# Zero-Shot Reconstruction of Ocean Sound Speed Field Tensors: A Deep Plug-and-Play Approach

Siyuan Li,[1] Lei Cheng,[1] Xiao Fu,[2] and Jianlong Li[1]

[1] *College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China*

[2] *School of Electrical Engineering and Computer Science, Oregon State University, OR, United States*

(Dated: 19 June 2024)

Reconstructing a three-dimensional ocean sound speed field (SSF) from limited and noisy measurements presents an ill-posed and challenging inverse problem. Existing methods used a number of pre-specified priors (e.g., low-rank tensor and tensor neural network structures) to address this issue. However, the SSFs are often too complex to be accurately described by these pre-defined priors. While utilizing neural network-based priors trained on historical SSF data may be a viable workaround, acquiring SSF data remains a nontrivial task. This work starts with a key observation: Although natural images and SSFs admit fairly different characteristics, their denoising processes appear to share similar traits—as both remove random components from more structured signals. This observation allows us to incorporate *deep denoisers* trained using extensive natural images to realize *zero-shot* SSF reconstruction, without any extra training or network modifications. To implement this idea, an alternating direction method of multipliers (ADMM) algorithm using such a deep denoiser is proposed, which is reminiscent of the *plug-and-play* (PnP) scheme from medical imaging. Our PnP framework is tailored for SSF recovery such that the learned denoiser can be simultaneously used with other handcrafted SSF priors. Extensive numerical studies show that the new framework largely outperforms state-of-the-art baselines, especially under widely recognized challenging scenarios, e.g., when the SSF samples are taken as tensor fibers. The code is available at https://github.com/OceanSTARLab/DeepPnP.

## I. INTRODUCTION

Three-dimensional (3D) sound speed field (SSF) reconstruction is crucial in various ocean acoustic applications, including underwater target localization,[1] object tracking,[2] and acoustic communications.[3] From a signal processing perspective, this task poses significant challenges—as only limited and noisy measurements are available. For instance, when multiple conductivity, temperature, and depth (CTD) chains and/or pressure inverted echo sounders (PIES) are deployed over an area,[4,5] a significant portion of the SSF remains unobserved; see Fig. 1. Despite recent advancements in high-dimensional 3D data (e.g., SSF, radio map and hyperspectral image) reconstruction (see, e.g., Refs. [6–11]), existing methods still struggle to attain reasonable SSF reconstruction under challenging scenarios, e.g., when the target SSF exhibits complex dynamics.

To tackle the SSF reconstruction problem—which is an ill-posed inverse problem—it is essential to integrate a wealth of prior information into the reconstruction process. This often leads to reconstruction formulations incorporating multiple regularization terms.[12] These regularizers can either be *hand-crafted* or *data-driven*. Hand-crafted regularizers are based on relatively simple hypotheses of the structures of the SSFs (e.g., sparsity,[7] low rank,[8] and local smoothness[9]) that can be expressed using analytical functions. Hand-crafted priors make a lot of sense, as they are analytical summaries of the empirical observations. However, hand-crafted priors often struggle to provide detailed information on fine-grained sound speed variations. In contrast, the machine learning and vision literature recently advocated a class of new priors, namely, data-driven learned priors[13–17]. These priors make

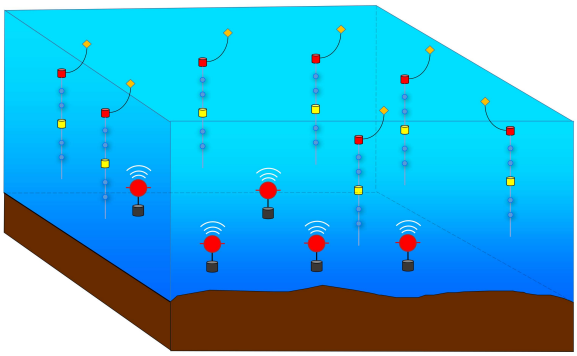


FIG. 1. A graphical illustration of multiple randomly deployed conductivity, temperature, and depth (CTD) chains and pressure inverted echo sounders (PIES), resulting in sparse and fiber-wise sampling.

use of information learned from data (normally through a carefully designed neural network). Compared to hand-crafted priors, data-driven priors are usually more flexible and adaptable in capturing detailed characteristics of data acquired under complex scenarios of interest.

Roughly speaking, in machine learning, there are two types of data-driven learned priors, i.e., supervised priors and unsupervised priors. Supervised data-driven priors utilize the knowledge acquired from training SSF data. The incorporation of (often a large amount of) training data enables such priors to provide a precise characterization of specific sound speed details.[6,7,18] Nonetheless, in the application of SSF reconstruction, obtaining high-quality training SSF data for prior training is a highly nontrivial task. Another caveat of using supervised priors lies in the difficulty of *generalization* under distribution drift. That is, when the testing data has different characteristics from the training data, using such learned priors may cause undesirable results (see, e.g., Fig. 14 in Ref. [6]).

Due to the difficulties of acquiring SSF training data, recent advances in SSF reconstruction primarily relied on unsupervised data-driven priors,[9,19,20] which are reminiscent of the deep image prior from computer vision.[17] This type of priors is not trained over data, but leverages neural structures to represent the intricate generating processes of complex data. If the neural structures are properly chosen, then such unsupervised/untrained priors can capture rich detail of the SSF data while using no training data—see Ref. [9] for an example using tensor neural network-based untrained prior to tackle the SSF reconstruction problem. Nonetheless, the challenge of using untrained deep prior is that the choices of neural structures seem to be overly abundant—and there has been no metric to measure the "optimality" of such choices. The untrained priors are also prone to overfit to noise,[17] which may require multiple heuristics (e.g., early stopping) to intervene.

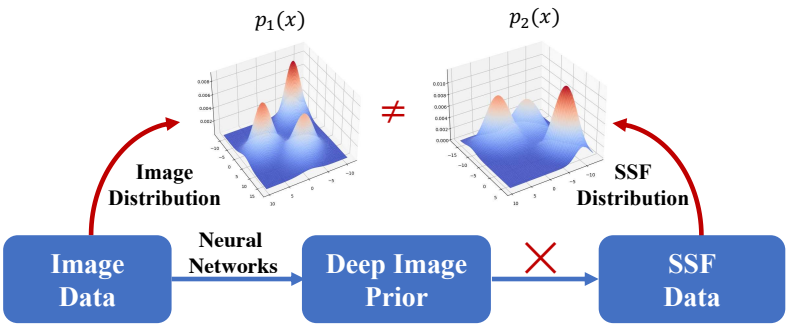
This work puts forth an alternative framework for deep prior-based SSF reconstruction. Instead of using trained or untrained priors as regularization terms in the SSF reconstruction criteria, we propose to employ neural denoisers that are trained over a massive amount of natural images in SSF recovery. The rationale is that although natural images and SSFs have fairly different characteristics/data distributions (see the illustration in Fig. 2(a)), the denoising process of both types of data share similar traits—both aim to remove relatively random components from more structured "signal" components. Fig. 2(b) presents the ocean SSF reconstruction results obtained using a state-of-the-art deep generative model (diffusion model) that has been pre-trained using natural images. The deviation between the reconstructed SSF and the ground-truth SSF showcases the limitations of employing pre-trained deep image priors for SSF reconstruction. The advantage of using the natural image

denoisers is that the training data can be easily acquired. The idea, illustrated in Fig. 3, is reminiscent of the deep plug-and-play (PnP) framework in computational imaging.[15,16,21,22]

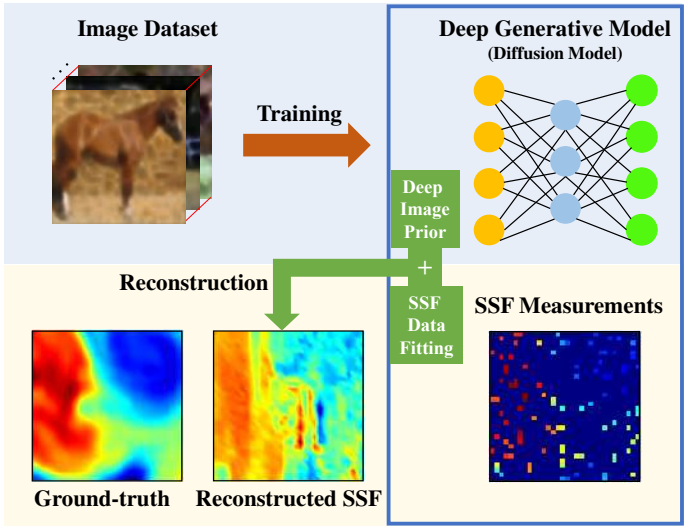
However, several challenges need to be addressed in order to adapt the PnP approach to 3D ocean SSF reconstruction. Specifically, deep denoisers are primarily designed for gray or color images, which typically have one or three channels. In contrast, 3D SSFs typically consist of multiple channels (or, equivalently, depths). To address this dimension mismatch issue, by noticing that the image denoiser can effectively remove the noise of the 2D SSFs at different depths (see Fig. 6 in Sec. IV), we unfold the 3D SSFs along one horizontal axis and denoise the unfolded matrix using a gray image denoiser. Furthermore, to preserve the spatial correlations across the three dimensions, we incorporate a tensor t-SVD-based low-rank prior into our approach. This prior promotes global coherency in the reconstructed SSFs while ensuring computational efficiency. Experimental results using real-life ocean SSFs demonstrate the superior performance of our method over state-of-the-art (SOTA) methods, particularly in realistic settings, e.g., fiber-wise sampling scenarios.

The remainder of this paper is organized as follows. In Sec. II, we formulate the reconstruction problem and show its connection with the denoising problem. In Sec. III, we propose to plug the pre-trained image denoiser into the reconstruction process and propose a zero-shot PnP deep tensor-based reconstruction algorithm. Experiments using real-life SSF data are reported in Sec. IV, followed by the conclusions in Sec. V.

