Neural OTFS Receiver Without Training Requirements

Hao Chang, *Graduate Student Member, IEEE*, Gaoyang Pang, *Member, IEEE*, Branka Vucetic, *Life Fellow, IEEE*, and Wibowo Hardjawana, *Member, IEEE*

***Abstract*—Orthogonal time frequency space (OTFS) modulation is a promising 6G air interface for high-mobility communications. Recent OTFS receivers employ deep neural networks (DNNs) to mitigate air interface interference, but these schemes require large, environment-specific datasets for training before deployment. In this letter, we propose a neural OTFS receiver that performs joint channel estimation and symbol detection (JCED), requiring no datasets and training, referred to as untrained neural network (UNN)-JCED. Simulation results demonstrate that UNN-JCED outperforms existing OTFS receivers by over 1 dB in terms of normalized mean square error for channel estimation and symbol error rate for detection.**

***Index Terms*—Channel estimation, OTFS, symbol detection, untrained neural network.**

I. Introduction

THE EXISTING receiver for the 5G air interface that is based on orthogonal frequency-division multiplexing (OFDM), performs well in frequency-selective fading channels due to multi-path delay spread. However, when objects such as the transmitter, receiver, or reflectors are highly mobile, Doppler shifts make the channel path delay to also be time-variant, leading to performance degradation, creating a doubly selective fading channel. Orthogonal time frequency space (OTFS) modulation has been proposed in [1]. OTFS enables identification of both delay and Doppler shifts for each channel path at the receiver, making it a promising candidate for 6G air interfaces in high-mobility communications.

Recent advances in OTFS receivers [2], [3] employ deep neural networks (DNNs) to simplify the symbol detection design. These receivers require massive datasets to learn the symbol detection process by optimizing a large number of NN parameters in training before deployment. The Bayesian parallel interference cancellation network (BPICNet) proposed in [4] reduces the NN parameters to support high-mobility vehicular communications. It uses only three parameters in each NN layer and achieves symbol error rate (SER) performance close to near-optimal expectation propagation (EP) [5] with much lower complexity, but still requires massive training datasets. To further reduce the complexity, a message passing (MP) based symbol detection and threshold-based channel estimation were proposed in [6]. In these OTFS receivers, the channel estimation is done separately, resulting in sub-optimal performance as compared to joint channel estimation and symbol detection (JCED) [7], [8] that optimizes both processes. JCED has been studied by using trained DNN-based [7] and Bayesian-based [8], [9], with applications in Internet of vehicles, low earth orbit satellite, and underwater communications, respectively. Trained DNN-based JCED [7] requires large training datasets and struggles to generalize to channel conditions that differ from the ones used in training. The mismatch between the channel conditions used in training and testing leads to over 1 dB SER performance degradation for trained DNN-based receivers in [10]. Bayesian-based JCED [8], [9] needs no training and does not have the generalization problem, but involves iterative matrix inversions, resulting in high complexity. These limitations provide the motivation for a new JCED approach that requires no training datasets and is able to generalize with low complexity.

Recently, an untrained neural network (UNN) was proposed in [11] for channel estimation, eliminating the need for training datasets and matrix inversion. The scheme in [11] follows a similar idea in [12], where convolutional neural networks (CNNs) are used to recover the clean signal directly from its noisy observation. The performance of UNNs in [11], [12] relies on correlated features of the channel coefficients and images, respectively. This limits UNN's effectiveness for OTFS receiver design, as the transmitted OTFS symbols and channel coefficients are generally uncorrelated. To address this, a fully connected neural network (FCNN) based UNN was proposed in our preliminary work [13], [14] to support high-mobility vehicular communications. They perform symbol detection without the need for correlations between transmitted symbols. Specifically, the FCNN-based UNN was integrated with the BPIC scheme [15], forming UNN-BPIC. It achieves SER performance on par with near-optimal EP [5] and outperforms the BPICNet [4] with no use of training datasets and matrix inversion. In [13], [14], the NN parameters are optimized for each received signal, avoiding generalization issues across different channel conditions. Nevertheless, existing UNN-based approaches [13], [14] have not exploited data symbols to refine the channel estimate as done in JCED, thereby leaving a research gap in extending UNN to JCED.

To fill this research gap, we propose a novel UNN-based OTFS receiver for JCED in high-mobility vehicular communications, referred to as UNN-JCED. The main contribution of this letter is the first neural OTFS receiver for JCED without requiring training datasets and matrix inversion. Simulation results demonstrate that UNN-JCED outperforms existing OTFS receivers by over 1 dB in both normalized mean square error (NMSE) for channel estimation and SER for detection.

