Learning Multiscale Convolutional Dictionaries for Image Reconstruction

Tianlin Liu, Anadi Chaman, David Belius, and Ivan Dokmanić, *Member, IEEE*

***Abstract*—Convolutional neural networks (CNNs) have been tremendously successful in solving imaging inverse problems. To understand their success, an effective strategy is to construct simpler and mathematically more tractable convolutional sparse coding (CSC) models that share essential ingredients with CNNs. Existing CSC methods, however, underperform leading CNNs in challenging inverse problems. We hypothesize that the performance gap may be attributed in part to how they process images at different spatial scales: While many CNNs use multiscale feature representations, existing CSC models mostly rely on single-scale dictionaries. To close the performance gap, we thus propose a multiscale convolutional dictionary structure. The proposed dictionary structure is derived from the U-Net, arguably the most versatile and widely used CNN for image-to-image learning problems. We show that incorporating the proposed multiscale dictionary in an otherwise standard CSC framework yields performance competitive with state-of-the-art CNNs across a range of challenging inverse problems including CT and MRI reconstruction. Our work thus demonstrates the effectiveness and scalability of the multiscale CSC approach in solving challenging inverse problems.**

***Index Terms*—Computed tomography, convolutional neural networks, convolutional sparse coding, dictionary learning, inverse problems, multiscale representation, U-Net.**

I. Introduction

Convolutional neural networks (CNNs) obtain state-of-the-art performance in many image processing tasks. To understand their success, an active line of recent research reduces CNNs into conceptually simpler and mathematically better-understood building blocks. Examples of these simplified convolutional models include *convolutional kernels* [1]–[3], *convolutional scattering transforms* [4]–[7], and *convolutional sparse coding* [8]–[10]. In addition to being mathematically tractable, these models have achieved remarkable empirical success, sometimes matching state-of-the-art CNNs.

This work studies convolutional representations arising from the convolutional sparse coding (CSC) paradigm, which provides a natural connection between sparse representation models and CNNs. Indeed, many CNN instances can be interpreted as optimizing a CSC objective through cascaded layers [8]. Moreover, CSC models compete favorably with state-of-the-art CNNs in several image processing tasks including denoising, single image super-resolution, and inpainting [10]–[18].

While these emerging results are promising, the successful applications of CSC in imaging inverse problems are still confined to problems with relatively simple forward operators, including Gaussian noise addition, blurring, and uniformly random pixel removal. Common to these forward operators is their *spatial locality* – they introduce artifacts that are spatially correlated only, if at all, within small pixel neighbourhoods. By contrast, a broad range of imaging inverse problems involve forward models that mix distant parts of the image and are *highly spatially heterogeneous*; examples include the Radon transform for computed tomography, which computes line integrals along radiating paths, and the Fourier transform for magnetic resonance imaging, which computes inner products with globally-supported sinusoids. Working with these forward models presents different challenges since they introduce structured noise, such as streak artifacts, with long-range spatial correlations. We thus ask a natural question: Can CSC models also yield strong performance on such inverse problems with non-local operators?

To deal with spatially heterogeneous imagery data, one natural strategy is to employ *multiscale* dictionaries. Indeed, seminal works have shown that multiscale dictionaries, either analytical or learned, are advantageous in representing and processing images [19]–[24]. Separating scales is useful because it gives efficient descriptions of structural correlations at different distances. Yet, these existing CSC models [10], [25]–[28] mostly employ single-scale dictionaries, whose dictionary atoms all have the same size. While there exist proposals for multiscale CSC architectures, they are tailored for specific tasks [29], [30]. In addition, CSC models do away with flexible skip connections between non-consecutive layers, which are nonetheless essential for many successful CNNs such as the U-Net and its variants [31]–[33] to fuse features across scales. This challenge of harnessing multiscale features in the CSC paradigm motivates our work.

Manuscript received August 19, 2021; revised January 17, 2022 and April 2, 2022; accepted April 7, 2022. Date of publication May 16, 2022; date of current version May 31, 2022. This work was supported by the European Research Council Starting under Grant 852821—SWING. The Associate Editor coordinating the review of this manuscript and approving it for publication was Prof. Stanley H. Chan. *(Corresponding author: Ivan Dokmanić.)*

Tianlin Liu and David Belius are with the Department of Mathematics and Computer Science, University of Basel, 4001 Basel, Switzerland (e-mail: t.liu@unibas.ch; david.belius@unibas.ch).

Anadi Chaman is with the Department of Electrical, Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: achaman2@illinois.edu).

Ivan Dokmanić is with the Department of Mathematics and Computer Science, University of Basel, 4001 Basel, Switzerland, and also with the Department of Electrical, Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: ivan.dokmanic@unibas.ch).

The code to reproduce our experiments is available at https://github.com/liutianlin0121/MUSC.

This article has supplementary downloadable material available at https://doi.org/10.1109/TCI.2022.3175309, provided by the authors.

Digital Object Identifier 10.1109/TCI.2022.3175309

TABLE I
THE NOTATIONS USED IN THIS PAPER

d	the dimension of an image
N	the dimension of a sparse code
M	the number of training samples
K	the number of sparse coding steps
$\sigma(\cdot)$	the ReLU non-linearlity
$\boldsymbol{z}$	a noisy image to be processed
$\boldsymbol{D}$	an overcomplete dictionary
$\boldsymbol{x}^\dagger$	a ground-truth image
$\widehat{\boldsymbol{x}}$	a predicted image
$\boldsymbol{\alpha}$	a sparse code
$\boldsymbol{\lambda}$	the thresholding parameters of ISTA

To address the challenge, we introduce a multiscale convolutional dictionary inspired by the highly successful U-Net [31]. We then apply the multiscale convolutional dictionary to challenging, large-scale inverse problems in imaging. The main contribution of this paper is twofold:

- We propose a new convolutional dictionary, whose representation incorporates atoms of different spatial scales. The proposed multiscale dictionary augments standard, single-scale convolutional dictionaries to exploit the spatially-heterogeneous properties of images.
- We study the effectiveness of the multiscale convolutional dictionary through experiments on large-scale datasets. We find that the performance of the multiscale CSC approach is competitive with leading CNNs on datasets including two major CT and MRI benchmarks. We additionally show that our model matches (and slightly improves) the state-of-the-art performance on the deraining task achieved by a deep neural network [34]. Notably, the single-scale CSC model performs significantly worse on this task [27].

Overall, our work makes a step forward in closing the performance gap between end-to-end CNNs and sparsity-driven dictionary models. At a meta level, it (re)validates the fundamental role of sparsity in representations of images and imaging operators [20], [35], [36].

The rest of this article is organized as follows. In Section II, we first briefly review the sparse representation model and its relationship to CNNs. Section III explains how we incorporate multiscale atoms in a dictionary model; we also explain how to learn the multiscale dictionary from data under the task-driven dictionary learning framework. Section IV reports experimental results on tasks including CT reconstruction and MRI reconstruction.

II. BACKGROUND AND RELATED WORK

In this section, we briefly review the related work; a summary of notation is given in Table I.

A. Sparse Representation Models

Sparse representation has been extensively studied and widely used in imaging inverse problems [37]–[39]. It is motivated by the idea that many signals, images being a prime example, can be approximated by a linear combination of a few elements from a suitable overcomplete basis. The sparse representation framework posits that we can decompose a signal of interest[1] $\boldsymbol{z} \in \mathbb{R}^d$ as $\boldsymbol{z} = \boldsymbol{D}\boldsymbol{\alpha}$, where $\boldsymbol{D} \in \mathbb{R}^{d\times N}$ is an overcomplete dictionary of N atoms ($N > d$) and $\boldsymbol{\alpha} \in \mathbb{R}^N$ is a sparse vector with few non-zero entries. Learning a sparse representation model thus comprises two sub-problems: (i) given a dictionary $\boldsymbol{D}$, encode the signal $\boldsymbol{z}$ into a sparse vector $\boldsymbol{\alpha}$ (the *sparse coding* problem), and (ii) given a set of signals, learn an appropriate dictionary $\boldsymbol{D}$ that sparsifies them (the *dictionary learning* problem). We briefly review these two problems and show how they are related to neural network models such as CNNs.

B. The Sparse Coding Problem

The sparse coding problem is often formulated as basis pursuit denoising [40] or Lasso regression [41]. Most relevant to our work is its formulation with non-negative constraints on the sparse code $\boldsymbol{\alpha}$:

$$\underset{\boldsymbol{\alpha}\geq 0}{\text{minimize}} \ \frac{1}{2}\left\|\boldsymbol{z}-\boldsymbol{D}\boldsymbol{\alpha}\right\|_2^2 + \lambda\left\|\boldsymbol{\alpha}\right\|_1. \tag{1}$$

Here, the first term $\frac{1}{2}\|\boldsymbol{z}-\boldsymbol{D}\boldsymbol{\alpha}\|_2^2$ ensures that the code $\boldsymbol{\alpha}$ yields a faithful representation of $\boldsymbol{z}$, the second term $\lambda\|\boldsymbol{\alpha}\|_1$ controls the sparsity of the code, and the two terms are balanced by a parameter $\lambda > 0$. An effective solver for the minimization problem (1) is the iterative shrinkage-thresholding algorithm (ISTA) [42], which executes the following iteration

$$\begin{aligned}\boldsymbol{\alpha}^{[k+1]} &:= \mathcal{S}(\boldsymbol{\alpha}^{[k]}, \boldsymbol{z}; \boldsymbol{D}, \boldsymbol{\lambda})\\ &:= \sigma(\boldsymbol{\alpha}^{[k]} + \eta\boldsymbol{D}^\top(\boldsymbol{z}-\boldsymbol{D}\boldsymbol{\alpha}^{[k]}) - \eta\boldsymbol{\lambda}),\end{aligned} \tag{2}$$

where the superscript $[k]$ denotes the iteration number, η is a step-size parameter, $\boldsymbol{\lambda}$ is a vector whose entries are all λ, and $\sigma(x) := \max(x, 0)$ is a component-wise rectifier function. For simplicity, we use $\mathcal{S}(\boldsymbol{\alpha}, \boldsymbol{z}; \boldsymbol{D}, \boldsymbol{\lambda})$ to denote one execution of ISTA with measurement $\boldsymbol{z}$, sparse code $\boldsymbol{\alpha}$, dictionary $\boldsymbol{D}$, and threshold $\boldsymbol{\lambda}$. The ISTA algorithm is a composition of such executions; we write ISTA_K for the K-fold composition of $\mathcal{S}$ with itself:

$$\begin{aligned}&\text{ISTA}_K(\boldsymbol{z}; \boldsymbol{D}, \boldsymbol{\lambda})\\ &:= \underbrace{\Big(\mathcal{S}(\cdot, \boldsymbol{z}; \boldsymbol{D}, \boldsymbol{\lambda}) \circ \cdots \circ \mathcal{S}(\cdot, \boldsymbol{z}; \boldsymbol{D}, \boldsymbol{\lambda})\Big)}_{K \text{ times}}(\boldsymbol{\alpha}^{[0]}),\end{aligned} \tag{3}$$

where $\boldsymbol{\alpha}^{[0]}$ is the initial sparse code; throughout this work, this initial code $\boldsymbol{\alpha}^{[0]}$ is assumed to contain zero in all entries. We emphasize that ISTA is a nonlinear transform of its input $\boldsymbol{z}$.

