D^{3}C^2-NET: DUAL-DOMAIN DEEP CONVOLUTIONAL CODING NETWORK FOR COMPRESSIVE SENSING

Anonymous authors
Paper under double-blind review

ABSTRACT

Mapping optimization algorithms into neural networks, deep unfolding networks (DUNs) have achieved impressive success in compressive sensing (CS). From the perspective of optimization, DUNs inherit a well-defined and interpretable structure from iterative steps. However, from the viewpoint of neural network design, most existing DUNs are inherently established based on traditional image-domain unfolding, which takes single-channel images as inputs and outputs between adjacent stages, resulting in insufficient information transmission capability and the inevitable loss of the image details. In this paper, to break the above bottleneck, we propose a generalized dual-domain optimization framework, which is general for inverse imaging problems and integrates the merits of both (1) image-domain and (2) convolutional-coding-domain priors to constrain the feasible region of the solution space. By unfolding the proposed optimization framework into deep neural networks, we further design a novel **D**ual-**D**omain **D**eep **C**onvolutional **C**oding **Net**work (D^{3}C^2-Net)[1] for CS imaging with the ability of transmitting high-capacity feature through all the unfolded stages. Experiments on multiple natural and MR image datasets demonstrate that our D^{3}C^2-Net achieves higher performance and better accuracy-complexity trade-offs than other state-of-the-art.

1 INTRODUCTION

As a new paradigm of signal acquisition, compressive sensing (CS) aims to recover the original signal from a small number of measurements obtained by linear random projection (Candès et al., 2006; Baraniuk, 2007), which has been successfully used in many applications, like single-pixel imaging (Duarte et al., 2008; Rousset et al., 2016), accelerating magnetic resonance imaging (MRI) (Lustig et al., 2007) and snapshot compressive imaging (SCI) (Wu et al., 2021a;b; Zhang et al., 2022).

Mathematically, given the original vectorized image $\mathbf{x} \in \mathbb{R}^N$ and a sampling matrix $\mathbf{\Phi} \in \mathbb{R}^{M\times N}$, the CS measurement of $\mathbf{x}$, denoted by $\mathbf{y} \in \mathbb{R}^M$ is formulated as $\mathbf{y} = \mathbf{\Phi}\mathbf{x} + \mathbf{n}$, where $\mathbf{n}$ is the additive white Gaussian noise (AWGN) with standard deviation σ. The purpose of CS reconstruction is to infer $\mathbf{x}$ from the obtained $\mathbf{y}$. CS is a typical ill-posed inverse problem due to the common setup of $M \ll N$, and the CS ratio (or sampling rate) is defined as $\gamma = M/N$. Generally, the conventional model-based methods reconstruct the latent clean $\mathbf{x}$ by solving the following optimization problem:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{\Phi}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\phi(\mathbf{x}), \tag{1}$$

where $\phi(\cdot)$ denotes a prior term with a regularization parameter λ. For traditional CS methods (Zhang et al., 2014b;a; Kim et al., 2010; Zhao et al., 2018; Metzler et al., 2016), $\phi(\cdot)$ is usually handcrafted with some pre-defined basis, like wavelet and discrete cosine transform (DCT) (Zhao et al., 2014; 2016a). Although these model-based methods enjoy the advantages of interpretability and convergence guarantees, they inevitably suffer from high computational complexity and the difficulty of choosing optimal transforms and hyper-parameters (Zhao et al., 2016c;b).

With the recent vigorous development of deep learning, many network-based image CS methods have been proposed, generally divided into deep non-unfolding networks (DNUNs) and deep unfolding networks (DUNs). By treating the CS recovery as a de-aliasing problem, DNUNs directly learn the inverse mapping from measurement $\mathbf{y}$ to image $\mathbf{x}$ through end-to-end "black-box" networks (Mousavi et al., 2015; Iliadis et al., 2018; Hyun et al., 2018; Kulkarni et al., 2016; Sun et al., 2020; Shi et al.,

[1] For reproducible research, the source code with pre-trained models of our D^{3}C^2-Net will be made available.

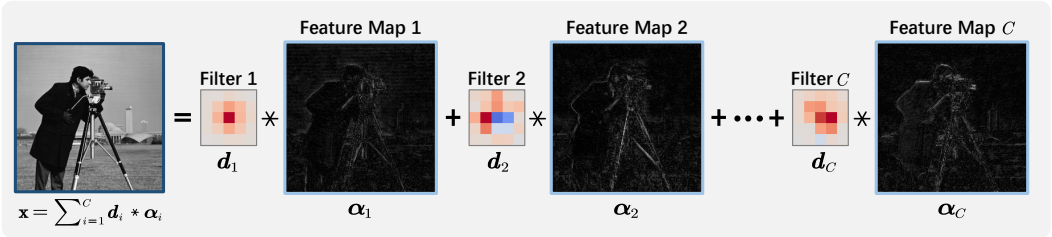


Figure 1: Illustration of the idea of convolutional coding. An image $\mathbf{x}$ is represented by the combination (sum) of multiple image-level convolution results, *i.e.*, $\mathbf{x} = \sum_{i=1}^{C} \boldsymbol{d}_i * \boldsymbol{\alpha}_i$, where $\boldsymbol{d}_i \in \mathbb{R}^{k \times k}$ is the i^{th} dictionary filter, $\boldsymbol{\alpha}_i \in \mathbb{R}^{h \times w}$ is the i^{th} feature map, $*$ is the convolution operator and C is the number of feature channels. The darker red (or blue) colors in each visualized filter correspond to the positive (or negative) entries with larger absolute values. Compared with the trivial single-channel image data, this type of feature-level representation naturally enjoys higher capacity and flexibility.

2019a;b), which highly depend on the careful tunings and lead to the extreme difficulty of analysis. DUNs combine the merits of networks and optimization frameworks by training a truncated unfolding inference (Zhang & Ghanem, 2018; Zhang et al., 2020a; You et al., 2021b; Song et al., 2021; You et al., 2021a). They often consist of a fixed number of iterative stages in series. Due to the well-defined structure and superior performance, DUNs have become the mainstream in the CS field.

However, most existing DUNs are designed based on trivial image-domain unfolding, where the stage input and output are single-channel images with poor representation capacity. There are often some operations of channel number reduction from multiple to one at the end of each unfolded stage, leading to inevitable limited feature transmission and the information loss of image details (Zhang & Ghanem, 2018; Zhang et al., 2020a; You et al., 2021a;b).

Recently, some convolutional coding methods have been successfully adapted to DUNs (Fu et al., 2019; Wang et al., 2020; Zheng et al., 2021). As shown in Fig. 1, the main idea of convolutional coding model is to represent an image $\mathbf{x} \in \mathbb{R}^{h \times w}$ as $\mathbf{x} = \mathbf{D} \circledast \boldsymbol{\alpha} = \sum_{i=1}^{C} \boldsymbol{d}_i * \boldsymbol{\alpha}_i$, where $*$ is the 2D convolution operator and C is the feature channel number; $\mathbf{D} \in \mathbb{R}^{C \times k \times k}$ is the convolutional dictionary and $\boldsymbol{d}_i$ is the i^{th} dictionary filter; $\boldsymbol{\alpha} \in \mathbb{R}^{C \times h \times w}$ is the feature map of $\mathbf{x}$ and $\boldsymbol{\alpha}_i$ is the i^{th} channel of $\boldsymbol{\alpha}$. Taking the natural advantage of $\boldsymbol{\alpha}$ being C-channel, these DUNs transmit high-capacity features among all stages. But they focus on specific tasks like rain removal (Wang et al., 2020) and image denoising (Zheng et al., 2021) without considering more general cases.

To address the above issues, in this paper, we propose a **D**ual-**D**omain **D**eep **C**onvolutional **C**oding **Net**work, dubbed D^{3}C^2-Net, focusing on the general CS problems. Specifically, we design a novel dual-domain unfolding framework, which resolves the lack of generalizability of existing methods, allows our D^{3}C^2-Net to transmit high-throughput information, and inherits the advantages of image and convolutional-coding domain constraints. The proposed D^{3}C^2-Net can be regarded as an attempt to bridge the gap between the convolutional coding methods and neural networks for the CS reconstruction problem, with the merits of interpretability and sufficient information throughput.

Our main contributions are three-fold: **(1)** We propose a novel general dual-domain optimization framework, which integrates the merits of both image-domain and convolutional-coding-domain priors to constrain the feasible solution space and can be easily applied to other inverse imaging problems. **(2)** We design a new **D**ual-**D**omain **D**eep **C**onvolutional **C**oding **Net**work (D^{3}C^2-Net) for general CS problems based on the proposed framework. Our D^{3}C^2-Net transmits high-capacity feature-level image representation through all the unfolded stages to capture sufficient features adaptively, thus recovering more details and textures. **(3)** Experiments on natural and MR image CS tasks show that our D^{3}C^2-Net outperforms existing state-of-the-art networks by large margins.

2 Related Work

Deep unfolding network. DUNs have been proposed to solve various inverse imaging problems (Chen & Pock, 2016; Lefkimmiatis, 2017; Metzler et al., 2017; Zheng et al., 2021; Kruse et al., 2017; Kokkinos & Lefkimmiatis, 2018). For CS and compressive sensing MRI (CS-MRI) tasks, DUNs usually combine convolutional neural network (CNN) denoisers with some optimization algorithms, like alternating minimization (AM) (Schlemper et al., 2017; Sun et al., 2018; Zheng et al., 2019), half quadratic splitting (HQS) (Zhang et al., 2017; Dong et al., 2018; Aggarwal et al., 2018),

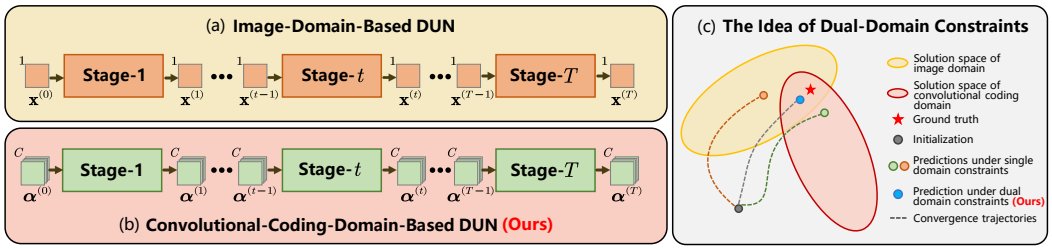


Figure 2: Illustration of the main ideas of dual-domain DUN design. (a) shows the architecture of the image-domain-based DUN and (b) shows the architecture of the convolutional-coding-domain-based DUN. Compared with (a) recovering the target image step-by-step ($\{\mathbf{x}^{(t)}\}$), our (b) transmits high-dimensional features ($\{\boldsymbol{\alpha}^{(t)}\}$) through all the unfolded stages. (c) further compares the converge trajectories under single and our dual-domain constraints, where the latter achieves more accurate recoveries than the former by employing the combination of learned dual-domain knowledge.

iterative shrinkage-thresholding algorithm (ISTA) (Gilton et al., 2019; Zhang & Ghanem, 2018; Zhang et al., 2020a; Song et al., 2021), alternating direction method of multipliers (ADMM) (Yang et al., 2018) and inertial proximal algorithm for nonconvex optimization (iPiano) (Su & Lian, 2020). Although existing DUNs benefit from well-defined frameworks, their inherent design of image-domain-based unfolding limits the feature transmission capability. Some DUNs take the previous intermediate features as auxiliary information in all stages but keep the idea of image-domain-based unfolding, which limits their further improvement (Chen et al., 2020; Song et al., 2021).

Deep convolutional coding. Convolutional coding has been widely studied in image restoration (Gu et al., 2015; Deng & Dragotti, 2020; Li et al., 2018). Compared with other sparse coding methods, the convolutional coding model is shift-invariant and flexible. Nevertheless, most existing convolutional coding methods use hand-crafted sparsity priors (Fu et al., 2019; Xu et al., 2020; Sreter & Giryes, 2018; Gao et al., 2022), *e.g.*, ℓ_0- or ℓ_1-regularization, instead of learning abundant priors from data. Recently, the convolutional coding model has been integrated into deep unfolding methods. Wang *et al.* (2020) design an interpretable deep network for rain removal. Zheng *et al.* (2021) propose a deep convolutional dictionary learning framework for denoising. However, they only develop for the special cases where the measurement (or degradation) matrix $\boldsymbol{\Phi}$ in Eq. (1) is the identity, *i.e.*, $\boldsymbol{\Phi} = \mathbf{I}$.