Received 7 July 2025; revised 15 August 2025; accepted 1 September 2025. Date of publication 8 September 2025; date of current version 10 November 2025. This work was supported in part by the University of Sydney International Scholarship. The associate editor coordinating the review of this article and approving it for publication was N. Zhao. *(Corresponding author: Hao Chang.)*

The authors are with the School of Electrical and Computer Engineering, The University of Sydney, Sydney, NSW 2006, Australia (e-mail: hao.chang@sydney.edu.au; gaoyang.pang@sydney.edu.au; branka.vucetic@sydney.edu.au; wibowo.hardjawana@sydney.edu.au).

Data is available on-line at https://github.com/HaoChang1/UNN-JCED.
Digital Object Identifier 10.1109/LWC.2025.3606969

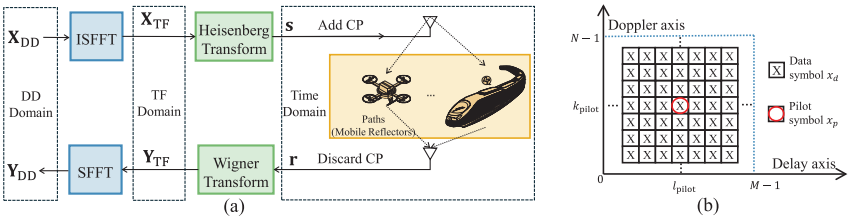

Fig. 1. (a) OTFS system model, and (b) $\mathbf{X}_{\mathrm{DD}}$.

Notations: We define a, $\mathbf{a}$, and $\mathbf{A}$ as scalar, vector, and matrix, respectively, and the estimates of them are denoted as $\hat{a}$, $\hat{\mathbf{a}}$, and $\hat{\mathbf{A}}$. The (i, j)-th entry of $\mathbf{A}$ is $\mathbf{A}[i, j]$. The transpose of matrix $\mathbf{A}$ is denoted as $\mathbf{A}^{\top}$. We define $\mathbb{C}^{M\times N}$ and $\mathbb{R}^{M\times N}$ as M × N matrices with complex and real values, respectively. A M × M identical matrix is denoted as $\mathbf{I}_M$. The mod-M operation is $[\,\cdot\,]_M$. The L2 and L1 norm of vector $\mathbf{x}$ is $\|\mathbf{x}\|_2$ and $\|\mathbf{x}\|_1$, respectively. A Gaussian distribution of $\mathbf{x}$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ is denoted as $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. We define $\mathbf{x} = vec(\mathbf{A})$ as the row-wise vectorization of matrix $\mathbf{A}$. The operations $\Re(\cdot)$ and $\Im(\cdot)$ take the real and imaginary parts, respectively. The Kronecker product operator is denoted as $\otimes$.

II. System Model

A. OTFS System Model

We consider a single-input single-output (SISO) OTFS system model in [16], as shown in Fig. 1a. The DD domain OTFS frame has N Doppler and M delay indices with the quantization step size $1/(\mathrm{NT})$ and $1/(M\Delta f)$ along the Doppler and delay axes, respectively. The subcarrier spacing is denoted by Δf in Hz, and NT is the OTFS frame length in seconds, and $T = 1/\Delta f$. The pilot and data symbols are allocated in the DD domain following the superimposed architecture in [17], denoted as $\mathbf{X}_{\mathrm{DD}} \in \mathbb{C}^{N\times M}$, as shown in Fig. 1b. The symbol transmitted in the Doppler index $k \in \{0, \ldots, N-1\}$ and delay index $l \in \{0, \ldots, M-1\}$ is given as

$$\mathbf{X}_{\mathrm{DD}}[k, l] = \begin{cases} \sqrt{E_p}x_p + \sqrt{E_d}x_d, & k = k_{\mathrm{pilot}}, l = l_{\mathrm{pilot}}, \\ \sqrt{E_d}x_d, & \text{otherwise}, \end{cases} \tag{1}$$

where k_{pilot} and l_{pilot} represent the DD domain pilot position where the data and pilot are superimposed. The pilot x_p and data symbol x_d follow $\tilde{M}$-ary quadrature amplitude modulation ($\tilde{M}-$QAM) with average power E_p and E_d, respectively, and $\tilde{M}$ is the modulation order. With the superimposed architecture in (1), all DD indices are used for data transmission, leading to high spectral efficiency. The interference between pilot and data symbols degrades the SER performance, requiring additional complexity for interference cancellation using JCED. We can split $\mathbf{X}_{\mathrm{DD}} = \mathbf{X}_p + \mathbf{X}_d$ into the transmitted pilot and data symbols only matrices, $\mathbf{X}_p$ and $\mathbf{X}_d$, respectively.

