Bilevel Methods for Image Reconstruction

Other titles in Foundations and Trends® in Signal Processing

Foundations of User-Centric Cell-Free Massive MIMO
Özlem Tugfe Demir, Emil Björnson and Luca Sanguinetti
ISBN: 978-1-68083-790-2

Data-Driven Multi-Microphone Speaker Localization on Manifolds
Bracha Laufer-Goldshtein, Ronen Talmon and Sharon Gannot
ISBN: 978-1-68083-736-0

Recent Advances in Clock Synchronization for Packet-Switched Networks
Anantha K. Karthik and Rick S. Blum
ISBN: 978-1-68083-726-1

Biomedical Image Reconstruction: From the Foundations to Deep Neural Networks
Michael T. McCann and Michael Unser
ISBN: 978-1-68083-650-9

Compressed Sensing with Applications in Wireless Networks
Markus Leinonen, Marian Codreanu and Georgios B. Giannakis
ISBN: 978-1-68083-646-2

Bilevel Methods for Image Reconstruction

Caroline Crockett
University of Michigan
cecroc@umich.edu

Jeffrey A. Fessler
University of Michigan
fessler@umich.edu

now
the essence of knowledge
Boston — Delft

Foundations and Trends® in Signal Processing

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

C. Crockett and J. A. Fessler. *Bilevel Methods for Image Reconstruction*. Foundations and Trends® in Signal Processing, vol. 15, no. 2-3, pp. 121–289, 2022.

ISBN: 978-1-63828-003-3

Foundations and Trends® in Signal Processing
Volume 15, Issue 2-3, 2022

Editorial Scope

Topics

Foundations and Trends® in Signal Processing publishes survey and tutorial articles in the following topics:

- Adaptive signal processing
- Audio signal processing
- Biological and biomedical signal processing
- Complexity in signal processing
- Digital signal processing
- Distributed and network signal processing
- Image and video processing
- Linear and nonlinear filtering
- Multidimensional signal processing
- Multimodal signal processing
- Multirate signal processing
- Multiresolution signal processing
- Nonlinear signal processing
- Randomized algorithms in signal processing
- Sensor and multiple source signal processing, source separation
- Signal decompositions, subband and transform methods, sparse representations
- Signal processing for communications
- Signal processing for security and forensic analysis, biometric signal processing
- Signal quantization, sampling, analog-to-digital conversion, coding and compression
- Signal reconstruction, digital-to-analog conversion, enhancement, decoding and inverse problems
- Speech/audio/image/video compression
- Speech and spoken language processing
- Statistical/machine learning
- Statistical signal processing
  - Classification and detection
  - Estimation and regression
  - Tree-structured methods

Information for Librarians

Foundations and Trends® in Signal Processing, 2022, Volume 15, 4 issues. ISSN paper version 1932-8346. ISSN online version 1932-8354. Also available as a combined paper and online subscription.

Contents

Bilevel Methods for Image Reconstruction

Caroline Crockett and Jeffrey A. Fessler

Department of EECS, University of Michigan, Ann Arbor, Michigan, USA; cecroc@umich.edu, fessler@umich.edu

ABSTRACT

This review discusses methods for learning parameters for image reconstruction problems using bilevel formulations. Image reconstruction typically involves optimizing a cost function to recover a vector of unknown variables that agrees with collected measurements and prior assumptions. State-of-the-art image reconstruction methods learn these prior assumptions from training data using various machine learning techniques, such as bilevel methods.

One can view the bilevel problem as formalizing hyperparameter optimization, as bridging machine learning and cost function based optimization methods, or as a method to learn variables best suited to a specific task. More formally, bilevel problems attempt to minimize an upper-level loss function, where variables in the upper-level loss function are themselves minimizers of a lower-level cost function.

This review contains a running example problem of learning tuning parameters and the coefficients for sparsifying filters used in a regularizer. Such filters generalize the popular total variation regularization method, and learned filters are closely related to convolutional neural networks approaches that are rapidly gaining in popularity. Here, the lower-level

Caroline Crockett and Jeffrey A. Fessler (2022), “Bilevel Methods for Image Reconstruction”, Foundations and Trends® in Signal Processing: Vol. 15, No. 2-3, pp 121–289. DOI: 10.1561/2000000111.

problem is to reconstruct an image using a regularizer with learned sparsifying filters; the corresponding upper-level optimization problem involves a measure of reconstructed image quality based on training data.

This review discusses multiple perspectives to motivate the use of bilevel methods and to make them more easily accessible to different audiences. We then turn to ways to optimize the bilevel problem, providing pros and cons of the variety of proposed approaches. Finally we overview bilevel applications in image reconstruction.

1

Introduction

Methods for image recovery aim to estimate a good-quality image from noisy, incomplete, or indirect measurements. Such methods are also known as computational imaging. For example, image denoising and image deconvolution attempt to recover a clean image from a noisy and/or blurry input image, and image inpainting tries to complete missing measurements from an image. Medical image reconstruction aims to recover images that humans can interpret from the indirect measurements recorded by a system like a Magnetic Resonance Imaging (MRI) or Computed Tomography (CT) scanner. Such image reconstruction applications are a type of inverse problem [52].

New methods for image reconstruction attempt to lower complexity, decrease data requirements, or improve image quality for a given input data quality. For example, in CT, one goal is to provide doctors with information to help their patients while reducing radiation exposure [127]. To achieve these lower radiation doses, the CT system must collect data with lower beam intensity or fewer views. Similarly, in MRI, collecting fewer k-space samples can reduce scan times. Such "undersampling" leads to an under-determined problem, with fewer knowns (measurements from a scanner) than unknowns (pixels in the reconstructed image), requiring advanced image reconstruction methods.

Existing reconstruction methods make different assumptions about the characteristics of the images being recovered. Historically, the assumptions are based on easily observed (or assumed) characteristics of the desired output image, such as a tendency to have smooth regions with few edges or to have some form of sparsity [49]. More recent machine learning approaches use training data to discover image characteristics. These learning-based methods often outperform traditional methods, and are gaining popularity in part because of increased availability of training data and computational resources [84], [184].

There are many design decisions in learning-based reconstruction methods. How many parameters should be learned? What makes a set of parameters "good?" How can one learn these good parameters? Using a bilevel methodology is one systematic way to address these questions.

Bilevel methods are so named because they involve two "levels" of optimization: an upper-level loss function that defines a goal or measure of goodness (equivalently, badness) for the learnable parameters and a lower-level cost function that uses the learnable parameters, typically as part of a regularizer. The main benefits of bilevel methods are learning task-based hyperparameters in a principled approach and connecting machine learning techniques with image reconstruction methods that are defined in terms of optimizing a cost function, often called model-based image reconstruction methods. Conversely, the main challenge with bilevel methods is the computational complexity. However, like with neural networks, that complexity is highest during the training process, whereas deployment has lower complexity because it uses only the lower-level problem.

The methods in this review are broadly applicable to bilevel problems, but we focus on formulations and applications where the lower-level problem is an image reconstruction cost function that uses regularization based on analysis sparsity. The application of bilevel methods to image reconstruction problems is relatively new, but there are a growing number of promising research efforts in this direction. We hope this review serves as a primer and unifying treatment for readers who may already be familiar with image reconstruction problems and traditional regularization approaches but who have not yet delved into bilevel methods.

This review lies at the intersection of a specific machine learning method, bilevel, and a specific application, filter learning for image reconstruction. For overviews of machine learning in image reconstruction, see [84], [151]. For an overview of image reconstruction methods, including classical, variational, and learning-based methods, see [125]. Finally, for historical overviews of bilevel optimization and perspectives on its use in a wide variety of fields, see [41], [42]. Within the image recovery field, bilevel methods have also been used, *e.g.*, in learning synthesis dictionaries [122].

The structure of this review is as follows. The remainder of the introduction defines our notation and presents a running example bilevel problem. Section 2 provides background information on the lower-level image reconstruction cost function and analysis regularizers. Section 3 provides background information on the upper-level loss function, specifically loss function design and hyperparameter optimization strategies. These background sections provide motivation and context for the rest of the review; they are not exhaustive overviews of these broad topics. Section 4 presents building blocks for optimizing a bilevel problem. Section 5 uses these building blocks to discuss optimization methods for the upper-level loss function. Section 6 discusses previous applications of the bilevel method in image recovery problems, including signal denoising, image inpainting, and medical image reconstruction. It also overviews bilevel formulations for blind learning and learning space-varying tuning parameters. Finally, Section 7 offers summarizing commentary on the benefits and drawbacks of bilevel methods for computational imaging, connects and compares bilevel methods to other machine learning approaches, and proposes future directions for the field.

1.1 Notation

This review focuses on continuous-valued, discrete space signals. Some papers, *e.g.*, [14], [38], analyze signals in function space, arguing that the goal of high resolution imagery is to approximate a continuous space reality and that analysis in the continuous domain can yield insights and optimization algorithms that are resolution independent. However, the majority of bilevel methods are motivated and described in discrete space. The review does not include discrete-valued settings, such as image segmentation; those problems often require different techniques to optimize the lower-level cost function, although some recent work uses dual formulations to bridge this gap [109], [137].

The literature is inconsistent in how it refers to variables in machine learning problems. For consistency within this document, we define the following terms:

- **Hyperparameters**: Any adjustable parameters that are part of a model. Tuning parameters and model parameters are both sub-types of hyperparameters. This document uses $\boldsymbol{\gamma}$ to denote a vector of hyperparameters.
- **Tuning parameters**: Scalar parameters that weight terms in a cost function to determine the relative importance of each term. This review uses β to denote individual tuning parameters.
- **Model parameters**: Parameters, generally in vector or matrix form, that are used in the structure of a cost or loss function, typically as part of the regularization term. In the running example in the next section, the model parameters are typically filter coefficients, denoted $\boldsymbol{c}$.

We write vectors as column vectors and use bold to denote matrices (uppercase letters) and vectors (lowercase letters). Subscripts index vector elements, so x_i is the ith element in $\boldsymbol{x}$. For functions that are applied element-wise to vectors, we use notation following the Julia programming language [8], where $f.(\boldsymbol{x})$ denotes the function f applied element wise to its argument:

$$\boldsymbol{x} \in \mathbb{F}^N \implies f.(\boldsymbol{x}) = \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_N) \end{bmatrix} \in \mathbb{F}^N.$$

We will often use this notation in combination with a transposed vector of ones to sum the result of a function applied element-wise to a vector, *i.e.*,

$$\mathbf{1}' f.(\boldsymbol{x}) = \sum_{i=1}^{N} f(x_i). \tag{1.1}$$

For example, the standard Euclidean norm is equivalent to $\mathbf{1}'f.(\boldsymbol{x})$ when $f(x) = |x|^2$ and and the vector 1-norm can be similarly written when $f(x) = |x|$. This notation is helpful for regularizers that do not correspond to norms. The field $\mathbb{F}$ can be either $\mathbb{R}$ or $\mathbb{C}$, depending on the application.

Convolution between a vector, $\boldsymbol{x}$, and a filter, $\boldsymbol{c}$, is denoted as $\boldsymbol{c} \circledast \boldsymbol{x}$. This review assumes all convolutions use circular boundary conditions. Thus, convolution is equivalent to multiplication with a square, circulant matrix:

$$\boldsymbol{c} \circledast \boldsymbol{x} = \boldsymbol{C}\boldsymbol{x}.$$

The conjugate mirror reversal of $\boldsymbol{c}$ is denoted as $\tilde{\boldsymbol{c}}$ and its application is equivalent to multiplying with the adjoint of $\boldsymbol{C}$:

$$\tilde{\boldsymbol{c}} \circledast \boldsymbol{x} = \boldsymbol{C}'\boldsymbol{x},$$

where the prime indicates the Hermitian transpose operation.

Finally, for partial derivatives, we use the notation that:

$$\begin{aligned}
\nabla_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{y}) &= \frac{\partial f(\boldsymbol{x}, \boldsymbol{y})}{\partial \boldsymbol{x}} \in \mathbb{F}^N, \\
\nabla_{\boldsymbol{xy}} f(\boldsymbol{x}, \boldsymbol{y}) &= \left[\frac{\partial^2 f(\boldsymbol{x}, \boldsymbol{y})}{\partial x_i \partial y_j} \right] \in \mathbb{F}^{N \times M}, \text{ and} \\
\nabla_{\boldsymbol{xy}} f(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) &= \nabla_{\boldsymbol{xy}} f(\boldsymbol{x}, \boldsymbol{y}) \Big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}, \boldsymbol{y}=\hat{\boldsymbol{y}}} \in \mathbb{F},
\end{aligned} \tag{1.2}$$

where $f : \mathbb{F}^N \times \mathbb{F}^M \to \mathbb{F}$.

Tables 1.1 and 1.2 summarize our frequently used notation for variables and functions.

1.2 Defining a Bilevel Problem

This section introduces a generic bilevel problem; the next presents a specific bilevel problem that serves as a running example throughout the review. Later sections discuss many of the ideas presented here more thoroughly. Our hope is that an early introduction to the formal problem motivates readers and that this section acts as a quick-reference guide to our notation.

Table 1.1: Overview of frequently used symbols in the review.

Variable	Dim	Description
$\boldsymbol{x}_j^{\text{true}}$	N	One of J clean, noiseless training signals. Often used in a supervised training set-up.
$\boldsymbol{A}$	$M \times N$	Forward operator for the system of interest.
$\boldsymbol{y}_j$	M	During the bilevel learning process, $\boldsymbol{y}_j$ refers to simulated measurements, where $\boldsymbol{y}_j = \boldsymbol{A}\boldsymbol{x}_j^{\text{true}} + \boldsymbol{n}_j$. Once $\boldsymbol{\gamma}$ is learned, $\boldsymbol{y}$ refers to collected measurements.
$\boldsymbol{n}_j$	N	A noise realization.
$\hat{\boldsymbol{x}}_j$	N	A reconstructed image.
$\boldsymbol{\gamma}$	R	The vector of parameters to learn using bilevel methods. This often includes $\boldsymbol{c}_k$ and/or β_k.
$\boldsymbol{c}_k$	S	One of K convolutional filters. A 2D filter might be $\sqrt{S} \times \sqrt{S}$.
$\tilde{\boldsymbol{c}}_k$	S	Conjugate mirror reversal of filter $\boldsymbol{c}_k$.
$\boldsymbol{C}_k$	$N \times N$	The convolution matrix such that $\boldsymbol{C}_k\boldsymbol{x} = \boldsymbol{c}_k \circledast \boldsymbol{x}$ and $\boldsymbol{C}_k'\boldsymbol{x} = \tilde{\boldsymbol{c}}_k \circledast \boldsymbol{x}$.
β_k	$\mathbb{R}$	The tuning parameter associated with $\boldsymbol{c}_k$.
β_0	$\mathbb{R}$	An overall regularization (tuning) parameter, appearing as e^{β_0} in (Ex).
$\boldsymbol{\Omega}$	$F \times N$	A matrix with filters in each row. For the stacked convolution matrices in (2.7) $F = KN$.
$\boldsymbol{z}$	Varies	A sparse vector, often from $\boldsymbol{C}_k\boldsymbol{x}$.
ϵ	$\mathbb{R}_+$	Parameter used to define ϕ. Typically determines the amount of corner-rounding.
t	$0, \dots, T$	Iteration counter for the lower-level optimization iterates, *e.g.*, $\boldsymbol{x}^{(t)}$ is the estimate of the lower-level optimization variable $\boldsymbol{x}$ at the tth iteration.
u	$0, \dots, U$	Iteration counter for the upper-level optimization iterates, *e.g.*, $\boldsymbol{\gamma}^{(u)}$.

Table 1.2: Overview of frequently used functions in the review.

Function	Description
$\ell(\boldsymbol{\gamma}) \mapsto \mathbb{R}$ or $\ell(\boldsymbol{\gamma}, \boldsymbol{x}) \mapsto \mathbb{R}$	Upper-level loss function used as a fitness measure of $\boldsymbol{\gamma}$. Although ℓ is a function of $\boldsymbol{\gamma}$, it is often helpful to write it with two inputs, where typically $\boldsymbol{x} = \hat{\boldsymbol{x}}$.
$\Phi(\boldsymbol{x}\,; \boldsymbol{\gamma}) \mapsto \mathbb{R}$	Lower-level cost function used for reconstructing an image.
$R(\boldsymbol{x}) \mapsto \mathbb{R}$	Regularization function. Incorporates prior information about likely image characteristics.
$d(\boldsymbol{x}, \boldsymbol{y}) \mapsto \mathbb{R}$	Data-fit term.
$\phi(z) \mapsto \mathbb{R}$	Sparsity promoting function, *e.g.*, 0-norm, 1-norm, or corner-rounded 1-norm. Typically used in R.

This review considers the image reconstruction problem where the goal is to form an estimate $\hat{\boldsymbol{x}} \in \mathbb{F}^N$ of a (vectorized) latent image, given a set of measurements $\boldsymbol{y} \in \mathbb{F}^M$. For denoising problems, $N = M$, but the two dimensions may differ significantly in more general image reconstruction problems. The forward operator, $\boldsymbol{A} \in \mathbb{F}^{M \times N}$ models the physics of the system such that one would expect $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ in an ideal (noiseless) system. We focus on linear imaging systems here, but the concepts generalize readily to nonlinear forward models. When known (in a supervised training setting), we denote the true, underlying signal as $\boldsymbol{x}^{\text{true}} \in \mathbb{F}^N$. Most bilevel methods are supervised, but Section 6.2 presents a few examples of unsupervised bilevel methods.

We focus on model-based image reconstruction methods where the goal is to estimate $\boldsymbol{x}$ from $\boldsymbol{y}$ by solving an optimization problem of the form:

$$\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}(\boldsymbol{\gamma}) = \underset{\boldsymbol{x} \in \mathbb{F}^N}{\operatorname{argmin}}\, \Phi(\boldsymbol{x}\,; \boldsymbol{\gamma}, \boldsymbol{y}). \tag{1.3}$$

To simplify notation, we drop $\boldsymbol{y}$ from the list of Φ arguments except where needed for clarity. The quality of the estimate $\hat{\boldsymbol{x}}$ can depend greatly on the choice of the hyperparameters $\boldsymbol{\gamma}$. Historically there have been numerous approaches pursued for choosing $\boldsymbol{\gamma}$, such as cross validation [176], generalized cross validation [75], the discrepancy principle [145] and Bayesian methods [160], among others.

