Proceedings of Machine Learning Research 101:694–709, 2019 ACML 2019

SPoD-Net: Fast Recovery of Microscopic Images Using Learned ISTA

Satoshi Hara* SATOHARA@AR.SANKEN.OSAKA-U.AC.JP
Osaka University, Japan

Weichih Chen*,† R05942094@NTU.EDU.TW
National Taiwan University, Taiwan

Takashi Washio WASHIO@AR.SANKEN.OSAKA-U.AC.JP
Tetsuichi Wazawa WZ8@SANKEN.OSAKA-U.AC.JP
Takeharu Nagai NG1@SANKEN.OSAKA-U.AC.JP
Osaka University, Japan

Editors: Wee Sun Lee and Taiji Suzuki

Abstract

Recovering high quality images from microscopic observations is an essential technology in biological imaging. Existing recovery methods require solving an optimization problem by using iterative algorithms, which are computationally expensive and time consuming. The focus of this study is to accelerate the image recovery by using deep neural networks (DNNs). In our approach, we first train a certain type of DNN by using some observations from microscopes, so that it can well approximate the image recovery process. The recovery of a new observation is then computed thorough a single forward propagation in the trained DNN. In this study, we specifically focus on observations obtained by SPoD (Super-resolution by Polarization Demodulation), a recently developed microscopic technique, and accelerate the image recovery for SPoD by using DNNs. To this end, we propose *SPoD-Net*, a specifically tailored DNN for fast recovery of SPoD images. Unlike general DNNs, SPoD-Net can be parameterized using a small number of parameters, which is helpful in two ways: (i) it can be stored in a small memory, and (ii) it can be trained efficiently. We also propose a method to stabilize the training of SPoD-Net. In the experiments with the real SPoD observations, we confirmed the effectiveness of SPoD-Net over existing recovery methods. Specifically, we observed that SPoD-Net could recover images with more than a hundred times faster than the existing method.

Keywords: sparse coding, convolutional sparse coding, learned iterative soft-thresholding algorithm, deep neural network

1. Introduction

Observing nanoscopic structures of live cells is a challenging topic in biological imaging (Sahl et al., 2017). Conventional super-resolution nanoscopy is not suitable for this purpose because the detailed inspection requires high-power illumination that is harmful to live cells.

∗ These authors contributed equally.
† This work was done when the author was staying at Osaka University as an exchange student.

To observe live cells, three key technologies have been developed in the last few years. The first technology is SPoD (Super-resolution by Polarization Demodulation), a microscopic technique which improved the resolution of observations by incorporating information of molecular orientation (Hafi et al., 2014). The second technology is the use of a positively-photoswitchable fluorescent protein, Kohinoor (Tiwari et al., 2015), that can color cells which enabled to observe cells with low-power illumination in SPoD (Wazawa et al., 2018). The third technology is the image recovery methods (Hafi et al., 2014; Wazawa et al., 2018) that can recover high quality images from the observations obtained by SPoD.

While the development of these technologies enabled us to observe cells in detail, the long computation time required for image recovery is found to be a bottleneck in practice. Hafi et al. (2014) and Wazawa et al. (2018) proposed to formulate the image recovery as an optimization problem, which is then solved by using Iterative Soft-Thresholding Algorithm (ISTA) (Daubechies et al., 2004) and Fast ISTA (FISTA) (Beck and Teboulle, 2009). These ISTA and FISTA require several tens of minutes to recover a single observation. This is because these are iterative algorithms that usually require thousands of iterations to recover an image. Because of the long computation time of these methods, the scientists need to wait for several tens of minutes every time after they obtained observations by SPoD. This problem is crucial especially when the scientists are interested in the dynamics of cells. In such a case, they record several hundreds of observations over time to trace the dynamics. With the current methods, the recovery of hundreds of observations are apparently computationally prohibitive. A fast recovery method is therefore in high demand to accelerate the entire process of the science.

To overcome the problem of computation time, here we propose *SPoD-Net*, a specifically tailored deep neural network (DNN) that enables a fast image recovery for SPoD. In our approach, we first train the SPoD-Net by using some observations from SPoD as a training set, so that the SPoD-Net can well approximate the recovery process of ISTA. The recovery of a new observation is then computed thorough a single forward propagation in the trained SPoD-Net, which usually takes less than one second. Thus, the image recovery using SPoD-Net is significantly faster than ISTA and FISTA.

The proposed SPoD-Net is based on the idea of Learned Iterative Soft-Thresholding Algorithm (LISTA) (Gregor and LeCun, 2010; Kavukcuoglu et al., 2010), which is a typical surrogate of ISTA. In LISTA, we express the recovery process of ISTA as a DNN with a fixed number of layers (e.g., DNN with ten layers). DNN is trained using a training set so that it can well approximate ISTA. We follow the same approach to construct the SPoD-Net for SPoD. The contribution of our study is in twofolds.

1. We propose SPoD-Net, a DNN specifically tailored for SPoD. Unlike general DNNs used in LISTA, SPoD-Net can be parameterized using a small number of parameters. This property is helpful in two ways: (i) it can be stored in a small memory, and (ii) it can be trained efficiently.

2. We propose a method to stabilize the training of SPoD-Net. We observed that the soft-thresholding operator in the network can zero out the signal during the forward propagation, which spoils the training of the network. In the proposed method, we use *leaky ReLU* (Maas et al., 2013) as an alternative of the soft-thresholding operator. Because leaky ReLU does not zero out signals, the training process gets stable.

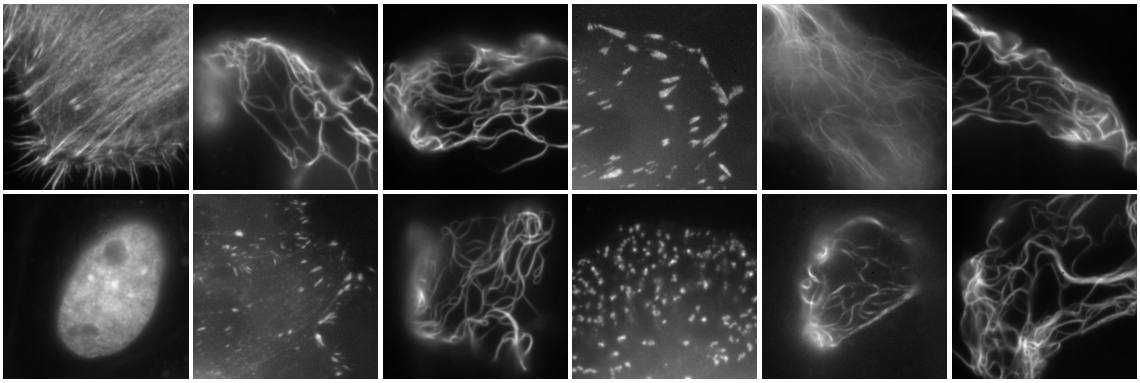


Figure 1: 12 observations obtained by SPoD. One observation consists of 18 images, where the average image is shown here. See Figure 2 for each of 18 images.