The DD domain symbol $\mathbf{X}_{\mathrm{DD}}$ is first transformed into the time-frequency (TF) domain using the inverse symplectic finite Fourier transform (ISFFT), resulting in $\mathbf{X}_{\mathrm{TF}} \in \mathbb{C}^{M\times N}$. Pulse-shaping operation by using a rectangular waveform with amplitude $1/\sqrt{T}$ and duration T and the Heisenberg transform are then applied to $\mathbf{X}_{\mathrm{TF}}$ to generate the time domain transmitted signal $\mathbf{s} \in \mathbb{C}^{MN}$. Our model uses a rectangular waveform, recommended in [18] for a realistic scenario to deliver excellent symbol detection performance, to transmit $\mathbf{s}$. The cyclic prefix (CP) of length M-1 is inserted at the beginning of $\mathbf{s}$ to prevent interference between different OTFS frames. The combined signal is then sampled at $1/(M\Delta f)$ seconds and sent via a wireless channel consisting of mobile reflectors, as shown in Fig. 1a. The received signal is then sampled every $1/(M\Delta f)$ seconds with CP removed, resulting in the time domain received signal $\mathbf{r} \in \mathbb{C}^{MN}$. Matched filter and the Wigner transform are then applied to $\mathbf{r}$ to produce the TF domain received signal $\mathbf{Y}_{\mathrm{TF}} \in \mathbb{C}^{M\times N}$. Finally, the SFFT is applied to $\mathbf{Y}_{\mathrm{TF}}$, resulting in the DD domain received signal $\mathbf{Y}_{\mathrm{DD}} \in \mathbb{C}^{M\times N}$, which is then used to perform JCED.

B. Signal Processing Model for Channel Estimation

We present a model to be used to perform channel estimation in Section III-A. Given the additive-white-Gaussian-noise (AWGN) $\tilde{\mathbf{n}} \sim \mathcal{N}(\mathbf{0}, \tilde{\sigma}^2\mathbf{I}_{MN})$, the received signal $\mathbf{Y}_{\mathrm{DD}}$ can be restructured as a vector $\tilde{\mathbf{y}} = vec(\mathbf{Y}_{\mathrm{DD}})$ [8],

$$\tilde{\mathbf{y}} = \left(\tilde{\boldsymbol{\Phi}}_p + \tilde{\boldsymbol{\Phi}}_d\right)\tilde{\mathbf{h}} + \tilde{\mathbf{n}}, \tag{2}$$

where $\tilde{\mathbf{h}} = [\tilde{h}_{0,-k_{\max}}, \ldots, \tilde{h}_{0,k_{\max}}, \ldots, \tilde{h}_{l_{\max},k_{\max}}] = [\tilde{h}_1, \ldots, \tilde{h}_L] \in \mathbb{C}^L$ is the channel vector and $L = (2k_{\max} + 1)(l_{\max} + 1)$. We use $\tilde{h}_{a,b} \sim \mathcal{N}(0, E_a)$ to represent a possible channel path with delay index $a \in \{0, \ldots, l_{\max}\}$ and Doppler shift index $b \in \{-k_{\max}, \ldots, k_{\max}\}$. Each non-zero $\tilde{h}_{a,b}$ indicates a channel path (mobile reflector) that has a channel delay of aT/M seconds and Doppler shift of $b\Delta f/N$ Hz with power E_a. Translating to a practical system, our model assumes multi-path wireless channel with a maximum delay of $l_{\max}T/M$ seconds and Doppler shifts of $\pm k_{\max}\Delta f/N$ Hz, $l_{\max} < M$ and $k_{\max} \le N/2$. This is equivalent to maximum supportable mobility of $c\Delta f k_{\max}/(Nf_c)$ m/s, where c is the speed of light and f_c is the carrier frequency. We assume that not necessarily all possible L channel paths are used by P mobile reflectors, $P \le L$. We use matrix $\tilde{\boldsymbol{\Phi}}_e = [\boldsymbol{\phi}^e_1, \ldots, \boldsymbol{\phi}^e_{MN}]^{\top} \in \mathbb{C}^{MN\times L}, \forall e \in \{p, d\}$ to capture the impact of Doppler shift and channel delay to the (i, j)-th entry of the pilot and data symbol matrices, $\mathbf{X}_e[i, j], \forall e \in \{p, d\}$. The $(Nl + k + 1)$-th row of the matrix $\tilde{\boldsymbol{\Phi}}_e$ can be written as $\boldsymbol{\phi}^e_{Nl+k+1} = [\phi^e_{0,-k_{\max}}(k, l), \ldots, \phi^e_{0,k_{\max}}(k, l), \ldots, \phi^e_{l_{\max},k_{\max}}(k, l)]^{\top}$,

$$\phi^e_{a,b}(k, l) = \mathbf{X}_e[[k-b]_N, [l-a]_M]\beta_{a,b}(k, l)e^{j2\pi\frac{(l-a)b}{MN}},$$
$$\beta_{a,b}(k, l) = \begin{cases} 1, & a \le l < M, \\ \frac{N-1}{N}e^{-j2\pi\left(\frac{[k-b]_N}{N}\right)}, & 0 \le l < a. \end{cases} \tag{3}$$