Bilevel methods provide a framework for choosing hyperparameters. A bilevel problem for learning hyperparameters $\boldsymbol{\gamma}$ has the following "double minimization" form:

$$\hat{\boldsymbol{\gamma}} = \operatorname*{argmin}_{\boldsymbol{\gamma}\in\mathbb{F}^R} \underbrace{\ell(\boldsymbol{\gamma}\,;\hat{\boldsymbol{x}}(\boldsymbol{\gamma}))}_{\ell(\boldsymbol{\gamma})} \text{ where} \tag{UL}$$

$$\hat{\boldsymbol{x}}(\boldsymbol{\gamma}) = \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{F}^N} \Phi(\boldsymbol{x}\,;\boldsymbol{\gamma}). \tag{LL}$$

Fig. 1.1 depicts a generic bilevel problem for image reconstruction. The upper-level (UL) loss function, $\ell : \mathbb{R}^R \times \mathbb{F}^N \mapsto \mathbb{R}$, quantifies how (not) good is a vector $\boldsymbol{\gamma}$ of learnable parameters. The upper-level depends on the solution to the lower-level (LL) cost function, Φ, which depends on $\boldsymbol{\gamma}$. The upper-level can also be called the outer optimization, with the lower-level being the inner optimization. Another terminology is leader-follower, as the minimizer of the lower-level follows where the upper-level loss leads. We will also write the upper-level loss function with a single parameter as $\ell(\boldsymbol{\gamma}) := \ell(\boldsymbol{\gamma}\,;\hat{\boldsymbol{x}}(\boldsymbol{\gamma}))$.

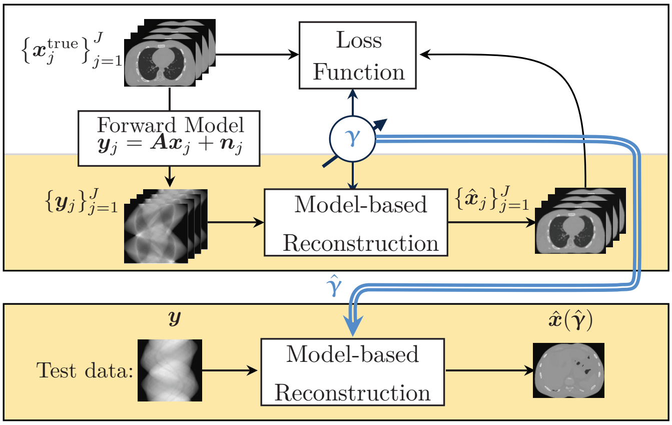

$\{\boldsymbol{x}_j^{\text{true}}\}_{j=1}^J$
Loss Function
Forward Model
$\boldsymbol{y}_j = \boldsymbol{A}\boldsymbol{x}_j + \boldsymbol{n}_j$
γ
$\{\boldsymbol{y}_j\}_{j=1}^J$
Model-based Reconstruction
$\{\hat{\boldsymbol{x}}_j\}_{j=1}^J$
$\hat{\gamma}$
$\boldsymbol{y}$
$\hat{\boldsymbol{x}}(\hat{\gamma})$
Test data:
Model-based Reconstruction

Figure 1.1: Depiction of a typical bilevel problem for image reconstruction, illustrated using XCAT phantom from [162]. The upper box represents the training process, with the upper-level loss and lower-level cost function. During training, one minimizes the upper-level loss with respect to a vector of parameters, $\boldsymbol{\gamma}$, that are used in the image reconstruction task. Once learned, $\hat{\boldsymbol{\gamma}}$ is typically deployed in the same image reconstruction task, shown in the lower box.

We write the lower-level cost as an optimization problem with "argmin" and thus implicitly assume that Φ has unique minimizer, $\hat{\boldsymbol{x}}$. The lower-level is guaranteed to have a unique minimizer when Φ is a strictly convex function of $\boldsymbol{x}$. (See Section 4 for more discussion of this point). More generally, there may be a set of lower-level minimizers, each having some possibly distinct upper-level loss function value. For more discussion, [41] defines optimistic and pessimistic versions of the bilevel problem for the case of multiple lower-level solutions.

Bilevel methods typically use training data. Specifically, one often assumes that a given set of J good quality images $\boldsymbol{x}_1^{\text{true}}, \ldots, \boldsymbol{x}_J^{\text{true}} \in \mathbb{F}^N$ are representative of the images of interest in a given application. (For simplicity of notation we assume the training images have the same size, but they can have different sizes in practice.) We typically generate corresponding simulated measurements for each training image using the imaging system model:

$$\boldsymbol{y}_j = \boldsymbol{A}\boldsymbol{x}_j^{\text{true}} + \boldsymbol{n}_j, \quad j = 1, \ldots, J, \tag{1.4}$$

where $\boldsymbol{n}_j \in \mathbb{F}^M$ denotes an appropriate random noise realization[1]. In (1.4), we add one noise realization to each of the J images; in practice one could add multiple noise realizations to each $\boldsymbol{x}_j^{\text{true}}$ to augment the training data. We then use the training pairs $(\boldsymbol{x}_j^{\text{true}}, \boldsymbol{y}_j)$ to learn a good value of $\boldsymbol{\gamma}$. After those parameters are learned, we reconstruct subsequent test images using (1.3) with the learned hyperparameters $\hat{\boldsymbol{\gamma}}$.

An alternative to the upper-level formulation (UL) is the following stochastic formulation of bilevel learning:

$$\hat{\boldsymbol{\gamma}} = \underset{\boldsymbol{\gamma} \in \mathbb{F}^R}{\operatorname{argmin}} \underbrace{\mathbb{E}\left[\ell(\boldsymbol{\gamma})\right]}_{\approx \frac{1}{J}\sum_{j=1}^{J} \ell(\boldsymbol{\gamma}\,;\hat{\boldsymbol{x}}_j(\boldsymbol{\gamma}))} \tag{1.5}$$

$$\text{where } \hat{\boldsymbol{x}}_j(\boldsymbol{\gamma}) = \underset{\boldsymbol{x} \in \mathbb{F}^N}{\operatorname{argmin}} \Phi(\boldsymbol{x}\,;\boldsymbol{\gamma}, \boldsymbol{y}_j). \tag{1.6}$$

The expectation, taken with respect to the training data and noise distributions, is typically approximated as a sample mean over J training examples.

[1] A more general system model allows the noise to depend on the data and system model, *i.e.*, $\boldsymbol{n}_j(\boldsymbol{A}, \boldsymbol{x}_j)$. This generality is needed for applications with certain noise distributions such as Poisson noise.

The definition of bilevel methods used in (UL) is not universal in the literature. In some works, bilevel methods refer to nested optimization problems with two levels, even when the two levels result from reformulating a single-level problem, *e.g.*, [146]. That definition is much more encompassing, and includes primal-dual reformulations, Lagrangian reformulations of constrained optimization problems, and alternating methods that introduce then minimize over an auxiliary variable.

Another term in the literature, sometimes used interchangeably with a bilevel problem, is a mathematical program with equilibrium constraints (MPEC). As shown in Section 4, many bilevel optimization methods start by transforming the two-level problem into an equivalent single-level problem by replacing the lower-level optimization with a set of constraints based on optimally conditions. Bilevel problems are thus a subset of MPECs. MPECs are generally challenging due to their non-convex nature; even when the lower-level cost function is convex, the upper-level loss function is rarely convex. Importantly, $\ell(\cdot, \cdot)$ is often convex with respect to both arguments. However, $\ell(\boldsymbol{\gamma}) = \ell(\boldsymbol{\gamma}\,; \hat{\boldsymbol{x}}(\boldsymbol{\gamma}))$ is generally non-convex in $\boldsymbol{\gamma}$ due to how the lower-level minimizer depends on $\boldsymbol{\gamma}$. There is a large literature on MPEC problems, *e.g.*, [30], [41], [61], and on non-convex optimization more generally [97]. Bilevel methods are one sub-field in this large literature.

1.3 Running Example

To offer a concrete example, this review will frequently refer to the following running example (Ex), a filter learning bilevel problem:

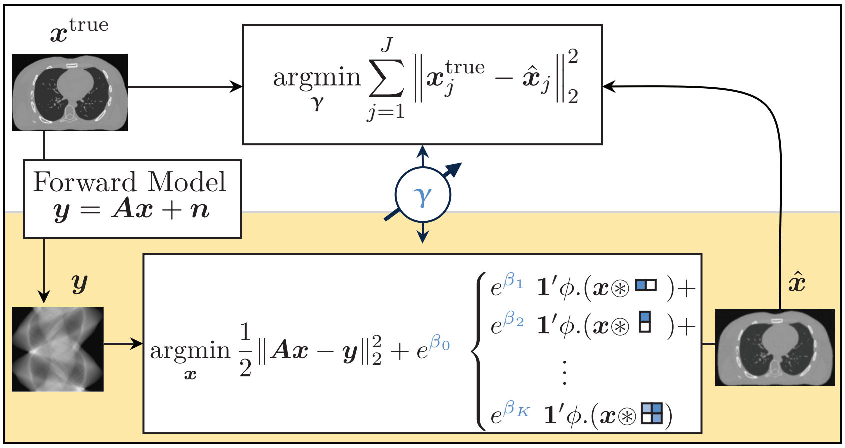
$$\hat{\boldsymbol{\gamma}} = \operatorname*{argmin}_{\boldsymbol{\gamma} \in \mathbb{F}^R} \frac{1}{2} \|\hat{\boldsymbol{x}}(\boldsymbol{\gamma}) - \boldsymbol{x}^{\text{true}}\|_2^2, \text{ where}$$

$$\hat{\boldsymbol{x}}(\boldsymbol{\gamma}) = \operatorname*{argmin}_{\boldsymbol{x} \in \mathbb{F}^N} \frac{1}{2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + e^{\beta_0} \sum_{k=1}^{K} e^{\beta_k} \mathbf{1}' \phi.(\boldsymbol{c}_k \circledast \boldsymbol{x}; \epsilon), \qquad \text{(Ex)}$$

where $\boldsymbol{\gamma} \in \mathbb{F}^R$ contains all variables that we wish to learn: the filter coefficients $\boldsymbol{c}_k \in \mathbb{F}^S$ and tuning parameters $\beta_k \in \mathbb{R}$ for all $k \in [1, K]$. We include an auxiliary tuning parameter, $\beta_0 \in \mathbb{R}$, for easier comparison to other models. Fig. 1.2 depicts the running example and Fig. 1.3 shows example learned filters for a toy training image. Ref. [45] demonstrates how a spectral analysis of learned filters and penalty functions can be interpreted to provide insight into real-world problems.

The learnable hyperparameters can also include the sparsifying function ϕ, its corner rounding parameter ϵ, the forward model $\boldsymbol{A}$, or some aspect of the data-fit term. For example, [45], [82] learn the regularization functional and [46], [167] learn part of the forward model. Such examples are relatively rare in the bilevel methods literature to date.

Unlike many learning problems (see examples in Section 7.4), the running example (Ex) does not include any constraints on $\boldsymbol{\gamma}$. Learned filters should be those that are best at the given task, where "best" is defined by the upper-level loss function. Therefore, a zero mean or norm constraint is not generally required, though some authors have found such constraints helpful, *e.g.*, [25], [111]. Following previous literature, *e.g.*, [159], the tuning parameters in (Ex) are written in terms of an exponential function to ensure positivity. One could re-write (Ex) without this exponentiation "trick" and then add a non-negativity



$\boldsymbol{x}^{\text{true}}$
$\underset{\gamma}{\operatorname{argmin}} \sum_{j=1}^{J} \left\| \boldsymbol{x}_j^{\text{true}} - \hat{\boldsymbol{x}}_j \right\|_2^2$
Forward Model
$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}$
γ
$\boldsymbol{y}$
$\hat{\boldsymbol{x}}$
$\underset{\boldsymbol{x}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + e^{\beta_0} \begin{cases} e^{\beta_1} \ \mathbf{1}'\phi.(\boldsymbol{x} \circledast \ ) + \\ e^{\beta_2} \ \mathbf{1}'\phi.(\boldsymbol{x} \circledast \ ) + \\ \vdots \\ e^{\beta_K} \ \mathbf{1}'\phi.(\boldsymbol{x} \circledast \ ) \end{cases}$

Figure 1.2: Bilevel problem in (Ex). The vector of learnable hyperparameters, $\boldsymbol{\gamma}$, includes the tuning parameters, β_k, and the filter coefficients, $\boldsymbol{c}_k$, shown as example filters. Although this review will generally consider learning filters of a single size, the figure depicts how the framework easily extends to 2d filters of different sizes.

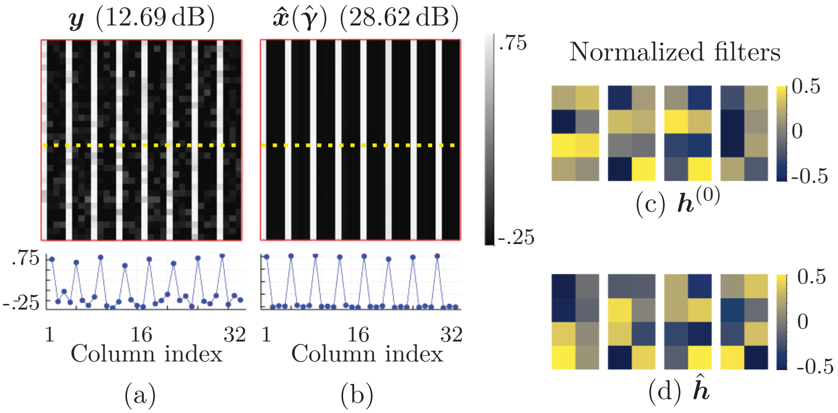


Figure 1.3: Example learned filters for a simple training image, normalized for easier visualization. The true image is zero-mean and repeats three columns of signal value -0.25 and one column of signal value 0.75. (a) Noisy image. The lower plot shows a profile of one row of the image (marked by a dotted line). The signal-to-noise ratio, as defined in (3.2), is given in parenthesis. (b) The denoised image using learned filters as in (Ex). (c) Randomly initialized filters for the bilevel method ($K = 4$ and $S = 4 \cdot 2$). (d) Corresponding learned filters. As expected based on the training image, the learned filters primarily involve vertical differences. Appendix D.1 provides further details including the regularization strength of each learned filter.

constraint to the upper-level problem; most of the methods discussed in this review generalize to this common variation by substituting gradient methods for projected gradient methods.

In (Ex), we drop the sum over J training images for simplicity; the methods easily extend to multiple training signals. For ease of notation, we further simplify by considering $\boldsymbol{c}_k$ to be of length S for all k, *e.g.*, a 2D filter might be $\sqrt{S} \times \sqrt{S}$. In practice, the filters may be of different lengths with minimal impact on the methods presented in this review.

The function ϕ in (Ex) is a sparsity-promoting function. If we were to choose $\phi(z) = |z|$, then the regularizer would involve 1-norm terms of the type common in compressed sensing formulations:

$$\mathbf{1}'\phi.(\boldsymbol{c}_k \circledast \boldsymbol{x}) = \left\|\boldsymbol{c}_k \circledast \boldsymbol{x}\right\|_1 .$$

However, to satisfy differentiability assumptions (see Section 4), this review will often consider ϕ to denote the following "corner rounded"

1-norm having the shape of a hyperbola with the corresponding first and second derivative:

$$\begin{aligned}\phi(z) &= \sqrt{z^2+\epsilon^2} \\ \dot{\phi}(z) &= \frac{z}{\sqrt{z^2+\epsilon^2}} \in [0,1) \\ \ddot{\phi}(z) &= \frac{\epsilon^2}{(z^2+\epsilon^2)^{3/2}} \in (0, \frac{1}{\epsilon}],\end{aligned} \tag{CR1N}$$

where ϵ is a small, relative to the expected range of z, parameter that controls the amount of corner rounding. (Here, we use a dot over the function rather than ∇ to indicate a derivative because ϕ has a scalar argument.)

1.4 Conclusion

Bilevel methods for selecting hyperparameters offer many benefits. Previous papers motivate them as a principled way to approach hyperparameter optimization [42], [94], as a task-based approach to learning [38], [82], [143], and/or as a way to combine the data-driven improvements from learning methods with the theoretical guarantees and explainability provided by cost function-based approaches [14], [26], [111]. A corresponding drawback of bilevel methods are their computational cost; see Sections 4 and 5 for further discussion.

The task-based nature of bilevel methods is a particularly important advantage; Section 7.4 exemplifies why by comparing the bilevel problem to single-level, non-task-based approaches for learning sparsifying filters. Task-based refers to the hyperparameters being learned based on how well they work in the lower-level cost function–the image reconstruction task in our running example. The learned hyperparameters can also adapt to the training dataset and noise characteristics. The task-based nature yields other benefits, such as making constraints or regularizers on the hyperparameters generally unnecessary; Section 6.2 presents some exceptions and [42] further discusses bilevel methods for applications with constraints.

There are three main elements to a bilevel approach. First, the lower-level cost function in a bilevel problem defines a goal, such as image reconstruction, including what hyperparameters can be learned, such as filters for a sparsifying regularizer. Section 2 provides background on this element specifically for image reconstruction tasks, such as the one in (Ex). Section 6.1 reviews example cost functions used in bilevel methods.

Second, the upper-level loss function determines how the hyperparameters should be learned. While the squared error loss function in the running example is a common choice, Section 3 discusses other loss functions based on supervised and unsupervised image quality metrics. Section 6.2 then reviews example loss functions used in bilevel methods.

While less apparent in the written optimization problem, the third main element for a bilevel problem is the optimization approach, especially for the upper-level problem. Section 3.2 briefly discusses various hyperparameter optimization strategies, then Sections 4 and 5 present multiple gradient-based bilevel optimization strategies. Throughout the review, we refer to the running example to show how the bilevel optimization strategies apply.

Acknowledgements

This work was made possible in part due to the support of NIH grant R01 EB023618, NSF grant IIS 1838179, and the Rackham Predoctoral Fellowship. The authors would like to thank Lindon Roberts for helpful email discussion of [46], Mike McCann for general discussion of bilevel approaches and of [72] specifically, and Avrajit Ghosh and Saiprasad Ravishankar for discussion of [72]. The authors would also like to thank Qing Qu and the anonymous reviewers, whose suggestions were a great help in strengthening and clarifying this review.