C. The Task-Driven Dictionary Learning Problem

We now briefly recall the task-driven dictionary learning framework [43]. Consider a supervised learning setting, in which we aim to identify a parametric function that associates each input $\boldsymbol{z}$ (e.g., a corrupted image) with its target $\boldsymbol{x}^\dagger$ (e.g., a

[1]For simplicity we write all signals as 1 d vectors, but the formulation is valid in any dimension.

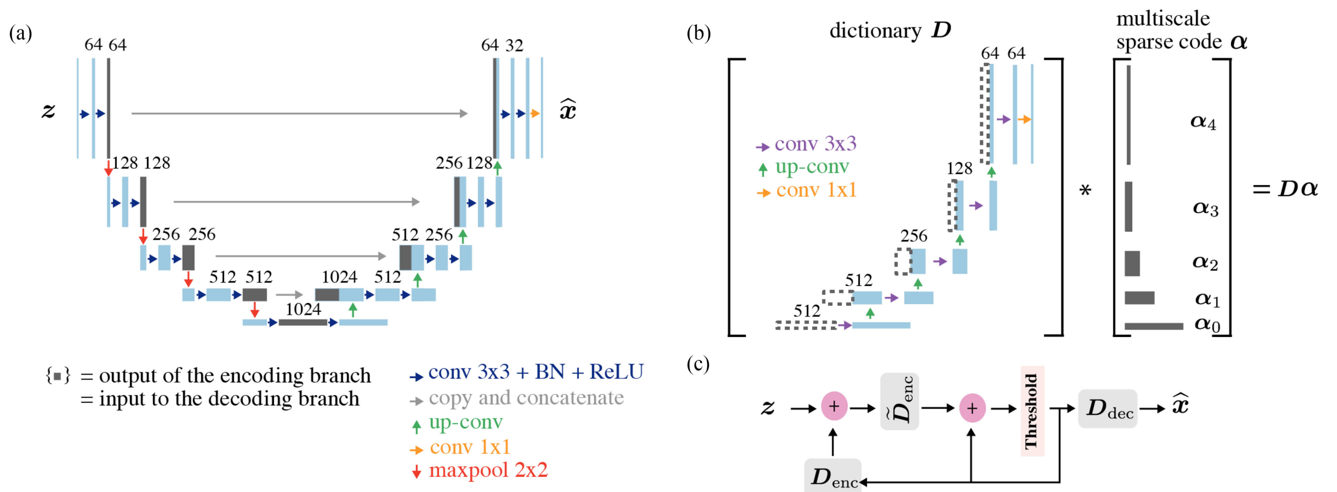


Fig. 1. Schematic illustration of the U-Net (left panel) and the dictionary model considered in this work (right panel). (a): The U-Net processes input images using convolution, scale separations, and skip connections in conjunction with ReLU non-linearities and batch-normalization (BN) modules indicated by colored arrows. The multi-channel feature maps produced by these operations are illustrated as boxes with the channel numbers indicated on the top of these boxes. Dark grey boxes indicate the feature maps produced by the encoding branch of the U-Net, which are sent to the decoding branch either through channel-wise concatenation ("skip connection") or through the bottleneck layer. (**b**): The dictionary considered in this work is a simplification of the decoder branch of the U-Net: We retain convolution and multiscale representation from the decoder branch of the U-Net but remove all non-linearities, batch-normalization, and additive biases; additionally, we remove a convolution at each spatial resolution level and halve the number of convolutional channels for all convolutions. Grey boxes indicate the multiscale sparse code $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_0, \ldots, \boldsymbol{\alpha}_4)$ that the dictionary takes as input. Dashed boxes indicate the position that each $\boldsymbol{\alpha}_i$ feed into the dictionary. (**c**): The proposed as a computational graph that uses multiscale dictionaries $\boldsymbol{D}_{\text{enc}}$, $\widetilde{\boldsymbol{D}}_{\text{enc}}$, and $\boldsymbol{D}_{\text{dec}}$; although each dictionary is linear, the computational graph is nonlinear due to the thresholding operator.

clean image) for all $(\boldsymbol{z}, \boldsymbol{x}^\dagger) \in \mathbb{R}^d \times \mathbb{R}^d$ drawn from some joint distribution. In the task-driven framework, we proceed by first representing the signal $\boldsymbol{z}$ by a sparse code $\boldsymbol{\alpha}_{\boldsymbol{z}}$ with respect to a dictionary $\boldsymbol{D}$. One way to achieve this is to let

$$\boldsymbol{\alpha}_{\boldsymbol{z}} := \arg\min_{\boldsymbol{\alpha} \geq 0} \frac{1}{2} \|\boldsymbol{z} - \boldsymbol{D}\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1, \tag{4}$$

which can be approximated by K iterations of ISTA as in (3). Next, we approximate the desired target $\boldsymbol{x}^\dagger$ using the sparse code $\boldsymbol{\alpha}_{\boldsymbol{z}}$ through a regression model $f(\cdot, \boldsymbol{w})$ with learnable parameter $\boldsymbol{w}$. For instance, $f(\cdot, \boldsymbol{w})$ could be a linear regression model with weights and biases $\boldsymbol{w}$. The model output $f(\boldsymbol{\alpha}_{\boldsymbol{z}}, \boldsymbol{w})$ thus depends on the regression model parameters $\boldsymbol{w}$ as well as the sparse code $\boldsymbol{\alpha}_{\boldsymbol{z}}$, which in turn depends on the dictionary $\boldsymbol{D}$ through the ISTA iterations. In this way, the regression parameters $\boldsymbol{w}$ and dictionary $\boldsymbol{D}$ can be *jointly* optimized, for instance, with respect to the quadratic loss objective evaluated on a dataset of M input-target pairs $\{(\boldsymbol{z}_i, \boldsymbol{x}_i^\dagger)\}_{i=1}^M$:

$$\underset{\boldsymbol{w}, \boldsymbol{D}}{\text{minimize}} \ \frac{1}{2\,M} \sum_{i=1}^{M} \|f(\boldsymbol{\alpha}_{\boldsymbol{z}_i}, \boldsymbol{w}) - \boldsymbol{x}_i^\dagger\|_2^2. \tag{5}$$

Importantly, the task-driven objective in (5) implies that the dictionary $\boldsymbol{D}$ is optimized to solve the supervised learning task and not just to sparsely represent data.

D. Convolutional Sparse Coding

Our work is inspired by the convolutional sparse coding (CSC) model [8], [44]–[47], which bridges deep CNNs and sparse representation models. Concretely, Papyan *et al.* [8] noticed that if the dictionary $\boldsymbol{D}$ has a convolutional structure and if the sparse code $\boldsymbol{\alpha}$ is assumed to be non-negative, a single iteration of ISTA with $\boldsymbol{\alpha}^{[0]}$ initialized as a zero vector and step-size $\eta = 1$ is equivalent to the forward pass of a single-layer convolutional network

$$\boldsymbol{\alpha} = \sigma(\boldsymbol{D}^\top \boldsymbol{z} + \boldsymbol{b}), \tag{6}$$

where $\boldsymbol{b}$ is a vector whose components are $-\lambda$ (cf. Equation (2)). This single-layer formulation can be extended to characterize a deep CNN of multiple layers. Specifically, the forward-pass of a deep CNN of L-layers can be interpreted to approximate the sparse codes $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_L$ sequentially with respect to different dictionaries $\boldsymbol{D}_1, \ldots, \boldsymbol{D}_L$; the back-propagation pass is interpreted as an update to these dictionaries $\{\boldsymbol{D}_i\}_{i=1}^L$ in a task-driven way.

E. CNNs for Solving Inverse Problems

Deep CNNs achieve state-of-the-art performance in many image processing tasks [48]–[51]. In particular, the U-Net [31] and its variants [32], [33], [52] are among the most extensively used CNN architectures in solving image-to-image learning tasks. U-Nets represent images via multiscale features computed from measurements using an encoding (or downsampling) branch and a synthesized into an estimated image in a decoding (or upsampling) branch (Fig. 1**a**). In the downsampling branch, the spatial resolutions of feature maps are reduced while the number of feature maps is increased; in the decoding branch, these features are recombined with previous high-resolution features via channel concatenation ("skip connections") and convolution. Heuristically, low-resolution feature maps of a U-Net capture large-scale image properties, whereas the high-resolution feature maps capture more fine-grained image properties [52]. In a related line of work, Ye *et al.* [53]–[55] proposed to use the framelets formalism [56] to study aspects of U-Net-like encoder-decoder CNNs. A key observation they make is that a U-Net model is closely related to convolutional framelets whose frame basis selection depends non-linearly on input data.

III. CSC With Multiscale Dictionaries

The structure of a convolutional dictionary is crucial to a CSC model since the dictionary atoms characterize the signals that can be represented sparsely. In the existing formulation of CSC, atoms of a convolutional dictionary have a single scale, in the sense that they all share the same spatial shape. However, many image classes and imaging artifacts exhibit structured correlations over multiple scales. To exploit these correlations in imaging inverse problems, we construct multiscale convolutional dictionaries.

Our construction is based on the U-Net reviewed in Section II. Indeed, the tremendous success of U-Nets has in part been attributed to their ability to represent images at multiple scales [33], [55], which is achieved by using up- and downsampling operations together with skip connections as in Fig. 1**a**. Another property of the U-Net is its shared parameters across scales: Low-resolution features (the grey boxes at the bottom of Fig. 1**a**) and high-resolution features (the grey boxes at the top of Fig. 1**a**) undergo an overlapping synthesizing path parameterized by shared weights. This weight-sharing strategy has not been employed by existing proposals for multiscale CSC dictionaries [29], [30]. In what follows, we describe the construction process of a *linear* dictionary inspired by and closely following the standard U-Net.