**Notations:** Lower- and uppercase bold letters (e.g., $\mathbf{x}$ and $\mathbf{X}$) are used to denote the vectors and matrices. Higher-order tensors (order three or higher) are denoted by upper-case bold calligraphic letters. The tensor $t$-product is denoted by $*$. The Kronecker product is




(a)




(b)

FIG. 2. (a) Illustration of the unsuitability of pre-trained deep image priors trained on natural images for ocean sound speed field (SSF) reconstruction, which highlights the evident differences in data distributions between natural images and SSF data. (b) Illustration of the ocean SSF reconstruction results obtained using a state-of-the-art deep generative model (diffusion model) that has been pre-trained using natural images. The deviation between the reconstructed SSF and the ground-truth SSF showcases the limitations of employing pre-trained deep image priors for SSF reconstruction.

denoted by $\otimes$. The Hadamard product is denoted by $\circledast$. $\|\cdot\|_{\mathrm{F}}$ stands for the Frobenius

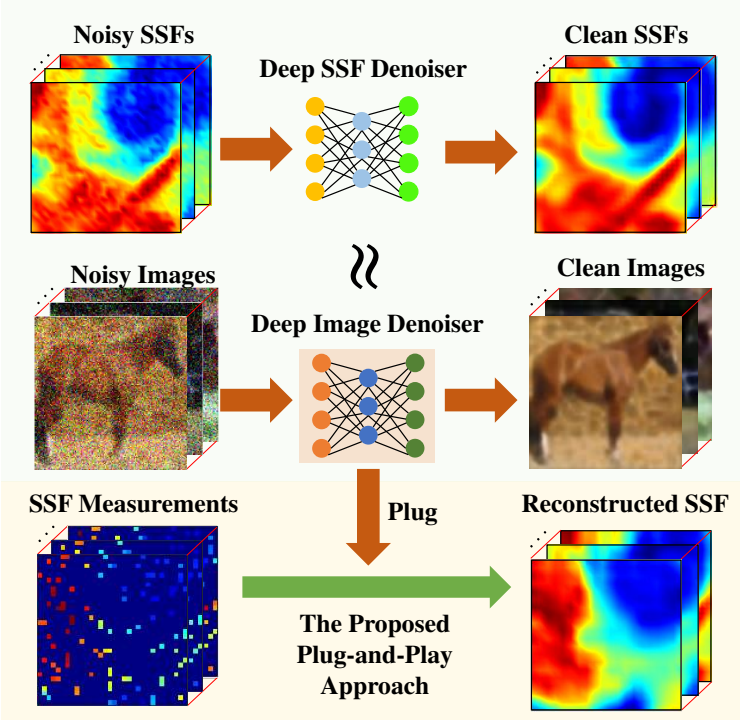



FIG. 3. This illustration depicts the rationale behind the proposed approach, which utilizes a pre-trained deep image denoiser to realize zero-shot ocean SSF reconstruction. The adopted pre-trained denoiser is capable of capturing shared characteristics between natural images and ocean sound speed fields that are distinct from noise, including explicit features like continuous variations, as well as implicit deep features. By leveraging this powerful data-driven regularizer, the accuracy of the reconstructed SSF can be significantly enhanced.

---

norm. $\|\cdot\|_*$ stands for the matrix nuclear norm and $\|\cdot\|_{\mathrm{T}*}$ stands for the tensor nuclear norm. $\langle\cdot,\cdot\rangle$ denotes the tensor inner product. $\boldsymbol{\mathcal{X}}^T$ denotes the transpose of $\boldsymbol{\mathcal{X}}$. $\mathbb{R}$ is the field of real numbers. The folding and unfolding operations are denoted as $\mathrm{Fold}(\mathbf{X})$ and $\mathrm{Unfold}(\boldsymbol{\mathcal{X}})$, respectively.

## II. RECONSTRUCTION AND CONNECTIONS TO DENOISING

This section presents the mathematical formulation of the 3D SSF reconstruction problem, followed by a discussion of its connection with the denoising problem.

### A. Reconstruction Problem Formulation

The 3D SSF reconstruction problem aims to estimate the unknown 3D SSF from the limited and noisy measurements. Specifically, we seek to solve the optimization problem:

$$\min_{\boldsymbol{\mathcal{X}}} \|\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{O}} \circledast \boldsymbol{\mathcal{X}}\|_{\mathrm{F}}^2, \tag{1}$$

where $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I\times J\times K}$ is the optimization variable (or SSF to be reconstructed) and $\boldsymbol{\mathcal{O}}$ is a binary indicator tensor with $\boldsymbol{\mathcal{O}}_{i,j,k} = 1$ if the corresponding entry $(i, j, k)$ is observed. Here $I$ and $J$ refer to the dimension size of the data in the horizontal direction, while $K$ represents the depth number. Symbol $\circledast$ denotes the Hadamard product. The observation tensor $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I\times J\times K}$ collects noisy and limited SSF samples, i.e., $\boldsymbol{\mathcal{Y}}_{i,j,k}$ is the sampled sound speed value if $\boldsymbol{\mathcal{O}}_{i,j,k} = 1$, and otherwise equals to zero. The Frobenius norm $\|\cdot\|_{\mathrm{F}}$ is utilized to quantify the error between the reconstructed and the observed data. Given the measurements $\boldsymbol{\mathcal{Y}}$, we aim to reconstruct the ground-truth SSF $\boldsymbol{\mathcal{X}}$.

Since problem (1) is an ill-posed inverse problem due to the limited measurements, prior information should be incorporated by adding multiple regularization terms to the objective function. Our first postulate is that the reconstructed SSF tensor should have a low tensor rank. Hence, we use a regularization term that is based on the tensor nuclear norm,[23] denoted as $\|\mathcal{X}\|_{\mathrm{T}*}$, to promote this property. The low tensor rank modeling of SSFs makes

sense, as there are clear correlations along different modes of the SSF tensor. The low tensor rank property of SSFs was observed in recent papers[6,7], but in this work we use a different low tensor rank promoter that is based on the so-called t-SVD[23] for computational efficiency—as we will see later.

The low tensor rank model is often considered a more "global" prior that describes the overall correlations—it does not capture detailed local variations of the SSF data. Therefore, we also include a data-driven regularizer that characterizes the fine-grained details of the SSF data. We denote this regularizer as $\Phi(\boldsymbol{\mathcal{X}})$ and will explain its form in the next section.

With these two regularization terms, the reconstruction problem can be reformulated as

$$\min_{\boldsymbol{\mathcal{X}}} \|\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{O}} \circledast \boldsymbol{\mathcal{X}}\|_{\mathrm{F}}^2 + \lambda_1 \|\boldsymbol{\mathcal{X}}\|_{\mathrm{T}*} + \lambda_2 \Phi(\boldsymbol{\mathcal{X}}), \tag{2}$$

where $\lambda_1$ and $\lambda_2$ are the regularization parameters that control the balance between the data fidelity term $\|\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{O}} \circledast \boldsymbol{\mathcal{X}}\|_{\mathrm{F}}^2$ and two regularization terms. Tensor nuclear norm $\|\boldsymbol{\mathcal{X}}\|_{\mathrm{T}*}$ is defined as $\sum_{k=1}^{K} \|\widetilde{\boldsymbol{\mathcal{X}}}(:,:,k)\|_*$, where $\widetilde{\boldsymbol{\mathcal{X}}}$ is the Fourier transformation of $\boldsymbol{\mathcal{X}}$ along the third mode.[23]

### B. Reconstruction as A Series of Denoising

At first glance, the reconstruction problem defined in Eq. (2) seems unrelated to the denoising problem. However, by implementing the classical ADMM optimization framework, the reconstruction problem can be transformed into a series of denoising problems.[21]

Specifically, following the standard ADMM procedure, two auxiliary variables are introduced, denoted by $\{\boldsymbol{\mathcal{Z}}, \boldsymbol{\mathcal{W}}\}$, and the reconstruction problem is recast as follows:

$$\begin{aligned} \min_{\boldsymbol{\mathcal{X}},\boldsymbol{\mathcal{Z}},\boldsymbol{\mathcal{W}}} & \|\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{O}} \circledast \boldsymbol{\mathcal{X}}\|_{\mathrm{F}}^2 + \lambda_1 \|\boldsymbol{\mathcal{W}}\|_{\mathrm{T}*} + \lambda_2 \Phi(\boldsymbol{\mathcal{Z}}), \\ & \text{s.t.} \quad \boldsymbol{\mathcal{W}} = \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Z}} = \boldsymbol{\mathcal{X}}. \end{aligned} \tag{3}$$

The augmented Lagrangian for this problem is then expressed as:

$$\begin{aligned} & \|\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{O}} \circledast \boldsymbol{\mathcal{X}}\|_{\mathrm{F}}^2 + \lambda_1 \|\boldsymbol{\mathcal{W}}\|_{\mathrm{T}*} + \langle \boldsymbol{\mathcal{E}}_1, \boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{W}} \rangle + \\ & \frac{\beta}{2} \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{W}}\|_{\mathrm{F}}^2 + \lambda_2 \Phi(\boldsymbol{\mathcal{Z}}) + \langle \boldsymbol{\mathcal{E}}_2, \boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{Z}} \rangle + \frac{\beta}{2} \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{Z}}\|_{\mathrm{F}}^2, \end{aligned} \tag{4}$$

where $\boldsymbol{\mathcal{E}}_1$ and $\boldsymbol{\mathcal{E}}_2$ denote the Lagrangian multipliers, and $\beta$ is a nonnegative penalty parameter.

Based on Eq. (4), problem (3) can be solved by tackling the following sequence of subproblems.

■ **Updating $\boldsymbol{\mathcal{W}}$:** First, we tackle the subproblem regarding $\boldsymbol{\mathcal{W}}$. The subproblem for $\boldsymbol{\mathcal{W}}$ can be obtained by minimizing the augmented Lagrangian in Eq. (4) with other variables fixed, and can be formulated as:

$$\min_{\boldsymbol{\mathcal{W}}} \lambda_1 \|\boldsymbol{\mathcal{W}}\|_{\mathrm{T}*} + \langle \boldsymbol{\mathcal{E}}_1^l, \boldsymbol{\mathcal{X}}^l - \boldsymbol{\mathcal{W}} \rangle + \frac{\beta^l}{2} \|\boldsymbol{\mathcal{X}}^l - \boldsymbol{\mathcal{W}}\|_{\mathrm{F}}^2, \tag{5}$$

where the superscript $l$ represents the iteration number. Using basic algebra, this problem can be equivalently simplified as:

$$\min_{\boldsymbol{\mathcal{W}}} \|\boldsymbol{\mathcal{W}}\|_{\mathrm{T}*} + \frac{\beta^l}{2\lambda_1} \left\| \boldsymbol{\mathcal{W}} - \left( \boldsymbol{\mathcal{X}}^l + \frac{\boldsymbol{\mathcal{E}}_1^l}{\beta^l} \right) \right\|_{\mathrm{F}}^2. \tag{6}$$

Note that problem (6) can be interpreted as a regularized additive white Gaussian noise (AWGN) denoising problem by treating the term $\boldsymbol{\mathcal{X}}^l + \frac{\boldsymbol{\mathcal{E}}_1^l}{\beta^l}$ as noisy observations and $\boldsymbol{\mathcal{W}}$ as

the signal to recovery, see more details in Appendix A. The regularization term is modeled by the tensor nuclear norm (i.e., $\|\boldsymbol{\mathcal{W}}\|_{\mathrm{T}*}$) to encode the low-rank of $\boldsymbol{\mathcal{W}}$.

Denote the denoiser based on tensor nuclear norm as $\boldsymbol{\mathcal{D}}_{\mathrm{T}*}(\cdot)$. Then, the solution to the denoising problem defined in (6) can be represented as $\boldsymbol{\mathcal{D}}_{\mathrm{T}*}(\boldsymbol{\mathcal{X}}^l + \frac{\boldsymbol{\mathcal{E}}_1^l}{\beta^l})$.