We use an equivalent real-valued model of (2), given as

$$\mathbf{y} = \left(\boldsymbol{\Phi}_p + \boldsymbol{\Phi}_d\right)\mathbf{h} + \mathbf{n} = \boldsymbol{\Phi}\mathbf{h} + \mathbf{n} \tag{4}$$

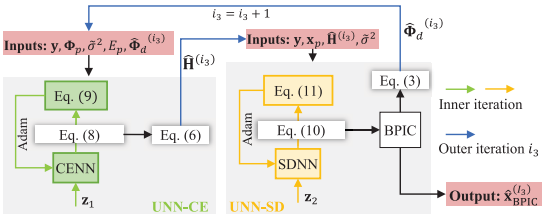

Fig. 2. The proposed UNN-JCED architecture.

where $\mathbf{\Phi}_e = \begin{bmatrix} \Re(\tilde{\mathbf{\Phi}}_e) & -\Im(\tilde{\mathbf{\Phi}}_e) \\ \Im(\tilde{\mathbf{\Phi}}_e) & \Re(\tilde{\mathbf{\Phi}}_e) \end{bmatrix} \in \mathbb{R}^{2MN \times 2L}, \forall e \in \{p, d\}$, $\mathbf{y} = \left[\Re(\tilde{\mathbf{y}})^\top \ \Im(\tilde{\mathbf{y}})^\top\right]^\top \in \mathbb{R}^{2MN}$, $\mathbf{h} = \left[\Re(\tilde{\mathbf{h}})^\top \ \Im(\tilde{\mathbf{h}})^\top\right]^\top \in \mathbb{R}^{2L}$, $\mathbf{n} = \left[\Re(\tilde{\mathbf{n}})^\top \ \Im(\tilde{\mathbf{n}})^\top\right]^\top \in \mathbb{R}^{2MN}$, $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \tilde{\sigma}^2/2\mathbf{I}_{2MN})$.

C. Signal Processing Model for Symbol Detection

We present a model to be used to perform symbol detection in Section III-B. The received signal $\tilde{\mathbf{y}}$ is given as

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}}\left(\tilde{\mathbf{x}}_p + \tilde{\mathbf{x}}_d\right) + \tilde{\mathbf{n}}, \tag{5}$$

where $\tilde{\mathbf{x}}_d = vec(\mathbf{X}_d)$ and $\tilde{\mathbf{x}}_p = vec(\mathbf{X}_p)$ are the vectorized DD domain transmitted data and pilot symbol, respectively. The OTFS channel matrix $\tilde{\mathbf{H}}$ that captures the impact of Doppler shift and delay of the channel paths is given as

$$\tilde{\mathbf{H}} = \sum_{a=0}^{l_{\max}} \sum_{b=-k_{\max}}^{k_{\max}} h_{a,b} (\mathbf{F}_N \otimes \mathbf{G}_{\text{rx}}) \mathbf{I}_{MN}(a) \mathbf{\Delta}^b \left(\mathbf{F}_N^H \otimes \mathbf{G}_{\text{tx}}\right), \tag{6}$$

where $\mathbf{F}_N$ and $\mathbf{F}_N^H$ are the N point discrete Fourier transform (DFT) and inverse DFT (IDFT) matrix. The term $\mathbf{I}_{MN}(a)$ is obtained by circularly left shifting the columns of $\mathbf{I}_{MN}$ by a and $\mathbf{\Delta}^b = \text{diag}[z^0, z^1, \ldots, z^{MN-1}]$, $z = e^{\frac{j2\pi b}{MN}}$. The use of rectangular pulse-shaping waveform results in $\mathbf{G}_{\text{tx}} = \mathbf{G}_{\text{rx}} = \mathbf{I}_M$. The real-valued model of (5) is given as

$$\mathbf{y} = \mathbf{H}\left(\mathbf{x}_p + \mathbf{x}_d\right) + \mathbf{n}, \tag{7}$$

where $\mathbf{x}_e = \left[\Re(\tilde{\mathbf{x}}_e)^\top \ \Im(\tilde{\mathbf{x}}_e)^\top\right]^\top \in \mathbb{R}^{2MN}, \forall e \in \{p, d\}$ and $\mathbf{H} = \begin{bmatrix} \Re(\tilde{\mathbf{H}}) & -\Im(\tilde{\mathbf{H}}) \\ \Im(\tilde{\mathbf{H}}) & \Re(\tilde{\mathbf{H}}) \end{bmatrix} \in \mathbb{R}^{2MN \times 2MN}$.