Appendices

A

Background: Primal-Dual Formulations

This appendix briefly reviews primal-dual analysis as it applies to (Ex). Section 3.3 in [19] provides a more general but brief introduction to the notion of conjugate functions and duality and [10] goes into more depth on duality.

The conjugate of a function $f : \mathbb{R}^N \to \mathbb{R} \cup \{\text{-}\infty, \infty\}$ is denoted $f^* : \mathbb{R}^N \to \mathbb{R} \cup \{\text{-}\infty, \infty\}$, and is defined as

$$f^*(\boldsymbol{d}) = \sup_{\boldsymbol{x} \,\in\, \mathrm{domain}(f)} \boldsymbol{d}'\boldsymbol{x} - f(\boldsymbol{x}), \tag{A.1}$$

where $\boldsymbol{d} \in \mathbb{R}^N$ is a dual variable. The derivations below use the following two conjugate function relations.

1. When $f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2$ for $\boldsymbol{y} \in \mathbb{R}^N$, the conjugate function is

   $$f^*(\boldsymbol{d}) = \sup_{\boldsymbol{x} \,\in\, \mathbb{R}^N} \boldsymbol{d}'\boldsymbol{x} - \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2.$$

   The maximizer of the quadratic cost function f^* is

   $$\hat{\boldsymbol{x}} = \boldsymbol{y} + \boldsymbol{d} \tag{A.2}$$

   and the maximum value simplifies to

   $$f^*(\boldsymbol{d}) = \frac{1}{2}\|\boldsymbol{d} + \boldsymbol{y}\|^2 - \frac{1}{2}\|\boldsymbol{y}\|^2. \tag{A.3}$$

2. When $\phi(z) = |z|$ is defined on $\mathbb{R}$, the conjugate function is

$$\phi^*(d) = \sup_{z \in \mathbb{R}} dz - |z|.$$

One can verify that the conjugate is

$$\phi^*(d) = \begin{cases} 0 & \text{if } |d| \leq 1 \\ \infty & \text{else} \end{cases} \tag{A.4}$$

and the corresponding sets of suprema are

$$\operatorname*{argmax}_{z \in \mathbb{R}} dz - |z| = \begin{cases} \operatorname{sign}(d) \cdot \infty & \text{if } |d| > 1 \\ 0 & \text{if } |d| < 1 \\ [0, \infty) & \text{if } d = 1 \\ (\text{-}\infty, 0] & \text{if } d = \text{-}1. \end{cases} \tag{A.5}$$

Generalizing (A.4) to a vector, the conjugate function of the 1-norm is a characteristic function that is infinity if any element of the input vector is larger than 1 in absolute value.

Ref. [10, p. 50] provides a table with many more conjugate functions.

The biconjugate, denoted f^{**}, is the conjugate of f^*, *i.e.*,

$$f^{**}(\boldsymbol{x}) = \sup_{\boldsymbol{d} \in \operatorname{domain}(f^*)} \boldsymbol{x}'\boldsymbol{d} - f^*(\boldsymbol{d}), \tag{A.6}$$

and is the largest convex, lower semi-continuous function below f. When f is convex and lower semi-continuous, the biconjugate is equal to the original function, *i.e.*, $f^{**} = f$. One can use the equality of the original function and the biconjugate to derive the saddle point and dual problems when f is convex.

Consider the specific lower-level problem with an analysis-based regularizer

$$\operatorname*{argmin}_{\boldsymbol{x} \in \mathbb{R}^N} \frac{1}{2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2 + \mathbf{1}'\phi_{.}(\boldsymbol{\Omega}\boldsymbol{x}), \tag{A.7}$$

where $\boldsymbol{\Omega} \in \mathbb{R}^{F \times N}$. When ϕ is convex, the corresponding saddle-point problem is

$$\operatorname*{argmin}_{\boldsymbol{x} \in \mathbb{R}^N} \frac{1}{2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2 + \underbrace{\sup_{\boldsymbol{d} \in \mathbb{R}^F} \langle \boldsymbol{d}, \boldsymbol{\Omega}\boldsymbol{x} \rangle - \mathbf{1}'\phi^*_{.}(\boldsymbol{d})}_{\mathbf{1}'\phi^{**}_{.}(\boldsymbol{\Omega}\boldsymbol{x})},$$

where $\langle \cdot, \cdot, \rangle$ is the standard inner product. Under very mild conditions (satisfied for the absolute value function) [19], one can swap the minimum and supremum operations and write the **saddle-point problem** as

$$\sup_{\boldsymbol{d}\,\in\,\mathbb{R}^F}\ \min_{\boldsymbol{x}\,\in\,\mathbb{R}^N} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2 + \langle \boldsymbol{d}, \boldsymbol{\Omega}\boldsymbol{x}\rangle - \mathbf{1}'\phi^*.(\boldsymbol{d}).$$

Substituting the conjugate of the 1-norm (A.4), the saddle-point problem is thus

$$\min_{\boldsymbol{x}\in\mathbb{R}^N}\ \min_{\boldsymbol{d}\,\in\,\mathbb{R}^F} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2 - \langle \boldsymbol{d}, \boldsymbol{\Omega}\boldsymbol{x}\rangle \text{ s.t. } |d_i| \leq 1\ \forall i. \tag{A.8}$$

We hereafter assume $\boldsymbol{A} = \boldsymbol{I}$ to derive the dual problem from the saddle-point problem. By grouping terms and re-arranging negative signs, the dual problem can be derived from the saddle point problem. For a general ϕ, the saddle-point problem is equivalent to

$$\begin{aligned} &\max_{\boldsymbol{d}\,\in\,\mathbb{R}^F} \text{-}\mathbf{1}'\phi^*.(\boldsymbol{d}) + \left(\min_{\boldsymbol{x}\in\mathbb{R}^N} \langle \boldsymbol{d}, \boldsymbol{\Omega}\boldsymbol{x}\rangle + \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2\right) \\ &= \max_{\boldsymbol{d}\,\in\,\mathbb{R}^F} \text{-}\mathbf{1}'\phi^*.(\boldsymbol{d}) - \underbrace{\left(\max_{\boldsymbol{x}\in\mathbb{R}^N} \langle \text{-}\boldsymbol{\Omega}'\boldsymbol{d}, \boldsymbol{x}\rangle - \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2\right)}_{f^*(\text{-}\boldsymbol{\Omega}'\boldsymbol{d})}, \end{aligned}$$

where the last line follows from properties of inner products. The expression in parenthesis is the conjugate function for the data-fit term, given in (A.3). Therefore, the dual problem for a general, convex ϕ is

$$\max_{\boldsymbol{d}\,\in\,\mathbb{R}^F} \text{-}\mathbf{1}'\phi^*.(\boldsymbol{d}) - f^*(\text{-}\boldsymbol{\Omega}'\boldsymbol{d}) = \text{-}\min_{\boldsymbol{d}\,\in\,\mathbb{R}^F} \mathbf{1}'\phi^*.(\boldsymbol{d}) + f^*(\text{-}\boldsymbol{\Omega}'\boldsymbol{d}).$$

Substituting the conjugates for the data-fit term (A.3) and the conjugate for the 1-norm regularizer (A.4), the **dual problem** for (A.7) with $\phi(z) = |z|$ becomes

$$\min_{\boldsymbol{d}\,\in\,\mathbb{R}^F} \frac{1}{2}\left\|\text{-}\boldsymbol{\Omega}'\boldsymbol{d} + \boldsymbol{y}\right\|^2 - \frac{1}{2}\left\|\boldsymbol{y}\right\|^2 \text{ s.t. } |d_i| \leq 1\ \forall i. \tag{A.9}$$

When we require only the minimizer (not the minimum), an equivalent dual problem is

$$\hat{\boldsymbol{d}} = \operatorname*{argmin}_{\boldsymbol{d}\,\in\,\mathbb{R}^F} \frac{1}{2}\left\|\text{-}\boldsymbol{\Omega}'\boldsymbol{d} + \boldsymbol{y}\right\|^2 \text{ s.t. } |d_i| \leq 1\ \forall i. \tag{A.10}$$

This dual problem is a constrained least squares problem and can be solved with a projected gradient descent method, optionally with momentum [104]. From (A.2), the primal minimizer can be recovered from the dual minimizer by

$$\hat{\boldsymbol{x}} = \boldsymbol{y} - \boldsymbol{\Omega}' \hat{\boldsymbol{d}}. \tag{A.11}$$

Finally, from (A.5), the dual variable is related to the filtered signal by

$$d_i \in \begin{cases} 1 & \text{if } [\boldsymbol{\Omega}\hat{\boldsymbol{x}}]_i > 0 \\ \text{-1} & \text{if } [\boldsymbol{\Omega}\hat{\boldsymbol{x}}]_i < 0 \\ [0, \infty) & \text{if } [\boldsymbol{\Omega}\hat{\boldsymbol{x}}]_i = 1 \\ (\text{-}\infty, 0] & \text{if } [\boldsymbol{\Omega}\hat{\boldsymbol{x}}]_i = \text{-1}. \end{cases} \tag{A.12}$$

Ref. [181] provides a more general version of the dual function for non-identity system matrices.

Above, we derived the saddle-point and dual problems using the equality of the biconjugate and the original function for a convex regularizer. The dual problem can also be derived using Lagrangian theory, as shown in [181]. Define an auxiliary (split) variable that is constrained to equal the filtered signal, *i.e.*, $\boldsymbol{z} = \boldsymbol{\Omega x}$. Considering the specific case of the 1-norm regularizer, the Lagrangian of the constrained version of (A.7) is

$$\frac{1}{2} \|\boldsymbol{x} - \boldsymbol{y}\|^2 + \|\boldsymbol{z}\|_1 + \boldsymbol{d}'(\boldsymbol{\Omega x} - \boldsymbol{z}),$$

where $\boldsymbol{d} \in \mathbb{R}^F$ is a vector of Lagrange multipliers and we have omitted the KKT conditions. Minimizing the Lagrangian with respect to $\boldsymbol{x}$ and $\boldsymbol{z}$ yields the conjugate functions for the data-fit term and 1-norm and thus the dual problem.

Using the Lagrangian perspective to derive the dual problem yields a useful relation between the filtered signal and the dual variable [181]. Because the split variable $\boldsymbol{z}$ is constrained to equal $\boldsymbol{\Omega x}$, $[\boldsymbol{\Omega x}]_i > 0$ implies $z_i > 0$. From (A.5), z_i is only positive and finite when $d_i = 1$. A similar argument holds for $[\boldsymbol{\Omega x}]_i < 0$. Therefore, the dual variable and $\hat{\boldsymbol{x}}$ are related by

$$d_i \in \begin{cases} \operatorname{sign}([\boldsymbol{\Omega x}]_i) & \text{if } [\boldsymbol{\Omega}\hat{\boldsymbol{x}}]_i \neq 0 \\ [\text{-1}, 1] & \text{if } [\boldsymbol{\Omega}\hat{\boldsymbol{x}}]_i = 0. \end{cases} \tag{A.13}$$

The second case follows from observing that d_i can take any value in its constrained range when $z_i = 0$ as the minimum in (A.9) will be 0 regardless of d_i.

The primal-dual results reviewed in this appendix are referenced in Section 2.2.3 to relate analysis and synthesis regularizers, Section 4.3 to re-write the lower-level minimizer as a differentiable function of itself and $\boldsymbol{\gamma}$, and in Section 4.4.2 to unroll a differentiable algorithm for a non-smooth cost function.

B

Forward and Reverse Approaches to Unrolling

This appendix provides background on the forward and backward approaches to the unrolled gradient computation introduced in Section 4.4. From (4.18), the gradient of interest is:

$$\nabla \ell(\boldsymbol{\gamma}) = \nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}\,; \boldsymbol{x}^{(T)}) + \left(\sum_{t=1}^{T} \left(\boldsymbol{H}_T \cdots \boldsymbol{H}_{t+1} \right) \boldsymbol{J}_t \right)' \nabla_{\boldsymbol{x}} \ell(\boldsymbol{\gamma}\,; \boldsymbol{x}^{(T)}) \in \mathbb{F}^R. \tag{B.1}$$

If one uses a gradient descent based algorithm to optimize the lower-level cost function Φ, then $\boldsymbol{H}_t = \nabla_{\boldsymbol{x}} \Psi(\boldsymbol{x}^{(t-1)}\,; \boldsymbol{\gamma}) \in \mathbb{F}^{N \times N}$ is closely related to the Hessian of Φ and $\boldsymbol{J}_t = \nabla_{\boldsymbol{\gamma}} \Psi(\boldsymbol{x}^{(t-1)}\,; \boldsymbol{\gamma}) \in \mathbb{F}^{N \times R}$ is proportional to the Jacobian of the gradient.

To compare the forward and reverse approaches to gradient computation for unrolled methods, we introduce notation for an ordered product of matrices. We indicate the arrangement of the multiplications by the set endpoints, $s \in [s_1 \leftrightarrow s_2]$ with the left endpoint, s_1, corresponding to the index for the left-most matrix in the product and the right endpoint, s_2, corresponding to the right-most matrix. Thus, for any sequence of square matrices $\{\boldsymbol{A}\}_i$:

$$\prod_{s\in[t\leftrightarrow T]} \boldsymbol{A}_s := \boldsymbol{A}_t\boldsymbol{A}_{t+1}\cdots\boldsymbol{A}_T = (\boldsymbol{A}_T'\boldsymbol{A}_{T-1}'\cdots\boldsymbol{A}_t')' = \left(\prod_{s\in[T\leftrightarrow t]}\boldsymbol{A}_s'\right)'.$$

The above double arrow notation does not indicate order of operations. In the following notation the arrow direction does not affect the product result (ignoring finite precision effects), but rather signifies the direction (order) of calculation:

$$\prod_{s\in[T\leftarrow t]} \boldsymbol{A}_s := \boldsymbol{A}_T\left(\boldsymbol{A}_{T-1}\cdots\left(\boldsymbol{A}_{t+1}\left(\boldsymbol{A}_t\right)\right)\right)$$

$$\prod_{s\in[T\rightarrow t]} \boldsymbol{A}_s := \left(\left(\left(\boldsymbol{A}_T\boldsymbol{A}_{T-1}\right)\cdots\right)\boldsymbol{A}_{t+1}\right)\boldsymbol{A}_t.$$

We use a similar arrow notation to denote the order that terms are computed for sums; as above, the order is only important for computational considerations and does not affect the final result.

Using this notation, the reverse gradient calculation of (B.1) is

$$\nabla_{\boldsymbol{\gamma}}\ell(\boldsymbol{\gamma}\,;\boldsymbol{x}^{(T)}) + \sum_{t\in[T\rightarrow 1]} \boldsymbol{J}_t{}'\left(\prod_{s\in[(t+1)\leftarrow T]}\boldsymbol{H}_s'\right)\nabla_{\boldsymbol{x}}\ell(\boldsymbol{\gamma}\,;\boldsymbol{x}^{(T)}). \tag{B.2}$$

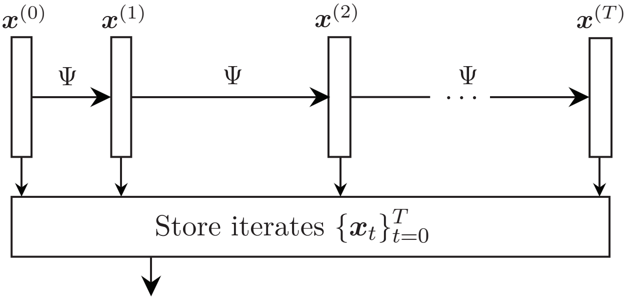
This expression requires $\prod_{s\in[(T+1)\leftarrow T]}\boldsymbol{H}_s' = \boldsymbol{I}$, because $\boldsymbol{H}_{T+1}$ is not defined. For example, for $T=3$, we have

$$\nabla_{\boldsymbol{\gamma}}\ell(\boldsymbol{\gamma}\,;\boldsymbol{x}^{(3)}) + \underbrace{\boldsymbol{J}_3'(\boldsymbol{I})\boldsymbol{g}}_{t=3} + \underbrace{\boldsymbol{J}_2'\left(\boldsymbol{H}_3'\right)\boldsymbol{g}}_{t=2} + \underbrace{\boldsymbol{J}_1'\left(\boldsymbol{H}_2'\boldsymbol{H}_3'\right)\boldsymbol{g}}_{t=1},$$

where $\boldsymbol{g}$ is shorthand for $\nabla_{\boldsymbol{x}}\ell(\boldsymbol{\gamma}\,;\boldsymbol{x}^{(T)})$ here. This version is called reverse as all computations (arrows) begin at the end, T.

The primary benefit of the reverse mode comes from the ability to group $\nabla_{\boldsymbol{x}}\ell(\boldsymbol{\gamma}\,;\boldsymbol{x}^{(T)})$ with the right-most $\boldsymbol{H}_T$, such that all products are matrix-vector products, as seen in Fig. B.1 Further, one can save the matrix-vector products for use during the next iteration and avoid duplicating the computation. Continuing the example for $T=3$, we have

$$\nabla_{\boldsymbol{\gamma}}\ell(\boldsymbol{\gamma}\,;\boldsymbol{x}^{(3)}) + \underbrace{\boldsymbol{J}_3'(\boldsymbol{I})\boldsymbol{g}}_{t=1} + \underbrace{\boldsymbol{J}_2'(\overbrace{\boldsymbol{H}_3'\boldsymbol{g}}^{\boldsymbol{\Delta}})}_{t=2} + \underbrace{\boldsymbol{J}_1'(\boldsymbol{H}_2'\overbrace{(\boldsymbol{H}_3'\boldsymbol{g})}^{\boldsymbol{\Delta}})}_{t=3},$$



```
$\boldsymbol{\Delta} = \nabla_{\boldsymbol{x}} \ell \left( \boldsymbol{\gamma} \; ; \; \boldsymbol{x}^{(T)} \right)$
$\boldsymbol{r} = 0$
For $t = T - 1 \ldots 0$
    Use $\boldsymbol{x}^{(t)}$ to compute $\boldsymbol{J}_{t+1}$ and $\boldsymbol{H}_{t+1}$
    $\boldsymbol{r} = \boldsymbol{r} + \boldsymbol{J}'_{t+1} \boldsymbol{\Delta}$
    $\boldsymbol{\Delta} = \boldsymbol{H}'_{t+1} \boldsymbol{\Delta}$
```

Figure B.1: Reverse mode computation of the unrolled gradient from (B.1). The first gradient computation requires $\boldsymbol{x}^{(T)}$, so all computations occur after the lower-level optimization algorithm is complete. The final gradient is $\nabla \ell(\boldsymbol{\gamma}) = \nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma} \, ; \boldsymbol{x}^{(T)}) + \boldsymbol{r}$.

where one only needs to compute $\boldsymbol{\Delta}$ once. This ability to rearrange the parenthesis to compute matrix-vector products greatly decreases the computational requirement compared to matrix-matrix products. Excluding the costs of the optimization algorithm steps and forming the $\boldsymbol{H}_s$ and $\boldsymbol{J}_t$ matrices (these costs will be the same in the forward mode computation), reverse mode requires $\mathcal{O}(T)$ Hessian-vector multiplies and $\mathcal{O}(TNR)$ additional multiplies. The trade-off is that reverse mode requires storing all T iterates, $\boldsymbol{x}^{(t)}$, so that one can compute the corresponding Hessians and Jacobians from them as needed, and thus has a memory complexity $\mathcal{O}(TN)$.