A. Encoder–Decoder Dictionaries

We denote the encoding branch of the U-Net by $f_{\text{enc}}(\cdot, \boldsymbol{\theta}_{\text{enc}}) : \mathbb{R}^d \to \mathbb{R}^N$ with parameters $\boldsymbol{\theta}_{\text{enc}}$; the encoding branch maps the input $\boldsymbol{z} \in \mathbb{R}^d$ to convolutional feature maps $\boldsymbol{\alpha}_{\boldsymbol{z}} = f_{\text{enc}}(\boldsymbol{z}, \boldsymbol{\theta}_{\text{enc}}) \in \mathbb{R}^N$, illustrated as the dark grey boxes in Fig. 1**a**. Note that, for a U-Net, the intermediate feature map dimension N (number of scalar coefficients in $\boldsymbol{\alpha}$) is typically much greater than the image dimension d. These feature maps are then fed into the decoding branch of the U-Net either through skip connections or through the bottleneck layer. To describe this process, we write the decoding branch of the U-Net as a function $f_{\text{dec}}(\cdot, \boldsymbol{\theta}_{\text{dec}}) : \mathbb{R}^N \to \mathbb{R}^d$ with parameters $\boldsymbol{\theta}_{\text{dec}}$. That is, the function $f_{\text{dec}}(\cdot, \boldsymbol{\theta}_{\text{dec}})$ takes the convolutional feature maps produced by the encoding branch and transforms them to produce the model output. We can thus write the output produced by a U-Net as

$$\widehat{\boldsymbol{z}} := f_{\text{dec}}(\boldsymbol{\alpha}_{\boldsymbol{z}}, \boldsymbol{\theta}_{\text{dec}}) = f_{\text{dec}}\left(f_{\text{enc}}(\boldsymbol{z}, \boldsymbol{\theta}_{\text{enc}}), \boldsymbol{\theta}_{\text{dec}}\right).$$

We now focus on the image synthesis process of the U-Net, described by the decoding function $f_{\text{dec}}(\cdot, \boldsymbol{\theta}_{\text{dec}})$. This function synthesizes convolutional feature maps at different spatial scales through skip connections and upsampling. As such, the decoding branch of the U-Net approximates an image $\boldsymbol{x}^\dagger \in \mathbb{R}^d$ using multiscale feature maps $\boldsymbol{\alpha}_{\boldsymbol{z}} \in \mathbb{R}^N$ of a much higher dimension, so that $\boldsymbol{x}^\dagger \approx f_{\text{dec}}(\boldsymbol{\alpha}_{\boldsymbol{z}}, \boldsymbol{\theta}_{\text{dec}})$. Conceptually, this representation is similar to the sparse and overcomplete representation in a dictionary, except that the U-Net decoder is non-linear.

To construct a multiscale dictionary, we thus consider a stripped-down version of the image synthesis process of U-Net by removing all non-linearities, batch normalization, and additive biases from the function $f_{\text{dec}}(\cdot, \boldsymbol{\theta}_{\text{dec}})$, as shown in Fig. 1**b**; to further simplify the architecture, at each spatial scale, we additionally remove a convolution and halve the number of convolutional channels for all convolutions. The resulting function is then simply a *linear* transformation

$$\boldsymbol{\alpha} := (\boldsymbol{\alpha}_0, \ldots, \boldsymbol{\alpha}_4) \mapsto \boldsymbol{D}_{\text{dec}}\boldsymbol{\alpha}, \tag{7}$$

where $\boldsymbol{\alpha}_0, \ldots, \boldsymbol{\alpha}_4$ are sparse code having different resolutions (visualized as the grey boxes in Fig. 1**b**). This dictionary shares the essential ingredients of convolution, multiscale representation, and skip connections with the U-Net decoding branch and therefore we refer to it as the decoder dictionary. A precise definition of the decoder dictionary $\boldsymbol{D}_{\text{dec}}$ through convolution and upsampling is provided in Appendix A.

B. The Dictionary-Based Sparsity Prior

With a given decoder dictionary $\boldsymbol{D}_{\text{dec}}$ to describe the image synthesis process, we next consider how to infer an associated sparse code $\boldsymbol{\alpha}$, so that $\boldsymbol{D}_{\text{dec}}\boldsymbol{\alpha}$ is a good approximation of the image we wish to model. In a supervised learning setting where the input image $\boldsymbol{z}$ is given, it is natural to interpret $\boldsymbol{\alpha}$ as an encoded representation of $\boldsymbol{z}$. Since the encoding must produce a coefficient vector whose structure is compatible with $\boldsymbol{\alpha}$, we endow an encoder dictionary $\boldsymbol{D}_{\text{enc}} \in \mathbb{R}^{d\times N}$ with the same structure of $\boldsymbol{D}_{\text{dec}}$ albeit with a different set of atoms. This setup is analogous to U-Net's encoding and decoding branches: the encoder and decoder dictionaries $\boldsymbol{D}_{\text{enc}}$ and $\boldsymbol{D}_{\text{dec}}$ are employed to process input signals and produce output signals, respectively. The sparse code $\boldsymbol{\alpha}_{\boldsymbol{z}}$ induced by an input $\boldsymbol{z}$ and the encoder dictionary $\boldsymbol{D}_{\text{enc}}$ then facilitate the subsequent task for approximating the ground-truth image $\boldsymbol{x}$:

$$\boldsymbol{z} \xrightarrow{\text{Sparse coding with } \boldsymbol{D}_{\text{enc}}} \boldsymbol{\alpha}_{\boldsymbol{z}} \xrightarrow{\text{Synthesis with } \boldsymbol{D}_{\text{dec}}} \widehat{\boldsymbol{x}} := \boldsymbol{D}_{\text{dec}}\boldsymbol{\alpha}_{\boldsymbol{z}}. \tag{8}$$

In what follows, we derive a supervised learning method that turns each $\boldsymbol{z}$ into a prediction $\widehat{\boldsymbol{x}}$ using encoder and decoder dictionaries.

C. The Task-Driven Dictionary Learning Objective

Under the task-driven framework introduced in Section II, we formulate a supervised learning problem via sparse coding and dictionary learning. We consider the following minimization problem over a dataset of M input-target pairs $\{(\boldsymbol{z}_i, \boldsymbol{x}_i^\dagger)\}_{i=1}^M$:

$$\underset{\{\boldsymbol{D}_{\text{enc}}, \boldsymbol{D}_{\text{dec}}\}, \boldsymbol{\lambda}>0}{\text{minimize}} \quad \frac{1}{2M}\sum_{i=1}^{M} \|\boldsymbol{D}_{\text{dec}}\boldsymbol{\alpha}_{\boldsymbol{z}_i} - \boldsymbol{x}_i^\dagger\|_2^2$$
$$\text{where} \quad \boldsymbol{\alpha}_{\boldsymbol{z}_i} := \text{ISTA}_K(\boldsymbol{z}_i; \boldsymbol{D}_{\text{enc}}, \boldsymbol{\lambda}). \tag{9}$$

The objective in (9) penalizes the discrepancy between the ground-truth signal $\boldsymbol{x}^\dagger$ and the model prediction $\boldsymbol{D}_{\text{dec}}\boldsymbol{\alpha}_{\boldsymbol{z}}$, where the latter is a signal synthesized from a sparse code $\boldsymbol{\alpha}_{\boldsymbol{z}}$ via the decoder dictionary $\boldsymbol{D}_{\text{dec}}$; the code $\boldsymbol{\alpha}_{\boldsymbol{z}}$ is a sparse representation of the input image $\boldsymbol{z}$ with respect to the encoder dictionary $\boldsymbol{D}_{\text{enc}}$ by unrolling a fixed number K of ISTA iterations. The sparsity-controlling parameter $\boldsymbol{\lambda}$ is multi-dimensional, weighting codes component-wise. The intuition behind this choice is that the different convolutional features, especially those at different scales, should be thresholded differently. The sparse code $\boldsymbol{\alpha}$, illustrated as the grey boxes in Fig. 1**b**, is a collection of multi-dimensional tensors, each corresponds a spatial scale.

The task driven objective (9) defines a computational graph that transforms an input image $\boldsymbol{z}$ into a prediction $\boldsymbol{D}_{\text{dec}}\boldsymbol{\alpha}_{\boldsymbol{z}}$. We term this computational graph MUSC, since it involves **mu**ltiscale **U**-Net-like **s**parse **c**oding. We note the MUSC is an instance of *optimization-driven networks* [26] derived by unrolling an optimization algorithm. It incorporates two modules with meaningful objectives, one implementing sparse coding and the other dictionary-based synthesis. This composition is arguably conceptually more interpretable than end-to-end layerwise composition of deep networks.

While a traditional compressed sensing approach uses *a single dictionary* for reconstruction, our approach uses *two dictionaries* $\boldsymbol{D}_{\text{enc}}$ and $\boldsymbol{D}_{\text{dec}}$ in the task-driven learning objective (9). This discrepancy is due to different assumptions in measurement-to-image reconstruction (the compressed sensing approach) and image-to-image reconstruction (our approach). Consider an inverse problem with a forward operator $\boldsymbol{A}$, a unknown ground-truth signal $\boldsymbol{x}^{\dagger}$, and measurements $\boldsymbol{y} := \boldsymbol{A}\boldsymbol{x}^{\dagger}$; in CT reconstruction, $\boldsymbol{A}$ is the Radon transform and $\boldsymbol{y}$ is the measured sinogram. The compressed sensing approach estimates $\boldsymbol{x}^{\dagger}$ as $\boldsymbol{D}\boldsymbol{\alpha}^{*}$ for some dictionary $\boldsymbol{D}$, where

$$\boldsymbol{\alpha}^{*} = \arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{A}\boldsymbol{D}\boldsymbol{\alpha} - \boldsymbol{y}\|_2 + \lambda \|\boldsymbol{\alpha}\|_1 \tag{10}$$

is the inferred sparse code based on the dictionary $\boldsymbol{D}$. Note that (10) and the synthesis $\boldsymbol{D}\boldsymbol{\alpha}^{*}$ require only a single dictionary $\boldsymbol{D}$. However, this approach assumes that we know the measurements $\boldsymbol{y}$ and the forward operator $\boldsymbol{A}$.

