Sparse Prior-Guided Deep Learning for OTFS Channel Estimation

Xiaoqi Zhang, Chang Liu, *Member, IEEE*, Weijie Yuan, *Member, IEEE*,
J. Andrew Zhang, *Senior Member, IEEE*, and Derrick Wing Kwan Ng, *Fellow, IEEE*

***Abstract*—In this paper, we propose a deep residual shrinkage neural network (DRSNN) based on sparse prior for solving the orthogonal time frequency space (OTFS) channel estimation problem. Specifically, w e f ormulate t he p roblem a s a denoising task and subsequently develop a residual learning-based denoiser to adeptly learn the residual noise. To harness the sparsity of the channel in the delay-Doppler (DD) domain, we insert proximal mapping as a new layer into the deep network, which directly and explicitly produces well-regularized outputs. In particular, the layer is implemented by a trainable soft shrinkage function with a threshold vector, where the thresholds can be adapted by a dedicated sub-network. Moreover, we derive a mathematical expression for the proposed network and conduct a thorough analysis exploiting Bayesian philosophy, which provides valuable insights into the interpretability of neural networks. Finally, simulation results demonstrate the superiority of the proposed method for sparse channel recovery in terms of both estimation performance and computational complexity.**

***Index Terms*—Deep learning, OTFS, channel estimation, Bayesian statistics, regularization.**

I. Introduction

Orthogonal time frequency space (OTFS) modulation emerges as a potential solution for satisfying the heterogeneous requirements of next-generation wireless communication systems, due to its excellent performance in high-mobility environments [1], [2]. To fully exploit the advantages of OTFS modulation, accurate estimation of channel state information (CSI) is of great importance. A threshold-based method was proposed to estimate CSI in [3], where a single pilot impulse with guarding zero symbols are employed in the DD domain at the transmitter. To exploit the structured sparsity of effective Delay-Doppler-angle domain channel, a structured orthogonal matching pursuit algorithm (OMP)-based channel estimation scheme for massive multiple-input multiple-output (MIMO)-OTFS systems was proposed [4]. Among various existing estimation methods, sparse Bayesian learning (SBL), formulated in a Bayesian framework, further exploits sparsity information of wireless channels and has been widely adopted for OTFS channel estimation [5], [6]. In particular, theoretical results demonstrated that SBL can be treated as a specific form of *maximum a posteriori* (MAP) estimation when a Laplace prior is applied to the channel model [7]. However, despite these advancements, the processing delay inherent in iterative algorithms and the need for rigorous initialization parameters still hinder the practical implementation of these methods [7], especially in high-dimensional signal recovery applications.

Recently, thanks to the remarkable data-driven capability and the offline training mechanism of neural networks, deep learning (DL) has presented a promising prospect of designing efficient and low-complexity algorithms for online estimation. In particular, completely data-driven approaches directly map input features of the network to the estimated results. Benefiting from a large training dataset, they exhibit superior performance over conventional iterative algorithms derived from classic optimization methods, especially in complicated applications [8], [9]. However, unlike the latter that have a clear or explicit problem statement and can incorporate prior knowledge into the solution, completely data-driven networks often lack sufficient interpretation and require a large number of parameters to learn a specific function mapping for the dedicated tasks.

To circumvent the above challenges, researchers have designed various regularization methods to incorporate prior knowledge into data-driven networks. For instance, the authors in [10] introduced extra terms in the objective function to impose sparse constraints on hidden layers, thereby enhancing network performance. Meanwhile, a deep network with total variation (TV) regularization was designed for image recovery, improving the training robustness and interoperability [11]. These regularization methods facilitate a versatile fusion of model priors with neural network architectures, promoting interpretable, generalizable, and high-performance neural networks.

Despite their rich potential, the adoption of these promising regularization methods for wireless communications is still in its infancy. In this work, we propose a deep residual shrinkage neural network (DRSNN) that adopts a more effective regularization method by integrating a proximal mapping layer based on the model prior into the deep network [12]. The proposed DRSNN leverages the powerful data-driven capability, the offline training mechanism, and the model

This work is supported in part by National Natural Science Foundation of China under Grant 62471208, in part by Guangdong Provincial Natural Science Foundation under Grant 2022A1515011257 and 2024A151510098, in part by Shenzhen Science and Technology Program under Grant JCYJ20220530114412029, and in part by Shenzhen Key Laboratory of Robotics and Computer Vision under Grant ZDSYS20220330160557001, in part by La Trobe University Research Project under Grant 325011150.

X. Zhang and J. Andrew Zhang are with the School of Electrical and Data Engineering, University of Technology Sydney, Sydney 2007, Australia (e-mail: Xiaoqi.Zhang@student.uts.edu.au; andrew.zhang@uts.edu.au)

C. Liu is with the Department of Computer Science and Information Technology, La Trobe University, Melbourne, Victoria 3086, Australia (e-mail: C.Liu6@latrobe.edu.au).

W. Yuan is with the School of System Design and Intelligent Manufacturing and the Shenzhen Key Laboratory of Robotics and Computer Vision, Southern University of Science and Technology, Shenzhen 518055, China. (e-mail: yuanwj@sustech.edu.cn).

