V}}$ agrees with the original energy E. In particular, this implies that an obtained Dirac solution $(\boldsymbol{\varphi}(u_v^*))_{v \in \mathcal{V}}$ of the discretization corresponds to a solution of the original problem in the same way integer solutions of linear programming relaxations are certificates of optimality for the corresponding integer linear program.

Proposition 2.1.19. *Let $\emptyset \neq \Gamma \subset \mathbb{R}^m$. Let the metric $\mathbf{d}$ be induced by a norm and assume the space of linear functions on Γ is contained in Λ. Furthermore, assume Λ is spanned by an extremal curve*

$\boldsymbol{\varphi} : \Gamma \to \mathcal{M}_\Lambda$ and f_v lsc. Then for any $\mathbf{u} \in \mathcal{M}_\Lambda^{\mathcal{V}}$, with $\mathbf{u}_v = \boldsymbol{\varphi}(u_v)$, $u_v \in \Gamma$ the following identity holds true:

$$\mathbf{E}_\Lambda(\mathbf{u}) = E(u).$$

In particular, whenever $\hat{\mathbf{u}}$ is a solution of problem (dR-P) *such that $\hat{\mathbf{u}} \in \mathcal{M}_\Lambda^{\mathcal{V}}$, with $\hat{\mathbf{u}}_v = \boldsymbol{\varphi}(\hat{u}_v)$, for some $\hat{u}_v \in \Gamma$, $\hat{u}$ is a solution of the original problem* (P).

Proof. Let $\mathbf{d}$ be induced by some norm $\|\cdot\|$ and denote its dual norm by $\|\cdot\|_*$. As shown in Theorem 2.1.18, the unaries preserve the original cost functions at $\boldsymbol{\varphi}(\Gamma)$. Hence it remains to show that the for the pairwise costs it holds $\sigma_{\mathcal{K}\cap\Lambda}((\nabla_\Lambda \mathbf{u})_{(v,w)}) = \|u_v - u_w\|$. By assumption $(\nabla_\Lambda \mathbf{u})_{(v,w)} = \boldsymbol{\varphi}(u_v) - \boldsymbol{\varphi}(u_w)$. We rewrite

$$\sigma_{\mathcal{K}\cap\Lambda}(\boldsymbol{\varphi}(u_v) - \boldsymbol{\varphi}(u_w)) = \sup_{\lambda \in \mathrm{Lip}_{\mathbf{d}}(\Gamma)\cap\Lambda} \lambda(u_v) - \lambda(u_w) \leq \|u_v - u_w\|.$$

For any $u_v, u_w \in \Gamma$ we have

$$\|u_v - u_w\| = \sup_{p \in \mathbb{R}^m : \|p\|_* \leq 1} |\langle u_v - u_w, p\rangle| = \langle u_v - u_w, \hat{p}\rangle,$$

where $\hat{p}$ denotes the maximizer in the supremum, which exists due to the compactness of the unit ball in a finite-dimensional space. Define the linear function $\hat{\lambda} := \langle \cdot\,, \hat{p}\rangle$ and note that by assumption $\hat{\lambda} \in \Lambda$. In addition we have shown that $\hat{\lambda} \in \mathrm{Lip}_{\mathbf{d}}(\Gamma)$ is 1-Lipschitz. This implies $\|u_v - u_w\| = \hat{\lambda}(u_v) - \hat{\lambda}(u_w) \leq \sup_{\lambda \in \mathrm{Lip}_d(\Gamma)\cap\Lambda} \lambda(u_v) - \lambda(u_w) \leq \|u_v - u_w\|$ and hence equality holds. □

2.1.5.4 A Generalized Conjugacy Perspective

The previous results can be obtained from a generalized conjugacy point of view. In particular, the convex conjugate of the lifted function f_Λ^* is comprised by the notion of Φ-conjugacy, see [111, Ch. 11L*]:

Definition 2.1.20 (generalized conjugate functions)**.** *Let $\mathcal{X}$ and $\mathcal{Y}$ be nonempty sets. Let $\Phi : \mathcal{X} \times \mathcal{Y} \to \overline{\mathbb{R}}$ be any function. Let $f : \mathcal{X} \to \overline{\mathbb{R}}$. Then the Φ-conjugate of f on $\mathcal{Y}$ at $y \in \mathcal{Y}$ is defined by*

$$f^\Phi(y) := \sup_{x \in \mathcal{X}} \Phi(x, y) - f(x), \tag{2.31}$$

and the Φ-biconjugate of f back on $\mathcal{X}$ at $x \in \mathcal{X}$ is given by

$$f^{\Phi\Phi}(x) := \sup_{y \in \mathcal{Y}} \Phi(x, y) - f^\Phi(y). \tag{2.32}$$

We say that f is a Φ-envelope on $\mathcal{X}$ if f can be written in terms of a pointwise supremum of a collection of elementary functions $x \mapsto \Phi(x, y) - \alpha$, where $(\alpha, y) \in \overline{\mathbb{R}} \times \mathcal{Y}$ is the parameter element.

For $\mathcal{X} = \Gamma$, $\mathcal{Y} = \Lambda$, and $\Phi(x, y) = \langle \boldsymbol{\varphi}(x), y\rangle$, the convex conjugate f_Λ^* is identical to the Φ-conjugate f^Φ of f, while its biconjugate f_Λ^{**} is the tightest lsc convex extension of f_Λ to $\operatorname{con} \boldsymbol{\varphi}(\Gamma)$. The Φ-biconjugate $f^{\Phi\Phi}$ of f at a point $x \in \Gamma$ is equal to the classical biconjugate of f_Λ, evaluated at $\boldsymbol{\varphi}(x)$, i.e., $f^{\Phi\Phi} = f_\Lambda^{**} \circ \varphi$ on Γ, showing that the Φ-biconjugate is a convexly composite function and, therefore, it is nonconvex in general.

As a consequence of [111, Ex. 11.63], the considered Φ-conjugacy can be interpreted in terms of under-approximation by functions in Λ. In analogy to the biconjugate f^{**}, which is the pointwise supremum of affine-linear functions majorized by f, the Φ-biconjugate $f^{\Phi\Phi}$ is the pointwise supremum of functions in Λ up to constant translation majorized by f. This point of view also relates $(f^\Phi)^*$ and $f^{\Phi\Phi}$ by each other.

Remark 2.1.21. The function $(f^\Phi)^* = f_\Lambda^{**}$ is the pointwise supremum of all affine-linear functions $l_{\lambda,\beta} := \langle \cdot\,, \lambda\rangle - \beta$ for which $q_{\lambda,\beta} := \Phi(\cdot\,, \lambda) - \beta$ is majorized by f. This can be seen as follows: The Legendre–Fenchel conjugate can be characterized via the identity $(f^\Phi)^*(y) = \sup_{(\lambda,\beta)\in \mathrm{epi}(f^\Phi)} \langle y, \lambda\rangle - \beta$. The observation now follows from the fact that $(\lambda, \beta) \in \mathrm{epi}(f^\Phi)$ if and only if $q_{\lambda,\beta}$ is majorized by f.

The correspondence between f_Λ^{**} and $(f^\Phi)^*$ and between the minorizers $l_{\lambda,\beta}$ and $q_{\lambda,\beta}$ is illustrated in Fig. 2.2. Note that this is closely related to the idea of feature maps $\boldsymbol{\varphi}$ in linear classifiers.

Theorem 2.1.18 identifies all lsc functions $f : X \to \mathbb{R}$ as Φ-envelopes whenever $\boldsymbol{\varphi}$ is extremal:

Corollary 2.1.22. *Let $\Gamma \subset \mathbb{R}^m$ be nonempty and compact and let $f : \Gamma \to \mathbb{R}$ be bounded from below. Furthermore, let $\boldsymbol{\varphi} : \Gamma \to \mathcal{M}_\Lambda$ be an extremal curve. Then we have*

$$f^{\Phi\Phi} = f_\Lambda^{**} \circ \boldsymbol{\varphi} = \operatorname{cl} f, \tag{2.33}$$

on Γ.

Proof. Since f is finite-valued and bounded from below on Γ we have $(\operatorname{cl} f)(x) > -\infty$ for $x \in \Gamma$ and therefore $\operatorname{cl} f$ is finite-valued. By [111, Ex. 11.63] $f^{\Phi\Phi}$ is the largest Φ-envelope below f. Since $\boldsymbol{\varphi}$ is continuous, $f^{\Phi\Phi} = f_\Lambda^{**} \circ \boldsymbol{\varphi}$ is lsc. Since $\operatorname{cl} f$ is the largest lsc function below f we have $f^{\Phi\Phi} \leq \operatorname{cl} f$. Since $\operatorname{cl} f \leq f$ we also have $(\operatorname{cl} f)^{\Phi\Phi} \leq f^{\Phi\Phi}$. Invoking Theorem 2.1.18 we have

$$\operatorname{cl} f = (\operatorname{cl} f)^{\Phi\Phi} \leq f^{\Phi\Phi} \leq \operatorname{cl} f,$$

on Γ. Therefore $f^{\Phi\Phi} = \operatorname{cl} f$ on Γ. □

Up to the presence of the compact set Γ, this result generalizes the basic quadratic transform [111, Ex. 11.66] originally due to [101, Prop. 3.4] (for lsc functions only), which is obtained by choosing $\varphi(x) = (x_1, x_2, \ldots, x_m, \|x\|^2)^T$. In [7, Thm. 1] a similar duality formula is shown for Φ-couplings of a certain "needle-type". Our result is instead based on the extremality condition, which, from a primal point of view, captures an intuitive and sharp (sufficient) condition for the above result for the one-sided linear couplings we consider. For the component functions of φ being the hat basis, see Example 2.1.11, the class of Φ-envelopes are the piecewise convex functions, see, Fig. 2.2 middle.

We can formulate an interpretation of f_Λ^{**} in terms of probability measures:

Proposition 2.1.23. *Let $f \in \mathcal{C}(\Gamma)$ and $\Lambda \subset \mathcal{C}(\Gamma)$. Then the biconjugate of f_Λ is*

$$f_\Lambda^{**}(y) = \min\{\, \langle \mu, f \rangle \mid \mu \in \mathcal{P}(\Gamma),\ [\mu]_\Lambda = y \,\}, \tag{2.34}$$

with $\operatorname{dom} f_\Lambda^{**} = \mathcal{P}_\Lambda$.

Proof. We first define $g(y) := \inf\{\, \langle \mu, f \rangle \mid \mu \in \mathcal{P}(\Gamma),\ [\mu]_\Lambda = y \,\}$. We will show that g is proper convex lsc.

Proposition 2.1.8 implies $\operatorname{dom} g = \mathcal{P}_\Lambda$ and $g(y) \geq \min_{x\in\Gamma} f(x)$ by Lemma 2.1.1. Hence g is proper.

For showing convexity of g we will show that the epigraph $\operatorname{epi} g := \{\, (y, \alpha) \in \mathcal{M}_\Lambda \times \mathbb{R} \mid g(y) \leq \alpha \,\}$ of g is convex. Let $(y, \alpha), (z, \beta) \in \operatorname{epi} g$. Then there exist $\mu, \nu \in \mathcal{P}(\Gamma)$ such that $y = [\mu]_\Lambda, z = [\nu]_\Lambda$ and $\langle \mu, f \rangle \leq \alpha, \langle \nu, f \rangle \leq \beta$. By convexity of $\mathcal{P}(\Gamma)$ we get $\lambda\mu + (1-\lambda)\nu \in \mathcal{P}(\Gamma)$ and $[\lambda\mu + (1-\lambda)\nu]_\Lambda = \lambda y + (1-\lambda)z$ for $\lambda \in [0, 1]$. Furthermore $\langle \lambda\mu + (1-\lambda)\nu \rangle = \lambda\langle \mu, f \rangle + (1-\lambda)\langle \nu, f \rangle \leq \lambda\alpha + (1-\lambda)\beta$. Hence $(\lambda y + (1-\lambda)z, \lambda\alpha + (1-\lambda)\beta) \in \operatorname{epi} g$ and g is convex.

Now consider $(y, \alpha) \in \mathcal{M}_\Lambda \times \mathbb{R}$ and a sequence $(y^t, \alpha^t) \in \operatorname{epi} g$ such that $(y^t, \alpha^t) \to (y, \alpha)$. Then there exists a sequence $\mu^t \in \mathcal{P}(\Gamma)$ such that $[\mu^t]_\Lambda = y^t$ and $\langle \mu^t, f \rangle \leq \alpha^t$. Since Γ is compact, again by Prokhorov's Theorem, see [119, Sec. 1.1], there exists a weakly* convergent subsequence $\mu^{t_j} \overset{*}{\rightharpoonup} \hat{\mu}$ for some $\hat{\mu} \in \mathcal{P}(\Gamma)$. Therefore $[\hat{\mu}]_\Lambda = \lim_{j\to\infty}[\mu^{t_j}]_\Lambda = \lim_{j\to\infty} y^{t_j} = y$ where the first equality follows from the definition of the weak* topology and the fact that $\Lambda \subset \mathcal{C}(\Gamma)$. In addition $\alpha - \langle \hat{\mu}, f \rangle = \lim_{j\to\infty} \alpha^{t_j} - \langle \mu^{t_j}, f \rangle \geq 0$ as $f \in \mathcal{C}(\Gamma)$. Therefore $(y, \alpha) \in \operatorname{epi} g$, $\operatorname{epi} g$ is closed and thus g is lsc. Furthermore consider a sequence $\mu^t \in \mathcal{P}(\Gamma)$ such that $[\mu^t]_\Lambda = y$ for some $y \in \mathcal{P}_\Lambda$ and $\lim_{t\to\infty}\langle \mu^t, f \rangle = \inf\{\, \langle \mu, f \rangle \mid \mu \in \mathcal{P}(\Gamma),\ [\mu]_\Lambda = y \,\}$. Again by Prokhorov's Theorem there exists $\hat{\mu} \in \mathcal{P}(\Gamma)$ and a subsequence $\mu^{t_j} \overset{*}{\rightharpoonup} \hat{\mu}$. Furthermore $[\hat{\mu}]_\Lambda = \lim_{j\to\infty}[\mu^{t_j}]_\Lambda = \lim_{j\to\infty} y = y$ and $\langle \hat{\mu}, f \rangle = \lim_{j\to\infty}\langle \mu^{t_j}, f \rangle = \inf\{\, \langle \mu, f \rangle \mid \mu \in \mathcal{P}(\Gamma),\ [\mu]_\Lambda = y \,\}$ by construction. Hence $\hat{\mu}$ is a minimizer of $\{\, \langle \mu, f \rangle \mid \mu \in \mathcal{P}(\Gamma),\ [\mu]_\Lambda = y \,\}$.

Now we calculate the conjugate of g as

$$\begin{aligned}
g^*(\lambda) &= \sup_{y \in \mathcal{P}_\Lambda} \langle y, \lambda \rangle - \inf\{\, \langle \mu, f \rangle \mid \mu \in \mathcal{P}(\Gamma),\ [\mu]_\Lambda = y \,\} \\
&= \sup_{y \in \mathcal{P}_\Lambda} \sup_{\mu \in \mathcal{P}(\Gamma),\ [\mu]_\Lambda = y} \langle y, \lambda \rangle - \langle \mu, f \rangle \\
&= \sup_{\mu \in \mathcal{P}(\Gamma)} \langle \mu, \lambda \rangle - \langle \mu, f \rangle \\
&= \max_{x \in \Gamma} (\lambda - f)(x) \\
&= f_\Lambda^*(\lambda).
\end{aligned}$$

The claim now follows from $f_\Lambda^{**} = g^{**} = g$ as g is proper convex lsc. □

2.1.6 A Tractable Conic Program for MRFs

2.1.6.1 Nonnegativity and Moments

After discretization a next step to obtain a practical implementation is to derive finite characterizations of the lifted biconjugates f_Λ^{**} and the constraint set $\mathcal{K} \cap \Lambda$. We will show that the formulations can be rewritten in terms of a semi-infinite conic program which can be implemented using semidefinite programming in the piecewise polynomial case.

For now let $f_v \in \Lambda$. Equation (2.27) then shows, that the challenging part is to characterize the moment space $\mathcal{P}_\Lambda$: The following result shows that up to normalization, $\mathcal{P}_\Lambda$ can be written in terms of the dual cone of the cone of functions in Λ that are nonnegative on Γ.

Lemma 2.1.24. *Let $(\mathcal{M}_\Lambda)_+$ be the cone of moments of nonnegative measures as defined in Eq.* (2.21) *and let $\mathcal{N}_\Lambda$ be the cone of the coefficients of the functions in $\Lambda = \langle \varphi_0, \ldots, \varphi_n \rangle$ that are nonnegative on Γ defined as:*

$$\mathcal{N}_\Lambda := \{\, \lambda \in \Lambda \mid \lambda(x) \geq 0,\ \forall\, x \in \Gamma \,\}. \tag{2.35}$$

Then $(\mathcal{M}_\Lambda)_+$ is equal to $\mathcal{N}_\Lambda^*$, where $\mathcal{N}_\Lambda^*$ denotes the dual cone of $\mathcal{N}_\Lambda$.

If, in addition, $\varphi_0 \equiv 1$ we also have $\mathcal{P}_\Lambda = \{\, y \in (\mathcal{M}_\Lambda)_+ \mid y_0 = 1 \,\}$.

Proof. Let $y \in (\mathcal{M}_\Lambda)_+$. This means there exists $\mu \in \mathcal{M}_+(\Gamma)$ such that $y = [\mu]_\Lambda$. Let $p \in \mathcal{N}_\Lambda$. By the definition of nonnegative measures $\mathcal{M}_+(\Gamma)$ it holds $\langle y, p \rangle = \langle [\mu]_\Lambda, p \rangle = \langle \mu, p \rangle > 0$. Since $p \in \mathcal{N}_\Lambda$ was an arbitrary choice from $\mathcal{N}_\Lambda$ we have $y \in \mathcal{N}_\Lambda^*$.

Next we show $(\mathcal{M}_\Lambda)_+^* \subseteq \mathcal{N}_\Lambda$ as this implies $\mathcal{N}_\Lambda^* \subseteq (\mathcal{M}_\Lambda)_+^{**} = (\mathcal{M}_\Lambda)_+$, where the last equality holds since $(\mathcal{M}_\Lambda)_+$ is convex by definition of convexity and closed by the same argument used in the proof of Proposition 2.1.8. Take $p \in (\mathcal{M}_\Lambda)_+^*$. For $x \in \Gamma$ it holds $p(x) = \langle \delta_x, p \rangle = \langle [\delta_x]_\Lambda, p \rangle > 0$. Since the choice $x \in \Gamma$ was arbitrary we have $p(x) \geq 0$ for all $x \in \Gamma$ and therefore $p \in \mathcal{N}_\Lambda$.

Finally, $\mathcal{P}_\Lambda = \{\, y \in (\mathcal{M}_\Lambda)_+ \mid y_0 = 1 \,\}$ follows from the fact that $\mu \in \mathcal{M}(\Gamma)$ is an element of $\mathcal{P}(\Gamma)$ if and only if $\mu \in \mathcal{M}_+(\Gamma)$ and $\langle \mu, \varphi_0 \rangle = 1$. □

Before we specialize Λ to the space of polynomials, we derive a cone programming formulation of the Lipschitz constraints $\lambda_{(v,w)} \in \mathrm{Lip}_{\mathbf{d}}(\Gamma)$. Here we restrict $\Gamma = [a, b]$ to be a compact interval. In the following, we provide an implementation for two specific metrics. As before, this boils down to nonnegativity of functions: Firstly, we consider total variation regularization, i.e. $\mathbf{d}(x, y) = |x - y|$: Assume that Λ is closed under differentiation, i.e., φ is differentiable and $\varphi_k' \in \Lambda$. Then, the condition $\lambda_{(v,w)} \in \mathrm{Lip}_{\mathbf{d}}([a, b])$ can be phrased in terms of the constraints $-1 \leq \lambda_{(v,w)}'(x) \leq 1$ for all $x \in [a, b]$, where $\lambda_{(v,w)}'$ is the derivative of $\lambda_{(v,w)}$. Equivalently, this means that the coefficients of the functions $1 + \lambda_{(v,w)}'$ and $1 - \lambda_{(v,w)}'$ are in $\mathcal{N}_\Lambda$. Secondly, we consider Potts regularization, i.e., $\mathbf{d}(x, y) = [\![x = y]\!]$. Since $\lambda_{(v,w)}$ is a univariate function, and constant terms in the dual variable do not matter, the condition $\lambda_{(v,w)} \in \mathrm{Lip}_{\mathbf{d}}([a, b])$ can be equivalently phrased as $0 \leq \lambda_{(v,w)} \leq 1$, see, [136, Ex. 1.17]. Equivalently, this means that the coefficients of the functions $\lambda_{(v,w)}$ and $1 - \lambda_{(v,w)}$ are in $\mathcal{N}_\Lambda$.

2.1.6.2 Semidefinite Programming and Nonnegative Polynomials

As we have seen in the previous section, an important ingredient for a tractable formulation is the efficient characterization of nonnegativity of functions in a finite-dimensional subspace $\Lambda \subset \mathcal{C}(\Gamma)$. A promising choice of Λ in that regards is the space of polynomials. Indeed, the characterization of nonnegativity of polynomials is a fundamental problem in *convex algebraic geometry* surveyed in [12]: Let $\mathbb{R}[x_1, \ldots, x_m]$ denote the ring of possibly multivariate polynomials with $p \in \mathbb{R}[x_1, \ldots, x_m]$. Then $p = \sum_{\alpha \in I} p_\alpha x^\alpha$ for monomials x^α. Let $\deg p$ denote its degree. A key result from real algebraic geometry is the Positivstellensatz due to [64] and [126] refined in [121] and [103]. It characterizes polynomials $p \in \mathbb{R}[x_1, \ldots, x_m]$ that are positive on semi-algebraic sets Γ, i.e. $p(x) > 0$ for all $x \in \Gamma$, where Γ is defined in terms of polynomial inequalities.

Key to such results is a *certificate* of nonnegativity of the polynomial p that involves *sum-of-squares* (SOS) multipliers q, where q is SOS if $q(x) = \sum_{i=1}^N q_i^2(x)$ for polynomials $q_i \in \mathbb{R}[x_1, \ldots, x_m]$. For intervals $[a, b]$, thanks to [12, Thm. 3.72] originally due to [102, Cor. 2.3] we have following result:

Lemma 2.1.25. *Let $a < b$. Then the univariate polynomial $p \in \mathbb{R}[x]$ is nonnegative on $[a, b]$ if and only if it can be written as*

$$p(x) = \begin{cases} s(x) + (x-a) \cdot (b-x) \cdot t(x) & \text{if } \deg p \text{ is even,} \\ (x-a) \cdot s(x) + (b-x) \cdot t(x) & \text{if } \deg p \text{ is odd,} \end{cases} \tag{2.36}$$

where $s, t \in \mathbb{R}[x]$ are sum of squares. If $\deg p = 2n$, then we have $\deg s \le 2n$, $\deg t \le 2n - 2$, while if $\deg p = 2n + 1$, then $\deg s \le 2n, \deg t \le 2n$.

Remarkably, the above result provides us with explicit upper bounds of the degrees of the SOS multipliers s and t that are important to derive a practical implementation: Then the SOS constraints can be formulated in terms of semidefinite and affine inequalities: We adopt [12, Lem. 3.33] and [12, Lem. 3.34]:

Lemma 2.1.26. *A univariate polynomial $p \in \mathbb{R}[x]$ with $\deg p = 2n$, $n \ge 0$ is SOS if and only if there exists a positive semidefinite matrix $Q \in \mathbb{R}^{(n+1)\times(n+1)}$ such that*

$$z^\top Q z = p(x),$$

where $z^\top := (1, x, x^2, \ldots, x^n)$. Furthermore the equality is equivalent to

$$p_k = \langle A_{n,k}, Q \rangle = \sum_{\substack{1 \le i,j \le n+1, \\ i+j-2=k}} Q_{ij}, \quad \forall\, 0 \le k \le 2n \tag{2.37}$$

where $A_{n,k} \in \mathbb{R}^{(n+1)\times(n+1)}$, $0 \le k \le 2n$ is a Hankel matrix whose $(k+1)^{th}$ skew-diagonal is 1 and 0 elsewhere.

In analogy to [12, Lem. 3.148] and [12, Lem. 3.149], invoking the results above SDP-duality yields the following compact representation of $(\mathcal{M}_\Lambda)_+$:

Lemma 2.1.27. *Let $n \ge 0$. For $\dim \mathcal{M}_\Lambda = 2n + 2$, $y \in (\mathcal{M}_\Lambda)_+$ if and only if*

$$bM_{0,n}(y) \succeq M_{1,n}(y) \succeq aM_{0,n}(y), \tag{2.38}$$

for Hankel matrices

$$M_{i,n}(y) := \begin{bmatrix} y_i & y_{i+1} & \cdots & y_{i+n} \\ y_{i+1} & y_{i+2} & \cdots & y_{i+n+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{i+n} & y_{i+n+1} & \cdots & y_{i+2n} \end{bmatrix}. \tag{2.39}$$

For $\dim \mathcal{M}_\Lambda = 2n + 1$, $y \in (\mathcal{M}_\Lambda)_+$ if and only if

$$M_{0,n}(y) \succeq 0, \tag{2.40}$$

$$(a+b)\, M_{1,n-1}(y) - abM_{0,n-1}(y) \succeq M_{2,n-1}(y). \tag{2.41}$$

Proof. By Eq. (2.21) moment vectors of non-negative measures can be directly derived as the dual cone of non-negative polynomials. Therefore we can use SDP duality for characterizing $(\mathcal{M}_\Lambda)_+$:

$$\iota_{(\mathcal{M}_\Lambda)_+}(y) = \iota_{\mathcal{N}_\Lambda^\circ}(-y) = \sup_x \langle -y, x\rangle - \iota_{\mathcal{N}_\Lambda}(x), \tag{2.42}$$

where $x \in \mathbb{R}^{2n+2}$ for the odd case and $x \in \mathbb{R}^{2n+1}$ for the even case and $\mathcal{N}_\Lambda^\circ$ denotes the polar cone of $\mathcal{N}_\Lambda$. For notational convenience we will identify matrices emerging from the SOS characterization as vectors. We denote by A_n the stacked matrices $A_{n,k}$ from Lemma 2.1.26 adapted to vectorized matrices, i.e. $A \in \mathbb{R}^{(2n+1)\times(n+1)^2}$. Therefore the SOS constraint of a polynomial p of degree $2n$ can be stated as

$$\inf_{\substack{P \in \mathbb{R}^{(n+1)^2} \\ p = A_n P}} \iota_{\mathcal{S}_+^{n+1}}(P) = 0, \tag{2.43}$$

where $\mathcal{S}_+^n$ is the vectorized cone of positive semi-definite matrices in $\mathbb{R}^{n\times n}$. Non-negativity of a polynomial p on an interval $[a,b]$ can according to Lemma 2.1.25 hence be written as

$$\iota_{\mathcal{N}_\Lambda}(p) = \inf_{\substack{S,T \in \mathbb{R}^{n^2} \\ p = QA(S,T)^T}} \iota_{\mathcal{S}_+}(S) + \iota_{\mathcal{S}_+}(T), \tag{2.44}$$

where for the odd case $\deg p = 2n+1$ we pick $A = (A_n \, A_n)$ and

$$Q = \left(\begin{array}{cccc|cccc}
-a & & & & b & & & \\
1 & -a & & & -1 & b & & \\
 & \ddots & \ddots & & & \ddots & \ddots & \\
 & & 1 & -a & & & -1 & b \\
 & & & 1 & & & & -1
\end{array}\right) \in \mathbb{R}^{(2n+2)\times(2\cdot(2n+1))} \tag{2.45}$$

and for the even case $\deg p = 2n$ we pick $A = (A_n \, A_{n-1})$ and

$$Q = \left(\begin{array}{ccccc|ccc}
1 & & & & & -ab & & \\
 & \ddots & & & & a+b & \ddots & \\
 & & \ddots & & & -1 & \ddots & -ab \\
 & & & \ddots & & & \ddots & a+b \\
 & & & & 1 & & & -1
\end{array}\right) \in \mathbb{R}^{(2n+1)\times((2n+1)+(2n-1))}. \tag{2.46}$$

Now we can formulate the moment constraints as

$$\begin{aligned}
\iota_{(\mathcal{M}_\Lambda)_+}(y) &= \sup_x \langle -y, x\rangle - \inf_{\substack{S,T \in \mathbb{R}^{n^2} \\ x = QA(S,T)^T}} \iota_{\mathcal{S}_+}(S) + \iota_{\mathcal{S}_+}(T) & (2.47)\\
&= \sup_x \langle -y, x\rangle - \left((\iota_{\mathcal{S}_-} \times \iota_{\mathcal{S}_-}) \circ A^T Q^T\right)^* (x) & (2.48)\\
&= \left(\iota_{\mathcal{S}_-} \times \iota_{\mathcal{S}_-}\right)\left(-A^T Q^T y\right) & (2.49)\\
&= \left(\iota_{\mathcal{S}_+} \times \iota_{\mathcal{S}_+}\right)\left(A^T Q^T y\right), & (2.50)
\end{aligned}$$

where we used the image function rule and $\mathcal{S}_-$ is the polar cone of $\mathcal{S}_+$. We conclude by remarking that A^T exactly yields vectorized Hankel matrices. □

2.1.6.3 Convergence of a Piecewise Polynomial Hierarchy

In experiments, we will discretize the dual problem with a piecewise polynomial family of functions. For intervals, the following proposition shows that either by increasing the number of pieces or the degree of the polynomial the primal-dual gap can be reduced. In our case we approximate the Lipschitz dual variable in terms of a Lipschitz spline. As a consequence existing results such as [41, Thm. 2] do not apply. Instead, we use a construction based on Bernstein-polynomials. Then the result follows from [18, Thm. 1].

Proposition 2.1.28. *Assume that $\Gamma = [a, b] \subset \mathbb{R}$, $a < b$, and let the metric d be given by $d(x, y) = |x-y|$. Furthermore, let $\Lambda \subset \mathcal{C}(\Gamma)$ be the space spanned by continuous piecewise polynomials on intervals $[t_i, t_{i+1}]$ defined by a regularly spaced grid with nodes given by $t_i = a + (b-a) \cdot (i-1)/K$, $i = 1, \dots, K+1$. Then the optimality gap satisfies:*

$$\text{(P)} - \text{(dR-D)} = \mathcal{O}(1/(K \cdot \sqrt{\deg})),$$

where deg *is the degree of the polynomial on each piece.*

Proof. We consider the discretized dual problem (dR-D) where $\mathcal{K}_\Lambda$ is the set of coefficients corresponding to 1-Lipschitz piecewise polynomials on $[a, b]$ of degree deg with K pieces. Also recall that we have the following relations between the dual and primal problems: (dR-D) $\leq$ (R-D) $=$ (R-P) where the last equality follows from strong duality Proposition 2.1.6.