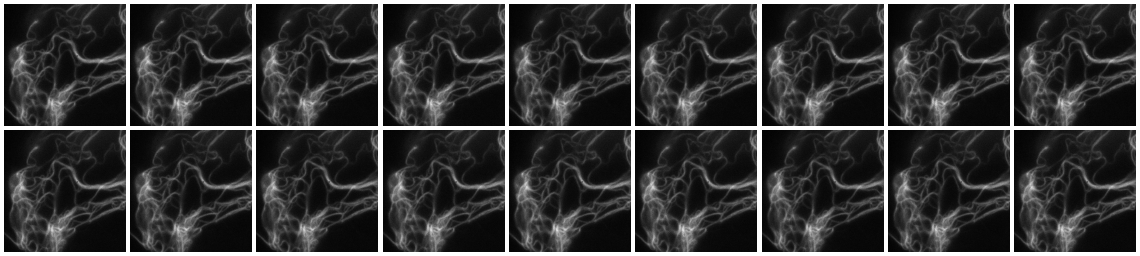


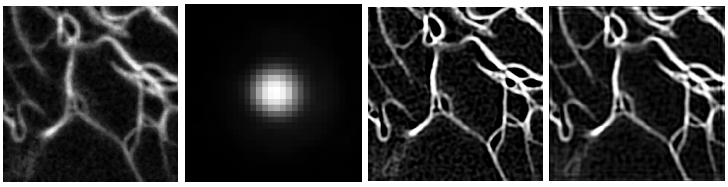
Figure 2: 18 images in the last observation in Figure 1. Although they may look similar, they are different images reflecting the change of polarization.

We demonstrate the effectiveness of SPoD-Net using observations obtained by SPoD. Specifically, we observed that SPoD-Net could recover images of comparable qualities as FISTA with only ten layers. This means that SPoD-Net was more than a hundred times faster than FISTA which requires thousands of iterations for recovery. With SPoD-Net, the scientists can obtain recovered images just after they obtained observations by SPoD.

SPoD Data In the paper, we use the twelve observations obtained by SPoD, which are shown in Figure 1. In SPoD, we observe cells with an illumination polarization changing over the time. Thus, one observation of SPoD consists of a set of images with different polarizations. Examples of images in one observation is shown in Figure 2. In the example, one observation consists of 18 images of size 512×512 with different polarizations. A high quality image is recovered from the observation (i.e. the set of images) by solving an optimization problem. Figure 3 shows an example of the recovered image.

Notations We denote the set of real values and positive integers by $\mathbb{R}$ and $\mathbb{N}_+$, respectively.

Let $x \in \mathbb{R}^{h \times w \times c}$ be an image with the height $h \in \mathbb{N}_+$, the width $w \in \mathbb{N}_+$, and the channel $c \in \mathbb{N}_+$. Here, each channel in x corresponds to an image with different polarization. For example, in Figure 2, the number of channels c is 18. Let $u \in \mathbb{R}^{a \times b \times c}$ be a filter with the same number of channels c as the image x. We denote the *convolution*, or *cross-correlation*, of the image x with the filter u by $u * x \in \mathbb{R}^{(h-a+1)\times(w-b+1)}$, where the (i, j)-th entry of



(*a*) Observation (*b*) PSF (*c*) FISTA (*d*) SPoD-Net

Figure 3: An example of the image recovery. (a) the first image out of the 18 images; (b) point spread function (PSF); (c) a recovered image using FISTA; (d) a recovered image using the proposed method SPoD-Net. See Section 2 and 3 for the details of the image recovery.

$u * x$ is given by

$$(u * x)_{i,j} := \sum_{i'=1}^{a} \sum_{j'=1}^{b} \sum_{k'=1}^{c} u_{i',j',k'} x_{i+i'-1,j+j'-1,k'}, \tag{1}$$

for $\forall i \in [1, h-a+1]$ and $\forall j \in [1, w-b+1]$. See Figure 4(a) for an example.

Let $y \in \mathbb{R}^{h \times w}$ be an image with the height $h \in \mathbb{N}_+$ and the width $w \in \mathbb{N}_+$. Let u be the same filter as above. We denote the *transposed convolution* of the image y with the filter u by $u \bar{*} y \in \mathbb{R}^{(h+a-1)\times(w+b-1)\times c}$, where the (i,j,k)-th entry of $u \bar{*} y$ is given by

$$(u \bar{*} y)_{i,j,k} := \sum_{i'=1}^{a} \sum_{j'=1}^{b} u_{a-i'+1,b-j'+1,k} y_{i-a+i',j-b+j'}, \tag{2}$$

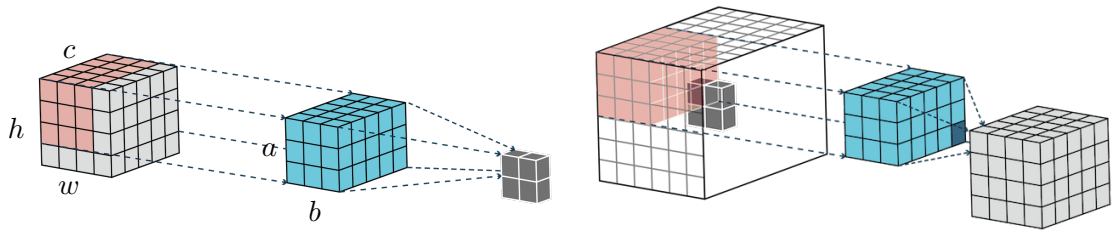
for $\forall i \in [1, h+a-1]$, $\forall j \in [1, w+b-1]$, and $\forall k \in [1, c]$. Here, we defined $y_{i,j} = 0$ if $i \notin [1,h]$ or $j \notin [1,w]$, which corresponds to zero padding of the image y. Note that, in (2), the filter u is flipped horizontally and vertically when taking a sum. See Figure 4(b) for an example.

We denote the thresholding of an image x by $[x]_+$ where $([x]_+)_{i,j,k} = \max\{x_{i,j,k}, 0\}$.