D. W. K. Ng is with the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, NSW 2052, Australia (e-mail: w.k.ng@unsw.edu.au).

prior, thereby enhancing channel estimation performance with low computational complexity. The main contributions of this paper are summarized as follows:

- We approach the task of OTFS channel estimation by framing it as a denoising problem utilizing the typical least square (LS) estimator and employ a convolutional layers-based denoiser with a residual structure to implicitly learn the noise. Furthermore, by exploiting the sparse characteristic of the channel, we design an optimizable soft shrinkage function based on proximal mapping techniques to generate expected sparse outputs, thereby significantly improving the estimation performance.
- To provide deeper insights, we also derive a mathematical expression for the proposed estimator to theoretically explain the rationale of the proposed algorithm within the framework of Bayesian statistics.
- Finally, simulation results show that our proposed DRSNN algorithm outperforms other channel estimation approaches in terms of both estimation performance and computational complexity.

The rest of this paper is organized as follows. Section II introduces the OTFS system model and the SBL-based solution. In Section III, we model the channel estimation as a denoising problem and develop a DRSNN-based algorithm. Our simulation results are provided in Section IV, while Section V concludes this paper.

Notations: Superscript H indicates the conjugate transpose. $\mathcal{CN}(\mathbf{a}, \mathbf{B})$ denotes the circularly symmetric complex Gaussian (CSCG) distribution with $\mathbf{a}$ being the mean vector and $\mathbf{B}$ being the covariance matrix. $\odot$ represents element-wise multiplication. $p(\cdot)$ represents the probability density function (PDF). $\Re(\cdot)$ and $\Im(\cdot)$ are real and imaginary operations, respectively. $[\cdot]_L$ and $|\cdot|_1$ denote modulo operations for L and ℓ_1 norms of a scalar, respectively. The function $\mathrm{sign}(\cdot)$ denotes the sign function that returns the sign of a real number and $E(\cdot)$ represents the statistical expectation. In addition, $\mathbb{1}_{\varpi}$ represents the indicator function of an event ϖ.

II. System Model and SBL-based Solution

A. OTFS Modulation

In OTFS modulation, the information symbols $\{x[k,l], k = 0, \cdots, N-1, l = 0, \cdots, M-1\}$ from a constellation set $\mathbb{A} = \{\chi_1, \cdots, \chi_q\}$ of size q are placed in the delay-Doppler (DD) domain. Here, M and N represent the numbers of subcarriers and time slots, respectively. The effective channel impulse in the DD domain can be written as

$$h(\tau, \nu) = \sum_{i}^{P} h_i \delta(\tau - \tau_i)\delta(\nu - \nu_i), \tag{1}$$

where P and $\delta(\cdot)$ are the total number of propagation paths and the Dirac delta function, respectively. The terms h_i, τ_i, and ν_i denote the complex channel coefficient, delay, and Doppler shift associated with the i-th path, respectively. The delay and Doppler taps for the i-th path are defined as $\tau_i = \frac{l_i}{M\Delta f}$ and $\nu_i = \frac{k_i + \kappa_i}{NT}$, respectively, where Δf represents subcarrier spacing, T denotes symbol period, $l_i \in [0, M-1]$ and $k_i \in [0, N-1]$ are integers, and $\kappa_i \in [-0.5, 0.5]$ represents fractional Doppler shift for the i-th path. Generally, the typical value of the sampling time $\frac{1}{M\Delta f}$ in the delay domain is sufficiently small. Therefore, the impact of fractional delays in typical wideband systems can be neglected [2].

In this work, we employ the pilot design proposed scheme in [4], where pilot symbols with size $P_m \times P_n$ are embedded in the DD domain with size $M \times N$. To avoid data interference, a guard space with zero symbols is inserted between the pilot symbols and data symbols. When the transmitted pulse waveform $g_{\mathrm{tx}}(t)$ and the received waveform $g_{\mathrm{rx}}(t)$ are both bi-orthogonal [6], the received signal $y[k,l]$ can be expressed as [5]

$$y[k,l] = \sum_{l'=0}^{l_{\max}} \sum_{k'=-k_{\max}}^{k_{\max}} h[l',k'] e^{-j2\pi \frac{l'(k+\kappa_{k'})}{MN}} \sum_{q=-Q}^{Q} \eta(q, \kappa_{k'}) x[[k-k'+q]_N, [l-l']_M] + v[k,l], \tag{2}$$

where $\eta(q, \kappa_{k'}) = \frac{1 - e^{-j2\pi(-q-\kappa_{k'})}}{N - Ne^{-j\frac{2\pi}{N}(-q-\kappa_{k'})}}$, $l_{\max}$ and $k_{\max}$ represent the maximum on-grid delay and Doppler shift, respectively. The terms $h[l',k']$ and Q are the effective channel impulse and number of paths. For the convenience of analysis, we model $v[k,l]$ as an additive CSCG noise following $\mathcal{CN}(0, \sigma_v^2)$. To facilitate the channel estimation problem, we reformulate (2) in a matrix form as [13]

$$\mathbf{y} = \mathbf{X}\mathbf{h} + \mathbf{v}, \tag{3}$$

where $\mathbf{y}, \mathbf{v} \in \mathbb{C}^{L_m \times 1}$ with $L_m = (P_m + l_{\max})(P_n + 2k_{\max} + 2Q)$. The pilot matrix $\mathbf{X} \in \mathbb{C}^{L_m \times L_n}$ is derived from the pilot symbols, where $L_n = (l_{\max}+1)(2k_{\max}+2Q+1)$. Our goal is to accurately reconstruct the channel coefficient $\mathbf{h} \in \mathbb{C}^{L_n \times 1}$ based on $\mathbf{y}$ and $\mathbf{X}$. As for the vector $\mathbf{h} = [h_1, \cdots, h_{L_n}]$, the support of $\mathbf{h}$ is defined to be the set $\mathrm{supp}(\mathbf{h}) = \{i : h_i \neq 0\}$ and $\mathbf{h}$ is *k-sparse* if $|\mathrm{supp}(\mathbf{h})| \leq k$.