3 Proposed D^{3}C^2-Net for Compressive Sensing

3.1 Convolutional-Coding-Inspired Dual-Domain Formulation

Being different from the existing works based on image-domain DUNs, we draw inspiration from convolutional coding methods to enhance the information transmission capability. Figs. 2 (a) and (b) show the architectures of the image-domain-based and convolutional-coding-domain-based DUNs, respectively. We can observe that in the inherent design of image-domain-based DUNs, the single-channel image $\mathbf{x}$ in Eq. (1) is taken as the input and output of each stage and greatly hampers the information transmission capability. By taking the natural advantage of feature maps $\boldsymbol{\alpha}$ being C-channel, convolutional-coding-based DUNs can transmit high-capacity informative features among stages. Notably, the prior term in Eq. (1) plays an essential role in the reconstructing process as it narrows the feasible region of the solution space. This idea leads to the model-level integration of image-domain and convolutional-coding-domain priors as follows:

$$\{\mathbf{D}, \mathbf{z}, \boldsymbol{\alpha}\} = \underset{\{\mathbf{D},\mathbf{z},\boldsymbol{\alpha}\}}{\arg\min} \frac{1}{2}\|\boldsymbol{\Phi}\mathbf{z} - \mathbf{y}\|_2^2 + \frac{\mu_{\mathbf{z}}}{2}\|\mathbf{z} - \mathbf{D} \circledast \boldsymbol{\alpha}\|_2^2 + \lambda\psi(\boldsymbol{\alpha}) + \tau\phi(\mathbf{z}), \tag{2}$$

where $\mathbf{z} \in \mathbb{R}^{h\times w}$ is the image, $\boldsymbol{\alpha} \in \mathbb{R}^{C\times h\times w}$ is the convolutional coefficients, $\phi(\mathbf{z})$ and $\psi(\boldsymbol{\alpha})$ are the prior terms of image domain and convolutional-coding domain respectively, and $\mu_{\mathbf{z}}$, λ and τ are the trade-off parameters. The advantages of dual-domain priors are illustrated in Fig. 2(c). One can observe that the introduction of dual-domain priors constrains the solution feasible region, leading to more accurate reconstruction results than single-domain-based models. Besides, compared with the objective functions in (Wang et al., 2020) and (Zheng et al., 2021) where the measurement matrix $\boldsymbol{\Phi}$ in $\mathbf{y} = \boldsymbol{\Phi}\mathbf{x} + \mathbf{n}$ is specially the identity matrix $\mathbf{I}$, our method is generalizable to other cases.

3.2 DUAL-DOMAIN OPTIMIZATION FRAMEWORK

To simplify the overall optimization process, we collaboratively learn a universal $\mathbf{D}$ and the other network components through end-to-end training and solve $\mathbf{z}$ and $\boldsymbol{\alpha}$ in Eq. (2) iteratively as follows:

$$\mathbf{z}^{(t)} = \arg\min_{\mathbf{z}} \frac{1}{2}\|\mathbf{\Phi z} - \mathbf{y}\|_2^2 + \frac{\mu_{\mathbf{z}}}{2}\|\mathbf{z} - \mathbf{D} \circledast \boldsymbol{\alpha}^{(t-1)}\|_2^2 + \tau\phi(\mathbf{z}), \tag{3a}$$

$$\boldsymbol{\alpha}^{(t)} = \arg\min_{\boldsymbol{\alpha}} \frac{\mu_{\mathbf{z}}}{2}\|\mathbf{D} \circledast \boldsymbol{\alpha} - \mathbf{z}^{(t)}\|_2^2 + \lambda\psi(\boldsymbol{\alpha}). \tag{3b}$$

Image-level optimization. The image-domain optimization and the convolutional-coding-domain optimization are decoupled into Eqs. (3a) and (3b), respectively. The $\mathbf{z}$-subproblem in Eq. (3a) can be solved through proximal gradient descent (PGD) by iterating between the following two steps:

$$\begin{aligned}\mathbf{r}^{(t)} &= \mathcal{G}_{\mathrm{GDM}}\left(\boldsymbol{\alpha}^{(t-1)}, \mathbf{z}^{(t-1)}, \mathbf{D}, \rho, \mu_{\mathbf{z}}\right) \\ &= \mathbf{z}^{(t-1)} - \rho\left(\mathbf{\Phi}^{\top}\left(\mathbf{\Phi z}^{(t-1)} - \mathbf{y}\right) + \mu_{\mathbf{z}}\left(\mathbf{z}^{(t-1)} - \mathbf{D} \circledast \boldsymbol{\alpha}^{(t-1)}\right)\right),\end{aligned} \tag{4a}$$

$$\mathbf{z}^{(t)} = \mathcal{G}_{\mathrm{PMN}}(\mathbf{r}^{(t)}) = \mathrm{prox}_{\tau\phi}(\mathbf{r}^{(t)}) = \arg\min_{\mathbf{z}^*} \frac{1}{2}\|\mathbf{z}^* - \mathbf{r}^{(t)}\|_2^2 + \tau\phi(\mathbf{z}^*), \tag{4b}$$

where $\mathcal{G}_{\mathrm{GDM}}$ and $\mathcal{G}_{\mathrm{PMN}}$ denote the gradient descent module (GDM) and proximal mapping network (PMN), respectively. Their structural details will be elaborated on in the next subsection.

Feature-level optimization. For the $\boldsymbol{\alpha}$-subproblem in Eq. (3b), where $\frac{\mu_{\mathbf{z}}}{2}\|\mathbf{D} \circledast \boldsymbol{\alpha} - \mathbf{z}^{(t)}\|_2^2$ is the data term, $\psi(\boldsymbol{\alpha})$ is the prior term, and λ is a trade-off parameter. To separate the data term and the prior term, we apply the HQS algorithm, which tackles Eq. (3b) by introducing an auxiliary variable $\tilde{\boldsymbol{\alpha}}$, leading to the following objective function:

$$\{\boldsymbol{\alpha}, \tilde{\boldsymbol{\alpha}}\} = \arg\min_{\boldsymbol{\alpha}, \tilde{\boldsymbol{\alpha}}} \frac{\mu_{\mathbf{z}}}{2}\|\mathbf{D} \circledast \tilde{\boldsymbol{\alpha}} - \mathbf{z}^{(t)}\|_2^2 + \lambda\psi(\boldsymbol{\alpha}) + \frac{\mu_{\boldsymbol{\alpha}}}{2}\|\boldsymbol{\alpha} - \tilde{\boldsymbol{\alpha}}\|_2^2, \tag{5}$$

where $\mu_{\boldsymbol{\alpha}}$ is the penalty parameter for the distance between $\boldsymbol{\alpha}$ and $\tilde{\boldsymbol{\alpha}}$. The above Eq. (5) can be also solved iteratively as follows:

$$\tilde{\boldsymbol{\alpha}}^{(t)} = \arg\min_{\boldsymbol{\alpha}^*} \frac{1}{2}\|\mathbf{D} \circledast \boldsymbol{\alpha}^* - \mathbf{z}^{(t)}\|_2^2 + \frac{\eta}{2}\|\boldsymbol{\alpha}^* - \boldsymbol{\alpha}^{(t-1)}\|_2^2, \tag{6a}$$

$$\boldsymbol{\alpha}^{(t)} = \arg\min_{\boldsymbol{\alpha}^*} \frac{1}{2}\|\boldsymbol{\alpha}^* - \tilde{\boldsymbol{\alpha}}^{(t)}\|_2^2 + \beta\psi(\boldsymbol{\alpha}^*), \tag{6b}$$

where $\eta = \mu_{\boldsymbol{\alpha}}/\mu_{\mathbf{z}}$ and $\beta = \lambda/\mu_{\boldsymbol{\alpha}}$. For solving the Eq. (6a), the Fast Fourier Transform (FFT) can be utilized by assuming the convolution is carried out with circular boundary conditions. Let $\boldsymbol{\mathcal{D}} = \mathcal{F}(\mathbf{D})$, $\boldsymbol{\mathcal{Z}}^{(t)} = \mathcal{F}(\mathbf{z}^{(t)})$, and $\boldsymbol{\mathcal{A}}^{(t-1)} = \mathcal{F}(\boldsymbol{\alpha}^{(t-1)})$, where $\mathcal{F}(\cdot)$ denotes the 2D FFT. Following (Zheng et al., 2021), we apply the data-term solving module (DTSM), leading to the following closed-form solution:

$$\tilde{\boldsymbol{\alpha}}^{(t)} = \mathcal{G}_{\mathrm{DTSM}}\left(\boldsymbol{\alpha}^{(t-1)}, \mathbf{z}^{(t)}, \mathbf{D}, \eta\right) = \frac{1}{\eta}\mathcal{F}^{-1}\left(\boldsymbol{\mathcal{H}}^{(t)} - \boldsymbol{\mathcal{D}} \circ \left(\frac{\left(\bar{\boldsymbol{\mathcal{D}}} \odot \boldsymbol{\mathcal{H}}^{(t)}\right)}{\eta + \left(\bar{\boldsymbol{\mathcal{D}}} \odot \boldsymbol{\mathcal{D}}\right)} \uparrow_C\right)\right), \tag{7}$$

where $\circ$ is the Hadamard product, $\mathbf{X} \odot \mathbf{Y} = \sum_{i=1}^{C} \mathbf{X}_i \circ \mathbf{Y}_i$, $\mathbf{X} \uparrow_C$ expands the channel dimension of $\mathbf{X}$ to C, $\div$ is the Hadamard division, $\mathcal{F}^{-1}(\cdot)$ denotes the inverse of FFT, $\bar{\boldsymbol{\mathcal{D}}}$ denotes the complex conjugate of $\boldsymbol{\mathcal{D}}$, and $\boldsymbol{\mathcal{H}}^{(t)}$ is defined as $\boldsymbol{\mathcal{H}}^{(t)} = \boldsymbol{\mathcal{D}} \circ \left(\boldsymbol{\mathcal{Z}}^{(t)} \uparrow_C\right) + \eta\boldsymbol{\mathcal{A}}^{(t-1)}$.

For solving the Eq. (6b), we apply a prior-term solving network (PTSN) to estimate $\boldsymbol{\alpha}^{(t)}$ as follows:

$$\boldsymbol{\alpha}^{(t)} = \mathcal{G}_{\mathrm{PTSN}}\left(\tilde{\boldsymbol{\alpha}}^{(t)}, \bar{\boldsymbol{\beta}}\right), \tag{8}$$

and the structural design of PTSN will be presented in the following.