The forward mode calculation of (B.1), depicted in Fig. B.2, has all computations (arrows) starting at the earlier iterate:

$$\nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma} \, ; \boldsymbol{x}^{(T)}) + \left(\sum_{t \in [1 \to T]} \left(\prod_{s \in [T \leftarrow (t+1)]} \boldsymbol{H}_s \right) \boldsymbol{J}_t \right)' \nabla_{\boldsymbol{x}} \ell(\boldsymbol{\gamma} \, ; \boldsymbol{x}^{(T)}). \quad \text{(B.3)}$$

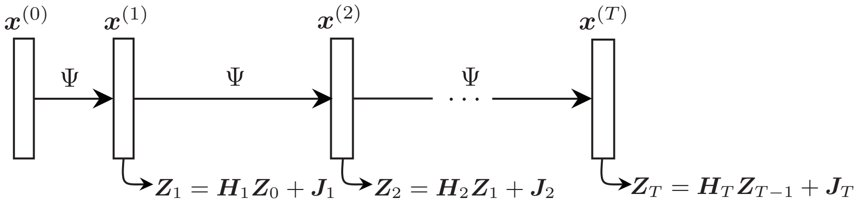

$\boldsymbol{x}^{(0)}$
$\boldsymbol{x}^{(1)}$
$\boldsymbol{x}^{(2)}$
$\boldsymbol{x}^{(T)}$
Ψ
Ψ
Ψ
$\boldsymbol{Z}_1 = \boldsymbol{H}_1 \boldsymbol{Z}_0 + \boldsymbol{J}_1$
$\boldsymbol{Z}_2 = \boldsymbol{H}_2 \boldsymbol{Z}_1 + \boldsymbol{J}_2$
$\boldsymbol{Z}_T = \boldsymbol{H}_T \boldsymbol{Z}_{T-1} + \boldsymbol{J}_T$

Figure B.2: Forward mode computation of the unrolled gradient from (B.3). The intermediate computation matrix, $\boldsymbol{Z}$, is initialized to zero ($\boldsymbol{Z}_0 = \mathbf{0}$) then updated every iteration. The final gradient is $\nabla \ell(\boldsymbol{\gamma}) = \nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}\,; \boldsymbol{x}^{(T)}) + \boldsymbol{Z}_T' \nabla_{\boldsymbol{x}} \ell(\boldsymbol{\gamma}\,; \boldsymbol{x}^{(T)})$.

As before, $\boldsymbol{H}_{T+1}$ is not defined, so we take $\prod_{s \in [T \leftarrow (T+1)]} \boldsymbol{H}_s = \boldsymbol{I}$. For example, for $T = 3$ we have

$$\nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}\,; \boldsymbol{x}^{(T)}) + \left(\underbrace{((\boldsymbol{H}_3 \boldsymbol{H}_2) \boldsymbol{J}_1)'}_{t=1} + \underbrace{((\boldsymbol{H}_3) \boldsymbol{J}_2)'}_{t=2} + \underbrace{((\boldsymbol{I}) \boldsymbol{J}_3)'}_{t=3} \right) \boldsymbol{g}.$$

How the forward mode avoids storing $\boldsymbol{x}$ iterates is evident after rearranging the parenthesis to avoid duplicate calculations, as illustrated in Fig. B.2. Continuing the example for $T = 3$, we have

$$\nabla_{\boldsymbol{\gamma}} \ell(\boldsymbol{\gamma}\,; \boldsymbol{x}^{(T)}) + \left[\underbrace{\boldsymbol{H}_3 \left(\overbrace{\boldsymbol{H}_2 \underbrace{(\boldsymbol{H}_1 \cdot \mathbf{0} + \boldsymbol{J}_1)}_{\boldsymbol{Z}_1} + \boldsymbol{J}_2}^{\boldsymbol{Z}_2} \right) + \boldsymbol{J}_3}_{\boldsymbol{Z}_3} \right]' \boldsymbol{g},$$

where $\boldsymbol{Z}_s = \boldsymbol{H}_s \boldsymbol{Z}_{s-1} + \boldsymbol{J}_s \in \mathbb{F}^{N \times R}$ stores the intermediate calculations. The above formula also illustrates why $\boldsymbol{H}_1$ is not needed in (4.17); $\nabla_{\boldsymbol{\gamma}} \boldsymbol{x}^{(0)} = \mathbf{0}$ is the last element from applying the chain rule.

There is no way to rearrange the terms in the forward mode formula to achieve matrix-vector products (while preserving the computation order). Therefore, the computation requirement is much higher at $\mathcal{O}(TR)$ Hessian-vector multiplications. The corresponding benefit of the forward mode method is that it does not require storing iterates, thus decreasing (in the common case when $T > R$) the memory requirement to $\mathcal{O}(NR)$ for storing the intermediate matrix $\boldsymbol{Z}_s$ during calculation.

As with the minimizer approach in Section 4.2, the computational complexity of the unrolled approach is lower than the generic bound when we consider the specific example of learning convolutional filters according to (Ex). Nevertheless, the general comparison that reverse mode takes more memory but less computation holds true. See Tab. 4.1 for a comparison of the computational and memory complexities.

C

Additional Running Example Results

This appendix derives some results that are relevant to the running example used throughout the survey.

C.1 Derivatives for Convolutional Filters

This section proves the result

$$\frac{\partial}{\partial c_{\boldsymbol{s}}} \left(\tilde{\boldsymbol{c}}_k \circledast f.(\boldsymbol{c}_k \circledast \boldsymbol{x})\right) = f.(\boldsymbol{c}_k \circledast \boldsymbol{z}^{\langle \boldsymbol{s} \rangle}) + \tilde{\boldsymbol{c}}_k \circledast \left(\dot{f}.(\boldsymbol{c}_k \circledast \boldsymbol{x}) \odot \boldsymbol{x}^{\langle -\boldsymbol{s} \rangle}\right), \tag{C.1}$$

when considering $\mathbb{F} = \mathbb{R}$. This equation is key to finding derivatives of the lower-level cost function in (Ex) with respect to the filter coefficients.

To simplify notation, we drop the indexing over k, so $\boldsymbol{c}$ is a single filter and $c_{\boldsymbol{s}}$ denotes the $\boldsymbol{s}$th element in the filter for $\boldsymbol{s} \in \mathbb{Z}^D$. Here, $\boldsymbol{s}$ indexes every dimension of $\boldsymbol{c}$, *e.g.*, for a two-dimensional filter, we could equivalently write $\boldsymbol{s}$ as $\langle s_1, s_2 \rangle$. Recall that the notation $\tilde{\boldsymbol{c}}$ signifies a reversed version of $\boldsymbol{c}$, as needed for the adjoint of convolution.

Define the notation $\boldsymbol{x}^{\langle \boldsymbol{i} \rangle}$ as the vector $\boldsymbol{x}$ circularly shifted according to the index $\boldsymbol{i}$. Thus, if $\boldsymbol{x}$ is 0-indexed and we use circular indexing,

$$(\boldsymbol{x}^{\langle \boldsymbol{s} \rangle})_{\boldsymbol{i}} = \boldsymbol{x}_{\boldsymbol{i}-\boldsymbol{s}}.$$

As two examples,

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix} \to \boldsymbol{x}^{\langle\text{-}1\rangle} = \begin{bmatrix} x_2 \\ x_3 \\ \vdots \\ x_N \\ x_1 \end{bmatrix},$$

and, in two dimensions, if $\boldsymbol{X} \in \mathbb{F}^{M\times N}$

$$\boldsymbol{X}^{\langle 1,2\rangle} = \begin{bmatrix} x_{M,N-1} & x_{M,N} & x_{M,1} & \cdots & x_{M,3} \\ x_{1,N-1} & x_{1,N} & x_{1,1} & \cdots & x_{1,3} \\ x_{2,N-1} & x_{2,N} & x_{2,1} & \cdots & x_{2,3} \\ \vdots & & \ddots & & \vdots \\ x_{M-1,N-1} & x_{M-1,N} & x_{M-1,1} & \cdots & x_{M-1,3} \end{bmatrix}.$$

This circular shift notation is useful in the derivation and statement of the desired gradient.

Define $\boldsymbol{z} = \boldsymbol{c} \circledast \boldsymbol{x}$, where $\boldsymbol{c}$ and $\boldsymbol{x}$ are both N-dimensional. By the definition of convolution, $\boldsymbol{z}$ is given by

$$\boldsymbol{z} = \sum_{i_1} \cdots \sum_{i_N} c_{i_1,\ldots,i_N} \boldsymbol{x}^{\langle\text{-}i_1,\ldots,\text{-}i_N\rangle} := \sum_{i_1,\ldots,i_N} c_{i_1,\ldots,i_N} \boldsymbol{x}^{\langle\text{-}\boldsymbol{i}\rangle},$$

where, for each sum, the indexing variable i_n iterates over the size of $\boldsymbol{c}$ in the ith dimension and we simplify the index for circularly shifting vectors, $i_1, \ldots, i_N$, as simply $\langle \boldsymbol{i} \rangle$. This expression shows that the derivative of $\boldsymbol{c} \circledast \boldsymbol{x}$ with respect to the $\boldsymbol{s}$th filter coefficient is the -$\boldsymbol{s}$th coefficient in $\boldsymbol{x}$, *i.e.*,

$$\frac{\partial}{\partial c_{\boldsymbol{s}}}(\boldsymbol{c} \circledast \boldsymbol{x}) = \boldsymbol{x}^{\langle -\boldsymbol{s}\rangle}. \tag{C.2}$$

We can now find the partial derivative of interest:

$$\begin{aligned}
\tilde{\boldsymbol{c}} \circledast f.(\boldsymbol{z}) &= \sum_{i_1,\dots,i_N} [\tilde{\boldsymbol{c}}]_{i_1,\dots,i_N} f.(\boldsymbol{z})^{\langle \text{-}\boldsymbol{i}\rangle} && \text{by the convolution formula}\\
&= \sum_{i_1,\dots,i_N} [\tilde{\boldsymbol{c}}]_{i_1,\dots,i_N} f.\left(\boldsymbol{z}^{\langle \text{-}\boldsymbol{i}\rangle}\right) && \text{since } f \text{ operates point-wise}\\
&= \sum_{i_1,\dots,i_N} c_{\text{-}i_1,\dots,\text{-}i_N} f.\left(\boldsymbol{z}^{\langle \text{-}\boldsymbol{i}\rangle}\right) && \text{by definition of } \tilde{\boldsymbol{c}}\\
&= \sum_{i_1,\dots,i_N} c_{i_1,\dots,i_N} f.\left(\boldsymbol{z}^{\langle \boldsymbol{i}\rangle}\right) && \text{reverse summation order.}
\end{aligned}$$

Recall that $\boldsymbol{z}$ is a function of $c_{\boldsymbol{s}}$. Therefore, using the chain rule to take the derivative,

$$\begin{aligned}
&\frac{\partial}{\partial c_{\boldsymbol{s}}} \left(\tilde{\boldsymbol{c}} \circledast f.(\boldsymbol{z})\right)\\
&\quad = f.(\boldsymbol{z}^{\langle s\rangle}) + \sum_{i_1} \cdots \sum_{i_N} c_{i_1,\dots,i_N} \dot{f}.(\boldsymbol{z}^{\langle i_1,\dots,i_N\rangle}) \odot \nabla_{c_{\boldsymbol{s}}} \left(\boldsymbol{z}^{\langle \boldsymbol{i}\rangle}\right)\\
&\quad = f.(\boldsymbol{z}^{\langle s\rangle}) + \sum_{i_1} \cdots \sum_{i_N} [\tilde{\boldsymbol{c}}]_{\text{-}i_1,\dots,\text{-}i_N} \dot{f}.(\boldsymbol{z}^{\langle i_1,\dots,i_N\rangle}) \odot \boldsymbol{x}^{\langle \boldsymbol{i}-\boldsymbol{s}\rangle},
\end{aligned}$$

where the second equality follows from (C.2) and the definition of $\tilde{\boldsymbol{c}}$. Recognizing the convolution formula in the second summand, the expression can be simplified to

$$f.(\boldsymbol{z}^{\langle \boldsymbol{s}\rangle}) + \tilde{\boldsymbol{c}} \circledast \left(\dot{f}.(\boldsymbol{z}) \odot \boldsymbol{x}^{\langle -\boldsymbol{s}\rangle}\right).$$

This proves the claim. Note that the provided formula is for a single element in $\boldsymbol{c}$. One can concatenate the partial derivative result for each value of $\boldsymbol{s}$ to get the full Jacobian.

C.2 Evaluating Assumptions for the Running Example

To better understand the upper-level assumptions Aℓ 1-Aℓ 3 and lower-level assumptions AΦ 1-AΦ 6 in Section 5.3.1, this section examines whether the filter learning example (Ex) meets each assumption.

C.2.1 Upper-level Loss Assumptions

Recall the upper-level loss function in (Ex) is squared error:

$$\ell(\boldsymbol{\gamma}\,;\,\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{x}^{\text{true}}\|_2^2, \tag{C.3}$$

where ℓ is typically evaluated at $\boldsymbol{x} = \hat{\boldsymbol{x}}(\boldsymbol{\gamma})$.

The loss function (C.3) satisfies Aℓ1. Because there is no dependence on $\boldsymbol{\gamma}$ in the upper-level, $L_{\boldsymbol{x},\nabla_{\boldsymbol{\gamma}}\ell} = 0$. The gradient with respect to $\boldsymbol{x}$ is $\nabla_{\boldsymbol{x}}\ell(\boldsymbol{\gamma}\,;\,\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{x}^{\text{true}}$, so $L_{\boldsymbol{x},\nabla_{\boldsymbol{x}}\ell} = 1$.

The norm of the upper-level gradient with respect to $\boldsymbol{x}$,

$$\|\nabla_{\boldsymbol{x}}\ell(\boldsymbol{\gamma}\,;\,\boldsymbol{x})\| = \left\|\boldsymbol{x} - \boldsymbol{x}^{\text{true}}\right\|,$$

can grow arbitrarily large, so condition Aℓ 2 is not met in general. However, in most applications, one can assume an upper bound (possibly quite large) on the elements of $\boldsymbol{x}^{\text{true}}$ and impose that bound as a box constraint when computing $\hat{\boldsymbol{x}}$. Then the triangle inequality provides a bound on $\|\boldsymbol{x} - \boldsymbol{x}^{\text{true}}\|$ for all $\boldsymbol{x}$ within the constraint box.

Finally, Aℓ3 is met by any loss function, including (C.3), that lacks cross terms between $\boldsymbol{x}$ and $\boldsymbol{\gamma}$. We are unaware of any bilevel method papers using such cross terms.

C.2.2 Lower-level Cost Assumptions

One property used below in many of the bounds for the lower-level cost function is that

$$\sigma_1(\boldsymbol{C}_k) = \|\boldsymbol{c}_k\|_1\,, \tag{C.4}$$

where $\sigma_1(\cdot)$ is a function that returns the first singular value of its matrix argument. This property follows from Young's inequality and is related to bounded-input bounded-output stability of linear and time invariant systems [182].

As with the upper-level assumptions considered above, (Ex) meets the lower-level assumptions AΦ1-AΦ6 if we impose additional constraints on the maximum norm of variables. In addition to bounding the elements in $\boldsymbol{x}$, as we did to ensure Aℓ2, imposing bounds on $\|\boldsymbol{c}_k\|$ and $|\beta_k|$ is sufficient to meet all the lower-level assumptions. We now examine each condition individually.

Recall from (Ex) that the example lower-level cost function is

$$\hat{\boldsymbol{x}}(\boldsymbol{\gamma}) = \underset{\boldsymbol{x}\in\mathbb{F}^N}{\operatorname{argmin}} \frac{1}{2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + e^{\beta_0} \sum_{k=1}^{K} e^{\beta_k} \boldsymbol{1}' \phi.(\boldsymbol{c}_k \circledast \boldsymbol{x}; \epsilon),$$

where ϕ is a corner-rounded 1-norm (CR1N).

As described in Section 4.2, the minimizer approach requires Φ to be twice differentiable. Thus, Φ satisfies AΦ1. This condition limits the choices of ϕ to twice differentiable functions.

Considering AΦ2, the gradient of Φ with respect to $\boldsymbol{x}$ is Lipschitz continuous in $\boldsymbol{x}$ if the norm of the Hessian, $\|\nabla_{\boldsymbol{x}\boldsymbol{x}}\Phi(\boldsymbol{x}\,; \boldsymbol{\gamma})\|_2$, is bounded. Using (4.9) and assuming the Lipschitz constant of the derivative of ϕ is $L_{\dot{\phi}}$ (for (CR1N), $L_{\dot{\phi}} = \frac{1}{\epsilon}$), a Lipschitz constant for $\nabla_{\boldsymbol{x}}\Phi$ is

$$\begin{aligned} L_{\boldsymbol{x},\nabla_{\boldsymbol{x}}\Phi} &= \sigma_1^2(\boldsymbol{A}) + L_{\dot{\phi}} e^{\beta_0} \sum_k e^{\beta_k} \sigma_1(\boldsymbol{C}_k' \boldsymbol{C}_k) \\ &= \sigma_1^2(\boldsymbol{A}) + L_{\dot{\phi}} e^{\beta_0} \sum_k e^{\beta_k} \|\boldsymbol{c}_k\|_1^2 \text{ by (C.4).} \end{aligned} \tag{C.5}$$

The Lipschitz constant $L_{\boldsymbol{x},\nabla_{\boldsymbol{x}}\Phi}$ depends on the values in $\boldsymbol{\gamma}$ and therefore does not strictly satisfy AΦ2. Here if β_0, β_k, and $\boldsymbol{c}_k$ have upper bounds, then one can upper bound $L_{\boldsymbol{x},\nabla_{\boldsymbol{x}}\Phi}$. All of the bounds below have similar considerations.