B. Sparse Bayesian Learning-based Solution

In the classic Bayesian modeling framework, all unknowns are treated as stochastic variables characterized by specific probability distributions. As a result, the joint distribution $p(\mathbf{y}, \mathbf{h}, \sigma_v^2, \boldsymbol{\zeta})$ of all unknown and observed variables can be factorized as

$$p(\mathbf{y}, \mathbf{h}, \sigma_v, \boldsymbol{\zeta}) = p(\mathbf{y}|\mathbf{h}, \sigma_v^2) p(\mathbf{h}|\boldsymbol{\zeta}) p(\boldsymbol{\zeta}). \tag{4}$$

Here, $\boldsymbol{\zeta}$ denotes the *hyper* parameters. It is proved that all entries h_i in the OTFS channel share the common Laplace sparse prior $p_{\mathrm{L}}(h_i)$, which is governed by $\zeta \in [0, +\infty]$, with [14]

$$p(h_i) = p_{\mathrm{L}}(h_i) = p(h_i|\zeta) = \frac{\zeta}{2} e^{(-\frac{\zeta}{2}|h_i|_1)}. \tag{5}$$

The Laplacian prior is equivalent to a two-level hierarchical model, which can be expressed as

$$p_{\mathrm{L}}(h_i) = \mathcal{CN}(h_i|0, \gamma_i) p(\gamma_i|\zeta), \tag{6}$$

where $p(\gamma_i|\zeta) = \frac{\zeta}{2} \exp\{-\frac{\zeta}{2}\gamma_i\}$, for $\gamma_i \geq 0$, is an exponential *hyper* prior. The term γ_i denotes the i-th, $i \in \{1, 2, \cdots, L_n\}$ sample variance that controls the sparsity of sample h_i, i.e., $\gamma_i \to 0$ resulting in $h_i \to 0$.

Given fixed values of the sample variance vector $\boldsymbol{\gamma} = (\gamma_1, \cdots, \gamma_n)$ and noise variance σ_v^2, the SBL-based solution can be written as [15]

$$\hat{\mathbf{h}}_{\mathrm{SBL}} = \sigma_v^{-2} \left(\sigma_v^{-2}\mathbf{X}^H\mathbf{X} + \boldsymbol{\Gamma}\right)^{-1} \mathbf{X}^H\mathbf{y}, \tag{7}$$

where $\boldsymbol{\Gamma} = \mathrm{diag}(\gamma_1, \gamma_2, \cdots, \gamma_n)$ is the covariance matrix of $\mathbf{h}$. To acquire optimal $\boldsymbol{\Gamma}$ and σ_v^2, the iterative expectation maximization (EM) algorithm is typically applied. However, the algorithm entails exceedingly high computational complexity that hinders its practiced implementation [5].

III. Proposed Method

In this section, we develop a DRSNN method to learn the residual noise from the noisy pilot signals. In contrast to existing completely data-based methods, we specifically design denoising and proximal mapping-based regularization blocks for learning the residual noise and generating explicitly regularized channel coefficients. This approach can achieve significantly reduced complexity and enhanced estimation performance.

A. Structure of Neural Network

As shown in Fig. 1(a), we design T denoising blocks and a sparse prior-based regularization module to generate sparse output. All hyperparameters of the neural network are summarized in TABLE I. Here, the filter size is $f_n \times (f_w \times f_h \times f_c)$, where f_n is the numbers of convolution kernels, f_w and f_h are the spatial width and height of the kernel, respectively, and f_c represents the channel numbers of input data.

*1) **Input Layer**:* We first adopt an LS estimator to obtain a coarse estimate

$$\hat{\mathbf{h}}_{\mathrm{LS}} = \mathbf{X}^{\dagger}\mathbf{y} = \mathbf{h} + \hat{\mathbf{v}}, \tag{8}$$

where $\mathbf{X}^{\dagger} = \mathbf{X}^H(\mathbf{X}\mathbf{X}^H)^{-1}$ and $\hat{\mathbf{v}}$ represent the pseudo-inverse of $\mathbf{X}$ and the additive noise in the DD domain, respectively. Specifically, the inverse fast Fourier transform (IFFT) is adopted to realize the LS method, reducing the computational complexity [8]. To effectively deal with complex-valued data, we adopt two neural network channels for the real and imaginary parts of the noisy channel matrix, which are stacked to form a real-valued two-dimensional tensor $\boldsymbol{\Xi}_0 \in \mathbb{R}^{L_m \times 2}$ as [2]

$$\boldsymbol{\Xi}_0 = [\Re(\hat{\mathbf{h}}_{\mathrm{LS}}), \Im(\hat{\mathbf{h}}_{\mathrm{LS}})]. \tag{9}$$