3.3 $\mathbf{D^3C^2}$-NET UNFOLDING ARCHITECTURE DESIGN

As discussed above, the unfolding optimization consists of an image-domain optimization subproblem (*i.e.*, Eq. (3a)) and a convolutional-coding-domain optimization subproblem (*i.e.*, Eq. (3b)). Mapping the unfolding process into a deep neural network, we propose our $\mathrm{D}^3\mathrm{C}^2$-Net, which alternates

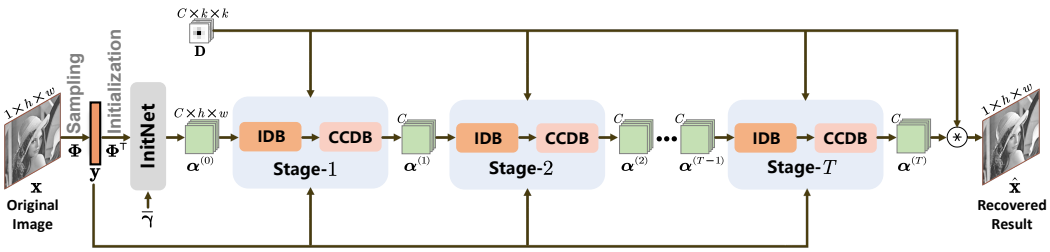


Figure 3: The overall architecture of our proposed D^{3}C^2-Net with T stages. Each stage consists of an image domain block (IDB) and a convolutional coding domain block (CCDB). $\mathbf{x}$ denotes the full-sampled image, $\mathbf{y}$ is the under-sampled data, and $\hat{\mathbf{x}}$ denotes the output of the D^{3}C^2-Net. $\mathbf{D}$ is the convolutional dictionary, and $\boldsymbol{\alpha}$ is the feature map. k denotes the filter size of $\mathbf{D}$, h and w denote the height and width of $\mathbf{x}$ and $\boldsymbol{\alpha}$, and C is the number of channels.

between the image domain block (IDB) and the convolutional coding domain block (CCDB). Fig. 3 illustrates the overall architecture of D^{3}C^2-Net with T stages, whereby the recovered result $\hat{\mathbf{x}}$ is obtained by $\hat{\mathbf{x}} = \mathbf{D} \circledast \boldsymbol{\alpha}^{(T)}$. It can be seen that the proposed D^{3}C^2-Net can transmit C-channel high-throughput information between each two adjacent stages. Fig. 4(a) gives more details about each stage. As shown in Fig. 4(a), each IDB is composed of a gradient descent module (GDM in Eq. (4a)) and a proximal mapping network (PMN in Eq. (4b)), while each CCDB is composed of a data-term solving module (DTSM in Eq. (7)) and a prior-term solving network (PTSN in Eq. (8)). Besides, for hyper-parameters $\{\rho, \mu_{\mathbf{z}}, \eta, \beta\}$, inspired by (Zhang et al., 2020b) and (Zheng et al., 2021), we adopt a hyper-parameter network (HPN) to predict them for each stage. Fig. 4(b) illustrates the architectures of the sub-networks, including InitNet, PMN, PTSN and HPN. More details are shown below.

InitNet takes the concatenation of $\mathbf{x}^{\text{init}}$ and $\bar{\boldsymbol{\gamma}}$ as input to obtain a feature map initialization $\boldsymbol{\alpha}^{(0)}$, where $\mathbf{x}^{\text{init}} = \boldsymbol{\Phi}^{\top}\mathbf{y}$, and $\bar{\boldsymbol{\gamma}}$ is the CS ratio map generated from γ with a same dimension as $\mathbf{x}$. It consists of two convolutional layers ($\text{Conv}_1(\cdot)$ and $\text{Conv}_2(\cdot)$). The former one receives 2-channel inputs and generates C-channel outputs with ReLU activation. InitNet is formulated as:

$$\boldsymbol{\alpha}^{(0)} = \mathcal{G}_{\text{InitNet}}\left(\mathbf{x}^{\text{init}}, \bar{\boldsymbol{\gamma}}\right) = \text{Conv}_2(\text{ReLU}(\text{Conv}_1(\text{Concat}(\mathbf{x}^{\text{init}}, \bar{\boldsymbol{\gamma}})))). \tag{9}$$

PMN solves the proximal mapping problem $\text{prox}_{\tau\phi}(\mathbf{r}^{(t)})$. It consists of two convolutional layers ($\text{Conv}_1(\cdot)$ and $\text{Conv}_2(\cdot)$) and two residual blocks ($\text{RB}_1(\cdot)$ and $\text{RB}_2(\cdot)$), which generate residual outputs by the structure of Conv-ReLU-Conv. Specifically, $\text{Conv}_1(\cdot)$ takes single-channel $\mathbf{r}^{(t)}$ as input and generates C-channel outputs. Then two $\text{RB}(\cdot)$s are used to extract deep representation. Finally, $\text{Conv}_2(\cdot)$ outputs the result by feature conversions from C-channel to single-channel under a residual learning strategy. Accordingly, PMN can be formulated as:

$$\mathbf{z}^{(t)} = \mathcal{G}^{(t)}_{\text{PMN}}\left(\mathbf{r}^{(t)}\right) = \mathbf{r}^{(t)} + \text{Conv}_2(\text{RB}_2(\text{RB}_1(\text{Conv}_1(\mathbf{r}^{(t)})))). \tag{10}$$

PTSN takes the concatenation of $\tilde{\boldsymbol{\alpha}}^{(t)}$ and $\bar{\boldsymbol{\beta}}^{(t)}$ as input to learn the implicit prior on feature map $\boldsymbol{\alpha}$, where $\bar{\boldsymbol{\beta}}^{(t)}$ is generated from $\beta^{(t)}$ as $\bar{\boldsymbol{\gamma}}$ does. It consists of one convolutional layer ($\text{Conv}_1(\cdot)$) and two residual blocks ($\text{RB}_1(\cdot)$ and $\text{RB}_2(\cdot)$). The convolutional layer receives $(C+1)$-channel inputs and generates C-channel outputs. A residual learning strategy is applied. PTSN is formulated as:

$$\boldsymbol{\alpha}^{(t)} = \mathcal{G}^{(t)}_{\text{PTSN}}\left(\tilde{\boldsymbol{\alpha}}^{(t)}, \bar{\boldsymbol{\beta}}^{(t)}\right) = \tilde{\boldsymbol{\alpha}}^{(t)} + \text{RB}_2(\text{RB}_1(\text{Conv}_1(\text{Concat}(\tilde{\boldsymbol{\alpha}}^{(t)}, \bar{\boldsymbol{\beta}}^{(t)})))). \tag{11}$$

HPN takes CS ratio map $\bar{\boldsymbol{\gamma}}$ as input and predicts hyper-parameters for each stage. It consists of two 1×1 convolutional layers with Sigmoid as the first activation function and Softplus as the last, ensuring all hyper-parameters are positive. HPN can be formulated as:

$$\left(\rho^{(t)}, \mu^{(t)}_{\mathbf{z}}, \eta^{(t)}, \beta^{(t)}\right) = \mathcal{G}^{(t)}_{\text{HPN}}\left(\bar{\boldsymbol{\gamma}}\right) = \text{Softplus}(\text{Conv}_2(\text{Sigmoid}(\text{Conv}_1(\bar{\boldsymbol{\gamma}})))). \tag{12}$$

To sum up, with jointly taking the sampling matrix $\boldsymbol{\Phi}$ and the recovery network as combined learnable parts, the collection of all parameters incorporated in D^{3}C^2-Net, denoted by $\boldsymbol{\Theta}$, can be collaboratively learned and expressed as $\boldsymbol{\Theta} = \{\boldsymbol{\Phi}, \mathbf{D}, \mathcal{G}_{\text{InitNet}}(\cdot)\} \cup \{\mathcal{G}^{(t)}_{\text{PMN}}(\cdot), \mathcal{G}^{(t)}_{\text{PTSN}}(\cdot), \mathcal{G}^{(t)}_{\text{HPN}}(\cdot)\}_{t=1}^{T}$.

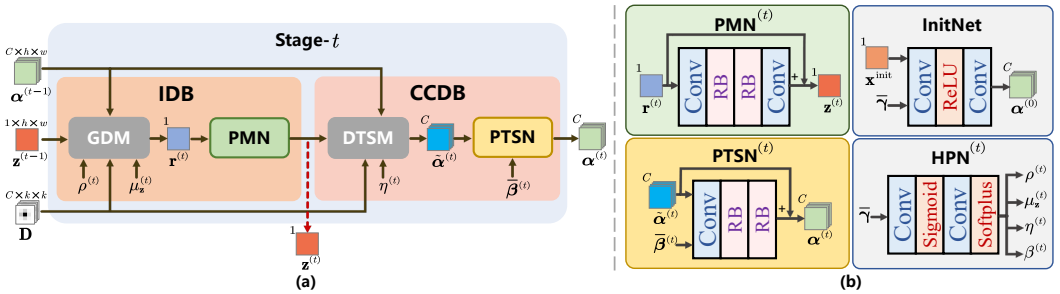


Figure 4: Illustration of the structural design of the unfolded stage and its components. (a) is the structure of t-th stage in D^{3}C^2-Net. An image domain block (IDB) consists of a gradient descent module (GDM) and a proximal mapping network (PMN). A convolutional coding domain block (CCDB) consists of a data-term solving module (DTSM) and a prior-term solving network (PTSN). (b) shows the architectures of four sub-networks in each D^{3}C^2-Net stage.

3.4 Relationship to other works

Compared with most existing DUNs, our framework minimizes the original objective function by decoupling it into image-level and feature-level optimizations. Specifically, ISTA-Net$^+$ (Zhang & Ghanem, 2018) and ISTA-Net^{++} (You et al., 2021a) are two special cases of our method when optimizing only in the image domain. DCDicL (Zheng et al., 2021) is the special case of our method applied to the image denoising task (the sampling matrix is the identity $\mathbf{I}$) and optimized only in the convolutional-coding domain. Our framework integrates their strengths to some extent so that it can be easily generalized to other image inverse problems and allow our D^{3}C^2-Net to transmit high-capacity features. Besides, we employ a universal dictionary instead of the adaptive one (Zheng et al., 2021), which avoids training-time instability and collapse, improves training speed and convergence, and follows the original definition of dictionary learning better.

4 Experiments

4.1 Implementation details

Loss function. For each batch of full-sampled images $\{\mathbf{x}_j\}_{j=1}^{N_b}$ and CS ratio γ, the measurement is obtained by $\mathbf{y}_j = \mathbf{\Phi}\mathbf{x}_j$. Our D^{3}C^2-Net takes $\mathbf{y}_j$ as the input of its recovery network and outputs the reconstruction $\hat{\mathbf{x}}_j$ with the parameter-free initialization $\mathbf{x}_j^{\text{init}} = \mathbf{\Phi}^\top \mathbf{y}_j$. Following (Zhang et al., 2020a; You et al., 2021a), we adopt the block-based CS sampling setup, where a high-dimensional image is divided into non-overlapping blocks and sampled independently, *i.e.*, $\mathbf{x}_j$, $\mathbf{x}_j^{\text{init}}$ and $\hat{\mathbf{x}}_j$ are tensors of size $1\times\sqrt{N}\times\sqrt{N}$. More details about sampling and initialization are provided in the appendix A. To reduce the discrepancy between $\mathbf{x}_j$ and $\hat{\mathbf{x}}_j$, an ℓ_2 discrepancy loss is defined by the mean square error (MSE), *i.e.*, $\mathcal{L}_{disc} = \frac{1}{NN_b}\sum_{j=1}^{N_b}\|\hat{\mathbf{x}}_j - \mathbf{x}_j\|_F^2$, where N_b and N represent the number of each training batch and the size of each image, respectively. For the orthogonal constraint of the jointly learned sampling matrix $\mathbf{\Phi}$, the orthogonal loss term is designed as $\mathcal{L}_{orth} = \frac{1}{M^2}\|\mathbf{\Phi}\mathbf{\Phi}^\top - \mathbf{I}\|_F^2$. Therefore, the end-to-end loss for D^{3}C^2-Net is defined as $\mathcal{L}(\mathbf{\Theta}) = \mathcal{L}_{disc} + \xi\mathcal{L}_{orth}$, where ξ is the regularization parameter, which is set to 0.01 in experiments.

Training. We use the combination of BSD400 (Martin et al., 2001; Chen & Pock, 2016), DIV2K training set (Timofte et al., 2017), and WED (Ma et al., 2016) for training. Training data samples are obtained by extracting the luminance component of each image block of size 32×32, *i.e.*, $N = 1024$. The data augmentation technique is applied to increase the data diversity. Our D^{3}C^2-Net is implemented in PyTorch (Paszke et al., 2019). All the experiments are performed on one NVIDIA GeForce RTX 3090. The Adam optimizer is used for updating the learnable parameters. The batch size is set to 32, and we train the network for 2.8×10^5 iterations. The learning rate starts from 1×10^{-4} and decays a factor by 0.1 after 1.6×10^5 and 2.4×10^5 iterations. The default filter size k of each dictionary filter is set to 5, the number of feature maps C is set to 64, and the stage number T is set to 8. The number of filters in $\mathbf{D}$ is determined by the number of feature maps (*i.e.*, same as C). The selection of k, C and T is discussed in Section 4.2.