III. Untrained Neural Network Based JCED

In this section, we explain the proposed UNN-JCED based on (4) and (7), as shown in Fig. 2. It consists of UNN-CE and UNN-SD blocks. The two blocks exchange information and are iteratively updated I_3 times, referred to as outer iterations. At each outer iteration $i_3 \in \{1, \ldots, I_3\}$, UNN-CE calculates the estimate of the channel matrix $\mathbf{H}$ in (6), as $\hat{\mathbf{H}}^{(i_3)}$, using the measurement matrix $\hat{\mathbf{\Phi}}_d^{(i_3-1)}$ obtained from UNN-SD in the previous iteration. Based on that, UNN-SD then calculates the estimate of measurement matrix $\mathbf{\Phi}_d$ in (3), as $\hat{\mathbf{\Phi}}_d^{(i_3)}$, and returns $\hat{\mathbf{x}}_{\text{BPIC}}^{(I_3)}$ at iteration I_3. The estimated measurement matrix $\hat{\mathbf{\Phi}}_d^{(i_3)}$ is then used by UNN-CE to improve its estimates for the next outer iteration, and the whole process is repeated.

For $i_3 = 1$, UNN-CE can not get the estimate of the measurement matrix $\hat{\mathbf{\Phi}}_d^{(0)}$ for channel estimation. Thus, we adopt the threshold scheme with a threshold $\tau = 3\sqrt{E_d + \tilde{\sigma}^2}$ [17] to obtain the estimated channel for its low complexity. The role of the threshold is to identify the delay index a, the Doppler shift index b for existing P out of L possible channel paths by comparing the magnitude of all possible received pilot symbols in (2) with the threshold τ. The corresponding magnitude $h_{a,b}$ is obtained by dividing the received pilot symbols by the transmitted pilot. The estimated a, b, and $h_{a,b}$ are used to construct $\hat{\mathbf{H}}^{(1)}$ by using (6). The true noise variance $\tilde{\sigma}^2$ in the threshold τ is assumed known, as in [6], [17]. If not, we estimate as $(\hat{\sigma}^2)^{(i_3)} = var(\mathbf{y} - \hat{\mathbf{H}}^{(i_3)}(\hat{\mathbf{x}}_{\text{BPIC}}^{(i_3)} + \mathbf{x}_p))$ for every i_3 outer iteration. We set the initial estimate $(\hat{\sigma}^2)^{(1)} = 1$, and $var(\cdot)$ calculates the variance of a vector.

A. UNN-CE at Outer Iteration $i_3 > 1$

In UNN-CE, a channel estimation NN (CENN) is represented by the function f_{CENN}. It performs channel estimation using four fully connected (FC) layers with ReLU activation used after the first three FC layers. This is given as

$$\hat{\mathbf{h}}^{(i_1)} = f_{\text{CENN}}\left(\mathbf{z}_1, \mathbf{\Theta}_{\text{CENN}}^{(i_1)}\right), \tag{8}$$

where $\hat{\mathbf{h}}^{(i_1)} \in \mathbb{R}^{2L}$ is the CENN output representing the estimated channel vector in (4) at inner iteration $i_1 \in \{1, \ldots, I_1\}$. The CENN input is $\mathbf{z}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{1}) \in \mathbb{R}^4$. The term $\mathbf{\Theta}_{\text{CENN}}^{(i_1)}$ represents the NN parameters at inner iteration i_1, and it is initialized with a uniform distribution following [19].

For each i_1 iteration, we identify $\hat{P}$ out of L possible channel paths in $\hat{\mathbf{h}}^{(i_1)}$ by applying a threshold function to set $L - \hat{P}$ non-existing paths to 0. This means for $\hat{\mathbf{h}}^{(i_1)} = [\hat{h}_1^{(i_1)}, \ldots, \hat{h}_j^{(i_1)}, \ldots, \hat{h}_{2L}^{(i_1)}], 1 < j < L$, if $(\hat{h}_j^{(i_1)})^2 + (\hat{h}_{j+L}^{(i_1)})^2 < \epsilon, \hat{h}_j^{(i_1)} = \hat{h}_{j+L}^{(i_1)} = 0$. We set $\epsilon = ((i_3 - 2)I_1 + i_1)/((I_3 - 1)I_1)((\hat{\sigma}^2)^{(i_3)}/E_p)$ as a dynamic threshold, increasing with the number of iterations and approaching Cramer–Rao Lower Bound (CRLB) of estimation error $\tilde{\sigma}^2/E_p$ [20]. We set $(\hat{\sigma}^2)^{(i_3)} = \tilde{\sigma}^2$ if the true noise variance is known. The rationale behind the design is to avoid aggressive selection that leads to wrongly excluding channel paths during earlier iterations.