If we were to apply a single dictionary $\boldsymbol{D} := \boldsymbol{D}_{\text{enc}} = \boldsymbol{D}_{\text{dec}}$ in our image-to-image learning approach in (9), we would find a sparse code $\boldsymbol{\alpha}$ such that $\boldsymbol{D}\boldsymbol{\alpha} \approx \boldsymbol{x}^{\dagger}$ and $\boldsymbol{D}\boldsymbol{\alpha} \approx \boldsymbol{A}^{+}\boldsymbol{A}\boldsymbol{x}^{\dagger}$. This is difficult when $\boldsymbol{A}^{+}\boldsymbol{A}$ significantly differs from the identity operator as in the case of highly ill-posed problems. On the other hand, using two dictionaries $\boldsymbol{D}_{\text{enc}}$ and $\boldsymbol{D}_{\text{dec}}$ in (9) requires finding a sparse code $\boldsymbol{\alpha}$ such that $\boldsymbol{D}_{\text{dec}}\boldsymbol{\alpha} \approx \boldsymbol{x}^{\dagger}$ and $\boldsymbol{D}_{\text{enc}}\boldsymbol{\alpha} \approx \boldsymbol{A}^{+}\boldsymbol{A}\boldsymbol{x}^{\dagger}$, a formulation that is more flexible when $\boldsymbol{A}^{+}\boldsymbol{A}$ substantially differs from the identity. Experiments in Section IV-E confirm that allowing $\boldsymbol{D}_{\text{enc}} \neq \boldsymbol{D}_{\text{dec}}$ yields better performance. We note that our approach is morally related to setting $\boldsymbol{D}_{\text{enc}} = \boldsymbol{A}\boldsymbol{D}$ in (10), but since we do not know $\boldsymbol{A}$ we have to learn $\boldsymbol{D}_{\text{enc}}$ from samples together with $\boldsymbol{D}_{\text{dec}}$. Such a learned encoder dictionary captures information about $\boldsymbol{A}$, entangled with information about the data distribution.

D. Relaxation on Dictionaries

We now describe computational techniques that stabilize the gradient-descent-based dictionary learning of MUSC. Following earlier work [6], [10], [25], [26], [57], we untie the encoder dictionary from its adjoint during dictionary update. That is, we replace the execution in (2) by

$$\widetilde{\mathcal{S}}(\boldsymbol{\alpha}, \boldsymbol{z}; \boldsymbol{D}_{\text{enc}}, \widetilde{\boldsymbol{D}}_{\text{enc}}, \boldsymbol{\lambda}) := \sigma(\boldsymbol{\alpha} + \eta \widetilde{\boldsymbol{D}}_{\text{enc}}^{\top}(\boldsymbol{z} - \boldsymbol{D}_{\text{enc}}\boldsymbol{\alpha}) - \eta\boldsymbol{\lambda}), \tag{11}$$

where the dictionary $\widetilde{\boldsymbol{D}}_{\text{enc}}$ is initialized to be identical to $\boldsymbol{D}_{\text{enc}}$ but is allowed to evolve independently during training. Even though the theoretical effects of this relaxation remain unclear, the dictionary $\widetilde{\boldsymbol{D}}_{\text{enc}}$ can be interpreted as a learned preconditioner that accelerates training [25], [26]; see also the investigation in [6], [58], [59]. The *learned ISTA* (LISTA) algorithm [57] corresponding to (11) is written as

$$\text{LISTA}_K(\boldsymbol{z}; \boldsymbol{D}_{\text{enc}}, \widetilde{\boldsymbol{D}}_{\text{enc}}, \boldsymbol{\lambda}) := \underbrace{\Big(\widetilde{\mathcal{S}}(\cdot, \boldsymbol{z}; \boldsymbol{D}_{\text{enc}}, \widetilde{\boldsymbol{D}}_{\text{enc}}, \boldsymbol{\lambda}_K) \circ \cdots \circ \widetilde{\mathcal{S}}(\cdot, \boldsymbol{z}; \boldsymbol{D}_{\text{enc}}, \widetilde{\boldsymbol{D}}_{\text{enc}}, \boldsymbol{\lambda}_1)\Big)}_{K \text{ times}}(\boldsymbol{\alpha}^{[0]}), \tag{12}$$

where $\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_K$ are the soft-thresholding parameters for each ISTA execution. Note that, in (12), the soft-thresholding parameters $\{\boldsymbol{\lambda}_i\}_{i=1}^{K}$ depend on the execution step. As shown in [6], incorporating step-dependent soft-thresholding parameters can be beneficial. While [6] uses a homotopy continuation strategy to adjust these parameters we treat them as learnable parameters for simplicity. Taking these considerations into account, we define a new regression loss:

$$\mathcal{L}(\boldsymbol{D}_{\text{enc}}, \boldsymbol{D}_{\text{dec}}, \boldsymbol{\lambda}) := \frac{1}{2M}\sum_{i=1}^{M} \|\boldsymbol{D}_{\text{dec}}\boldsymbol{\alpha}_{\boldsymbol{z}_i} - \boldsymbol{x}_i^{\dagger}\|_2^2,$$
$$\text{where} \quad \boldsymbol{\alpha}_{\boldsymbol{z}_i} = \text{LISTA}_K(\boldsymbol{z}_i; \boldsymbol{D}_{\text{enc}}, \widetilde{\boldsymbol{D}}_{\text{enc}}, \boldsymbol{\lambda}). \tag{13}$$

Unless mentioned otherwise, we use the loss (13) to train MUSC throughout our paper. In Section IV-E, we compare the performance of trained model using (13) and (9).

E. Training the MUSC

Training the MUSC entails the following three steps:

1) *Dictionary initialization:* We randomly initialize the dictionary $\boldsymbol{D}_{\text{enc}}$ and initialize $\boldsymbol{D}_{\text{dec}}$, and $\widetilde{\boldsymbol{D}}_{\text{enc}}$ as identical copies of $\boldsymbol{D}_{\text{enc}}$.
2) *Model forward pass:* For each input image $\boldsymbol{z}_i$, we evaluate the model prediction $\boldsymbol{D}_{\text{dec}}\boldsymbol{\alpha}_{\boldsymbol{z}_i}$ as in (13). For ISTA executions, we initialize all sparse code $\boldsymbol{\alpha}_{\boldsymbol{z}}^{[0]}$ as a collection of all-zero tensors; the ISTA step-size parameter η is initialized as the inverse of the dominant eigenvalue of the matrix $\boldsymbol{D}_{\text{enc}}^{\top}\boldsymbol{D}_{\text{enc}}$, which can be approximated using by power iteration (Appendix C).
3) *Task-driven dictionary learning:* For a mini-batch of input-target pairs, solve the optimization problem in (13) with gradient descent.

IV. Experiments

We report the performance of MUSC on deraining, CT reconstruction, and MRI reconstruction tasks. The motivations for choosing these tasks are as follows. First, we note that single-scale CSC models have recently been applied to the deraining task, achieving performance slightly worse than state-of-the-art deep networks [27]; we thus aim to test the capability of our multiscale approach on the same task. We additionally choose CT and MRI reconstruction tasks as there exist challenging, large-scale, and up-to-date benchmark datasets for these tasks. Two such datasets that we use are the LoDoPaB-CT [64] and the fastMRI [65]. An additional strength of these two datasets is that the model evaluation process is carefully controlled: The evaluation on the challenge fold (for LoDoPaB-CT) or the test fold (for fastMRI) is restricted through an online submission portal with the ground truth hidden from the public. As a result, overfitting to these evaluation folds is difficult and quantitative comparisons are transparent.

Throughout this section, we use the MUSC architecture whose encoder and decoder dictionaries are displayed in Fig. 1**b** and mathematically defined in Appendix A. Hyper-parameter choices for the experiments are provided in Appendix D. For each task, we use well-known CNN models as baselines. We note that, for the CT and MRI reconstruction tasks, there are two major approaches to employ CNNs. In the first, *model-based* approach, one applies neural networks on raw measurement data (sinogram data in CT and k-space data in MRI) by embedding a task-dependent forward operator (the Radon transform for CT and the Fourier transform for MRI) into multiple layers or iterations of the network. Learning methods of this approach can be highly performant at the cost of being computationally expensive, especially during training, since one needs to apply the forward operator (and the adjoint of its derivative) repeatedly [49]. In the second, *model-free* approach, the (pseudoinverse of the) forward operator is used at most *once* during data preprocessing and is never used during subsequent supervised training. These preprocessed images contain imaging artifacts. During supervised learning, one applies a CNN directly on these preprocessed images. The proposed MUSC is in this sense a model-free approach and we compare it to model-free baselines. We note that in this case one does not need to know the forward operator at all. The leading model-free baseline CNN methods in this approach are typically U-Net variants tuned to the task at hand. For a more thorough comparison, we also implemented the original U-Net architecture proposed in [31] (schematically illustrated in Fig. 1**a**) in these tasks as additional baselines.

While model-free approaches perform somewhat worse than model-based ones, our purpose here is to show that a general-purpose multiscale convolutional model can perform as well as convolutional neural networks *ceteris paribus*, rather than to propose state-of-the-art reconstruction algorithms for specific problems. This general-purpose approach further allows us to tackle structured denoising problems such as deraining where the forward operator is simply the identity.

A. Deraining

Image deraining aims to remove rain streaks from an image. Formally, a rainy image $\boldsymbol{z}$ is expressed as $\boldsymbol{z} = \boldsymbol{x}^\dagger + \boldsymbol{s}$, where $\boldsymbol{x}^\dagger$ is a clean image and $\boldsymbol{s}$ is the rain streaks component. The goal is to reconstruct the clean image $\boldsymbol{x}^\dagger$ based on the rainy image $\boldsymbol{z}$. Recently, single-scale CSC models have been applied to the draining task [27]. Despite theoretical progress, these single-scale CSC models still fall short competing with leading deep learning models, as remarked in [27]. In this section, we demonstrate that our multiscale CSC model closes this performance gap.

Throughout this subsection, we follow the experiment setup of [27]. We use 200 clean and rainy image pairs as the training dataset. A rainy image is created by adding synthesized rain streaks to its clean version. We use two test sets, Rain12 [60] and Rain100L [63], to benchmark our results. Similar to [27], we train our model to restore rain streaks based on rainy images; a derained image is then the difference between a rainy image and the restored rain streaks. To be consistent with [27], [34], [63], the evaluation result is calculated after transforming the image into the luma component in the YCbCr domain using the software provided by [34]. Additional details of the experiment are provided in Appendix D.

We report the reconstruction performance in Table II and visualize the reconstruction results in Fig. 2. Our multiscale convolutional dictionary approach matches or outperforms baseline methods. Notably, it improves upon the LGM method (the single-scale CSC approach of [27]) by a non-trivial margin.

B. CT Reconstruction

Computed tomography (CT) aims to recover images from their sparse-view sinograms. We use the LoDoPaB-CT dataset [64] to benchmark our results. This dataset contains more than 40000 pairs of human chest CT images and their simulated low photon count measurements. The ground truth images of this dataset are human chest CT scans corresponding to the LIDC/IDRI dataset [66], cropped to 362×362 pixels. The low-dose projections are simulated using the default setting of [64].