To consider the strong convexity condition in AΦ3, we consider the Hessian,

$$\nabla_{\boldsymbol{x}\boldsymbol{x}}\Phi(\boldsymbol{x}\,; \boldsymbol{\gamma}) = \underbrace{\boldsymbol{A}'\boldsymbol{A}}_{\text{From data-fit term}} + \underbrace{e^{\beta_0} \sum_k e^{\beta_k} \boldsymbol{C}_k' \operatorname{diag}(\ddot{\phi}.(\boldsymbol{c}_k \circledast \boldsymbol{x}))\boldsymbol{C}_k}_{\text{From regularizer}}. \tag{C.6}$$

We assume that $\ddot{\phi}(z) \geq 0\,\forall z$, as is the case for the corner rounded 1-norm. If $\boldsymbol{A}'\boldsymbol{A}$ is positive-definite with $\sigma_N(\boldsymbol{A}'\boldsymbol{A}) > 0$ (this is equivalent to $\boldsymbol{A}$ having full column rank), then the Hessian is positive-definite and $\mu_{\boldsymbol{x},\Phi} = \sigma_N^2(\boldsymbol{A})$ suffices as a strong convexity parameter. In applications like compressed sensing, $\boldsymbol{A}$ does not have full column rank. In such cases, $\sigma_N(\boldsymbol{A}'\boldsymbol{A}) = 0$ and as $e^{\beta_0} \to 0$ the regularizer term vanishes, so there does not exist any universal $\mu_{\boldsymbol{x},\Phi} > 0$ for all $\boldsymbol{\gamma} \in \mathbb{F}^R$, so the strong convexity condition AΦ3 is not satisfied. However, as discussed in Section 4.2.3, the condition may hold in practice for many values of $\boldsymbol{\gamma}$. How to adapt the complexity theory to rigorously address these subtleties is an open question.

The fourth condition, AΦ4, is that $\nabla_{\boldsymbol{x}\boldsymbol{x}}\Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})$ and $\nabla_{\boldsymbol{\gamma}\boldsymbol{x}}\Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})$ are Lipschitz continuous with respect to $\boldsymbol{x}$ for all $\boldsymbol{\gamma}$. For the first part part, a Lipschitz constant results from bounding the difference in the Hessian evaluated at two points, $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$:

$$\begin{aligned}&\left\|\nabla_{\boldsymbol{x}\boldsymbol{x}}\Phi(\boldsymbol{x}^{(1)}\,;\boldsymbol{\gamma}) - \nabla_{\boldsymbol{x}\boldsymbol{x}}\Phi(\boldsymbol{x}^{(2)}\,;\boldsymbol{\gamma})\right\|_2 \\ &\qquad = \left\| e^{\beta_0}\sum_k e^{\beta_k}\boldsymbol{C}_k'\operatorname{diag}(\ddot{\phi}.(\boldsymbol{c}_k \circledast \boldsymbol{x}^{(1)}) - \ddot{\phi}(\boldsymbol{c}_k \circledast \boldsymbol{x}^{(2)}))\boldsymbol{C}_k \right\|_2 .\end{aligned}$$

Since every element of $\ddot{\phi}$ is bounded in $(0, L_{\dot{\phi}})$, the difference between any two evaluations of $\ddot{\phi}$ is at most $L_{\dot{\phi}}$. Thus

$$\begin{aligned}\left\|\nabla_{\boldsymbol{x}\boldsymbol{x}}\Phi(\boldsymbol{x}^{(1)}\,;\boldsymbol{\gamma}) - \nabla_{\boldsymbol{x}\boldsymbol{x}}\Phi(\boldsymbol{x}^{(2)}\,;\boldsymbol{\gamma})\right\|_2 &\le e^{\beta_0}L_{\dot{\phi}}\sum_k e^{\beta_k}\left\|\boldsymbol{C}_k'\boldsymbol{C}_k\right\|_2 \\ &\le e^{\beta_0}L_{\dot{\phi}}\sum_k e^{\beta_k}\left\|\boldsymbol{c}_k\right\|_1^2 .\end{aligned}$$

The final simplification again uses (C.4). Thus,

$$L_{\boldsymbol{x},\nabla_{\boldsymbol{x}\boldsymbol{x}}\Phi} = e^{\beta_0}L_{\dot{\phi}}\sum_k e^{\beta_k}\left\|\boldsymbol{c}_k\right\|_1^2 .$$

For the second part of AΦ4, we must look at the tuning parameters and filter coefficients separately. When considering learning a tuning parameter, β_k,

$$\nabla_{\beta_k\boldsymbol{x}}\Phi(\boldsymbol{x}\,;\boldsymbol{\gamma}) = e^{\beta_0+\beta_k}\boldsymbol{C}_k'\dot{\phi}.(\boldsymbol{C}_k\boldsymbol{x}).$$

To find a Lipschitz constant, consider the Jacobian:

$$\nabla_{\boldsymbol{x}}\left(\nabla_{\beta_k\boldsymbol{x}}\Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})\right) = e^{\beta_0+\beta_k}\boldsymbol{C}_k'\operatorname{diag}(\ddot{\phi}.(\boldsymbol{C}_k\boldsymbol{x}))\boldsymbol{C}_k.$$

A Lipschitz constant of $\nabla_{\beta_k\boldsymbol{x}}\Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})$ is given by the bound on the norm of this matrix (we chose to use the matrix 2-norm, also called the spectral norm). Using similar steps as above to simplify the expression, $L_{\boldsymbol{x},\nabla_{\beta_k\boldsymbol{x}}\Phi} = e^{\beta_0+\beta_k}L_{\dot{\phi}}\left\|\boldsymbol{c}_k\right\|_1^2$.

When considering learning the $\boldsymbol{s}$th element of the kth filter,

$$\begin{aligned}\nabla_{c_{k,\boldsymbol{s}}\boldsymbol{x}}\Phi(\boldsymbol{x}\,;\boldsymbol{\gamma}) &= e^{\beta_0+\beta_k}\left(\dot{\phi}.((\boldsymbol{C}_k\boldsymbol{x})^{\langle s\rangle}) + \boldsymbol{C}_k'\left(\ddot{\phi}.(\boldsymbol{C}_k\boldsymbol{x})\odot\boldsymbol{x}^{\langle\text{-}s\rangle}\right)\right)\\ &= e^{\beta_0+\beta_k}\left(\underbrace{\dot{\phi}.(\boldsymbol{R}_1\boldsymbol{C}_k\boldsymbol{x})}_{\text{Expression 1}} + \underbrace{\boldsymbol{C}_k'\left(\ddot{\phi}.(\boldsymbol{C}_k\boldsymbol{x})\odot\boldsymbol{R}_2\boldsymbol{x}\right)}_{\text{Expressions 2-3}}\right)\in\mathbb{F}^N,\end{aligned}$$

where $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ are rotation matrices that depends on $\boldsymbol{s}$ such that $\boldsymbol{R}_1\boldsymbol{x} = \boldsymbol{x}^{\langle s\rangle}$ and $\boldsymbol{R}_2\boldsymbol{x} = \boldsymbol{x}^{\langle\text{-}s\rangle}$. For taking the gradient, it is convenient to note that the last term can be expressed in multiple ways:

$$\ddot{\phi}.(\boldsymbol{C}_k\boldsymbol{x})\odot\boldsymbol{x}^{\langle\text{-}s\rangle} = \underbrace{\operatorname{diag}(\ddot{\phi}.(\boldsymbol{C}_k\boldsymbol{x}))\boldsymbol{R}_2\boldsymbol{x}}_{\text{Expression 2}} = \underbrace{\operatorname{diag}(\boldsymbol{R}_2\boldsymbol{x})\ddot{\phi}.(\boldsymbol{C}_k\boldsymbol{x})}_{\text{Expression 3}}.$$

Using the alternate expressions to perform the chain rule with respect to the $\boldsymbol{x}$ term that is not in the diag$(\cdot)$ statement, the gradient with respect to $\boldsymbol{x}$ is:

$$\begin{aligned}\nabla_{\boldsymbol{x}}\left(\nabla_{c_{k,s}\boldsymbol{x}}\Phi(\boldsymbol{x}\,;\boldsymbol{\gamma})\right) = e^{\beta_0+\beta_k}(&\underbrace{\boldsymbol{C}_k'\boldsymbol{R}_1'\operatorname{diag}(\ddot{\phi}.(\boldsymbol{R}_1\boldsymbol{C}_k\boldsymbol{x}))}_{\text{Expression 1}}\\ &+\underbrace{\boldsymbol{C}_k'\operatorname{diag}(\ddot{\phi}.(\boldsymbol{C}_k\boldsymbol{x}))\boldsymbol{R}_2}_{\text{Expression 2}}\\ &+\underbrace{\boldsymbol{C}_k'\operatorname{diag}(\dddot{\phi}(\boldsymbol{C}_k\boldsymbol{x}))\operatorname{diag}(\boldsymbol{R}_2\boldsymbol{x})'\boldsymbol{C}_k}_{\text{Expression 3}}).\end{aligned}$$

The bound on the spectral norm of the first and second expressions are both $\sigma_1(\boldsymbol{C}_k)L_{\dot{\phi}}$ because, for any $\boldsymbol{z}\in\mathbb{F}^N$,

$$\|\operatorname{diag}(\ddot{\phi}.(\boldsymbol{z}))\|_2 \le \max_z|\ddot{\phi}(z)| = L_{\dot{\phi}}.$$

The third expression is bounded by $\sigma_1^2(\boldsymbol{C}_k)\,\|\boldsymbol{x}\|_2\,L_{\ddot{\phi}}$, which requires a bound on the norm of $\boldsymbol{x}$, similar to Aℓ2. Summing the three expressions and including the tuning parameters gives the final Lipschitz constant

$$L_{\boldsymbol{x},\nabla_{c_{k,s}\boldsymbol{x}}\Phi} = e^{\beta_0+\beta_k}\sigma_1(\boldsymbol{C}_k)(2L_{\dot{\phi}} + \sigma_1(\boldsymbol{C}_k)L_{\ddot{\phi}}\,\|\boldsymbol{x}\|_2). \qquad \text{(C.7)}$$

The fifth assumption, AΦ5 states that the mixed second gradient of Φ is bounded. For the tuning parameters, the mixed second gradient is given in (4.9) as

$$\nabla_{\beta_k\boldsymbol{x}}\Phi(\hat{\boldsymbol{x}}\,;\boldsymbol{\gamma}) = e^{\beta_0}e^{\beta_k}\tilde{\boldsymbol{c}}_k\circledast\dot{\phi}.(\boldsymbol{c}_k\circledast\hat{\boldsymbol{x}}).$$

The bound given in AΦ5 follows easily by considering that

$$\|\mathrm{diag}(\dot{\phi}.(\boldsymbol{c}_k \circledast \hat{\boldsymbol{x}}))\|_2 \leq \max_z |\dot{\phi}(z)| = L_\phi.$$

For a filter coefficient, the mixed second gradient is more complicated:

$$\nabla_{c_{k,\boldsymbol{s}}\boldsymbol{x}} \Phi(\hat{\boldsymbol{x}}\,;\boldsymbol{\gamma}) = e^{\beta_0+\beta_k}\Big(\underbrace{\dot{\phi}.((\boldsymbol{c}_k \circledast \hat{\boldsymbol{x}})^{\langle \boldsymbol{s}\rangle})}_{\text{Bounded by } L_\phi} + \tilde{\boldsymbol{c}}_k \circledast \Big(\underbrace{\ddot{\phi}.(\boldsymbol{c}_k \circledast \hat{\boldsymbol{x}})}_{\text{Bounded by } L_{\dot{\phi}}} \odot \hat{\boldsymbol{x}}^{\langle \text{-}\boldsymbol{s}\rangle}\Big)\Big).$$

Assuming that the bounds L_ϕ and $L_{\dot{\phi}}$ exist (they are 1 and $\frac{1}{\epsilon}$ respectively for (CR1N)), a bound on the norm of the mixed gradient is

$$\|\nabla_{c_{k,\boldsymbol{s}}\boldsymbol{x}} \Phi(\hat{\boldsymbol{x}}\,;\boldsymbol{\gamma})\|_2 \leq e^{\beta_0+\beta_k}\left(L_\phi + L_{\dot{\phi}} \left\|\boldsymbol{c}_k\right\|_1 \left\|\boldsymbol{x}\right\|_2\right).$$

The sixth assumption, AΦ6, is that $L_{\boldsymbol{\gamma},\nabla_{\boldsymbol{\gamma}\boldsymbol{x}}\Phi}$ and $L_{\boldsymbol{\gamma},\nabla_{\boldsymbol{x}\boldsymbol{x}}\Phi}$ exist. Lipschitz constants for the tuning parameters are

$$L_{\beta_k,\nabla_{\beta_k \boldsymbol{x}}\Phi} = e^{\beta_0+\beta_k} \left\|\boldsymbol{c}_k\right\|_1 L_\phi \text{ and } L_{\beta_k,\nabla_{\boldsymbol{x}\boldsymbol{x}}\Phi} = e^{\beta_0+\beta_k} \left\|\boldsymbol{c}_k\right\|_1^2 L_{\dot{\phi}}.$$

Using similar derivations as shown above, corresponding Lipschitz constants for the filter coefficients are

$$\begin{aligned} L_{c_{k,s},\nabla_{c_{k,s}\boldsymbol{x}}\Phi} &= e^{\beta_0+\beta_k}\left(L_\phi + \left\|\boldsymbol{x}\right\|_2 \left(L_{\dot{\phi}} + L_{\ddot{\phi}} \left\|\boldsymbol{c}_k\right\|_1 \left\|\boldsymbol{x}\right\|_2\right)\right) \\ L_{c_{k,s},\nabla_{\boldsymbol{x}\boldsymbol{x}}\Phi} &= e^{\beta_0+\beta_k}\left(2L_{\dot{\phi}} \left\|\boldsymbol{c}_k\right\|_1 + L_{\ddot{\phi}} \left\|\boldsymbol{c}_k\right\|_1^2 \left\|\boldsymbol{x}\right\|_2\right). \end{aligned}$$

This is the last lower-level condition in Section 5.3.1 for the single-loop and double-loop bilevel optimization method analysis.

D

Implementation Details

This appendix describes the experimental settings used throughout this review. We first present the common settings; the following sub-sections detail any differences specifically for the results in Fig. 1.3 and for the series of figures using the cameraman image (Fig. 5.2, Fig. 6.1, and Fig. 6.2). The code for all experiments is available on github [32].

The experiments consider the denoising problem ($\boldsymbol{A} = \boldsymbol{I}$) and use (CR1N) as the sparsifying function ϕ with $\epsilon = 0.01$. The training data is typically on the scale [0, 1] and noisy samples are generated from the clean training data using (1.4) with zero-mean Gaussian noise with a standard deviation of $\sigma = 25/255$, following [25].

The lower-level optimizer is the optimized gradient method (OGM) with gradient-based restart [104]. We calculate the step-size based on the Lipschitz constant of the lower-level gradient using (C.5) every upper-level iteration. Each experiment sets a maximum number of lower-level iterations, but the lower-level optimization will terminate early if it converges, defined as if $\|\nabla_{\boldsymbol{x}} \Phi(\boldsymbol{x}\,; \boldsymbol{\gamma})\| < 10^{-5}$.

The upper-level optimizer follows the general structure of the double-loop procedure outlined in Alg. 3. To compute $\nabla \ell(\boldsymbol{\gamma})$, we use the minimizer formulation (4.8), with the conjugate gradient (CG) method to compute the Hessian-inverse-vector product (4.10). As suggested in [98], the initialization for the lower-level optimization is the estimated minimizer from the previous outer loop iteration, $\boldsymbol{x}^{(T)}(\boldsymbol{\gamma}^{(u\text{-}1)})$ and the initialization for the CG method is the solution from the previous CG iteration. Following [26] and other bilevel works, the experiments use Adam with the default parameters [107] to determine the size of the upper-level gradient descent; this choice avoids introducing the tuning parameter α_ℓ.

The learnable parameters include the filter coefficients and the tuning parameters β_k for $k \in [1, K]$. The experiments either use random or DCT filters to initialize $\boldsymbol{h}$. An initial grid search determines the tuning parameter β_0; β_k for $k \in [1, K]$ are initialized as 0 such that $e^{\beta_k} = 1$.

D.1 Vertical Bar Training Image

This section describes additional details for Fig. 1.3. This simple proof of concept used 50 lower-level iterations ($T = 50$) and 4,000 upper-level iterations ($U = 4,000$). The initial grid search for β_0 yielded -4.6.

When $\phi(z) = |z|$, one can absorb the kth filter's magnitude into the tuning parameter β_k because $\|\boldsymbol{c}_k \circledast \boldsymbol{x}\|_1 = \|\boldsymbol{c}_k\|_2 \left\| \frac{1}{\|\boldsymbol{c}_k\|_2} \boldsymbol{c}_k \circledast \boldsymbol{x} \right\|_1$. When using (CR1N), this equality no longer holds, but

$$e^{\beta_0 + \beta_k} \|\boldsymbol{c}_k\|_2 \tag{D.1}$$

still provides a reasonable approximation for the overall regularization strength for the kth filter. From left to right, the approximate regularization strengths of the filters in Fig. 1.3 are 0.77, 0.49, 0.17, and 0.05.

The learned filters reflect that the training data is constant along the columns. Visually, the filters resemble vertical (extended) finite differences. This matches our expectations as a filter that takes vertical finite differences will exactly sparsify the noiseless signal. Further, the maximum sum of the columns of the learned filters is $10^{\text{-}5}$. In contrast, the sum of the rows of the learned filters varies from -2.6 to 3.0.

D.2 Cameraman Training Image

This section describes the experimental settings for Fig. 5.2, Fig. 6.2, and Fig. 6.1.

To reduce computation, we selected three 50×50 patches from the "cameraman" image in Fig. 6.2 to use as the training data. We hand selected the training patches to contain structure. Fig. D.1 shows the training image patches.