Now, let us denote a maximizer of (R-D) as $\hat{\lambda} \in \operatorname{Lip}_d(\Gamma)^{\mathcal{E}}$. Existence of such a dual maximizer follows by Proposition 2.1.6. Then, one has for any $\lambda \in \Lambda^{\mathcal{E}}$:

$$\begin{aligned} \min_{x \in \Gamma}\ f_v(x) - (\operatorname{Div} \hat{\lambda})_v(x) &= \min_{x \in \Gamma}\ f_v(x) - (\operatorname{Div} \lambda)_v(x) - (\operatorname{Div} \hat{\lambda})_v(x) + (\operatorname{Div} \lambda)_v(x) \\ &\leq \min_{x \in \Gamma}\ f_v(x) - (\operatorname{Div} \lambda)_v(x) + \| - (\operatorname{Div}(\hat{\lambda} - \lambda))_v \|_\infty. \end{aligned} \tag{2.51}$$

This allows us to bound the optimality gap by:

$$\begin{aligned} \text{(R-D)} - \text{(dR-D)} &\leq \sum_{v \in \mathcal{V}} \| - (\operatorname{Div}(\lambda - \hat{\lambda}))_v \|_\infty \\ &\leq \sum_{v \in \mathcal{V}} |d(v)| \cdot \sup_{e \in \mathcal{E}}\ \|\lambda_e - \hat{\lambda}_e\|_\infty \\ &\leq 2|\mathcal{E}| \cdot \sup_{e \in \mathcal{E}} \|\lambda_e - \hat{\lambda}_e\|_\infty, \end{aligned} \tag{2.52}$$

where $d(v)$ denotes the degree of the vertex v.

For a L-Lipschitz function $f : [0, 1] \to \mathbb{R}$ there exists a Bernstein polynomial $p : [0, 1] \to \mathbb{R}$ with $p(0) = f(0)$ and $p(1) = f(1)$ such that $\|p - f\|_\infty \leq \frac{5L}{4} \deg^{-1/2}$ [79, Thm. 1.6.1]. By [18, Thm. 1], this polynomial is L-Lipschitz as well. For each $e \in \mathcal{E}$ we pick the coefficients of the function λ_e such that it approximates the optimal dual variable $\hat{\lambda}_e$ with such a polynomial individually on each interval $[t_i, t_{i+1}]$. Then one obtains an overall 1-Lipschitz polynomial with the following bound:

$$\|\lambda_e - \hat{\lambda}_e\|_\infty \leq \frac{5(b-a)}{4K\sqrt{\deg}}. \tag{2.53}$$

Inserting this into Eq. (2.52) yields via (R-P) = (R-D):

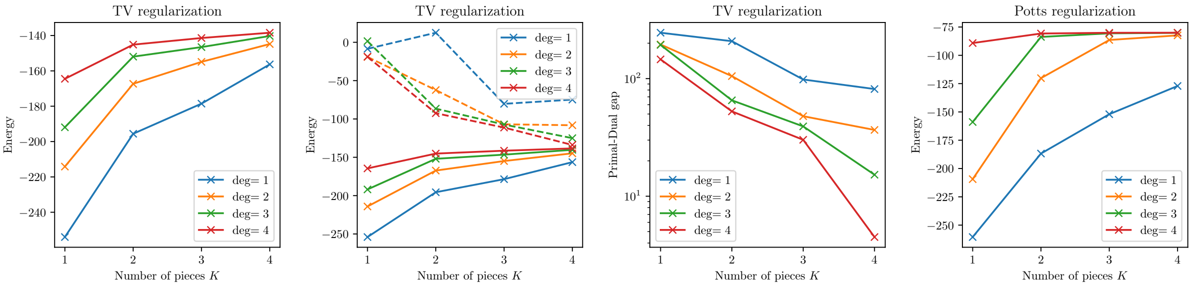
$$\text{(R-P)} - \text{(dR-D)} \leq |\mathcal{E}| \frac{5(b-a)}{2K\sqrt{\deg}}, \tag{2.54}$$

which due to tightness (R-P) = (P) from Proposition 2.1.2 gives the stated $\mathcal{O}(1/(K \cdot \sqrt{\deg}))$ rate. □

2.1.6.4 A First-Order Primal-Dual Algorithm

We are now ready to describe the algorithm for solving the resulting semidefinite program. We first consider the case $f_v \in \Lambda$ and Λ is the space of univariate polynomials. We propose to use the PDHG [27] algorithm, as it can exploit the partially separable structure of our SDP. The primal-dual algorithm optimizes the problem (dR-P) via alternating projected gradient descent/ascent steps applied to the saddle-point formulation of (dR-P):

$$\min_{\mathbf{u} \in (\mathcal{P}_\Lambda)^{\mathcal{V}}}\ \max_{\lambda \in (\mathcal{K} \cap \Lambda)^{\mathcal{E}}}\ \langle \mathbf{u}, f - \operatorname{Div}_\Lambda \lambda \rangle, \tag{2.55}$$



(a) (b) (c) (d)

Figure 2.3: Primal and dual energies for MAP-inference in a continuous MRF with TV regularization using a piecewise polynomial hierarchy of dual variables. (a) shows the dual energy for TV. In (b) the dashed lines correspond to the primal energy at the rounded solution and the solid lines correspond to the dual energy. (c) shows the gap between the nonconvex primal energy at the rounded solution and the dual energy for TV regularization. (d) shows the dual energies for Potts regularization. Figure taken from [Pub2].

which is obtained by expanding the support function in problem (dR-P) and substituting the expression (2.27) for the lifted biconjugates. Writing $F(u) := \langle u, f\rangle + \iota_{(\mathcal{P}_\Lambda)^\mathcal{V}}$ and $G^*(\lambda) := \iota_{(\mathcal{K}\cap\Lambda)^\mathcal{E}}$ we can rewrite Eq. (2.55) as

$$\min_{\mathbf{u}} \max_{\lambda} \quad F(u) + \langle \mathbf{u}, \operatorname{Div}_\Lambda \lambda\rangle - G^*(\lambda). \tag{2.56}$$

The PDHG algorithm at iteration $i+1$ performs the following steps

$$\lambda^{i+1} = \operatorname{prox}_{\tau_\mathrm{p} G^*}(\lambda^i + \tau_\mathrm{p} \nabla_\Lambda \overline{\mathbf{u}}^i), \tag{2.57}$$

$$\mathbf{u}^{i+1} = \operatorname{prox}_{\tau_\mathrm{d} F}(\mathbf{u}^i - \tau_\mathrm{d} \operatorname{Div}_\Lambda \lambda^{i+1}), \tag{2.58}$$

$$\overline{\mathbf{u}}^{i+1} = \mathbf{u}^{i+1} + (\mathbf{u}^{i+1} - \mathbf{u}^i), \tag{2.59}$$

where $\operatorname{prox}_g(x)$ is the proximal operator $\operatorname{prox}_g(x) := \arg\min_{\bar{x}} g(\bar{x}) + \frac{1}{2}\|\bar{x} - x\|^2$ of a proper, lsc, convex function g. In each iteration the algorithm performs a projected gradient ascent step in the dual λ followed by a projected gradient descent step in the primal variable $\mathbf{u}$. Subsequently it performs an extrapolation step in the primal. The algorithm converges to a saddle point $(\hat{\mathbf{u}}, \hat{\lambda})$ for step sizes $\tau_\mathrm{d}, \tau_\mathrm{p} \in \mathbb{R}_+$ such that $\tau_\mathrm{d}\tau_\mathrm{p}\|\nabla_\Lambda\|^2 < 1$. The projections onto the sets $(\mathcal{P}_\Lambda)^\mathcal{V}$ and $(\mathcal{K}\cap\Lambda)^\mathcal{E}$ are separable and can therefore be carried out in parallel on a GPU using the SDP characterizations derived above. For practicality, we introduce additional auxiliary variables and linear constraints to decouple the affine constraints (2.37) and the SDP constraints. The projection operator of the semidefinite cone can then be solved using an eigenvalue decomposition.

2.1.6.5 Piecewise Polynomial Duals and Nonlinear Lifted Biconjugates

The polynomial discretization can be extended by means of a continuous piecewise polynomial representation of the dual variables resulting in a possibly more accurate approximation of the dual subspace Λ. Then, both, nonnegativity and Lipschitz continuity can be enforced on each piece Γ_k individually. Continuity of the piecewise polynomial dual variables can be enforced via linear constraints. The corresponding primal variable y belongs to $y \in (\mathcal{M}_{\Lambda_1})_+ \times \cdots \times (\mathcal{M}_{\Lambda_K})_+$. Then the restriction that y is a moment vector of a probability measure supported on the whole space Γ yields an additional sum-to-one constraint on the 0th moments $1 = \sum_{k=1}^K y_{k,0}$.

Another issue to address is when $f_v \notin \Lambda$ which results in a nonlinear lifted biconjugate over the moment-space as in Fig. 2.2.

The formulation which is derived next addresses both: In particular it allows one to choose Λ independently from f_v which can even be discontinuous, as long as f_v has a piecewise polynomial structure. Key to the formulation is to rewrite the inner minimum in the dual formulation (dR-D) exploiting a duality between nonnegativity and minimization of functions: Let $\Gamma = [a, b]$, $a < b$ be a compact interval. Let $a = \gamma_1 < \gamma_2 < \gamma_3 < \cdots < \gamma_{K+1} = b$ be a sequence of knots, where $\Gamma_k := [\gamma_k, \gamma_{k+1}]$. Let Λ_k be the space of univariate polynomials on Γ_k with some maximum degree n. Let $f : \Gamma \to \mathbb{R}$ be a possibly discontinuous lsc piecewise polynomial function defined by $f(x) = \min_{1 \leq k \leq K} f_k(x) + \iota_{\Gamma_k}(x)$ with $f_k \in \Lambda_k$. First observe the following duality between nonnegativity and minimization of a lsc function:

$$\min_{x \in \Gamma} \; f(x) = \max_{q \in \mathbb{R}} \; q - \iota_{\mathcal{N}_\Lambda}(f - q),$$

Then we obtain for $Aq = (q\varphi_{1,0}, \ldots, q\varphi_{K,0})$, where $\varphi_{k,0} \equiv 1$ is the constant 1 function on Γ_k:

$$\min_{x \in \Gamma} \; f(x) = \max_{q \in \mathbb{R}} \; q - \iota_{\mathcal{N}_\Lambda}(f - q) \tag{2.60}$$

$$= \max_{q \in \mathbb{R}} \; q - \sum_{k=1}^{K} \iota_{\mathcal{N}_{\Lambda_k}}(f_k - A_k q). \tag{2.61}$$

Fenchel–Rockafellar duality then yields:

$$\begin{aligned} \min_{x \in \Gamma} \; f(x) &= \min_{\mathbf{u} \in \mathcal{M}_{\Lambda_1} \times \cdots \times \mathcal{M}_{\Lambda_K}} \; \iota_{\{1\}}(A^* \mathbf{u}) + \sum_{k=1}^{K} \langle \mathbf{u}_k, f_k \rangle + \iota_{\left(\mathcal{M}_{\Lambda_k}\right)_+}(\mathbf{u}_k) \\ &= \min_{\substack{\mathbf{u} \in \left(\mathcal{M}_{\Lambda_1}\right)_+ \times \cdots \times \left(\mathcal{M}_{\Lambda_K}\right)_+ \\ \sum_{k=1}^{K} \mathbf{u}_{k,0} = 1}} \; \sum_{k=1}^{K} \langle \mathbf{u}_k, f_k \rangle. \end{aligned}$$

This formulation can be substituted in the dual problem (dR-D) and we obtain

$$\sup_{\lambda \in (\Lambda_1 \times \cdots \times \Lambda_K)^{\mathcal{E}}} \sum_{v \in \mathcal{V}} \; \min_{\substack{\mathbf{u} \in \left(\mathcal{M}_{\Lambda_1}\right)_+ \times \cdots \times \left(\mathcal{M}_{\Lambda_K}\right)_+ \\ \sum_{k=1}^{K} \mathbf{u}_{k,0} = 1}} \; \sum_{k=1}^{K} \langle \mathbf{u}_k, f_{v,k} - (\operatorname{Div}_\Lambda \lambda)_{v,k} \rangle - \sum_{e \in \mathcal{E}} \iota_{\mathcal{K} \cap \Lambda}(\lambda_e). \tag{2.62}$$

Here the dual variables λ are chosen such that $(\operatorname{Div}_\Lambda \lambda)_v$ represents a piecewise polynomial with knots $a = \gamma_1 < \gamma_2 < \gamma_3 < \cdots < \gamma_{K+1} = b$ such that for each piece we have $(\operatorname{Div}_\Lambda \lambda)_{v,k} \in \Lambda_k$.

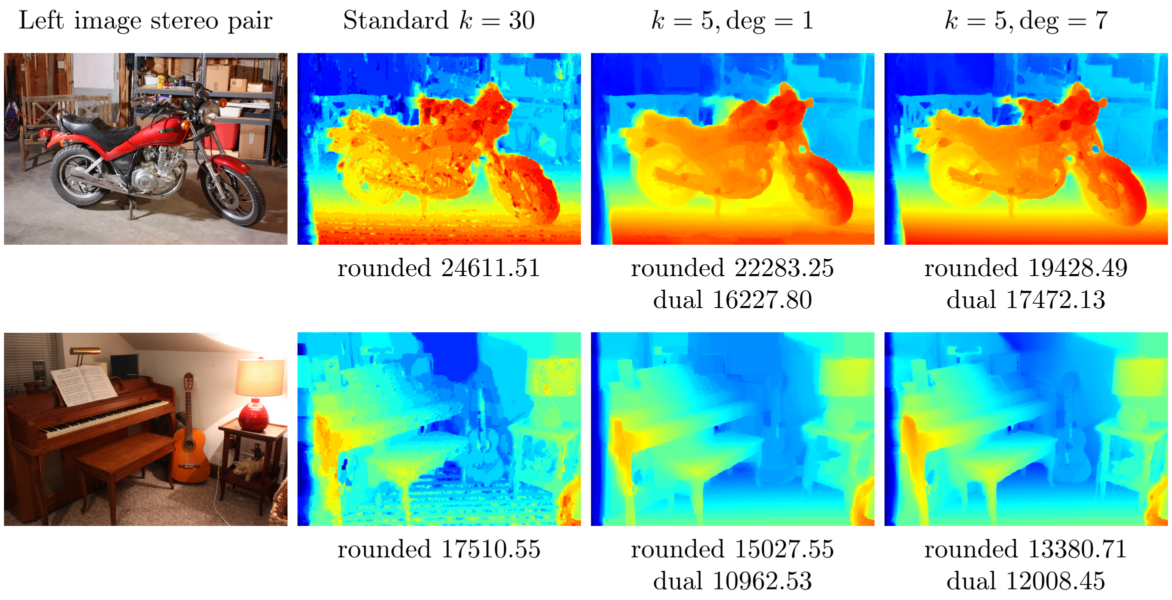
2.1.7 Numerical Experiments

2.1.7.1 Empirical Convergence Study

In this first experiment we evaluate the local marginal polytope relaxation of the MRF formulation Eq. (P) using a piecewise polynomial hierarchy of dual variables. We choose the graph $(\mathcal{V}, \mathcal{E})$ to be a square grid of size 16×16. I.e., the vertices $\mathcal{V}$ correspond to the points in the plane with its x- and y-coordinates being integers in the range $1, 2, \ldots, 16$, and two vertices are connected by an edge whenever the corresponding points are at distance 1. We fix a random polynomial data term $f_v : [-1, 1] \to \mathbb{R}$ of degree 4 at each vertex v by fitting a random sample of data points. To obtain a high-accuracy solution we solve the primal SDP formulation corresponding to the saddle-point formulation Eq. (2.62) with MOSEK[1]. For recovering a primal solution at each vertex v we compute the mode w.r.t. the 0^{th} moments to select the best interval denoted by $k^* = \arg\max_{1 \leq k \leq K} (y_v)_{k,0}$. Then we compute the mean of the discretized measure corresponding to the $(k^*)^{\text{th}}$ interval as $u_v = (y_v)_{k^*,1}$.

Figure 2.3 visualizes the primal and dual energies for varying degrees and/or number of pieces of the dual variable. While the dual energy strictly increases with higher degrees and/or number of pieces the primal energy is evaluated at the rounded solution and therefore does not strictly decrease in general. While for TV increasing the degree vs. increasing the number of pieces (for $K \cdot \deg$ constant) leads to similar performance, for Potts, in many situations, increasing the degree leads to larger dual energies, e.g., consider $\deg = 4, K = 1$ vs. $\deg = 1, K = 4$, red curve vs. blue curve in Fig. 2.3(d). In further experiments, we observed, that this holds in particular when the structure of the dual variables and the unaries match, i.e., $f_v, \lambda_e \in \Lambda$. Note that for Potts, since the dual variables are uniformly bounded on Γ and the derivative can be unbounded we drop the continuity constraint which leads to a more

[1] https://www.mosek.com/products/academic-licenses



Left image stereo pair | Standard $k = 30$ | $k = 5, \deg = 1$ | $k = 5, \deg = 7$

rounded 24611.51 | rounded 22283.25
dual 16227.80 | rounded 19428.49
dual 17472.13

rounded 17510.55 | rounded 15027.55
dual 10962.53 | rounded 13380.71
dual 12008.45

Figure 2.4: Stereo disparity estimation from a stereo image pair: Left: Standard MRF/OT discretization implemented using a continuous piecewise linear under-approximation for the unaries and piecewise linear duals. middle: piecewise linear duals. right: piecewise polynomial duals. The visual appearance of the solution to the standard MRF/OT discretization shows a strong grid bias. The dual energy gap increases for increasing the degree and/or the number of pieces. Likewise the energy at the rounded solution decreases. Figure taken from [Pub2].

compact formulation and larger dual energies.

2.1.7.2 Stereo Matching

In this experiment we consider stereo matching using the anisotropic relaxation Eq. (R-P). We consider the Motorcycle and the Piano image pairs from the Middlebury stereo benchmark [120]. We downsample the images by factor 4. The disparity cost term is first calculated using 135 discrete disparities obtained by shifting the images by the corresponding amount of pixels and comparing the image gradients. More specifically, given a RGB image I mapping from $[1, \dots, n_y] \times [1, \dots, n_x]$ to a RGB value in $\mathbb{R}^3$, the x and y derivatives are calculated as $I_x(i,j) = I(i, \min\{j+1, n_x\}) - I(i,j)$ and $I_y(i,j) = I(\min\{i+1, n_y\}, j) - I(i,j)$. For a stereo image pair $(I^{\text{left}}, I^{\text{right}})$ and a disparity $l \in \mathbb{N}$ the

Dual energies

deg	$K = 1$	$K = 3$	$K = 5$
1	14180.08	15733.71	16227.80
2	15052.18	16430.97	16773.75
3	15601.89	16778.87	17055.25
4	15938.92	16998.63	17235.21
5	16191.40	17147.92	17346.49
6	16369.95	17243.47	17422.26
7	16480.22	17308.91	17472.13

Energies rounded

deg	$K = 1$	$K = 3$	$K = 5$
1	28982.45	25005.05	22283.26
2	31038.14	22525.12	21049.32
3	28505.41	21500.65	20428.34
4	27255.48	20841.62	20049.04
5	25795.56	20344.17	19764.23
6	24081.51	20032.77	19559.00
7	23142.67	19869.71	19428.50

Table 2.1: Energies for stereo matching Motorcycle. Left: Dual energies. Right: Primal energies at the rounded solution.

cost at pixel (i, j) is then calculated as

$$D(i,j,l) = \min\{\|I_x^{\text{left}}(i, \min\{j+l, n_x\}) - I_x^{\text{right}}(i,j)\|_1, 0.1\} + \tag{2.63}$$

$$\min\{\|I_y^{\text{left}}(i, \min\{j+l, n_x\}) - I_y^{\text{right}}(i,j)\|_1, 0.1\}. \tag{2.64}$$

Then, the cost dataterm is approximated from below in terms of a continuous piecewise cubic polynomial $f_v : [1, 135] \to \mathbb{R}$ using 30 pieces at each $v \in \mathcal{V}$. This dataterm is precomputed once during a preprocessing and subsequently used as a benchmark for the different methods that we compare. For the coupling, we use a total variation-like regularization, i.e., $f_{vw}(u_v, u_w) = \alpha|u_v - u_w|$ with weight $\alpha = 0.2$. In Fig. 2.4 we compare the standard MRF/OT discretization as described in Example 2.1.10 with our framework using piecewise linear and piecewise polynomial dual variables with degree 7 both with 5 pieces. The standard MRF/OT discretization is equivalent to a piecewise linear approximation of the data term with piecewise linear duals in our framework. The piecewise linear approximation is obtained by sampling the piecewise cubic polynomial f_v at the interval boundaries of the pieces. The reported energy for the solution of the standard approach in Fig. 2.4 is also evaluated using the piecewise linear cost. As the resulting optimization problem is large-scale we solve the saddle-point formulation Eq. (2.62) with PDHG [27] as described in Section 2.1.6.4 using the GPU-based PDHG framework prost[2]. In contrast to the previous experiment which uses a combined mode and mean rounding procedure we found the plain mean of the discretized measure to produce better results on real data: More explicitly we recover a solution according to $u_v = \sum_{k=1}^{K} t_k(y_v)_{k,0}$ at each vertex $v \in \mathcal{V}$. In Table 2.1 we compare both, dual and nonconvex primal energies, for a larger hierarchy of dual subspaces. As predicted by the theory the dual energies are strictly increasing for higher degrees and a higher number of pieces in the piecewise polynomial dual functions. Furthermore, except for one case, also the rounded primal energies decrease for higher degrees and numbers of pieces. Overall a significant drop in the primal-dual gap can be observed vindicating the proposed hierarchy of dual subspaces.

2.1.7.3 Primal Solution Ambiguity

As mentioned before the evaluation of the nonconvex primal energy of primal solutions bears the inherent difficulty that the found moments on the vertices may not correspond to Dirac measures. Here we demonstrate this effect on a toy example. Therefor we consider a MRF on two vertices with quadratic unary data terms f, see Fig. 2.5. Picking a TV regularization scale of 0.1 and recovering the primal solutions by the first moment of the found solutions yields the solution depicted in Fig. 2.5. This solution, however, is far from the optimum as the recovered moment vector does not correspond to a Dirac measure on the left vertex $\{1\}$. It is instructive to explain this result in terms of the corresponding dual variable λ in Fig. 2.6. The dual function tries to penalize the distance between the solutions on the vertices by an approximating TV regularization by a 0.1-Lipschitz continuous function. This approximation is exact if we may use affine linear functions with slope 0.1 as dual variables. In view of (R-D), however, an affine dual variable would cause a new minimum of $f_{\{1\}} + (\operatorname{Div}\lambda)_{\{1\}}$ at $\mu_{\{1\}} = \delta_1$. Therefore the supremizing λ, as depicted in Fig. 2.7, of (R-D) results in $f_{\{1\}} + (\operatorname{Div}\lambda)_{\{1\}}$ having two global minima, as depicted in Fig. 2.7. Hence $f_{\{1\}} + (\operatorname{Div}\lambda)_{\{1\}}$ has multiple global minimizers corresponding to convex combinations of Diracs at $x_{\{1\}} = -1$ and $x_{\{1\}} = 1$ with the depicted solution being one of them.

2.1.8 Summary

This section has studied functional lifting under the viewpoint of discretizations of a more general infinite-dimensional formulation involving probability measure related to the original problems. While focusing on a specific MRF structure, we provided theoretical insights and intuitions which convey to the more general concept of functional lifting and explain why it can yield more faithful solutions to the original problem. For the remainder of this work, we will come back to the theoretical concepts developed in this section and motivate the use of lifting techniques in a broader context.

[2] https://github.com/tum-vision/prost

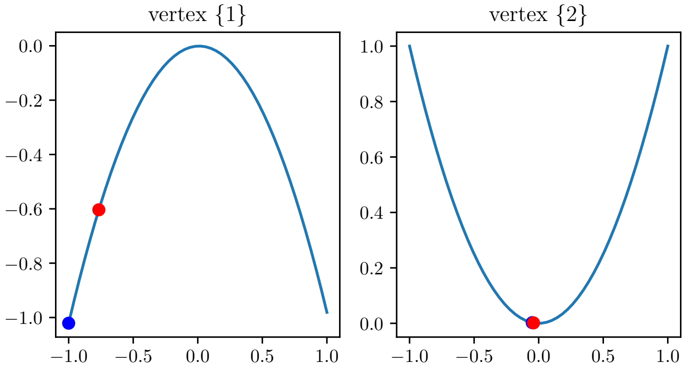


Figure 2.5: Depicted here are the unaries on two vertices of the considered toy problem. The blue dots are the true solution whereas the red dots correspond to the recovered solution of the relaxed problem.

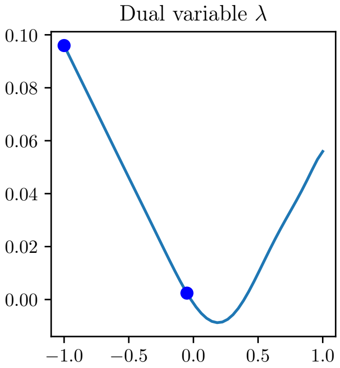


Figure 2.6: The dual variable corresponding to the unaries of Fig. 2.5 approximates the TV distance between the solutions on the vertices (blue dots) quite well as it almost resembles an affine linear function on the interval between those solutions. Here a dual variable of degree 10 was used.

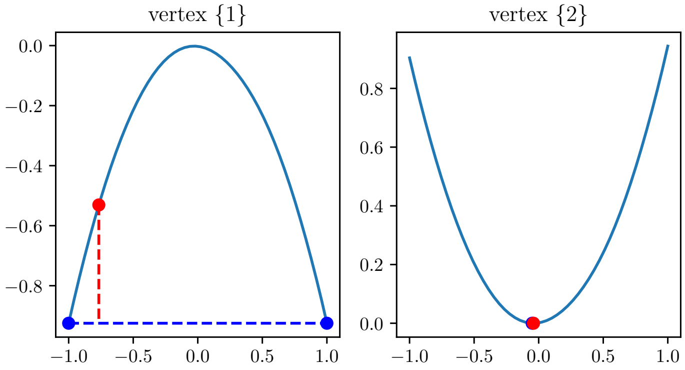


Figure 2.7: The resulting inner optimization problem of Eq. (R-D) reveals why the found solution (red dot) is ambiguous in terms of convex combinations of global solutions: The inner problem $f_{\{1\}} + (\text{Div}\,\lambda)_{\{1\}}$ at vertex $\{1\}$ has two global minima. Any convex combination of Diracs at $x_{\{1\}} = -1$ and $x_{\{1\}} = 1$ yields the same dual energy. However only the solution $x_{\{1\}} = -1$ corresponds to the true solution of the nonconvex problem. The existence of the second global minimum of the inner problem at vertex $\{1\}$ is best understood in the context of the dual variable in Fig. 2.6: The dual variable tries to approximate an affine-linear function between the two blue dots in Fig. 2.6 but at the same time prevents a minimum of the inner problem at vertex $\{1\}$ having a lower energy than the true global optimum at $x_{\{1\}} = -1$. Therefore the dual variable is "bent" upwards to the right.

2.2 Fast Convex Relaxations using Graph Discretizations

As discussed in Section 2.1 nonconvex energy minimization problems such as stereo matching can be solved near-globally optimal by convex lifting approaches. Yet, applying these techniques comes with a significant computational effort, reducing their feasibility in practical applications. In this section we discuss the spatial discretization of continuous partitioning problems into a graph structure, generalizing discretization onto a Cartesian grid. This setup allows us to faithfully work on super-pixel graphs constructed by SLIC or Cut-Pursuit, massively decreasing the computational effort for lifted partitioning problems compared to a Cartesian grid, while optimal energy values remain similar. Whereas we have previously expended effort in increasing the dimensionality of the label space Γ we will now pursue making the overall problem more efficient by discretizing the spatial domain Ω. We discuss this methodology in detail and show examples in stereo estimation, where we demonstrate that the proposed graph discretization can reduce runtime as well as memory consumption of convex relaxations. This section is based on [Pub5].

2.2.1 Graph Discretizations for Convex Relaxations

2.2.1.1 Problem Description

We consider Eq. (2.1) for a spatial continuous domain $\Omega \subset \mathbb{R}^d$ (e.g. $d = 2$ for image data) and a one-dimensional label space, i.e. $\Gamma \subset \mathbb{R}$, and a total variation regularization

$$E(u) = \int_\Omega f(x, u(x))\,\mathrm{d}x + \mathrm{TV}(u). \tag{2.65}$$

Formally, we define the total variation of a function $u \in L^1(\Omega, \Gamma)$ in this case as

$$\mathrm{TV}(u) = \sup\left\{ -\int_\Omega u(x)\operatorname{div} p(x)\,\mathrm{d}x\colon \quad p \in C_c^1(\Omega, \mathbb{R}^d), \quad \|p(x)\|^2 \leq 1 \right\}.$$

Note that the above definition reduces to $\mathrm{TV}(u) = \int_\Omega \|\nabla u(x)\|\,\mathrm{d}x$ for smooth u. We define u to be an element of the space of bounded variation $\mathrm{BV}(\Omega, \mathbb{R})$ if $\mathrm{TV}(u)$ is finite. We can then identify this value with the mass of the distributional derivative Du, i.e. $\int_\Omega |Du| = \mathrm{TV}(u)$. For $\Omega \subset \mathbb{R}^d$ the bounded Radon measure Du can be decomposed [5, Thm. 10.4.1] into

$$Du = \nabla u\,\mathcal{L}^d + Cu + (u^+ - u^-) \otimes \nu_u \mathcal{H}^{d-1} \llcorner J_u, \tag{2.66}$$

where $\mathcal{L}^d$ is the Lebesgue measure, $\mathcal{H}^{d-1}$ denotes the $d-1$-dimensional Hausdorff measure, J_u is the jump set of u, where $u^+ \neq u^-$, i.e. the values at the boundary differ, ν_u the normal of the boundary and Cu a remainder Cantor part. In the following we will consider functions $u \in \mathrm{SBV}(\Omega, \mathbb{R})$, which is the space of functions for which $Cu = 0$ [3, 91]. We further define the perimeter of a measurable set $P \subset \Omega$, $\mathrm{Per}(\Omega, P)$, in turn by the total variation of its characteristic function $\chi_P : \Omega \to \mathbb{R}$ [26], the boundary of a set as $\partial S = \bar{S} \setminus \mathrm{int}(S)$, and the length of the boundary $B_{k,l} = \partial P_k \cap \partial P_l$ between two sets P_k, P_l via

$$|B_{k,l}| = \mathcal{H}^{d-1}(B_{k,l}). \tag{2.67}$$

These definitions allow us to examine the continuous boundary of shapes, seeFig. 2.8.