To train our MUSC, we use the default dataset split as recommended in [64]: The dataset is divided into 35820 training samples, 3522 validation samples, 3553 test samples, and 3678 challenge samples. Here, the ground-truth samples from the challenge dataset are hidden from the public; the evaluation on this fold is performed through the online submission system of the LoDoPaB-CT challenge[2].

We compare the reconstruction performance of MUSCs with five modern CNN baselines, namely CINN [67], U-Net++ [68], MS-D-CNN [69], U-Net [31], and LoDoPaB U-Net [64]; the LoDoPaB U-Net refers to a modified U-Net architecture tailored to the LoDoPaB-CT task. Fig. 3 shows the reconstruction results of a test sample. In Table III, we quantitatively compare MUSC with two classic methods (FBP and TV) together with five CNN baseline methods mentioned above. As shown in Table III, MUSC outperforms all baselines. The metrics PSNR and PSNR-FR are taken from [49]: For a ground-truth signal $\boldsymbol{x}^\dagger$

[2]https://lodopab.grand-challenge.org/challenge/

TABLE II
PERFORMANCE ON THE DERAINING TEST SET. BOLDFACE INDICATES THE BEST PERFORMANCE; SECOND-BEST RESULTS ARE HIGHLIGHTED IN GREY. ALL RESULTS ARE COLLECTED FROM [34] AND [27] EXCEPT MUSC.

	Rain12		Rain100L	
	PSNR	SSIM	PSNR	SSIM
LP [60]	32.02	0.91	29.11	0.88
DDN [61]	31.78	0.90	32.16	0.94
DSC [62]	30.02	0.87	24.16	0.87
JORDER [63]	33.92	0.95	36.61	0.97
PReNet [34]	36.69	**0.96**	37.10	**0.98**
LGM (single-scale CSC; [27])	35.46	0.95	34.07	0.96
MUSC (multiscale CSC; ours)	**36.77**	**0.96**	**37.25**	**0.98**

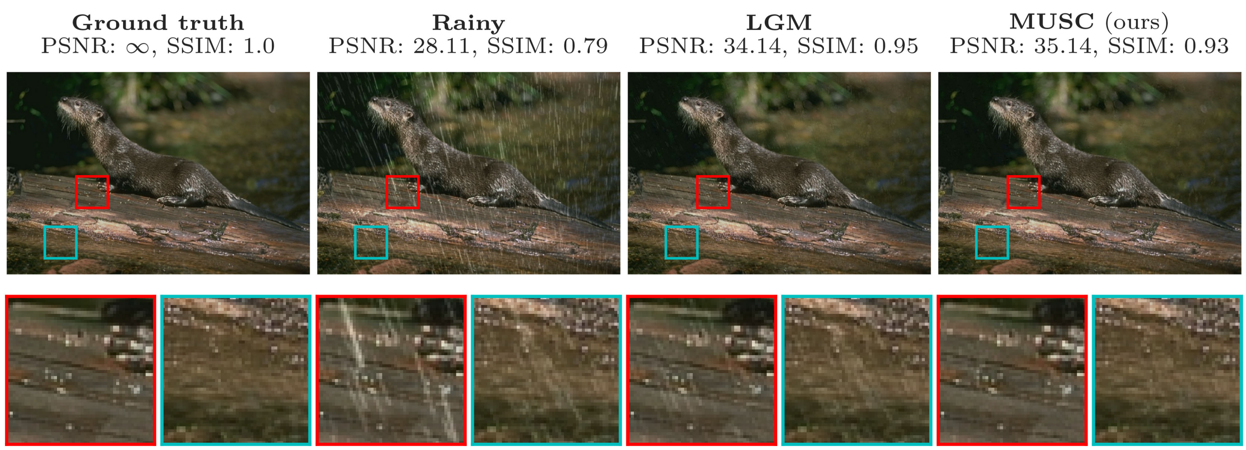


Fig. 2. Reconstructions of a test sample from the Rain12 dataset.

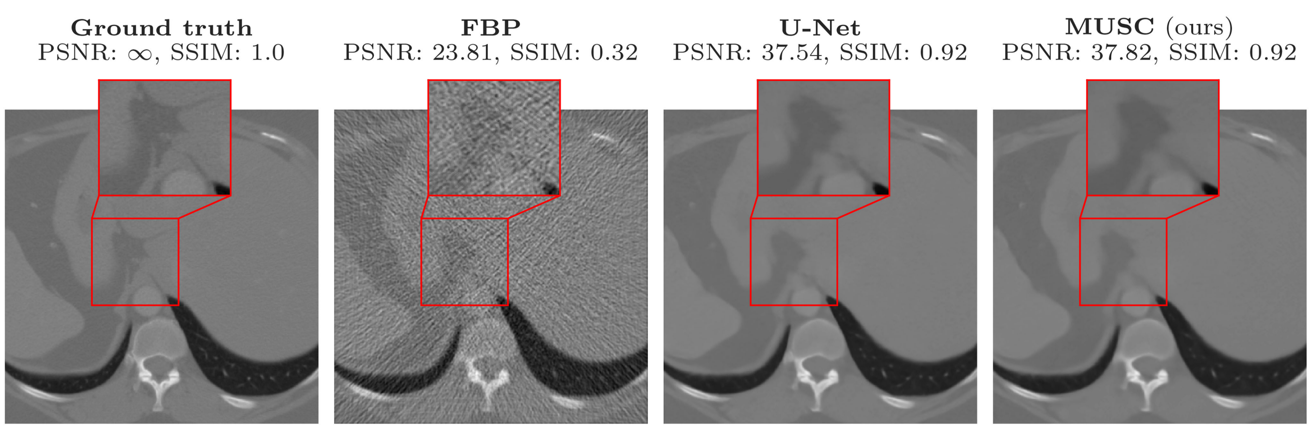


Fig. 3. Reconstructions of a test sample from the LoDoPaB-CT dataset.

and its approximation $\widehat{\boldsymbol{x}}$, we define

$$\mathrm{PSNR}\left(\widehat{\boldsymbol{x}}, \boldsymbol{x}^{\dagger}\right) := 10 \log_{10}\left(\frac{\max(\boldsymbol{x}^{\dagger}) - \min(\boldsymbol{x}^{\dagger})}{\mathrm{MSE}\left(\widehat{\boldsymbol{x}}, \boldsymbol{x}^{\dagger}\right)}\right),$$

$$\mathrm{PSNR\text{-}FR}\left(\widehat{\boldsymbol{x}}, \boldsymbol{x}^{\dagger}\right) := 10 \log_{10}\left(\frac{1}{\mathrm{MSE}\left(\widehat{\boldsymbol{x}}, \boldsymbol{x}^{\dagger}\right)}\right).$$

C. MRI Reconstruction

We further considered the task of accelerated magnetic resonance imaging (MRI) reconstruction using the fastMRI dataset [65] procured by Facebook and NYU. Specifically, we used the single-coil knee dataset with a 4-fold acceleration factor. This dataset contains 973 volumes or 34742 slices in the training set, 199 volumes or 7135 slices in the validation set, and 108 volumes or 3903 slices in the test set. The ground-truth

TABLE III
PERFORMANCE ON THE LODOPAB-CT CHALLENGE SET. ALL VALUES ARE TAKEN FROM THE OFFICIAL CHALLENGE LEADERBOARD

	Number of parameters	PSNR	PSNR-FR	SSIM	SSIM-FR
FBP	-	30.19 ± 2.55	34.46 ± 2.18	0.727 ± 0.127	0.836 ± 0.085
TV	-	33.36 ± 2.74	37.63 ± 2.70	0.830 ± 0.121	0.903 ± 0.082
CINN	6.43M	35.54 ± 3.51	39.81 ± 3.48	0.854 ± 0.122	0.919 ± 0.081
U-Net++	9.17M	35.37 ± 3.36	39.64 ± 3.40	0.861 ± 0.119	0.923 ± 0.080
MS-D-CNN	181.31K	35.85 ± 3.60	40.12 ± 3.56	0.858 ± 0.122	0.921 ± 0.082
U-Net	31.04M	35.87 ± 3.59	40.14 ± 3.57	0.859 ± 0.121	0.922 ± 0.081
LoDoPaB U-Net	613.32K	36.00 ± 3.63	40.28 ± 3.59	0.862 ± 0.119	0.923 ± 0.079
MUSC (ours)	13.87M	**36.08** ± **3.68**	**40.35** ± **3.64**	**0.863** ± **0.119**	**0.924** ± 0.080

Boldface indicates the best performance; second-best results are highlighted in grey.

TABLE IV
PERFORMANCE ON FASTMRI SINGLE-COIL KNEE VALIDATION DATA. RESULTS ARE COLLECTED FROM [65] EXCEPT U-NET AND MUSC. THE FASTMRI U-NET-32 MODEL REFERS TO A U-NET VARIANT DEFINED IN [65] WHOSE OUTPUT AFTER THE FIRST CONVOLUTION HAS 32 CHANNELS. OTHER MODELS ARE DEFINED SIMILARLY. PDFS AND PD CORRESPOND TO TWO MRI ACQUISITION PROTOCOLS WITH FAT SUPPRESSION (PDFS) AND WITHOUT FAT SUPPRESSION (PD) [65]

	Number of parameters	NMSE		PSNR		SSIM	
		PD	PDFS	PD	PDFS	PD	PDFS
TV	-	0.0287	0.0900	31.4	27.7	0.645	0.494
U-Net	31.04M	0.0161	0.0538	33.8	29.9	0.809	0.631
fastMRI U-Net-256	214.16M	**0.0154**	**0.0525**	**34.0**	**30.0**	**0.815**	**0.636**
MUSC (ours)	13.87M	0.0156	0.0526	**34.0**	**30.0**	0.811	0.631

Boldface indicates the best performance; second-best results are highlighted in grey.