*2) **Denoising Module**:* According to (8), the channel estimation can be regarded as a denoising problem when the initial coarse estimate is based on the LS-based channel estimation. Consequently, we exploit T denoising blocks to gradually enhance the recovery performance and all blocks adopt an identical structure. Each denoising block consists of N_l layers, as depicted in Fig 1(b). For the first layer, the 1D convolution, the batch normalization, and the rectified linear unit (ReLU) denoted by "Conv+BN+ReLU" are exploited to extract spatial features from the input, and a single convolution is used to obtain the residual noise matrix in the last layer. In particular, a denoising block consists of a residual sub-network and an element-wise subtraction operator. Accordingly, the output of the denoising module $\hat{\mathbf{h}}_{\mathrm{Dn}}$ can be expressed as

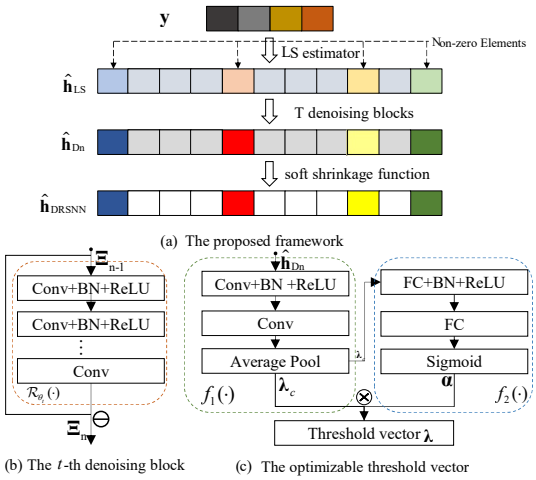


Fig. 1. The proposed DL-based approach.

$$\hat{\mathbf{h}}_{\mathrm{Dn}} = \boldsymbol{\Xi}_0 - \sum_{t=1}^{T} \left(\mathcal{R}_{\theta_{t-1}}(\boldsymbol{\Xi}_{t-1})\right), \tag{10}$$

where $\mathcal{R}_{\theta_{t-1}}(\cdot)$ represents the function of the t-th residual sub-network with network parameters θ. The terms $\boldsymbol{\Xi}_{t-1}$ and $\boldsymbol{\Xi}_t$ denote the input and output of the t-th denoising block, respectively.

*3) **Sparse Prior-based Regularization Module**:* To formulate the optimization of the soft shrinkage function, two sub-networks are exploited to generate adaptive thresholds. In the first sub-module, a pooling operation is adopted to generate a coarse estimated threshold vector $\boldsymbol{\lambda}_c \in \mathbb{R}^{1\times 2}$ from "Conv+BN+ReLU+Conv" operations, which is given by

$$\boldsymbol{\lambda}_c = f_1(\hat{\mathbf{h}}_{\mathrm{Dn}}), \tag{11}$$

where $f_1(\cdot)$ denotes the sub-network as shown in Fig. 1(c). To refine the thresholds, the second sub-module is then applied to get a scaling factor

$$\boldsymbol{\alpha} = f_2(\boldsymbol{\lambda}_c) = f_2(f_1(\hat{\mathbf{h}}_{\mathrm{Dn}})), \tag{12}$$

where $f_2(\cdot)$ denotes the second sub-network as shown in Fig. 1(c). In addition, a sigmoid function is used at the end of the fully connected (FC) network to limit the range of $\boldsymbol{\alpha}$. Finally, the adaptive thresholds can be expressed as $\boldsymbol{\lambda} = \boldsymbol{\alpha} \odot \boldsymbol{\lambda}_c$. Based on the threshold vector $\boldsymbol{\lambda}$, the soft shrinkage-based layer is adopted at the output of DRSNN, which is given by

$$\hat{\mathbf{h}}_{\mathrm{DRSNN}} = \mathcal{L}_{\lambda}(\hat{\mathbf{h}}_{\mathrm{Dn}}) = \mathrm{sign}(\boldsymbol{\Xi}_T)\max\{\mathrm{abs}(\boldsymbol{\Xi}_T) - \mathbf{1}_m\boldsymbol{\lambda}, 0\}, \tag{13}$$

where $\mathcal{L}_{\lambda}(\cdot)$ and $\mathrm{sign}(\cdot)$ represents the soft shrinkage function with the adaptive thresholds and a function that returns the sign of the variable, respectively.

B. Training Process

The training dataset of the network is obtained as follows

$$(\mathcal{Y}, \mathcal{H}) = \{(\mathbf{y}_1, \mathbf{h}_1), \cdots, (\mathbf{y}_{N_s}, \mathbf{h}_{N_s})\}, \tag{14}$$

Input Layer: The noisy input with the size of $L_m \times 2$		
Denoising Module: It has T identical denoising blocks		
Layers	**Operation**	**Filter Size** ($f_n \times f_w \times f_h \times f_c$)
1	Conv + BN + ReLU	$32 \times (3 \times 1 \times 2)$
$2 \sim (N_l - 1)$	Conv + BN + ReLU	$32 \times (3 \times 1 \times 32)$
N_l	Conv	$2 \times (3 \times 1 \times 32)$
Sparse Prior-based Regularization Module: Eliminating zero-elements		
Layers	**Operation**	**Hybrid Parameters**
1	Conv + BN + ReLU + Conv	$2 \times (3 \times 1 \times 2)$
2	Global Average Pool	$L_m \times 2 \mapsto 1 \times 2$
3	FC + BN + ReLU + FC + Sigmoid	$1 \times 2 \mapsto 1 \times 2$
Output Layer: The output ($\mathcal{L}_\lambda(\hat{\mathbf{h}}_{\rm Dn})$) with the size of $L_m \times 2$		