2. Image Recovery Problem for SPoD

Here, we review the image recovery problem for SPoD. SPoD microscopy (Hafi et al., 2014; Wazawa et al., 2018) enabled to observe cells with improved spatial resolution. SPoD achieved this goal by taking temporal information into account. While ordinary microscope obtain a single image as one observation, SPoD obtains a set of images over several different polarizations changing over the time. In this way, SPoD collects information necessary for microscopy using only low-power illumination. A high quality image is then recovered from the observation by solving an optimization problem. Popular image recovery methods for SPoD are ℓ_1 and ℓ_p regularization methods (Hafi et al., 2014; Wazawa et al., 2018).

The recovery problem for SPoD is formulated as follows. In SPoD, observed images are decayed through the point spread function (PSF). PSF expresses how a single dot is blurred in the observation. Note that, in SPoD, we can obtain the PSF by observing a



(a) Convolution (b) Transposed convolution

Figure 4: (a) Convolution of input shape $h \times w \times c = 4 \times 4 \times 5$ with filter shape $a \times b \times c = 3 \times 3 \times 5$ and output shape $(h-a+1) \times (w-b+1) \times 1 = 2 \times 2 \times 1$. (b) Transposed convolution of input shape $2 \times 2 \times 1$ with filter shape $3 \times 3 \times 5$ and output shape $4 \times 4 \times 5$. The white voxels correspond to zero padding around the input.

known dot pattern. See Figure 3 for an example of PSF. Let $\phi \in \mathbb{R}^{a \times b}$ be a PSF, and let $x \in \mathbb{R}^{h \times w \times c}$ be a *true* image we want to obtain. Moreover, let $y_t \in \mathbb{R}^{(h-a+1) \times (w-b+1)}$ be an observed image at time t, and let the entire observation $Y := \{y_1, y_2, \ldots, y_c\}$ be a set of the observed images. Then, by using convolution, the observed image y_t at time t is expressed as follows (Hafi et al., 2014; Wazawa et al., 2018):

$$y_t := a_t * x + \epsilon, \quad (a_t)_{i,j,k} := \phi_{ij} \cos^2\left(\frac{t-k}{c}\pi\right), \tag{3}$$

where ϵ is an observation noise.[1] Here, the term $\cos^2\left(\frac{t-k}{c}\pi\right)$ in a_t expresses the effect of the polarization changing over time. Note that, the true image x is sparse, i.e. most pixels are black, in general. This is because, in SPoD, we observe cells colored by photoswitchable fluorescent protein (Tiwari et al., 2015). The remaining cells that are not colored are observed as black backgrounds. By assuming the sparseness of x, the image recovery problem for (3) is formulated as the following optimization problem (Hafi et al., 2014; Wazawa et al., 2018).

$$\hat{x} = \underset{x \in \mathbb{R}^{w \times h \times c}}{\operatorname{argmin}} \underbrace{\sum_{t=1}^{c} r(y_t, a_t * x)}_{\text{reconstruction error}} + \underbrace{\Omega(x)}_{\text{sparsity inducing term}} \quad \text{subject to } x \geq 0. \tag{4}$$

The first term enforces the recovered image $\hat{x}$ to well represent the observations y_t. Specifically, the recovered image $\hat{x}$ should reconstruct the observation if it is decayed along the decaying process (3). The second term enforces the recovered image $\hat{x}$ to be sparse. Here, in (4), the element-wise non-negative constraint $x \geq 0$ is added because the value of each pixel should be zero (i.e. black) or larger. As the error term r, one can use the Poisson loss $r(y, z) := \sum_{i,j}(-y_{i,j} \log z_{i,j} + z_{i,j})$ as well as the squared loss $r(y, z) := 1/2 \sum_{i,j}(y_{i,j} - z_{i,j})^2$. As the sparsity inducing term Ω, one can use the ℓ_1 regularization $\Omega(x) := \lambda \sum_{i,j,k} |x_{i,j,k}|$ as well as the ℓ_p regularization $\Omega(x) := \lambda \sum_{i,j,k} |x_{i,j,k}|^p$ with $p \geq 1$, where $\lambda \geq 0$ is a trade-off parameter determined by the user balancing the reconstruction error and the sparsity. To

1. One can adopt generalized linear models instead of the additive noise model if needed.

solve the problem (4), optimization algorithms such as ISTA (Daubechies et al., 2004) and FISTA (FISTA) (Beck and Teboulle, 2009) are used (Hafi et al., 2014; Wazawa et al., 2018).

In this paper, for simplicity, we consider the problem (4) with the squared loss as r and the ℓ_1 regularization as Ω, as follows:

$$\hat{x} = \underset{x}{\operatorname{argmin}} \underbrace{\frac{1}{2A}\sum_{t=1}^{c} \|y_t - a_t * x\|^2 + \frac{\lambda}{B}\|x\|_1}_{L(Y,x)}, \text{ subject to } x \geq 0, \tag{5}$$

where $\|z\|^2 := \sum_{i,j} z_{i,j}^2$ is an Euclid norm. Here, we introduced the normalization constants $A := (h-a+1)(w-b+1)c$ and $B := hwc$ to balance the two terms. We denote the objective function by $L(Y,x)$. Note that the following discussions in this paper including the proposed method can be extended also to other loss functions and regularizations.

3. LISTA

Learned ISTA (LISTA) (Gregor and LeCun, 2010; Kavukcuoglu et al., 2010) is a general framework for approximating the optimization process of ISTA using DNN. Here, we briefly review ISTA and LISTA for the problem (5).

ISTA ISTA is a common algorithm for solving ℓ_1-regularized problems such as (5). With ISTA, we can solve the problem (5) by repeating the following two steps.

$$x \leftarrow x - \frac{\eta}{A}\sum_{t=1}^{c} a_t \bar{*} (a_t * x - y_t), \tag{6}$$

$$x \leftarrow R_{\lambda/B\eta}(x), \tag{7}$$

where $R_\theta(z) := [z-\theta]_+$ is a non-negative soft-thresholding operator. The first step (6) minimizes the first term of (5) using gradient descent with a step size $\eta > 0$. The second step (7) reflects the effect of the second term in (5), and shrinks x towards zeros while incorporating the non-negative constraint. Note that R_θ with $\theta = 0$ is a well-known ReLU function (Glorot et al., 2011). ISTA repeats these two steps until x converges.