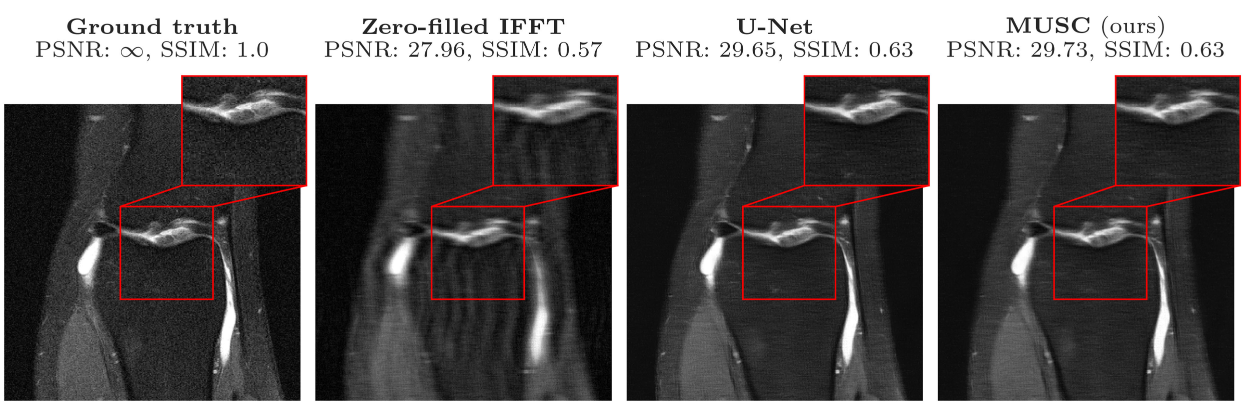


Fig. 4. Reconstructions of a test sample from the fastMRI single-coil knee dataset.

images in the test set are not provided to the public and the evaluation must be made through the fastMRI online submission system[3].

Following the training protocol of [65], we first transformed the undersampled k-space measurements into the image space using zero-filled Inverse Fast Fourier Transform (IFFT); we use the transformed images as input to MUSC and other CNN baselines. Consistent with previous work [65], we found that U-Net variants deliver exceptional performance on validation samples (Table IV). Remarkably, MUSC performs on-par with U-Net variants, yielding visually indistinguishable results (Fig. 4). We next evaluate the U-Net and the MUSC on test samples through the fastMRI submission system. On the test data, the proposed MUSC produces results comparable to the best-performing U-Net result (fastMRI U-Net-256) provided by the fastMRI challenge organizer while having an order of magnitude fewer parameters (Table V).

[3]https://fastmri.org/

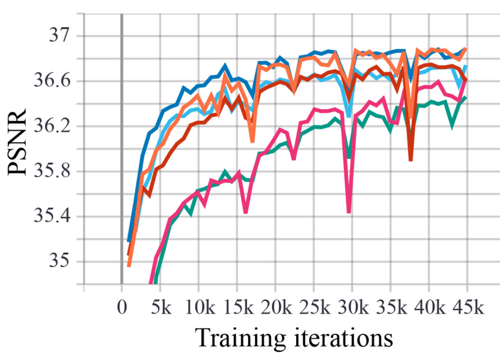
D. Single-Image Super-Resolution

We have additionally tested the MUSC on a standard super-resolution task, whose results are deferred to Appendix G. The goal of this task is to recover a high-resolution image from its degraded, low-resolution version. Unlike tasks such as CT and MRI reconstruction, in which the image degradation processes introduce long-range spatially correlated noise like streak artifacts, the blurring process in the super-resolution task is spatially local.

TABLE V
PERFORMANCE ON SINGLE-COIL KNEE TEST DATA. RESULTS ARE COLLECTED FROM THE FASTMRI PUBLIC LEADERBOARD

	Number of parameters	NMSE		PSNR		SSIM	
		PD	PDFS	PD	PDFS	PD	PDFS
Zero-filled IFFT	-	0.0266	0.0597	32.0	29.2	0.7765	0.6045
TV	-	0.0221	0.0716	33.1	28.5	0.7036	0.5096
U-Net	31.04M	0.0130	0.0430	35.3	30.6	0.8513	0.6626
fastMRI U-Net-256	214.16M	**0.0115**	**0.0414**	**35.9**	**30.8**	**0.8618**	**0.6680**
MUSC (ours)	13.87M	0.0126	0.0421	35.5	30.7	0.8537	0.6633

Boldface indicates the best performance; second-best results are highlighted in grey.



Legend	Tied adjoints? (Is $D_{\text{enc}} = \widetilde{D}_{\text{enc}}$?)	Tied en/decoders? (Is $D_{\text{enc}} = D_{\text{dec}}$?)	Fixed threshold? (Is λ fixed?)	Non-neg code? (Is $\alpha \geq 0$?)
Case 1	✗	✗	✗	✗
Case 2	✗	✗	✗	✓
Case 3	✓	✗	✗	✓
Case 4	✗	✓	✗	✓
Case 5	✓	✓	✗	✓
Case 6	✓	✓	✓	✓

Different choices of model components

Fig. 5. Ablation study on how different choices of model components affect the overall performance. Left panel shows the PSNR (evaluated on validation samples) of six trained models as the training progresses; right panel shows the configuration of each trained model, where Case 2 corresponds to the usual setting used in other sections of this paper. For training, we used first 10% of training samples of the LoDoPab-CT dataset; the validation samples are 50 samples in the validation fold of the dataset.

In this case, we do not observe a performance gain of using a multiscale model – either U-Net or MUSC – over state-of-the-art single-scale CSC models. Interestingly, MUSC outperforms the U-Net, but is up to 0.5 dB worse than single-scale CSC.

In subsection IV-F, we study this phenomenon by analyzing the sparse code yielded by MUSC. In the super-resolution tasks, the nonzeros in sparse codes are confined to high-resolution channels, or, equivalently, small filter supports which only leverage local information. This is well aligned with the intuition that the blurring forward operator mixes information only locally. It suggests that the right strategy is to use a large number of small-support filters just like CSC does, instead of "wasting" trainable parameters on unused large scales. We similarly find that a single-scale CSC model works better than MUSC on a denoising (Gaussian noise removal) task. Together, these findings suggest that multiscale features are no panacea for imaging inverse problems; the configuration of scales needs to resonate with the task-dependent forward operator that we aim to invert.

E. Ablations on the Choices of Model Components

In Fig. 5, we show ablation experiments that demonstrate how different choices of model components affect the overall performance. There, Case 2 is the off-the-shelf setup we have used in all other sections of this paper; this option has the fastest learning speed and highest end-point accuracy. Consistent with findings in [6], [25], [26], we find it advantageous to use untied adjoints as described in (11): Untied dictionaries (Cases 1, 2, and 4 in Fig. 5) in general perform much better than tied dictionaries with $\widetilde{D}_{\text{enc}} = D_{\text{enc}} = D_{\text{dec}}$ (Cases 5 and Case 6). What is more, we find that learnable threshold λ gives better results than fixed threshold. The non-negative constraint of sparse code $\alpha \geq 0$ does not greatly influence the end-point performance of models, although with the constraint the model learns slightly faster (Case 2) than without (Case 1).

F. Probing Multiscale Dictionary-Based Representations

Thus far, we have shown that our proposed multiscale CSC approach, dubbed MUSC, performs comparably to state-of-the-art CNNs in a range of imaging inverse problems. This is noteworthy, as the strong performance is achieved simply by employing a multiscale dictionary – as opposed to a single-scale one – in an otherwise standard CSC paradigm. The strong performance suggests the usefulness of the multiscale representation. We now analyze our learned dictionaries and their induced sparse representations.

a) Visualizing dictionary atoms: We visualize dictionary atoms of the MUSC. To extract a dictionary atom from a dictionary D, we first prepare an *indicator code* δ, which is a collection of multichannel tensors that takes a value 1 at a certain entry and 0 elsewhere; a dictionary atom corresponds

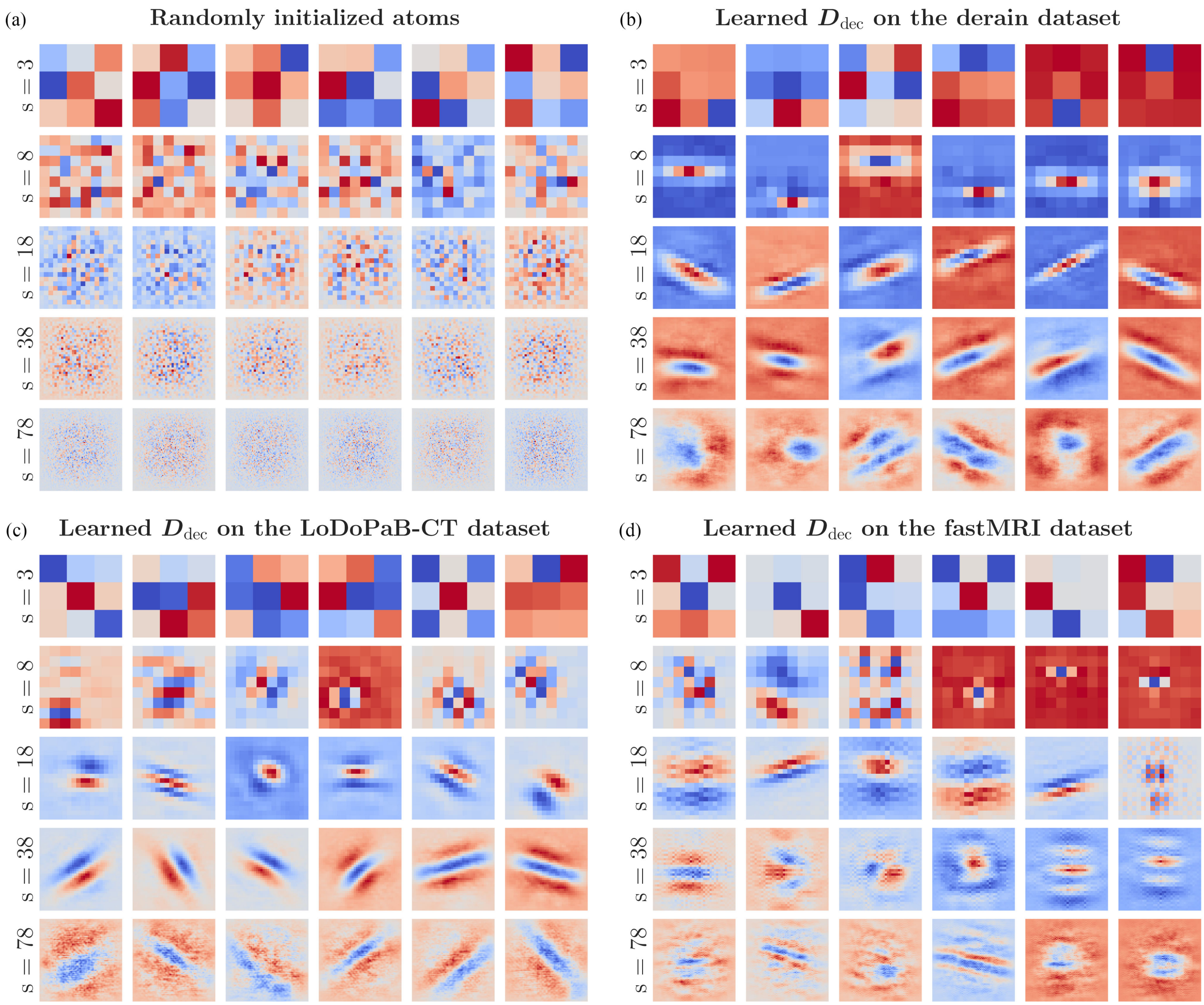


Fig. 6. Atoms in a randomly initialized (panel a) and learned decoder dictionary based on the derain dataset (panel b), LoDoPab-CT (panel c) and fastMRI (panel d) dataset. For all panels, each row corresponds to a support size (denoted by s) of atoms. Top rows are atoms that have a small support size; bottom rows are atoms that have a large support size. In each row, atoms are displayed in a sorted order of decreasing ℓ_2 norm; for the visualization purpose, they are normalized into the range $[-1, 1]$.

to that entry is computed as $\boldsymbol{D\delta}$. Note that, different positions of the nonzero entry may give rise to atoms of different support sizes. This can be seen in Fig. 1**b**: The indicator code is illustrated as the grey boxes; depending on the grey box the nonzero entry resides in, the sparse code activates different receptive fields under composite convolutions and transposed convolutions. If the nonzero entry resides in the top-most box, then the support of the atom is 3 as it undergoes only a single 3×3 convolution; if the nonzero entry is in one of the lower boxes, the support of the atom is larger as the code undergoes multiple convolutions and one or more transposed convolutions.