The NN parameters in CENN are then updated by using the following loss function and Adam optimizer

$$\mathcal{L}_1^{(i_1)} = \frac{1}{2MN} \|\mathbf{y} - \left(\mathbf{\Phi}_p + \hat{\mathbf{\Phi}}_d^{(i_3-1)}\right)\hat{\mathbf{h}}^{(i_1)}\|_2^2 + \lambda \|\hat{\mathbf{h}}^{(i_1)}\|_1, \tag{9}$$

where $\hat{\mathbf{\Phi}}_d^{(i_3-1)}$ is the estimated measurement matrix at outer iteration $i_3 - 1$. The regularization factor λ is used to penalize the sparsity of the estimated channel $\hat{\mathbf{h}}^{(i_1)}$. The received signal $\mathbf{y}$ and measurement matrices $\mathbf{\Phi}_p$ and $\hat{\mathbf{\Phi}}_d^{(i_3-1)}$ are the inputs of UNN-CE. After I_1 inner iterations, by applying (6) to $\hat{\mathbf{h}}^{(I_1)}$, we obtain $\hat{\mathbf{H}}^{(i_3)}$, which is then forwarded to UNN-SD.

B. UNN-SD at Outer Iteration i_3

In UNN-SD, a symbol detection NN (SDNN) is represented by the function f_{SDNN}. It performs symbol detection using four FC layers with ReLU and Tanh used after the first three FC layers and the last FC layer. This is given as

$$\hat{\mathbf{x}}_d^{(i_2)} = \alpha f_{\text{SDNN}}\left(\mathbf{z}_2, \mathbf{\Theta}_{\text{SDNN}}^{(i_2)}\right), \tag{10}$$

where $\hat{\mathbf{x}}_d^{(i_2)} \in \mathbb{R}^{2MN}$ is the estimated data symbol at inner iteration $i_2 \in \{1, \ldots, I_2\}$. The SDNN input is $\mathbf{z}_2 \sim \mathcal{N}(\mathbf{0}, \mathbf{1}) \in \mathbb{R}^4$, and α is used to restrict the output range

within the minimum and maximum values of $\tilde{M}$-QAM symbol $\mathbf{x}_d$ in (7).

The NN parameters in SDNN are then initialized and updated similarly to CENN by using the following loss function

$$\mathcal{L}_2^{(i_2)} = \frac{1}{2MN}\|\mathbf{y} - \hat{\mathbf{H}}^{(i_3)}\left(\hat{\mathbf{x}}_d^{(i_2)} + \mathbf{x}_p\right)\|_2^2. \tag{11}$$

With the received signal $\mathbf{y}$, channel matrix $\hat{\mathbf{H}}^{(i_3)}$ and pilot $\mathbf{x}_p$ as inputs, UNN-SD outputs $\hat{\mathbf{x}}_d^{(I_2)}$ after I_2 inner iterations.

BPIC based on [13], and [15, Algorithm 1], is then run T_1 times to improve the symbol estimates $\hat{\mathbf{x}}_d^{(I_2)}$, which is assumed to be correct initially. Other inputs for BPIC are the estimated noise variance $(\hat{\sigma}^2)^{(i_3)}$, channel matrix $\hat{\mathbf{H}}^{(i_3)}$, and $\mathbf{y}-\hat{\mathbf{H}}^{(i_3)}\mathbf{x}_p$. The latter is the received signal of the data components. BPIC output is then used to compute $\hat{\mathbf{\Phi}}_d^{(i_3)}$ by using (3), which is then forwarded to UNN-CE for the next outer iteration.

IV. Performance Evaluation

A. Simulation Settings

We set $M = N = 16$, $l_{\max} = 3$, $k_{\max} = 3$, $P = 8$, $k_{\text{pilot}} = l_{\text{pilot}} = 7$, $\tilde{M} = 4$, $\Delta f = 15$kHz and $f_c = 6$GHz, leading to a maximum supported mobility of 506 km/h. We use the 3GPP tapped delay line (TDL) channel model [21] with delay spread $1/(M\Delta f)$, and the channel power E_a for path $\tilde{h}_{a,b}$ is set according to the power delay profile. All simulation results are collected under non-line-of-sight (NLoS) TDL-B and LoS TDL-E channels. We define the transmit data signal-to-noise ratio (SNR) as $SNR_d = E_d/\tilde{\sigma}^2 = 1/\tilde{\sigma}^2$, and pilot SNR as $SNR_p = E_p/\tilde{\sigma}^2$. The pilot, data, and noise power, E_p, E_d, and $\tilde{\sigma}^2$ impacting the SNR definition are time-invariant. However, when interference at the receiver is considered, the SNR becomes time-varying and can be low. We refer to it as the signal-to-interference-plus-noise ratio (SINR) to reflect the impact of interference on SNR. The interference arises due to the use of superimposed architecture in (1) and the impact of the Doppler shift on each channel path. We assume the true noise variance $\tilde{\sigma}^2$ is known for all schemes. We define UNN-JCED and UNN-JCED-est for the use of true and estimated noise variance, respectively. We set $\alpha = 1/\sqrt{2}$ corresponding to the maximum value of $\mathbf{x}_d$ in (7), and $\lambda = 0.01$. The learning rate of CENN and SDNN is 0.01. The output size of each FC layer in CENN and SDNN is set to $\{8, 16, 32, 2L\}$ and $\{8, 16, 32, 2MN\}$, respectively.