LISTA LISTA approximates the repetition of the steps (6) and (7) in ISTA using DNN. To this end, LISTA interprets the iterations of ISTA as a feed forward network, as follows. Suppose x is initialized as zeros in ISTA. Then, the M iterations of ISTA is expressed as

$$\begin{aligned}
x^{(1)} &= R_{\theta^{(1)}}\left(\eta^{(1)}\sum_{t=1}^{c} p_t^{(1)} \bar{*}\, y_t\right), \\
x^{(2)} &= R_{\theta^{(2)}}\left(x^{(1)} - \eta^{(2)}\sum_{t=1}^{c} p_t^{(2)} \bar{*} \left(p_t^{(2)} * x^{(1)} - y_t\right)\right), \\
&\vdots \\
x^{(M)} &= R_{\theta^{(M)}}\left(x^{(M-1)} - \eta^{(M)}\sum_{t=1}^{c} p_t^{(M)} \bar{*} \left(p_t^{(M)} * x^{(M-1)} - y_t\right)\right),
\end{aligned}$$

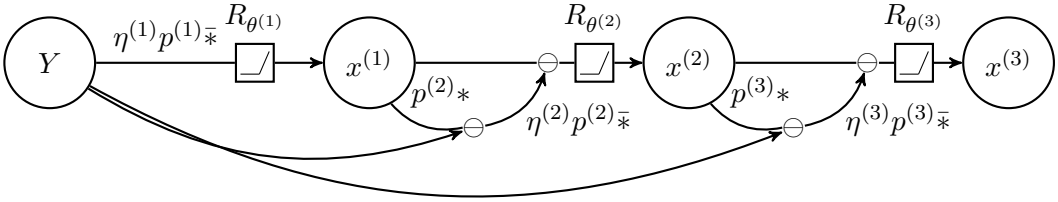


Figure 5: An example of the LISTA-Net for the problem (5). Three iterations of ISTA is expressed as a deep neural network.

where $p_t^{(1)} = p_t^{(2)} = \ldots = p_t^{(M)} = a_t$, $\eta^{(1)} = \eta^{(2)} = \ldots = \eta^{(M)} = \eta/A$, and $\theta^{(1)} = \theta^{(2)} = \ldots = \theta^{(M)} = \lambda/B\eta$. This can be interpreted a feed forward network as shown in Figure 5. The network accepts a set of images Y as its input, and outputs $x^{(M)}$ after M iterations of the ISTA update steps. We refer to this network as *LISTA-Net* hereafter.

LISTA approximates ISTA by training the LISTA-Net. While the parameters in each layer $(\{p_t^{(m)}\}_{t=1}^c, \eta^{(m)}, \theta^{(m)})$ are equivalent to the constants $(\{a_t\}_{t=1}^c, \eta/A, \lambda/B\eta)$ in ISTA, we treat them as *trainable parameters* in LISTA. By optimizing these parameters through training, LISTA-Net can approximate the optimization process of ISTA. Moreover, while ISTA usually requires more than a thousand iterations, LISTA-Net can approximate this exhaustive iteration with a limited number of layers, e.g. with the number of layers $M = 10$. Here, let $\Theta := \{(\{p_t^{(m)}\}_{t=1}^c, \eta^{(m)}, \theta^{(m)})\}_{m=1}^M$ be a set of parameters of the LISTA-Net, and $x = f(Y; \Theta)$ be the LISTA-Net parametrized by Θ. Suppose we have a set of observations $\{(Y_n, \hat{x}_n)\}_{n=1}^N$ where $\hat{x}_n$ is obtained by solving the problem (5). We can then train the LISTA-Net by optimizing the parameter Θ so that the following training loss to be minimized.

$$\hat{\Theta} = \operatorname*{argmin}_{\Theta} \frac{1}{N} \sum_{n=1}^{N} \|\hat{x}_n - f(Y_n; \Theta)\|^2. \tag{8}$$

In practice, we can use typical optimization algorithms such as mini-batch SGD and Adam (Kingma and Ba, 2014). Several studies have reported that LISTA-Net and its variations can approximate the recovered image $\hat{x}$ well with a few layers, while requiring significantly smaller computation (Gregor and LeCun, 2010; Kavukcuoglu et al., 2010; Borgerding and Schniter, 2016; Borgerding et al., 2017; Metzler et al., 2017; Sreter and Giryes, 2018; Chen et al., 2018).

LISTA without supervision One practical difficulty of LISTA is that its training requires a set of observations $\{(Y_n, \hat{x}_n)\}_{n=1}^N$. To prepare $\hat{x}_n$, we need to solve the problem (5) for each Y_n, e.g. by using ISTA. Because ISTA is computationally expensive, the collection of data is computationally highly demanding especially when we want to collect many data for training. Fortunately, there is a way to bypass this difficulty. Prior studies (Kavukcuoglu et al., 2010; Sreter and Giryes, 2018) have proposed a way to train LISTA-Net by using only observations $\{Y_n\}_{n=1}^N$ without $\hat{x}_n$. The idea here is that $\hat{x}_n$ is defined as the minimizer of (5). It is therefore sufficient if LISTA-Net can well approximate the minimizer of (5). We

can thus train the LISTA-Net so that it can minimize (5), as follows.

$$\hat{\Theta} = \operatorname*{argmin}_{\Theta} \frac{1}{N} \sum_{n=1}^{N} L(Y_n, f(Y_n; \Theta)). \tag{9}$$

Note that the training based on the loss function (9) does not require $\hat{x}_n$ as the supervised signal. Hence, we do not need to solve (5), and we can thus save the computation.

4. SPoD-Net: Image Recovery DNN for SPoD

In this section, we propose SPoD-Net, a specifically tailored DNN for the image recovery problem of SPoD (5). We then present the proposed network modification to stabilize the training as well as the layer-wise training algorithm.

4.1. SPoD-Net

We propose SPoD-Net, a specifically tailored DNN for SPoD. Recall that each layer of LISTA-Net has the set of filters $\{p_t^{(m)}\}_{t=1}^{c}$ each of which has abc parameters. Thus, the number of parameters in the set of filters is abc^2. SPoD-Net is a variant of LISTA-Net whose filter $p_t^{(m)}$ has a specific structure. In SPoD, as shown in (3), the filter a_t is of the form $(a_t)_{i,j,k} = \phi_{ij} \cos^2((t-k)\pi/c)$, where ϕ is the PSF. An important observation is that the spatial indices i and j appear only in the PSF ϕ, and the temporal index k appears only in the $\cos^2$. In SPoD-Net, we follow this observation and decompose the filter $p_t^{(m)}$ into the spatial part and the temporal part. Specifically, we consider the decomposition $(p_t^{(m)})_{i,j,k} := g_{i,j}^{(m)} u_{t,k}^{(m)}$ with the parameters $g^{(m)} \in \mathbb{R}^{a \times b}$ and $u_t^{(m)} \in \mathbb{R}^c$. Here, the first part $g_{i,j}^{(m)}$ represents the dependency to the spatial indices i and j, and the second part $u_{t,k}^{(m)}$ represents the dependency to the temporal index k. Moreover, recall that $\cos^2((t-k)\pi/c)$ is shift-invariant over t and k. Thus, the term $u_{t,k}^{(m)}$ should depend only on $t-k$. To implement the shift-invariance, we introduce the parameter $h^{(m)} \in \mathbb{R}^c$, and express $u_{t,k}^{(m)}$ by $u_{t,k}^{(m)} := h_{t-k}^{(m)}$ with a circular indexing $h_{-\ell}^{(m)} := h_{c-\ell+1}^{(m)}$ for $\ell \geq 1$.