In Fig. 6, we show samples of multiscale atoms in $\boldsymbol{D}_{\text{dec}}$ of varying sizes – we crop these atoms to only show their nonzero support regions. As can be seen in Fig. 6**b-d**, the learned dictionaries contain Gabor-like or curvelet-like atoms with different spatial widths, resolutions, and orientations. Thus the learned dictionaries indeed exploit multiscale features. For comparison, we also show a randomly initialized dictionary (Fig. 6**a**). Unlike a learned dictionary, a random dictionary does not exhibit structures in atoms. We also visualize atoms of encoder dictionaries $\boldsymbol{D}_{\text{enc}}$ and $\widetilde{\boldsymbol{D}}_{\text{enc}}$ in Appendix E Using a similar technique, we also probe the multiscale representations learned by U-Nets in Appendix F.

b) Sparsity levels of representations: We anticipate that the trained dictionaries induce different sparsity levels at different resolution levels in a task-dependent manner: More non-zeros associated with large-support atoms are useful when imaging artifacts have long-range correlations (e.g., streak artifacts in CT) than when the artifacts are localized (e.g., deraining or super-resolution).

Fig. 7 shows the sparsity levels across tasks, both before and after dictionary learning. We observe that, prior to any learning, the sparsity levels induced by randomly initialized

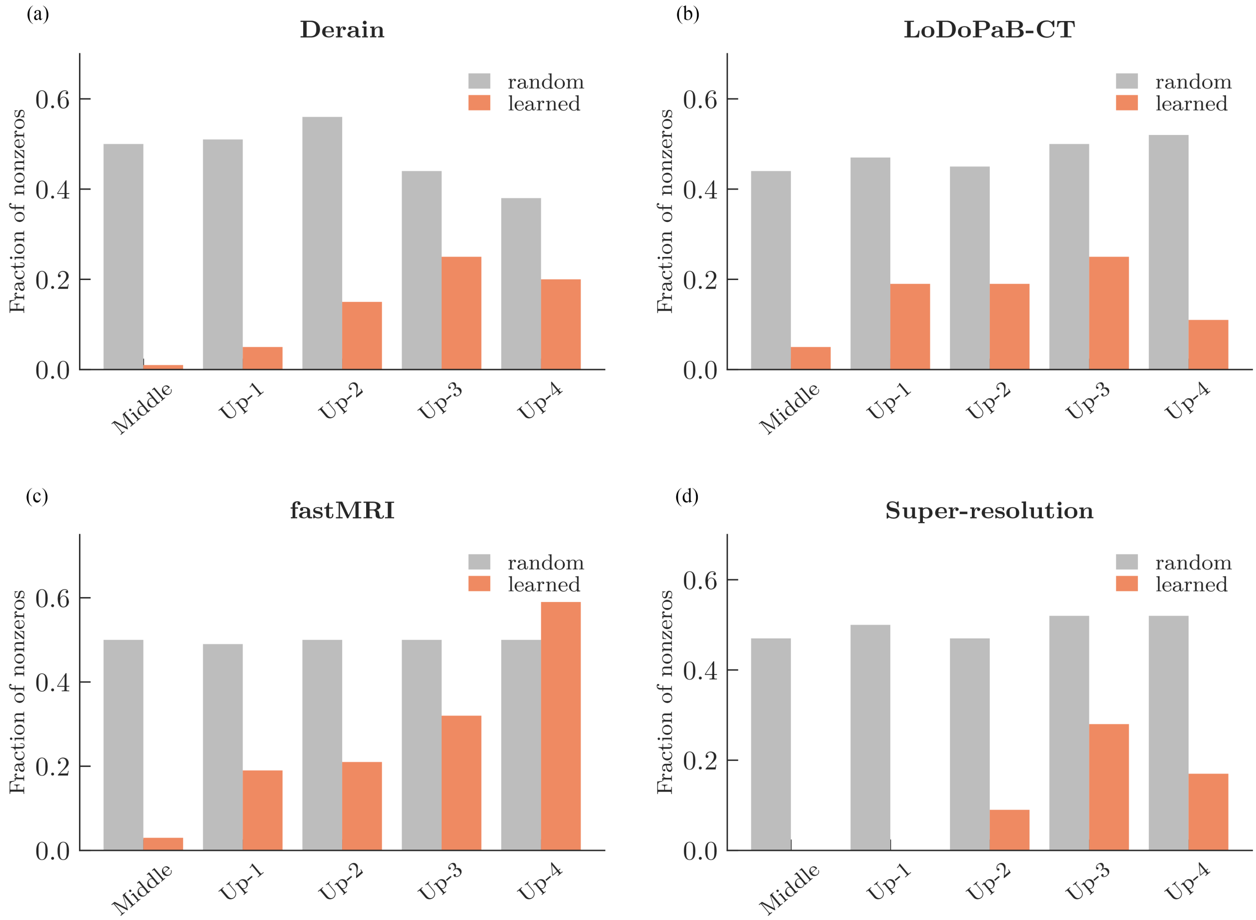


Fig. 7. Sparsity of dictionary-induced convolutional features maps. Each bar corresponds to the sparsity level of a feature map tensor from the "deepest" activations corresponding to large-support atoms ("Middle") to the "shallowest' activations corresponding to small-support atoms ("Up-4").

dictionaries (grey bars) are approximately uniform across scales. After learning, the sparsity levels of feature maps differentiate in a task-dependent way (orange bars in all panels). This task-dependent differentiation suggests the usefulness of multiscale representations – the learned sparsity levels are neither collapsed to a single scale nor remain uniform across spatial scales; instead, they are weighted and combined across scales in a problem-dependent way. A curious effect of multiscale learning arises in super-resolution (panel **d**): the activations are nonzero *only in high-resolution features* ("Up-2," "Up-3," and "Up-4"), corroborating the intuition that low-resolution features are not important for this task. Additionally, comparing the "Middle" bars across panels, we see that CT and MRI reconstruction tasks indeed use more nonzero coefficients on large-support atoms than tasks such as deraining and super-resolution.

V. Discussion

The CSC paradigm provides a natural connection between sparse modeling and CNNs. Despite being mathematical principled, existing CSC models still fall short competing with CNNs in terms of empirical performance on challenging inverse problems. In this work, we report one simple and effective way to close the performance gap between CSC and CNN models: incorporating a multiscale structure in the CSC dictionaries. Crucial to our approach is the structure of our constructed multiscale dictionary: It takes inspiration from and closely follows the highly successful U-Net model. We show that the constructed multiscale dictionary performs on par with leading CNNs in major imaging inverse problems. These results suggest a strong link between dictionary learning and CNNs – in both cases, multiscale structures are essential ingredients.

Beyond empirical performance, we believe that the interpretability of the proposed MUSC is showing the way towards an interpretable deep learning model. An interpretable model consists of components whose objectives and functionality have nominal values. The MUSC fulfills this desideratum by incorporating two modules with well-understood objectives, one implementing sparse coding and the other dictionary-based synthesis.

Overall, our work demonstrates the effectiveness and scalability of CSC models on imaging inverse problems. While deep neural networks are profoundly influencing image reconstruction, our work shows promise in a different direction: the principles of sparsity and multiscale representation developed decades ago are still useful in designing performant, parameter-efficient (compared to mainstream CNNs), and interpretable architectures that push the current limits of machine learning for imaging inverse problems.

Acknowledgment

The authors would like to thank Johannes Leuschner for helpful discussions regarding the LoDoPaB-CT dataset. Numerical experiments were partly performed at the sciCORE (http://scicore.unibas.ch/) scientific computing center at the University of Basel. They thank the outstanding anonymous reviewers whose comments greatly impoved this manuscript.