We set the lower-level initialization $\hat{\boldsymbol{x}}(\boldsymbol{\gamma}^{(0)})$ by optimizing the lower-level cost function until the norm of the gradient fell below a threshold for each training patch, *i.e.*, until $\frac{1}{\sqrt{N}} \left\| \nabla_{\boldsymbol{x}} \Phi \left(\hat{\boldsymbol{x}}_j(\boldsymbol{\gamma}^{(0)})\,;\, \boldsymbol{\gamma}^{(0)} \right) \right\|_2 < 10^{\text{-}7}$ for $j \in [1, J]$. The lower-level optimizer consisted of 10 iterations of OGM [104].

As shown in Fig. 6.1, the initial filters are the 48 non-constant DCT filters of size 7×7. The initial grid search for β_0 yielded -4. In summary, the settings are $J = 3$, $N = 50 \cdot 50$, $S = 7 \cdot 7$, $K = 48$, $R = 48(49 + 1) = 2400$, $\beta_0 = \text{-}4$, $T = 10$, and $U = 10{,}000$.

Fig. 6.1 shows the learned filters. To visualize the filters when $\boldsymbol{\gamma}$ includes $\boldsymbol{h}$, Fig. 6.1c scales each learned filter $\hat{\boldsymbol{c}}_k$ to have unit norm. Fig. D.2 shows the learned filters with the effective regularization strength printed above each filter.

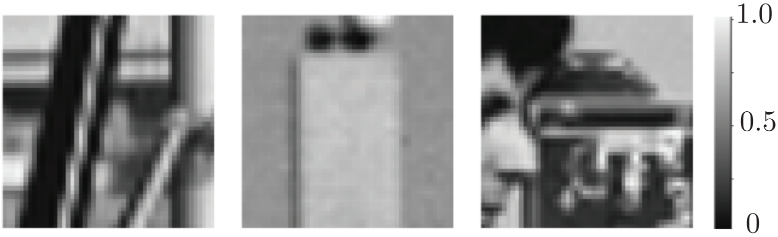


Figure D.1: Patches from the cameraman test images used as the training dataset.

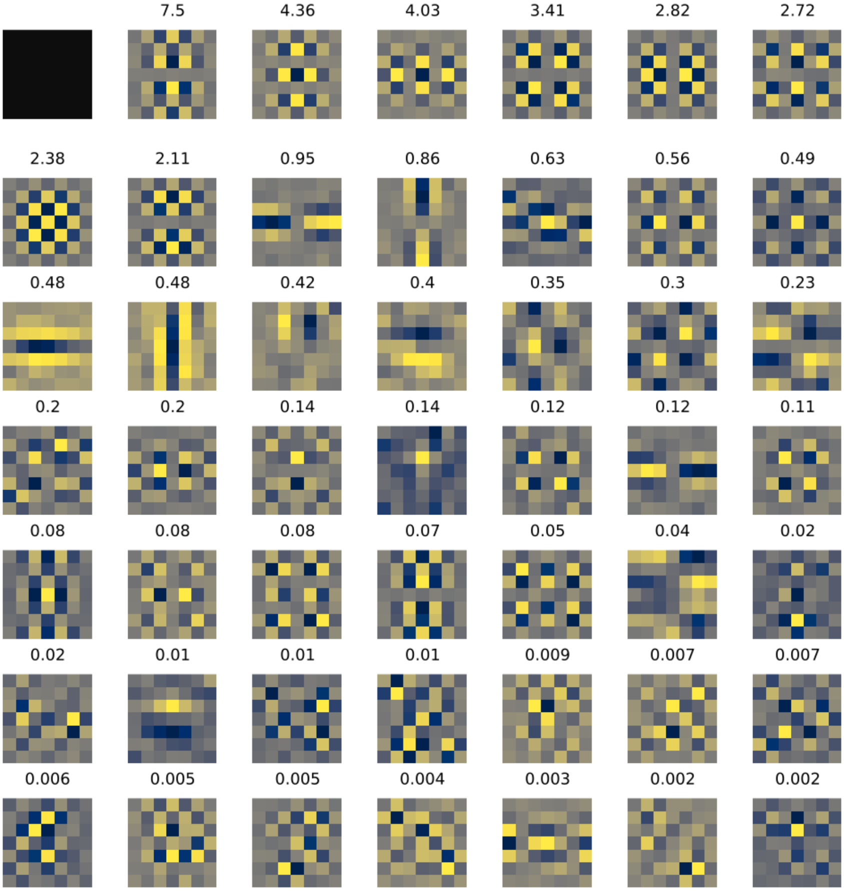


Figure D.2: Learned filers for (Ex) when $\boldsymbol{\gamma}$ includes $\boldsymbol{h}$ and $\boldsymbol{\beta}$, ordered by their effective regularization strength $e^{\beta_k} \|\boldsymbol{c}_k\|_2$, which is printed above each filter. This effective regularization does not include the influence of e^{β_0}, which is uniform across all filters.

References

[1] B. M. Afkham, J. Chung, and M. Chung, "Learning regularization parameters of inverse problems via deep neural networks," *Inverse Problems*, vol. 37, no. 10, Sep. 2021, p. 105 017. DOI: 10.1088/1361-6420/ac245d.

[2] H. Antil, Z. Di, and R. Khatri, "Bilevel optimization, deep learning and fractional laplacian regularization with applications in tomography," *Inverse Problems*, Mar. 18, 2020. DOI: 10.1088/1361-6420/ab80d7.

[3] S. Bai, J. Z. Kolter, and V. Koltun, "Deep equilibrium models," in *Advances in Neural Information Processing Systems*, vol. 32, Curran Associates, Inc., 2019, URL: https://proceedings.neurips.cc/paper/2019/hash/01386bd6d8e091c2ab4c7c7de644d37b-Abstract.html.

[4] H. H. Barrett, "Objective assessment of image quality: Effects of quantum noise and object variability," *J. Opt. Soc. Am. A*, vol. 7, no. 7, Jul. 1990, 1266–1278. DOI: 10.1364/JOSAA.7.001266.

[5] M. Benning, C. Brune, M. Burger, and J. Müller, "Higher-order TV methods—Enhancement via Bregman iteration," *Journal of Scientific Computing*, vol. 54, no. 2-3, Feb. 2013, pp. 269–310. DOI: 10.1007/s10915-012-9650-3.

[6] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, Feb. 2012, pp. 281–305. DOI: 10.5555/2188385.2188395.

[7] H.-G. Beyer, *The Theory of Evolution Strategies*, ser. Natural Computing Series. Springer, 2001.

[8] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM Review*, vol. 59, no. 1, 2017, 65–98. DOI: 10.1137/141000671.

[9] W. Bian, Y. Chen, and X. Ye, "Deep parallel MRI reconstruction network without coil sensitivities," in *Machine Learning for Medical Image Reconstruction*, F. Deeba, P. Johnson, T. Würfl, and J. C. Ye, Eds., ser. Lecture Notes in Computer Science, pp. 17–26, Springer International Publishing, 2020. DOI: 10.1007/978-3-030-61598-7_2.

[10] J. Borwein and A. Lewis, "Fenchel Duality," in *Convex Analysis and Nonlinear Optimization: Theory and Examples*, ser. CMS Books in Mathematics, New York, NY: Springer, 2006, pp. 33–63. DOI: 10.1007/978-0-387-31256-9_3.

[11] S. Bosse, D. Maniry, K.-R. Muller, T. Wiegand, and W. Samek, "Deep neural networks for no-reference and full-reference image quality assessment," *IEEE Transactions on Image Processing*, vol. 27, no. 1, Jan. 2018, pp. 206–219. DOI: 10.1109/TIP.2017.2760518.

[12] K. Bredies, K. Kunisch, and T. Pock, "Total generalized variation," *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, Jan. 2010, pp. 492–526. DOI: 10.1137/090769521.

[13] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," *SIAM J. Sci. Comp.*, vol. 16, no. 5, 1995, 1190–208. DOI: 10.1137/0916069.

[14] L. Calatroni, C. Chung, J. C. De los Reyes, C.-B. Schönlieb, and T. Valkonen, "Bilevel approaches for learning of variational imaging models," in *Variational Methods in Imaging and Geometric Control*, ser. Radon Series on Computational and Applied Mathematics, vol. 18, De Gruyter, 2017, URL: http://arxiv.org/abs/1505.02120.

[15] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, Feb. 2006, pp. 489–509. DOI: 10.1109/TIT.2005.862083.

[16] E. J. Candès, Y. C. Eldar, D. Needell, and P. Randall, "Compressed sensing with coherent and redundant dictionaries," *Applied and Computational Harmonic Analysis*, vol. 31, no. 1, Jul. 2011, pp. 59–73. DOI: 10.1016/j.acha.2010.10.002.

[17] C. Cartis and L. Roberts. (Feb. 23, 2021). "Scalable subspace methods for derivative-free nonlinear least-squares optimization." arXiv: 2102.12016.

[18] A. Chambolle and P.-L. Lions, "Image recovery via total variation minimization and related problems," *Numerische Mathematik*, vol. 76, no. 2, Apr. 1, 1997, pp. 167–188. DOI: 10.1007/s002110050258.

[19] A. Chambolle and T. Pock, "An introduction to continuous optimization for imaging," *Acta Numerica*, vol. 25, May 2016, pp. 161–319. DOI: 10.1017/S096249291600009X.

[20] A. Chambolle and T. Pock, "On the ergodic convergence rates of a first-order primal—dual algorithm," *Mathematical Programming: Series A and B*, vol. 159, no. 1-2, Sep. 2016, pp. 253–287. DOI: 10.1007/s10107-015-0957-3.

[21] A. Chambolle and T. Pock, "Learning consistent discretizations of the total variation," vol. 14, no. 2, 2021, pp. 778–813. DOI: 10.1137/20M1377199.

[22] T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural Ordinary Differential Equations," in *Advances in Neural Information Processing Systems*, vol. 31, Curran Associates, Inc., 2018, URL: https://papers.nips.cc/paper/2018/hash/69386f6bb1dfed68692a24c8686939b9-Abstract.html.

[23] T. Chen, Y. Sun, and W. Yin. (Feb. 22, 2021). "A single-timescale stochastic bilevel optimization method." arXiv: 2102.04671.

[24] Y. Chen, T. Pock, R. Ranftl, and H. Bischof, "Revisiting loss-specific training of filter-based MRFs for image restoration," in *Pattern Recognition*, J. Weickert, M. Hein, and B. Schiele, Eds., pp. 271–281, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. DOI: 10.1007/978-3-642-40602-7_30.

[25] Y. Chen, R. Ranftl, and T. Pock, "Insights into analysis operator learning: From patch-based sparse models to higher order MRFs," *IEEE Transactions on Image Processing*, vol. 23, no. 3, Mar. 2014, pp. 1060–1072. DOI: 10.1109/TIP.2014.2299065.

[26] Y. Chen, H. Liu, X. Ye, and Q. Zhang, "Learnable descent algorithm for nonsmooth nonconvex image reconstruction," *SIAM Journal on Imaging Sciences*, vol. 14, no. 4, 2021, pp. 1532–1564. DOI: 10.1137/20M1353368.

[27] C. Christof, "Gradient-based solution algorithms for a class of bilevel optimization and optimal control problems with a nonsmooth lower level," *SIAM Journal on Optimization*, vol. 30, no. 1, Jan. 2020, pp. 290–318. DOI: 10.1137/18M1225707.

[28] I. Y. Chun and J. A. Fessler, "Convolutional dictionary learning: Acceleration and convergence," *IEEE Trans. Im. Proc.*, vol. 27, no. 4, Apr. 2018, 1697–712. DOI: 10.1109/TIP.2017.2761545.

[29] I. Y. Chun and J. A. Fessler, "Convolutional analysis operator learning: Acceleration and convergence," *IEEE Transactions on Image Processing*, vol. 29, 2020, pp. 2108–2122. DOI: 10.1109/TIP.2019.2937734.

[30] B. Colson, P. Marcotte, and G. Savard, "An overview of bilevel optimization," *Annals of Operations Research*, vol. 153, no. 1, Jun. 2007, pp. 235–256. DOI: 10.1007/s10479-007-0176-2.

[31] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust Region Methods*, ser. MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, Jan. 1, 2000, 960 pp. DOI: 10.1137/1.9780898719857.

[32] C. Crockett, *BilevelFilterLearningForImageRecon*, 2022, URL: https://github.com/cecroc/BilevelFilterLearningForImageRecon.

[33] C. Crockett and J. A. Fessler, “Motivating bilevel approaches to filter learning: A case study,” in *2021 IEEE International Conference on Image Processing (ICIP)*, pp. 2803–2807, IEEE, Sep. 19, 2021. DOI: 10.1109/ICIP42928.2021.9506489.

[34] C. Crockett, D. Hong, I. Y. Chun, and J. A. Fessler, “Incorporating handcrafted filters in convolutional analysis operator learning for ill-posed inverse problems,” in *2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pp. 316–320, Dec. 2019. DOI: 10.1109/CAMSAP45676.2019.9022669.

[35] M. D’Elia, J. C. De los Reyes, and A. M. Trujillo, *Bilevel parameter optimization for learning nonlocal image denoising models*, Apr. 29, 2020. arXiv: 1912.02347.

[36] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-D transform-domain collaborative filtering,” *IEEE Transactions on Image Processing*, vol. 16, no. 8, Aug. 2007, pp. 2080–2095. DOI: 10.1109/TIP.2007.901238.

[37] B. Dauvergne and L. Hascoet, “The data-flow equations of checkpointing in reverse automatic differentiation,” in *International Conference on Computational Science*, pp. 566–573, 2006. DOI: 10.1007/11758549_78.

[38] J. C. De los Reyes, C.-B. Schönlieb, and T. Valkonen, “Bilevel parameter learning for higher-order total variation regularisation models,” *Journal of Mathematical Imaging and Vision*, vol. 57, no. 1, Jan. 2017, pp. 1–25. DOI: 10.1007/s10851-016-0662-8.

[39] C.-A. Deledalle, S. Vaiter, J. Fadili, and G. Peyré, “Stein Unbiased GrAdient estimator of the Risk (SUGAR) for multiple parameter selection,” *SIAM Journal on Imaging Sciences*, vol. 7, no. 4, Jan. 2014, pp. 2448–2487. DOI: 10.1137/140968045.

[40] S. Dempe and J. Dutta, “Is bilevel programming a special case of a mathematical program with complementarity constraints?” *Mathematical Programming*, vol. 131, no. 1-2, Feb. 2012, pp. 37–48. DOI: 10.1007/s10107-010-0342-1.

[41] S. Dempe, "Annotated bibliography on bilevel programming and mathematical programs with equilibrium constraints," *Optimization*, vol. 52, no. 3, Jun. 2003, pp. 333–359. DOI: 10.1080/0233193031000149894.

[42] S. Dempe and A. Zemkoho, Eds., *Bilevel Optimization: Advances and next Challenges*, vol. 161, ser. Springer Optimization and Its Applications. Springer International Publishing, 2020. DOI: 10.1007/978-3-030-52119-6.

[43] Y. Drori, "The exact information-based complexity of smooth convex minimization," *J. Complexity*, vol. 39, Apr. 2017, 1–16. DOI: 10.1016/j.jco.2016.11.001.

[44] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, no. 1-3, Apr. 1992, 293–318. DOI: 10.1007/BF01581204.

[45] A. Effland, E. Kobler, K. Kunisch, and T. Pock, "Variational networks: An optimal control approach to early stopping variational methods for image restoration," *Journal of Mathematical Imaging and Vision*, vol. 62, no. 3, Apr. 2020, pp. 396–416. DOI: 10.1007/s10851-019-00926-8.

[46] M. J. Ehrhardt and L. Roberts, "Inexact derivative-free optimization for bilevel learning," *Journal of Mathematical Imaging and Vision*, vol. 63, Feb. 6, 2021, pp. 580–600. DOI: 10.1007/s10851-021-01020-8.

[47] M. Elad, *Sparse and redundant representations: from theory to applications in signal and image processing*. Berlin: Springer, 2010. DOI: 10.1007/978-1-4419-7011-4.

[48] M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors," *Inverse Problems*, vol. 23, no. 3, Jun. 2007, pp. 947–68. DOI: 10.1088/0266-5611/23/3/007.

[49] Y. Eldar and G. Kutyniok, *Compressed sensing: Theory and applications*. Cambridge, 2012. DOI: 10.1017/CBO9780511794308.

[50] Y. C. Eldar, "Rethinking biased estimation: Improving maximum likelihood and the Cramer-Rao bound," *Found. & Trends in Sig. Pro.*, vol. 1, no. 4, 2008, 305–449. DOI: 10.1561/2000000008.

[51] Y. Eldar, "Generalized SURE for exponential families: Applications to regularization," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, Feb. 2009, pp. 471–481. DOI: 10.1109/TSP.2008.2008212.

[52] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of inverse problems*. Dordrecht: Kluwer, 1996.

[53] FDA, *510k premarket notification of Deep Learning Image Reconstruction (GE Medical Systems)*, 2019, URL: https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfpmn/pmn.cfm?ID=K183202.

[54] J. Fehrenbach, M. Nikolova, G. Steidl, and P. Weiss, "Bilevel image denoising using gaussianity tests," in *International Conference on Scale Space and Variational Methods in Computer Vision*, vol. 9087, pp. 117–128, 2015. DOI: 10.1007/978-3-319-18461-6_10.

[55] J. A. Fessler, "Mean and variance of implicitly defined biased estimators (such as penalized maximum likelihood): Applications to tomography," *IEEE Trans. Im. Proc.*, vol. 5, no. 3, Mar. 1996, pp. 493–506. DOI: 10.1109/83.491322.

[56] J. A. Fessler, "Model-based image reconstruction for MRI," *IEEE Sig. Proc. Mag.*, vol. 27, no. 4, Jul. 2010, 81–9. DOI: 10.1109/MSP.2010.936726.

[57] J. A. Fessler, "Optimization methods for MR image reconstruction," *IEEE Sig. Proc. Mag.*, vol. 37, no. 1, Jan. 2020, 33–40. DOI: 10.1109/MSP.2019.2943645.

[58] J. A. Fessler and W. L. Rogers, "Spatial resolution properties of penalized-likelihood image reconstruction methods: Space-invariant tomographs," *IEEE Trans. Im. Proc.*, vol. 5, no. 9, Sep. 1996, 1346–58. DOI: 10.1109/83.535846.