■ **Updating $\boldsymbol{\mathcal{Z}}$:** Similarly, we can express the $\boldsymbol{\mathcal{Z}}$-subproblem as follows:

$$\min_{\boldsymbol{\mathcal{Z}}} \Phi(\boldsymbol{\mathcal{Z}}) + \frac{\beta^l}{2\lambda_2} \left\| \boldsymbol{\mathcal{X}}^l - \boldsymbol{\mathcal{Z}} + \frac{\boldsymbol{\mathcal{E}}_2^l}{\beta^l} \right\|_{\mathrm{F}}^2. \tag{7}$$

This problem can also be viewed as an AWGN denoising problem. The prior information utilized in this step is captured by the regularizer function $\Phi(\boldsymbol{\mathcal{Z}})$. We can represent the solution to problem (7) as $\boldsymbol{\mathcal{D}}_{\Phi}(\boldsymbol{\mathcal{X}}^l + \frac{\boldsymbol{\mathcal{E}}_2^l}{\beta^l})$, where $\boldsymbol{\mathcal{D}}_{\Phi}(\cdot)$ denotes the denoiser based on $\Phi(\boldsymbol{\mathcal{Z}})$.

Additionally, it is worth noting that (7) is a *non-blind* denoising problem, of which the objective is to recover $\boldsymbol{\mathcal{Z}}$ from noisy observations $\boldsymbol{\mathcal{X}}^l + \frac{\boldsymbol{\mathcal{E}}_2^l}{\beta^l}$ under AWGN with power $\frac{\lambda_2}{\beta^l}$. Therefore, the assumed noise power adapts to the value of $\lambda_2$. However, the assumed noise power may differ from the ground-truth noise power, leading to a modeling mismatch in problem (7). This type of mismatch cannot be eliminated, but empirically it does not significantly affect the results, see the experimental results in Sec. IV.

■ **Updating $\boldsymbol{\mathcal{X}}$:** The $\boldsymbol{\mathcal{X}}$-subproblem is formulated as:

$$\begin{aligned}\min_{\boldsymbol{\mathcal{X}}} \|\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{O}} \circledast \boldsymbol{\mathcal{X}}\|_{\mathrm{F}}^2 + \langle \boldsymbol{\mathcal{E}}_1^l, \boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{W}}^{l+1} \rangle + \langle \boldsymbol{\mathcal{E}}_2^l, \boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{Z}}^{l+1} \rangle \\ + \frac{\beta^l}{2} \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{W}}^{l+1}\|_{\mathrm{F}}^2 + \frac{\beta^l}{2} \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{Z}}^{l+1}\|_{\mathrm{F}}^2,\end{aligned} \tag{8}$$

which is a least square problem and has a closed-form solution.

■ **Updating Lagrangian Multipliers:** Finally, the multipliers $\boldsymbol{\mathcal{E}}_1$ and $\boldsymbol{\mathcal{E}}_2$ are updated as follows:

$$\begin{cases} \boldsymbol{\mathcal{E}}_1^{l+1} = \boldsymbol{\mathcal{E}}_1^{l} + \beta^l(\boldsymbol{\mathcal{X}}^{l+1} - \boldsymbol{\mathcal{W}}^{l+1}), \\ \boldsymbol{\mathcal{E}}_2^{l+1} = \boldsymbol{\mathcal{E}}_2^{l} + \beta^l(\boldsymbol{\mathcal{X}}^{l+1} - \boldsymbol{\mathcal{Z}}^{l+1}), \end{cases} \tag{9}$$

and the penalty parameter $\beta$ is updated by

$$\beta^{l+1} = t\beta^l, \tag{10}$$

where $t$ is a hyper-parameter, and its value can be determined based on the difficulty of solving the subproblems in each iteration.[24] In our experiments, we have empirically determined that $t \in (1.0, 1.5)$ is appropriate. The rationale is twofold: 1) Choosing $t > 1$ ensures that $\beta$ increases, driving the iterates towards feasibility. 2) Keeping $t < 1.5$ prevents excessively rapid increases that could lead to convergence issues or oscillatory behavior.

The ADMM algorithm for solving the SSF reconstruction problem is summarized in **Algorithm 1**, highlighting the fact that the SSF reconstruction problem can be viewed as a sequence of denoising problems based on the aforementioned procedures.

## III. PLUG-AND-PLAY APPROACH

In the previous section, the SSF reconstruction problem was formulated and connected with a series of denoising problems. Deriving the ADMM algorithm is a standard practice, but how to choose problem-tailored regularization terms and how to solve the denoising steps are often an art. In the following subsections, the mathematical forms of the denoisers $\boldsymbol{\mathcal{D}}_{\mathrm{T}*}(\cdot)$ and $\boldsymbol{\mathcal{D}}_{\Phi}(\cdot)$ used in the denoising steps are specified.

**Algorithm 1: ADMM Algorithm for 3D Ocean SSF Reconstruction**

**Input:** $\boldsymbol{\mathcal{Y}}$, $\boldsymbol{\mathcal{O}}$, maximum iteration number $L$.

**Initialize:** $l \leftarrow 0$, $\boldsymbol{\mathcal{X}}$, $\boldsymbol{\mathcal{W}}$, $\boldsymbol{\mathcal{Z}}$, $\boldsymbol{\mathcal{E}}_1$, $\boldsymbol{\mathcal{E}}_2$, $\beta^0, \lambda_1, \lambda_2$.

**While** $l < L$ or not converged **do**

$\boldsymbol{\mathcal{W}}^{l+1} \leftarrow \boldsymbol{\mathcal{D}}_{\mathrm{T}*}(\boldsymbol{\mathcal{X}}^l + \frac{\boldsymbol{\mathcal{E}}_1^l}{\beta^l})$, [Denoising Step]

$\boldsymbol{\mathcal{Z}}^{l+1} \leftarrow \boldsymbol{\mathcal{D}}_{\Phi}(\boldsymbol{\mathcal{X}}^l + \frac{\boldsymbol{\mathcal{E}}_2^l}{\beta^l})$, [Denoising Step]

Update $\boldsymbol{\mathcal{X}}$ by solving (8),

Update $\boldsymbol{\mathcal{E}}$ via (9),

Update $\beta$ via (10),

$l \leftarrow l + 1$,

**end while**

**Return** $\boldsymbol{\mathcal{X}}^l$.

### A. Closed-form Tensor Nuclear Norm Denoiser

As we mentioned, using the tensor nuclear norm $\|\cdot\|_{\mathrm{T}*}$ to promote low tensor rank leads to computationally efficient updates. To see this, note that Ref. [23] showed that the denoising problem with respect to the tensor nuclear norm has a closed-form solution. Specifically, let the t-SVD of tensor $\boldsymbol{\mathcal{X}}^l + \frac{\boldsymbol{\mathcal{E}}_1^l}{\beta^l}$ be expressed as

$$\boldsymbol{\mathcal{X}}^l + \frac{\boldsymbol{\mathcal{E}}_1^l}{\beta^l} = \boldsymbol{\mathcal{U}} * \boldsymbol{\mathcal{S}} * \boldsymbol{\mathcal{V}}^T. \tag{11}$$

Then, the solution to the denoising problem can be expressed as $\boldsymbol{\mathcal{W}}^{l+1} = \boldsymbol{\mathcal{U}} * \mathrm{SVT}(\boldsymbol{\mathcal{S}}) * \boldsymbol{\mathcal{V}}^T$, where $\mathrm{SVT}(\cdot)$ represents the singular value thresholding operator in the Fourier domain. It is defined as $\mathrm{SVT}(\boldsymbol{\mathcal{S}}) = \mathrm{IFT}(\max\{\widetilde{\boldsymbol{\mathcal{S}}}(i,i,k) - \frac{\lambda_1}{\beta^l}, 0\})$, with $\mathrm{IFT}(\cdot)$ representing the inverse Fourier transform along the third dimension. Therefore, the mathematical form of denoiser $\boldsymbol{\mathcal{D}}_{\mathrm{T}*}(\cdot)$ can be expressed as

$$\boldsymbol{\mathcal{D}}_{\mathrm{T}*}(\boldsymbol{\mathcal{X}}^l + \frac{\boldsymbol{\mathcal{E}}_1^l}{\beta^l}) = \boldsymbol{\mathcal{U}} * \mathrm{SVT}(\boldsymbol{\mathcal{S}}) * \boldsymbol{\mathcal{V}}^T. \tag{12}$$

The above solution only consists of fast inverse Fourier transform and matrix SVD, which can be carried out very efficiently. On the contrary, if one chooses other low-rank tensor models (e.g., the Tucker model as in Ref. [6]), the denoising step becomes an NP-hard optimization problem, and no elegant and semi-analytical optimal solutions like (12) exist.

### B. Pre-trained Deep Image Denoiser

Dealing with the denoiser $\boldsymbol{\mathcal{D}}_{\Phi}(\cdot)$ is more challenging. The denoiser is associated with an ideally data-driven regularizer. So far, we have not even introduced the explicit representation of the regularizer—optimizing (7) seems intractable.

A data-driven way to tackle (7) is to make $\Phi$ implicit and to view $\boldsymbol{\mathcal{D}}_{\Phi}(\cdot)$ as a neural network. Ideally, the neural network is designed to map noisy SSFs to their clean counterparts. If such a neural network can be trained using SSF data, it naturally incorporates characteristic information of SSF data and serves for SSF denoising. However, training such a neural network can be quite challenging in practice, if not outright impossible. This is because training a denoising neural network designated to SSF may need a massive amount of SSF

data. For example, the denoiser trained for natural images in Ref. [25] used ImageNet[26] and Waterloo Exploration Database[27] training samples. However, acquiring such an amount of high-quality training samples in the domain of SSF analysis is a daunting challenge.

To address this issue, as illustrated in Fig. 3, we can leverage the similarity between the denoising processes of natural images and SSFs, and use a pre-trained deep image denoiser instead. This "zero-shot" approach avoids the need for extensive SSF historical data and neural network training. The pre-trained deep image denoiser can be straightforwardly applied to the SSF denoising task and then be *plugged* into the ADMM algorithm to aid the reconstruction task.

The next step is to select a deep image denoiser suitable for our task. As shown in **Algorithm 1**, the deep denoiser should be capable of removing noise with different variances, as the noise variance changes due to the variation of $\beta$. In this paper, we consider FFDNet,[25] a high-performance and flexible deep denoiser that primarily consists of convolutional layers with batch normalization and ReLU activation. The FFDNet has been pre-trained on various high-quality image datasets, including the Berkeley Segmentation Dataset (BSD),[28] Waterloo Exploration Database,[27] and ImageNet[26] dataset. The extensive pre-training enables FFDNet to achieve superior denoising performance, ultimately enhancing the overall reconstruction performance. Besides, as a non-blind denoiser, it can handle denoising problems with a wide range of noise levels via a single network. Moreover, it is worth noting that since FFDNet is specifically designed as a denoiser and the subproblem is a denoising problem, we utilize the output of FFDNet instead of its intermediate layers.

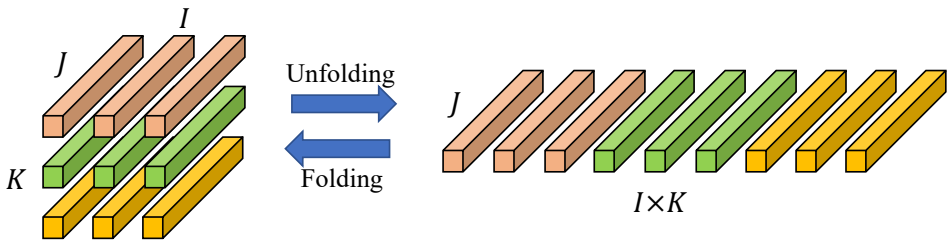



FIG. 4. Illustration of the folding and unfolding operations.