4.2 Ablation study

In this section, we first discuss the selection of filter size k of $\mathbf{D}$, the number of feature maps C, and the number of stages T. Then we investigate the contribution of each domain in our dual-domain network. All the experiments are performed with CS ratio $\gamma = 30\%$.

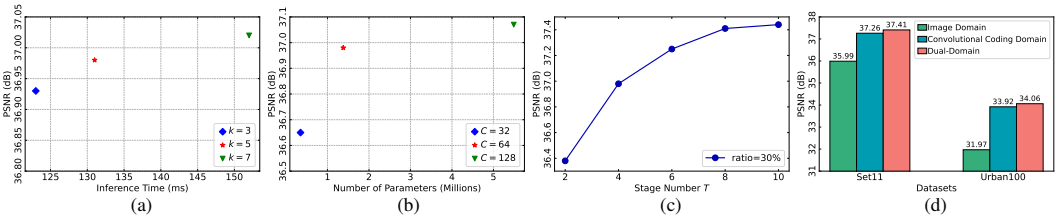


Figure 5: Ablation studies on (a) filter size k, (b) feature number C, (c) stage number T and (d) effect of each domain. The experiments of (a), (b) and (c) are performed on the Set11 benchmark.

Dictionary filter size k. We first explore the effects of dictionary filter size $k \in \{3, 5, 7\}$. As shown in Fig. 5(a), the recovery performance is improved with a larger k while the inference time increases. To balance the performance and efficiency, we choose $k = 5$ in our default $\mathrm{D}^3\mathrm{C}^2$-Net setting.

Number of feature maps C. We analyze the effect of $C \in \{32, 64, 128\}$. Fig. 5(b) provides the experimental comparison of PSNR and parameter number with different Cs. With the increase of C, on the one hand, the throughput of $\mathrm{D}^3\mathrm{C}^2$-Net improves, leading to better reconstruction performance. On the other hand, feature maps become redundant, resulting in huge network parameters and being hard to be sufficiently trained. To get a better trade-off between reconstruction performance and the network computational complexity, we choose $C = 64$ by default in $\mathrm{D}^3\mathrm{C}^2$-Net.

Number of unfolded stages T. Since each $\mathrm{D}^3\mathrm{C}^2$-Net stage corresponds to one iteration in our dual-domain unfolding framework, it is expected that a larger T will lead to a higher reconstruction accuracy. Fig. 5(c) investigates the performances of five $\mathrm{D}^3\mathrm{C}^2$-Net variants with $T \in \{2, 4, 6, 8, 10\}$. We observe that PSNR rises as T increases, but the improvement becomes minor when $T \geq 8$. Considering the recovery accuracy-efficiency trade-offs, we employ $T = 8$ in $\mathrm{D}^3\mathrm{C}^2$-Net by default.

Effect of dual-domain constraints. To analyze the effectiveness of dual-domain priors, we compare our $\mathrm{D}^3\mathrm{C}^2$-Net with two single-domain-based networks. A representative image-domain-only network OPINE-Net$^+$ (Zhang et al., 2020a) is adopted for evaluation, each stage of which is similar to our IDB composed of a GDM and a PMN. To conduct a convolutional-coding-domain-only network, we remove the image-domain prior $\phi(\mathbf{z})$ in Eq. (3a), leading to the removal of PMN in $\mathrm{D}^3\mathrm{C}^2$-Net for comparison. Fig. 5(d) shows the recovery performances of three different networks. It is clear to see that due to the enhancement of information transmission capability, the convolutional-coding-domain-only network boosts performance by 1.27dB on Set11 and 1.95dB on Urban100 over the image-domain-only network. Moreover, the combination of image and convolutional-coding domain priors (*i.e.*, our default $\mathrm{D}^3\mathrm{C}^2$-Net design) further improves the performance by about 0.15dB on both benchmarks, which demonstrates the effectiveness of dual-domain constraints.

4.3 Comparison with state-of-the-art methods

We compare our $\mathrm{D}^3\mathrm{C}^2$-Net with five advanced CS methods, including CSNet$^+$ (Shi et al., 2019a), SCSNet (Shi et al., 2019b), OPINE-Net$^+$ (Zhang et al., 2020a), AMP-Net (Zhang et al., 2020c), and MADUN (Song et al., 2021). The average PSNR and SSIM reconstruction performance on Set11 (Kulkarni et al., 2016) and Urban100 (Huang et al., 2015) datasets with respect to five CS ratios are summarized in Table 2. It can be observed that our $\mathrm{D}^3\mathrm{C}^2$-Net outperforms all the other competing methods both in PSNR and SSIM under all given CS ratios, especially for lower ones. Fig. 6 further shows the visual comparison of two challenging images from Set11 and Urban100 datasets, respectively. As we can see, our $\mathrm{D}^3\mathrm{C}^2$-Net recovers richer textures and details than all other methods. Furthermore, we verify that the parameters in $\mathrm{D}^3\mathrm{C}^2$-Net are used more rationally than MADUN (Song et al., 2021), which directly introduces intermediate results as auxiliary information to transmit between stages without changing the idea of image-domain-based unfolding. As shown in Table 1, compared with MADUN, $\mathrm{D}^3\mathrm{C}^2$-Net uses fewer parameters while improving PSNR by 1.31dB on Urban100 with $\gamma = 10\%$, which validates the stronger learning capability of $\mathrm{D}^3\mathrm{C}^2$-Net from our dual-domain unfolding principle.

Table 1: Parameter number (Millions) and recovery PSNR (dB) comparisons between MADUN and our $\mathrm{D}^3\mathrm{C}^2$-Net on Urban100 dataset with $\gamma = 10\%$.

Methods	#Param.	PSNR
MADUN	3.13	26.23
$\mathrm{D}^3\mathrm{C}^2$-Net (Ours)	**2.72**	**27.54**

4.4 Application to compressive sensing MRI

To demonstrate the generality of $\mathrm{D}^3\mathrm{C}^2$-Net, we directly extend it to the practical problem of CS-MRI reconstruction, which aims at restoring MR images from a small number of under-sampled data in

Table 2: Average PSNR (dB) and SSIM performance comparisons on Set11 and Urban100 datasets with five different levels of CS ratios (or sampling rates). We compare our D^{3}C^2-Net with five prior arts. The best and second best results are highlighted in red and blue colors, respectively.

Dataset	Methods	CS Ratio γ				
		10%	20%	30%	40%	50%
Set11	CSNet$^+$	28.34/0.8580	31.66/0.9203	34.30/0.9490	36.48/0.9644	38.52/0.9749
	SCSNet	28.52/0.8616	31.82/0.9215	34.64/0.9511	36.92/0.9666	39.01/0.9769
	OPINE-Net$^+$	29.81/0.8904	33.43/0.9392	35.99/0.9596	38.24/0.9718	40.19/0.9800
	AMP-Net	29.40/0.8779	33.33/0.9345	36.03/0.9586	38.28/0.9715	40.34/0.9804
	MADUN	29.89/0.8982	34.09/0.9478	36.90/0.9671	39.14/0.9769	40.75/0.9831
	D^{3}C^2-Net (Ours)	30.80/0.9061	34.64/0.9512	37.41/0.9684	39.49/0.9773	41.29/0.9836
Urban100	CSNet$^+$	23.96/0.7309	26.95/0.8449	29.12/0.8974	30.98/0.9273	32.76/0.9484
	SCSNet	24.22/0.7394	27.09/0.8485	29.41/0.9016	31.38/0.9321	33.31/0.9534
	OPINE-Net$^+$	25.90/0.7979	29.38/0.8902	31.97/0.9309	34.27/0.9548	36.28/0.9697
	AMP-Net	25.32/0.7747	29.01/0.8799	31.63/0.9248	33.88/0.9511	35.91/0.9673
	MADUN	26.23/0.8250	30.24/0.9108	33.00/0.9457	35.10/0.9639	36.69/0.9746
	D^{3}C^2-Net (Ours)	27.54/0.8464	30.98/0.9161	34.06/0.9522	36.11/0.9676	37.89/0.9771

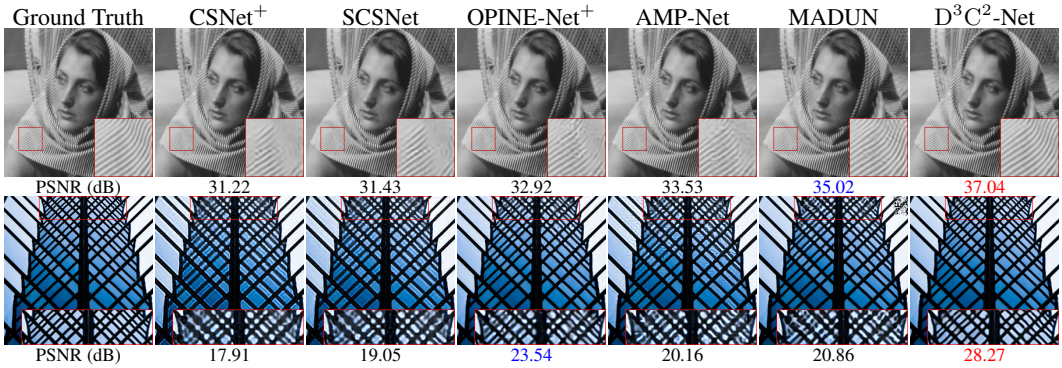


Figure 6: Visual comparisons on recovering an image named "Barbara" from Set11 dataset with CS ratio $\gamma = 30\%$ (top) and an image from Urban100 dataset with CS ratio $\gamma = 10\%$ (bottom).

k-space. We follow the common practices in this application, setting the sampling matrix $\mathbf{\Phi}$ in Eq. (1) to $\mathbf{\Phi} = \mathbf{BF}$, where $\mathbf{B}$ is an under-sampling matrix and $\mathbf{F}$ is the discrete Fourier transform. We follow MADUN (Song et al., 2021) to use the same 100 fully sampled brain MR images as the training set. To avoid overfitting on this small data collection, we reduce the width and increase the depth of the network, yielding D^{3}C^2-Net for MRI with $C = 32$ and $T = 20$, whose parameter number (1.72M) is fewer than D^{3}C^2-Net for natural images (2.72M). The geometric data augmentation technique is also applied to increase the data diversity. As shown in Table 3, our D^{3}C^2-Net outperforms the state-of-the-art methods on the brain dataset under all given ratios. It is worth emphasizing that the PSNR is already high under a big ratio, thus making PSNR improvement more difficult. What is more, due to the introduction of CS ratio information in InitNet and HPN, our D^{3}C^2-Net for MRI is scalable for different ratios, *i.e.*, it can handle five ratios by a single model, which significantly reduces the overall parameter number. Compared with MADUN (3.13M for each ratio), our D^{3}C^2-Net leverages only about $\times\frac{1}{9}$ parameters while achieving better reconstruction performance on the CS-MRI task.

4.5 Analysis of the learned dictionary $\mathbf{D}$ and feature map $\boldsymbol{\alpha}$

To further analyze the image representation capability of our D^{3}C^2-Net, we visualize the learned convolutional dictionary $\mathbf{D}$ of the model and the final estimated feature maps on Set11 in the case of $\gamma = 30\%$. Figs. 7(a) and (b) show some randomly picked dictionary filters $\boldsymbol{d}_i$ and feature maps $\boldsymbol{\alpha}_i$, respectively. It can be seen that our feature maps are not so sparse compared with those in convolutional sparse coding methods (Fu et al., 2019; Gao et al., 2022) that explicitly impose sparsity priors. Interestingly, we observe that there is always one channel to preserve low-frequency information in our learned feature maps across the unfolded stage-by-stage inferences,

Table 3: Average PSNR/SSIM performance comparisons on testing brain MR images with eight recent methods. The best and second best results are highlighted in red and blue colors, respectively.