[59] J. A. Fessler, *MIRT-demo: 01-recon*, Jul. 25, 2020, URL: https://github.com/JeffFessler/mirt-demo/blob/master/isbi-19/01-recon.jl.

[60] M. Feurer and F. Hutter, “Chapter 1: Hyperparameter optimization,” in *Automated Machine Learning: Methods, Systems, Challenges*, ser. The Springer Series on Challenges in Machine Learning, F. Hutter, L. Kotthoff, and J. Vanschoren, Eds., Springer International Publishing, 2019, pp. 3–33. DOI: 10.1007/978-3-030-05318-5.

[61] R. Fletcher and S. Leyffer, “Numerical experience with solving MPECs as NLPs,” Department of Mathematics and Computer Science, University of Dundee, Dundee, 2002, URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.57.6674.

[62] C.-s. Foo, C. B., and A. Ng, “Efficient multiple hyperparameter learning for log-linear models,” in *Advances in Neural Information Processing Systems*, vol. 20, Curran Associates, Inc., 2007, URL: https://proceedings.neurips.cc/paper/2007/hash/851ddf5058cf22df63d3344ad89919cf-Abstract.html.

[63] L. Franceschi, M. Donini, P. Frasconi, and M. Pontil, “Forward and reverse gradient-based hyperparameter optimization,” in *Proceedings of the International Conference on Machine Learning*, pp. 1165–1173, PMLR, Dec. 12, 2017, URL: http://proceedings.mlr.press/v70/franceschi17a.html.

[64] L. Franceschi, P. Frasconi, S. Salzo, R. Grazzi, and M. Pontil, “Bilevel programming for hyperparameter optimization and meta-learning,” in *International Conference on Machine Learning*, pp. 1568–1577, PMLR, Jul. 3, 2018, URL: http://proceedings.mlr.press/v80/franceschi18a.html.

[65] P. I. Frazier. (Jul. 8, 2018). “A tutorial on bayesian optimization.” arXiv: 1807.02811.

[66] S. W. Fung, H. Heaton, Q. Li, D. McKenzie, S. Osher, and W. Yin, “JFB: Jacobian-free backpropagation for implicit networks,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. arXiv: 2103.12803.

[67] C. Garcia-Cardona and B. Wohlberg, “Convolutional dictionary dearning: A comparative review and new algorithms,” *IEEE Transactions on Computational Imaging*, vol. 4, no. 3, Sep. 2018, pp. 366–381. DOI: 10.1109/TCI.2018.2840334.

[68] O. Gencoglu, M. van Gils, E. Guldogan, C. Morikawa, M. Süzen, M. Gruber, J. Leinonen, and H. Huttunen. (Apr. 16, 2019). "HARK side of deep learning – From grad student descent to automated machine learning." arXiv: 1904.07633.

[69] S. Ghadimi and M. Wang. (Feb. 6, 2018). "Approximation methods for bilevel programming." arXiv: 1802.02246.

[70] M. Gholizadeh-Ansari, J. Alirezaie, and P. Babyn, "Deep learning for low-dose CT denoising using perceptual loss and edge detection layer," *J. Digital Im.*, vol. 33, no. 2, 2020, 504–15. DOI: 10.1007/s10278-019-00274-4.

[71] A. Ghosh, *Questions about BLORC*, E-mail, Feb. 21, 2022.

[72] A. Ghosh, M. T. Mccann, and S. Ravishankar, *Bilevel learning of l1-regularizers with closed-form gradients(BLORC)*, Nov. 21, 2021. arXiv: 2111.10858.

[73] D. Gilton, G. Ongie, and R. Willett, "Deep equilibrium architectures for inverse problems in imaging," *IEEE Transactions on Computational Imaging*, vol. 7, 2021, pp. 1123–1133. DOI: 10.1109/TCI.2021.3118944.

[74] R. Giryes, M. Elad, and Y. C. Eldar, "The projected GSURE for automatic parameter tuning in iterative shrinkage methods," *Applied and Computational Harmonic Analysis*, vol. 30, no. 3, May 2011, 407–22. DOI: 10.1016/j.acha.2010.11.005.

[75] G. H. Golub, M. Heath, and G. Wahba, "Generalized cross-validation as a method for choosing a good ridge parameter," *Technometrics*, vol. 21, no. 2, May 1979, 215–23, URL: http://www.jstor.org/stable/1268518.

[76] G. H. Golub and C. F. Van Loan, "An analysis of the total least squares problem," *SIAM J. Numer. Anal.*, vol. 17, no. 6, Dec. 1980, 883–93. DOI: 10.1137/0717073.

[77] S. Gould, B. Fernando, A. Cherian, P. Anderson, R. S. Cruz, and E. Guo. (Jul. 20, 2016). "On differentiating parameterized argmin and argmax problems with application to bi-level optimization." arXiv: 1607.05447.

[78] B. Gozcu, R. K. Mahabadi, Y.-H. Li, E. Ilicak, T. Cukur, J. Scarlett, and V. Cevher, "Learning-based compressive MRI," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, Jun. 2018, 1394–406. DOI: 10.1109/TMI.2018.2832540.

[79] R. Grazzi, L. Franceschi, M. Pontil, and S. Salzo, "On the iteration complexity of hypergradient computation," in *Proceedings of the 37th International Conference on Machine Learning*, p. 11, 2020, URL: http://proceedings.mlr.press/v119/grazzi20a.html.

[80] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. Intl. Conf. Mach. Learn*, 2010, URL: http://yann.lecun.com/exdb/publis/pdf/gregor-icml-10.pdf.

[81] C. Guillemot and O. Le Meur, "Image inpainting: Overview and recent advances," *IEEE Sig. Proc. Mag.*, vol. 31, no. 1, Jan. 2014, 127–44. DOI: 10.1109/MSP.2013.2273004.

[82] E. Haber and L. Tenorio, "Learning regularization functionals a supervised training approach," *Inverse Problems*, vol. 19, no. 3, Jun. 1, 2003, pp. 611–626. DOI: 10.1088/0266-5611/19/3/309.

[83] K. Hammernik, T. Klatzer, E. Kobler, M. P. Recht, D. K. Sodickson, T. Pock, and F. Knoll, "Learning a variational network for reconstruction of accelerated MRI data," *Magnetic Resonance in Medicine*, vol. 79, no. 6, 2018, pp. 3055–3071. DOI: 10.1002/mrm.26977.

[84] K. Hammernik and F. Knoll, "Machine learning for image reconstruction," in *Handbook of Medical Image Computing and Computer Assisted Intervention*, Elsevier, 2020, pp. 25–64. DOI: 10.1016/B978-0-12-816176-0.00007-7.

[85] S. Hawe, M. Kleinsteuber, and K. Diepold, "Analysis operator learning and its application to image reconstruction," *IEEE Transactions on Image Processing*, vol. 22, no. 6, Jun. 2013, pp. 2138–2150. DOI: 10.1109/TIP.2013.2246175.

[86] S. Haykin, "Neural networks expand SP's horizons," *IEEE Sig. Proc. Mag.*, vol. 13, no. 2, Mar. 1996, 24–49. DOI: 10.1109/79.487040.

[87] J. He, Y. Yang, Y. Wang, D. Zeng, Z. Bian, H. Zhang, J. Sun, Z. Xu, and J. Ma, "Optimizing a parameterized plug-and-play ADMM for iterative low-dose CT reconstruction," *IEEE Transactions on Medical Imaging*, vol. 38, no. 2, Feb. 2019, pp. 371–382. DOI: 10.1109/TMI.2018.2865202.

[88] H. Heaton, S. Wu Fung, A. Gibali, and W. Yin, "Feasibility-based fixed point networks," *Fixed Point Theory and Algorithms for Sciences and Engineering*, vol. 2021, no. 1, Dec. 2021, p. 21. DOI: 10.1186/s13663-021-00706-3.

[89] T. J. Hebert and R. Leahy, "Statistic-based MAP image reconstruction from Poisson data using Gibbs priors," *IEEE Trans. Sig. Proc.*, vol. 40, no. 9, Sep. 1992, 2290–303. DOI: 10.1109/78.157228.

[90] M. Hintermüller, K. Papafitsoros, C. N. Rautenberg, and H. Sun. (Feb. 13, 2020). "Dualization and automatic distributed parameter selection of total generalized variation via bilevel optimization." arXiv: 2002.05614.

[91] M. Hintermüller and T. Wu, "Bilevel optimization for calibrating point spread functions in blind deconvolution," *Inverse Problems & Imaging*, vol. 9, no. 4, 2015, pp. 1139–1169. DOI: 10.3934/ipi.2015.9.1139.

[92] L. Hoeltgen, S. Setzer, and J. Weickert, "An optimal control approach to find sparse data for Laplace interpolation," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, A. Heyden, F. Kahl, C. Olsson, M. Oskarsson, and X.-C. Tai, Eds., vol. 8081, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 151–164. DOI: 10.1007/978-3-642-40395-8_12.

[93] P. W. Holland and R. E. Welsch, "Robust regression using iteratively reweighted least-squares," *Comm. in Statistics—Theory and Methods*, vol. 6, no. 9, 1977, 813–27, DOI: 10.1080/03610927708827533.

[94] G. Holler, K. Kunisch, and R. C. Barnard, "A bilevel approach for parameter learning in inverse problems," *Inverse Problems*, vol. 34, no. 11, Nov. 1, 2018, p. 115 012. DOI: 10.1088/1361-6420/aade77.

[95] M. Hong, H.-T. Wai, Z. Wang, and Z. Yang, *A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic*, Dec. 20, 2020. arXiv: 2007.05170.

[96] J.-N. Hwang, S.-Y. Kung, M. Niranjan, and J. C. Principe, "The past, present, and future of neural networks for signal processing," *IEEE Sig. Proc. Mag.*, vol. 14, no. 6, Nov. 1997, 28–48. DOI: 10.1109/79.637299.

[97] P. Jain and P. Kar, "Non-convex optimization for machine learning," *Found. & Trends in Machine Learning*, vol. 10, no. 3-4, 2017, 142–336. DOI: 10.1561/2200000058.

[98] K. Ji, J. Yang, and Y. Liang, "Bilevel optimization: Convergence analysis and enhanced design," in *Proceedings of the 38th International Conference on Machine Learning*, pp. 4882–4892, Jul. 2021, URL: http://proceedings.mlr.press/v139/ji21c.html.

[99] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," *IEEE Trans. Im. Proc.*, vol. 26, no. 9, Sep. 2017, 4509–22. DOI: 10.1109/TIP.2017.2713099.

[100] J. Kaipioa and E. Somersalo, "Statistical inverse problems: Discretization, model reduction and inverse crimes," *J. Comp. Appl. Math.*, vol. 198, no. 2, Jan. 2007, 493–504. DOI: 10.1016/j.cam.2005.09.027.

[101] L. Kang, P. Ye, Y. Li, and D. Doermann, "Convolutional neural networks for no-reference image quality assessment," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1733–1740, Jun. 2014. DOI: 10.1109/CVPR.2014.224.

[102] M. Kellman, K. Zhang, E. Markley, J. Tamir, E. Bostan, M. Lustig, and L. Waller, "Memory-efficient learning for large-scale computational imaging," *IEEE Transactions on Computational Imaging*, vol. 6, 2020, pp. 1403–1414. DOI: 10.1109/TCI.2020.3025735.

[103] P. Khanduri, H.-T. Wai, S. Zeng, M. Hong, Z. Wang, and Z. Yang, "A near-optimal algorithm for stochastic bilevel optimization via double-momentum," in *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, p. 13, 2021, URL: https://proceedings.neurips.cc/paper/2021/hash/fe2b421b8b5f0e7c355ace66a9fe0206-Abstract.html.

[104] D. Kim and J. A. Fessler, "Adaptive restart of the optimized gradient method for convex optimization," *J. Optim. Theory Appl.*, vol. 178, no. 1, Jul. 2018, 240–63. DOI: 10.1007/s10957-018-1287-4.

[105] D. Kim and J. A. Fessler, "On the convergence analysis of the optimized gradient method," *Journal of Optimization Theory and Applications*, vol. 172, no. 1, Jan. 2017, pp. 187–205. DOI: 10.1007/s10957-016-1018-7.

[106] K. Kim, S. Soltanayev, and S. Y. Chun, "Unsupervised training of denoisers for low-dose CT reconstruction without full-dose ground truth," *IEEE J. Sel. Top. Sig. Proc.*, vol. 14, no. 6, Oct. 2020, 1112–25. DOI: 10.1109/JSTSP.2020.3007326.

[107] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations*, vol. abs/1412.6980, May 2015. arXiv: 1412.6980.

[108] A. Klein, S. Falkner, S. Bartels, P. Hennig, and F. Hutter, "Fast Bayesian hyperparameter optimization on large datasets," *Electron. J. Statist.*, vol. 11, no. 2, 2017, pp. 4945–68. DOI: 10.1214/17-EJS1335SI.

[109] P. Knöbelreiter, C. Sormann, A. Shekhovtsov, F. Fraundorfer, and T. Pock, "Belief propagation reloaded: Learning BP-layers for labeling problems," presented at the The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7897–7906, Jun. 2020. DOI: 10.1109/CVPR42600.2020.00792.

[110] F. Knoll, K. Bredies, T. Pock, and R. Stollberger, "Second order total generalized variation (TGV) for MRI," *Mag. Res. Med.*, vol. 65, no. 2, 2011, 480–91. DOI: 10.1002/mrm.22595.

[111] E. Kobler, A. Effland, K. Kunisch, and T. Pock, “Total deep variation: A stable regularization method for inverse problems,” *IEEE transactions on pattern analysis and machine intelligence*, Nov. 2021. DOI: 10.1109/TPAMI.2021.3124086, Advance online publication. PMID: 34727026.

[112] F. K. Kopp, M. Catalano, D. Pfeiffer, A. A. Fingerle, E. J. Rummeny, and P. B. Noel, “CNN as model observer in a liver lesion detection task for x-ray computed tomography: A phantom study,” *Med. Phys.*, vol. 45, no. 10, Oct. 2018, 4439–47. DOI: 10.1002/mp.13151.

[113] K. Kunisch and T. Pock, “A bilevel optimization approach for parameter learning in variational models,” *SIAM Journal on Imaging Sciences*, vol. 6, no. 2, Jan. 2013, pp. 938–983. DOI: 10.1137/120882706.

[114] J. Larson, M. Menickelly, and S. M. Wild, “Derivative-free optimization methods,” *Acta Numerica*, vol. 28, May 1, 2019, pp. 287–404. DOI: 10.1017/S0962492919000060.

[115] B. Lecouat, J. Ponce, and J. Mairal, “A flexible framework for designing trainable priors with adaptive smoothing and game encoding,” in *Advances in Neural Information Processing Systems*, vol. 33, pp. 15 664–15 675, 2020, URL: https://papers.nips.cc/paper/2020/hash/b4edda67f0f57e218a8e766927e3e5c5-Abstract.html.

[116] R. M. Lewitt and S. Matej, “Overview of methods for image reconstruction from projections in emission computed tomography,” *Proc. IEEE*, vol. 91, no. 10, Oct. 2003, 1588–611. DOI: 10.1109/JPROC.2003.817882.

[117] H. Lim, I. Y. Chun, Y. K. Dewaraja, and J. A. Fessler, “Improved low-count quantitative PET reconstruction with an iterative neural network,” *IEEE Transactions on Medical Imaging*, vol. 39, no. 11, Nov. 2020, pp. 3512–3522. DOI: 10.1109/TMI.2020.2998480.

[118] G. W. Lindsay, “Convolutional neural networks as a model of the visual system: Past, present, and future,” *Journal of Cognitive Neuroscience*, Feb. 6, 2020, pp. 1–15. DOI: 10.1162/jocn_a_01544.

[119] T. Liu, A. Chaman, D. Belius, and I. Dokmanić, *Learning multi-scale convolutional dictionaries for image reconstruction*, Aug. 19, 2021. arXiv: 2011.12815.

[120] J. Lorraine, P. Vicol, and D. Duvenaud, "Optimizing millions of hyperparameters by implicit differentiation," in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, pp. 1540–1552, PMLR, Jun. 3, 2020, URL: https://proceedings.mlr.press/v108/lorraine20a.html.

[121] A. Lucas, M. Iliadis, R. Molina, and A. K. Katsaggelos, "Using deep neural networks for inverse problems in imaging: Beyond analytical methods," *IEEE Sig. Proc. Mag.*, vol. 35, no. 1, Jan. 2018, 20–36. DOI: 10.1109/msp.2017.2760358.

[122] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, Apr. 2012, pp. 791–804. DOI: 10.1109/TPAMI.2011.156.

[123] J. Mairal, G. Sapiro, and M. Elad, "Learning multiscale sparse representations for image and video restoration," *Multiscale Modeling & Simulation*, vol. 7, no. 1, Jan. 2008, pp. 214–241. DOI: 10.1137/070697653.

[124] A. Mason, J. Rioux, S. E. Clarke, A. Costa, M. Schmidt, V. Keough, T. Huynh, and S. Beyea, "Comparison of objective image quality metrics to expert radiologists' scoring of diagnostic quality of MR images," *IEEE Transactions on Medical Imaging*, vol. 39, no. 4, Apr. 2020, pp. 1064–1072. DOI: 10.1109/TMI.2019.2930338.

[125] M. T. McCann and M. Unser, "Biomedical image reconstruction: From the foundations to deep neural networks," *Foundation and Trends in Signal Processing*, vol. 13, no. 3, 2019, pp. 283–359. DOI: 10.1561/2000000101.

[126] M. T. McCann and S. Ravishankar, "Supervised learning of sparsity-promoting regularizers for denoising," *arXiv Computing Research Repository*, Jun. 9, 2020. arXiv: 2006.05521.

[127] C. H. McCollough, A. C. Bartley, R. E. Carter, B. Chen, T. A. Drees, P. Edwards, D. R. Holmes, A. E. Huang, F. Khan, S. Leng, K. L. McMillan, G. J. Michalak, K. M. Nunez, L. Yu, and J. G. Fletcher, "Low-dose CT for the detection and classification of metastatic liver lesions: Results of the 2016 Low Dose CT Grand Challenge," *Med. Phys.*, vol. 44, no. 10, Oct. 2017, e339–52. DOI: 10.1002/mp.12345.

[128] A. Mehra and J. Hamm, "Penalty method for inversion-free deep bilevel optimization," in *Proceedings of The 13th Asian Conference on Machine Learning*, pp. 347–362, PMLR, Nov. 28, 2021, URL: https://proceedings.mlr.press/v157/mehra21a.html.