---

Using FFDNet, we can specify $\boldsymbol{\mathcal{D}}_\Phi(\cdot)$ as follows:

$$\boldsymbol{\mathcal{D}}_\Phi(\boldsymbol{\mathcal{X}}^l + \frac{\boldsymbol{\mathcal{E}}_2^l}{\beta^l}) = \text{Fold}[\text{FFDNet}(\text{Unfold}(\boldsymbol{\mathcal{X}}^l + \frac{\boldsymbol{\mathcal{E}}_2^l}{\beta^l}), \eta^l)], \tag{13}$$

where $\eta^l = \sqrt{\lambda_2/\beta^l}$ is the noise variance in the $l$th iteration. Note that FFDNet is an image denoiser designed to handle data with one or three channels. However, in our case, the SSF data has multiple channels. To address the dimension mismatch issue, we use the unfolded SSF data as input and fold the denoising result back to the original 3D structure. The folding and unfolding operations are illustrated in Fig. 4. For more in-depth information and detailed explanations, readers can refer to Appendix A of Ref. [6]. Additionally, it is important to mention that the unfolded SSF data may have different spatial sizes compared to the training images. However, this does not pose a problem as FFDNet is a fully convolutional neural network composed of multiple convolutional layers. The convolution operations in these layers are applied across the entire input data, regardless of its size. Furthermore, FFDNet is a non-blind denoiser that effectively utilizes the noise variance as an input, resulting in an improved solution for problem (7).

It is worth noting that instead of explicitly formulating the regularizer $\Phi(\cdot)$, we leverage a deep denoiser that implicitly specifies $\Phi(\cdot)$. This approach allows the incorporation of

data-driven prior information and yields enhanced reconstruction performance, as will be demonstrated by the experimental results in the next section.

In this paper, we employ FFDNet as the denoiser due to its state-of-the-art performance in denoising. However, it is important to mention that other pre-trained image denoisers can also be plugged into the ADMM framework.

### C. Algorithm Summary

After *"plugging"* in the deep image denoiser, the ADMM framework can *"play"* with it by incorporating it into the denoising step. This technique is known as "plug-and-play (PnP)" in the deep learning literature and has recently been widely used in various tasks.[15,21,22] However, PnP was mostly applied to domains like medical imaging,[29] remote sensing[30] and optical tomographic imaging[31]. These domains are fairly far away from SSF reconstruction as the problems are still within the image processing domains. SSF reconstruction deals with a very different type of data that is a physical acoustics-environmental field. It has been unclear whether or not denoisers trained for natural images are useful for denoising SSF.

This paper represents *the first attempt* to harness the power of the PnP method in this area. Although images and SSFs have distinct data distributions, it is important to note that they share certain fundamental "structural" characteristics that differentiate them from noise. Notably, both images and SSFs demonstrate approximate low-rank properties and exhibit strong spatial correlations, as demonstrated in Ref. [8]. These shared properties establish a meaningful connection between the denoising processes of images and SSFs,

**Algorithm 2: PnP Deep Tensor Algorithm for Zero-Shot 3D Ocean SSF Reconstruction**

**Input:** $\boldsymbol{\mathcal{Y}}$, $\boldsymbol{\mathcal{O}}$, maximum iteration number $L$.

**Initialize:** $l \leftarrow 0$, $\boldsymbol{\mathcal{X}}$, $\boldsymbol{\mathcal{W}}$, $\boldsymbol{\mathcal{Z}}$, $\boldsymbol{\mathcal{E}}_1$, $\boldsymbol{\mathcal{E}}_2$, $\beta^0, \lambda_1, \lambda_2$.

**While** $l < L$ or not converged **do**

$\boldsymbol{\mathcal{X}}^l + \frac{\boldsymbol{\mathcal{E}}_1^l}{\beta^l} = \boldsymbol{\mathcal{U}} * \boldsymbol{\mathcal{S}} * \boldsymbol{\mathcal{V}}^T$,

$\boldsymbol{\mathcal{W}}^{l+1} \leftarrow \boldsymbol{\mathcal{U}} * \mathrm{SVT}(\boldsymbol{\mathcal{S}}) * \boldsymbol{\mathcal{V}}^T$,

$\boldsymbol{\mathcal{Z}}^{l+1} \leftarrow \mathrm{FFDNet}(\boldsymbol{\mathcal{X}}^l + \frac{\boldsymbol{\mathcal{E}}_2^l}{\beta^l}, \eta^l)$, where $\eta^l = \sqrt{\lambda_2/\beta^l}$

Update $\boldsymbol{\mathcal{X}}$ by solving (8),

Update $\boldsymbol{\mathcal{E}}$ via (9),

Update $\beta$ via (10),

$l \leftarrow l + 1$,

**end while**

**Return** $\boldsymbol{\mathcal{X}}^l$.

despite the variations in their data distributions. Additionally, the integration of tensor t-SVD-based low-rank regularization is essential in fostering global coherency, particularly in scenarios with highly sparse measurements. Through our ablation study in Sec. V, we confirm that the combination of tSVD and deep denoiser provides a compelling blend of

global and detailed prior information for SSF. The proposed algorithm, based on the formulations described above, is summarized in **Algorithm 2**. Note that in **Algorithm 1** is a general algorithm while **Algorithm 2** explicitly details the denoising steps using tensor SVD and the deep denoiser.

*Remark 1(Computational Complexity)*: As shown in **Algorithm 2**, the method comprises three main steps, with the first two steps being the most computationally intensive. Specifically, The first step involves solving a t-SVD problem, which has a computational complexity of $\mathcal{O}(K\min(IJ^2, I^2J) + IJK\log(K))$. The second step is the denoising process, which primarily entails convolution operations. The computational complexity of passing through the neural network can be expressed as $\mathcal{O}(IJKn_fn_ln_k)$, where $n_f$ represents the number of features, $n_l$ represents the number of layers, and $n_k$ represents the kernel size. Hence, the total computational complexity in one iteration of our method is given by $\mathcal{O}(K\min(IJ^2, I^2J) + IJK\log(K) + IJKn_fn_ln_k)$. More discussions about the running time can be found in Appendix C.

## IV. NUMERICAL RESULTS AND DISCUSSIONS

In this section, numerical results based on 3D ocean SSF data are reported to demonstrate the effectiveness of the proposed PnP deep tensor approach (labeled as PnP).

### A. Experimental Settings

**3D SSF Data**: In this paper, we analyze the Philippines SSF data denoted as $\mathbf{X} \in \mathbb{R}^{100\times100\times10}$. The data are derived from the hybrid coordinate ocean model (HYCOM) and

cover the geographical region of 17°N-21°N and 123°E-131°E, with a time stamp of 11:00 on June 6, 2022. Spanning an area of 440 km × 879 km × 180 m, the data exhibit a latitude resolution of 4.40 km, a longitude resolution of 8.79 km, and a vertical resolution of 20 m. The depth of the data ranges from 0 m to 180 m.

**Sampling Scenarios**: This paper mainly focuses on challenging scenarios where multiple CTD chains and PIES are deployed over an area, leading to a sparse and fiber-wise sampling of sound speeds, as illustrated in Fig. 5. Furthermore, the paper investigates the impact of sensor placement, whether random or regular, on the reconstruction performance of different methods. In the case of regular sampling patterns, the measurements are evenly spaced horizontally, whereas random patterns do not adhere to such regularity.

In all scenarios, the data is assumed to be corrupted by i.i.d. Gaussian noise with a zero mean and standard deviation of $\sigma$. The sampling ratio is defined as $\rho = \frac{\sum_{i,j,k} \boldsymbol{\mathcal{O}}_{i,j,k}}{IJK}$, which measures the ratio of observed entries to the total number of unknown entries.

**Implementation**: The pre-trained FFDNet model used in the experiments is the Matlab version. All experiments were conducted on a computer equipped with a 3.7 GHz 4-Core Intel i3 CPU and 16GB memory. The running environment is Matlab 2021, which also serves as the programming language.

**Performance Measure**: The metric that assesses the reconstruction performance of different methods is the root mean squared error (RMSE), defined as

$$\text{RMSE} = \sqrt{\frac{1}{IJK}\|\hat{\boldsymbol{\mathcal{X}}} - \boldsymbol{\mathcal{X}}\|_{\text{F}}^2}, \tag{14}$$

where $\boldsymbol{\mathcal{X}}$ is the ground-truth and $\hat{\boldsymbol{\mathcal{X}}}$ is the reconstructed SSF.

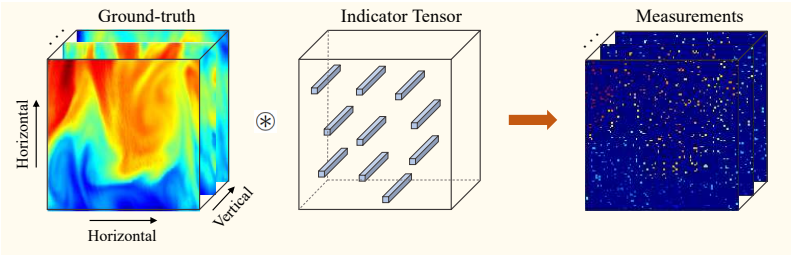



FIG. 5. Illustration of the sparse and fiber-wise sampling.

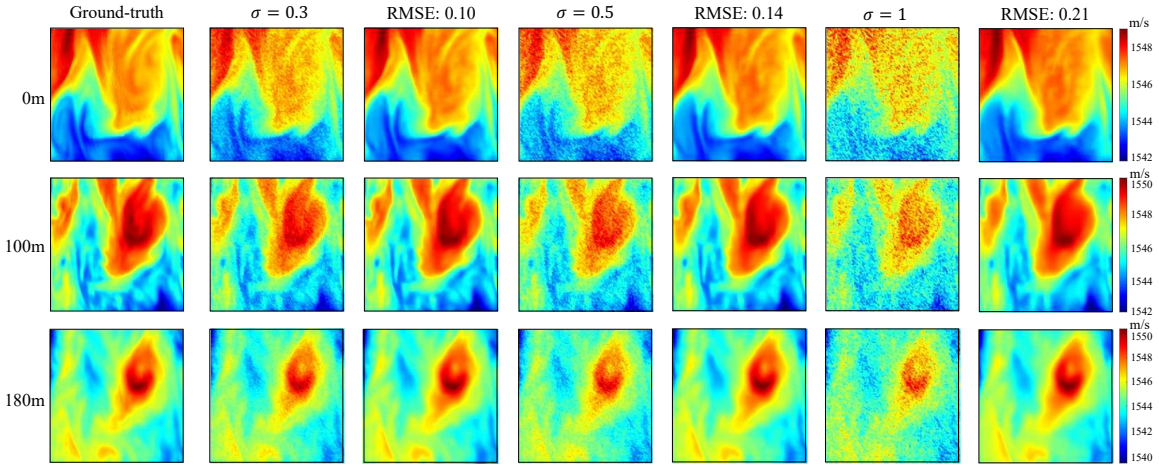



FIG. 6. Visual effects of the denoising results under different noise standard deviations $\sigma$. The first column is the ground-truth SSF data, followed by noisy data with different $\sigma$ and the denoising results. The RMSEs are shown above the denoising results.