With the proposed parametrization, in SPoD-Net, the convolution with the filter $p_t^{(m)}$ is expressed as

$$\begin{aligned}
(p_t^{(m)} * x)_{i,j} &= \sum_{i'=1}^{a} \sum_{j'=1}^{b} \sum_{k'=1}^{c} g_{i',j'}^{(m)} h_{t-k'}^{(m)} x_{i+i'-1,j+j'-1,k'} \\
&= \sum_{k'=1}^{c} h_{t-k'}^{(m)} \left(\sum_{i'=1}^{a} \sum_{j'=1}^{b} g_{i',j'}^{(m)} x_{i+i'-1,j+j'-1,k'} \right) \\
&= \sum_{k'=1}^{c} h_{t-k'}^{(m)} (g^{(m)} * x_{:,:,k'})_{i,j},
\end{aligned} \tag{10}$$

where $x_{:,:,k'}$ denotes the k'-th channel of x. Note that this is a specific variant of the *depth-wise separable convolution* (Kaiser et al., 2017; Chollet, 2017). Here, the term $g * x_{:,:,k'}$

with independent convolutions for each channel is known as *depth-wise convolution*. The term $\sum_{k'=1}^{c} h_{t-k'}^{(m)}(g * x_{:,:,k'})_{i,j}$ that applies the convolution with h over each index (i, j) is known as *point-wise convolution*.

In summary, in SPoD-Net, the filter $p_t^{(m)}$ is composed of the two basic parameters $g^{(m)} \in \mathbb{R}^{a\times b}$ and $h^{(m)} \in \mathbb{R}^c$. The number of parameters is thus $ab + c$. Compared to LISTA-Net with abc^2 parameters, SPoD-Net has a far less parameters: the number of parameters is nearly c^2 times smaller than LISTA-Net. This property of SPoD-Net is favorable in two ways. First, it can be saved in a small memory. Second, the training of the network requires less data.

4.2. Stabilizing Training by Network Modification

We propose a modification to SPoD-Net so that its training to be more stable. One difficulty when training LISTA-Net and SPoD-Net is that the operator R_θ can zero out signals during the forward propagation if the parameter of the network is badly conditioned. If the signals are zeroed out for all the training data, the gradients of all the parameters vanish during backpropagation. Thus, the training of the network completely stops once this phenomenon occurs. To avoid zeroing out signals, we need to tune the learning rate carefully so that the parameter does not fall into such an unfavorable condition.

To avoid the cumbersome tuning of the learning rate, we propose to modify the structure of the network. Recall that zeroing out occurs because of the operator R_θ where $R_\theta(z) := [z - \theta]_+$, i.e. if the signal z gets smaller than the threshold θ, the signal is zeroed out. Our idea is to use *leaky ReLU* (Maas et al., 2013) instead of the operator R_θ. More specifically, we replace the operator R_θ with the following *leaky soft-thresholding* operator $\hat{R}_{\theta,\alpha}$:

$$\hat{R}_{\theta,\alpha}(z_i) := \begin{cases} z_i - \theta & \text{if } z_i \geq \theta, \\ \alpha(z_i - \theta) & \text{otherwise,} \end{cases} \tag{11}$$

where $\alpha \geq 0$ is a level of relaxation determined by the user. While the signals smaller than the threshold are zeroed out in R_θ, the operator $\hat{R}_{\theta,\alpha}$ does not zero out small signals but allow them to be slightly negative with the factor α. Consequently, the gradients do not vanish during backpropagation with the operator $\hat{R}_{\theta,\alpha}$ as long as α is set to be strictly positive. Hence, with this modification, we can stabilize the training of the network.

Note that we achieve stable training by relaxing the non-negative constraint in the problem (5). Because the operator $\hat{R}_{\theta,\alpha}$ can output negative values, the output $x^{(M)}$ of the modified network may not be a feasible solution to the problem (5) anymore. To obtain a feasible output, we post-threshold the output to be non-negative by $\hat{x}^{(M)} = [x^{(M)}]_+$. The next theorem assures that this post-thresholding does not degenerate the quality of the recovered image so much. The proof of the theorem can be found in Appendix A.

Theorem 1 *For $\delta \geq 0$, let $x_\delta \geq -\delta$ and $\hat{x} := [x_\delta]_+$. Then, we have*

$$L(Y, \hat{x}) \leq \frac{1}{2A} \sum_{t=1}^{c} (\|y_t - a_t * x_\delta\| + \delta\kappa_t)^2 + \frac{\lambda}{B}\|x_\delta\|_1, \tag{12}$$

where $\kappa_t := \|a_t * 1_{h\times w\times c}\|$.

Theorem 1 indicates that the post-thresholding can change the quality of the recovered image only up to the factor $\delta\kappa_t$. Thus, as long as δ is kept small, the relaxation induced by $\hat{R}_{\theta,\alpha}$ is almost negligible. Overall, the use of the relaxed operator $\hat{R}_{\theta,\alpha}$ can stabilize the training with almost no lose in the quality of the recovered image.

4.3. Layer-wise Training

To train the proposed SPoD-Net, we use the layer-wise training. A typical approach for training DNN is to train the entire parameters by using the gradient-based methods. However, in the context of LISTA, several studies have reported that layer-wise training is much more effective (Borgerding and Schniter, 2016; Borgerding et al., 2017; Metzler et al., 2017; Chen et al., 2018). To train SPoD-Net, we follow the method proposed by Chen et al. (2018).