[129] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a "completely blind" image quality analyzer," *IEEE Signal Processing Letters*, vol. 20, no. 3, Mar. 2013, pp. 209–212. DOI: 10.1109/LSP.2012.2227726.

[130] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Processing Magazine*, vol. 38, no. 2, Mar. 2021, pp. 18–44. DOI: 10.1109/MSP.2020.3016905.

[131] G. Muniraju, B. Kailkhura, J. J. Thiagarajan, and T. Bremer, "Controlled random search improves sample mining and hyperparameter optimization," in *Thirty-Third AAAI Conference on Artificial Intelligence*, 2019, URL: https://www.osti.gov/servlets/purl/1497973.

[132] S. Nam, M. Davies, M. Elad, and R. Gribonval, "The cosparse analysis model and algorithms," *Applied and Computational Harmonic Analysis*, vol. 34, no. 1, Jan. 2013, pp. 30–56. DOI: 10.1016/j.acha.2012.03.006.

[133] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $O(1/k^2)$," *Soviet Math. Dokl.*, vol. 27, no. 2, 1983, 372–76.

[134] L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takác, "SARAH: A novel method for machine learning problems using stochastic recursive gradient," in *34th International Conference on Machine Learning*, p. 9, 2017, URL: https://proceedings.mlr.press/v70/nguyen17b.html.

[135] M. Nikolova and ,CMLA, ENS Cachan, CNRS, PRES UniverSud, 61 Av. President Wilson, F-94230 Cachan, “Model distortions in Bayesian MAP reconstruction,” *Inverse Problems & Imaging*, vol. 1, no. 2, 2007, pp. 399–422. DOI: 10.3934/ipi.2007.1.399.

[136] S. Nowozin and C. H. Lampert, “Structured learning and prediction in computer vision,” *Foundations and Trends® in Computer Graphics and Vision*, vol. 6, no. 3-4, 2011, pp. 185–365. DOI: 10.1561/0600000033.

[137] P. Ochs, R. Ranftl, T. Brox, and T. Pock, “Techniques for gradient-based bilevel optimization with non-smooth lower level problems,” *Journal of Mathematical Imaging and Vision*, vol. 56, no. 2, Oct. 2016, pp. 175–194. DOI: 10.1007/s10851-016-0663-7.

[138] B. Ophir, M. Elad, N. Bertin, and M. D. Plumbley, “Sequential minimal eigenvalues - an approach to analysis dictionary learning,” *19th European Signal Processing Conference*, 2011, pp. 1465–1469, URL: https://ieeexplore.ieee.org/document/7074010.

[139] D. P. Palomar and Y. C. Eldar, *Convex optimization in signal processing and communications.* Cambridge, 2011. DOI: 10.1017/CBO9780511804458.

[140] F. Pedregosa, “Hyperparameter optimization with approximate gradient,” in *Proceedings International Conference on Machine Learning*, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48, pp. 737–46, PMLR, Jun. 20–22, 2016, URL: http://proceedings.mlr.press/v48/pedregosa16.html.

[141] K. B. Petersen and M. S. Pedersen, *The Matrix Cookbook.* Technical University of Denmark, Nov. 2012, URL: http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=3274.

[142] G. Peyre, “A review of adaptive image representations,” *IEEE J. Sel. Top. Sig. Proc.*, vol. 5, no. 5, Sep. 2011, 896–911. DOI: 10.1109/JSTSP.2011.2120592.

[143] G. Peyré and J. M. Fadili, “Learning analysis sparsity priors,” in *IEEE Intl. Conf. on Sampling Theory and Appl. (SampTA)*, 2011, URL: https://hal.archives-ouvertes.fr/hal-00542016.

[144] L. Pfister and Y. Bresler, "Learning filter bank sparsifying transforms," *IEEE Transactions on Signal Processing*, vol. 67, no. 2, Jan. 2019, pp. 504–519. DOI: 10.1109/TSP.2018.2883021.

[145] D. L. Phillips, "A technique for the numerical solution of certain integral equations of the first kind," *J. Assoc. Comput. Mach.*, vol. 9, no. 1, Jan. 1962, 84–97. DOI: 10.1145/321105.321114.

[146] C. Poon and G. Peyré, "Smooth Bilevel Programming for Sparse Regularization," in *35th Conference on Neural Information Processing Systems*, 2021, URL: https://proceedings.neurips.cc/paper/2021/hash/0bed45bd5774ffddc95ffe500024f628-Abstract.html.

[147] J. Qi and R. H. Huesman, "Penalized maximum-likelihood image reconstruction for lesion detection," *Phys. Med. Biol.*, vol. 51, no. 16, Aug. 2006, 4017–30. DOI: 10.1088/0031-9155/51/16/009.

[148] S. Ramani, T. Blu, and M. Unser, "Monte-carlo sure: A black-box optimization of regularization parameters for general denoising algorithms," *IEEE Transactions on Image Processing*, vol. 17, no. 9, Sep. 2008, pp. 1540–1554. DOI: 10.1109/TIP.2008.2001404.

[149] Z. Ramzi, F. Mannel, S. Bai, J.-L. Starck, P. Ciuciu, and T. Moreau, *SHINE: SHaring the INverse Estimate from the forward pass for bi-level optimization and implicit models*, Jun. 24, 2021. arXiv: 2106.00553.

[150] S. Ravishankar and Y. Bresler, "MR image reconstruction from highly undersampled k-space data by dictionary learning," *IEEE Transactions on Medical Imaging*, vol. 30, no. 5, May 2011, pp. 1028–1041. DOI: 10.1109/TMI.2010.2090538.

[151] S. Ravishankar, J. C. Ye, and J. A. Fessler, "Image reconstruction: From sparsity to data-adaptive methods and machine learning," *Proc. IEEE*, vol. 108, no. 1, Jan. 2020, 86–109. DOI: 10.1109/JPROC.2019.2936204.

[152] S. Ravishankar and Y. Bresler, "Learning sparsifying transforms," *IEEE Transactions on Signal Processing*, vol. 61, no. 5, Mar. 2013, pp. 1072–1086. DOI: 10.1109/TSP.2012.2226449.

[153] G. P. Renieblas, A. T. Nogués, A. M. González, N. G. León, and E. G. . Castillo, "Structural similarity index family for image quality assessment in radiological images," *J. Med. Im.*, vol. 4, no. 3, Jul. 2017, p. 035 501. DOI: 10.1117/1.JMI.4.3.035501.

[154] L. Roberts, *Inexact DFO for Bilevel Learning: Dimension Question*, E-mail, Jul. 11, 2021.

[155] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, 234–41, 2015. DOI: 10.1007/978-3-319-24574-4_28.

[156] S. Roth and M. Black, "Fields of experts: A framework for learning image priors," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, pp. 860–867, 2005. DOI: 10.1109/CVPR.2005.160.

[157] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithm," *Physica D*, vol. 60, no. 1-4, Nov. 1992, 259–68. DOI: 10.1016/0167-2789(92)90242-F.

[158] B. Sahiner, A. Pezeshk, L. M. Hadjiiski, X. Wang, K. Drukker, K. Cha, R. Summers, and M. L. Giger, "Deep learning in medical imaging and radiation therapy," *Medical Physics*, Nov. 2018. DOI: 10.1002/mp.13264.

[159] K. G. G. Samuel and M. F. Tappen, "Learning optimized MAP estimates in continuously-valued MRF models," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 477–484, Jun. 2009. DOI: 10.1109/CVPR.2009.5206774.

[160] S. S. Saquib, C. A. Bouman, and K. Sauer, "ML parameter estimation for Markov random fields, with applications to Bayesian tomography," *IEEE Trans. Im. Proc.*, vol. 7, no. 7, Jul. 1998, 1029–44. DOI: 10.1109/83.701163.

[161] S. Scholtes and M. Stöhr, "How stringent is the linear independence assumption for mathematical programs with complementarity constraints?" *Mathematics of Operations Research*, vol. 26, no. 4, Nov. 2001, pp. 851–863. DOI: 10.1287/moor.26.4.851.10007.

[162] W. P. Segars, G. Sturgeon, S. Mendonca, J. Grimes, and B. M. W. Tsui, "4D XCAT phantom for multimodality imaging research," *Medical Physics*, vol. 37, no. 9, Aug. 2010, pp. 4902–15. DOI: 10.1118/1.3480985.

[163] G. Seif and D. A., "Edge-based loss function for single image super-resolution," in *Proc. IEEE Conf. Acoust. Speech Sig. Proc.*, 1468–72, 2018. DOI: 10.1109/ICASSP.2018.8461664.

[164] S. Setzer, G. Steidl, and T. Teuber, "Infimal convolution regularizations with discrete ℓ1-type functionals," *Comm. Math. Sci.*, vol. 9, no. 3, 2011, 797–827. DOI: 10.4310/CMS.2011.v9.n3.a7.

[165] A. Shaban, C.-A. Cheng, N. Hatch, and B. Boots, "Truncated back-propagation for bilevel optimization," in *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pp. 1723–1732, PMLR, Apr. 11, 2019, URL: https://proceedings.mlr.press/v89/shaban19a.html.

[166] L. A. Shepp and B. F. Logan, "The Fourier reconstruction of a head section," *IEEE Trans. Nuc. Sci.*, vol. 21, no. 3, Jun. 1974, 21–43. DOI: 10.1109/TNS.1974.6499235.

[167] F. Sherry, M. Benning, J. C. De los Reyes, M. J. Graves, G. Maierhofer, G. Williams, C.-B. Schonlieb, and M. J. Ehrhardt, "Learning the sampling pattern for MRI," *IEEE Transactions on Medical Imaging*, vol. 39, no. 12, Dec. 2020, pp. 4310–4321. DOI: 10.1109/TMI.2020.3017353.

[168] N. Shlezinger, J. Whang, Y. C. Eldar, and A. G. Dimakis, *Model-based deep learning*, Dec. 15, 2020. arXiv: 2012.08405.

[169] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, Jul. 2019, p. 60. DOI: 10.1186/s40537-019-0197-0.

[170] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, May 2015, URL: http://arxiv.org/abs/1409.1556.

[171] B. Sixou, "Adaptative regularization parameter for poisson noise with a bilevel approach: Application to spectral computerized tomography," *Inverse Problems in Science and Engineering*, Dec. 22, 2020, pp. 1–18. DOI: 10.1080/17415977.2020.1864348.

[172] J. Solomon, P. Lyu, D. Marin, and E. Samei, "Noise and spatial resolution properties of a commercially available deep learning-based CT reconstruction algorithm," *Med. Phys.*, vol. 47, no. 9, 2020, 3961–71. DOI: 10.1002/mp.14319.

[173] S. Soltanayev and S. Y. Chun, "Training deep learning based denoisers without ground truth data," in *Neural Information Processing Systems*, vol. 31, 2018, URL: https://papers.nips.cc/paper/7587-training-deep-learning-based-denoisers-without-ground-truth-data.

[174] P. Sprechmann, R. Litman, T. B. Yakar, A. M. Bronstein, and G. Sapiro, "Supervised sparse analysis and synthesis operators," in *Neural Information Processing Systems*, pp. 908–916, 2013, URL: https://papers.nips.cc/paper/2013/hash/7380ad8a673226ae47fce7bff88e9c33-Abstract.html.

[175] C. M. Stein, "Estimation of the mean of a multivariate normal distribution," *The Annals of Statistics*, vol. 9, no. 6, Nov. 1, 1981. DOI: 10.1214/aos/1176345632.

[176] M. Stone, "Cross-validation: A review," *Math Oper Stat Ser Stat.*, vol. 9, no. 1, 1978, 127–139. DOI: 10.1080/02331887808801414.

[177] M. F. Tappen, C. Liu, E. H. Adelson, and W. T. Freeman, "Learning gaussian conditional random fields for low-level vision," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, Jun. 2007. DOI: 10.1109/CVPR.2007.382979.

[178] The University of Texas at Austin: Laboratory for Image and Video Engineering. (n.d.). "Image & video quality assessment at LIVE." URL: http://live.ece.utexas.edu/research/quality/.

[179] M. Thies, F. Wagner, M. Gu, L. Folle, L. Felsner, and A. Maier, *Learned Cone-Beam CT Reconstruction Using Neural Ordinary Differential Equations*, Jan. 19, 2022. arXiv: 2201.07562.

[180] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, Jan. 1996, pp. 267–288. DOI: 10.1111/j.2517-6161.1996.tb02080.x.

[181] R. J. Tibshirani and J. Taylor, "The solution path of the generalized lasso," *The Annals of Statistics*, vol. 39, no. 3, Jun. 2011. DOI: 10.1214/11-AOS878.

[182] M. Unser and T. Blu, "Generalized smoothing splines and the optimal discretization of the Wiener filter," *IEEE Trans. Sig. Proc.*, vol. 53, no. 6, Jun. 2005, 2146–59. DOI: 10.1109/TSP.2005.847821.

[183] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-Play priors for model based reconstruction," in *2013 IEEE Global Conference on Signal and Information Processing*, pp. 945–948, IEEE, Dec. 2013. DOI: 10.1109/GlobalSIP.2013.6737048.

[184] G. Wang, "A perspective on deep imaging," *IEEE Access*, vol. 4, Nov. 2016, 8914–24. DOI: 10.1109/ACCESS.2016.2624938.

[185] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Im. Proc.*, vol. 13, no. 4, Apr. 2004, 600–612. DOI: 10.1109/TIP.2003.819861.

[186] Z. Wang, E. Simoncelli, and A. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, pp. 1398–1402, IEEE, 2003. DOI: 10.1109/ACSSC.2003.1292216.

[187] Z. Wang and A. Bovik, "Reduced- and no-reference image quality assessment," *IEEE Signal Processing Magazine*, vol. 28, no. 6, Nov. 2011, pp. 29–40. DOI: 10.1109/MSP.2011.942471.

[188] B. Wen, S. Ravishankar, L. Pfister, and Y. Bresler, "Transform learning for magnetic resonance image reconstruction: From model-based learning to building neural networks," *IEEE Sig. Proc. Mag.*, vol. 37, no. 1, Jan. 2020, 41–53. DOI: 10.1109/MSP.2019.2951469.

[189] J. Xu and F. Noo, "Patient-specific hyperparameter learning for optimization-based CT image reconstruction," *Physics in Medicine & Biology*, vol. 66, no. 19, Sep. 2021, 19NT01. DOI: 10.1088/1361-6560/ac0f9a.

[190] M. Yaghoobi, S. Nam, R. Gribonval, and M. E. Davies, "Constrained overcomplete analysis operator learning for cosparse signal modelling," *IEEE Transactions on Signal Processing*, vol. 61, no. 9, May 2013, pp. 2341–2355. DOI: 10.1109/TSP.2013.2250968.

[191] M. Yaghoobi, S. Nam, R. Gribonval, and M. E. Davies, "Analysis operator learning for overcomplete cosparse representations," presented at the 2011 19th European Signal Processing Conference, pp. 1470–1474, IEEE, 2011, URL: https://ieeexplore.ieee.org/document/7074220.

[192] J. Yang, K. Ji, and Y. Liang, "Provably faster algorithms for bilevel optimization," in *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, 2021, URL: https://proceedings.neurips.cc/paper/2021/hash/71cc107d2e0408e60a3d3c44f47507bd-Abstract.html.

[193] L. Yang, J. Zhou, A. Ferrero, R. D. Badawi, and J. Qi, "Regularization design in penalized maximum-likelihood image reconstruction for lesion detection in 3D PET," *Phys. Med. Biol.*, vol. 59, no. 2, Jan. 2014, 403–20. DOI: 10.1088/0031-9155/59/2/403.

[194] J. C. Ye, Y. Han, and E. Cha, "Deep convolutional framelets: A general deep learning framework for inverse problems," *SIAM J. Imaging Sci.*, vol. 11, no. 2, Jan. 2018, 991–1048. DOI: 10.1137/17m1141771.

[195] A. Yendiki and J. A. Fessler, "Analysis of observer performance in unknown-location tasks for tomographic image reconstruction," *J. Opt. Soc. Am. A*, vol. 24, no. 12, Dec. 2007, B99–109. DOI: 10.1364/JOSAA.24.000B99.

[196] L. Ying and J. Sheng, "Joint image reconstruction and sensitivity estimation in SENSE (JSENSE)," *Mag. Res. Med.*, vol. 57, no. 6, Jun. 2007, 1196–1202. DOI: 10.1002/mrm.21245.

[197] C. You, Q. Yang, H. Shan, L. Gjesteby, G. Li, S. Ju, Z. Zhang, Z. Zhao, Y. Zhang, W. Cong, and G. Wang, "Structure-sensitive multi-scale deep neural network for low-dose CT denoising," *IEEE Access*, vol. 6, 2018, pp. 41 839–41 855. DOI: 10.1109/ACCESS.2018.2858196.

[198] H. Zhang, X. Chen, X. Zhang, and X. Zhang, "A bi-level nested sparse optimization for adaptive mechanical fault feature detection," *IEEE Access*, vol. 8, 2020, pp. 19 767–19 782. DOI: 10.1109/ACCESS.2020.2968726.

[199] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "A comprehensive evaluation of full reference image quality assessment algorithms," in *2012 19th IEEE International Conference on Image Processing*, pp. 1477–1480, Sep. 2012. DOI: 10.1109/ICIP.2012.6467150.

[200] W. Zhang, K. Ma, J. Yan, D. Deng, and Z. Wang, "Blind image quality assessment using a deep bilinear convolutional neural network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 1, Jan. 2020, pp. 36–47. DOI: 10.1109/TCSVT.2018.2886771.

[201] P. Zhou, C. Zhang, and Z. Lin, "Bilevel model-based discriminative dictionary learning for recognition," *IEEE Trans. Im. Proc.*, vol. 26, no. 3, Mar. 2017, 1173–87. DOI: 10.1109/tip.2016.2623487.

[202] M. Zhussip, S. Soltanayev, and S. Y. Chun, "Training deep learning based image denoisers from undersampled measurements without ground truth and without image prior," in *Proc. IEEE Conf. on Comp. Vision and Pattern Recognition*, 10247–56, 2019. DOI: 10.1109/CVPR.2019.01050.

[203] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Royal Stat. Soc. Ser. B*, vol. 67, no. 2, 2005, 301–20. DOI: 10.1111/j.1467-9868.2005.00503.x.