## B. Deep Image Denoiser for SSF Denoising

Sec. II has established the connection between the reconstruction and denoising problems, highlighting the importance of successful denoising in improving SSF reconstruction performance. In this subsection, we empirically demonstrate that a pre-trained deep image denoiser, specifically the FFDNet, can effectively mitigate noises from SSF data.

Particularly, consider an AWGN denoising problem

$$\boldsymbol{\mathcal{P}} = \boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{N}}, \tag{15}$$

where $\boldsymbol{\mathcal{P}}$ is the observed noisy SSF data and $\boldsymbol{\mathcal{N}}$ is a noise tensor with each element following $\mathcal{N}(0, \sigma^2)$. To test the denoising performance of FFDNet on SSF data, we use $\boldsymbol{\mathcal{P}}$ as the input of FFDNet and calculate the RMSE between the output of FFDNet and the ground-truth data. The denoising results under different noise standard deviations ($\sigma$) are presented in Fig. 6. Additional denosing results under much larger noise powers ($\sigma > 1$) are given in Appendix C.

Fig. 6 demonstrates that the FFDNet, *despite not being trained on SSF data*, effectively removes the noises with varying $\sigma$ and *restores the fine-grained details of SSF*. This result lays the foundation of the encouraging reconstruction results presented in the next subsection.

### C. Comparisons with SOTA Methods

In this subsection, we assess the reconstruction performance of the proposed PnP deep tensor approach under different sampling scenarios and compare it with the SOTA methods.

**SOTA Methods**: The SOTA methods compared in this paper include: 1) matrix-based methods, represented by the recently proposed graph-guided Bayesian matrix completion (BMCG);[8] 2) tensor-based methods, including the low-rank tensor completion (LRTC),[32] LRTC with total variation (LRTC-TV),[33] and the recently proposed tensor neural network-based method (TensorNN);[9] 3) nonparametric statistical learning-based methods, represented by the Gaussian process regression (GPR).[34,35]

**Model Settings**: The kernel of GPR is the widely used radial basis function (RBF), with the hyper-parameters being learned via evidence optimization.[36] In BMCG, the graph Laplacian matrix is constructed using the same kernel. The tensor rank surrogate function used in LRTC is the tensor nuclear norm. TensorNN has three layers, with dimensions being (20, 20, 2), (50, 50, 5), and (100, 100, 10), respectively. Further details about the number of parameters in different models are presented in Appendix B.

### *1. Random Fiber-wise Sampling*

We begin by evaluating the reconstruction performance of various methods under random fiber-wise sampling scenarios. We consider this scenario for two reasons: First, sampling the SSF as vertical line array is the most realistic setting if sensors like multiple CTD chains and PIES are used; see Fig. 1. Second, the random fiber sampling appears to present a more challenging inverse problem compared to random entry sampling (see Refs. [37, 38]), as many fibers are completely unobserved under the former. Table I shows the averaged RMSEs of different methods under various sampling ratios and noise powers. It is evident that the proposed PnP approach outperforms all other methods in terms of reconstruction quality. Moreover, we provide visual inspections of the reconstructed SSFs under different settings in Fig. 7 to Fig. 8. And the error surfaces are depicted in Fig. 9. Based on these figures and the table, discussions are provided below to gain further insights into the performance of different methods.

Among the methods we evaluated, LRTC, which only utilizes a hand-crafted low-rank tensor regularizer, gives the worst reconstruction results in all scenarios. It can only provide

TABLE I. The averaged RMSEs over three Monte-Carlo trials of different algorithms under different sampling ratios and noise powers.

| $\rho$ | $\sigma$ | **BMCG** | **GPR** | **LRTC** | **LRTC-TV** | **TensorNN** | **PnP** |
|---|---|---|---|---|---|---|---|
| 0.05 | 0.1 | 0.46 | 0.35 | 1.59 | 1.08 | 0.64 | **0.33** |
| | 0.3 | 0.47 | 0.40 | 1.60 | 1.09 | 0.64 | **0.38** |
| | 0.5 | 0.51 | 0.44 | 1.61 | 1.12 | 0.65 | **0.43** |
| 0.10 | 0.1 | 0.31 | 0.25 | 1.07 | 0.46 | 0.47 | **0.20** |
| | 0.3 | 0.35 | 0.31 | 1.10 | 0.51 | 0.48 | **0.25** |
| | 0.5 | 0.39 | 0.37 | 1.13 | 0.57 | 0.48 | **0.31** |
| 0.15 | 0.1 | 0.26 | 0.21 | 0.69 | 0.31 | 0.24 | **0.16** |
| | 0.3 | 0.31 | 0.27 | 0.74 | 0.38 | 0.26 | **0.22** |
| | 0.5 | 0.36 | 0.33 | 0.81 | 0.45 | 0.31 | **0.28** |
| 0.20 | 0.1 | 0.22 | 0.18 | 0.46 | 0.26 | 0.26 | **0.13** |
| | 0.3 | 0.26 | 0.24 | 0.54 | 0.33 | 0.28 | **0.20** |
| | 0.5 | 0.32 | 0.30 | 0.64 | 0.42 | 0.31 | **0.26** |

the mean value of the SSF under very sparse sampling (e.g., $\rho = 0.05$, as shown in Fig. 8), and loses all the sound speed variation details. This suggests that in very sparse and fiber-wise sampling scenarios, the low-rank tensor regularizer, which only enforces global coher-

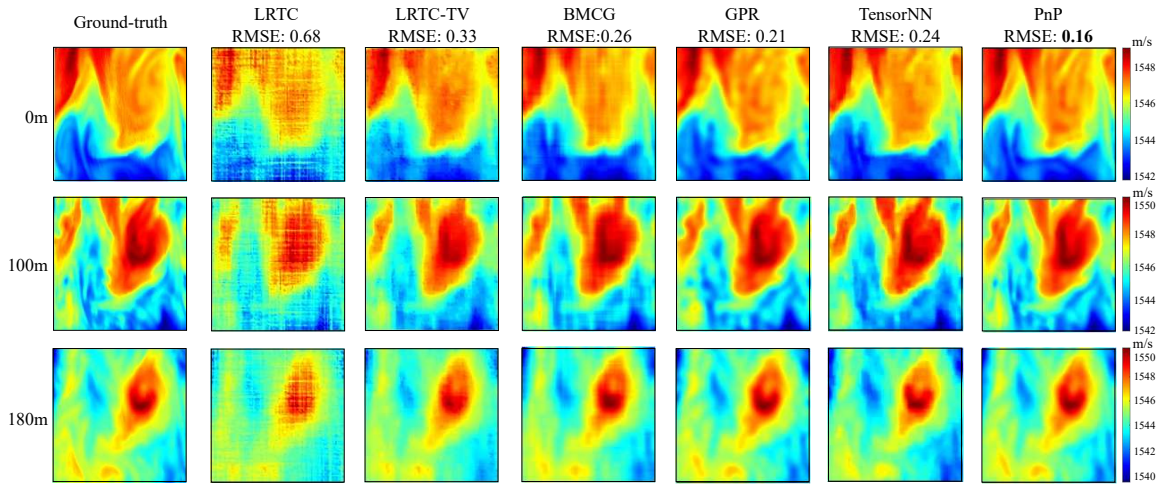



FIG. 7. Visual effects of the reconstructed SSFs under $\rho = 0.15$ and $\sigma = 0.1$ in one single Monte-Carlo trial. The RMSEs of different methods are shown above the subfigures in the top row.

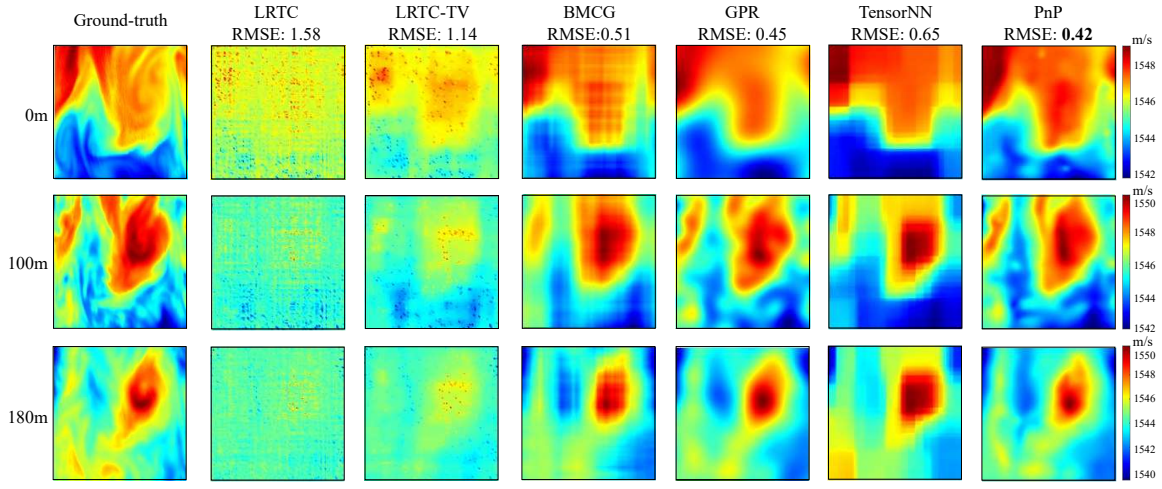



FIG. 8. Visual effects of the reconstructed SSFs under $\rho = 0.05$ and $\sigma = 0.5$ in one single Monte-Carlo trial. The RMSEs of different methods are shown above the subfigures in the top row.

ence, is insufficient for successfully reconstructing the fine-grained SSF variations. Although augmenting the TV regularizer can improve the performance, the reconstruction results of LRTC-TV are still unsatisfactory, as shown in Fig. 7 and Fig. 8. These results demonstrate

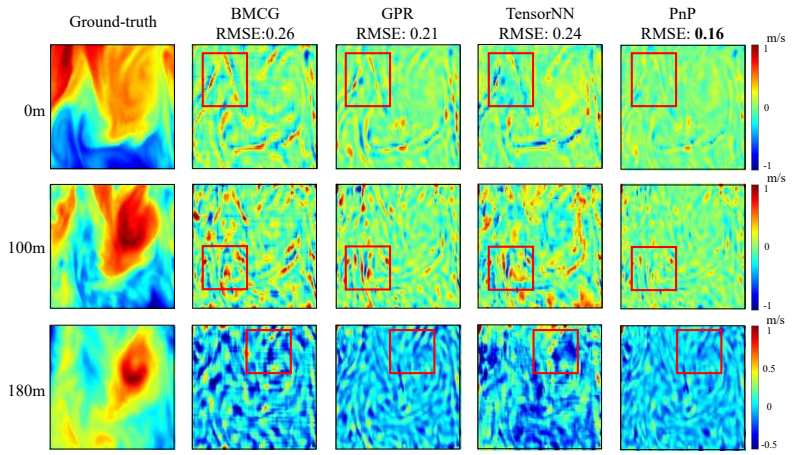



FIG. 9. Visual effects of the error surface in one single Monte-Carlo trial of different methods under $\rho = 0.15$ and $\sigma = 0.1$. The RMSEs are shown above the subfigures in the top row.

that without data-driven regularizers, commonly-used hand-crafted ones cannot rescue the SSF reconstruction under challenging fiber-wise sampling scenarios.