2.2.1.2 Graph Discretization

We are interested in solving Eq. (2.65) numerically. To do so we need to translate the problem into the discrete setting. To take a step from the continuous definitions to a discrete problem, we make use of the fact that we expect solutions $\hat{u}$ of total variation regularized problems to be piecewise constant with a finite number of pieces. Although this might not hold in general for the spatial continuous case, we end up with a finite number of grid points after discretizations anyways. Therefore we will assume $\hat{u}$ from now on to have finitely many constant pieces. A good discretization to a finite setting mimics this piecewise constant structure exactly. We hence represent the discretization by a graph of candidate constant sets, the nodes of which represent each separate constant piece and where neighboring pieces are connected by edges in the graph. After solving the problem on this discrete graph, the final solution

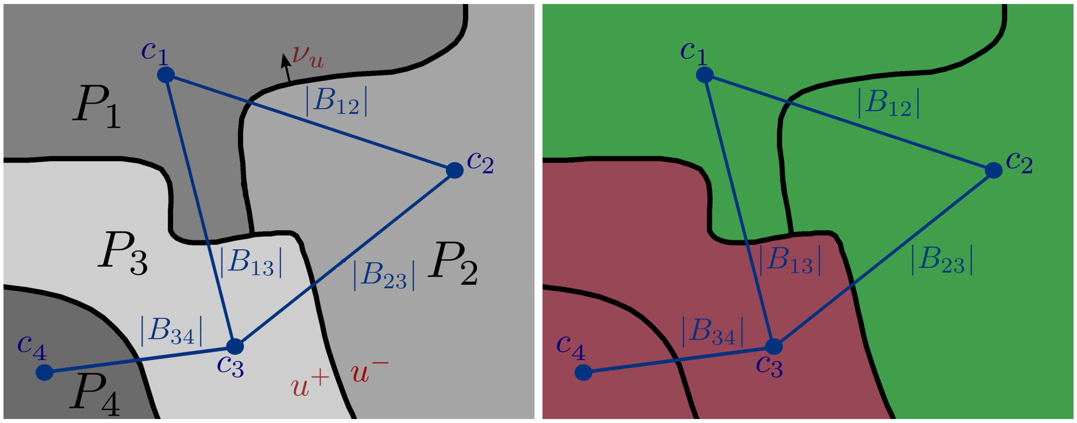


Figure 2.8: Sketch of the discretization process for a graph-based discretization. *Left:* The underlying continuous function $u \in BV_\Pi(\Omega, \mathbb{R})$ is pictured in black, with piecewise-constant partitions $\Pi = \{P_1, P_2, P_3, P_4\}$ shown in gray as well as the discrete graph structure in blue. *Right:* A minimal partitions problem with two potentials is solved on this graph structure. Pictured is the solution $\hat{u}$ which now corresponds to a piecewise constant solution (in red and green). Figure adapted from [Pub5].

$\hat{u}$ can be reassembled by assigning to each constant piece its matched value according to the respective value of the node that represents it. This setup is sketched in Fig. 2.8.

Note that this is a generalization of a classical discretization to a Cartesian grid. Placing a continuous function on a pixel grid corresponds to claiming that the function is piecewise-constant on every image pixel. Hence the boundaries of the solution $\hat{u}$ to the minimal partitions problem will be a subset of the boundaries imposed by the image pixels.

To connect the continuous formulation of Eq. (2.65) to a discrete setting we define the discretization as a finite set $\Pi = \{P_v\}$ of candidate sets $P_v \subset \Omega$ with $M = |\Pi|$ partitions, such that $\bigcup_v \overline{P_v} = \Omega$, $P_v \cap P_w = \emptyset$ for all $v \neq w$. The continuous function $u \in \mathrm{SBV}(\Omega, \mathbb{R})$ is assumed to be constant on every partition, so that we can denote its value on partition $P_v \in \Pi$ by a vector $c_v \in \mathbb{R}$. Thus, $u(x) = c_v$ for every $x \in P_v \subset \Omega$. The partition Π can be represented by a set of nodes $\mathcal{V} = \{1, \dots, M\}$ where each node corresponds to a segment $P_v \in \Pi$. Furthermore we can describe every boundary between sets P_v and P_w as B_{vw} and by that define an edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ as $\mathcal{E} = \{(v, w) \in \mathcal{V} \times \mathcal{V} \mid |B_{vw}| > 0, v \neq w\}$. Note, that the perimeter of some partition $P_v \in \Pi$ is given by $\mathrm{Per}_{P_v} = \sum_{(v,w)\in\mathcal{E}} |B_{vw}|$.

Let us assume that our desired solution $\hat{u}$, which minimizes Eq. (2.65), is piecewise constant. More formally, given some partition Π let us write $u \in \mathrm{SBV}_\Pi(\Omega, \mathbb{R})$ to denote continuous functions in BV which are piecewise constant on the regions in Π, and assume $\hat{u} \in \mathrm{SBV}_\Pi(\Omega, \mathbb{R})$. This implies that the jump set J_u is a subset of $\cup_{(v,w)\in E} B_{vw}$ and that $\Omega \setminus J_u$ is a subset $\cup_{v \in V} P_v$, or, in other words, the discrete partitioning by Π is able to represent the continuous structure of $\hat{u}$.

Under the above assumption we can restrict the minimization of E over all functions $u \in \mathrm{SBV}(\Omega, \mathbb{R})$ to those in $\mathrm{SBV}_\Pi(\Omega, \mathbb{R})$ which allows to simplify Eq. (2.65) to a problem in which merely the values c_v inside the piecewise constant regions are the unknowns. Let us discuss the two main components of Eq. (2.65) separately.

Data Term:

Considering E for any piece-wise constant $u \in \mathrm{SBV}_\Pi(\Omega, \mathbb{R})$ allows us to rewrite the first term of Eq. (2.65) as

$$K(u) = \int_\Omega f(x, u(x))\, \mathrm{d}x = \sum_{v=1}^{M} \int_{P_v} f(x, c_v)\, \mathrm{d}x =: K_\Pi(c) \tag{2.68}$$

which is the discrete representation $K_\Pi(c) : \mathbb{R}^M \to \mathbb{R}$ of this term that mere depends on the values c_v.

Regularization Term:

For the jump regularization, we can write $\mathrm{TV}(u)$ for any piecewise constant $u \in \mathrm{SBV}_\Pi(\Omega, \mathbb{R})$ as (cf.

[46, 17])

$$\begin{aligned}\mathrm{TV}(u) &= \int_{\Omega\setminus J_u} \|\nabla u\|\,\mathrm{d}x + \int_{J_u} |u^+ - u^-|\,\mathrm{d}\mathcal{H}^{d-1} = \sum_{(v,w)\in\mathcal{E}} \int_{B_{vw}} |c_v - c_w|\,\mathrm{d}\mathcal{H}^{d-1} \\ &= \sum_{(v,w)\in\mathcal{E}} |c_v - c_w| \int_{B_{vw}} \mathrm{d}\mathcal{H}^{d-1} = \sum_{(v,w)\in\mathcal{E}} |B_{vw}|\,|c_v - c_w| =: R_\Pi(c)\end{aligned} \tag{2.69}$$

using the fact that $\int_{B_{vw}} d\mathcal{H}^{d-1} = |B_{vw}|$. With this weighting we can define the weighted finite graph $G = (\mathcal{V}, \mathcal{E}, \{|B_{vw}|\}_{(v,w)\in\mathcal{E}})$ as the discrete graph structure with which any continuous piece-wise constant function $u \in \mathrm{SBV}_\Pi(\Omega, \mathbb{R})$ can be represented. We can now define the discretized energy as

$$E_\Pi := K_\Pi + R_\Pi. \tag{2.70}$$

Interestingly, the above restriction from the minimization of E over $\mathrm{SBV}(\Omega, \mathbb{R})$ to its minimization over $\mathrm{SBV}_\Pi(\Omega, \mathbb{R})$ (which translates into the minimization of E_Π over $c \in C_\Pi$) remains valid as long as the jump set of the true solution is a subset of the jumps in the partition Π, independent of what exactly the "super" jump-set of Π is. Let us formalize this result:

Proposition 2.2.1. *Assume a discretization Π and its assorted partitions P_v to be given. Let $\hat{u}$ be a minimizer to the continuous problem Eq. (2.65) for a given data dependent f. If the jump-set $J_{\hat{u}}$ of $\hat{u}$ is a subset of the jump set of Π given as the union of boundaries $\cup_{(v,w)\in\mathcal{E}} B_{vw}$, then*

$$\min_{u\in\mathrm{SBV}(\Omega,\mathbb{R})} E(u) = \min_{c\in\mathbb{R}^M} E_\Pi(c),$$

for the discrete energy $E_\Pi = K_\Pi + R_\Pi$, i.e. the continuous minimum $E(\hat{u})$ is equal to the minimum $E_\Pi(\hat{c})$ of the discrete energy of E_Π.

Proof. We consider the space $\mathrm{SBV}_\Pi(\Omega, \mathbb{R})$ of functions in $\mathrm{SBV}(\Omega, \mathbb{R})$ that are piecewise-constant on partition Π. From the assumption that $J_{\hat{u}}$ is a subset of the jump set of Π, given by $\bigcup_{(v,w)\in\mathcal{E}} B_{vw}$, we deduce $\hat{u} \in \mathrm{SBV}_\Pi(\Omega, \mathbb{R}^L)$. For partitions Π' and Π the relation $\Pi' \le \Pi$, where "$\le$" refers to the partial order on partitions meaning Π' is a finer partition than Π, implies $\mathrm{SBV}_\Pi(\Omega, \mathbb{R}) \subseteq \mathrm{SBV}_{\Pi'}(\Omega, \mathbb{R})$. The assumption that the jump set of $\hat{u}$ is a subset of the jump set of Π hence translates to $\hat{u} \in \mathrm{SBV}_\Pi(\Omega, \mathbb{R})$. For $u \in \mathrm{SBV}_\Pi(\Omega, \mathbb{R})$ and the according c we already have shown $E_\Pi(c) = E(u)$.

Hence, $\hat{u} \in \mathrm{SBV}_\Pi(\Omega, \mathbb{R})$ allows us to write

$$\begin{aligned}\min_{c\in\mathbb{R}^M} E_\Pi(c) &= \min_{\tilde{u}\in\mathrm{SBV}_\Pi(\Omega,\mathbb{R})} E(\tilde{u}) \\ &\le E(\hat{u}) = \min_{u\in\mathrm{SBV}(\Omega,\mathbb{R})} E(u).\end{aligned}$$

Equality now follows due to $\mathrm{SBV}_\Pi(\Omega, \mathbb{R}) \subseteq \mathrm{SBV}(\Omega, \mathbb{R})$ from

$$\min_{u\in\mathrm{SBV}(\Omega,\mathbb{R})} E(u) \le \min_{\tilde{u}\in\mathrm{SBV}_\Pi(\Omega,\mathbb{R})} E(\tilde{u}) = \min_{c\in\mathbb{R}^M} E_\Pi(c).$$

□

Proposition 2.2.1 shows that if the jump set of $\hat{u}$ is contained in Π, then the exact optimum $\hat{u}$ of the continuous problem can actually by found by computing a discrete solution $\hat{c}$ of the function F_Π numerically on a finite graph. Practically however, we now need to find some partition Π that approximates (or ideally overestimates) the true jump set $J_{\hat{u}}$, but consists of a limited number of segments. On a Cartesian grid, the equivalent operation is to subsample the image, result in the "superpixel" seen in Fig. 2.9 on the left, which are not well aligned with edges in the images. However approaches such as SLIC (middle) or Cut-Pursuit (right, in the variant of [131]) are more adept at finding a superset of candidate partitions.

2.2.2 Functional Lifting on Graphs

After formulating the problem of minimizing the reduced energy in Eq. (2.70) we can now apply ideas from functional lifting [89] in order to derive a faithful convex relaxation of Eq. (2.65) combined with the computational benefits achieved by the super-pixel graph discretization.

We adapt the lifting strategy from [89], see Example 2.1.11, to the discretized grid. For the sake of clarity we reintroduce the relevant notation in the following. We assume Γ is subdivided by $K+1$ labels $\gamma_1 < \ldots < \gamma_{K+1}$ such that $\Gamma = [\gamma_1, \gamma_{K+1}]$. For e_k the k^{th} unit vector let $\mathbf{1}_k := \sum_{i=1}^{k} e_i$. Furthermore define $\gamma_k^\alpha := \alpha\gamma_{k+1} + (1-\alpha)\gamma_k$ and $\mathbf{1}_k^\alpha := \alpha\mathbf{1}_k + (1-\alpha)\mathbf{1}_{k-1}$. The lifting strategy now uses the lifting map $\varphi(\gamma_k^\alpha) = \mathbf{1}_k^\alpha$ and Λ according to φ. Analogous to Eq. (2.70), the lifted energy for $\mathbf{u} \in \text{SBV}(\Omega, \mathcal{M}_\Lambda)$, where $\mathcal{M}_\Lambda \cong \mathbb{R}^K$, in [89] is stated as

$$\hat{E} = \hat{K} + \hat{R} \tag{2.71}$$

$$\hat{K}(\mathbf{u}) = \int_\Omega f_\Lambda^{**}(x, \mathbf{u}(x)) \,\mathrm{d}x \tag{2.72}$$

$$\hat{R}(\mathbf{u}) = \int_{\Omega \setminus J_\mathbf{u}} \psi^{**}(\nabla\mathbf{u}(x)) \,\mathrm{d}x + \int_{J_\mathbf{u}} \psi^{**}\left((\mathbf{u}^+(x) - \mathbf{u}^-(x)) \otimes \nu_\mathbf{u}(x)\right) \,\mathrm{d}\mathcal{H}^{d-1}(x), \tag{2.73}$$

where the jump regularizer ψ is defined in a way such that for $u \in \text{SBV}(\Omega, \mathbb{R})$ the total variation of u is equal to $\hat{R}(\varphi \circ u)$ and $\psi^{**} \colon \mathbb{R}^{\mathbf{M} \times d} \to \mathbb{R}$ is given as

$$\psi^{**}(g) = \sum_{i=1}^{K} \|g_i\| \cdot (\gamma_{i+1} - \gamma_i), \tag{2.74}$$

where $g_i = \mathbf{m}_i^T g$ for $\mathbf{M} := \{\mathbf{m}_1, \ldots, \mathbf{m}_K\}$ the dual basis of $\{\varphi_1, \ldots, \varphi_K\}$. Note that the notation of the lifted energy is different from the notation used in Section 2.1 as we treat the lifted regularizer slightly differently.

The discretized energy now operates on the lifted space $\hat{E}_\Pi \colon (\mathcal{M}_\Lambda)^M \to \mathbb{R}$. Define $f_{P_v} \colon \Gamma \to \mathbb{R}$ as the integral (of functions) $f_{P_v} = \int_{P_v} f(x, \cdot\,) \,\mathrm{d}x$ in the sense of $f_{P_v}(\mathbf{c}_v) = \int_{P_v} f(x, \mathbf{c}_v) \,\mathrm{d}x$ for $\mathbf{c} \in (\mathcal{M}_\Lambda)^M$. The data term is constructed analogously to Eq. (2.68) as

$$\hat{K}_\Pi(\mathbf{c}) = \sum_{v=1}^{M} (f_{P_v})_\Lambda^{**}(\mathbf{c}_v). \tag{2.75}$$

Note that it is sensible to apply the lifting to the integrated function f_{P_v} instead of applying it to $f(x, \cdot\,)$ directly as is holds that

$$(f_{P_v})_\Lambda^{**} \geq \int_{P_v} f(x, \cdot\,)_\Lambda^{**} \,\mathrm{d}x \tag{2.76}$$

and $(f_{P_v})_\Lambda^{**}$ hence might be a more faithful convex underapproximation of f_{P_v}.

In order to preserve the regularizer in Eq. (2.69) for $\boldsymbol{\varphi}(c_v)$ and $\boldsymbol{\varphi}(c_w)$ we introduce a lifted jump regularizer $\psi_\Pi \colon \mathcal{M}_\Lambda \to \mathbb{R}$ such that $|c_v - c_w| = \psi_\Pi(\boldsymbol{\varphi}(c_v) - \boldsymbol{\varphi}(c_w))$ in analogy to [89] as

$$\psi_\Pi(g) = \begin{cases} |c_v - c_w| \cdot |\xi|, & \text{if } g = (\boldsymbol{\varphi}(c_v) - \boldsymbol{\varphi}(c_w)) \cdot \xi \text{ for } \xi \in \mathbb{R}, \\ \infty, & \text{else.} \end{cases} \tag{2.77}$$

Following the lines of [89] the convex biconjugate of ψ is then given as

$$\psi_\pi^{**}(g) = \sum_{i=1}^{L-1} |g_i| \cdot (\gamma_{i+1} - \gamma_i). \tag{2.78}$$

We can now formulate the lifted energy $\hat{E}_\Pi := \hat{K}_\Pi + \hat{R}_\Pi$ using the lifted regularization term defined as

$$\hat{R}_\Pi(\mathbf{c}) := \sum_{(v,w) \in \mathcal{E}} |B_{vw}| \; \psi_\Pi^{**}(\mathbf{c}_v - \mathbf{c}_w). \tag{2.79}$$

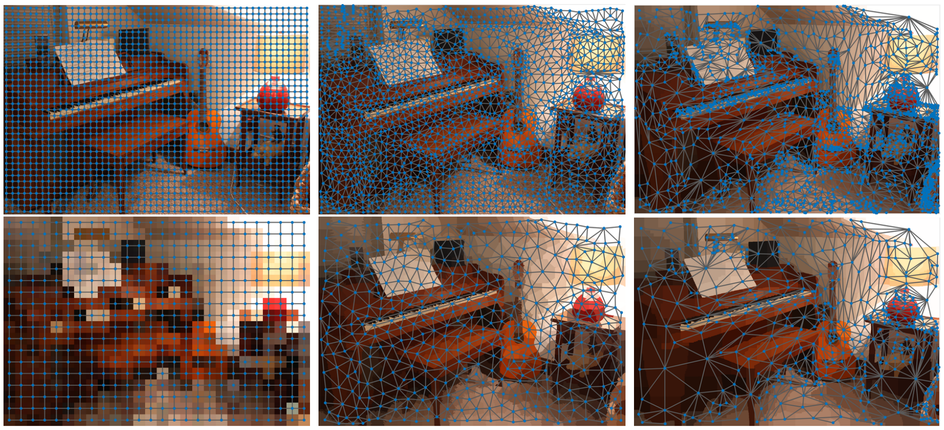


Figure 2.9: From left to right: Grid Sampling, SLIC Superpixels and L^0 Cut-Pursuit. Images from the Middlebury dataset [120]. The top row shows a fine discretization into the same number of nodes for every method, whereas the lower row shows a coarse discretization with the same number of nodes for every method. Figure taken from [Pub5].

The following proposition shows how the lifted energy $\hat{E}$ and the discretized lifted energy $\hat{E}_\Pi$ relate to each other for piecewise constant solutions w.r.t. Π:

Proposition 2.2.2. *Assume* $\mathbf{u} \in \mathrm{SBV}_\Pi(\Omega, \mathcal{M}_\Lambda)$ *piecewise constant w.r.t.* Π *and let* $\mathbf{c} \in (\mathcal{M}_\Lambda)^M$ *denote the corresponding discrete vector representing the values of* $\mathbf{u}$ *on the elements of* Π. *Then it holds* $\hat{E}_\Pi(\mathbf{c}) \geq \hat{E}(\mathbf{u})$.

Proof. Using Eq. (2.76) we immediately can show the desired unequality for the data term

$$\hat{K}_\Pi(\mathbf{c}) = \sum_{v=1}^{M} (f_{P_v})_\Lambda^{**}(\mathbf{c}_v) \geq \sum_{v=1}^{M} \int_{P_v} f(x, \mathbf{u}(x))_\Lambda^{**} \, \mathrm{d}x = \hat{K}(\mathbf{u}). \tag{2.80}$$

It remains to show the equality of the regularizers $\hat{R}$ and $\hat{R}_\Pi$ by calculating

$$\hat{R}(\mathbf{u}) = \int_{J_\mathbf{u}} \psi^{**}\left((\mathbf{u}^+(x) - \mathbf{u}^-(x)) \otimes \nu_\mathbf{u}(x)\right) \, \mathrm{d}\mathcal{H}^{d-1}(x) \tag{2.81}$$

$$= \sum_{(v,w)\in\mathcal{E}} \int_{B_{vw}} \psi^{**}\left((\mathbf{c}_v - \mathbf{c}_w) \otimes \nu_\mathbf{u}(x)\right) \mathrm{d}\mathcal{H}^{d-1}(x) \tag{2.82}$$

$$= \sum_{(v,w)\in\mathcal{E}} \int_{B_{vw}} \sum_{i=1}^{K} |(\mathbf{c}_v - \mathbf{c}_w)_i| \cdot \|\nu_\mathbf{u}(x)\| \cdot (\gamma_{i+1} - \gamma_i) \, \mathrm{d}\mathcal{H}^{d-1}(x) \tag{2.83}$$

$$= \sum_{(v,w)\in\mathcal{E}} \int_{B_{vw}} \sum_{i=1}^{K} |(\mathbf{c}_v - \mathbf{c}_w)_i| \cdot (\gamma_{i+1} - \gamma_i) \, \mathrm{d}\mathcal{H}^{d-1}(x) \tag{2.84}$$

$$= \sum_{(v,w)\in\mathcal{E}} |B_{vw}| \; \psi_\Pi^{**}(\mathbf{c}_v - \mathbf{c}_w) \tag{2.85}$$

$$= \hat{R}_\Pi(\mathbf{c}). \tag{2.86}$$

□

2.2.3 Partitioning

The choice of efficient discretization of the input image data is directly related to superpixel approaches, e.g. [1, 135]. Their general idea is to generically reduce the computational complexity of any (pixel-based) numerical algorithm, by locally grouping pixels of similar color to larger superpixels. The most prominent algorithm in current practice is SLIC (Simple Linear Iterative Clustering) [1]. Ideally, the superpixel setup should also be chosen by an appropriate minimization procedure that adheres object edges. However, edge adherence is often costly. An interesting exception is the Cut-Pursuit algorithm [66, 67], which solves total variation minimization and related problems in a fast sequence of binary graph cuts, making it competitive as a discretization step and leading to boundaries that better adhere with minimal partitions. This approach has been successfully applied in practice in such works as [68, 49].

For implementation reference we replicate some parts of the L^0 Cut-Pursuit [66] variant of [131] in the continuous setting.

To obtain a good trade-off between having as few segments as possible but still constructing a partition whose jump set is a super set of the jump set of a minimizer $\hat{u}$, we exploit a modification of the Cut-Pursuit (CP) algorithm of [66] discussed in [131]. In [66] Landrieu and Obozinski develop an approach to solve total variation problems [114, 21] with an alternating method solving graph cuts and reduced problems on the smaller graphs generated by these cuts. This is an efficient method with superior performance compared to more classical optimization methods as primal-dual [27] or Douglas-Rachford [39] algorithms minimizing the total variation problem. The Cut-Pursuit algorithm can be further extended to a variant minimizing the L^0 norm of the graph gradient. This strategy is able to quickly return approximate solutions to partitioning problems for a relatively high number of partitions compared. The final number of partitions depends on regularization parameters and is data-dependent. In [131] Tenbrinck et. al. modify the Cut-Pursuit for L^0 by simplifying the algorithm to alternating between a graph cut and solving the data term separately on each generated partitions. We will denote this method as L^0-Cut-Pursuit (L^0-CP). We discuss this algorithm in a continuous setting with an application specific data fidelity term g, which we will discuss later, resulting in the following alternating algorithm: For a given function $u^k \in \mathrm{SBV}(\Omega, \mathbb{R})$ and some given data $g \in L^1(\Omega, \mathbb{R})$, one step of the algorithm consists of the two alternating optimization steps. The first one is

$$B^{k+1} = \operatorname*{argmin}_{B \subset \Omega} \int_\Omega (u^k(x) - g(x)) 1_B(x)\; dx + \alpha_c \int_\Omega |D 1_B|, \tag{2.87}$$

where 1_B denotes a characteristic function on B. This set minimization in Eq. (2.87) is binary and thus globally solvable. Then compute Π^{k+1} as the connected components of B^{k+1} and, in a second step, find the mean over every partition,

$$c_v^{k+1} = \frac{1}{|P_v|} \int_{P_v} g(x)\; dx. \tag{2.88}$$

From the values c_i^{k+1} of the partitions $P_i \in \Pi^{k+1}$, we can compute the continuous solution via

$$u^{k+1} = \sum_{P_v \in \Pi^{k+1}} c_v^{k+1}\; 1_{P_v}. \tag{2.89}$$

A visualization of the L^0-CP partitioning is given in Fig. 2.9. In comparison to a naive subsampling on a grid (left) and the SLIC superpixels from [1], the L^0-CP generates less uniformly-sized regions, allowing to combine large constant regions into a single node in a graph and thus being well suited for an efficient coarsification with accurate edges. Figure 2.10 shows the fidelity of the L0-Cut Pursuit representation

Algorithm 1: L0-Pursuit from [131]

```
Result: Discretization $\Pi$
$c^0 \leftarrow \mathrm{mean}(g)$;
$\Pi^0 \leftarrow \{\Omega\}$;
while $\Pi^k \neq \Pi^{k+1}$ do
    $B^{k+1} \leftarrow$ Solve Eq. (2.87) for given $u^k$;
    $\Pi^{k+1} \leftarrow$ connected components of $B^{k+1}$;
    $c^{k+1} \leftarrow$ Solve Eq. (2.88) for given $\Pi^{k+1}$;
    $u^{k+1} \leftarrow$ Compute as in Eq. (2.89);
    $k \leftarrow k + 1$;
end
$\Pi \leftarrow \Pi^{k+1}$
```

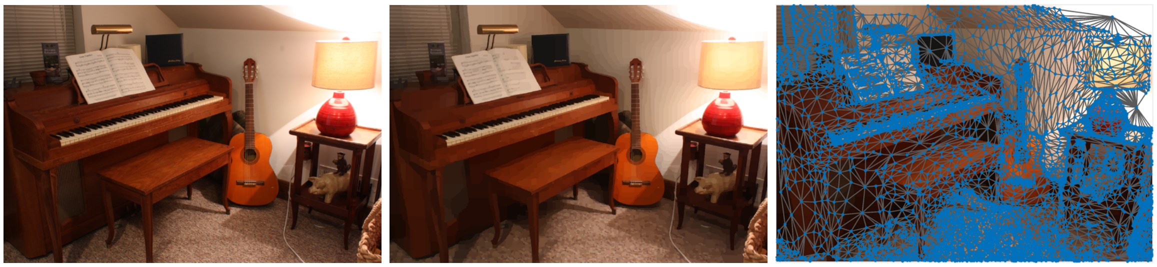


Figure 2.10: Reduction of the raw image (left) to 6031 nodes in the graph structure (Right). In the middle we reproject the graph structure onto the original image, visualizing the high fidelity of the representation, even as the number of nodes reduces to 0.45% of the full grid. Figure taken from [Pub5].

of an RGB image. The reduction 0.45% in comparison to the full grid is hardly noticeable without zooming in. For further details we refer to [Pub5]. Algorithm 1 shows the steps that this algorithm follows for further clarification.

2.2.3.1 Implementation

On a discrete image grid generated by sensor data, Eq. (2.87) becomes a binary partitioning on a discrete graph, which can be solved efficiently by a *maxflow* algorithm, e.g. Boykov-Kolmogorov [16]. In the end variants, such as L^1-Cut-Pursuit [66] or using a real-time Mumford-Shah such as [128] would also be possible candidates to find a candidate partition, yet we did not find these variants to yield either sufficient speed or sufficient accuracy around edges to be applicable - for algorithms that do not explicitly track the partitioning, the partition also has to be computed from the final result in an additional post-processing step.

We have chosen the `search-trees` implementation of [15] to solve the discrete version of the binary partition problem stated in Eq. (2.87). This can be done by reformulating the energy into a flow-graph structure with two additional terminal nodes *sink* and *source*. How to assign the right capacities to the edges can be taken from [66] or [131]. A significant bottleneck of this straightforward *maxflow* implementation is that the computation is difficult to parallelize. Thus, the computational time can increase drastically for very large images or other input data. For real-time applications with access to parallelization via GPUs or CPUs with sufficient cores we would recommend porting the entire pipeline into a single framework and using a primal dual algorithm with diagonal preconditioned stepsizes as in [97] not only for the minimal partitions problem but also the binary cuts. Especially running both subroutines on the GPU is potentially highly beneficial for large images. On the other hand, solving the binary cut with a primal-dual algorithm only approximates the solution in finite time and convergence criteria have to be chosen carefully to guarantee accurate results. In contrast *maxflow* termination criteria are straightforward, which is why we focus on *maxflow* in this work, aside from its applicability to weaker hardware with low specifications.

2.2.4 Numerical Evaluation

We consider the experimental setup for stereo matching in Section 2.1.7.2 and apply the piecewise linear lifting discussed in Section 2.2.2 with $L = 8$ labels. As the data term used for running the L^0-CP partitioning we use

$$g(x) = \arg\min_k D(x, k) \tag{2.90}$$

where with a slight abuse of notation D refers to the discrete cost term Eq. (2.63) identifying x with its pixel coordinates (i, j). This is motivated by the intuition that constant regions of pointwise minimizing disparities likely induce constant regions of the original data term. Figure 2.11 (top) gives a visual impression of the approximation behavior of the graph reduction on an exemplary stereo image. Figure 2.11 (bottom) visualizes the time vs. the achieved energy values of our method compared to the full stereo matching problem. Note the scale of the x-axis. We can easily reduce the necessary time and memory costs by using the graph-based discretization. Despite of the significant speedup for stereo matching the proposed method still is capable of producing visually pleasing results, as the matching is still computed with respect to all variables, just with an optimally chosen discretization.

We implement the graph-structured optimization of the stereo matching problem via a primal-dual algorithm [27] with diagonal preconditioning [97]. The preconditioning allows us to reconcile the step sizes of the algorithm with the varying sizes of the graph partitions. We use the implementation of this algorithm from https://github.com/tum-vision/prost, which conducts GPU computations with a Matlab wrapper. The L^0-CP implementation is written in Matlab using just the internal *maxflow* implementation.

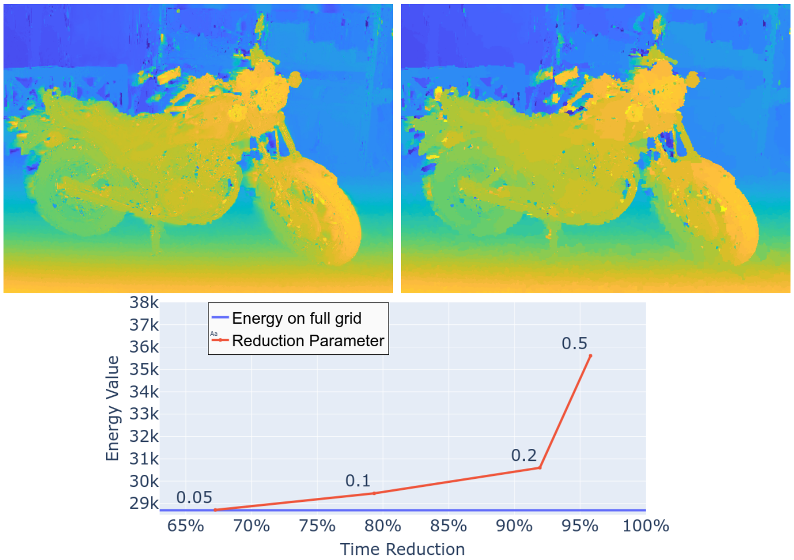


Figure 2.11: Results on stereo matching baselines. *Top:* Comparison of full (left) and proposed reduced (right) matching. Both methods use sublabels [92] between 32 labels. The proposed method uses Cut-Pursuit (with parameter $\alpha_c = 0.1$, a higher parameter corresponds to fewer vertices in the reduced graph) to find a reduced graph, amounting to a time reduction by a factor of 4.8, although the matching quality is near indistinguishable. *Bottom:* Time reduction and energy levels for different parameters α_c, showing the granular relationship between graph reduction and difference in energy value of the matching algorithm. Figure taken from [Pub5].