The layer-wise training algorithm is shown in Algorithm 1. In the algorithm, we sequentially add a new layer Θ_m and train it. The training of each layer Θ_m consists of the two steps, each of which requires an optimizer $\texttt{Opt}(\Theta; \{Y_n\}_{n=1}^N, \mathcal{L})$ that optimizes the parameter Θ so that the loss $\mathcal{L}$ on the set $\{Y_n\}_{n=1}^N$ to be small. The choice of the optimization algorithms such as SGD and Adam, as well as the size of the learning rate and the number of the optimization epochs are determined by the user. In the fist step, we train the parameter of the new layer Θ_m using the optimizer $\texttt{Opt}_1$. This aim of this step is to find a good initial parameter of the new layer for the second step. In the second step, we train all the parameters in the entire network Θ using the optimizer $\texttt{Opt}_2$. In this step, we refine the entire network so that it can output high quality images. After these steps, the parameters in the new layer are shrunk by the factor $\gamma \in (0, 1]$ before the next layer is added.

5. Experiment

We now present the effectiveness of the proposed SPoD-Net using the SPoD data.[2] Here, we evaluate the two aspects of SPoD-Net. In the first experiment, we demonstrate the effectiveness of the two proposed approaches which are (i) the use of the depth-wise separable convolution and (ii) the use of the leaky soft-thresholding. In the second experiment, we show that SPoD-Net can recover images more than a hundred times faster than FISTA.

Dataset Across the experiments, we used the twelve SPoD observations shown in Figure 1. Each observation consists of 18 images of size 512×512 and one PSF of size 32×32. Here, $c = 18$ corresponds to the number of channels in (3). All the observations are preprocessed by the median filter of size five to remove noisy dots in the observations, followed by the minimum filter of size seven to remove noisy backgrounds. In the experiment, we split twelve observations into four groups each of which consists of three observations. We used three of them (i.e. nine observations) as a training set for training SPoD-Net, and the remaining one (three observations) as a test set for evaluation. Thus, each experiment is conducted under four different data configurations depending on which group of observations is used as the test set. Because the PSFs of all the observations were similar, we used the average of twelve PSFs as ϕ in (3). Figure 3(b) shows the PSF we used in the experiments.

2. The codes are available at https://github.com/sato9hara/SPoD-Net

Algorithm 1 Layer-wise Training of SPoD-Net

Require: Training set $\{Y_n\}_{n=1}^N$ and optimizers $\texttt{Opt}_1$ and $\texttt{Opt}_2$
  $\Theta \leftarrow \emptyset$
  for $m = 1, 2, \ldots, M$ **do**
    `// add new layer`
    $\Theta_m \leftarrow \{\{g^{(m)}, h^{(m)}\}_{t=1}^c, \eta^{(m)}, \theta^{(m)}\}$
    Let $f_m(Y; \Theta \cup \Theta_m)$ be the SPoD-Net with m layers.
    Let the loss for $Y = \{y_1, y_2, \ldots, y_c\}$ be $\mathcal{L}_m := L(Y, f_m(Y; \Theta \cup \Theta_m))$.

    `// step1: train the new layer`
    $\Theta_m \leftarrow \texttt{Opt}_1(\Theta_m; \{Y_n\}_{n=1}^N, \mathcal{L}_m)$
    $\Theta \leftarrow \Theta \cup \Theta_m$

    `// step2: train the entire network`
    if $m \geq 2$ **then**
      $\Theta \leftarrow \texttt{Opt}_2(\Theta; \{Y_n\}_{n=1}^N, \mathcal{L}_m)$
    end if

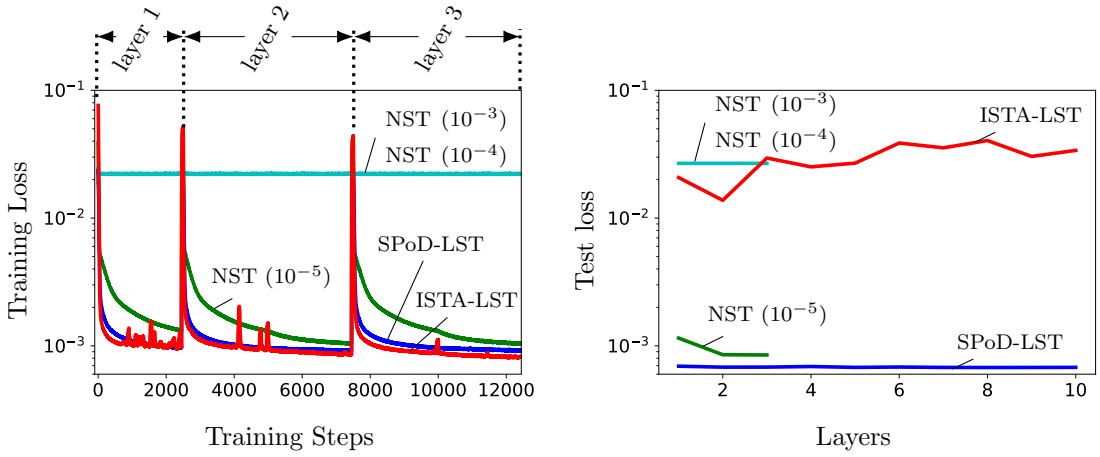
    `// shrink parameters`
    if $m \neq M$ **then**
      $\Theta_m \leftarrow \gamma \Theta_m$
    end if
  end for

5.1. Effectiveness of the Proposed Approaches

Here, we evaluate the effectiveness of the proposed approaches. Specifically, we evaluate the two key components of the proposed method, which are (i) the use of the depth-wise separable convolution and (ii) the use of the leaky soft-thresholding.

Setup We used SPoD-Net with $M = 10$ layers. For each data configuration, to train the SPoD-Net, we generated $N = 100$ observations from the nine observations as the training set by using data augmentation as follows. We first randomly selected one observation, and randomly clipped 96×96 images from it. We then randomly flipped the images horizontally or vertically. As a result, each observation $Y = \{y_1, y_2, \ldots, y_c\}$ in the training set consists of $c = 18$ images each of which has a size 96×96. We also applied the same data augmentation to the test set, and generated 12 test observations. We set $\lambda = 10^{-6}$ in the loss function (5).