We consider the following schemes for comparison: **1) Threshold:** The classical threshold based channel estimation [6] discussed in Section III. **2) MMSE:** It applies the minimum-mean-square-error (MMSE) technique to (4) to obtain the channel estimates with only information of the received signal $\mathbf{y}$ and pilot information $\mathbf{\Phi}_p$. **3) CRLB:** We use CRLB in [20] to obtain the channel estimation error of $\tilde{h}_{a,b}$ in terms of NMSE. The delay index a and Doppler shift index b in $\mathbf{h}$, as well as $\mathbf{\Phi}$, are assumed to be perfectly known in (4). The NMSE of CRLB is then given as $\tilde{\sigma}^2/2\sum_{j=1}^{2P}((\mathbf{\Phi}^\top\mathbf{\Phi})[j,j])^{-1}/\|\mathbf{h}\|_2^2$. **4) Threshold-MP:** It obtains the estimated channel based on the threshold scheme, followed by MP with $T_2 = 10$ iterations for symbol detection in [6]. **5) BPICNet-JCED-X**: We combine the BPICNet OTFS detector in [4] with the MMSE channel estimator as a representative for a trained DNN-based JCED. The BPICNet is trained following [4] under $\mathrm{X} \in \{\mathrm{TDL-B}, \mathrm{TDL-E}\}$ channels. This is similar to UNN-JCED, where the UNN-CE is replaced by MMSE, and the UNN-SD is replaced by BPICNet with $T_1 = 10$ layers, and the number of outer iterations is $T_4 = 20$. **6) Threshold-EP, Ideal-EP**: We use the EP OTFS detector in [5] with $T_4 = 10$ iterations, assuming either threshold-based channel estimation or ideal channel state information (CSI). **7) Threshold-UNN-BPIC, Ideal-UNN-BPIC**: We only perform UNN-SD in JCED, assuming either threshold-based channel estimation or ideal CSI. This is the same as our preliminary work in [13].

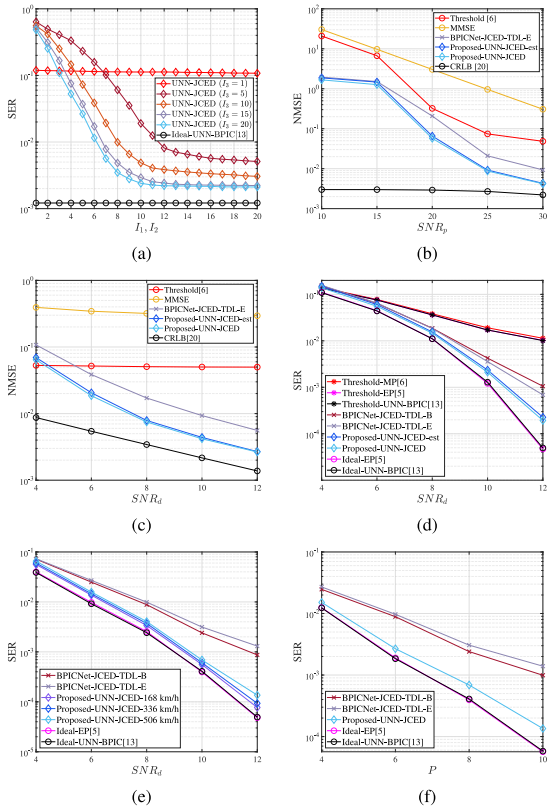

Fig. 3. Simulation results under TDL-E (a, b, c, d) and TDL-B (e, f) channels. (a) $I_1 = I_2, SNR_d = 10$dB, $SNR_p = 30$ dB. (b) $SNR_d = 10$dB. (c-e) $SNR_p = SNR_d + 20$dB. (f) $SNR_d = 10$dB, $SNR_p = 30$dB.

B. Results and Discussion

The convergence analysis of UNN-JCED for different I_1, I_2, I_3, measured in terms of SER, is shown in Fig. 3(a). It shows the dependency between I_1, I_2, and I_3, underlining the need to set a sufficiently high number of iterations to guarantee the convergence of the proposed UNN-JCED. $I_1 = I_2 = I_3 = 20$ are used for the subsequent simulations. Fig. 3(b) shows NMSE simulation results for all schemes across a range of SNR_p. The proposed UNN-JCED and UNN-JCED-est outperform comparison schemes for all SNR_p and approach the CRLB when SNR_p is high due to superior management of interference caused by superimposed pilot and data symbols. We also note that not knowing the true noise variance at the receiver does not degrade the NMSE performance. The NMSE performance for different SNR_d is shown in Fig. 3(c).