TABLE I: Hyperparameters of the proposed DRSNN

where $(\mathbf{y}_i, \mathbf{h}_i)$ and N_s denote the i-th training example of $(\mathcal{Y}, \mathcal{H})$ and sample size, respectively. At the output, the cost function of the offline training phase can be expressed as

$$J_{\rm MSE}(\theta) = \frac{1}{N_s}\sum_{i=1}^{N_s} \left\| \mathcal{U}\left(\mathcal{L}_\lambda(f_\theta(\mathbf{y}_i))\right) - \mathbf{h}_i \right\|_2, \tag{15}$$

where $\mathcal{U}(\cdot) : \mathbb{R}^{N_s \times 2} \mapsto \mathbb{C}^{N_s \times 1}$ denotes a mapping function that converts a two-column real matrix to a complex vector. Finally, the back propagation (BP) algorithm is exploited to progressively update the network parameters to generate the well-trained DRSNN, i.e.,

$$\hat{\mathbf{h}}_{\rm DRSNN} = \mathcal{U}\left(\mathcal{L}_\lambda(f_{\theta^*}(\boldsymbol{\Xi}_0))\right), \tag{16}$$

where $f_{\theta^*}(\cdot)$ denotes the well-trained network with the optimal network parameters θ^*.

C. Qualitative Characterization of DRSNN

In this section, we analyze and qualitatively characterize the properties of the proposed DRSNN, offering more insights by comparing it with the SBL algorithm [14]. Let us consider the training phase of a neural network $\Phi_{\boldsymbol{\theta}}(\cdot)$ with weight parameters $\boldsymbol{\theta}$, i.e.,

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \left[J(\Phi_{\boldsymbol{\theta}}(\cdot)) + \lambda R(\cdot) \right], \tag{17}$$

where $J(\cdot)$ denotes the error function and $R(\cdot)$ is a regularization function weighted by $\lambda > 0$ that is associated with the prior distribution of the parameters, respectively. Note that there are no regularization constraints on the weight parameters in completely data-driven networks. In this work, we formulate a deterministic prior based on the OTFS channel and integrate it directly into the final layer using proximal mapping, ensuring that the output of network $\mathbf{h}_{\rm DRSNN}$ aligns closely with the specified prior $R(\cdot)$, while maintaining vicinity to $\mathbf{h}_{\rm Dn}$. The proximal mapping $\mathrm{prox}(\cdot)$ is defined as [16]

$$\mathrm{prox}(\mathbf{h}_{\rm Dn}) = \arg\min_{\mathbf{z}} \{\lambda R(\mathbf{z}) + \|\mathbf{z} - \mathbf{h}_{\rm Dn}\|_2^2\}, \tag{18}$$

where $R(\mathbf{z}) = \|\mathbf{z}\|_1$ is a ℓ_1-norm constraint. In this case, the output $\mathbf{h}_{\rm DRSNN}$ can be written as

$$\mathbf{h}_{\rm DRSNN} = \mathrm{prox}(\mathbf{h}_{\rm Dn}) = \mathcal{L}_\lambda(\mathbf{h}_{\rm Dn}). \tag{19}$$

In particular, the weighting parameter λ is generated by a lightweight neural network. As stated in [12], this handicraft structure allows upstream layers to approach the weight values by themselves, thereby producing well-regularized output. Therefore, we denote $\mathbf{h}_{\rm Dn} = f_{\theta^*}(\cdot)$ and the $\mathbf{h}_{\rm DRSNN} = f_{\Omega(\theta^*)}(\cdot)$, with $\Omega(\cdot)$ indicating the regularization of weight values for upstream layers based on $R(\cdot)$.

Next, let us examine the whole structure of the proposed DRSNN. The convolution operation of the network can be expressed as the production of two matrices. Therefore, for each denoising block, we have

$$\mathcal{R}_{\theta_t}(\boldsymbol{\Xi}_{t-1}) = \boldsymbol{\Xi}_{t-1}\boldsymbol{\Theta}_t, \tag{20}$$

where $\boldsymbol{\Theta}_t$ represents the network weights at the t-th block to be optimized. Note that although the output of the d-th residual subnetwork can be expressed as the product of two matrices, the elements in $\boldsymbol{\Theta}$ are obtained through the non-linear operation, including the activation function, and the residual subnetwork is still a non-linear network [17]. Consequently, the expression of the denoising module can be expressed as

$$\begin{aligned}\hat{\mathbf{h}}_{\rm Dn} &= \boldsymbol{\Xi}_0 - \sum_{t=1}^{T} \mathcal{R}_{\theta_t}\left(\boldsymbol{\Xi}_{t-1}\right) \\ &= \boldsymbol{\Xi}_0 - \boldsymbol{\Xi}_0\boldsymbol{\Theta}_1 - \sum_{t=2}^{T}\prod_{i=1}^{t-1} \boldsymbol{\Xi}_0\tilde{\boldsymbol{\Theta}}_i\boldsymbol{\Theta}_t = \boldsymbol{\Xi}_0 - \boldsymbol{\Xi}_0\boldsymbol{\Theta}, \end{aligned} \tag{21}$$