Networks We refer to the proposed SPoD-Net with the leaky soft-thresholding (LST) as *SPoD-LST*, where we set α in the LST (11) to be 0.1. To evaluate the effectiveness of SPoD-LST, we prepared two baselines to be compared with. The first baseline, *ISTA-LST*, is the ordinary LISTA-Net presented in Section 3, with LST used instead of the non-negative soft-thresholding. As we have pointed out in Section 4, LISTA-Net has filters with the number of parameters abc^2 in each layer, while the SPoD-Net with the depth-wise separable convolution has only $ab + c$ parameters. We show the effectiveness of the depth-wise separable convolution by comparing SPoD-LST with ISTA-LST. The second baseline, *SPoD-NST*, is SPoD-Net with the non-negative soft-thresholding (NST) instead of the leaky



(*a*) Learning curves (*b*) Test losses

Figure 6: Comparisons of networks: (a) Learning curves of the training of the first three layers. (b) Test losses in each step of the training. The test loss is measured every time after the new layer is trained. NST denotes SPoD-NST, where the number in the parenthesis is the learning rate η of $\texttt{Opt}_1$. We stopped the training of SPoD-NST after three layers because of its instability in training.

soft-thresholding. We show the effectiveness of the leaky soft-thresholding by comparing SPoD-LST with SPoD-NST.

Training We used the layer-wise training in Algorithm 1 for SPoD-LST and the two baselines. In the algorithm, we used Adam (Kingma and Ba, 2014) as the optimizers $\texttt{Opt}_1$ and $\texttt{Opt}_2$. Here, we followed the setup proposed by Chen et al. (2018). For $\texttt{Opt}_1$, we set the learning rate η as $\eta = 10^{-4}$ for SPoD-LST and ISTA-LST, and $\eta = 10^{-3}, 10^{-4}$, and 10^{-5} for SPoD-NST. For $\texttt{Opt}_2$, we set the learning rate for the parameters in the new layer to be 0.2η, and 0.02η otherwise. We set the number of training epochs to be 500 for both $\texttt{Opt}_1$ and $\texttt{Opt}_2$. In the experiment, we set the batch size in one epoch to be 20, and thus the entire training step in each optimizer is 2,500. We set the shrinkage parameter γ to be 0.3.

Result Figures 6 (a) and (b) show the results. Here, we report one result out of the four data configurations because the results on other three cases were similar. There are two key observations in the figures regarding (i) the use of the depth-wise separable convolution and (ii) the use of the leaky soft-thresholding.

The first observation is the effectiveness of the depth-wise separable convolution. Figure 6 (a) shows that both SPoD-LST and ISTA-LST successfully minimized the training loss (9). Here, we observed that ISTA-LST attained slightly lower training loss than SPoD-LST. An important difference between these two networks can be found in Figure 6 (b). In the figure, we measured the loss of the recovered images on the test set using (9) every time after the new layer is trained. The figure shows the test loss of ISTA-LST was significantly larger than SPoD-LST. This means that ISTA-LST has overfitted to the training set and failed to generalize to the test set, while this was not the case for SPoD-LST. This difference

comes from the fact that ISTA-LST has abc^2 parameters in one layer, while SPoD-LST has only $ab + c$ parameters. Because ISTA-LST has many more trainable parameters, it can easily overfit to the training set. In particular, in the experiment, we only had limited number of observations, and it was therefore difficult for ISTA-LST to avoid overfitting. The use of the depth-wise separable convolution significantly reduced the number of trainable parameters, which enabled SPoD-LST to avoid overfitting caused by the limited number of observations. We also note that the size of the trained networks differed significantly: while the size of ISTA-LST was more than 13MB, the size of SPoD-LST was less than 50kB.

The second observation is the effectiveness of the leaky soft-thresholding. Figure 6 (a) shows that the training of SPoD-NST failed when $\eta = 10^{-3}$ and $\eta = 10^{-4}$. In these cases, the training losses were kept constant even if the number of training steps increases. We observed that this was because the signals were zeroed out, which made the gradients to be zeros and stopped the training. By setting the learning rate to be sufficiently small ($\eta = 10^{-5}$), we found that the signals were not zeroed out and the training did not stop. However, as shown in Figure 6 (a), it enforces the training to be slow. Compared to SPoD-LST that can use larger learning rate, the training loss of SPoD-NST decreased more slowly. This result clarified the disadvantage of the non-negative soft-threshold. To train SPoD-NST, we need to select appropriately small learning rate, which enforces the training to be slow. If we set too large learning rate, the training can stop in very early stages. By contrast, SPoD-LST that uses the leaky soft-thresholding does not require such careful tuning of the learning rate, which makes the training of the network faster and more stable.

5.2. Comparison with FISTA

We now demonstrate the advantage of SPoD-Net by comparing it with FISTA. Specifically, we show that SPoD-Net can recover images with comparable qualities as FISTA with more than a hundreds times faster than FISTA.

Setup We used SPoD-Net with $M = 10$ layers. We prepared the training data of $N = 100$ using the same data augmentation as in the previous section, except that the size of each image is set to 256×256 in this experiment. To train SPoD-Net, we used the same setup as in the previous section. For each of three observations in the test set, we split each image of size 512×512 into four patches of size 256×256 without overlap. In this way, we prepared 12 test observations. As the baseline, we adopted FISTA with the learning rate adjusted by a line search. Because FISTA does not require any training, we merely applied it to each of the test observations. To recover images, we ran FISTA for 10,000 steps.

Result Figure 7 shows the comparison of FISTA and SPoD-Net on their test losses measured by using (9). In the figure, we summarized the results on all the 48 test observations (12 observations for each of four data configurations). The figure clearly demonstrates the advantage of SPoD-Net over FISTA. It shows that, with SPoD-Net with the depth $M = 10$, we could recover images of almost the same quality as FISTA. The result indicates that SPoD-Net could recover images significantly faster than FISTA: SPoD-Net required only one forward propagation of the network of depth $M = 10$, while FISTA required more than a thousand iterations. That is, SPoD-Net could recover images with more than a hundred times faster than FISTA. Indeed, for the recovery of each test image, the 1,000 steps of

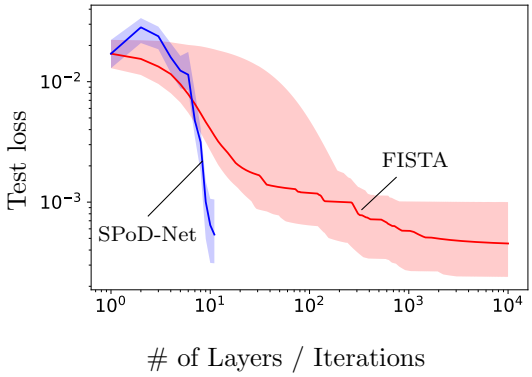
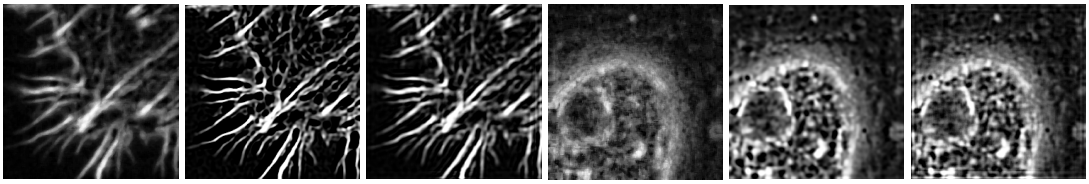


Figure 7: Comparison of FISTA and SPoD-Net on the test loss measured by using (9). The solid lines denote median losses, and shaded regions denote intervals between 25% and 75% quantiles.