Methods	CS Ratio				
	10%	20%	30%	40%	50%
Hyun et al.	32.78/0.8385	36.36/0.9070	38.85/0.9383	40.65/0.9539	42.35/0.9662
Schlemper et al.	34.23/0.8921	38.47/0.9457	40.85/0.9628	42.63/0.9724	44.19/0.9794
ADMM-Net	34.42/0.8971	38.60/0.9478	40.87/0.9633	42.58/0.9726	44.19/0.9796
RDN	34.59/0.8968	38.58/0.9470	40.82/0.9625	42.64/0.9723	44.18/0.9793
CDDN	34.63/0.9002	38.59/0.9474	40.89/0.9633	42.59/0.9725	44.15/0.9795
ISTA-Net$^+$	34.65/0.9038	38.67/0.9480	40.91/0.9631	42.65/0.9727	44.24/0.9798
MoDL	35.18/0.9091	38.51/0.9457	40.97/0.9636	42.38/0.9705	44.20/0.9776
MADUN	36.15/0.9237	39.44/0.9542	41.48/0.9666	43.06/0.9746	44.60/0.9810
D^{3}C^2-Net (Ours)	36.48/0.9289	39.66/0.9558	41.59/0.9671	43.14/0.9748	44.63/0.9811

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

Figure 7: Visualizations for analyzing the learned dictionary $\mathbf{D}$ and feature maps $\boldsymbol{\alpha}$, including (a) learned global dictionary filters $\boldsymbol{d}_i$ in every four channels, whose values are distributed in $[-0.17, 0.23]$, (b) feature maps $\boldsymbol{\alpha}_9$, $\boldsymbol{\alpha}_{25}$, $\boldsymbol{\alpha}_{41}$ and $\boldsymbol{\alpha}_{57}$, (c) low-frequency information $\boldsymbol{d}_{25} * \boldsymbol{\alpha}_{25}$ and the complementary (d) high-frequency information $\sum_{i\neq 25} \boldsymbol{d}_i * \boldsymbol{\alpha}_i$, with applying D^{3}C^2-Net to the image named "Cameraman" from Set11 with $\gamma = 30\%$. D^{3}C^2-Net learns diverse dictionary filters through end-to-end optimization with the recovery network trunk and obtains better image representations than prior arts by clearly separating the (c) low- and (d) high-frequency components.

e.g., the 25^{th}-channel shown in Fig. 7(b) with a red border. We thus visualize $\boldsymbol{d}_{25} * \boldsymbol{\alpha}_{25}$ and its complementary $\sum_{i=1,i\neq 25}^{64} \boldsymbol{d}_i * \boldsymbol{\alpha}_i$ in Figs. 7(c) and (d), respectively. (Please refer to Appendix D for more visualizations.) It is clear to see that our D^{3}C^2-Net represents the image as the sum of one-layer low-frequency information and multi-layer high-frequency information through convolutional coding, which may make D^{3}C^2-Net easier to keep and transmit high-frequency information among different stages in such a long trunk, thus achieving better reconstruction accuracies compared with prior arts.

5 Conclusion

Inspired by convolutional coding methods, we propose a generalized dual-domain unfolding framework that combines the merits of both image-domain and convolutional-coding-domain priors to constrain the feasible region of the solution space. Compared with most existing convolutional coding methods, on the one hand, our framework adopts deep priors rather than the traditional sparsity (Fu et al., 2019; Xu et al., 2020; Sreter & Giryes, 2018; Gao et al., 2022) to better leverage the learning capability of deep neural networks. On the other hand, our framework is more general, while existing deep convolutional coding methods for image restoration are exceptional cases where the degradation matrix $\boldsymbol{\Phi}$ is the identity $\mathbf{I}$ (Wang et al., 2020; Zheng et al., 2021). Based on our proposed framework, we further design a novel **D**ual-**D**omain **D**eep **C**onvolutional **C**oding **Net**work for compressive sensing (CS) imaging, dubbed D^{3}C^2-Net. Compared with most existing CS DUNs (Zhang & Ghanem, 2018; Zhang et al., 2020a; You et al., 2021b;a), our D^{3}C^2-Net transmits high-capacity feature-level representation through all stages and captures sufficient features adaptively. Extensive CS experiments on both natural and MR images demonstrate that D^{3}C^2-Net outperforms state-of-the-art network-based CS methods with large accuracy margins and lower complexities. In the future, we will extend our generalizable unfolding framework and D^{3}C^2-Net to more inverse imaging tasks and video applications.

REFERENCES

Hemant K Aggarwal, Merry P Mani, and Mathews Jacob. MoDL: Model-based deep learning architecture for inverse problems. *IEEE Transactions on Medical Imaging*, 38(2):394–405, 2018.

Richard G Baraniuk. Compressive sensing [lecture notes]. *IEEE Signal Processing Magazine*, 24(4): 118–121, 2007.

Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.

Jiwei Chen, Yubao Sun, Qingshan Liu, and Rui Huang. Learning memory augmented cascading network for compressed sensing of images. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.

Yunjin Chen and Thomas Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1256–1272, 2016.

Xin Deng and Pier Luigi Dragotti. Deep convolutional neural network for multi-modal image restoration and fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10): 3333–3348, 2020.

Weisheng Dong, Peiyao Wang, Wotao Yin, Guangming Shi, Fangfang Wu, and Xiaotong Lu. Denoising prior driven deep neural network for image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(10):2305–2318, 2018.

Marco F Duarte, Mark A Davenport, Dharmpal Takhar, Jason N Laska, Ting Sun, Kevin F Kelly, and Richard G Baraniuk. Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*, 25(2):83–91, 2008.

Xueyang Fu, Zheng-Jun Zha, Feng Wu, Xinghao Ding, and John Paisley. Jpeg artifacts reduction via deep convolutional sparse coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Fangyuan Gao, Xin Deng, Mai Xu, Jingyi Xu, and Pier Luigi Dragotti. Multi-modal convolutional dictionary learning. *IEEE Transactions on Image Processing*, 2022.

Davis Gilton, Greg Ongie, and Rebecca Willett. Neumann networks for linear inverse problems in imaging. *IEEE Transactions on Computational Imaging*, 6:328–343, 2019.

Shuhang Gu, Wangmeng Zuo, Qi Xie, Deyu Meng, Xiangchu Feng, and Lei Zhang. Convolutional sparse coding for image super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Chang Min Hyun, Hwa Pyung Kim, Sung Min Lee, Sungchul Lee, and Jin Keun Seo. Deep learning for undersampled MRI reconstruction. *Physics in Medicine & Biology*, 63(13):135007, 2018.

Michael Iliadis, Leonidas Spinoulas, and Aggelos K Katsaggelos. Deep fully-connected networks for video compressive sensing. *Digital Signal Processing*, 72:9–18, 2018.

Yookyung Kim, Mariappan S Nadar, and Ali Bilgin. Compressed sensing using a Gaussian scale mixtures model in wavelet domain. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, 2010.

Filippos Kokkinos and Stamatios Lefkimmiatis. Deep image demosaicking using a cascade of convolutional residual denoising networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

Jakob Kruse, Carsten Rother, and Uwe Schmidt. Learning to push the limits of efficient FFT-based image deconvolution. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

Kuldeep Kulkarni, Suhas Lohit, Pavan Turaga, Ronan Kerviche, and Amit Ashok. Reconnet: Non-iterative reconstruction of images from compressively sensed measurements. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Stamatios Lefkimmiatis. Non-local color image denoising with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Minghan Li, Qi Xie, Qian Zhao, Wei Wei, Shuhang Gu, Jing Tao, and Deyu Meng. Video rain streak removal by multiscale convolutional sparse coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Michael Lustig, David Donoho, and John M Pauly. Sparse MRI: The application of compressed sensing for rapid mr imaging. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 58(6):1182–1195, 2007.

Kede Ma, Zhengfang Duanmu, Qingbo Wu, Zhou Wang, Hongwei Yong, Hongliang Li, and Lei Zhang. Waterloo exploration database: New challenges for image quality assessment models. *IEEE Transactions on Image Processing*, 26(2):1004–1016, 2016.

David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2001.

Chris Metzler, Ali Mousavi, and Richard Baraniuk. Learned D-AMP: Principled neural network based compressive image recovery. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

Christopher A Metzler, Arian Maleki, and Richard G Baraniuk. From denoising to compressed sensing. *IEEE Transactions on Information Theory*, 62(9):5117–5144, 2016.

Ali Mousavi, Ankit B Patel, and Richard G Baraniuk. A deep learning approach to structured signal recovery. In *Proceedings of the Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2015.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Florian Rousset, Nicolas Ducros, Andrea Farina, Gianluca Valentini, Cosimo D'Andrea, and Françoise Peyrin. Adaptive basis scan by wavelet prediction for single-pixel imaging. *IEEE Transactions on Computational Imaging*, 3(1):36–46, 2016.

Jo Schlemper, Jose Caballero, Joseph V Hajnal, Anthony N Price, and Daniel Rueckert. A deep cascade of convolutional neural networks for dynamic MR image reconstruction. *IEEE Transactions on Medical Imaging*, 37(2):491–503, 2017.

Wuzhen Shi, Feng Jiang, Shaohui Liu, and Debin Zhao. Image compressed sensing using convolutional neural network. *IEEE Transactions on Image Processing*, 29:375–388, 2019a.

Wuzhen Shi, Feng Jiang, Shaohui Liu, and Debin Zhao. Scalable convolutional neural network for image compressed sensing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019b.

Jiechong Song, Bin Chen, and Jian Zhang. Memory-augmented deep unfolding network for compressive sensing. In *Proceedings of the 29th ACM International Conference on Multimedia*, 2021.

Hillel Sreter and Raja Giryes. Learned convolutional sparse coding. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

Yueming Su and Qiusheng Lian. iPiano-Net: Nonconvex optimization inspired multi-scale reconstruction network for compressed sensing. *Signal Processing: Image Communication*, 89:115989, 2020.

Liyan Sun, Zhiwen Fan, Yue Huang, Xinghao Ding, and John Paisley. Compressed sensing MRI using a recursive dilated network. In *Proceedings of the Conference on Association for the Advancement of Artificial Intelligence (AAAI)*, 2018.

Yubao Sun, Jiwei Chen, Qingshan Liu, Bo Liu, and Guodong Guo. Dual-path attention network for compressed sensing image reconstruction. *IEEE Transactions on Image Processing*, 29:9482–9495, 2020.

Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. NTIRE 2017 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017.

Hong Wang, Qi Xie, Qian Zhao, and Deyu Meng. A model-driven deep neural network for single image rain removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Zhuoyuan Wu, Jian Zhang, and Chong Mou. Dense deep unfolding network with 3d-cnn prior for snapshot compressive imaging. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021a.

Zhuoyuan Wu, Zhenyu Zhang, Jiechong Song, and Jian Zhang. Spatial-temporal synergic prior driven unfolding network for snapshot compressive imaging. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, 2021b.

Moran Xu, Dianlin Hu, Fulin Luo, Fenglin Liu, Shaoyu Wang, and Weiwen Wu. Limited-angle X-Ray CT reconstruction using image gradient ℓ_0-norm with dictionary learning. *IEEE Transactions on Radiation and Plasma Medical Sciences*, 5(1):78–87, 2020.

Yan Yang, Jian Sun, Huibin Li, and Zongben Xu. ADMM-CSNet: A deep learning approach for image compressive sensing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42 (3):521–538, 2018.

Di You, Jingfen Xie, and Jian Zhang. ISTA-Net++: flexible deep unfolding network for compressive sensing. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, 2021a.

Di You, Jian Zhang, Jingfen Xie, Bin Chen, and Siwei Ma. COAST: Controllable arbitrary-sampling network for compressive sensing. *IEEE Transactions on Image Processing*, 30:6066–6080, 2021b.

Jian Zhang and Bernard Ghanem. ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Jian Zhang, Chen Zhao, Debin Zhao, and Wen Gao. Image compressive sensing recovery using adaptively learned sparsifying basis via L0 minimization. *Signal Processing*, 103:114–126, 2014a.

Jian Zhang, Debin Zhao, and Wen Gao. Group-based sparse representation for image restoration. *IEEE Transactions on Image Processing*, 23(8):3336–3351, 2014b.