2.3 High-Dimensional Lifting using Network Priors

In contrast to the ideas of functional lifting elaborated in the previous sections, we will now present a very different idea based on [Pub6] for higher dimensional reparameterizations of Eq. (2.1). Instead of increasing the dimensionality by proposing a reformulation of the problem on a higher-dimensional label space Γ we now reformulate the problem by using a high-dimensional reparameterization . Instead of optimizing over functions $u\colon \Omega \to \Gamma$ directly, we parameterize u as $u(x) = \mathcal{N}(x;\theta)$ for $x \in \Omega$ and some parameterizable model $\mathcal{N}\colon \Omega \times \Theta \to \Gamma$. We optimize over θ where θ is element of a high-dimensional space Θ and the overall problem hence becomes

$$\min_{\theta\in\Theta} \int_\Omega f(x, \mathcal{N}(x;\theta))\,\mathrm{d}x + R(\mathcal{N}(\,\cdot\,;\theta)). \tag{2.91}$$

On the application side we consider a terahertz imaging reconstruction problem, for further details we refer to [151, 152]. We measure signals on a two dimensional, discrete grid $\Omega = \{1, \ldots, n_1\} \times \{1, \ldots, n_2\}$, where n_1 and n_2 are the spatial extents of the pixel gird, and observe a signal for n_z many discrete frequencies $\hat{g}\colon \Omega \times \mathbb{R}^{n_z} \to \mathbb{R}$. The goal is to find a set of parameters explaining the received signal for each pixel individually, i.e. we are looking for a function $u\colon \Omega \to \Gamma$, where Γ is now the space of parameters explaining $\hat{g}$. The quality of the predicted parameters is evaluated by a nonconvex loss function $f\colon \Omega \times \Gamma \to \mathbb{R}$ relating the parameters to the observed signal. We denote the acquired reflected electric field amplitude and phase f at lateral position $x \in \Omega$ by $\hat{g}(x, f)$. This signal is converted into time domain by a Fourier transform along the z axis, yielding

$$g_t = \mathcal{F}_z\{\hat{g}\}. \tag{2.92}$$

In a discrete way g_t can be seen as a complex valued tensor $g_t \in \mathbb{C}^{n_x \times n_y \times n_z}$. Equivalently, we may define g by considering the real and imaginary parts as two separate channels, resulting in a 4D real data tensor $G \in \mathbb{R}^{n_x \times n_y \times n_z \times 2}$.

For each pixel $x \in \Omega$ we now predict a set of parameters $u\colon x \mapsto (\hat{e}, \mu, \sigma, \phi)$ explaining the measurements $g_t(x, \cdot)$. Those parameters predict the time dependent signal using the model function

$$t \mapsto \hat{e}\,\mathrm{sinc}\left(\sigma(t-\mu)\right)\exp\left(-i(\omega t - \phi)\right), \tag{2.93}$$

where ω is some constant. The observed signal and the prediction are then compared using the ℓ_2 distance of the discrete measurements resulting in a nonconvex cost f.

2.3.1 Numerical Evaluation

A thorough numerical evaluation of directly optimizing Eq. (2.1) and Eq. (2.91) has been carried out by my colleague Tak Ming Wong in [Pub6] where we were able to demonstrate numerical advantages of the proposed reparameterization. Interestingly, it is not obvious why this is possible at all as reparameterizations can not avoid bad local. A theoretical analysis and discussion of this circumstance can be found in the next section

2.3.2 Theoretical Aspects of Reparametrizations

In this section we provide a theoretical analysis of the proposed parameterization using neural networks and show that it implicitly corresponds to a variable metric optimization strategy for problem (2.1).

Neglecting the regularizer, problem (2.1) is in itself not coupled on a pixel level. For the sake of simplicity consider for now the general uncoupled problem

$$\min_{u_x\in\Gamma} \sum_{x\in\Omega} h_x(u_x), \tag{2.94}$$

where h_x are (non-convex) cost functions at pixel x. Clearly, minimizing (2.94) reduces to minimizing problem h_x for each pixel x as the sum of the cost functions decouples on a pixel level. Therefore gradient descent on problem (2.94) corresponds to gradient descent on each of the subproblems h_x. Considering a reparameterization of the problem by a continuous function $\tilde{\mathcal{N}}\colon \Theta \to \Gamma^\Omega$ defined as

$\tilde{\mathcal{N}}(\theta)_x := \mathcal{N}(x;\theta)$ yields

$$\min_{\theta\in\Theta}(H \circ \tilde{\mathcal{N}})(\theta). \tag{2.95}$$

for $H(u) := \sum_{x\in\Omega} h_x(u_x)$, and thus generalizes (2.91). Although the problems at pixel level can share a common structure, reformulation (2.95) alone without knowledge of this structure is not advantageous in general due to the the preservation of local geometries, as stated in the following remark.

Remark 2.3.1. Let $\hat{u}$ be a local minimizer of H in the range of $\tilde{\mathcal{N}}$. Then each $\hat{\theta} \in \tilde{\mathcal{N}}^{-1}(\hat{u})$ is also a local minimizer of $H \circ \tilde{\mathcal{N}}$. Hence the local minima are preserved.

Furthermore assuming differentiability of $\tilde{\mathcal{N}}$, consider a continuous interpretation of gradient descent, the *gradient flow* w.r.t H, i.e. a $\theta(t)$ s.t. $\theta'(t) = -\nabla(H \circ \tilde{\mathcal{N}})(\theta(t))$. Then for $u(t) := \tilde{\mathcal{N}}(\theta(t))$ it holds

$$u'(t) = \nabla\tilde{\mathcal{N}}(\theta(t))^T\theta'(t) \tag{2.96}$$

$$= -\nabla\tilde{\mathcal{N}}(\theta(t))^T\nabla(H \circ \tilde{\mathcal{N}})(\theta(t)) \tag{2.97}$$

$$= -\nabla\tilde{\mathcal{N}}(\theta(t))^T\nabla\tilde{\mathcal{N}}(\theta(t))\nabla H(u(t)). \tag{2.98}$$

The matrix $M(t) := \nabla\tilde{\mathcal{N}}(\theta(t))^T\nabla\tilde{\mathcal{N}}(\theta(t)) \in \mathbb{R}^{|\Omega|\times|\Omega|}$ is positive semi-definite and hence $-M(t)\nabla H(u(t))$ is a descent direction. We hypothesize that for certain problem classes h_x as studied in the numerical experiments the temporally changing implicit gradient preconditioning with $M(t)$ is advantageous in terms of training dynamics. In particular, networks with a large receptive field such as a U-net typically yield dense matrices $\nabla\tilde{\mathcal{N}}(\theta(t))^T\nabla\tilde{\mathcal{N}}(\theta(t))$ and thus induce changes in predictions u_x even if $\frac{\partial H}{\partial u_x} = 0$. To obtain a deeper understanding of why a neural net reparameterization can indeed be advantageous it is instructive to consider the findings in [133]. There the studied problems are almost identically to Eq. (2.95) with the exception that H is even convex in the cases of e.g. denoising and image inpainting. In those cases, direct optimization of Eq. (2.94) yields undesired trivial solution whereas the solutions of Eq. (2.95) are of high quality considering that optimization of Eq. (2.95) is not a learning based approach as no training of the network is involved in the classical sense. The authors of [133] thus conclude that convolutional neural network architectures themselves induce a strong image prior on their outputs. The authors dub this phenomenon *deep image prior.* We hypothesize that in our terahertz experiments we experience a very similar effect and hence regard our approach as a generalization of [133] to non-convex functions h_x.

Chapter 3

Beyond Convex Problems: Provable Guarantees and Expressivity in the Light of Machine Learning

The optimization strategies discussed in the previous chapter commonly pursue guarantees of some sort for the obtained solutions. This is rendered possible by the precise mathematical formulations imposing an assumed structure of the problem. Depending on those assumptions guarantees can be invoked by tools from optimization theory. However, the less structured the problems are, the more difficult it gets to derive provable statements. In this section we will examine those aspects in the light of machine learning related problems. Therefore let us define a basic overview of supervised learning which will be used in various alterations in the course of this section. Given some data points $\{x_1, \dots, x_N\} \subset \mathbf{X}$ for some set $\mathbf{X}$ and some related target or label information $\{y_1, \dots, y_N\} \subset \mathbf{Y}$ for some set $\mathbf{Y}$ supervised machine learning is concerned with the task of finding a function explaining the data. More formally, we assume the data $\{(x_1, y_1), \dots, (x_N, y_N)\}$ to be i.i.d. drawn from a joint distribution $p(x, y)$. The conditional distribution $p(y \mid x)$ is approximated by a parameterized mapping $\mathcal{N}\colon \mathbf{X} \times \Theta \to \mathbf{Y}$ using some loss function $\mathcal{L}\colon \mathbf{Y}^2 \to \mathbb{R}$. The approximation problem can now be stated as

$$\min_{\theta \in \Theta} \; \mathbb{E}_{(x,y)\sim p} \left[\mathcal{L}(\mathcal{N}(x; \theta), y) \right]. \tag{3.1}$$

As the underlying distribution p, however, is unknown in practice only the approximation of the empirical loss is tractable by taking the mean over the data samples as estimator of the true expectation by

$$\min_{\theta \in \Theta} \; \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\mathcal{N}(x_i; \theta), y_i). \tag{3.2}$$

This training procedure, however, does per se not give any guarantees whatsoever that the fitted function $\mathcal{N}(\,\cdot\,, \theta)$ is a good approximation w.r.t. p on data that is not contained in the data set. Still Eq. (3.2) is at the heart of most supervised machine learning approaches which have proven successful in practice. The reason for the generalization capabilities hence has to be related to the space of functions $\{\, \mathcal{N}(\,\cdot\,; \theta) \mid \theta \in \Theta \,\}$ considered as candidates. Common models for vision tasks include convolutional deep neural network and indeed empirical evidence exists that this family of architectures contains viable approximations to the true unknown image prior p [159, 133].

Despite of the empiric generalization strength provided by those deep architectures they also come with several disadvantages. For one thing minimizing Eq. (3.1) becomes delicate as the depth and expressivity of the network architecture increases. Indeed there has been a lot of research on this field and also effective heuristics like, most prominently, skip connections [51] have been successfully developed. Still it is evident that complex deep architectures tend to be difficult in terms of optimization and mostly lack strong theoretical guarantees. This section discusses several aspects related to the intertwined connection of optimzation and neural network architecture and studies how they effect each other in terms of a network's expressivity, theoretical guarantees and privacy concerns in the field of federated learning.

Section 3.1 transfers ideas from functional lifting to neural network architecture. Particularly activation functions are regarded as a certain kind of splines related to the lifting approach in Example 2.1.11. Introducing more and more labels to these splines allows an adaptive and seamless increase of expressivity. Numerical advantages of this strategy are then shown in the domain of continual learning.

Many applications of machine learning in computer vision tackle inverse problems. In contrast to machine learning models classical approaches make use of mathematical assumptions and are able to derive mathematical guarantees of the obtained reconstructions. In Section 3.2 we present an approach for incorporating learnable models into a framework based on spectral regularization. This method allows for a learnable data prior and at the same time features provable convergence properties for the calculated solutions.

In Section 3.3 that, even though hardly any provable properties exist, the optimization of very high-dimensional problems in the paradigm of machine learning still yields solutions of a certain quality. This is especially interesting as the studied scenario is relevant for privacy in the context of federated learning and hence shows that relying on the difficulty of large optimization problems does not protect against attacks breaching data security.

3.1 Adaptive Network Architectures Via Linear Splines

The driving force of functional lifting as elaborated in Chapter 2 is the definition of the lifting map φ increasing the dimensionality of the reformulated problem and hence also yielding lower energy solutions. This section based on [Pub3] picks up the idea of using a lifting map for increasing dimensionality in a completely different context showing that the idea of increasing dimensionality and thus also increasing expressivity of the search space is beneficial as an idea in itself. We will interpret the commonly used activations functions called ReLU as a certain kind of dimensionality lifting in the hidden layers of artificial neural networks and extend the idea to spline based activation functions by means of drawing connections to the sampling based lifting map as used in Example 2.1.10. We evaluate this approach and show numerical advantages in the context of continual learning.

3.1.1 Problem Description

The power of modern machine learning approaches is based on deeply nested functions that typically alternate between a linear transfer function and certain non-linearities, often called *activation functions*. Rectified linear units (ReLUs), $\sigma(x) = \max(x, 0)$, are among the most popular non-linearities. However, they have a significant disadvantage. For all $x < 0$, the gradient is zero, which means that there is a loss of information. Therefore, researchers considered variants such as leaky ReLUs, parametrized ReLUs [50] and maxout units [48]. Interestingly, all aforementioned variants represent instances of the general class of continuous piecewise linear functions, also known as *linear splines.* Linear splines have recently attracted researchers attention, e.g. they form the *lifting layers* introduced in [93], see Proposition 1, and have interesting theoretical properties [134]. In this section we investigate the idea of using linear splines as activation functions and iteratively increase their number of kinks to adapt to a continuously increasing complexity of incoming training data.

To formalize our approach we follow an approach akin to Example 2.1.11 and define the lifting as a function mapping from $[\gamma_1, \gamma_n]$ to $\mathbb{R}^n$ by

$$\varphi(x) = \alpha e_k + (1-\alpha)e_{k+1} \tag{3.3}$$

where e_k represents the k-th unit vector in $\mathbb{R}^n$ and $\alpha \in [0,1]$ such that $\alpha\gamma_k + (1-\alpha)\gamma_{k+1} = x$. Note that for $x \in [\gamma_k, \gamma_{k+1}]$ the k-th and $k+1$-th components of $\varphi(x)$ express the convex combination of x w.r.t. the labels t_k and t_{k+1}. Thus, a spline interpolation between two given function values z_k at t_k and z_{k+1} at t_{k+1} is given by $\varphi(x)_k \cdot z_k + \varphi(x)_{k+1} \cdot z_{k+1}$. More generally, given n function values $z_1, \ldots, z_n$ we can calculate the linear spline interpolation at some $x \in [\gamma_1, \gamma_n]$ by

$$\langle z, \varphi(x) \rangle, \tag{3.4}$$

where $z = (z_1, \ldots, z_n)^T$, cf. [93, Proposition 1]. The restriction $x \in [\gamma_1, \gamma_n]$ can easily be avoided by linearly extrapolating the linear functions defined on $[\gamma_1, \gamma_2]$ and $[\gamma_{n-1}, \gamma_n]$ to the entire real line. Equation (3.4) is the type of activation function we will consider in this work.

3.1.2 Adaptive Architectures with Lifting Layers

While the work [93] considered the lifting dimension n to be fixed, we consider adapting it by iteratively adding new labels (kinks). In the particular case of a linear spline, the idea is to refine the spline on an interval $[\gamma_k, \gamma_{k+1}]$ by adding an additional kink $\tilde{\gamma} \in [\gamma_k, \gamma_{k+1}]$. However, this leads to the question of how to initialize the corresponding function value $\tilde{z}$ of the spline at $\tilde{\gamma}$?

Assuming the current spline has already been optimized for the given previous number of kinks, a natural idea is to introduce a new kink without changing the shape of the spline and thereby retaining the current quality of the predictions: By simply setting $\tilde{z} = \langle z, \varphi(\tilde{\gamma}) \rangle$ the lifting function $\hat{\varphi}$ on the new label set $\gamma_1 < \ldots < \gamma_k < \tilde{\gamma} < \gamma_{k+1} < \ldots < \gamma_n$ and $z = (z_1, \ldots, z_k, \tilde{z}, z_{k+1}, \ldots, z_n)^T$ meets

$$\langle \hat{z}, \hat{\varphi}(x) \rangle = \langle z, \varphi(x) \rangle, \text{ for } x \in [\gamma_1, \gamma_n].$$

The new degree of freedom in form of the parameter $\tilde{z}$ allows the spline to resemble the previous one. Therefore, using a descent algorithm, the training performance is provably as good as that of the previous spline. The effect on the test loss is, of course, highly dependent on the data and the true function to be approximated as a higher expressivity comes with the danger of overfitting.

Our approach offers the flexibility to easily adapt the architecture in form of resizing the output of the activation function and the following parameterized layers. Especially the widening of certain layers has proved successful in [144]. This flexible approach to architecture bears the potential of fitting a trained network to new data or even a new task.

3.1.3 Experiments

3.1.3.1 1-D Toy Example

To illustrate our idea we consider a simple 1-dimensional regression problem. Suppose we have an incoming stream of data points $x \in \mathbb{R}$ together with some targets $y = \sin(x) + \epsilon$, where ϵ is additive Gaussian noise with known standard deviation. To approximate the (unknown) function that computes y for a given x, we choose a shallow network of the form (3.4) for learnable parameters z, i.e. a linear spline and adapt the number of labels n, i.e. the number of kinks. For a fixed n and a current fixed set of training examples the parameters z that minimize

$$\sum_{i=1}^{N} (\langle z, \varphi(x_i) \rangle - y_i)^2,$$

can be computed via a linear equation. If any of the resulting predictions $\langle z, \varphi(x_j) \rangle$ differ from their target y_j by more than 3 times the standard deviation of the noise, we introduce a new label (kink), by first ensuring $x_i \in [\gamma_1, \gamma_n]$ for all training data, and subsequently splitting intervals $[\gamma_i, \gamma_{i+1}]$ in half if necessary. Over the course of this procedure the predicting function learns to adapt to new data and resembles the true distribution quite well as can be seen in Fig. 3.1.

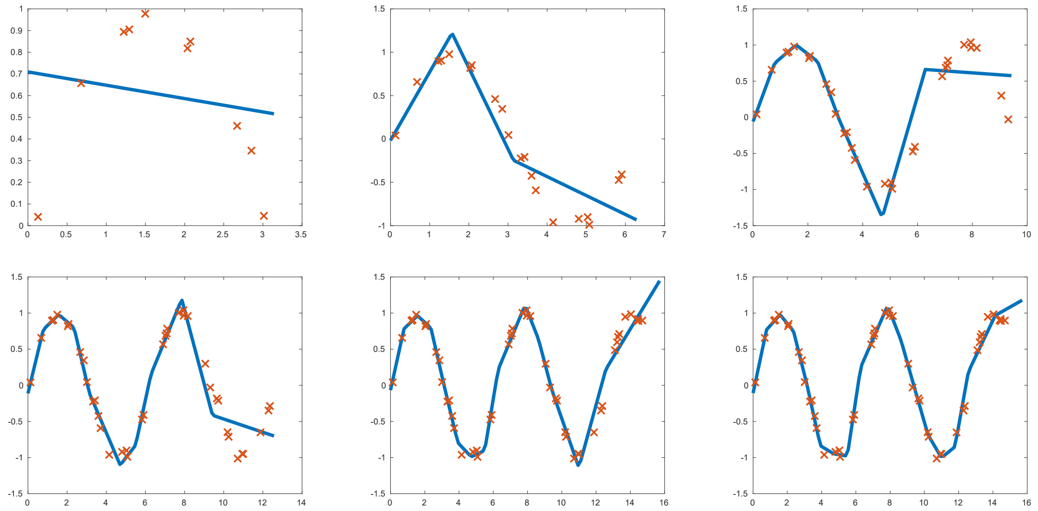


Figure 3.1: A stream of incoming data points (increasing number of red x in the above figures) is fitted by a linear spline (blue) that iteratively increases its number of kinks. Figure taken from [Pub3].

3.1.3.2 Adaptive Lifting in Networks

After the introductory example we apply the adaptive lifting approach in a deeper neural network on the CIFAR-10 classification problem. As starting point we use the “Deep MNIST for experts model” by TensorFlow[1] and replace all ReLUs by linear splines (represented by lifting layers followed by 1×1-convolutions resembling (3.4)). In our tests we consider three different networks which share the same architecture and only differ by the number of labels used for the spline activations. The first two models have a fixed number of labels where we use 2 for the small model and 10 for large model. The

[1] https://www.tensorflow.org/tutorials/layers

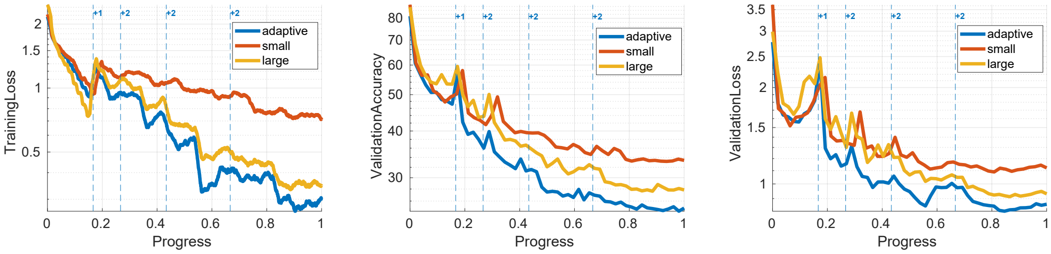


Figure 3.2: Comparison of the three described networks during the training procedure. Figure taken from [Pub3].

adaptive model, in contrast, starts with 2 labels, adds one label after 10 epochs training on 10% of the data set, two labels after 2 epochs on 30% of the data set, two further labels after 2 epochs on 50% of the data set and again two further labels after 2 epochs on 70% of the data set. Lastly we train for 2 epochs on the whole data set. The above training procedure is used for all three networks with two networks keeping the number of labels/kinks fixed. As depicted in Fig. 3.2 the training loss of the large network initially drops the fastest while the validation loss remains comparably high, which might indicate overfitting. In contrast, the adaptive net follows the learning curve of the small architecture at this first stage. As new data is added to the training set and the number of labels is increased for the adaptive network it pulls away from the small architecture and uses its newly gained expressivity. Interestingly, the validation error indicates that the adaptive enhancement of the label set can even lead to a better generalization compared to network that retained a large number of kinks throughout the entire training.

3.2 Learned spectral regularization

A variety of problems in computer vision can be formulated as linear *inverse problems*, i.e. the task is to reconstruct image data which explain observations most faithfully under a given linear forward model. One of the main challenges in linear inverse problems is that many of such problems are ill-posed in the sense that the solution does not depend on the data continuously. In order to achieve a continuous dependence on the data a prior information about the space of possible solutions has to be incorporated. Classically, hand crafted priors are used resulting in convex problems which provable yield continuity and convergence to the true solutions for vanishing noise levels. Often those solutions can even be calculated using *spectral regularization*, i.e. by constructions of pseudoinverse operators using regularized singular values of the forward operator. Whereas those handcrafted priors yield strong guarantees needed for robust reconstruction methods, they are often inferior modern learning based priors which, however, lack those guarantees. In the following we present an approach incorporating a learning based prior into classical spectral regularization methods thus unifying the benefits of learning based priors with provable guarantees. This section is based on [Pub1].

3.2.1 Introduction

The classical theory for linear inverse problems considers a linear operator $A : X \to Y$ between Hilbert spaces X and Y and asks the question how to reconstruct an unknown $x \in X$ from an observation $y^\delta \in Y$ given as $y^\delta = Ax + n^\delta$, where $n^\delta \in Y$ represents noise of magnitude $\|n^\delta\| = \delta$. As soon as A is a compact linear operator with infinite dimensional range, A admits a singular value decomposition (SVD)

$$Ax = \sum_{n=1}^{\infty} \sigma_n \langle x, u_n \rangle v_n \tag{3.5}$$

with singular vectors $u_n \in X$ and $v_n \in Y$, and zero being an accumulation point of the corresponding singular values $\sigma_n > 0$. Thus, the pseudoinverse $A^\dagger$ of A becomes

$$A^\dagger y = \sum_{n=1}^{\infty} \frac{1}{\sigma_n} \langle y, v_n \rangle u_n. \tag{3.6}$$

Since zero is an accumulation point of the σ_n, the pseudoinverse is unbounded, i.e. discontinuous, see [115, Thm. 4.18 (e)]. The idea of spectral regularization is to replace the pseudoinverse $A^\dagger$ by a family of operators $R_\alpha : Y \to X$ given by

$$R_\alpha y = \sum_{n=1}^{\infty} g_\alpha(\sigma_n) \langle y, v_n \rangle u_n \tag{3.7}$$

in such a way that R_α is a continuous on the entire space Y and that it converges pointwise to $A^\dagger$ as the *regularization parameter* α goes to zero. A suitable choice of α as a function of δ (a-priori choice) or of (δ, y^δ) (a-posteriori choice) allows to reestablish the continuous dependence of the solution on the data in the sense that

$$\|R_\alpha y^\delta - A^\dagger y\| \to 0 \qquad \text{as} \qquad \delta \to 0. \tag{3.8}$$

We investigate learning a function $\mathcal{N}$ parameterized by θ such that

$$g_\alpha(\sigma) = \mathcal{N}(\sigma, \delta, y^\delta; \theta) \tag{3.9}$$

provably satisfies (3.8). We discuss a-priori as well as a-posteriori choices of the regularization, demonstrate how learning improves the results over classical choices, and illustrate the performance boost when turning to nonlinear reconstruction operators.

3.2.2 Related Work

Classically, the spectral regularization approaches have been defined manually, e.g. via functions g_α of the form

$$g_\alpha(\sigma) = \begin{cases} \frac{1}{\sigma} & \text{if } \sigma \geq \alpha \\ 0 & \text{otherwise} \end{cases}, \tag{Truncated SVD}$$

$$g_\alpha(\sigma) = \frac{1}{\sigma + \alpha}, \tag{Lavrentiev}$$

$$g_\alpha(\sigma) = \frac{\sigma}{\sigma^2 + \alpha} \tag{Tikhonov}$$

along with suitable parameter choice rules α, see e.g. [40]. The fact that Tikhonov regularization is equivalent to solving a minimization problem with a quadratic penalty, subsequently gave rise to nonlinear variational methods for which convergent regularization methods similar to (3.8) can be guaranteed in different measures of distances, see e.g. [22, 8].

Due to the rise of deep learning techniques many benchmarks are currently dominated by directly learning a mapping from y^δ to a desired solution x via a neural network. Such techniques are, however, largely lacking a theoretical understanding and rarely even take the noise level δ into account explicitly (see e.g. [160] for an exception).

Several works have considered hybrid methods between regularization techniques that allow for detailed analysis and learning based approaches: For instance, [32, 2, 61] use minimization/regularization algorithms as a template for network architectures, [14, 77, 122, 71] optimize over a learned latent space that contains mostly realistic solutions, [160] consider regularization by parameterization via deep neural networks, and algorithmic schemes that replace the proximal operator of a regularizer with a neural network have been studied in [107, 84]. Such techniques do, however, not correspond to minimization problems anymore unless the network possesses very specific properties, see [112]. Beyond this, safeguarding techniques such as [86] or bilevel optimization problems that learn a parameterized variational regularization (e.g. [118, 33, 23, 45]) are the only way to remain in the regime of energy minimization methods. The additional difficulty of defining networks in such a way that they act on continuous functions rather than their fixed discretizations, makes a convergence analysis in the sense of (3.8) difficult and rare. On a related note, the work [113] considers learning itself as an ill-posed inverse problem to derive convergence properties similar to (3.8).

3.2.3 Learned Spectral Regularizers

For learning spectral regularizations we consider different types of parameterized functions to represent g_α:

- A-priori parameter choices: We parameterize $\mathcal{N}(\sigma, \delta, y^\delta; \theta)$ by two classical approaches, namely a learned Lavrentiev and a learned Tikhonov regularization given via

  $$\mathcal{N}_{\text{Lav}}(\sigma, \delta; \theta) = \frac{1}{\sigma + \delta^p \tilde{N}(\sigma, \delta; \theta)}, \qquad \mathcal{N}_{\text{Tik}}(\sigma, \delta; \theta) = \frac{\sigma}{\sigma^2 + \delta^q \tilde{N}(\sigma, \delta; \theta)}, \tag{3.10}$$

  with $p \leq 1$, $q \leq 2$, and a network

  $$\tilde{N}(\sigma, \delta; \theta) = \theta_{\text{scale}} \cdot \text{sigmoid}(\text{FCN}(\sigma, \delta; \theta)) \tag{3.11}$$

  with a 2-layer fully connected network *FCN*, and one additional scale parameter θ_{scale}. To also make a comparison to classical (but noise-level optimal) regularization choices, we additionally drop the dependence of $\tilde{\mathcal{N}}$ on σ in (3.11) and refer to these methods as the classical Lavrentiev and Tikhonov regularizations.

- A-posteriori parameter choice: Many papers have demonstrated great success in directly predicting a solution $\hat{x}$ for given data y^δ by exploiting spatial regularity of x, e.g. through convolutional neural networks in imaging applications. While the straight forward application of such networks lacks any kind of convergence guarantee, such techniques can be converted to a-posteriori spectral regularizations with convergence guarantees via suitable projectors. Let $\hat{x} = \mathcal{G}(y^\delta; \theta)$ be some prediction of a neural network $\mathcal{G}$ designed to solve the underlying problem directly. Then the choice

$$g_\alpha(\sigma_n) = \frac{\langle u_n, \mathcal{G}(y^\delta; \theta)\rangle}{\langle v_n, y^\delta\rangle} \tag{3.12}$$

results in the spectral regularization yielding $\mathcal{G}(y^\delta; \theta)$ as a reconstruction result (assuming that $\langle v_n, y^\delta\rangle \neq 0$, and that all σ_n are different). To ensure convergence, we modify such a prediction via the following projection

$$\mathcal{N}(\sigma_n, \delta, y^\delta; \theta) = \text{proj}_{\left[(1-\sqrt{\delta}\theta_l)\frac{\sigma}{\sigma^2+\alpha_l\delta}, (1+\sqrt{\delta}\theta_u)\frac{\sigma}{\sigma^2+\alpha_u\delta}\right]} \left(\frac{\langle u_n, \mathcal{G}(y^\delta; \theta)\rangle}{\langle v_n, y^\delta\rangle}\right) \tag{3.13}$$

with learnable parameters θ_l and θ_u. The projection can be interpreted as remaining in between a strong and a weak Tikhonov regularization ($\alpha_u << \alpha_l$) up to some tolerance with deceases to zero as $\delta \to 0$. For $\mathcal{G}$ we choose an architecture loosely motivated by unrolling a proximal gradient descent algorithm, see Section 3.2.5.