(*a*) Original (*b*) FISTA (*c*) SPoD-Net (*d*) Original (*e*) FISTA (*f*) SPoD-Net

Figure 8: Comparisons of the recovered images using FISTA and SPoD-Net. The figures show the first image out of the 18 images in each observation.

FISTA took more than five minutes, while SPoD-Net took less than one second with a single GPU GTX1080ti In Figure 8 as well as in Figure 3, we show examples of the recovered images. In the figures, both FISTA and SPoD-Net could recover sharp images than the original blurred observations. Moreover, the recovered images obtained from SPoD-Net were visually comparable with FISTA. These results confirm the effectiveness of SPoD-Net that can recover high quality images with significantly less computation than FISTA.

6. Conclusion

In this paper, we proposed SPoD-Net, a specifically tailored DNN for SPoD, as a computationally efficient surrogate of ISTA for recovering high quality images with much shorter time. SPoD-Net is based on the idea of LISTA while incorporating two key improvements. The first improvement is the use of tbe depth-wise separable convolution, which is a natural choice for SPoD. With this improvement, we could parameterize the DNN by using very small number of parameters. The second improvement is the use of the leaky soft-thresholding. We observed that ordinary soft-thresholding can zero out signals in forward propagation, which can stop the training of DNN. With the leaky soft-thresholding, we can avoid signals

to be zeroed out, and can stabilize the training. In the experiments with the SPoD data, we observed that these two improvements have led to better image recovery. Moreover, we found that SPoD-Net could recover images with comparable quality as FISTA with more than a hundred times faster than FISTA.

Acknowledgments

This work was supported by JST CREST Grant Number JPMJCR15N3, JPMJCR1666, and JSPS KAKENHI Grant Number JP17K00305, Japan. A part of this work is the outcome of the study conducted in Artificial Intelligence Research Center, the Institute of Scientific and Industrial Research, Osaka University.

References

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

Mark Borgerding and Philip Schniter. Onsager-corrected deep learning for sparse linear inverse problems. In *2016 IEEE Global Conference on Signal and Information Processing*, pages 227–231, 2016.

Mark Borgerding, Philip Schniter, and Sundeep Rangan. AMP-inspired deep networks for sparse linear inverse problems. *IEEE Transactions on Signal Processing*, 65(16):4293–4308, 2017.

Xiaohan Chen, Jialin Liu, Zhangyang Wang, and Wotao Yin. Theoretical linear convergence of unfolded ista and its practical weights and thresholds. In *Advances in Neural Information Processing Systems*, pages 9061–9071, 2018.

François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1251–1258, 2017.

Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57(11):1413–1457, 2004.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011.

Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on Machine Learning*, pages 399–406, 2010.

Nour Hafi, Matthias Grunwald, Laura S Van Den Heuvel, Timo Aspelmeier, Jian-Hua Chen, Marta Zagrebelsky, Ole M Schütte, Claudia Steinem, Martin Korte, Axel Munk, et al. Fluorescence nanoscopy by polarization modulation and polarization angle narrowing. *Nature methods*, 11(5):579, 2014.

Lukasz Kaiser, Aidan N Gomez, and Francois Chollet. Depthwise separable convolutions for neural machine translation. *arXiv preprint arXiv:1706.03059*, 2017.

Koray Kavukcuoglu, Pierre Sermanet, Y-Lan Boureau, Karol Gregor, Michaël Mathieu, and Yann L Cun. Learning convolutional feature hierarchies for visual recognition. In *Advances in Neural Information Processing Systems*, pages 1090–1098, 2010.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the 30th International Conference on Machine Learning*, volume 30, page 3, 2013.

Chris Metzler, Ali Mousavi, and Richard Baraniuk. Learned D-AMP: Principled neural network based compressive image recovery. In *Advances in Neural Information Processing Systems*, pages 1772–1783, 2017.

Steffen J Sahl, Stefan W Hell, and Stefan Jakobs. Fluorescence nanoscopy in cell biology. *Nature reviews molecular cell biology*, 18(11):685–701, 2017.

Hillel Sreter and Raja Giryes. Learned convolutional sparse coding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2191–2195, 2018.

Dhermendra K Tiwari, Yoshiyuki Arai, Masahito Yamanaka, Tomoki Matsuda, Masakazu Agetsuma, Masahiro Nakano, Katsumasa Fujita, and Takeharu Nagai. A fast-and positively photoswitchable fluorescent protein for ultralow-laser-power resolft nanoscopy. *Nature methods*, 12(6):515, 2015.

Tetsuichi Wazawa, Yoshiyuki Arai, Yoshinobu Kawahara, Hiroki Takauchi, Takashi Washio, and Takeharu Nagai. Highly biocompatible super-resolution fluorescence imaging using the fast photoswitching fluorescent protein kohinoor and spod-expan with l p-regularized image reconstruction. *Microscopy*, 67(2):89–98, 2018.

Appendix A. Proof of Theorem 1

Proof We first bound the term $\|\hat{x}\|_1$. By definition, $\|\hat{x}\|_1 = \sum_{(x_\delta)_{i,j,k}\geq 0}(x_\delta)_{i,j,k} \leq \sum_{(x_\delta)_{i,j,k}\geq 0}(x_\delta)_{i,j,k} - \sum_{(x_\delta)_{i,j,k}<0}(x_\delta)_{i,j,k} = \|x_\delta\|_1$. Next, we bound the term $\|y_t - a_t * \hat{x}\|^2$. By using the triangle inequality, we obtain

$$\|y_t - a_t * \hat{x}\|^2 = \|(y_t - a_t * x_\delta) - a_t * (\hat{x} - x_\delta)\|^2 \leq (\|y_t - a_t * x_\delta\| + \|a_t * (\hat{x} - x_\delta)\|)^2 . \tag{13}$$

For the second term, we further obtain $\|a_t * (\hat{x} - x_\delta)\| \leq \|a_t * \delta 1_{h\times w\times c}\| = \delta\kappa_t$. ■