Jian Zhang, Chen Zhao, and Wen Gao. Optimization-inspired compact deep compressive sensing. *IEEE Journal of Selected Topics in Signal Processing*, 14(4):765–774, 2020a.

Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep CNN denoiser prior for image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Kai Zhang, Luc Van Gool, and Radu Timofte. Deep unfolding network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020b.

Xuanyu Zhang, Yongbing Zhang, Ruiqin Xiong, Qilin Sun, and Jian Zhang. HerosNet: Hyperspectral explicable reconstruction and optimal sampling deep network for snapshot compressive imaging. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Zhonghao Zhang, Yipeng Liu, Jiani Liu, Fei Wen, and Ce Zhu. AMP-Net: Denoising-based deep unfolding for compressive image sensing. *IEEE Transactions on Image Processing*, 30:1487–1500, 2020c.

Chen Zhao, Siwei Ma, and Wen Gao. Image compressive-sensing recovery using structured laplacian sparsity in DCT domain and multi-hypothesis prediction. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME)*, 2014.

Chen Zhao, Siwei Ma, Jian Zhang, Ruiqin Xiong, and Wen Gao. Video compressive sensing reconstruction via reweighted residual sparsity. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(6):1182–1195, 2016a.

Chen Zhao, Jian Zhang, Siwei Ma, and Wen Gao. Compressive-sensed image coding via stripe-based DPCM. In *Proceedings of the Data Compression Conference (DCC)*, 2016b.

Chen Zhao, Jian Zhang, Siwei Ma, and Wen Gao. Nonconvex Lp nuclear norm based ADMM framework for compressed sensing. In *Proceedings of the Data Compression Conference (DCC)*, 2016c.

Chen Zhao, Jian Zhang, Ronggang Wang, and Wen Gao. CREAM: CNN-regularized ADMM framework for compressive-sensed image reconstruction. *IEEE Access*, 6:76838–76853, 2018.

Hao Zheng, Faming Fang, and Guixu Zhang. Cascaded dilated dense network with two-step data consistency for MRI reconstruction. In *Proceedings of the International Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

Hongyi Zheng, Hongwei Yong, and Lei Zhang. Deep convolutional dictionary learning for image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

APPENDIX

A IMPLEMENTATION DETAILS

In this section, we show our simulations of CS sampling and $\mathrm{D^3C^2}$-Net initialization and more implementation details of $\mathrm{D^3C^2}$-Net components.

A.1 SAMPLING AND INITIALIZATION SIMULATIONS

In our experiments, we follow Zhang et al. (2020a); You et al. (2021a) to adopt the block-based CS sampling-initialization setup and mimic the CS sampling and initialization processes using bias-free convolution operators, *i.e.*, we use the whole image $\mathbf{x} \in \mathbb{R}^{1\times h\times w}$ instead of one vectorized image block $\mathbf{x} \in \mathbb{R}^N$ in our $\mathrm{D^3C^2}$-Net implementations, where h and w are multiples of $\sqrt{N}$. Specifically, N is set to 1024. For CS sampling, we reshape the learned sampling matrix $\mathbf{\Phi} \in \mathbb{R}^{M\times 1024}$ into M filters with the kernel size of 32×32, mimic the block-wise sampling process $\mathbf{y} = \mathbf{\Phi x}$ equivalently by adopting the convolutional layer with a stride of 32 and extend it to the whole image, *i.e.*, each image $\mathbf{x} \in \mathbb{R}^{1\times h\times w}$ can be considered as $(h/32)\times(w/32)$ non-overlapped image blocks of size $1\times 32\times 32$, and measurement $\mathbf{y}$ is a tensor of size $M \times (h/32) \times (w/32)$ after sampling. Correspondingly, the block-wise initialization $\mathbf{x}^{\text{init}} = \mathbf{\Phi}^\top \mathbf{y}$ is implemented by the bias-free transposed convolution operation with the same kernel weights as $\mathbf{\Phi}$. Hence, $\mathbf{x}^{\text{init}}$ is an image-domain tensor of the same size $1 \times h \times w$ as $\mathbf{x}$. In the learning process of $\mathrm{D^3C^2}$-Net, patches of size 96×96 are randomly cropped and served as training samples, *i.e.*, h and w are set to 96 for network training.

A.2 STRUCTURAL DESIGN DETAILS OF $\mathrm{D^3C^2}$-NET

InitNet takes the concatenation of $\mathbf{x}^{\text{init}}$ and $\bar{\boldsymbol{\gamma}}$ as input to obtain a feature map initialization $\boldsymbol{\alpha}^{(0)}$, as

$$\boldsymbol{\alpha}^{(0)} = \mathcal{G}_{\text{InitNet}}\left(\mathbf{x}^{\text{init}}, \bar{\boldsymbol{\gamma}}\right) = \mathrm{Conv}_2(\mathrm{ReLU}(\mathrm{Conv}_1(\mathrm{Concat}(\mathbf{x}^{\text{init}}, \bar{\boldsymbol{\gamma}})))). \tag{13}$$

Specifically:

- $\bar{\boldsymbol{\gamma}}$ is the CS ratio map generated from γ with a same dimension as $\mathbf{x}$, all entries of which is filled by the value of γ and implemented by the `torch.repeat` API of PyTorch (Paszke et al., 2019) framework.
- $\mathrm{Conv}_1(\cdot)$ takes a 2-channel input (*i.e.*, the channel-wise concatenation of $\mathbf{x}^{\text{init}} \in \mathbb{R}^{1\times h\times w}$ and $\bar{\boldsymbol{\gamma}} \in \mathbb{R}^{1\times h\times w}$) and generates a C-channel output with a ReLU activation.
- $\mathrm{Conv}_2(\cdot)$ takes a C-channel input and generates a C-channel output ($\{\boldsymbol{\alpha}^{(0)}\}$).

PMN solves the proximal mapping problem $\mathrm{prox}_{\tau\phi}(\mathbf{r}^{(t)})$, as

$$\mathbf{z}^{(t)} = \mathcal{G}^{(t)}_{\text{PMN}}\left(\mathbf{r}^{(t)}\right) = \mathbf{r}^{(t)} + \mathrm{Conv}_2(\mathrm{RB}_2(\mathrm{RB}_1(\mathrm{Conv}_1(\mathbf{r}^{(t)})))). \tag{14}$$

Specifically:

- $\mathrm{Conv}_1(\cdot)$ takes an 1-channel $\mathbf{r}^{(t)}$ as input and generates a C-channel output.
- $\mathrm{RB}_1(\cdot)$ and $\mathrm{RB}_2(\cdot)$ are two residual blocks. Each residual block takes a C-channel input and generates a C-channel residual output by the structure of Conv-ReLU-Conv, *i.e.*, $\mathrm{RB}(\mathbf{x}) = \mathbf{x} + \mathrm{Conv}(\mathrm{ReLU}(\mathrm{Conv}(\mathbf{x})))$.
- $\mathrm{Conv}_2(\cdot)$ converts a C-channel input to an single-channel output $\mathbf{z}^{(t)}$ under a residual learning strategy.

PTSN takes the concatenation of $\tilde{\boldsymbol{\alpha}}^{(t)}$ and $\bar{\boldsymbol{\beta}}^{(t)}$ as input to learn the implicit prior on feature map $\boldsymbol{\alpha}$, as

$$\boldsymbol{\alpha}^{(t)} = \mathcal{G}^{(t)}_{\text{PTSN}}\left(\tilde{\boldsymbol{\alpha}}^{(t)}, \bar{\boldsymbol{\beta}}^{(t)}\right) = \tilde{\boldsymbol{\alpha}}^{(t)} + \mathrm{RB}_2(\mathrm{RB}_1(\mathrm{Conv}_1(\mathrm{Concat}(\tilde{\boldsymbol{\alpha}}^{(t)}, \bar{\boldsymbol{\beta}}^{(t)})))). \tag{15}$$

- $\bar{\boldsymbol{\beta}}^{(t)}$ is generated from $\beta^{(t)}$ as similar to the $\bar{\boldsymbol{\gamma}}$ generation.
- $\mathrm{Conv}_1(\cdot)$ takes a $(C+1)$-channel input (*i.e.*, the concatenation of $\tilde{\boldsymbol{\alpha}}^{(t)} \in \mathbb{R}^{C\times h\times w}$ and $\bar{\boldsymbol{\beta}}^{(t)} \in \mathbb{R}^{1\times h\times w}$) and generates a C-channel output.

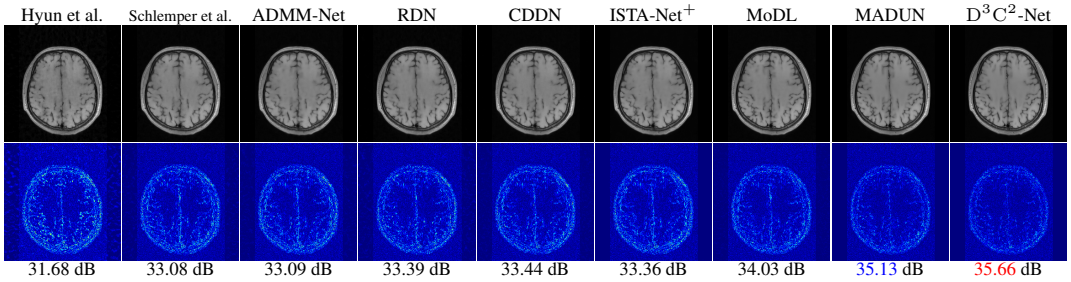


Figure 8: Visual comparisons on reconstructed images (top) and residual error to ground truth (bottom) of nine CS-MRI reconstruction methods when being applied to an image named "brain-test-13.png" from testing brain dataset Zhang & Ghanem (2018) in the case of CS ratio $\gamma = 10\%$.

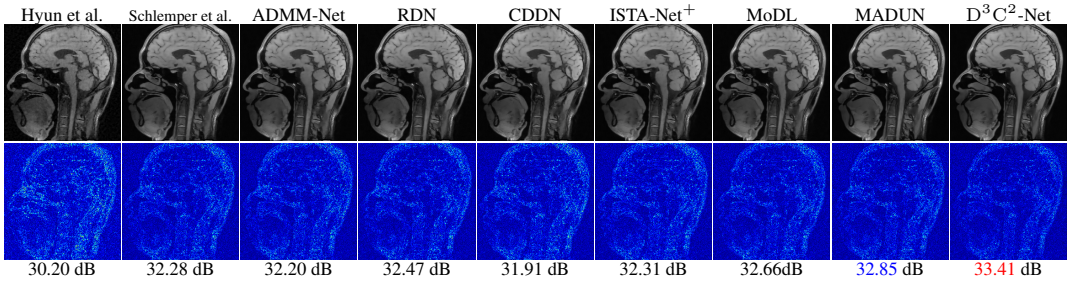


Figure 9: Visual comparisons on reconstructed images and error images of nine CS-MRI reconstruction methods when being applied to "brain-test-49.png" from testing brain dataset (Zhang & Ghanem, 2018) with $\gamma = 20\%$.

- Two residual blocks $\mathrm{RB}_1(\cdot)$ and $\mathrm{RB}_2(\cdot)$ are used to extract deep representations as those in PMN.
- The residual learning strategy is also applied in PTSN.

HPN takes CS ratio map $\bar{\boldsymbol{\gamma}}$ as input and predicts hyper-parameters for each stage, as

$$\left(\rho^{(t)}, \mu_{\mathbf{z}}^{(t)}, \eta^{(t)}, \beta^{(t)}\right) = \mathcal{G}_{\mathrm{HPN}}^{(t)}\left(\bar{\boldsymbol{\gamma}}\right) = \mathrm{Softplus}(\mathrm{Conv}_2(\mathrm{Sigmoid}(\mathrm{Conv}_1(\bar{\boldsymbol{\gamma}})))), \tag{16}$$

- $\mathrm{Conv}_1(\cdot)$ takes the CS ratio map $\bar{\boldsymbol{\gamma}}$ as input and generates a 256-channel output with Sigmoid activation (with a kennel size of 1).
- $\mathrm{Conv}_2(\cdot)$ takes a 256-channel input and generates four hyper-parameters with Softplus activation, ensuring all output elements are positive (with a kernel size of 1).