where $\boldsymbol{\Theta} = \boldsymbol{\Theta}_1 + \mathbb{1}_{\varpi}(T)\sum_{t=2}^{T}\prod_{i=1}^{t-1}\tilde{\boldsymbol{\Theta}}_i\boldsymbol{\Theta}_t$. Here, we have the event $\varpi = \{T | T \geq 2, T \in \mathbb{Z}\}$ and $\tilde{\boldsymbol{\Theta}}_i = \mathbf{I}_t - \boldsymbol{\Theta}_i$. Given the input from the LS estimator, i.e., $\boldsymbol{\Xi}_0 = \mathbf{h}_{\rm LS} = \mathbf{X}^\dagger\mathbf{h}$, the well-trained model can be formulated as

$$\hat{\mathbf{h}}_{\rm Dn} = \boldsymbol{\Xi}_T = f_\theta(\mathbf{h}_{\rm LS}) = \mathbf{h}_{\rm LS}(\mathbf{I}_t - \boldsymbol{\Theta}^*), \tag{22}$$

where $\boldsymbol{\Xi}_T$ and $\boldsymbol{\Theta}^*$ denote the output of the denoising module and the weighted matrix obtained from parameter θ^*, respectively. Based on the handicraft regularization, the output of the neural network can be expressed as

$$\hat{\mathbf{h}}_{\rm DRSNN} = \mathrm{prox}(\mathbf{h}_{\rm Dn}) = \mathbf{h}_{\rm LS}(\mathbf{I}_t - \Omega(\boldsymbol{\Theta}^*)). \tag{23}$$

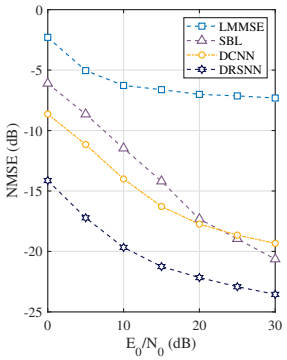
On the other hand, the solution (7) can be rewritten as [18]

$$\hat{\mathbf{h}}_{\rm SBL} = \hat{\mathbf{h}}_{\rm LS}\left(\mathbf{I}_t - \boldsymbol{\Gamma}_{\rm opt}^{-1}(\boldsymbol{\Gamma}_{\rm opt}^{-1} + \sigma_{\rm opt}^{-2}\mathbf{I}_t)^{-1}\right), \tag{24}$$

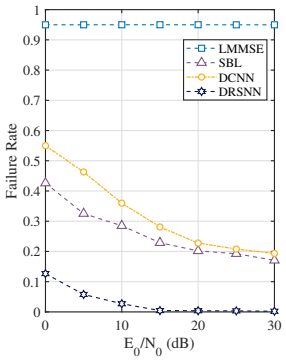
where $\mathbf{I}_t \in \mathbb{C}^{L_n \times L_n}$ denotes the identity matrix. By comparing (23) and (24), one can deduce that the output of the DRSNN is equivalent to the SBL estimator when $\Omega(\boldsymbol{\Theta}^*) = \boldsymbol{\Gamma}_{\rm opt}^{-1}(\boldsymbol{\Gamma}_{\rm opt}^{-1} + \sigma_{\rm opt}^{-2}\mathbf{I}_t)^{-1}$ is adopted. The DRSNN is capable of perfectly approximating the SBL estimator due to its learning capabilities [17]. Furthermore, we can see that the proposed DRSNN is a generalized regularization-based framework that subsumes an SBL estimator in a data-driven manner as a subcase.

IV. Simulation Results

In this section, we carry out extensive simulations to illustrate the effectiveness of our proposed algorithm. The system parameters are as follows. Unless otherwise specified, we set an OTFS frame with $M = 24$ subcarriers and $N = 20$ time slots in the time-frequency domain. The total number of paths is set as $P = 4$, with a normalized maximum Doppler shift of $k_{\max} = 4$ and a normalized maximum delay shift of $l_{\max} = 6$. The channel coefficients are generated by using a complex Gaussian distribution $\mathcal{CN}(0, \frac{1}{P})$. Moreover, the

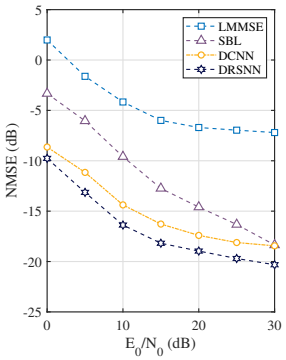


(a) NMSE performance.

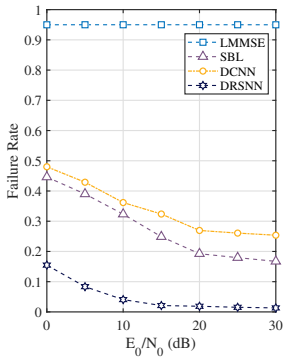


(b) Failure Rate performance.

Fig. 2. Peroformance comparison for multiple schemes with AWGN.



(a) NMSE performance.



(b) Failure Rate performance.

Fig. 3. Performance comparison for multiple schemes with t-distribution noise.

sparsity k is determined by $k = \frac{PQ}{L_n}$. The carrier frequency and subcarrier spacing are set to 3 GHz and 7.5 kHz, respectively. The number of pilot symbols is set as 4. Data symbols are normalized by ensuring that $E|x[k,l]|^2 = 1$. The training set and test set consisted of $27,000$ and $3,000$ samples, respectively, both generated by the Monte-Carlo method. Each presented results is obtained from averaging over $3,000$ Monte-Carlo realizations. We execute the proposed method on a desktop computer equipped with a Xeon Gold 6226R 2.9 GHz processing unit (CPU) and an NVIDIA GeForce RTX 3090 graphic processing unit (GPU), and the online estimation time for the proposed method is only 0.43 milliseconds. Note that the Adam optimizer is adopted for network training, with the learning rate set to 0.01.