TensorNN, although showing encouraging reconstruction performance in the randomly sampled scenario, fails to provide excellent reconstruction results in fiber-wise sampling scenarios. The reason for this is that the backbone of TensorNN still relies on low-rank tensor computations, despite incorporating several non-linear activation functions. Therefore, when a large portion of the 3D area is missing, the learning problem for model parameters becomes ill-posed, thus significantly degrading the reconstruction results.

The matrix-based method BMCG recovers the 3D SSF slice by slice. Therefore, fiber-wise sampling has no effect on its performance. Additionally, the graph structure characterized by the RBF kernel enables it to reconstruct smooth SSFs even under very sparse samples.[8] Consequently, BMCG achieves the third-best reconstruction performance in this challenging scenario, even without exploiting the 3D structure of the SSF.

Unlike the parametric model discussed above, GPR is a non-parametric statistical method that aims to model the functionals of SSF variations. Although no research work has formally discussed its application in SSF reconstruction, we still consider it a potent competitor for comparison with our proposed method. It is apparent that with the RBF kernel and evidence-maximization-based hyper-parameter optimization, GPR achieves the second-best performance in almost all scenarios. However, constrained by the RBF kernel, which can only model smoothness, GPR cannot recover the fine-grained SSF variation details as our proposed PnP method does, see Fig. 9. The performance of GPR can be further improved via the development of a data-driven deep kernel,[39] and its notorious high computational cost[8,40] can also be reduced through recent advances in machine learning. However, these aspects are beyond the scope of this paper.

It can be seen from Fig. 9 that the proposed PnP method outperforms other methods in restoring these fine-scale and sharp details, as highlighted by the red boxes. This can be attributed to the effectiveness of FFDNet in removing noise from complex natural images that contain numerous intricate details. Consequently, it is reasonable that these CNN denoisers can effectively denoise SSF data that exhibit similar sharp patterns. Furthermore, the utilization of the plug-and-play technique enables the proposed method to simultaneously exploit the hand-crafted prior information (i.e., low-rankness) and the data-driven prior information (encoded in the deep denoiser), thus achieving the best reconstruction performance in all scenarios.

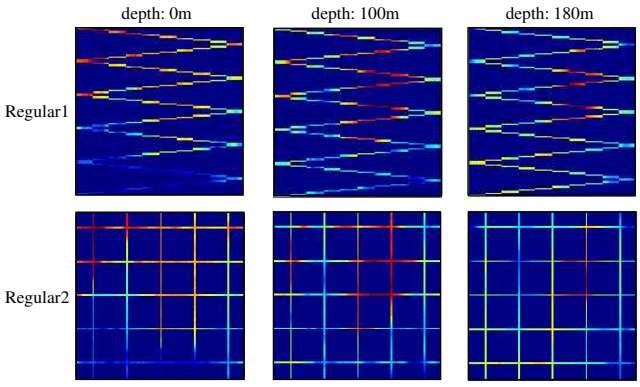



FIG. 10. Samples collected from the two regular sampling patterns considered in our experiments.

### 2. *Regular Fiber-wise Sampling*

Next, we examine the sampling scenarios where multiple CTD chains and PIES are deployed regularly. Specifically, we study two regular patterns denoted as *Regular1* and *Regular2*, which are illustrated in Fig. 10. For ease of comparison, both of these patterns have a sampling ratio of $\rho = 0.1$. Experimental results under different sampling ratios can be found in Appendix C. Such regular sampling patterns often happen when the sensors are equipped on ships, and thus the scenarios are of great interest.

Table II shows the RMSEs of different methods under different sampling patterns and noise standard deviations, and the visual inspections are depicted in Fig. 11 and Fig. 12. To facilitate comparison, we have included the results of random sampling in Table II. It is evident that the proposed PnP method achieves the highest reconstruction accuracies in all instances. Additionally, the following conclusions can be drawn.

*Random sampling outperforms regular sampling for reconstruction:* The results presented in Table II demonstrate that the reconstruction RMSEs are significantly higher for regular

TABLE II. RMSEs of different algorithms under different sampling patterns and noise powers.

| Pattern | $\sigma$ | **BMCG** | **GPR** | **LRTC** | **LRTC-TV** | **TensorNN** | **PnP** |
|---|---|---|---|---|---|---|---|
| Random | 0.1 | 0.31 | 0.25 | 1.07 | 0.46 | 0.47 | **0.20** |
| | 0.3 | 0.35 | 0.31 | 1.10 | 0.51 | 0.48 | **0.25** |
| | 0.5 | 0.39 | 0.37 | 1.13 | 0.57 | 0.48 | **0.31** |
| *Regular1* | 0.1 | 0.59 | 0.70 | 1.67 | 1.28 | 0.50 | **0.35** |
| | 0.3 | 0.52 | 0.52 | 1.67 | 1.29 | 0.51 | **0.37** |
| | 0.5 | 0.51 | 0.47 | 1.68 | 1.30 | 0.52 | **0.40** |
| *Regular2* | 0.1 | 0.81 | 0.70 | 1.59 | 1.32 | 0.61 | **0.44** |
| | 0.3 | 0.56 | 0.47 | 1.59 | 1.33 | 0.62 | **0.45** |
| | 0.5 | 0.52 | 0.48 | 1.60 | 1.36 | 0.66 | **0.47** |

sampling scenarios compared to random sampling scenarios, when the number of samples is the same. Therefore, deploying a fixed number of sensors randomly offers distinct advantages for reconstructing SSFs.

*Supervised data-driven regularization does matter:* Low-rank-based methods, including LRTC and LRTC-TV, give poor reconstruction results because the hand-crafted low-rank regularizer is not informative enough in the challenging regular sampling scenarios. On the other hand, TensorNN, although with unsupervised data-driven prior, still fails to capture the detailed variations of SSF from such limited data samples. Nevertheless, with the assis-

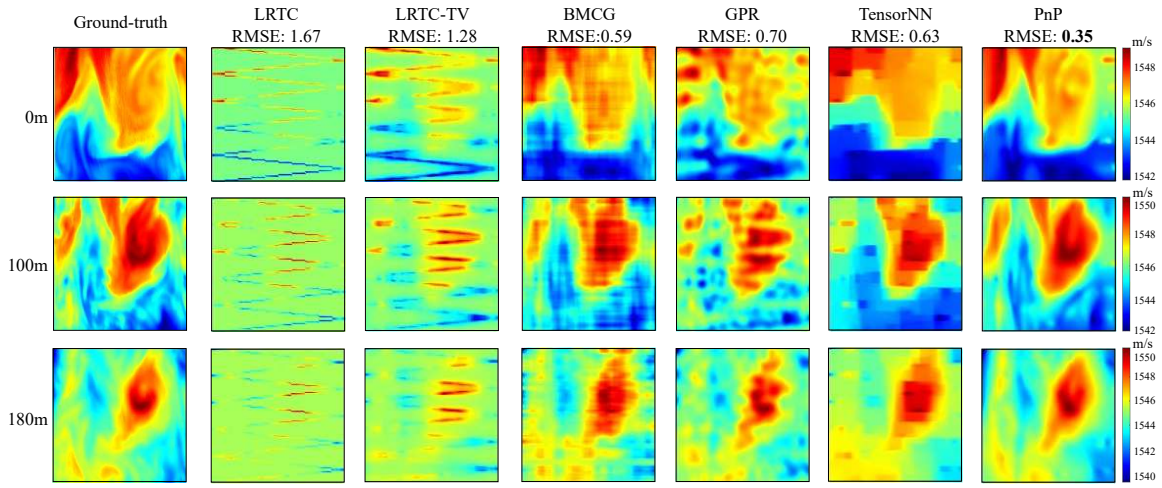



FIG. 11. Visual effects of the reconstructed SSFs under *Regular1* sampling pattern. The noise standard deviation $\sigma = 0.1$. The RMSEs of different methods are shown above the subfigures in the top row.

tance of the supervised data-driven prior information, the proposed PnP method still gives satisfactory results, demonstrating the importance of the supervised data-driven regularizer in such scenarios.

*Challenging optimization of scale parameters under regular sampling:* In regular sampling scenarios, BMCG and GPR perform worse with a higher signal-to-noise ratio (SNR). The reason is that their reconstruction performances are affected not only by the noise variance but also by the learned hyperparameters. These hyperparameters, such as the scale parameter in GPR and the rank parameter in BMCG, are automatically learned from the data using evidence maximization-based hyperparameter learning. Note that the evidence maximization-based hyper-parameter learning problem is typically non-convex. The quality of the obtained solution can vary depending on factors such as the noise variance and the

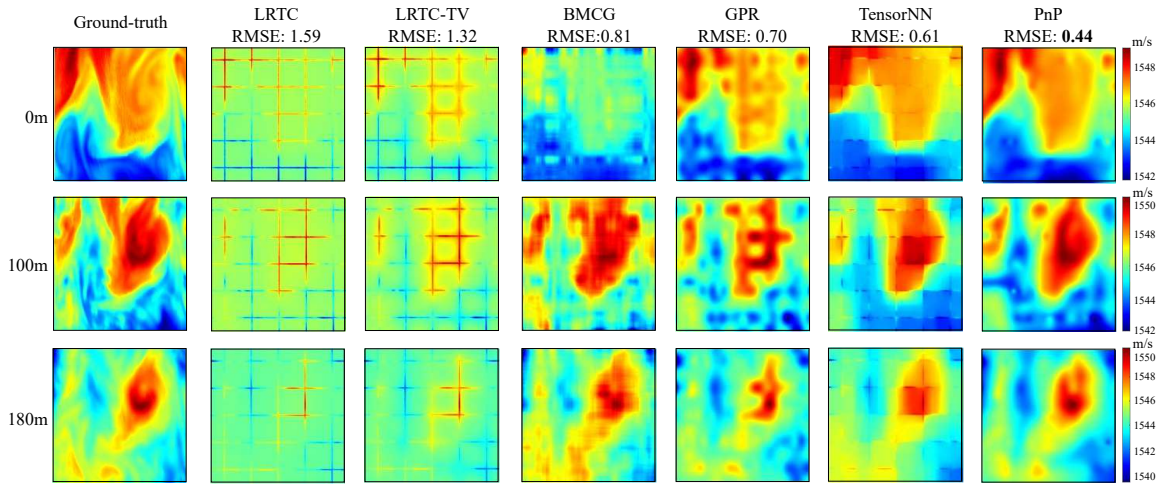



FIG. 12. Visual effects of the reconstructed SSFs under *Regular2* sampling pattern. The noise standard deviation $\sigma = 0.1$. The RMSEs of different methods are shown above the subfigures in the top row.

sampling pattern. In cases of regular sampling with sparse measurements (e.g., $\rho = 0.1$) and high SNR (e.g., $\sigma = 0.1$), the limited samples fail to capture the global coherence across the field, while the high SNR leads to an "over-confidence" in the model. Consequently, the scale parameter of GPR is underestimated, resulting in drastic variations in the reconstruction results, which are visible as artifacts and lead to high RMSE values (see Fig. 11 and Fig. 12). However, as the noise variance increases, the over-confidence diminishes, allowing for larger scale parameters and improved reconstruction performance.