With the above choices, we can state the following convergence result:

Proposition 3.2.1 (Convergent spectral regularization methods)**.** *Both learning based approaches, the a-priori choice* (3.10) *and the a-posteriori choice* (3.13)*, are convergent regularization methods in the sense of* (3.8) *provided that* $q < 2$ *and* $p < 1$*.*

Proof. The techniques used for proving our proposition are well known and follow the arguments in [40]. For the sake of completeness, we will still give an overview of the proof. The central idea is to show the following results that ensure a convergent spectral regularization method: For a regularization of the form (3.7) the following criteria ensure (3.8).

$$g_\alpha(\sigma) \leq C_\alpha \text{ for all } \sigma > 0, \tag{3.14a}$$

$$g_\alpha(\sigma) \overset{\alpha \to 0}{\to} \frac{1}{\sigma} \text{ for all } \sigma > 0, \tag{3.14b}$$

$$\sigma g_\alpha(\sigma) \leq \tilde{C} < \infty \text{ for all } \alpha,\ \sigma > 0, \tag{3.14c}$$

$$\delta C_{\alpha(\delta, y^\delta)} \overset{\delta \to 0}{\to} 0. \tag{3.14d}$$

The condition (3.14a) ensures that R_α is a continuous linear operator for any fixed $\delta > 0$ because $\|R_\alpha\| = C_\alpha$. For y in the domain of $A^\dagger$, $y \in \mathcal{D}(A^\dagger)$, and y^δ with $\|y - y^\delta\| \leq \delta$ we estimate

$$\|A^\dagger y - R_\alpha y^\delta\| = \|A^\dagger y - R_\alpha y + R_\alpha y - R_\alpha y^\delta\| \leq \|A^\dagger y - R_\alpha y\| + \|R_\alpha\|\delta. \tag{3.15}$$

Now using $\|R_\alpha\| = C_\alpha$ condition (3.14d) ensures that the second term in the above estimate converges to zero for $\delta \to 0$. As for the first term we find

$$\|R_\alpha y - A^\dagger y\|^2 = \sum_{n=1}^{\infty} \left(g_\alpha(\sigma_n) - \frac{1}{\sigma_n}\right)^2 |\langle v_n, y\rangle|^2, \tag{3.16}$$

$$= \sum_{n=1}^{\infty} (\sigma g_\alpha(\sigma_n) - 1)^2 \frac{1}{\sigma_n^2} |\langle v_n, y\rangle|^2. \tag{3.17}$$

Due to condition (3.14c) the above sum remains bounded independent of α and therefore is uniformly convergent, such that summation and a limit of $\delta \to 0$ can be exchanged. Finally, condition (3.14b) ensures that the above term converges to zero.

Left to verify is that our learnable architectures satisfy the conditions (3.14). For both a-priori choice rules (3.14a) holds due to the sigmoid function being strictly greater than zero for all inputs. Condition (3.14b) holds for $p, q > 0$ since the sigmoid function is bounded by 1. Condition (3.14c) holds with $\tilde{C} = 1$, and (3.14d) holds if $p < 1$ and $q < 2$. Similarly, the projection operator of the a-posteriori choice (3.13) ensures the conditions (3.14) to be met.

□

3.2.4 Numerical Experiments

To compute the solution of inverse problems numerically, we need to discretize any infinite dimensional problem to a finite one, e.g., by considering a suitable subspace. While this step alone reintroduces regularity as finite dimensional linear operators can never be discontinuous, the resulting problem still remains ill-conditioned due to a quick decay of the singular values σ_n.

To investigate the behavior of our regularization strategy, we consider two inverse problems: The differentiation as well as deblurring of a function $y : [0, 1] \to \mathbb{R}$, giving rise to the linear operators

$$A_{\text{int}}x(t) = \int_0^t x(s)\,\mathrm{d}s, \qquad A_{\text{blur}}x(t) = \int_0^1 g(s-t)x(s)\,\mathrm{d}s, \tag{3.18}$$

for g being a Gaussian kernel. In both cases we discretize x by evaluating it at positions $(x_n)_{1\leq n\leq N}$ and approximate the integrals of the operators by simple summations. Further details on the training can be found in Section 3.2.6.

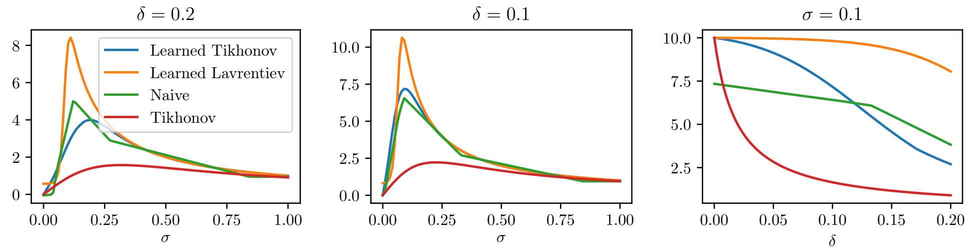


Figure 3.3: Exemplifying the results of the a-priori parameter choice rules. Left and middle: $\mathcal{N}(\sigma, \delta; \theta)$ as a function of σ for two different noise levels $\delta = 0.2$ and $\delta = 0.1$. Right: $\mathcal{N}(\sigma, \delta; \theta)$ as a function of δ for a fixed $\sigma = 0.1$. Figure taken from [Pub1].

Figure 3.3 (left and middle) shows the learned Tikhonov and Lavrentiev regularizations along with the classical Tikhonov regularization and a naive approach, in which g_α is directly parameterized as a fully connected network depending on σ and δ, for two different noise levels δ as a function of σ. As we can see, all learned regularizers choose a Tikhonov-type shape of g_α, but allow much larger $g_\alpha(\sigma)$ for σ on a medium scale. In particular, the learned Tikhonov and direct (naive) learning yield remarkably similar shapes of g_α. Looking at the spectral regularizers for fixed $\sigma = 0.1$ (right plot in Fig. 3.3) as a function of δ, one can see that the Tikhonov, learned Tikhonov and Lavrentiev regularizers yield a value of $g_\alpha(\sigma) = 1/\sigma$ for $\delta = 0$. The direct (naive) approach, however, fails to yield such a value, which directly implies that this method is not a convergent regularization in the sense of (3.8). We conclude that the choice of the architecture with built-in behavior is crucial for obtaining theoretical guarantees.

		Naive	Lav.	Tik.	Learned Lav.	Learned Tik.	A-Post.
Deblur	training	29.58	20.05	27.17	28.59	29.73	31.89
	test	29.56	19.98	27.00	28.55	29.67	31.51
Diff.	training	28.93	21.35	26.44	28.94	29.27	31.63
	test	29.00	21.30	26.45	28.99	29.31	30.74

Table 3.1: PSNR values during training and testing for deblurring and differentiation for various different regularization strategies.

As for the overall performance of each approach, Section 3.2.4 shows the PSNR values for all methods over training and testing in both applications for a fixed discretization of $N = 50$. As we can see, the learning-based methods clearly outperform the classical approach while still remaining provably convergent in the sense of Proposition 3.2.1. Moreover, the learned Tikhonov parameterization is superior to its Lavrentiev counterpart. The naive approach does yield good PSNR values (although with a high initialization-depending variance), but does not yield a convergent regularization, i.e., does not yield faithful results for small noise levels. The best results by far are obtained by the learned a-posteriori approach, which converts a direct prediction of the solution x to a spectral regularization.

Shown in Fig. 3.4 are two curves of $g_\alpha = \mathcal{N}(\cdot, \delta, y^\delta; \theta)$ for the same y and δ, such that merely the realizations of the noise differ. As we can see, the learned a-posteriori choice is non-monotone, does not yield smooth curves such as in Fig. 3.3, and differs significantly for different realizations of noisy data. Yet, the PSNR values of this approach is significantly higher, indicating the importance of non-linear regularizations that vary for different y^δ.

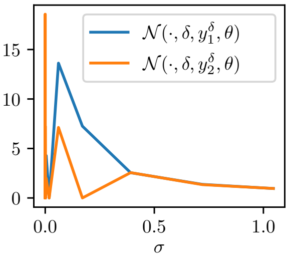


Figure 3.4: Comparison of noise instances. Figure taken from [Pub1].

3.2.5 Details on Network Architectures

The fully-connected networks in the learned Lavrentiev and learned Tikhonov models consist of two fully-connected layers with hidden dimension 10. The naive network consists of 3 fully-connected layers with hidden dimension 2 → 100 → 100 → 1.

For the a-posteriori prediction $\hat{x} = \mathcal{G}(y^\delta; \theta)$ we utilize an iteration inspired by proximal gradient descent

$$x_i = \mathcal{G}_i(x_{i-1} - \tau A^T(Ax_{i-1} - y); \theta_i) \tag{3.19}$$

where $x_0 = 0$ and $\tau = 1$. The networks $\mathcal{G}_i$ themselves are 3-layer convolutional networks with hidden layer sizes of 10. For the convolutions we use zero padding and we fix the minimum kernel size such that the receptive field of the output covers the entire input x_{i-1}.

3.2.6 Training

For each resolution $N \in \{10, 11, \dots, 50\}$ we generate $R = 5000$ many training examples $x_{i,N}$ by sampling random superpositions of sine and cosine functions, applying the discretized operator A and adding zero-mean Gaussian noise of different standard deviations $\delta_{i,N}$ with $\delta_{i,N} \in [0, 0.2]$ to obtain simulated data $y_{i,N}^{\delta_{i,N}}$. Then we train our networks in a supervised way via

$$\min_\theta \sum_{N=10}^{50} \frac{1}{R} \sum_{i=1}^{R} \left\| \sum_{n=1}^{N} \mathcal{N}(\sigma_{n,N}, \delta_{i,N}, y_{i,N}^{\delta_{i,N}}; \theta) \langle y_{i,N}^{\delta_{i,N}}, v_{n,N} \rangle u_{n,N} - x_{i,N} \right\|^2 . \tag{3.20}$$

The superpositions of sine and cosine functions are realized by

$$f(x) = \cos(\omega_1 x) + \gamma \sin(\omega_2 x) \tag{3.21}$$

where ω_1 and ω_2 are drawn from a standard normal distribution and γ is sampled uniformly from the interval $[-1, 1]$.

3.3 Inverting Gradients

Many of the previously presented methods provide some sort of guarantees which are often required in security critical applications. However, many real world problems are notoriously difficult to solve and are out of the range for algorithms providing robustness guarantees. Some of these problems indeed seems so difficult that they create the impression that it is practically impossible to find solutions of sufficient quality. This, however, may create a false sense of security in privacy critical contexts. Based on [Pub4], in the following we give an example of such false sense of security in an inverse problem arising from the application of federated learning. More specifically, we aim for the reconstruction of image data from update steps of neural network parameters and showcase, that even multiple update steps involving multiple data samples in the context of federated averaging do not prevent from a possible breach of privacy.

3.3.1 Problem Description

Federated or collaborative learning [34, 123] is a distributed learning paradigm that has recently gained significant attention as both data requirements and privacy concerns in machine learning continue to rise [82, 57, 154]. The basic idea is to train a machine learning model, for example a neural network, by optimizing the parameters θ of the network $\mathcal{N}$ using a loss function $\mathcal{L}$ and exemplary training data consisting of input images x_i and corresponding labels y_i in order to solve

$$\min_\theta \frac{1}{N}\sum_{i=1}^N \mathcal{L}(\mathcal{N}(x_i;\theta),y_i). \tag{3.22}$$

We consider a distributed setting in which a *server* wants to solve (3.22) with the help of multiple *users*

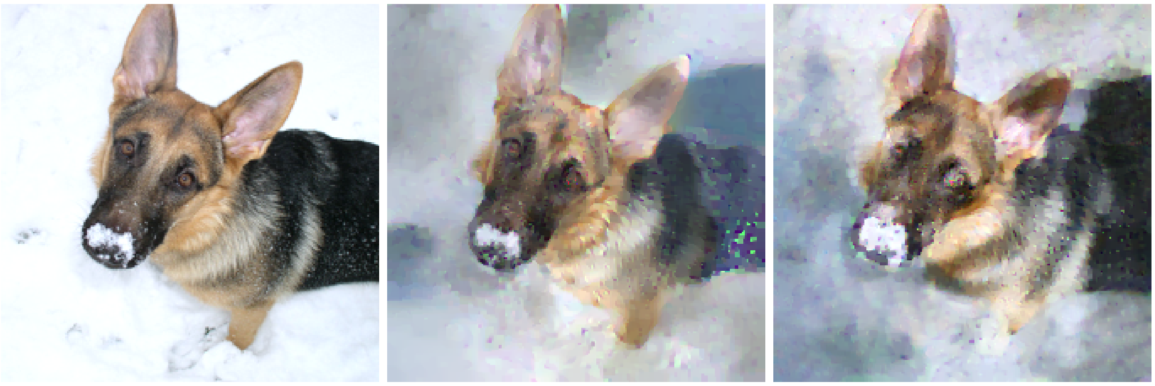


Figure 3.5: Reconstruction of an input image x from the gradient $\nabla_\theta \mathcal{L}_\theta(x,y)$. Left: Image from the validation dataset. Middle: Reconstruction from a trained ResNet-18 trained on ImageNet. Right: Reconstruction from a trained ResNet-152. In both cases, the intended privacy of the image is broken. Note that previous attacks cannot recover either ImageNet-sized data [163] or attack trained models. Figure taken from [Pub4].

that own training data (x_i,y_i). The idea of federated learning is to only share the gradients $\nabla_\theta \mathcal{L}_\theta(x_i,y_i)$ instead of the original data (x_i,y_i) with the server, where we define $\mathcal{L}_\theta(x_i,y_i) := \mathcal{L}(\mathcal{N}(x_i;\theta),y_i)$. The server then subsequently accumulates the gradients in order to update the overall weights. Using gradient descent the server's updates could, for instance, constitute

$$\theta^{k+1} = \underbrace{\theta^k - \tau\sum_{i=1}^N}_{\text{server}} \underbrace{\nabla_\theta \mathcal{L}_{\theta^k}(x_i,y_i)}_{\text{users}}. \tag{3.23}$$

The updated parameters θ^{k+1} are sent back to the individual users. The procedure in Eq. (3.23) is called *federated SGD*. In contrast, in *federated averaging* [63, 82] each user computes several gradient descent steps locally, and sends the updated parameters back to the server. Finally, information about (x_i, y_i) can be further obscured, by only sharing the mean $\frac{1}{t}\sum_{i=1}^{t} \nabla_\theta \mathcal{L}_{\theta^k}(x_i, y_i)$ of the gradients of several local examples, which we refer to as the *multi-image* setting.

Distributed learning of this kind has been used in real-world applications where user privacy is crucial, e.g. for hospital data [56] or text predictions on mobile devices [13], and it has been stated that "Privacy is enhanced by the ephemeral and focused nature of the [federated learning] updates" [13]: model updates are considered to contain less information than the original data, and through aggregation of updates from multiple data points, original data is considered impossible to recover. In this work we show analytically as well as empirically, that parameter gradients still carry significant information about the supposedly private input data as we illustrate in Fig. 3.5. We conclude by showing that even *multi-image federated averaging* on realistic architectures does not guarantee the privacy of all user data, showing that out of a batch of 100 images, several are still recoverable.

As threat model we investigate an *honest-but-curious* server with the goal of uncovering user data: The attacker is allowed to separately store and process updates transmitted by individual users, but may *not* interfere with the collaborative learning algorithm. The attacker may not modify the model architecture to better suit their attack, nor send malicious global parameters that do not represent the actually learned global model. The user is allowed to accumulate data locally in Section 3.3.6.

3.3.2 Overview

In this section we will discuss privacy limitations of federated learning first in an academic setting, where we prove that the input to any fully connected layer can be reconstructed analytically independent of the remaining network architecture. The we show that for the case of one image reconstruction of input data from gradient information is possible for realistic deep and non-smooth architectures with both, trained and untrained parameters.

Then we consider the implications that the findings have for practical scenarios, finding that reconstruction of multiple, separate input images from their averaged gradient is possible in practice, over multiple epochs, using local mini-batches, or even for a local gradient averaging of up to 100 images.

Previous related works that investigate recovery from gradient information have been limited to shallow networks of less practical relevance. Recovery of image data from gradient information was first discussed in [96, 95] for neural networks, who prove that recovery is possible for a single neuron or linear layer. For convolutional architectures, [145] show that recovery of a single image is possible for a 4-layer CNN, albeit with a significantly large fully-connected (FC) layer. Their work first constructs a "representation" of the input image, that is then improved with a GAN. [163] extends this, showing for a 4-layer CNN (with a large FC layer, smooth sigmoid activations, no strides, uniformly random weights), that missing label information can also be jointly reconstructed. They further show that reconstruction of multiple images from their averaged gradients is indeed possible (for a maximum batch size of 8). [163] also discuss deeper architectures, but provide no tangible results. A follow-up [162] notes that label information can be computed analytically from the gradients of the last layer. These works make strong assumptions on the model architecture and model parameters that make reconstructions easier, but violate the threat model that we consider in this work and lead to less realistic scenarios.

The central recovery mechanism discussed in [145, 163, 162] is the optimization of an Euclidean matching term. The cost function

$$\arg\min_{x} ||\nabla_\theta \mathcal{L}_\theta(x, y) - \nabla_\theta \mathcal{L}_\theta(\overline{x}, y)||^2 \tag{3.24}$$

is minimized to recover the original input image $\overline{x}$ from a transmitted gradient $\nabla_\theta \mathcal{L}_\theta(\overline{x}, y)$. This optimization problem is solved by an L-BFGS solver [78]. Note that differentiating the gradient of $\mathcal{L}$ w.r.t. to x requires a second-order derivative of the considered parameterized function and L-BFGS needs to construct a third-order derivative approximation, which is challenging for neural networks with ReLU units for which higher-order derivatives are discontinuous.

A related, but easier problem, compared to the full reconstruction of input images, is the retrieval of input attributes [85, 44] from local updates, e.g. does a person that is recognized in a face recognition system wear a hat. Information even about attributes unrelated to the task at-hand can be recovered from deeper layers of a neural network, which can be recovered from local updates.

Our problem statement is furthermore related to model inversion [43], where training images are recovered from network parameters after training. This provides a natural limit case for our setting. Model inversion generally is challenging for deeper neural network architectures [161] if no additional information is given [43, 161]. Another closely related task is inversion from visual representations [38, 37, 81], where, given the output of some intermediate layer of a neural network, a plausible input image is reconstructed. This procedure can leak some information, e.g. general image composition, dominating colors. However, depending on the given layer, it only reconstructs similar images if the neural network is not explicitly chosen to be (mostly) invertible [55]. As we prove later, inversion from visual representations is strictly more difficult than recovery from gradient information.

3.3.3 Theoretical Analysis: Recovering Images from their Gradients

To understand the overall problem of breaking privacy in federated learning from a theoretical perspective, let us first analyze the question if data $x \in \mathbb{R}^n$ can be recovered from its gradient $\nabla_\theta \mathcal{L}_\theta(x, y) \in \mathbb{R}^p$ analytically.

Due to the different dimensionality of x and $\nabla_\theta \mathcal{L}_\theta(x, y)$, reconstruction quality is surely is a question of the number of parameters p versus input pixels n. If $p < n$, then reconstruction is at least as difficult as image recovery from incomplete data [24, 8], but even when $p > n$, which we would expect in most computer vision applications, the difficulty of regularized "inversion" of $\nabla_\theta \mathcal{L}_\theta$ relates to the non-linearity of the gradient operator as well as its conditioning.

Interestingly, fully-connected layers take a particular role in our problem: As we prove below, the input to a fully-connected layer can always be computed from the parameter gradients analytically independent of the layer's position in a neural network (provided that a technical condition, which prevents zero-gradients, is met). In particular, the analytic reconstruction is independent of the specific types of layers that precede or succeed the fully connected layer, and a single input to a fully-connected network can always be reconstructed analytically without solving an optimization problem.The following statement is a generalization of Example 3 in [95] to the setting of arbitrary neural networks with arbitrary loss functions:

Proposition 3.3.1. *Consider a neural network containing a biased fully-connected layer preceded solely by (possibly unbiased) fully-connected layers. Furthermore assume for any of those fully-connected layers the derivative of the loss $\mathcal{L}_\theta$ w.r.t. to the layer's output contains at least one non-zero entry. Then the input to the network can be reconstructed uniquely from the network's gradients.*

The proposition is a direct consequence of the following two results:

Proposition 3.3.2. *Let a neural network contain a biased fully-connected layer at some point, i.e. for the layer's input $x_l \in \mathbb{R}^{n_l}$ its output $x_{l+1} \in \mathbb{R}^{n_{l+1}}$ is calculated as $x_{l+1} = \max\{y_l, 0\}$ for*

$$y_l = A_l x_l + b_l, \tag{3.25}$$

for $A_l \in \mathbb{R}^{n_{l+1} \times n_l}$ and $b_l \in \mathbb{R}^n_{l+1}$. Then the input x_l can be reconstructed from $\frac{\mathrm{d}\mathcal{L}_\theta}{\mathrm{d}A_l}$ and $\frac{\mathrm{d}\mathcal{L}_\theta}{\mathrm{d}b_l}$, if there exists an index i s.t. $\frac{\mathrm{d}\mathcal{L}_\theta}{\mathrm{d}(b_l)_i} \neq 0$.

Proof. It holds that $\frac{\mathrm{d}\mathcal{L}_\theta}{\mathrm{d}(b_l)_i} = \frac{\mathrm{d}\mathcal{L}_\theta}{\mathrm{d}(y_l)_i}$ and $\frac{\mathrm{d}y_i}{\mathrm{d}(A_l)_{i,:}} = x^T$. Therefore

$$\begin{aligned} \frac{\mathrm{d}\mathcal{L}_\theta}{\mathrm{d}\left(A_l\right)_{i,:}} &= \frac{\mathrm{d}\mathcal{L}_\theta}{\mathrm{d}\left(y_l\right)_i} \cdot \frac{\mathrm{d}\left(y_l\right)_i}{\mathrm{d}\left(A_l\right)_{i,:}} && (3.26) \\ &= \frac{\mathrm{d}\mathcal{L}_\theta}{\mathrm{d}\left(b_l\right)_i} \cdot x_l^T && (3.27) \end{aligned}$$

for $(A_l)_{i,:}$ denoting the i^{th} row of A_l. Hence x_l can can be uniquely determined as soon as $\frac{\mathrm{d}\mathcal{L}_\theta}{\mathrm{d}(b_l)_i} \neq 0$. □

Proposition 3.3.3. *Consider a fully-connected layer (not necessarily including a bias) followed by a ReLU activation function, i.e. for an input $x_l \in \mathbb{R}^{n_l}$ the output $x_{l+1} \in \mathbb{R}^{n_{l+1}}$ is calculated as $x_{l+1} = \max\{y_l, 0\}$ for*

$$y_l = A_l x_l, \tag{3.28}$$

where the maximum is computed element-wise. Now assume we have the additional knowledge of the derivative w.r.t. to the output $\frac{\mathrm{d}\mathcal{L}_\theta}{\mathrm{d}x_{l+1}}$. *Furthermore assume there exists an index i s.t.* $\frac{\mathrm{d}\mathcal{L}_\theta}{\mathrm{d}(x_{l+1})_i} \neq 0$. *Then the input v can be derived from the knowledge of* $\frac{\mathrm{d}\mathcal{L}_\theta}{\mathrm{d}A_l}$.

Proof. As $\frac{\mathrm{d}\mathcal{L}_\theta}{\mathrm{d}(x_{l+1})_i} \neq 0$ it holds that $\frac{\mathrm{d}\mathcal{L}_\theta}{\mathrm{d}(y_l)_i} = \frac{\mathrm{d}\mathcal{L}_\theta}{\mathrm{d}(x_{l+1})_i}$ and it follows that

$$\frac{\mathrm{d}\mathcal{L}_\theta}{\mathrm{d}\left(A_l\right)_{i,:}} = \frac{\mathrm{d}\mathcal{L}_\theta}{\mathrm{d}\left(y_l\right)_i} \cdot \frac{\mathrm{d}\left(y_l\right)_i}{\mathrm{d}\left(A_l\right)_{i,:}} \tag{3.29}$$

$$= \frac{\mathrm{d}\mathcal{L}_\theta}{\mathrm{d}\left(x_{l+1}\right)_i} \cdot x_l^T. \tag{3.30}$$

□

Another interesting aspect in view of the above considerations is that many popular network architectures use fully-connected layers (or cascades thereof) as their last prediction layers. Hence the input to those prediction modules being the output of the previous layers can be reconstructed. Those activations usually already contain some information about the input image thus exposing them to attackers. For example [85] show that these features representations can be mined for image attributes by training an auxiliary malicious classifier that recognizes attributes that are not part of the main task. Further interesting in this regard is the possibility to reconstruct the ground truth label information from the gradients of the last fully-connected layer as discussed in [162]. Finally, Proposition 3.3.1 allows to conclude that for any classification network that ends with a fully connected layer, reconstructing the input from a parameter gradient is strictly easier than inverting visual representations, as discussed in [38, 37, 81], from their last convolutional layer.

3.3.4 A Numerical Reconstruction Method

As image classification networks rarely start with fully connected layers, let us turn to the numerical reconstruction of inputs: Previous reconstruction algorithms relied on two components; the euclidean cost function of Eq. (3.24) and optimization via L-BFGS. We argue that these choices are not optimal for more *realistic* architectures and especially *arbitrary* parameter vectors. If we decompose a parameter gradient into its norm magnitude and its direction, we find that the magnitude only captures information about the state of training, measuring local optimality of the datapoint with respect to the current model (for strongly convex functions the gradient magnitude is even an upperbound on distance to the optimal solution). In contrast, the high-dimensional direction of the gradient can carry significant information, as the angle between two data points quantifies the change in prediction at one datapoint when taking a gradient step towards another [29, 62]. As such we propose to use a cost function based on angles, i.e. cosine similarity, $l(x, y) = \langle x, y\rangle/(||x||||y||)$. In comparison to Eq. (3.24), the objective is not to find images with a gradient that best fits the observed gradient, but to find images that lead to a similar change in model prediction as the (unobserved!) ground truth. This is equivalent to minimizing the euclidean cost function, if one additionally constrains both gradient vectors to be normalized to a magnitude of 1.

We further constrain our search space to images within $[0, 1]$ and add only total variation [114] as a simple image prior to the overall problem, cf. [145]:

$$\arg\min_{x\in[0,1]^n} 1 - \frac{\langle \nabla_\theta \mathcal{L}_\theta(x, y), \nabla_\theta \mathcal{L}_\theta(\overline{x}, y)\rangle}{||\nabla_\theta \mathcal{L}_\theta(x, y)||||\nabla_\theta \mathcal{L}_\theta(\overline{x}, y)||} + \alpha \operatorname{TV}(x). \tag{3.31}$$

Secondly, we note that our goal of finding some inputs x in a given interval by minimizing a quantity that depends (indirectly, via their gradients) on the outputs of intermediate layers, is related to the task of finding adversarial perturbations for neural networks [130, 80, 4]. As such, we minimize Eq. (3.31) only based on the sign of its gradient, which we optimize with Adam [60] with step size decay. Note though that signed gradients only affect the first and second order momentum for Adam, with the actual update step still being unsigned based on accumulated momentum, so that an image can still be accurately recovered.

Applying these techniques leads to the reconstruction observed in Fig. 3.5. We provide a `PyTorch` implementation at https://github.com/JonasGeiping/invertinggradients.

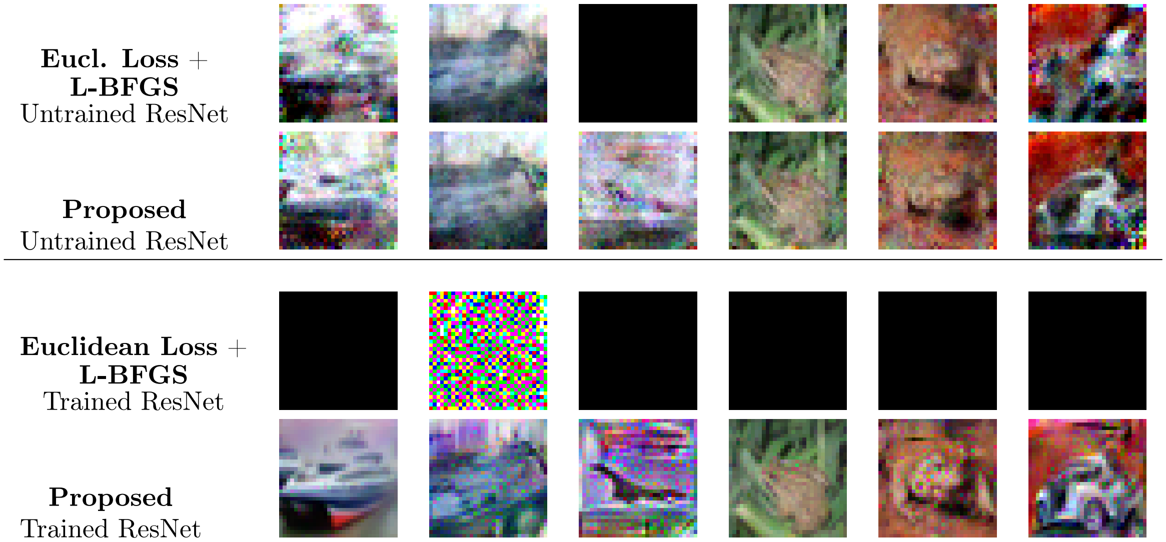


Figure 3.6: Baseline comparison for the network architectures shown in [145, 163].We show the first 6 images from the CIFAR-10 validation set. Figure taken from [Pub4].

This attack is, due to the double backpropagation, roughly twice as expensive as a single minibatch step per gradient step on the objective Eq. (3.31). In this work, we conservatively run the attack for up to 24000 iterations, with a relatively small step size, as computational costs are not our main concern at this moment (and we assume that the attacker that is breaking privacy potentially has order-of-magnitude more computational power than the user), yet we note that smarter step size rules and larger step sizes can lead to successful attacks with a budget of only several hundred iterations.

Remark 3.3.4 (Optimizing label information). While we could also consider the label y as unknown in Eq. (3.31) and optimize jointly for (x, y) as in [163], we follow [162] who find that label information can be reconstructed analytically for classification tasks. Thus, we consider label information to be known.

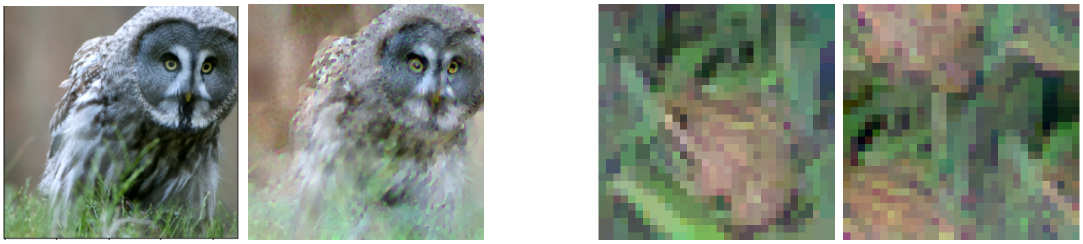
3.3.5 Single Image Reconstruction from a Single Gradient

Similar to previous works on breaking privacy in a federated learning setting, we first focus in the reconstruction of a single input image $x \in \mathbb{R}^n$ from the gradient $\nabla_\theta \mathcal{L}_\theta(x, y) \in \mathbb{R}^p$. This setting serves as a proof of concept as well as an upper bound on the reconstruction quality for the multi-image distributed learning settings we consider in Section 3.3.6. While previous works have already shown that a break of privacy is possible for single images, their experiments have been limited to rather shallow, smooth, and untrained networks. In the following, we compare our proposed approach to prior works, and conduct detailed experiments on the effect that architectural- as well as training-related choices have on the reconstruction.