We compare the performance of our proposed DRSNN scheme with the classic linear minimum mean-square error (LMMSE) [18], SBL, and denoising convolution neural network (DCNN), i.e., the DRSNN without the regularization layer. We adopt the normalized MSE (NMSE) as the performance metric, which is defined as $\text{NMSE} = 10\log_{10}\frac{E(\|\hat{\mathbf{h}}-\mathbf{h}\|_2^2)}{E(\|\mathbf{h}\|_2^2)}$, where $\hat{\mathbf{h}}$ and $\mathbf{h}$ are the estimated and true channel coefficients, respectively. Besides, in order to evaluate the capability of recovering channel coefficients, we also adopt the failure rate [15] as a performance metric. Here, the term failure rate refers to the percentage of cases where estimated indices do not match the true indices. Note that estimates with absolute values below 10^{-3} are considered negligible and treated as zero.

We first study the performance of OTFS channel estimation under additive white Gaussian noise (AWGN). Fig. 2a presents the NMSE performance under various $\mathrm{E}_0/\mathrm{N}_0$ values. It can be observed that the LMMSE estimator performs poorly, without exploiting the sparse prior information. Furthermore, in the low SNR region, the DCNN outperforms LMMSE and SBL because of its strong feature extraction capabilities, whereas the SBL's performance is affected by the noise level and the requirement for accurate initial conditions. In contrast, the proposed DRSNN achieves the best performance among four methods at all $\mathrm{E}_0/\mathrm{N}_0$ values. This is because the DRSNN can leverage the regularization method to exploit the prior statistical knowledge of input data, thereby enhancing the estimation accuracy. As demonstrated in Fig. 2b, the failure rate values of the SBL, DCNN, and DRSNN methods decrease with the increasing $\mathrm{E}_0/\mathrm{N}_0$, and the DRSNN method can always achieve the best performance. The reason is that our proposed method can efficiently generate a well-regularized sparse output with the proximal mapping layer for improving the accuracy of channel estimation.

Furthermore, we investigate the impact of t-distribution noise on the NMSE and failure rate performance. As shown in Fig. 3a, the performances of the LMMSE and SBL estimators degrade significantly when AWGN is replaced by t-distribution noise, particularly in the low SNR regions. This is because the model assumptions do not align with the characteristics of t-distribution noise. In contrast, the DL-based methods, i.e., DCNN and DRSNN, exhibit better performance than the model-based methods since the neural network can learn the channel statistical features from a large number of LS-based training examples. Furthermore, the proposed method enhances estimation performance by employing a regularization technique that ensures the output closely aligns with the specified prior $R(\cdot)$. To further evaluate the estimation performance, we present the failure rate results under various $\mathrm{E}_0/\mathrm{N}_0$ values in Fig. 3b. It is evident that the LMMSE

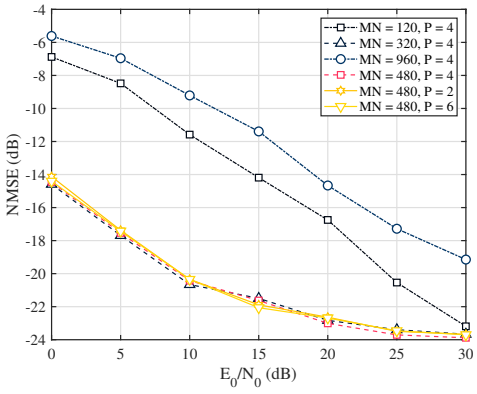


Fig. 4. NMSE performance with varying P and MN.

TABLE II: Computational Complexities of Different Estimation Algorithms

Algorithm	Online Estimation
LMMSE	$O(L_n^3)$
SBL	$O(kL_n^3)$
DCNN	$O(L_n \log L_n + TL_n \sum_i^{N_l} c_{i-1} s_i^2 c_i)$
DRSNN	$O(L_n \log L_n + TL_n \sum_i^{N_l} c_{i-1} s_i^2 c_i + L_n \sum_l^{L} c_{l-1} s_l^2 c_l)$

estimator shows a significant performance gap compared to the DRSNN. This is because the LMMSE estimator is developed based on the constrained linear estimator. Similar to the results in Fig. 2b, the proposed DRSNN consistently achieves the best performance in terms of failure rate. This is expected since the DRSNN is a generalized regularization-based framework that subsumes an SBL estimator in a data-driven manner as a subcase, as analyzed in Section III. C.

To verify the robustness of our proposed method, we conduct simulations with varying propagation path numbers P and network input dimensionalities MN, where M and N represent the numbers of subcarriers and time slots, respectively. Note that we regenerated the dataset with varying P and MN and retrained the model accordingly. As shown in Fig. 4, the variation in path numbers does not significantly impact the estimation performance. In addition to P, the channel dimensionality MN also plays an important in estimation performance. We can observe that our proposed DRSNN method still achieves satisfactory performance for $MN = 480$ and $MN = 320$. This is because minor variations in the statistics of channel models do not significantly affect the performance of channel estimation. However, it is noticed that the performance of the proposed method declines sharply when $MN = 120$ and $MN = 960$. This is attributed to a significant mismatch between the kernel size and input dimensionality when MN is either too large or too small [19]. This issue may typically be addressed through data preprocessing, adding pooling layers, or multi-scale feature extraction [17].