B VISUAL COMPARISONS ON MR IMAGES

The visual comparison results on two MR images and their error images compared with ground truth are shown in Figs. 8 and 9. One can see that our $\mathrm{D}^3\mathrm{C}^2$-Net can produce high-accuracy reconstructions with smaller overall errors and clearer brain tissue details than other competing methods, thus verifying the superiority and generalizability of the proposed method.

C MORE VISUAL COMPARISONS ON NATURAL IMAGES

More visual comparison results on four natural images are exhibited in Figs. 10, 11, 12, and 13. From Fig. 10, we observe that $\mathrm{D}^3\mathrm{C}^2$-Net recovers more reliable texture details (*e.g.*, patterns of walls and bricks) which are not captured in other methods. From Figs. 11, 12, and 13, one can see that our $\mathrm{D}^3\mathrm{C}^2$-Net recovers with better lines and stripes, fewer visible artifacts and less blurry effect than other competing methods.

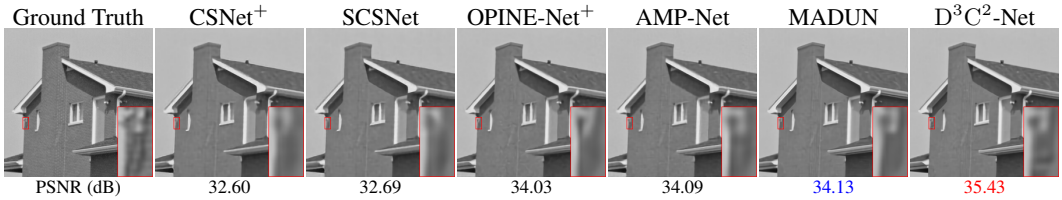


Figure 10: Visual comparisons on an image named "house" from Set11 (Kulkarni et al., 2016) with $\gamma = 10\%$.

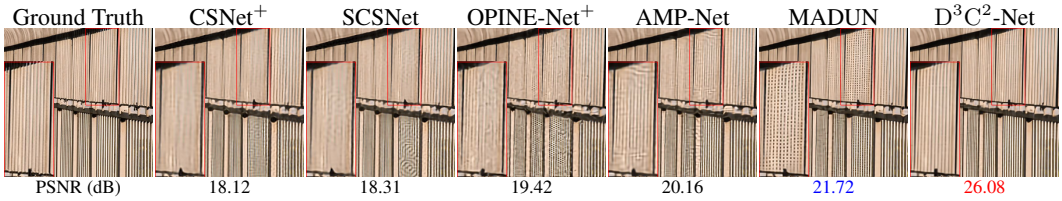


Figure 11: Visual comparisons on an image named "img_024" from Urban100 (Huang et al., 2015) with $\gamma = 10\%$.

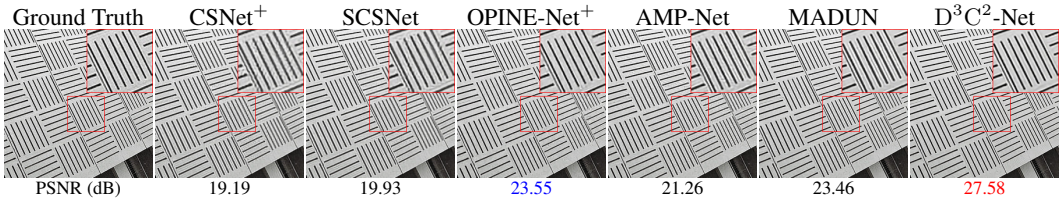


Figure 12: Visual comparisons on an image named "img_092" from Urban100 (Huang et al., 2015) with $\gamma = 30\%$.

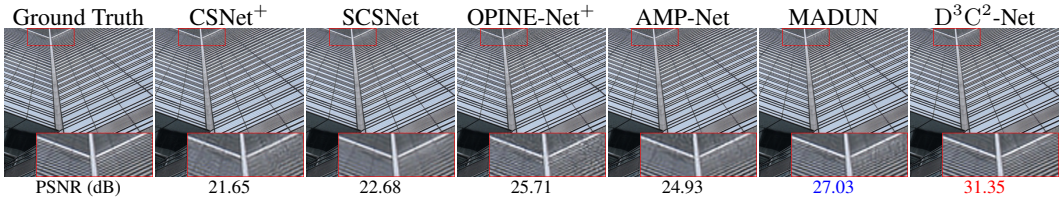
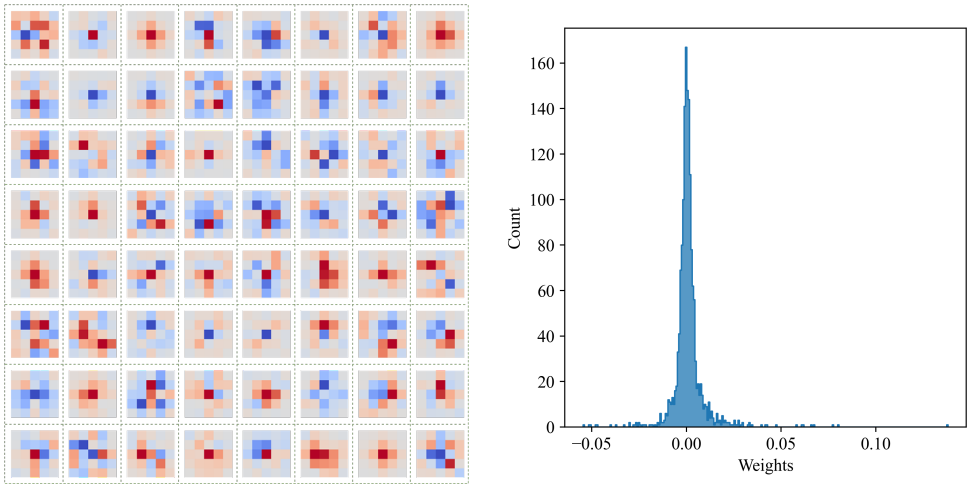


Figure 13: Visual comparisons on an image named "img_059" from Urban100 (Huang et al., 2015) with $\gamma = 50\%$.

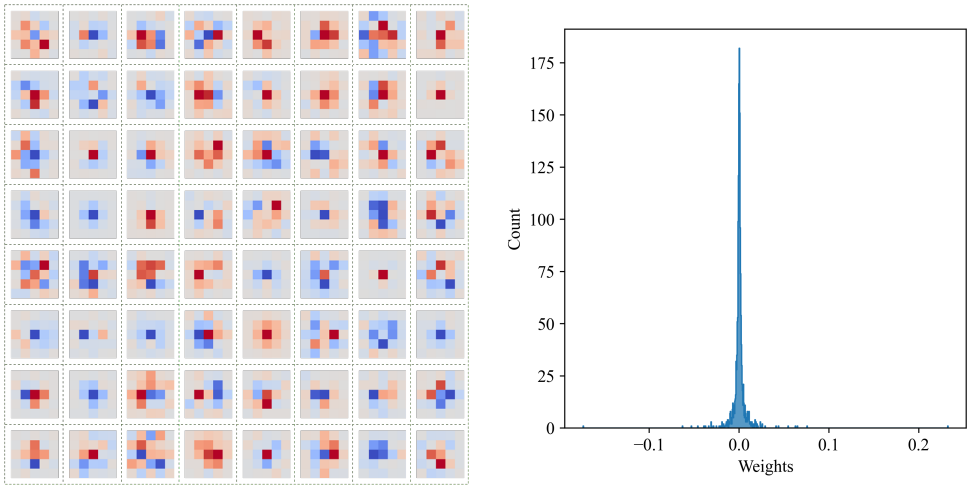
D MORE VISUALIZATIONS OF LEARNED DICTIONARY $\mathbf{D}$ AND FEATURE MAP $\boldsymbol{\alpha}$

More visualizations of the learned convolutional dictionary filters with different CS ratios and the corresponding distributions of their weights are shown in Figs 14, 15 and 16. It can be seen that the learned kernel weights are sparse. Moreover, one can observe that different filters exhibit diverse and anisotropic spatial distributions, which allows them to extract the gradients in different directions according to their learned patterns. More visualizations of the final estimated feature maps with different CS ratios are shown in Figs 17, 18 and 19. One can observe that there is always one channel to hold low-frequency information in our feature maps, *e.g.*, the 2^{th}-channel in Fig. 17, the 25^{th}-channel in Fig. 18 and the 8^{th}-channel in Fig. 19. More visualizations of the low-frequency information and their complementary high-frequency (sparse) information are shown in Figs 20, 21 and 22. It is clear to see that our D^{3}C^2-Net represents the image as the sum of one-layer low-frequency and multi-layer high-frequency information through convolutional coding, which may make D^{3}C^2-Net easier to keep and transmit high-frequency information among different stages in such a long trunk, thus achieving better reconstruction accuracies compare with prior arts.



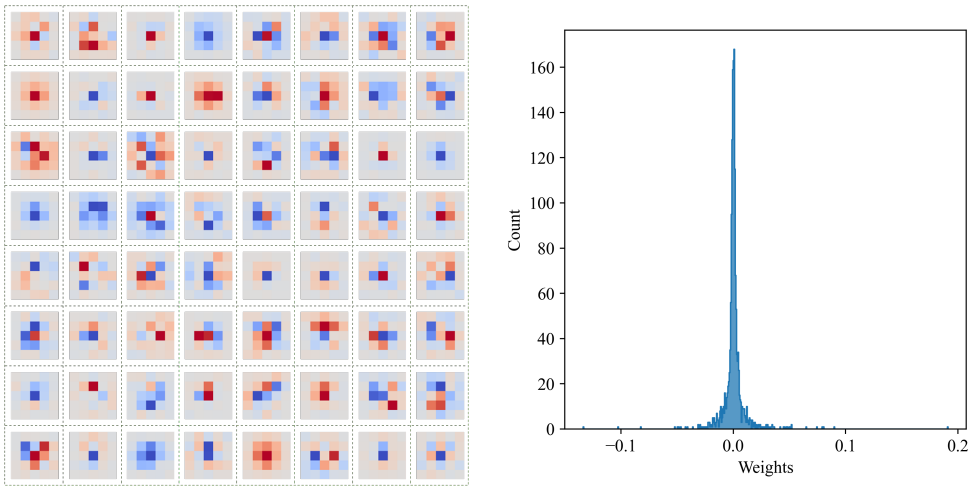
(a) (b)

Figure 14: Visualizations of (a) learned global dictionary filters ($\gamma = 0.1$) and (b) the distribution of their weights with the range of $[-0.06, 0.14]$.



(a) (b)

Figure 15: Visualizations of (a) learned global dictionary filters ($\gamma = 0.3$) and (b) the distribution of their weights with the range of $[-0.17, 0.23]$.



(a) (b)

Figure 16: Visualizations of (a) learned global dictionary filters ($\gamma = 0.5$) and (b) the distribution of their weights with the range of $[-0.13, 0.19]$.