Comparison to previous approaches. We first validate our approach by comparison to the Euclidean loss (3.24) optimized via L-BFGS considered in [145, 163, 162]. This approach can often fail due to a bad initialization, so we allow a generous setting of 16 restarts of the L-BFGS solver. For a quantitative comparison we measure the mean PSNR of the reconstruction of 32×32 CIFAR-10 images over the first 100 images from the validation set using the same shallow and smooth CNN as in [163], which we denote as “LeNet (Zhu)” as well as a ResNet architecture, both with trained and untrained parameters. Table 3.2 compares the reconstruction quality of Euclidean loss (3.24) with L-BFGS optimization (as in [145, 163, 162]) with the proposed approach. The former works extremely well for the untrained, smooth, shallow architecture, but completely fails on the trained ResNet. We note that [145] applied a GAN to enhance image quality from the LBFGS reconstruction, which, however, fails, when the representative is too distorted to be enhanced. Our approach provides recognizable images and works particularly well on the realistic setting of a trained ResNet as we can see in Fig. 3.6. Interestingly, the reconstructions on the trained ResNet have a better visual quality than those of the untrained ResNet, despite their lower PSNR values according to Table 3.2. Let us study the effect of trained network parameters in an even more realistic setting, i.e., for reconstructing ImageNet images

Table 3.2: PSNR mean and standard error for 100 experiments on the first images of the CIFAR-10 validation data set over two different networks with trained and untrained parameters.

Architecture	LeNet (Zhu)		ResNet20-4	
Trained	False	True	False	True
Eucl. Loss + L-BFGS	$\mathbf{33.11 \pm 2.33}$	7.84 ± 0.85	18.61 ± 0.67	2.44 ± 0.27
Proposed	23.27 ± 0.28	$\mathbf{21.67 \pm 0.45}$	$\mathbf{19.27 \pm 0.31}$	$\mathbf{14.57 \pm 0.43}$



(a) Class specific features appearing in reconstructions of trained networks.

(b) Positional invariance induced by a translational invariant network architecture.

Figure 3.7: Influences of training and architecture on image reconstruction: Reconstruction in a trained network (left); Reconstruction in a translational invariant network (right). Figure taken from [Pub4].

from a ResNet-152.

Trained vs. untrained networks. If a network is trained and has sufficient capacity for the gradient of the loss function $\mathcal{L}_\theta$ to be zero for different inputs, it is obvious that they can never be distinguished from their gradient. In practical settings, however, owing to stochastic gradient descent, data augmentation and a finite number of training epochs, the gradient of images is rarely entirely zero. While we do observe that image gradients have a much smaller magnitude in a trained network than in an untrained one, our magnitude-oblivious approach of (3.31) still recovers important visual information based only on the direction of the trained gradients.

We observe two general effects on trained networks that we illustrate with our ImageNet reconstructions in Fig. 3.8: First, reconstructions seem to become *implicitly biased* to typical features of the same class in the training data, e.g., the more blueish feathers of the capercaillie in the 5th image, or the large eyes of the owl in Fig. 3.7a. Thus, although the overall privacy of most images is clearly breached, this effect at least obstructs the recovery of fine scale details or the image's background. Second, we find that the data augmentation used during the training of neural networks leads to trained networks that make the *localization* of objects more difficult: Notice how few of the objects in Fig. 3.8 retain their original position and how the snake and gecko duplicate. Thus, although image reconstruction with networks trained with data augmentation still succeeds, some location information is lost.

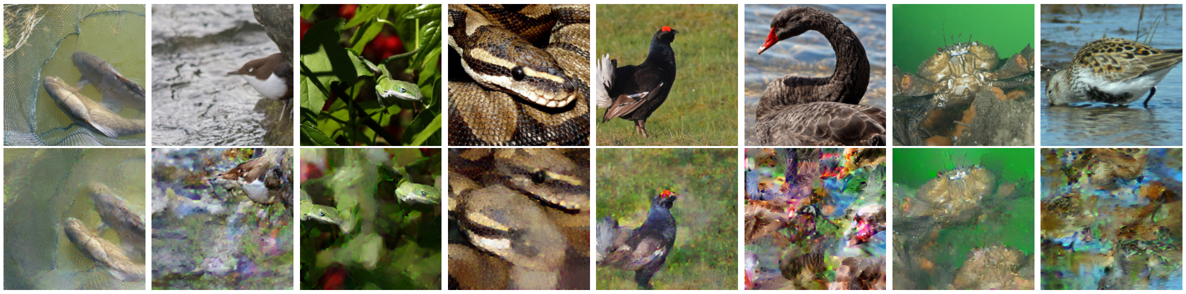


Figure 3.8: Single-Image Reconstruction from the parameter gradients of trained ResNet-152. Top row: Ground Truth. Bottom row: Reconstruction. We check every 1000th image of the ILSVRC2012 validation set. The amount of information leaked per image is highly dependent on image content - while some examples like the two tenches are highly compromised, the black swan (ironically) leaks almost no usable information. Noticeable is also the loss of positional information in several images. Figure taken from [Pub4].

Translational invariant convolutions. Let us study the ability to obscure the location of objects in more detail by testing how a conventional convolutional neural network, that uses convolutions with zero-padding, compares to a provably translationally invariant CNN, that uses convolutions with circular padding. As shown in Fig. 3.7b, while the conventional CNN allows for recovery of a rather high quality image (left), the translationally invariant network makes the localization of objects impossible (right) as the original object is separated. As such we identify the common zero-padding as a source of privacy risk.

Looking at the reconstruction results we obtain from ResNets with different depths, the proposed attack degrades very little with an increased depth of the network. In particular, as illustrated in Fig. 3.8, even faithful ImageNet reconstructions through a ResNet-152 are possible.

3.3.6 Distributed Learning with Federated Averaging and Multiple Images

So far we have only considered recovery of a single image from its gradient and discussed limitations and possibilities in this setting. We now turn to strictly more difficult generalized setting of *federated averaging* [82, 83, 106] and *multi-image* reconstruction, to show that the proposed improvements translate to this more practical case as well, discussing possibilities and limits in this application.

Instead of only calculating the gradient of a network's parameters based on local data, federated averaging performs multiple update steps on local data before sending the updated parameters back to the server. Following the notation of [82], we let the local data on the user's side consist of n images. For a number E of local epochs the user performs $\frac{n}{B}$ stochastic gradient update steps per epoch, where B denotes the local mini-batch size, resulting in a total number of $E\frac{n}{B}$ local update steps. Each user i then sends the locally updated parameters $\tilde{\theta}_i^{k+1}$ back to the server, which in turn updates the global parameters θ^{k+1} by averaging over all users.

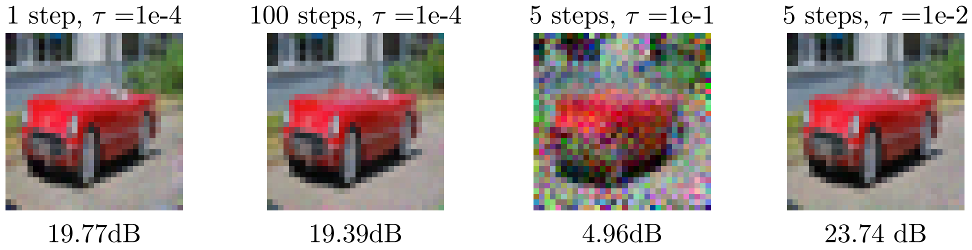


Figure 3.9: Illustrating the influence of the number of local update steps and the learning rate on the reconstruction: The left two images compare the influence of the number of gradient descent steps for a fixed learning rate of τ =1e-4. The two images on the right result from varying the learning rate for a fixed number of 5 gradient descent steps. PSNR values are shown below the images. Figure taken from [Pub4].

We empirically show that even the setting of federated averaging with $n \geq 1$ images is potentially amenable for attacks. To do so we try to reconstruct the local batch of n images by the knowledge of the local update $\tilde{\theta}_i^{k+1} - \theta^k$. In the following we evaluate the quality of the reconstructed images for different choices of n, E and B. We note that the setting studied in the previous sections corresponds to $n = 1$, $E = 1$, $B = 1$. For all our experiments we use an untrained ConvNet.

Multiple gradient descent steps, $B = n = 1$, $E > 1$:
Figure 3.9 shows the reconstruction of $n = 1$ image for a varying number of local epochs E and different choices of learning rate τ. Even for a high number of 100 local gradient descent steps the reconstruction quality is unimpeded. The only failure case we were able to exemplify was induced by picking a high learning rate of 1e-1. This setup, however, corresponds to a step size that would lead to a divergent training update, and as such does not provide useful model updates.

Multi-Image recovery, $B = n > 1$, $E = 1$:
So far we have considered the recovery of a single image only, and it seems reasonable to believe that averaging the gradients of multiple (local) images before sending an update to the server, restores the privacy of federated learning. While such a multi-image recovery has been considered in [163] for $B \leq 8$, we demonstrate that the proposed approach is capable of restoring some information from a batch of 100 averaged gradients: While most recovered images are unrecognizable (as shown in Fig. 3.22), Fig. 3.10

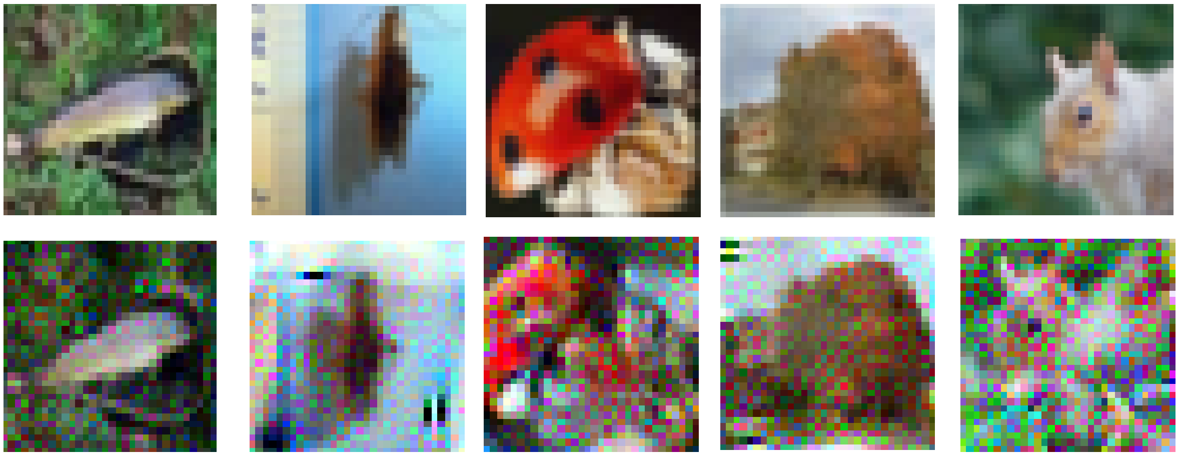


Figure 3.10: Information leakage from the aggregated gradient of a batch of 100 images on CIFAR-100 for a ResNet32-10. Shown are the 5 *most* recognizable images from the whole batch. Although most images are unrecognizable, privacy is broken even in a large-batch setting. Figure taken from [Pub4].

Table 3.3: PSNR statistics for various federated averaging settings, averaged over experiments on the first 100 images of the CIFAR-10 validation data set.

1 epoch			5 epochs	
4 images	8 images		1 image	8 images
batchsize 2	batchsize 2	batchsize 8	batchsize 1	batchsize 8
16.92 ± 2.10	**14.66 ± 1.12**	**16.49 ± 1.02**	**25.05 ± 3.28**	**16.58 ± 0.96**

shows the 5 most recognizable images and illustrates that even averaging the gradient of 100 images does not entirely secure the private data. Most surprising is that the distortions arising from batching are *non-uniform*. One could have expected all images to be equally distorted and near-irrecoverable, yet some images are highly distorted and others only to an extend at which the pictured object can still be recognized easily, which demonstrates that privacy leaks are conceivable even for large batches of image data.

Note that the attacker in this scenario only has knowledge about the average of gradients, however we assume the number of participating images to be known to the server. The server might request this information anyway (for example to balance heterogeneous data), but even if the exact number of images is unknown, the server (which we assume to have significantly more compute power than the user) could run reconstructions over a range of candidate numbers, given that the number of images is only a small integer value and then select the solution with minimal reconstruction loss.

General case

We also consider the general case of multiple local update steps using a subset of the whole local data in each mini batch gradient step. An overview of all conducted experiments is provided in Table 3.3. For each setting we perform 100 experiments on the CIFAR-10 validation set. For multiple images in a mini batch we only use images of different labels avoiding permutation ambiguities of reconstructed images of the same label. As to be expected, the single image reconstruction turns out to be most amenable to attacks in terms of PSNRs values. Despite a lower performance in terms of PSNR, we still observe privacy leakage for all multi-image reconstruction tasks, including those in which gradients in random mini-batches are taken. Comparing the full-batch, 8 images examples for 1 and 5 epochs, we see that our previous observation that multiple epochs do not make the reconstruction problem more difficult, extends to multiple images. We provide a qualitative assessment of reconstructed images of all experimental settings of Table 3.3 in Section 3.3.10.4

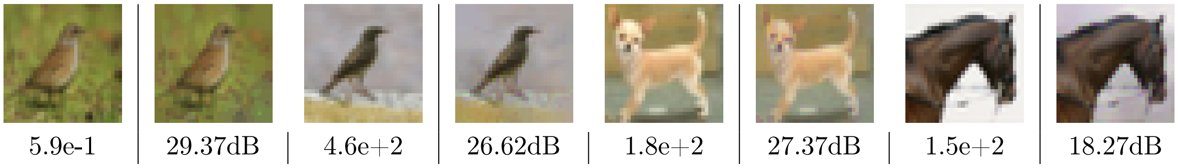


Figure 3.11: Label flipping. Images can be easily reconstructed when two rows in the parameters of the final classification layer are permuted. Below each input image is given the gradient magnitude, below each output image its PSNR. Compare these results to the additional examples in Fig. 3.13. Figure taken from [Pub4].

3.3.7 Variations of the Threat Model

In this work we consider a *honest-but-curious* threat model as discussed in the introduction. Straying from this scenario could be done primarily in two ways: First by changing the architecture, and second by keeping the architecture non-malicious, but changing the global parameters sent to the user.

3.3.7.1 Dishonest Architectures

So far we assumed that the server operates under an *honest-but-curious* model, and as such would not modify the model maliciously to make reconstruction easier. If we instead allow for this, then reconstruction becomes nearly trivial: Several mechanisms could be used: Following Proposition 3.3.1, the server could, for example, place a fully-connected layer in the first layer, or even directly connect the input to the end of the network by concatenation. Slightly less obvious, the model could be modified to contain reversible blocks [28, 55]. These blocks allow the recovery of input from their outputs. From Proposition 3.3.1 we know that we can reconstruct the input to the classification layer, so this allows for immediate access to the input image. If the server maliciously introduces separate weights or sub-models for each batch example, then this also allows for a recovery of an arbitrarily large batch of data. Operating in a setting, where such behavior is possible would require the user (or a provider trusted by the user) to vet any incoming model either manually or programmatically.

3.3.7.2 Dishonest Parameter Vectors

However, even with a fixed *honest* architecture, a malicious choice of global parameters can significantly influence reconstruction quality. For example, considering the network architecture in [163] which does not contain strides and flattens convolutional features, the dishonest server could set all convolution layers to represent the identity [47], moving the input through the network unchanged up to the classification layer, from which the input can be analytically computed as in Proposition 3.3.1. Likewise for an architecture that contains strides to a recognizable lower resolution [145], the input can be recovered immediately albeit in a smaller resolution when the right parameter vector is sent to the user.

Such a specific choice of parameters is however likely detectable. A subtler approach, as least possible in theory, would be to optimize the network parameters themselves that are sent to the user so that reconstruction quality from these parameters is maximized. While such an attack is likely to be difficult to detect on the user-side, it would also be very computationally intensive.

Label flipping. There is even a cheaper alternative. According to Section 3.3.5, very small gradient vectors may contain less information. A simple way for a dishonest server to boost these gradients is to permute two rows in the weight matrix and bias of the classification layer, effectively flipping the semantic meaning of a label. This attack is difficult to detect for the user (as long as the gradient magnitude stays within usual bounds), but effectively tricks him into differentiating his network w.r.t to the wrong label. Fig. 3.11 shows that this mechanism can allow for a reliable reconstruction with boosted PSNR scores, as the effect of the trained model is negated.

3.3.8 Experimental Details

3.3.8.1 Federated Averaging

The extension of Eq. (3.31) to the case of federated averaging (in which multiple local update steps are taken and sent back to the server) is straightforward. Notice first, that given old parameters θ^k, local

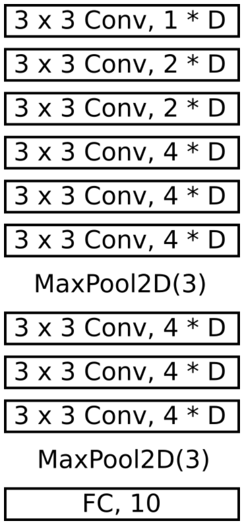


Figure 3.12: Network architecture *ConvNet*, consisting of 8 convolution layers, specified with corresponding number of output channels. Each convolution layer is followed by a batch normalization layer and a ReLU layer. D scales the number of output channels and is set to $D = 64$ by default. Figure taken from [Pub4].

Table 3.4: Ablation Study for the proposed approach for a trained ResNet-18 architecture, trained on CIFAR-10. Reconstruction PSNR scores are averaged over the first 10 images of the CIFAR-10 validation set (Standard Error in parentheses).

Basic Setup	20.12 dB (±1.02)
L2 Loss instead of cosine similarity	15.13 dB (±0.70)
Without total variation	19.96 dB (±0.75)
With L-BFGS instead of Adam	5.13 dB (±0.50)

updates θ^{k+l}, learning rate τ, and knowledge about the number of update steps[2], the update can be rewritten as the average of updated gradients.

$$\theta^{k+l} = \theta^k - \tau \sum_{m=1}^{l} \nabla_{\theta^{k+m}} \mathcal{L}_{\theta^{k+m}}(x, y) \tag{3.32}$$

Subtracting θ^k from θ^{k+l}, we simply apply the proposed approach to the resulting average of updates:

$$\arg \min_{x \in [0,1]^n} 1 - \frac{\langle \sum_{m=1}^{l} \nabla_{\theta^{k+m}} \mathcal{L}_{\theta^{k+m}}(x, y), \sum_{m=1}^{l} \nabla_{\theta^{k+m}} \mathcal{L}_{\theta^{k+m}}(\overline{x}, y) \rangle}{|| \sum_{m=1}^{l} \nabla_{\theta^{k+m}} \mathcal{L}_{\theta^{k+m}}(x, y) |||| \sum_{m=1}^{l} \nabla_{\theta^{k+m}} \mathcal{L}_{\theta^{k+m}}(\overline{x}, y) ||} + \alpha \, \mathrm{TV}(x). \tag{3.33}$$

Using automatic differentiation, we backpropagate the gradient w.r.t to x from the average of update steps.

3.3.8.2 ConvNet

We use a ConvNet architecture as a baseline for our experiments as it is relatively fast to optimize, reaches above 90% accuracy on CIFAR-10 and includes two max-pooling layers. It is a rough analogue to AlexNet [65]. The architecture is described in Fig. 3.12.

[2] We assume that the number of local updates is known to the server, yet this could also be found by brute-force, given that l is a small integer.

Architecture	LeNet (Zhu)		ResNet20-4	
Trained	False	True	False	True
TV	10^{-2}	10^{-3}	0	10^{-2}

Table 3.5: TV regularization values used for the proposed approach in the baseline experiments of Section 5.

3.3.8.3 Ablation Study

We provide an ablation for proposed choices in Table 3.4. We note that two things are central, the Adam optimizer and the similarity loss. Total variation is a small benefit, and using signed gradients is a minor benefit.

3.3.9 Hyperparameter Settings

In our experiments we reconstruct the network's input using Adam based on signed gradients as optimization algorithm and cosine similarity as cost function as described in Section 3.3.4. It is important to note that the optimal hyperparameters for the attack depend on the specific attack scenario - that the attack fails with default parameters is no guarantee for security. We always initialize our reconstructions from a Gaussian distribution with mean 0 and variance 1 (Note that the input data is normalized as usual for all considered datasets) and set the step size of the optimization algorithm within $[0.01, 1]$. We use a smaller step sizes of 0.1, for the wider and deeper networks in Section 3.3.5 and a larger step sizes of 1 for the federated averaging experiments in Section 3.3.6, with 0.1 being the default choice. The optimization runs for up to 24000 iterations. The step size decay is always fixed, occurring after $\frac{3}{8}$, $\frac{5}{8}$ and $\frac{7}{8}$ of iterations and reducing the learning rate by a factor of 0.1 each time. The number of iterations is a generally conservative estimate, privacy can often be broken much earlier.

We tweak the total variation parameter depending on the specific attack scenario, however note that its effect on avg. PSNR is mostly minor as seen in Table 3.4. When not otherwise noted we default to a value of 0.01.

Remark 3.3.5 (Restarts). Generally, multiple restarts of the attack from different random initializations can improve the attack success moderately. However they also increase the computational requirements significantly. To allow for quantitative experimental evaluations of multiple images, we do not consider restarts in this work (aside from Section 3.3.5 where we apply them to improve results of the competing LBFGS solver) - but stress that an attacker with enough resources could further improve his attack by running it with multiple restarts.

3.3.9.1 Comparison to Previous Approaches

For comparison with baselines in Section 3.3.5, we re-implement the network from [163], which we dub LeNet (Zhu) in the following, and additionally run all experiments for the ResNet20-4 architecture. We base both the network and the approach on code from the authors of [163], [3]. For the LBFGS-L2 optimization we use a learning rate of $1e-4$ and 300 iterations. For the ResNet experiments we use the generous amount of 8 restarts and for the faster to optimize LeNet (Zhu) architecture we use the even higher number of 16 restarts. All experiment conducted with the proposed approach only use one restart, 4800 iterations, a learning rate of 0.1 and TV regularization parameters as detailed in Table 3.5. Note that in the described settings the proposed method took significantly less time to optimize than the LBFGS optimization.

3.3.9.2 Spatial Information

The experiments on spatial information are performed on the ConvNet architecture with $D = 64$ channels.

[3] https://github.com/mit-han-lab/dlg

Number of epochs E	1	1	1	5	5
Number of local images n	4	8	8	1	8
Mini-batch size B	2	2	8	1	8
TV	10^{-6}	10^{-6}	10^{-4}	10^{-4}	10^{-4}

Table 3.6: Total variation weights for the reconstruction of network input in the experiments in Sec. 4.2

3.3.9.3 Setting for Experiments in Section 3.3.6

For the five cases consider in Table 2 we consider an untrained ConvNet, a learning rate of 1, 4800 iterations, one restart and the TV regularization parameters as given in Table 3.6. Each of the 100 experiments uses different images, i.e. each experiments uses the images of the CIFAR-10 validation set following the ones used in the previous experiment. As multiple images of the same label in one mini-batch cause an ambiguity in the ordering of images w.r.t. that label, we do not consider that case. If an image with an already encountered label is about to be added to the respective mini-batch we skip that image and use the next image of the validation set with a different label.

3.3.10 Additional Examples

3.3.10.1 CIFAR-10 Examples

Following the empirical study from Section 3.3.5 showing a slightly less visual quality of reconstructions obtained from trained networks in comparison to untrained networks, we conjecture that the gradient magnitude of images influences the reconstruction quality. Figure 3.13 shows additional “extreme” examples for CIFAR-10, reconstructing the image with lowest and the image with largest gradient magnitude for the training and validation set of CIFAR-10 for trained and untrained ConvNet and ResNet20-4 models. Indeed we can see from the examples of the trained ConvNet that extremely small magnitudes seem to negatively effect the results. However, we also observe that some images, even in the training data set, boast high gradient magnitudes and tend to be more susceptible to reconstruction of semantic image features. The presence of high gradient magnitudes in the training data images might be partly attributed to the stochastic optimization procedure which never evaluates the loss on the whole data set in the course of the training.

3.3.10.2 ImageNet Examples

Figure 3.14 shows further instructive examples of reconstructions for ImageNet validation images for a trained ResNet-18 (the same setup as Fig. 3.8) from Section 3.3.5. We show a very good reconstruction (German shepherd), a good, but translated reconstruction (giant panda) and two failure cases (ambulance and flower). For the ambulance, for example, the actual writing on the ambulance car is still hidden. For the flower, the exact number of petals is hidden. Also, note how the reconstruction of the giant panda is much clearer than that of the tree stump co-occurring in the image, which we consider an indicator of the self-regularizing effect described in Section 3.3.5.

Figure 3.15 and Fig. 3.16 show more examples. We note that the examples in these figures and in Fig. 3.8 are not handpicked, but chosen neutrally according to their ID in the ILSVRC2012, ImageNet, validation set. The ID for each image is obtained by sorting the synset that make up the dataset in increasing order according to their synset ID and sorting the images within each synset according to their synset ID in increasing order. This is the default order in `torchvision`.

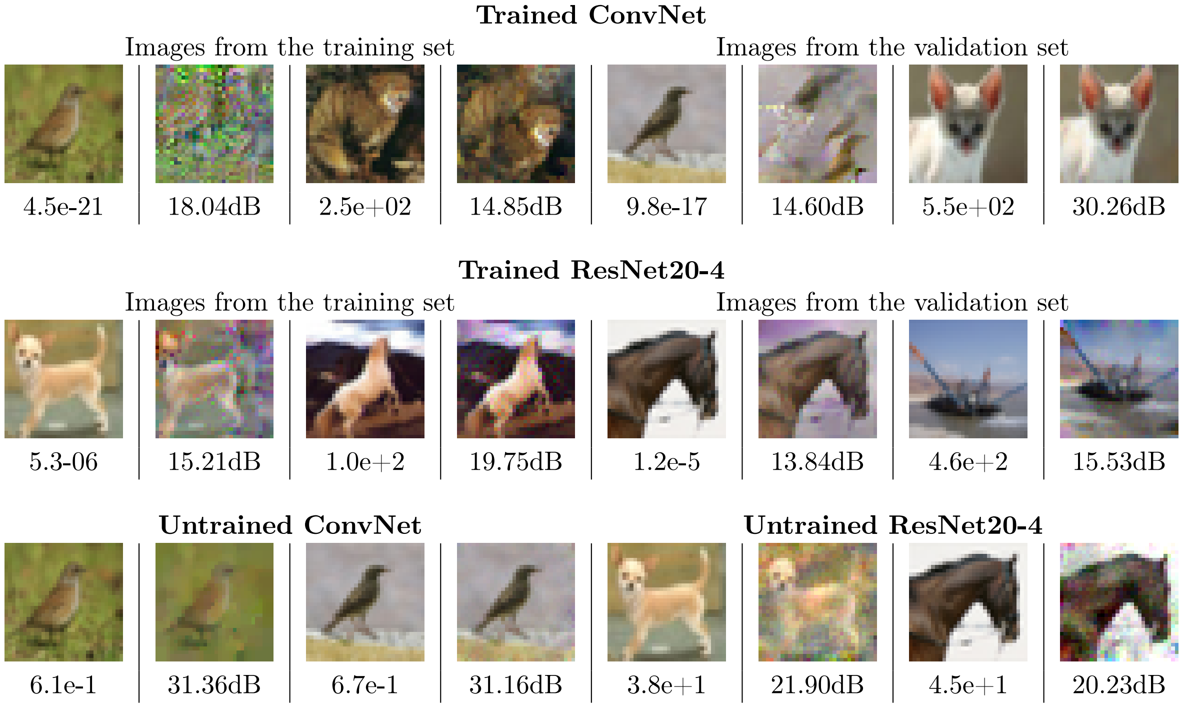


Figure 3.13: Reconstruction of images for the *trained* ConvNet model (Top) and ResNet20-4 (middle). We show reconstructions of the **worst-case** image and **best case** image from CIFAR-10, based on gradient magnitude for both the training and the validation set. Below each input image is given the gradient magnitude, below each output image its PSNR. The bottom row shows reconstructions for the worst-case examples for untrained models. Figure taken from [Pub4].

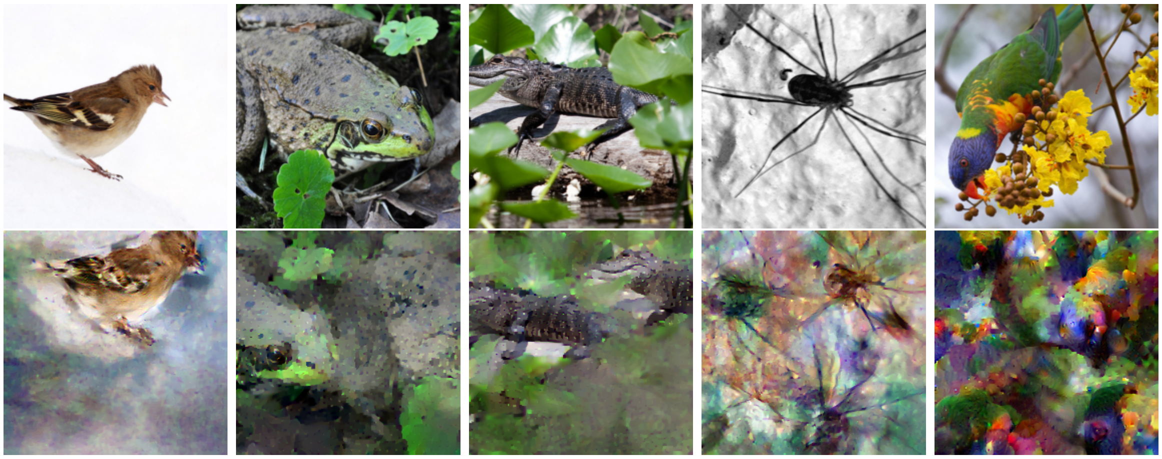


Figure 3.16: Additional single-image reconstruction from the parameter gradients of trained ResNet-152. Top row: Ground Truth. Bottom row: Reconstruction. These are images 500, 1500, 2500, 3500, 4500. Figure taken from [Pub4].