### *3. Ablation Study: Contributions of Two Regularizers*

As discussed in Sec. II, two regularizers are employed to enhance the reconstruction performance. In this subsection, we conduct experiments to demonstrate the contributions

TABLE III. RMSEs of different methods under different sampling ratio $\rho$. The noise standard deviation $\sigma = 0.3$.

| $\rho$ | **LRTC** | **DPTC** | **PnP** |
|---|---|---|---|
| 0.05 | 1.60 | 0.44 | **0.38** |
| 0.10 | 1.10 | 0.29 | **0.24** |
| 0.20 | 0.54 | 0.24 | **0.20** |

of these different terms. Specifically, we consider two cases: 1) utilizing only the tensor nuclear norm regularizer, and 2) utilizing only the data-driven regularizer. In the first case, the model is identical to LRTC. In the second case, we set $\lambda_1 = 0$ in Eq.(2) and refer to the model as deep prior-aided tensor completion (DPTC). The reconstruction RMSEs of various methods under different sampling ratios are presented in Table III.

It is evident that PnP, which incorporates both regularizers, outperforms the other two methods, each using a single regularizer, in all scenarios. This indicates that both of these regularizers contribute to the improvement in performance. Additionally, the impact of the TNN regularizer is more significant at lower sampling ratios. Notably, DPTC outperforms LRTC in all scenarios, highlighting the importance of incorporating the data-driven regularizer in challenging fiber-wise sampling scenarios. In Appendix B, the ablation studies concerning the impact and tuning of the regularization parameters, $\lambda_1$ and $\lambda_2$, on the reconstruction performance are presented.

## V. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we proposed a plug-and-play ADMM approach for accurately reconstructing ocean 3D SSFs in a zero-shot manner, that is, without requiring access to any historical SSF training data. Our method leverages a deep image denoiser (specifically, FFDNet), which was pre-trained on natural image datasets, and a low tensor rank prior. These two key ingredients are incorporated into an ADMM-based SSF reconstruction process. Our proposed method demonstrates promising SSF reconstruction results, particularly in challenging scenarios involving very sparse and fiber-wise sampling. Experimental evaluations using real-world SSF data have confirmed the favorable performance of our plug-and-play method, outperforming other SOTA techniques. This highlights the effectiveness of incorporating supervised data-driven prior information derived from a pre-trained deep image denoiser. Our algorithm construction and experiment results also for the first time—to our best knowledge—showed that image denoising and SSF denoising share similar characteristics, despite the two types of data being very different. Our discovery may open many doors for exploiting the wide availability of image data to come up with data-driven solutions that can be transferred to the SSF processing domain.

This study explores the use of FFDNet as a denoiser within the PnP framework. In future research, it would be valuable to explore alternative denoisers to further improve the reconstruction performance. For example, investigating the potential of using the diffusion model as a denoiser holds promise and deserves further investigation. Furthermore, the hyperparameters of the proposed model in our experiments have not been optimized.

Exploring methods to optimize hyperparameters from limited samples is also an intriguing area of research.

## VI. AUTHOR DECLARATIONS

Conflict of Interest Statement: The authors confirm that there are no conflicts of interest associated with the publication of this research article. Any potential conflicts arising from funding sources or affiliations have been disclosed accordingly.

## VII. DATA AVAILABILITY

The data and codes supporting the findings of this study can be found at:

https://github.com/OceanSTARLab/DeepPnP.

## VIII. ACKNOWLEDGEMENT

This work of L. Cheng was supported in part by the National Natural Science Foundation of China under Grant 62371418, in part by the Fundamental Research Funds for the Central Universities (226-2023-00012), and in part by the Zhejiang University Education Foundation Qizhen Scholar Foundation. The work of X. Fu was supported in part by the National Science Foundation (NSF) under Projects NSF ECCS-2024058 and NSF CCF-2210004.

## APPENDIX A: AWGN DENOISING PROBLEM

Consider the AWGN problem

$$\boldsymbol{\mathcal{P}} = \boldsymbol{\mathcal{W}} + \boldsymbol{\mathcal{N}}. \tag{A1}$$

where each element of $\boldsymbol{\mathcal{N}}$ follows a Gaussian distribution, i.e., $\boldsymbol{\mathcal{N}}_{i,j,k} \sim \mathcal{N}(0, \sigma^2)$. Then the likelihood function can be expressed as $p(\boldsymbol{\mathcal{P}}_{i,j,k}|\boldsymbol{\mathcal{W}}_{i,j,k}) = \mathcal{N}(\boldsymbol{\mathcal{W}}_{i,j,k}, \sigma^2)$. Thus the maximum a posteriori (MAP) estimation problem of $\boldsymbol{\mathcal{W}}$ can be expressed as

$$\max_{\boldsymbol{\mathcal{W}}} \log p(\boldsymbol{\mathcal{P}}|\boldsymbol{\mathcal{W}}) + \log p(\boldsymbol{\mathcal{W}}) \tag{A2}$$

$$= \max_{\boldsymbol{\mathcal{W}}} \log p(\boldsymbol{\mathcal{W}}) - \frac{1}{2\sigma^2}\|\boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{P}}\|_{\mathrm{F}}^2. \tag{A3}$$

Assuming that the prior distribution $p(\boldsymbol{\mathcal{W}}) \propto \exp(-||\boldsymbol{\mathcal{W}}||_{\mathrm{T}*})$. Let $\boldsymbol{\mathcal{P}} = \boldsymbol{\mathcal{X}}^l + \frac{\boldsymbol{\mathcal{E}}_1^l}{\beta^l}$ and the noise variance $\sigma^2 = \frac{\lambda_1}{\beta^l}$. Then, the MAP estimation problem can be formulated as

$$\max_{\boldsymbol{\mathcal{W}}} -\|\boldsymbol{\mathcal{W}}\|_{\mathrm{T}*} - \frac{\beta^l}{2\lambda_1}\|\boldsymbol{\mathcal{W}} - \left(\boldsymbol{\mathcal{X}}^l + \frac{\boldsymbol{\mathcal{E}}_1^l}{\beta^l}\right)\|_{\mathrm{F}}^2, \tag{A4}$$

which is mathematically equivalent to the denoising problem in Eq. (6).

## APPENDIX B: ADDITIONAL MODEL DETAILS

The number of parameters and hyper-parameters of different models are presented in Table IV. Specifically, BMCG is a Bayesian matrix completion method, and its parameters consist of the factor matrices. GPR is a non-parametric method, meaning it does not involve specific parameters. LRTC is a low-rank tensor completion method, and its parameters encompass the entire data tensor. LRTC-TV is an LRTC method incorporating total variation

(TV) regularization, which introduces an additional regularization hyper-parameter. TensorNN is a hierarchical tensor decomposition model with non-linear activation functions, and its parameters encompass the core tensor and factor matrices.

Regarding optimizing the hyperparameters, we ensure a fair comparison by employing either hyperparameter optimization methods or adopting empirically recommended hyperparameter values from the original papers of the state-of-the-art methods. Specifically, for GPR, the model evidence (or the marginal distribution of the measurements) is maximized with respect to the kernel parameters, by which the hyperparameters are optimized. Detailed descriptions of this approach can be found in Ref. [34]. For the models BMCG, LRTC, LRTC-TV, and TensorNN, we adopt the recommended hyperparameter values provided in the original papers. These values have been determined through extensive experimentation and analysis in their respective studies, ensuring a reliable and fair comparison.

For our proposed model, the hyperparameters are set via trial and error. To be specific, we test a range of hyperparameters offline and visually observe the reconstruction results. Our observation is that the reconstruction result is not sensitive to the hyperparameters within certain ranges; see the ablation study in Fig. 13. In our experiments, we manually select a hyperparameter from the range (without optimizing). Despite the absence of additional optimization, our approach consistently yields excellent reconstruction results.

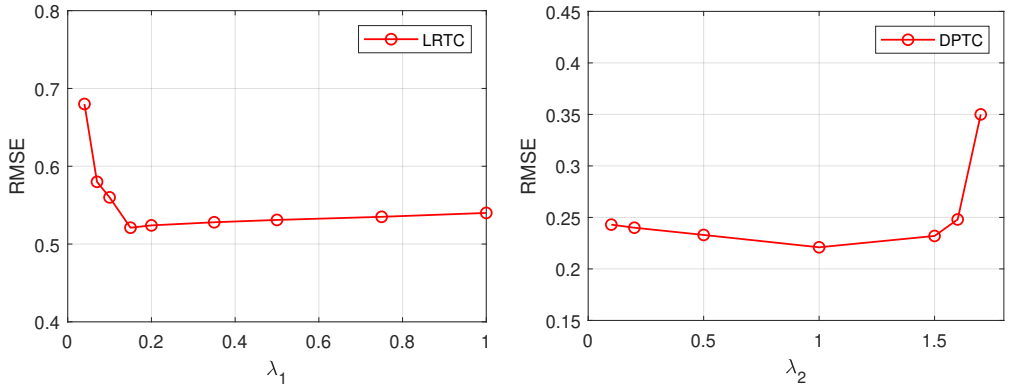



FIG. 13. Ablation studies of the regularization parameters $\lambda_1$ and $\lambda_2$.

TABLE IV. The number of parameters and hyper-parameters of different models. Note that GPR is a nonparametric model.

| Model | **# Parameters** | **# Hyper-parameters** |
|---|---|---|
| BMCG | 12000 | 2 |
| GPR | - | 2 |
| LRTC | 100000 | 1 |
| LRTC-TV | 100000 | 2 |
| TensorNN | 12860 | 1 |
| PnP | 100000 | 2 |

## APPENDIX C: ADDITIONAL EXPERIMENTAL RESULTS

### 1. SSF Denoising

Fig. 14 presents the denoising results under different noise powers, demonstrating the effectiveness of FFDNet in SSF denoising. The visual effects of the denoising results are shown in Fig.15.

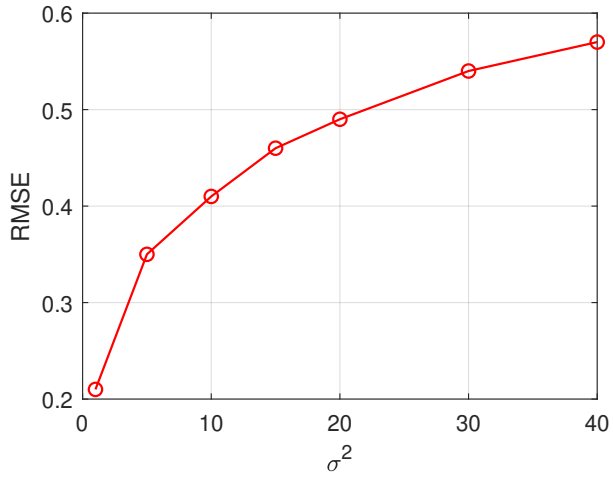



FIG. 14. The RMSEs of the denoised results with respect to different values of the noise variances.