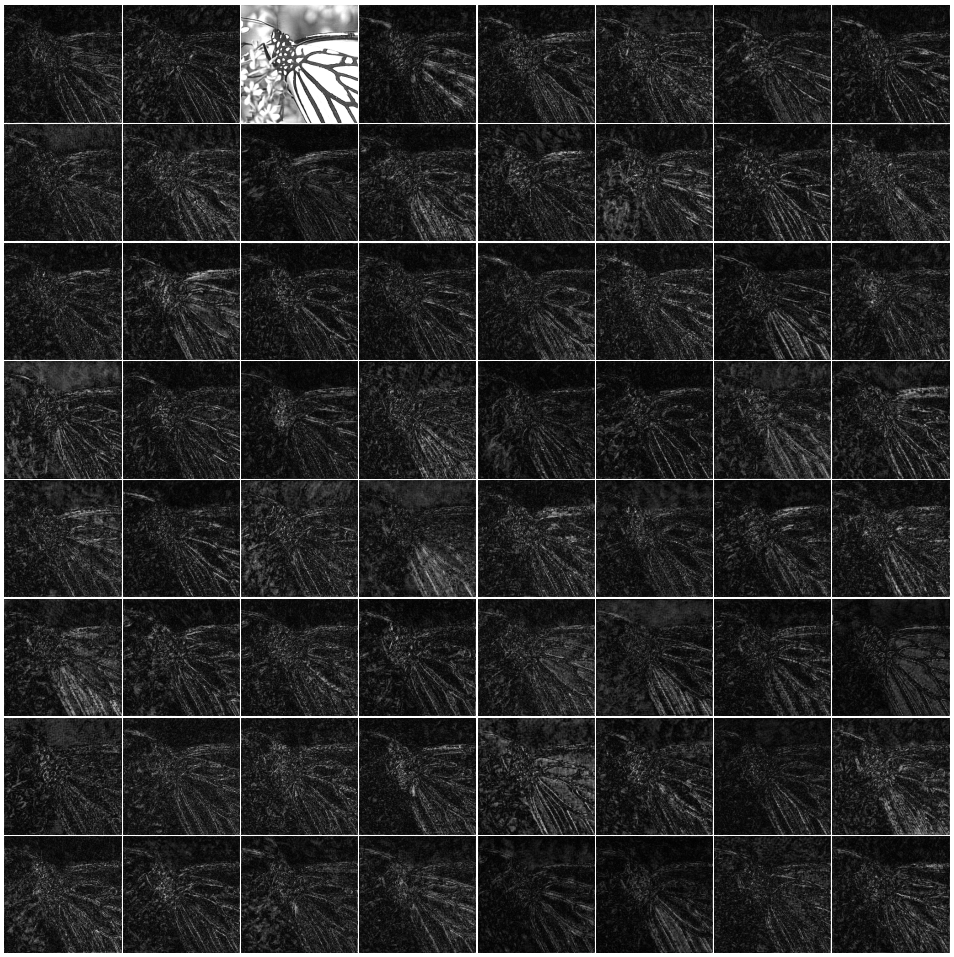


Figure 17: All feature maps $\boldsymbol{\alpha}_i$ of an image named "monarch" from Set11 (Kulkarni et al., 2016) with $\gamma = 10\%$.

Figure 18: All feature maps $\boldsymbol{\alpha}_i$ of an image named "cameraman" from Set11 (Kulkarni et al., 2016) with $\gamma = 30\%$.

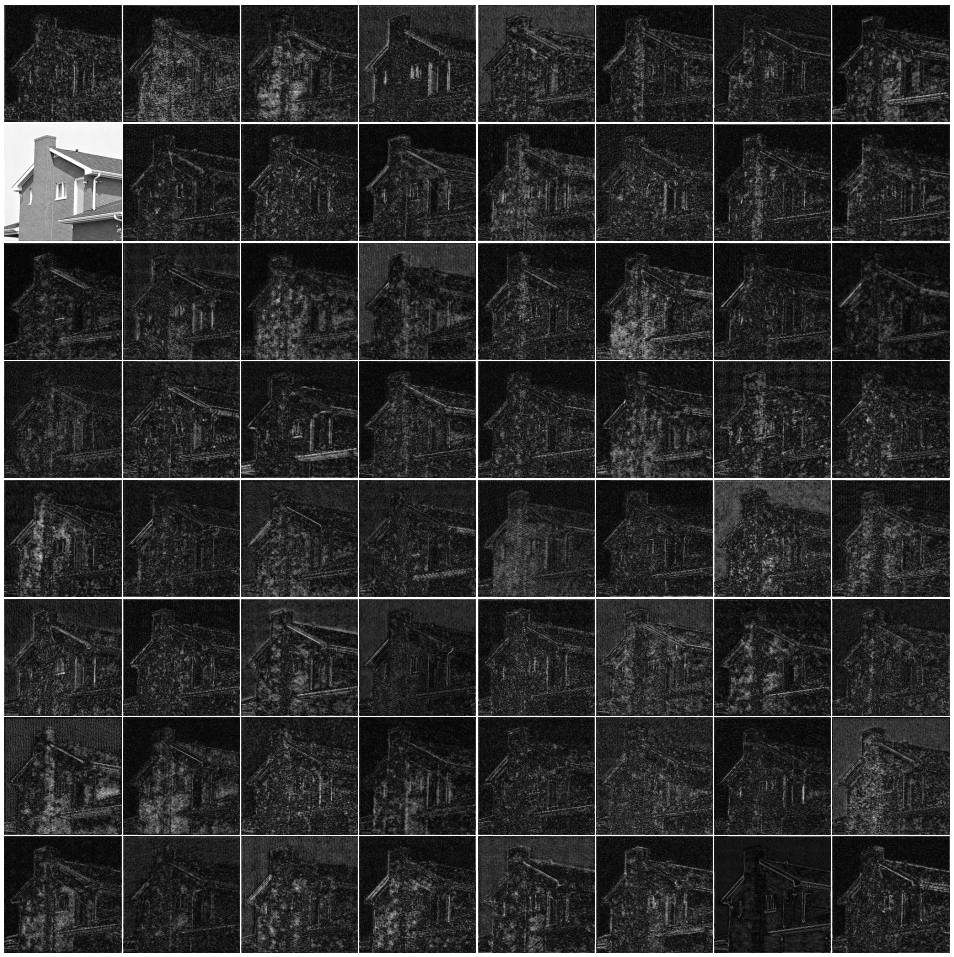


Figure 19: All feature maps $\boldsymbol{\alpha}_i$ of an image named "house" from Set11 (Kulkarni et al., 2016) with $\gamma = 50\%$.

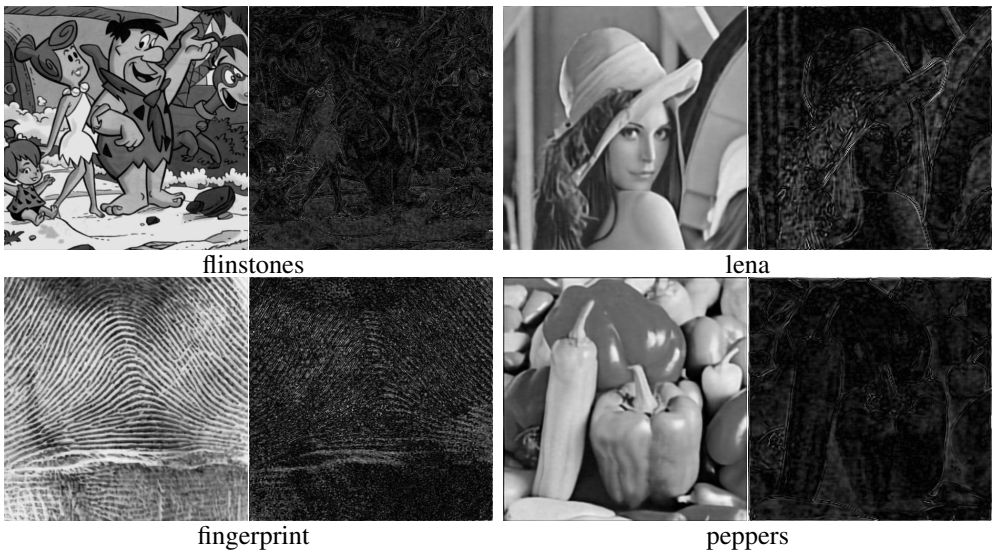


Figure 20: Low-frequency information $\boldsymbol{d}_2 * \boldsymbol{\alpha}_2$ (left) and the complementary high-frequency information $\sum_{i \neq 2} \boldsymbol{d}_i * \boldsymbol{\alpha}_i$ (right), with applying $\mathrm{D^3C^2}$-Net to four images from Set11 (Kulkarni et al., 2016) with $\gamma = 10\%$.

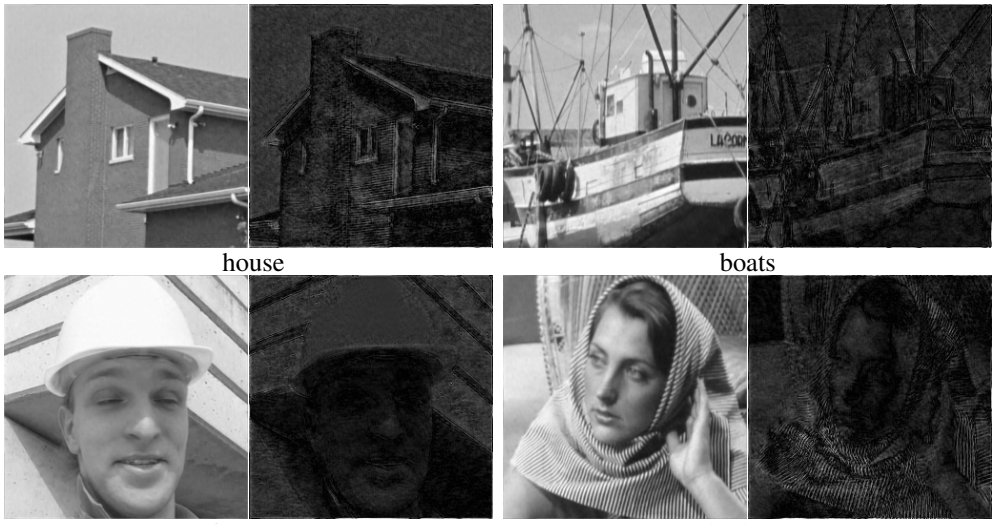


Figure 21: Low-frequency information $\boldsymbol{d}_{25} * \boldsymbol{\alpha}_{25}$ (left) and the complementary high-frequency information $\sum_{i \neq 25} \boldsymbol{d}_i * \boldsymbol{\alpha}_i$ (right), with applying $\mathrm{D^3C^2}$-Net to four images from Set11 (Kulkarni et al., 2016) with γ=30%.

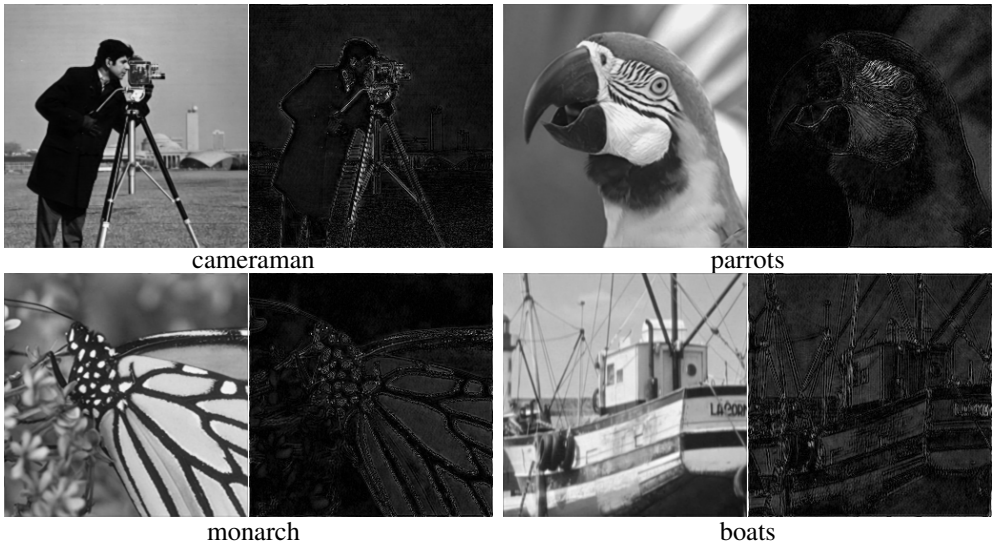


Figure 22: Low-frequency information $\boldsymbol{d}_8 * \boldsymbol{\alpha}_8$ (left) and the complementary high-frequency information $\sum_{i \neq 8} \boldsymbol{d}_i * \boldsymbol{\alpha}_i$ (right), with applying $\mathrm{D^3C^2}$-Net to four images from Set11 (Kulkarni et al., 2016) with $\gamma = 50\%$.