3.3.10.3 Multi-Image Recovery

For multi-image recovery of Section 3.3.6, we show the full set of 100 images in Fig. 3.22, we recommend to zoom in to a digital version of the figure. While most of the images are hardly or not recognizable we point out that the resulting visual quality of the reconstructed images also greatly depends on the initialization of the optimization. We repeated the experiment in several runs and found that in different runs some images are more recognizable than in the other runs. This, however, again shows that failure in optimization in one run for certain images does not guarantee that privacy can be preserved in other

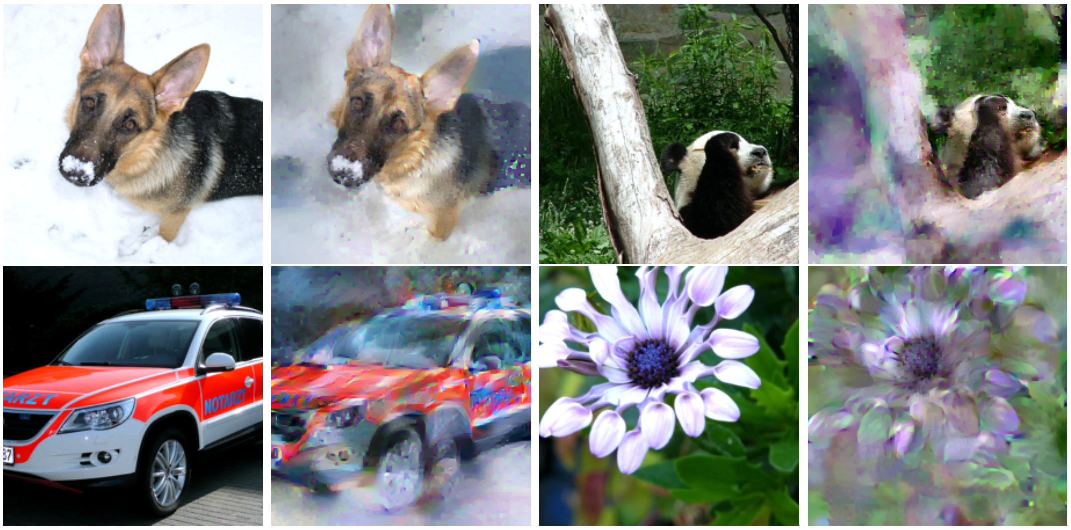


Figure 3.14: Additional qualitative ImageNet examples, failure cases and positive cases for a trained ResNet-18. Images taken from the ILSVRC2012 validation set. Figure taken from [Pub4].

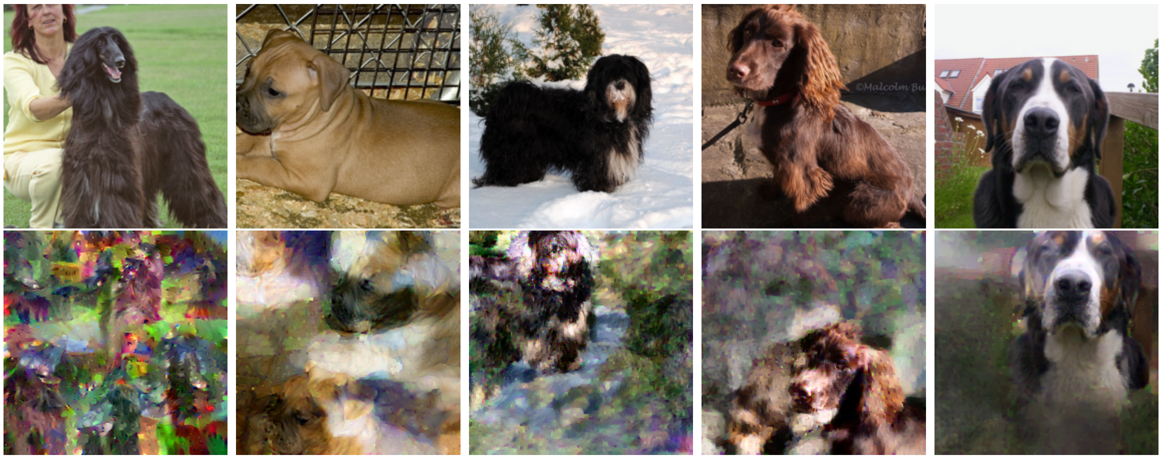


Figure 3.15: Additional single-image reconstruction from the parameter gradients of trained ResNet-152. Top row: Ground Truth. Bottom row: Reconstruction. Figure 3.8 showed images 0000, 1000, 2000, 3000, 4000, 5000, 6000, 7000 from the ILSVRC2012 validation set. These are images 8000-12000. Figure taken from [Pub4].

configurations. Also as future improvements in attacks of course pose critical source of privacy breech.

3.3.10.4 General Case

In the following we provide the reconstructed images from the experiments on the general case from Section 3.3.6. In contrast to the reported PSNR values, the reconstructed images themself allow for a better evaluation of how much visual information is preserved in the different settings of the federated averaging case.

In Fig. 3.17 we first show the results of reconstructing only one local image with 5 epochs ($n = 1$, $E = 5$) for each of the first 100 images of the CIFAR-10 validation data set. Basically all of the reconstructed images convey a considerable amount of semantic information about the original setting rendering this setting especially amenable to our attack.

Figure 3.18 shows the results of the scenario of $n = 4$ local images for the first 10 batches of 4 images in the CIFAR-10 validation data set. While most of the images are still recognizable the overall visual quality of the reconstruction suffers from the increased amount of local images. Note that in some batches images are dropped in order to avoid having images with the same semantic label in the same batch thus avoiding ambiguity in the reconstruction.

In Fig. 3.19 we again increase the number of local images to $n = 8$. Shown are the results for the first 10 batches of 8 images. Images with high contrast as for instance the black plane in front of the bright sky tend to still be easily discernible. Many images are, however, hard to identify as a result of the decreased visual quality of the reconstruction due to the increased batch sizes. In Fig. 3.20 we change the mini-batch size from $B = 2$ to $B = 8$. The visual quality of the reconstruction now seems to be slightly worse, the leaked semantic information, however, remains roughly the same. Increasing the number of local epochs from $E = 1$ to $E = 5$ in Fig. 3.21 interestingly improves the overall quality of the reconstructions.

Overall the best mechanism for making attacks harder in the federated averaging scenario turns out to be increasing the number of local images n. Changing the mini-batch size B hardly seems to affect the results. Increasing the number of local epochs E, interestingly, even seems to be beneficial for the obtained reconstructions.

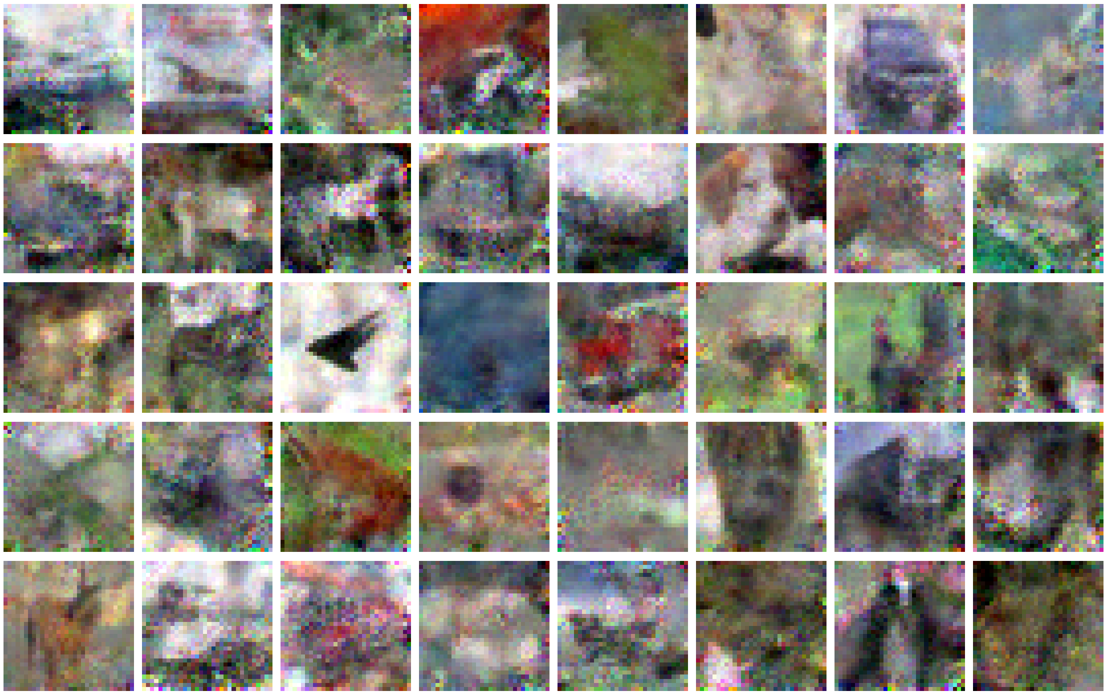


Figure 3.18: Results of the first ten experiments for $E = 1$, $n = 4$, $B = 2$. Figure taken from [Pub4].

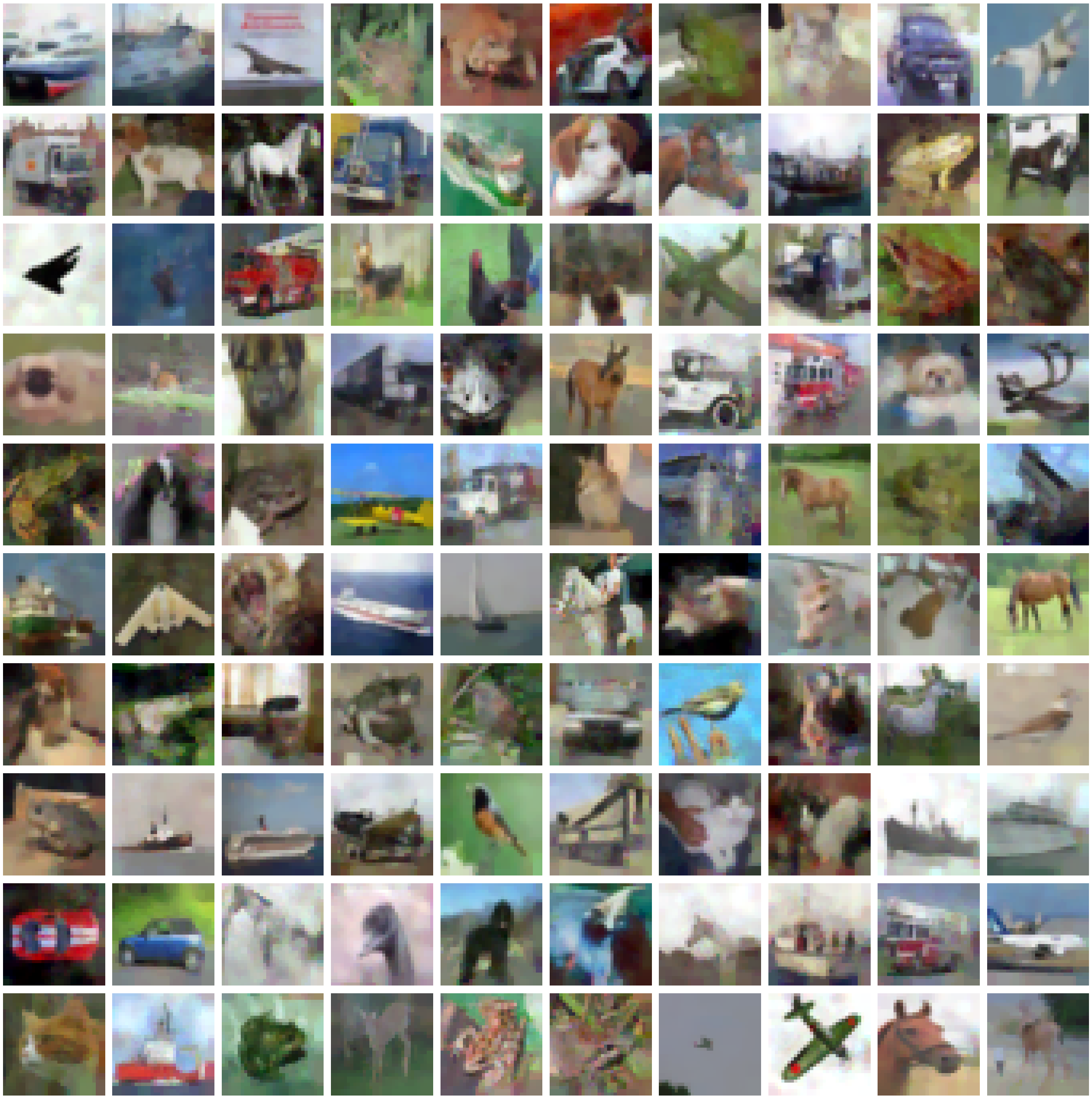


Figure 3.17: Results of the first 100 experiments for $E = 5$, $n = 1$, $B = 1$. Figure taken from [Pub4].

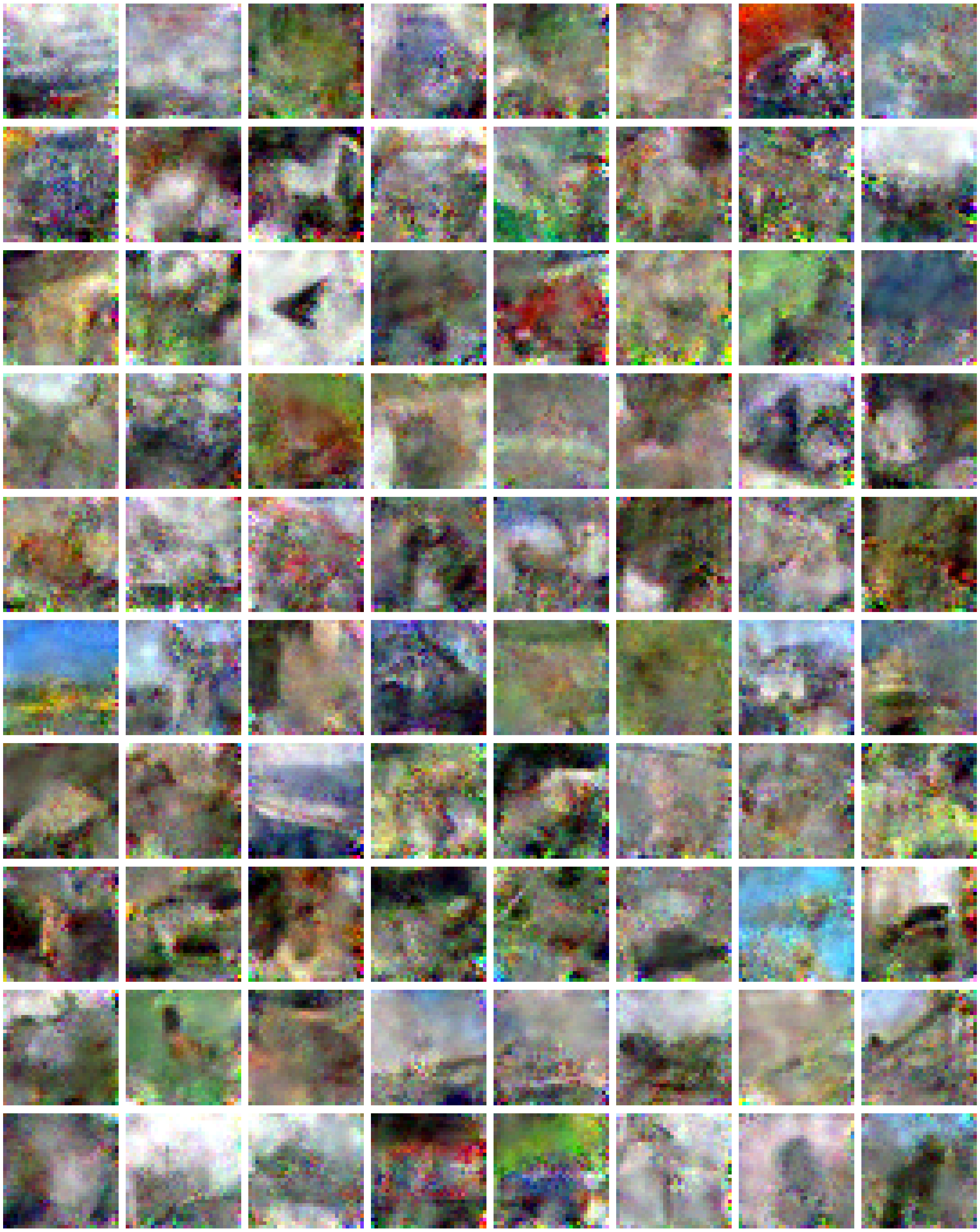


Figure 3.19: Results of the first ten experiments for $E = 1$, $n = 8$, $B = 2$. Figure taken from [Pub4].

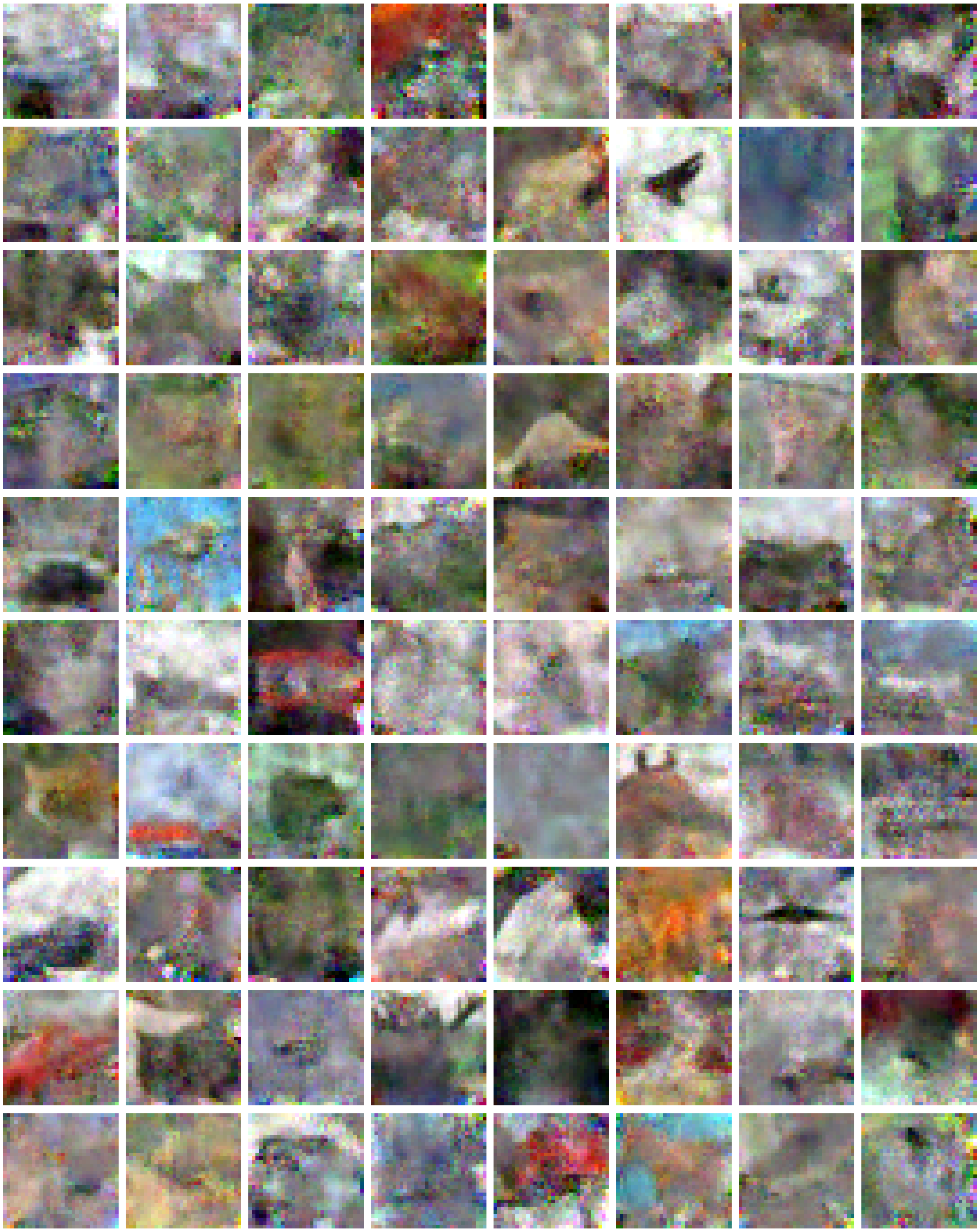


Figure 3.20: Results of the first ten experiments for $E = 1$, $n = 8$, $B = 8$. Figure taken from [Pub4].

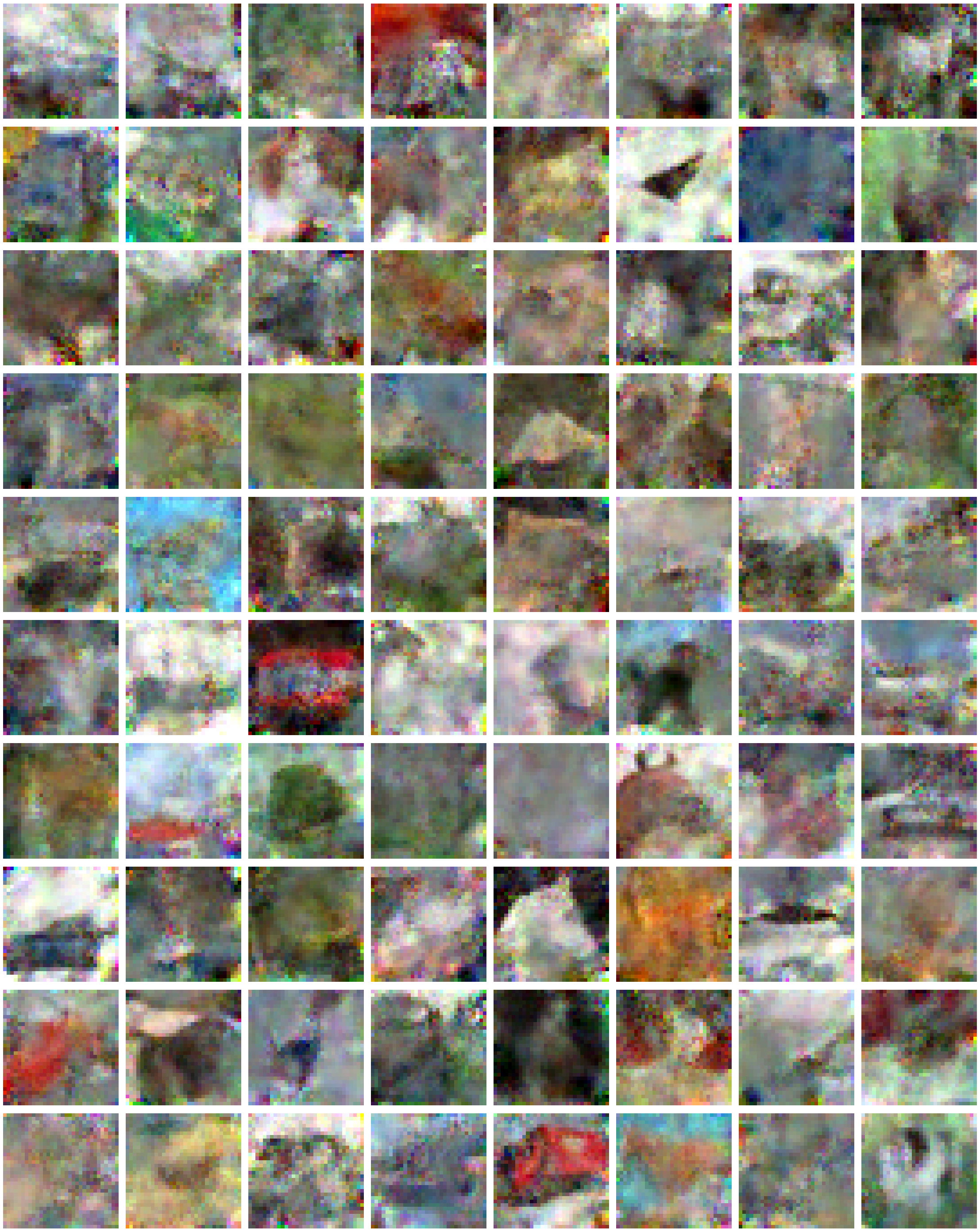


Figure 3.21: Results of the first ten experiments for $E = 5$, $n = 8$, $B = 8$. Figure taken from [Pub4].

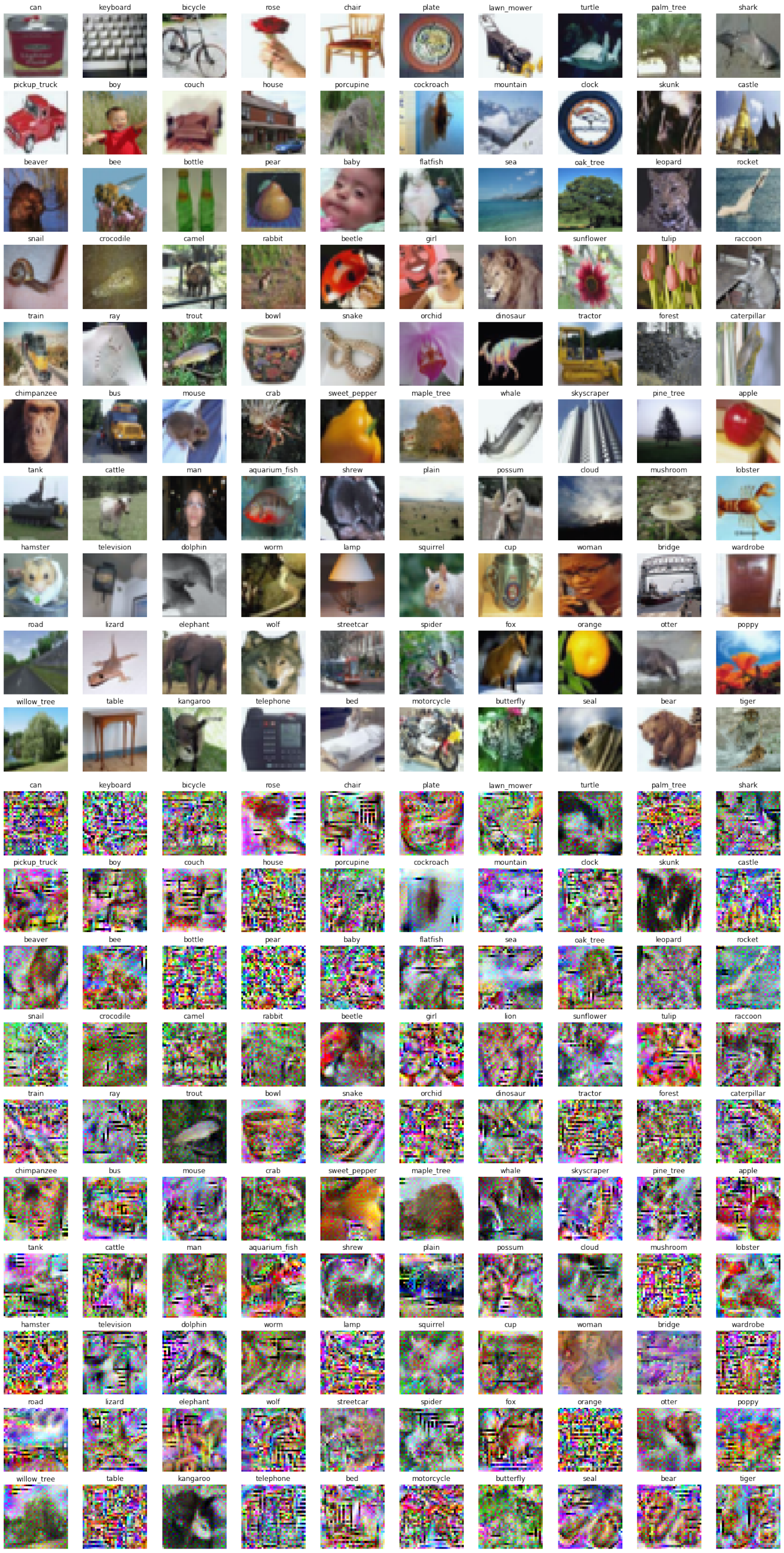


Figure 3.22: Full results for the batch of CIFAR-100 images. Same experiment as in Fig. 3.10. The top 10 rows show the input images and the bottom 10 rows the reconstructed images in the same order. Figure taken from [Pub4].

Chapter 4

Conclusion

This thesis encompasses a wide range of strategies for tackling optimization problems of various nature resulting in optimization strategies with different characteristics depending on the problem. Especially for model based approaches robust methods can be developed. We have shown how ideas from *functional lifting* are especially suitable for a wide range of nonconvex variational problems in computer vision making the problems amenable for tools from convex optimization. This in turn makes the proposed methods more stable and less dependent on initialization. Developing more advanced methods of convex relaxations applicable beyond the scope of MRF inference problems and to more complex regularizes remains an open question for future approaches.

Analyzing the concepts of functional lifting reveals that the proposed approaches are always a trade off between runtime efficiency and tighter guarantees. We have furthermore seen that this tradeoff persists also outside of the functional lifting setting in a wide variety of problems. Especially relevant in the light of the extensive emergence of machine learning models are guarantees for data driven applications particularly in security critical domains. We have shown how guarantees can be enforced for dedicated learnable models in the setting of linear inverse problems. However, such provable results are an exception and remain a major challenge for upcoming developments in the field of machine learning. As a negative result we highlighted the potential security issues in the context of federated learning where even large and hence seemingly safe network architectures fail to obfuscate image information contained in the network's gradients. Imposing robustness guarantees upon data driven approaches will be one of the paramount challenges in future research.

List of Publications

[Pub1] H. Bauermeister, M. Burger, and M. Moeller, *Learning Spectral Regularizations for Linear Inverse Problems*, in Workshop on Deep Learning and Inverse Problems, Conference on Neural Information Processing Systems (NeurIPS), 2020.

[Pub2] H. Bauermeister*, E. Laude*, T. Möllenhoff, M. Moeller, and D. Cremers, *Lifting the Convex Conjugate in Lagrangian Relaxations: A Tractable Approach for Continuous Markov Random Fields*, SIAM Journal on Imaging Sciences (SIIMS), 15 (2022), pp. 1253–1281.

[Pub3] H. Bauermeister, P. Ochs, T. Meinhardt, L. Leal-Taixe, and M. Moeller, *Adaptive Network Architectures via Linear Splines*, in Workshop on Interactive and Adaptive Learning in an Open World, European Conference on Computer Vision (ECCV), 2018.

[Pub4] J. Geiping*, H. Bauermeister*, H. Dröge*, and M. Moeller, *Inverting Gradients - How easy is it to break privacy in federated learning?*, in Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2020.

[Pub5] J. Geiping, F. Gaede, H. Bauermeister, and M. Moeller, *Fast Convex Relaxations using Graph Discretizations*, in Proceedings of the British Machine Vision Conference (BMVC), 2020.

[Pub6] T. M. Wong, H. Bauermeister, M. Kahl, P. H. Bolívar, M. Möller, and A. Kolb, *Deep Optimization Prior for THz Model Parameter Estimation*, in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2022.