A comparison of the computational complexity is provided in TABLE II. In the table, k denotes the number of iterations of SBL, c_i and s_i represent the numbers of channel and filter size at the i-th iteration, respectively, N_l and L denote the layers in a denoising block and the sparse prior-based regularization module, respectively. It can be seen that the LMMSE and SBL methods have high computational complexity with vector size L_n, while the DCNN and proposed DRSNN have linear complexity with L_n. Furthermore, DRSNN, which incorporates an additional lightweight network layer, has a relatively insignificant increase in complexity compared with the DCNN, yet leading to a notable improvement in performance.

V. Conclusions

This paper proposed a DL-based approach to address the sparse channel estimation problem. We formulated the original problem as a sparse signal denoising problem and developed the DRSNN scheme, which can effectively recover the channel at a lower complexity. Specifically, the proposed neural network consists of a denoising module that eliminates interference from noisy observations and a sparse prior-based regularization module that exploits channel sparsity. In particular, we derived a mathematical expression of DRSNN and theoretically compared it with SBL within the Bayesian theorem framework. Finally, simulation results demonstrated the superiority of DRSNN in terms of both estimation performance and computational complexity.

References

[1] S. Li, J. Yuan, W. Yuan, Z. Wei, B. Bai, and D. W. K. Ng, "Performance analysis of coded OTFS systems over high-mobility channels," *IEEE Trans. Wireless Commun.*, vol. 20, no. 9, pp. 6033–6048, Sept. 2021.

[2] C. Liu, S. Li, W. Yuan, X. Liu, and D. W. K. Ng, "Predictive precoder design for OTFS-enabled URLLC: A deep learning approach," *IEEE J. Sel. Areas Commun.*, vol. 41, no. 7, pp. 2245–2260, Jul. 2023.

[3] P. Raviteja, K. T. Phan, and Y. Hong, "Embedded pilot-aided channel estimation for OTFS in delay–doppler channels," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4906–4917, May. 2019.

[4] W. Shen, L. Dai, J. An, P. Fan, and R. W. Heath, "Channel estimation for orthogonal time frequency space (OTFS) massive MIMO," *IEEE Trans. Signal Process.*, vol. 67, no. 16, pp. 4204–4217, 2019.

[5] Z. Wei, W. Yuan, S. Li, J. Yuan, and D. W. K. Ng, "Off-grid channel estimation with sparse Bayesian learning for OTFS systems," *IEEE Trans. Wireless Commun.*, vol. 21, no. 9, pp. 7407–7426, Sept. 2022.

[6] W. Yuan, N. Wu, Q. Guo, Y. Li, C. Xing, and J. Kuang, "Iterative receivers for downlink MIMO-SCMA: Message passing and distributed cooperative detection," *IEEE Tran. Wireless Commun.*, vol. 17, no. 5, pp. 3444–3458, May. 2018.

[7] S. D. Babacan, R. Molina, and A. K. Katsaggelos, "Bayesian compressive sensing using Laplace priors," *IEEE Trans. Image Process.*, vol. 19, no. 1, pp. 53–63, Jan. 2009.

[8] C. Liu, X. Liu, D. W. K. Ng, and J. Yuan, "Deep residual learning for channel estimation in intelligent reflecting surface-assisted multi-user communications," *IEEE Trans. Wireless Commun.*, vol. 21, no. 2, pp. 898–912, Feb. 2022.

[9] X. Zhang, W. Yuan, C. Liu, F. Liu, and M. Wen, "Deep learning with a self-adaptive threshold for OTFS channel estimation," in *2022 Int. Sympos. Wireless Commun. Systems (ISWCS)*. IEEE, 2022, pp. 1–5.

[10] J. Mairal, "End-to-end kernel learning with supervised convolutional kernel networks," *Adv. Neural Inf. Process. Syst.*, vol. 29, 2016.

[11] J. Liu, Y. Sun, X. Xu, and U. S. Kamilov, "Image restoration using total variation regularized deep image prior," in *2019 IEEE Int. Conf. Acoust., Speech Signal Proces. (ICASSP)*. IEEE, 2019, pp. 7715–7719.

[12] M. Li, Y. Ma, and X. Zhang, "Proximal mapping for deep regularization," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 11 623–11 636, 2020.

[13] F. Liu, Z. Yuan, Q. Guo, Z. Wang, and P. Sun, "Message passing-based structured sparse signal recovery for estimation of OTFS channels with fractional doppler shifts," *IEEE Trans. Wireless Commun.*, vol. 20, no. 12, pp. 7773–7785, 2021.

[14] L. Zhao, W.-J. Gao, and W. Guo, "Sparse Bayesian learning of delay-Doppler channel for OTFS system," *IEEE Commun. Lett.*, vol. 24, no. 12, pp. 2766–2769, Dec. 2020.

[15] Z. Zhang and B. D. Rao, "Sparse signal recovery with temporally correlated source vectors using sparse bayesian learning," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 5, pp. 912–926, May. 2011.

[16] M. Teboulle, "Entropic proximal mappings with applications to nonlinear programming," *Mathematics of Operations Research*, vol. 17, no. 3, pp. 670–690, 1992.

[17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[18] S. M. Kay, *Fundamentals of statistical signal processing: estimation theory*. Prentice-Hall, Inc., 1993.

[19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 1–9.