### 2. Reconstruction under regular sampling

The RMSEs of different methods under different sampling ratios in the *Regular2* sampling pattern are presented in Table V. It can be seen that the proposed method consistently achieves the lowest reconstruction RMSEs across different sampling densities.

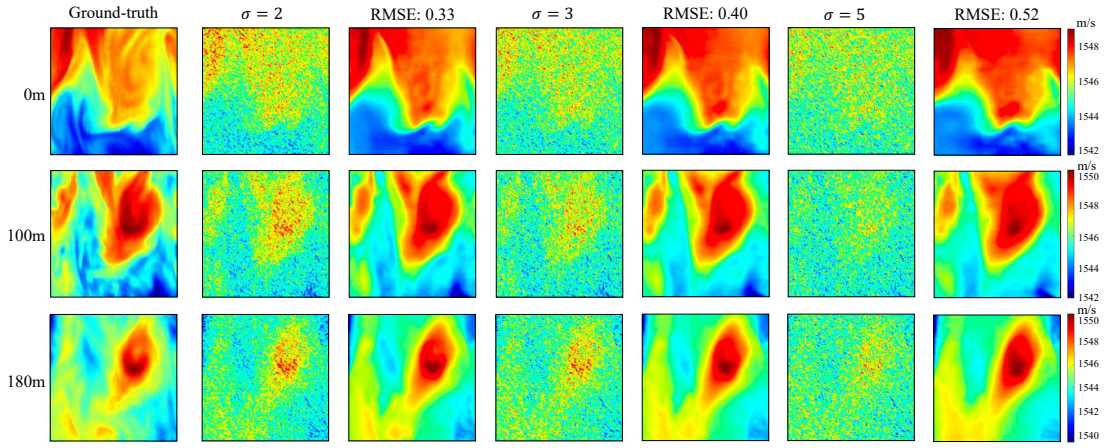



FIG. 15. Visual effects of the denoising results under different noise standard deviations $\sigma$. The first column is the ground-truth SSF data, followed by noisy data with different $\sigma$ and denoising results. The RMSEs are shown above the denoising results.

## 3. Running time and memory requirements

The running time of the PnP method on a computer equipped with an Intel i3 CPU with 4 cores is approximately 341 seconds. In our experiments, we observe that denoising with pre-trained image denoiser is the most time-consuming process. There are a couple of strategies that can be employed to expedite the running time. One option is to use a more advanced CPU with higher computational capabilities, which can speed up the computation. Alternatively, utilizing a GPU for the denoising step can significantly reduce the running time since GPUs are well-suited for parallel processing tasks like convolutions. These strategies can effectively enhance the overall efficiency of the PnP method.

The memory requirements of the proposed method primarily depend on the size of the pre-trained network and the memory needed during computation. According to our experiments,

TABLE V. RMSEs of different algorithms under different sampling ratios (*Regular2* sampling pattern, $\sigma = 0.3$).

| $\rho$ | **BMCG** | **GPR** | **LRTC** | **LRTC-TV** | **TensorNN** | **PnP** |
|---|---|---|---|---|---|---|
| 0.15 | 0.35 | 0.32 | 1.49 | 1.05 | 0.47 | **0.27** |
| 0.21 | 0.29 | 0.27 | 1.38 | 0.78 | 0.35 | **0.22** |

we have found that a personal laptop equipped with 8GB of RAM is sufficient to meet these requirements and ensure the smooth execution of the method.

## References


[1]Z.-H. Michalopoulou, P. Gerstoft, and D. Caviedes-Nozal, "Matched field source localization with gaussian processes," JASA Express Letters **1**(6), 064801 (2021).

[2]C.-X. Li, W. Xu, J.-L. Li, and X.-Y. Gong, "Time-reversal detection of multidimensional signals in underwater acoustics," IEEE Journal of Oceanic Engineering **36**(1), 60–70 (2011).

[3]F. Qu, X. Nie, and W. Xu, "A two-stage approach for the estimation of doubly spread acoustic channels," IEEE Journal of Oceanic Engineering **40**(1), 131–143 (2014).

[4]D. R. DelBalzo, L. Winsett, and E. R. Rike, "Long-term, large-scale acoustic fluctuations in the ulleung basin," The Journal of the Acoustical Society of America **114**(4), 2375–2375 (2003).

[5]X. Ji and H. Zhao, "Three-dimensional sound speed inversion in south china sea using ocean acoustic tomography combined with pressure inverted echo sounders," in *Global Oceans 2020: Singapore – U.S. Gulf Coast* (2020), pp. 1–6, doi: mu10.1109/IEEECONF38699.2020.9389208.

[6]L. Cheng, X. Ji, H. Zhao, J. Li, and W. Xu, "Tensor-based basis function learning for three-dimensional sound speed fields," The Journal of the Acoustical Society of America **151**(1), 269–285 (2022).

[7]P. Chen, L. Cheng, T. Zhang, H. Zhao, and J. Li, "Tensor dictionary learning for representing three-dimensional sound speed fields," The Journal of the Acoustical Society of America **152**(5), 2601–2616 (2022).

[8]S. Li, L. Cheng, T. Zhang, H. Zhao, and J. Li, "Graph-guided bayesian matrix completion for ocean sound speed field reconstruction," The Journal of the Acoustical Society of America **153**(1), 689–710 (2023).

[9]S. Li, L. Cheng, T. Zhang, H. Zhao, and J. Li, "Striking the right balance: Three-dimensional ocean sound speed field reconstruction using tensor neural networks," The Journal of the Acoustical Society of America **154**(2), 1106–1123 (2023).

[10]G. Zhang, X. Fu, J. Wang, X.-L. Zhao, and M. Hong, "Spectrum cartography via coupled block-term tensor decomposition," IEEE Transactions on Signal Processing **68**, 3660–3675 (2020).

[11]M. Ding, X. Fu, T.-Z. Huang, J. Wang, and X.-L. Zhao, "Hyperspectral super-resolution via interpretable block-term tensor modeling," IEEE Journal of Selected Topics in Signal

Processing **15**(3), 641–656 (2020).

[12]H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of inverse problems*, Vol. 375 (Springer Science & Business Media, 1996).

[13]Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," Advances in neural information processing systems **32** (2019).

[14]J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," Advances in Neural Information Processing Systems **33**, 6840–6851 (2020).

[15]K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte, "Plug-and-play image restoration with deep denoiser prior," IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(10), 6360–6376 (2021).

[16]U. S. Kamilov, C. A. Bouman, G. T. Buzzard, and B. Wohlberg, "Plug-and-play methods for integrating physical and learned models in computational imaging: Theory, algorithms, and applications," IEEE Signal Processing Magazine **40**(1), 85–97 (2023) doi: mu10.1109/MSP.2022.3199595.

[17]D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 9446–9454.

[18]X. Hua, L. Cheng, T. Zhang, and J. Li, "Interpretable deep dictionary learning for sound speed profiles with uncertainties," The Journal of the Acoustical Society of America **153**(2), 877–894 (2023).

[19]V. Saragadam, R. Balestriero, A. Veeraraghavan, and R. G. Baraniuk, "Deeptensor: Low-rank tensor decomposition with deep network priors," arXiv preprint arXiv:2204.03145

(2022).

[20]R. Heckel and P. Hand, "Deep decoder: Concise image representations from untrained non-convolutional networks," arXiv preprint arXiv:1810.03982 (2018).

[21]S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *2013 IEEE Global Conference on Signal and Information Processing*, IEEE (2013), pp. 945–948.

[22]U. S. Kamilov, H. Mansour, and B. Wohlberg, "A plug-and-play priors approach for solving nonlinear imaging inverse problems," IEEE Signal Processing Letters **24**(12), 1872–1876 (2017).

[23]C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis with a new tensor nuclear norm," IEEE transactions on pattern analysis and machine intelligence **42**(4), 925–938 (2019).

[24]J. Nocedal and S. J. Wright, *Numerical optimization* (Springer, 1999).

[25]K. Zhang, W. Zuo, and L. Zhang, "Ffdnet: Toward a fast and flexible solution for cnn-based image denoising," IEEE Transactions on Image Processing **27**(9), 4608–4622 (2018).

[26]J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, Ieee (2009), pp. 248–255.

[27]K. Ma, Z. Duanmu, Q. Wu, Z. Wang, H. Yong, H. Li, and L. Zhang, "Waterloo exploration database: New challenges for image quality assessment models," IEEE Transactions on Image Processing **26**(2), 1004–1016 (2016).

[28]D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001* (2001), Vol. 2, pp. 416–423 vol.2, doi: mu10.1109/ICCV.2001.937655.

[29]J. Liu, Y. Sun, C. Eldeniz, W. Gan, H. An, and U. S. Kamilov, "Rare: Image reconstruction using deep priors learned without groundtruth," IEEE Journal of Selected Topics in Signal Processing **14**(6), 1088–1099 (2020).

[30]C. J. Pellizzari, M. F. Spencer, and C. A. Bouman, "Coherent plug-and-play: digital holographic imaging through atmospheric turbulence using model-based iterative reconstruction and convolutional neural networks," IEEE Transactions on Computational Imaging **6**, 1607–1621 (2020).

[31]Z. Wu, Y. Sun, A. Matlock, J. Liu, L. Tian, and U. S. Kamilov, "Simba: Scalable inversion in optical tomography using deep denoising priors," IEEE Journal of Selected Topics in Signal Processing **14**(6), 1163–1175 (2020).

[32]Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer, "Novel methods for multilinear data completion and de-noising based on tensor-svd," in *Proceedings of the IEEE conference on computer vision and pattern recognition* (2014), pp. 3842–3849.

[33]F. Jiang, X.-Y. Liu, H. Lu, and R. Shen, "Anisotropic total variation regularized low-rank tensor completion based on tensor nuclear norm for color image inpainting," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE (2018), pp. 1363–1367.

[34]C. E. Rasmussen and C. K. Williams, "Gaussian processes in machine learning," Lecture notes in computer science **3176**, 63–71 (2004).

[35]D. Caviedes-Nozal, N. A. Riis, F. M. Heuchel, J. Brunskog, P. Gerstoft, and E. Fernandez-Grande, "Gaussian processes for sound field reconstruction," The Journal of the Acoustical Society of America **149**(2), 1107–1119 (2021).

[36]C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, Vol. 4 (Springer, 2006).

[37]J. Yang, X. Yang, X. Ye, and C. Hou, "Reconstruction of structurally-incomplete matrices with reweighted low-rank and sparsity priors," IEEE Transactions on Image Processing **26**(3), 1158–1172 (2016).

[38]J. Yang, Y. Zhu, K. Li, J. Yang, and C. Hou, "Tensor completion from structurally-missing entries by low-tt-rankness and fiber-wise sparsity," IEEE Journal of Selected Topics in Signal Processing **12**(6), 1420–1434 (2018).

[39]A. G. Wilson, Z. Hu, R. R. Salakhutdinov, and E. P. Xing, "Stochastic variational deep kernel learning," Advances in neural information processing systems **29** (2016).

[40]L. Xu, Y. Dai, J. Zhang, C. Zhang, and F. Yin, "Exact o (n 2) hyper-parameter optimization for gaussian process regression," in *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*, IEEE (2020), pp. 1–6.