*Equal contribution

Bibliography

[1] R. Achanta, A. S. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, *SLIC Superpixels Compared to State-of-the-Art Superpixel Methods*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 34 (2012), pp. 2274–2282.

[2] J. Adler and O. Öktem, *Solving ill-posed inverse problems using iterative deep neural networks*, Inverse Problems, 33 (2017).

[3] L. Ambrosio, N. Fusco, and D. Pallara, *Functions of Bounded Variation and Free Discontinuity Problems*, Oxford university press, Oxford, 2000.

[4] A. Athalye, N. Carlini, and D. Wagner, *Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples*, in Proceedings of the International Conference on Machine Learning (ICML), 2018.

[5] H. Attouch, G. Buttazzo, and G. Michaille, *Variational Analysis in Sobolev and BV Spaces*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2014.

[6] F. Bach, *Submodular functions: From discrete to continuous domains*, Mathematical Programming, 175 (2019), pp. 419–459.

[7] E. J. Balder, *An extension of duality-stability relations to nonconvex optimization problems*, SIAM Journal on Optimization (SIOPT), 15 (1977), pp. 329–343.

[8] M. Benning and M. Burger, *Modern regularization methods for inverse problems*, Acta Numerica, 27 (2018), pp. 1–111.

[9] R. Bergmann, R. H. Chan, R. Hielscher, J. Persch, and G. Steidl, *Restoration of manifold-valued images by half-quadratic minimization*, Inverse Problems and Imaging, 10 (2016), pp. 281–304.

[10] F. Bernard, F. R. Schmidt, J. Thunberg, and D. Cremers, *A combinatorial solution to non-rigid 3D shape-to-image matching*, in Proceedings of the IEEE/CVF Conference on computer vision and Pattern Recognition (CVPR), 2017.

[11] D. Bertsekas, *Nonlinear Programming*, Athena scientific optimization and computation series, Athena Scientific, 2016.

[12] G. Blekherman, P. A. Parrilo, and R. R. Thomas, *Semidefinite Optimization and Convex Algebraic Geometry*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2012.

[13] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, B. McMahan, et al., *Towards federated learning at scale: System design*, Proceedings of Machine Learning and Systems, 1 (2019), pp. 374–388.

[14] A. Bora, A. Jalal, E. Price, and A. Dimakis, *Compressed sensing using generative models*, in Proceedings of the International Conference on Machine Learning (ICML), 2017.

[15] Y. Boykov and G. Funka-Lea, *Graph Cuts and Efficient N-D Image Segmentation*, International Journal of computer vision (IJCV), 70 (2006), pp. 109–131.

[16] Y. Boykov, O. Veksler, and R. Zabih, *Fast approximate energy minimization via graph cuts*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 23 (2001), pp. 1222–1239.

[17] X. Bresson and T. F. Chan, *Non-local Unsupervised Variational Image Segmentation Models*, UCLA CAM report, University of California, Los Angeles, 2008.

[18] B. Brown, D. Elliott, and D. Paget, *Lipschitz constants for the Bernstein polynomials of a Lipschitz continuous function*, Journal of approximation theory, 49 (1987), pp. 196–199.

[19] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, *High Accuracy Optical Flow Estimation Based on a Theory for Warping*, in Proceedings of the European Conference on computer vision (ECCV), 2004.

[20] H. T. Bui, R. S. Burachik, A. Y. Kruger, and D. T. Yost, *Zero duality gap conditions via abstract convexity*, Optimization, 71 (2022), pp. 811–847.

[21] M. Burger and S. Osher, *A Guide to the TV zoo*, in PDE Based Reconstruction Methods in Imaging, no. 2090 in Lecture Notes in Mathematics, Springer International Publishing, Switzerland, 2013.

[22] M. Burger, E. Resmerita, and L. He, *Error estimation for bregman iterations and inverse scale space methods in image restoration*, Computing, 81 (2007), pp. 109–135.

[23] L. Calatroni, C. Cao, J. C. De Los Reyes, C.-B. Schönlieb, and T. Valkonen, *Bilevel approaches for learning of variational imaging models*, Variational Methods: In Imaging and Geometric Control, 18 (2017), p. 2.

[24] E. J. Candes, J. Romberg, and T. Tao, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, IEEE Transactions on Information Theory, 52 (2006), pp. 489–509.

[25] C. Carathéodory, *Über den Variabilitätsbereich der Fourier'schen Konstanten von positiven harmonischen Funktionen*, Rendiconti Del Circolo Matematico di Palermo (1884-1940), 32 (1911), pp. 193–217.

[26] A. Chambolle, D. Cremers, and T. Pock, *A Convex Approach to Minimal Partitions*, SIAM Journal on Imaging Sciences (SIIMS), 5 (2012), pp. 1113–1158.

[27] A. Chambolle and T. Pock, *A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging*, Journal of Mathematical Imaging and Vision, 40 (2011), pp. 120–145.

[28] B. Chang, L. Meng, E. Haber, L. Ruthotto, D. Begert, and E. Holtham, *Reversible Architectures for Arbitrarily Deep Residual Neural Networks*, in Proceedings of the AAAI conference on artificial intelligence, 2018.

[29] G. Charpiat, N. Girard, L. Felardos, and Y. Tarabalka, *Input Similarity from the Neural Network Perspective*, in Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2019.

[30] Q. Chen and V. Koltun, *Full flow: Optical flow estimation by global optimization over regular grids*, in Proceedings of the IEEE/CVF Conference on computer vision and Pattern Recognition (CVPR), 2016.

[31] T. Chen, J.-B. Lasserre, V. Magron, and E. Pauwels, *Semialgebraic Optimization for Lipschitz Constants of ReLU Networks*, in Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2020.

[32] Y. Chen and T. Pock, *Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 39 (2016), pp. 1–1.

[33] Y. Chen, T. Pock, and H. Bischof, *Learning ℓ_1-based analysis and synthesis sparsity priors using bi-level optimization*, in Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2012.

[34] T. Chilimbi, Y. Suzue, J. Apacible, and K. Kalyanaraman, *Project Adam: Building an Efficient and Scalable Deep Learning Training System*, in 11th USENIX symposium on operating systems design and implementation (OSDI 14), 2014, pp. 571–582.

[35] D. J. Crandall, A. Owens, N. Snavely, and D. P. Huttenlocher, *SfM with MRFs: Discrete-continuous optimization for large-scale structure from motion*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 35 (2012), pp. 2841–2853.

[36] C. Domokos, F. R. Schmidt, and D. Cremers, *MRF optimization with separable convex prior on partially ordered labels*, in Proceedings of the European Conference on computer vision (ECCV), 2018.

[37] A. Dosovitskiy and T. Brox, *Generating Images with Perceptual Similarity Metrics based on Deep Networks*, in Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2016.

[38] A. Dosovitskiy and T. Brox, *Inverting Visual Representations With Convolutional Networks*, in Proceedings of the IEEE/CVF Conference on computer vision and Pattern Recognition (CVPR), 2016.

[39] J. Eckstein and D. P. Bertsekas, *On the Douglas—Rachford splitting method and the proximal point algorithm for maximal monotone operators*, Mathematical Programming, 55 (1992), pp. 293–318.

[40] H. Engl, M. Hanke, and G. Neubauer, *Regularization of Inverse Problems*, Mathematics and Its Applications, Springer Netherlands, 1996.

[41] A. Fix and S. Agarwal, *Duality and the Continuous Graphical Model*, in Proceedings of the European Conference on computer vision (ECCV), 2014.

[42] G. Folland, *Real Analysis: Modern Techniques and Their Applications*, Pure and Applied Mathematics: A Wiley Series of Texts, Monographs and Tracts, Wiley, 2013.

[43] M. Fredrikson, S. Jha, and T. Ristenpart, *Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures*, in Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, 2015.

[44] K. Ganju, Q. Wang, W. Yang, C. A. Gunter, and N. Borisov, *Property Inference Attacks on Fully Connected Neural Networks using Permutation Invariant Representations*, in Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2018.

[45] J. Geiping and M. Moeller, *Parametric Majorization for Data-Driven Energy Minimization Methods*, in Proceedings of the International Conference on computer vision (ICCV), 2019.

[46] G. Gilboa and S. Osher, *Nonlocal Operators with Applications to Image Processing*, Multiscale Modelling and Simulation, 7 (2008), pp. 1005–1028.

[47] M. Goldblum, J. Geiping, A. Schwarzschild, M. Moeller, and T. Goldstein, *Truth or Backpropaganda? An Empirical Investigation of Deep Learning Theory*, in Proceedings of the International Conference on Learning Representations (ICLR), 2020.

[48] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, *Maxout networks*, in Proceedings of the International Conference on Machine Learning (ICML), 2013.

[49] S. Guinard, L. Landrieu, L. Caraffa, and B. Vallet, *Piecewise-Planar Approximation of Large 3D Data as Graph-Structured Optimization*, in ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. IV-2-W5, Copernicus GmbH, May 2019, pp. 365–372.

[50] K. He, X. Zhang, S. Ren, and J. Sun, *Delving deep into rectifiers: Surpassing human-level performance on imagenet classification*, in Proceedings of the International Conference on computer vision (ICCV), 2015.

[51] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, in Proceedings of the IEEE/CVF Conference on computer vision and Pattern Recognition (CVPR), 2016.

[52] B. K. P. Horn and B. G. Schunck, *Determining optical flow*, Artificial Intelligence, 17 (1981), pp. 185–203.

[53] H. Ishikawa, *Exact optimization for Markov random fields with convex priors*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 25 (2003), pp. 1333–1336.

[54] H. Ishikawa and D. Geiger, *Occlusions, discontinuities, and epipolar lines in stereo*, in Proceedings of the European Conference on computer vision (ECCV), 1998.

[55] J.-H. Jacobsen, A. Smeulders, and E. Oyallon, *I-RevNet: Deep Invertible Networks*, in Proceedings of the International Conference on Learning Representations (ICLR), 2018.

[56] A. Jochems, T. M. Deist, I. El Naqa, M. Kessler, C. Mayo, J. Reeves, S. Jolly, M. Matuszak, R. Ten Haken, J. van Soest, C. Oberije, C. Faivre-Finn, G. Price, D. de Ruysscher, P. Lambin, and A. Dekker, *Developing and Validating a Survival Prediction Model for NSCLC Patients Through Distributed Learning Across 3 Countries*, International Journal of Radiation Oncology*Biology*Physics, 99 (2017), pp. 344–352.

[57] A. Jochems, T. M. Deist, J. van Soest, M. Eble, P. Bulens, P. Coucke, W. Dries, P. Lambin, and A. Dekker, *Distributed learning: Developing a predictive model based on data from multiple hospitals without data leaving the hospital – A real life proof of concept*, Radiotherapy and Oncology, 121 (2016), pp. 459–467.

[58] L. V. Kantorovich, *Mathematical methods of organizing and planning production*, Management Science, 6 (1960), pp. 366–422.

[59] J. Kappes, B. Andres, F. Hamprecht, C. Schnorr, S. Nowozin, D. Batra, S. Kim, B. Kausler, J. Lellmann, N. Komodakis, et al., *A comparative study of modern inference techniques for discrete energy minimization problems*, in Proceedings of the IEEE/CVF Conference on computer vision and Pattern Recognition (CVPR), 2013.

[60] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, in Proceedings of the International Conference on Learning Representations (ICLR), 2015.

[61] E. Kobler, T. Klatzer, K. Hammernik, and T. Pock, *Variational networks: connecting variational methods and deep learning*, in Proceedings of the German Conference on Pattern Recognition (GCPR), 2017.

[62] P. W. Koh and P. Liang, *Understanding Black-box Predictions via Influence Functions*, in Proceedings of the International Conference on Machine Learning (ICML), 2017.

[63] J. Konečný, B. McMahan, and D. Ramage, *Federated Optimization: Distributed Optimization Beyond the Datacenter*, CoRR, abs/1511.03575 (2015).

[64] J.-L. Krivine, *Anneaux préordonnés*, Journal d'analyse mathématique, 12 (1964), pp. 307–326.

[65] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Imagenet classification with deep convolutional neural networks*, in Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2012.

[66] L. Landrieu and G. Obozinski, *Cut Pursuit: Fast algorithms to learn piecewise constant functions*, in Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, vol. 51 of Proceedings of Machine Learning Research, Cadiz, Spain, Apr. 2016, PMLR, pp. 1384–1393.

[67] L. Landrieu and G. Obozinski, *Cut Pursuit: Fast Algorithms to Learn Piecewise Constant Functions on General Weighted Graphs*, SIAM Journal on Imaging Sciences (SIIMS), 10 (2017), pp. 1724–1766.

[68] L. Landrieu and M. Simonovsky, *Large-scale point cloud semantic segmentation with superpoint graphs*, in Proceedings of the IEEE/CVF Conference on computer vision and Pattern Recognition (CVPR), 2018.

[69] J. B. Lasserre, *Global optimization with polynomials and the problem of moments*, SIAM Journal on Optimization (SIOPT), 11 (2001), pp. 796–817.

[70] J. B. Lasserre, *Semidefinite programming vs. LP relaxations for polynomial programming*, Mathematics of Operations Research, 27 (2002), pp. 347–360.

[71] F. Latorre, A. Eftekhari, and V. Cevher, *Fast and provable admm for learning with generative priors*, in Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2019.

[72] F. Latorre, P. T. Y. Rolland, and V. Cevher, *Lipschitz constant estimation for Neural Networks via sparse polynomial optimization*, in Proceedings of the International Conference on Learning Representations (ICLR), 2020.

[73] E. Laude, T. Möllenhoff, M. Moeller, J. Lellmann, and D. Cremers, *Sublabel-Accurate Convex Relaxation of Vectorial Multilabel Energies*, in Proceedings of the European Conference on computer vision (ECCV), 2016.

[74] J. Lellmann, J. Kappes, J. Yuan, F. Becker, and C. Schnörr, *Convex Multi-class Image Labeling by Simplex-Constrained Total Variation*, in Scale Space and Variational Methods in computer vision, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2009, pp. 150–162.

[75] J. Lellmann, E. Strekalovskiy, S. Koetter, and D. Cremers, *Total Variation Regularization for Functions with Values in a Manifold*, in Proceedings of the International Conference on computer vision (ICCV), 2013.

[76] C. Lemaréchal, *Lagrangian relaxation*, in Computational Combinatorial Optimization, Springer, 2001, pp. 112–156.

[77] Y. Li, S. Liu, J. Yang, and M.-H. Yang, *Generative face completion*, in Proceedings of the IEEE/CVF Conference on computer vision and Pattern Recognition (CVPR), 2017.

[78] D. C. Liu and J. Nocedal, *On the limited memory BFGS method for large scale optimization*, Mathematical Programming, 45 (1989), pp. 503–528.

[79] G. Lorentz, *Bernstein Polynomials*, AMS Chelsea Publishing Series, Chelsea Publishing Company, 1986.

[80] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, *Towards Deep Learning Models Resistant to Adversarial Attacks*, in Proceedings of the International Conference on Learning Representations (ICLR), 2018.

[81] A. Mahendran and A. Vedaldi, *Visualizing Deep Convolutional Neural Networks Using Natural Pre-images*, International Journal of computer vision (IJCV), 120 (2016), pp. 233–255.

[82] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, *Communication-Efficient Learning of Deep Networks from Decentralized Data*, in Proceedings of the Artificial Intelligence and Statistics (AISTATS), PMLR, 2017, pp. 1273–1282.

[83] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, *Learning Differentially Private Recurrent Language Models*, in Proceedings of the International Conference on Learning Representations (ICLR), 2018.

[84] T. Meinhardt, M. Moeller, C. Hazirbas, and D. Cremers, *Learning Proximal Operators: Using Denoising Networks for Regularizing Inverse Imaging Problems*, in Proceedings of the International Conference on computer vision (ICCV), 2017.

[85] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, *Exploiting Unintended Feature Leakage in Collaborative Learning*, in IEEE Symposium on Security and Privacy (SP), May 2019, pp. 691–706.

[86] M. Moeller, T. Moellenhoff, and D. Cremers, *Controlling Neural Networks via Energy Dissipation*, in Proceedings of the International Conference on computer vision (ICCV), 2019.

[87] T. Möllenhoff and D. Cremers, *Sublabel-Accurate Discretization of Nonconvex Free-Discontinuity Problems*, in Proceedings of the International Conference on computer vision (ICCV), 2017.

[88] T. Möllenhoff and D. Cremers, *Lifting Vectorial Variational Problems: A Natural Formulation based on Geometric Measure Theory and Discrete Exterior Calculus*, in Proceedings of the IEEE/CVF Conference on computer vision and Pattern Recognition (CVPR), 2019.

[89] T. Möllenhoff, E. Laude, M. Moeller, J. Lellmann, and D. Cremers, *Sublabel-accurate relaxation of nonconvex energies*, in Proceedings of the IEEE/CVF Conference on computer vision and Pattern Recognition (CVPR), 2016.

[90] J. J. Moreau, *Inf-convolution, sous-additivité, convexité des fonctions numériques*, Journal de Mathématiques Pures et Appliquées, (1970), pp. 33–41.

[91] T. Möllenhoff and D. Cremers, *Sublabel-Accurate Discretization of Nonconvex Free-Discontinuity Problems*, in Proceedings of the International Conference on computer vision (ICCV), 2017.

[92] T. Möllenhoff, E. Laude, M. Moeller, J. Lellmann, and D. Cremers, *Sublabel-Accurate Relaxation of Nonconvex Energies*, in Proceedings of the IEEE/CVF Conference on computer vision and Pattern Recognition (CVPR), 2016.

[93] P. Ochs, T. Meinhardt, L. Leal-Taixe, and M. Moeller, *Lifting Layers: Analysis and Applications*, in Proceedings of the European Conference on computer vision (ECCV), 2018.

[94] J. Peng, T. Hazan, D. McAllester, and R. Urtasun, *Convex Max-Product Algorithms for Continuous MRFs with Applications to Protein Folding*, in Proceedings of the International Conference on Machine Learning (ICML), 2011.

[95] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, *Privacy-Preserving Deep Learning: Revisited and Enhanced*, in Applications and Techniques in Information Security, Communications in Computer and Information Science, Singapore, 2017, Springer, pp. 100–110.

[96] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, *Privacy-Preserving Deep Learning via Additively Homomorphic Encryption*, IEEE Transactions on Information Forensics and Security, 13 (2018), pp. 1333–1345.

[97] T. Pock and A. Chambolle, *Diagonal Preconditioning for First Order Primal-Dual Algorithms in Convex Optimization*, in Proceedings of the International Conference on computer vision (ICCV), 2011.

[98] T. Pock, A. Chambolle, D. Cremers, and H. Bischof, *A Convex Relaxation Approach for Computing Minimal Partitions*, in Proceedings of the IEEE/CVF Conference on computer vision and Pattern Recognition (CVPR), 2009.

[99] T. Pock, D. Cremers, H. Bischof, and A. Chambolle, *Global Solutions of Variational Models with Convex Regularization*, SIAM Journal on Imaging Sciences (SIIMS), 3 (2010), pp. 1122–1145.

[100] T. Pock, T. Schoenemann, G. Graber, H. Bischof, and D. Cremers, *A Convex Formulation of Continuous Multi-Label Problems*, in Proceedings of the European Conference on computer vision (ECCV), 2008.

[101] R. Poliquin, *Subgradient monotonicity and convex functions*, Nonlinear Analysis: Theory, Methods & Applications, 14 (1990), pp. 305–317.

[102] V. Powers and B. Reznick, *Polynomials that are positive on an interval*, Transactions of the American Mathematical Society, 352 (2000), pp. 4677–4692.

[103] M. Putinar, *Positive polynomials on compact semi-algebraic sets*, Indiana University Mathematics Journal, 42 (1993), pp. 969–984.

[104] R. Ranftl, S. Gehrig, T. Pock, and H. Bischof, *Pushing the limits of stereo using variational stereo estimation*, in Intelligent Vehicles Symposium (IV), IEEE, 2012, pp. 401–407.

[105] R. Ranftl, T. Pock, and H. Bischof, *Minimizing TGV-Based Variational Models with Non-convex Data Terms*, in Scale Space and Variational Methods in computer vision, Lecture Notes in Computer Science, Springer Berlin Heidelberg, June 2013, pp. 282–293.

[106] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan, *Adaptive Federated Optimization*, in Proceedings of the International Conference on Learning Representations (ICLR), 2021.

[107] J. Rick Chang, C.-L. Li, B. Poczos, B. Vijaya Kumar, and A. Sankaranarayanan, *One Network to Solve Them All–Solving Linear Inverse Problems Using Deep Projection Models*, in Proceedings of the International Conference on computer vision (ICCV), 2017.

[108] R. T. Rockafellar, *Duality and stability in extremum problems involving convex functions.*, Pacific Journal of Mathematics, 21 (1967), pp. 167–187.

[109] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, New Jersey, 1970.

[110] R. T. Rockafellar, *Augmented Lagrange multiplier functions and duality in nonconvex programming*, SIAM Journal on Optimization (SIOPT), 12 (1974), pp. 268–285.

[111] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*, vol. 317, Springer Science & Business Media, 2009.

[112] Y. Romano, M. Elad, and P. Milanfar, *The Little Engine that Could: Regularization by Denoising (RED)*, SIAM Journal on Imaging Sciences (SIIMS), 10 (2017), pp. 1804–1844.

[113] L. Rosasco, A. Caponnetto, E. D. Vito, U. D. Giovannini, and F. Odone, *Learning, Regularization and Ill-Posed Inverse Problems*, in Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2004.

[114] L. I. Rudin, S. Osher, and E. Fatemi, *Nonlinear Total Variation Based Noise Removal Algorithms*, Physica D: Nonlinear Phenomena, 60 (1992), pp. 259–268.

[115] W. Rudin, *Functional Analysis*, International Series In Pure And Applied Mathematics, McGraw-Hill, 1991.

[116] N. Ruozzi, *Exactness of approximate MAP inference in continuous MRFs*, in Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2015.

[117] M. Salzmann, *Continuous Inference in Graphical Models with Polynomial Energies*, in Proceedings of the IEEE/CVF Conference on computer vision and Pattern Recognition (CVPR), 2013.

[118] K. G. G. Samuel and M. F. Tappen, *Learning Optimized MAP Estimates in Continuously-Valued MRF Models*, in Proceedings of the IEEE/CVF Conference on computer vision and Pattern Recognition (CVPR), 2009.

[119] F. Santambrogio, *Optimal Transport for Applied Mathematicians*, Birkhäuser, New York, 2015.

[120] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling, *High-Resolution Stereo Datasets with Subpixel-Accurate Ground Truth*, in Proceedings of the German Conference on Pattern Recognition (GCPR), 2014.

[121] K. Schmüdgen, *The k-moment problem for compact semi-algebraic sets*, Mathematische Annalen, 289 (1991), pp. 203–206.

[122] V. Shah and C. Hegde, *Solving Linear Inverse Problems Using GAN Priors: An Algorithm with Provable Guarantees*, in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2018, pp. 4609–4613.

[123] R. Shokri and V. Shmatikov, *Privacy-Preserving Deep Learning*, in Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS), Denver, Colorado, USA, 2015, ACM Press, pp. 1310–1321.

[124] M. Simões, A. Themelis, and P. Patrinos, *Lasry-Lions envelopes and nonconvex optimization: A homotopy approach*, in Proceedings of the European Signal Processing Conference (EUSIPCO), 2021.

[125] M. Souiai, M. R. Oswald, Y. Kee, J. Kim, M. Pollefeys, and D. Cremers, *Entropy Minimization for Convex Relaxation Approaches*, in Proceedings of the International Conference on computer vision (ICCV), 2015.

[126] G. Stengle, *A Nullstellensatz and a Positivstellensatz in semialgebraic geometry*, Mathematische Annalen, 207 (1974), pp. 87–97.

[127] E. Strekalovskiy, A. Chambolle, and D. Cremers, *Convex Relaxation of Vectorial Problems with Coupled Regularization*, SIAM Journal on Imaging Sciences (SIIMS), 7 (2014), pp. 294–336.

[128] E. Strekalovskiy and D. Cremers, *Real-Time Minimization of the Piecewise Smooth Mumford-Shah Functional*, in Proceedings of the European Conference on computer vision (ECCV), 2014.

[129] E. Strekalovskiy, B. Goldluecke, and D. Cremers, *Tight Convex Relaxations for Vector-Valued Labeling Problems*, in Proceedings of the International Conference on computer vision (ICCV), 2011.

[130] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, *Intriguing properties of neural networks*, arXiv:1312.6199 [cs.CV], preprint available at https://arxiv.org/pdf/1312.6199, (2013).

[131] D. Tenbrinck, F. Gaede, and M. Burger, *Variational Graph Methods for Efficient Point Cloud Sparsification*, arXiv:1903.02858 [math.NA], preprint available at https://arxiv.org/pdf/1903.02858, (2019).

[132] H. Trinh and D. McAllester, *Particle-based belief propagation for structure from motion and dense stereo vision with unknown camera constraints*, in International Workshop on Robot Vision, Springer, 2008, pp. 16–28.

[133] D. Ulyanov, A. Vedaldi, and V. Lempitsky, *Deep image prior*, in Proceedings of the IEEE/CVF Conference on computer vision and Pattern Recognition (CVPR), 2018.

[134] M. Unser, *A Representer Theorem for Deep Neural Networks*, Journal of Machine Learning Research, 20 (2019), pp. 1–30.

[135] R. Uziel, M. Ronen, and O. Freifeld, *Bayesian Adaptive Superpixel Segmentation*, in Proceedings of the International Conference on computer vision (ICCV), 2019.

[136] C. Villani, *Topics in Optimal Transportation*, no. 58 in Graduate studies in mathematics, American Mathematical Society, 2003.

[137] C. Villani, *Optimal Transport: Old and New*, Grundlehren der mathematischen Wissenschaften, Springer, 2008.

[138] T. Vogt, R. Haase, D. Bednarski, and J. Lellmann, *On the Connection between Dynamical Optimal Transport and Functional Lifting*, arXiv:2007.02587 [math.OC], preprint available at https://arxiv.org/pdf/2007.02587, (2020).

[139] T. Vogt, E. Strekalovskiy, D. Cremers, and J. Lellmann, *Lifting methods for manifold-valued variational problems*, in Handbook of Variational Methods for Nonlinear Geometric Data, Springer, 2020, pp. 95–119.

[140] M. J. Wainwright, M. I. Jordan, et al., *Graphical models, exponential families, and variational inference*, Foundations and Trends® in Machine Learning, 1 (2008), pp. 1–305.

[141] H. Waki, S. Kim, M. Kojima, and M. Muramatsu, *Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity*, SIAM Journal on Optimization (SIOPT), 17 (2006), pp. 218–242.

[142] Y. Wald and A. Globerson, *Tightness Results for Local Consistency Relaxations in Continuous MRFs.*, in Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI), 2014.

[143] S. Wang, A. Schwing, and R. Urtasun, *Efficient inference of continuous Markov random fields with polynomial potentials*, in Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2014.

[144] Y.-X. Wang, D. Ramanan, and M. Hebert, *Growing a brain: Fine-tuning by increasing model capacity*, in Proceedings of the IEEE/CVF Conference on computer vision and Pattern Recognition (CVPR), 2017.

[145] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, *Beyond inferring class representatives: User-level privacy leakage from federated learning*, in IEEE INFOCOM - IEEE Conference on Computer Communications, 2019, pp. 2512–2520.

[146] A. Weinmann, L. Demaret, and M. Storath, *Total variation regularization for manifold-valued data*, SIAM Journal on Imaging Sciences (SIIMS), 7 (2014), pp. 2226–2257.

[147] A. Weinmann, L. Demaret, and M. Storath, *Mumford–Shah and Potts regularization for manifold-valued data*, Journal of Mathematical Imaging and Vision, 55 (2016), pp. 428–445.

[148] T. Weisser, J. B. Lasserre, and K.-C. Toh, *Sparse-BSOS: A bounded degree SOS hierarchy for large scale polynomial optimization with sparsity*, Mathematical Programming Computation (MPC), 10 (2018), pp. 1–32.

[149] M. Werlberger, M. Unger, T. Pock, and H. Bischof, *Efficient Minimization of the Non-local Potts Model*, in Scale Space and Variational Methods in computer vision, Lecture Notes in Computer Science, Springer Berlin Heidelberg, May 2011, pp. 314–325.

[150] T. Werner, *A linear programming approach to max-sum problem: A review*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 29 (2007), pp. 1165–1179.

[151] T. M. Wong, M. Kahl, P. Haring Bolívar, and A. Kolb, *Computational image enhancement for frequency modulated continuous wave (FMCW) THz image*, Journal of Infrared, Millimeter, and Terahertz Waves, 40 (2019), pp. 775–800.

[152] T. M. Wong, M. Kahl, P. Haring Bolívar, A. Kolb, and M. Möller, *Training Auto-Encoder-Based Optimizers for Terahertz Image Reconstruction*, in Proceedings of the German Conference on Pattern Recognition (GCPR), 2019.

[153] K. Yamaguchi, T. Hazan, D. McAllester, and R. Urtasun, *Continuous Markov random fields for robust stereo estimation*, in Proceedings of the European Conference on computer vision (ECCV), 2012.

[154] Q. Yang, Y. Liu, T. Chen, and Y. Tong, *Federated Machine Learning: Concept and Applications*, ACM Transactions on Intelligent Systems and Technology (TIST), 10 (2019), pp. 1–19.

[155] C. Zach, *Dual Decomposition for Joint Discrete-Continuous Optimization*, in Proceedings of the Artificial Intelligence and Statistics (AISTATS), 2013.

[156] C. Zach, C. Häne, and M. Pollefeys, *What Is Optimized in Tight Convex Relaxations for Multi-Label Problems?*, in Proceedings of the International Conference on computer vision (ICCV), 2012.

[157] C. Zach and P. Kohli, *A Convex Discrete-Continuous Approach for Markov random fields*, in Proceedings of the European Conference on computer vision (ECCV), 2012.

[158] C. Zach, T. Pock, and H. Bischof, *A Duality Based Approach for Realtime TV-L1 Optical Flow*, in Joint pattern recognition symposium, 2007, pp. 214–223.

[159] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, *Understanding Deep Learning (Still) Requires Rethinking Generalization*, Communications of the ACM, 64 (2021), pp. 107–115.

[160] K. Zhang, W. Zuo, and L. Zhang, *FFDNet: Toward a Fast and Flexible Solution for CNN-Based Image Denoising*, IEEE Transactions on Image Processing, 27 (2018), pp. 4608–4622.

[161] Y. Zhang, R. Jia, H. Pei, W. Wang, B. Li, and D. Song, *The Secret Revealer: Generative Model-Inversion Attacks Against Deep Neural Networks*, in Proceedings of the IEEE/CVF Conference on computer vision and Pattern Recognition (CVPR), 2020.

[162] B. Zhao, K. R. Mopuri, and H. Bilen, *iDLG: Improved Deep Leakage from Gradients*, arXiv:2001.02610 [cs.LG], preprint available at http://arxiv.org/abs/2001.02610, (2020).

[163] L. Zhu, Z. Liu, and S. Han, *Deep Leakage from Gradients*, in Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2019.