References

[1] J. Mairal, P. Koniusz, Z. Harchaoui, and C. Schmid, "Convolutional kernel networks," in *Proc. Neural Inf. Process. Syst.*, 2020, pp. 2627–2635.
[2] J. Mairal, "End-to-end kernel learning with supervised convolutional kernel networks," in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 1399–1407.
[3] A. Bietti and J. Mairal, "Group invariance, stability to deformations, and complexity of deep convolutional representations," *J. Mach. Learn. Res.*, vol. 20, no. 25, pp. 1–49, 2019.
[4] S. Mallat, "Group invariant scattering," *Commun. Pure Appl. Math.*, vol. 65, no. 10, pp. 1331–1398, 2012.
[5] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1872–1886, Aug. 2013.
[6] J. Zarka, L. Thiry, T. Angles, and S. Mallat, "Deep network classification by scattering and homotopy dictionary learning," in *Int. Conf. Learn. Representations*, 2020, pp. 15664–15675.
[7] J. Zarka, F. Guth, and S. Mallat, "Separation and concentration in deep networks," in *Int. Conf. Learn. Representations*, 2021.
[8] V. Papyan, Y. Romano, and M. Elad, "Convolutional neural networks analyzed via convolutional sparse coding," *J. Mach. Learn. Res.*, vol. 18, no. 83, pp. 1–52, 2017.
[9] V. Papyan, Y. Romano, J. Sulam, and M. Elad, "Theoretical foundations of deep learning via sparse representations: A multilayer sparse model and its connection to convolutional neural networks," *IEEE Signal Process. Mag.*, vol. 35, no. 4, pp. 72–89, Jul. 2018.
[10] D. Simon and M. Elad, "Rethinking the CSC model for natural images," in *Proc. Neural Inf. Process. Syst.*, 2019, pp. 2271–2281.
[11] K. Degraux, U. S. Kamilov, P. T. Boufounos, and D. Liu, "Online convolutional dictionary learning for multimodal imaging," in *Proc. IEEE Int. Conf. Image Process.*, 2017, pp. 1617–1621.
[12] I. Y. Chun and J. A. Fessler, "Convolutional dictionary learning: Acceleration and convergence," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 1697–1712, Apr. 2018.
[13] E. Zisselman, J. Sulam, and M. Elad, "A local block coordinate descent algorithm for the CSC model," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8200–8209.
[14] I. Rey-Otero, J. Sulam, and M. Elad, "Variations on the convolutional sparse coding model," *IEEE Trans. Signal Process.*, vol. 68, pp. 519–528, 2020.
[15] N. Chodosh and S. Lucey, "When to use convolutional neural networks for inverse problems," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8223–8232.
[16] B. Tolooshams, A. H. Song, S. Temereanca, and D. E. Ba, "Convolutional dictionary learning based auto-encoders for natural exponential-family distributions," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 9493–9503.
[17] P. V. Arun, K. M. Buddhiraju, A. Porwal, and J. Chanussot, "CNN-based super-resolution of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 9, pp. 6106–6121, Sep. 2020.
[18] J.-J. Huang and P. L. Dragotti, "Learning deep analysis dictionaries for image super-resolution," *IEEE Trans. Signal Process.*, vol. 68, pp. 6633–6648, Nov. 2020.
[19] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, Jul. 1989.
[20] E. J. Candès and D. L. Donoho, "New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities," *Commun. Pure Appl. Math.*, vol. 57, no. 2, pp. 219–266, 2004.
[21] J.-L. Starck, E. Candes, and D. Donoho, "The curvelet transform for image denoising," *IEEE Trans. Image Process.*, vol. 11, no. 6, pp. 670–684, Jun. 2002.
[22] J. Mairal, G. Sapiro, and M. Elad, "Learning multiscale sparse representations for image and video restoration," *Multiscale Model. Simul.*, vol. 7, no. 1, pp. 214–241, 2008.
[23] B. Ophir, M. Lustig, and M. Elad, "Multi-scale dictionary learning using wavelets," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 5, pp. 1014–1024, Sep. 2011.
[24] M. Maggioni, S. Minsker, and N. Strawn, "Multiscale dictionary learning: Non-asymptotic bounds and robustness," *J. Mach. Learn. Res.*, vol. 17, no. 2, pp. 1–51, 2016.
[25] B. Lecouat, J. Ponce, and J. Mairal, "Fully trainable and interpretable non-local sparse models for image restoration," in *Eur. Conf. Comput. Vis.*, 2020, pp. 238–254.
[26] B. Lecouat, J. Ponce, and J. Mairal, "A flexible framework for designing trainable priors with adaptive smoothing and game encoding," in *Proc. Neural Inf. Process. Syst.*, 2020.
[27] R. Khatib, D. Simon, and M. Elad, "Learned greedy method (LGM): A novel neural architecture for sparse coding and beyond," *J. Vis. Commun. Image Representation*, vol. 77, 2021, Art. no. 103095.
[28] M. Scetbon, M. Elad, and P. Milanfar, "Deep K-SVD denoising," *IEEE Trans. Image Process.*, vol. 30, pp. 5944–5955, 2021.
[29] M. Li *et al.*, "Video rain streak removal by multiscale convolutional sparse coding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6644–6653.
[30] H. Chang, J. Han, C. Zhong, A. M. Snijders, and J.-H. Mao, "Unsupervised transfer learning via multi-scale convolutional sparse coding for biomedical applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1182–1194, May 2018.
[31] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image Comput. Comput.-Assist. Interv.*, 2015, pp. 234–241.
[32] T. Falk *et al.*, "U-Net: Deep learning for cell counting, detection, and morphometry," *Nature Methods*, vol. 16, no. 1, pp. 67–70, Jan. 2019.
[33] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet : Redesigning skip connections to exploit multiscale features in image segmentation," *IEEE Trans. Med. Imag.*, vol. 39, no. 6, pp. 1856–1867, Jun. 2020.
[34] D. Ren, W. Zuo, Q. Hu, P. Zhu, and D. Meng, "Progressive image deraining networks: A better and simpler baseline," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3932–3941.
[35] G. Beylkin, R. Coifman, and V. Rokhlin, "Fast wavelet transforms and numerical algorithms I," *Commun. Pure Appl. Math.*, vol. 44, pp. 141–183, 1991.
[36] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Rev.*, vol. 51, no. 1, pp. 34–81, 2009.
[37] M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly, "Compressed sensing MRI," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 72–82, Mar. 2008.
[38] J. Charlety *et al.*, "Global seismic tomography with sparsity constraints: Comparison with smoothing and damping regularization," *J. Geophysical Res., Solid Earth*, vol. 118, no. 9, pp. 4887–4899, 2013.
[39] H. Liebgott, R. Prost, and D. Friboulet, "Pre-beamformed RF signal reconstruction in medical ultrasound using compressive sensing," *Ultrasonics*, vol. 53, no. 2, pp. 525–533, 2013.
[40] C. Chen and D. Donoho, "Basis pursuit," in *Proc. 1994 28th Asilomar Conf. Signals, Syst. Comput.*, 1994, pp. 41–44.
[41] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *J. Roy. Stat. Soc., Ser. B. (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.
[42] I. Daubechies, M. Defrise, and C. D. Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, 2004.
[43] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 791–804, Apr. 2012.
[44] H. Bristow, A. Eriksson, and S. Lucey, "Fast convolutional sparse coding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 391–398.
[45] M. Šorel and F. Šroubek, "Fast convolutional sparse coding using matrix inversion lemma," *Digit. Signal Process.*, vol. 55, pp. 44–51, 2016.
[46] J. Liu, C. Garcia-Cardona, B. Wohlberg, and W. Yin, "Online convolutional dictionary learning," in *Proc. IEEE Int. Conf. Image Process.*, 2017, pp. 1707–1711.

[47] H. Sreter and R. Giryes, "Learned convolutional sparse coding," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 2191–2195.

[48] D. O. Baguer, J. Leuschner, and M. Schmidt, "Computed tomography reconstruction using deep image prior and learned reconstruction methods," *Inverse Problems*, vol. 36, no. 9, Sep. 2020, Art. no. 094004.

[49] J. Leuschner *et al.*, "Quantitative comparison of deep learning-based image reconstruction methods for low-dose and sparse-angle CT applications," *J. Imag.*, vol. 7, no. 3, 2021, Art. no. 44.

[50] T. Huang, W. Dong, X. Yuan, J. Wu, and G. Shi, "Deep Gaussian scale mixture prior for spectral compressive imaging," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 16211–16220.

[51] W. Dong, P. Wang, W. Yin, G. Shi, F. Wu, and X. Lu, "Denoising prior driven deep neural network for image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2305–2318, Oct. 2019.

[52] C. Etmann, R. Ke, and C.-B. Schönlieb, "iunets: Learnable invertible up- and downsampling for large-scale inverse problems," in *Proc. IEEE 30th Int. Workshop Mach. Learn. Signal Process.*, 2020, pp. 1–6.

[53] J. C. Ye, Y. Han, and E. Cha, "Deep convolutional framelets: A general deep learning framework for inverse problems," *SIAM J. Imag. Sci.*, vol. 11, no. 2, pp. 991–1048, 2018.

[54] Y. Han and J. C. Ye, "Framing U-Net via deep convolutional framelets: Application to sparse-view CT," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1418–1429, Jun. 2018.

[55] J. C. Ye and W. K. Sung, "Understanding geometry of encoder-decoder CNNs," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 7064–7073.

[56] R. Yin, T. Gao, Y. M. Lu, and I. Daubechies, "A tale of two bases: Local-nonlocal regularization on image patches with convolution framelets," *SIAM J. Imag. Sci.*, vol. 10, no. 2, pp. 711–750, 2017.

[57] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 399–406.

[58] T. Moreau and J. Bruna, "Understanding trainable sparse coding with matrix factorization," in *Proc. Int. Conf. Learn. Representations*, 2017.

[59] J. Liu, X. Chen, Z. Wang, and W. Yin, "ALISTA: Analytic weights are as good as learned weights in LISTA," in *Int. Conf. Learn. Representations*, 2019.

[60] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, "Rain streak removal using layer priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2736–2744.

[61] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, "Removing rain from single images via a deep detail network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1715–1723.

[62] Y. Luo, Y. Xu, and H. Ji, "Removing rain from a single image via discriminative sparse coding," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3397–3405.

[63] W. Yang, R. T. Tan, J. Feng, Z. Guo, S. Yan, and J. Liu, "Joint rain detection and removal from a single image with contextualized deep networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 6, pp. 1377–1393, Jun. 2020.

[64] J. Leuschner, M. Schmidt, D. O. Baguer, and P. Maass, "LoDoPaB-CT, a benchmark dataset for low-dose computed tomography reconstruction," *Sci. Data*, vol. 8, no. 1, Apr. 2021, Art. no. 109.

[65] J. Zbontar *et al.*, "fastMRI: An open dataset and benchmarks for accelerated MRI," 2018, *arXiv:1811.08839*.

[66] S. G. Armato *et al.*, "The Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI): A completed reference database of lung nodules on CT scans," *Med. Phys.*, vol. 38, no. 2, pp. 915–931, Feb. 2011.

[67] A. Denker, M. Schmidt, J. Leuschner, P. Maaß, and J. Behrmann, "Conditional normalizing flows for low-dose computed tomography image reconstruction," *ICML Workshop on Invertible Neural Networks, Normalizing Flows, Explicit Likelihood Models* 2020.

[68] Z. Zhou *et al.*, "UNet: A nested U-Net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, D. Stoyanov, Z. Taylor, G. Carneiro, T. Syeda-Mahmood, A. Martel, L. Maier-Hein, J. M. R. Tavares, A. Bradley, J. P. Papa, V. Belagiannis, J. C. Nascimento, Z. Lu, S. Conjeti, M. Moradi, H. Greenspan, and A. Madabhushi, Eds., Berlin, Germany: Springer, 2018, pp. 3–11.

[69] D. M. Pelt, K. J. Batenburg, and J. A. Sethian, "Improving tomographic reconstruction from limited data using mixed-scale dense convolutional neural networks," *J. Imag.*, vol. 4, no. 11, 2018, Art. no. 128.