TABLE I
COMPLEXITY COMPARISON

Receivers	Computational complexity	Run time ($\times 10^3 NT$)
Threshold-MP [6]	$\mathcal{O}(M^2N^2PT_2 + L)$	4.59
BPICNet-JCED-X	$\mathcal{O}((M^3N^3+M^2N^2T_1 + MNL + L^3)T_4 + L)$	4.55
Proposed UNN-JCED	$\mathcal{O}(M^2N^2(I_2+T_1)I_3 + MNLI_1(I_3 - 1) + L)$	2.58
Ideal-EP [5]	$\mathcal{O}((M^3N^3)T_3)$	1.51
Ideal-UNN-BPIC [13]	$\mathcal{O}(M^2N^2(I_2I_3+T_1))$	0.56

The threshold scheme fails to perform channel estimates with increasing SNR_d because the separate approach does not exploit data symbols as additional pilots to improve channel estimation. In contrast, BPICNet-JCED-TDL-E significantly outperforms the MMSE channel estimator. This is because the latter uses only the pilot symbol to estimate the channel, whereas the former uses both pilot and data symbols, leading to much better performance. Moreover, both UNN-JCED and UNN-JCED-est outperform BPICNet-JCED-TDL-E and approach CRLB, as the dynamic threshold filters non-existing paths, improving the channel estimation performance.

The SER performance of UNN-JCED and UNN-JCED-est is shown in Fig. 3(d). They outperform all schemes that use threshold-based channel estimation by over 3 dB and BPICNet-JCED-TDL-E by over 1 dB, and are only 0.5 dB away from Ideal-EP and Ideal-UNN-BPIC. The SER performance of BPICNet-JCED-TDL-B is worse than BPICNet-JCED-TDL-E, due to the mismatch between the training and testing channels of the former. This finding is consistent with [10]. Fig. 3(e) demonstrates that BPICNet-JCED-TDL-E suffers from similar performance degradation under a TDL-B channel. However, the proposed UNN-JCED remains an excellent SER performance, due to the independence of the UNN-JCED on channel environments. Fig. 3(e) shows that UNN-JCED achieves nearly identical SER performance under maximum supportable mobilities of 168, 336, and 506 km/h for $k_{\max} = 1, 2$, and 3, respectively, indicating its robustness to different mobilities. The impact of varying numbers of P mobile reflectors is shown in Fig. 3(f), where BPICNet-JCED-TDL-B is trained separately for each $P \in \{4, 6, 8, 10\}$, and BPICNet-JCED-TDL-E is trained only for $P = 4$. Fig. 3(f) demonstrates that the SER performance of BPICNet-JCED-TDL-E is worse than that of BPICNet-JCED-TDL-B for all P, due to the training datasets of the former not including TDL-B channels and $P \neq 4$. In contrast, UNN-JCED remains on par with Ideal-EP and Ideal-UNN-BPIC regardless of P. In conclusion, the proposed UNN-JCED can adapt to dynamic OTFS environments without modifying the architecture.

The comparison of the computational complexity is summarized in Table I, where the number of multiplications is used for evaluation. The complexity of UNN-JCED comes mainly from NNs related to M, N, L, and iterations I_1, I_2, I_3 to find the optimum NN parameters for UNN-CE and UNN-SD. Here, we provide the run time of the proposed and other schemes in terms of OTFS frame length ($NT \approx 1$ ms) following the settings in Section IV. A and B, obtained on a local computer with an Intel i7-12700K CPU and an NVIDIA RTX 3060Ti GPU. As shown in Table I, the run time of the proposed UNN-JCED is half of BPICNet-JCED-X and [6], indicating that UNN-JCED can potentially support lower coherence time and latency requirements. Due to additional channel estimation, the complexity of UNN-JCED is higher than [5] and [13].

V. CONCLUSION

We propose a neural OTFS receiver for JCED without training requirements. Simulation results demonstrate that UNN-JCED outperforms state-of-the-art OTFS receivers by over 1 dB in both NMSE for channel estimation and SER for detection. Future work includes extending the approach to fractional Doppler to better reflect practical environments.

REFERENCES

[1] R. Hadani et al., "Orthogonal time frequency space modulation," in *Proc. IEEE WCNC*, Mar. 2017, pp. 1–6.

[2] Y. K. Enku et al., "Two-dimensional convolutional neural network-based signal detection for OTFS systems," *IEEE Wireless Commun. Lett.*, vol. 10, no. 11, pp. 2514–2518, Nov. 2021.

[3] X. Zhang et al., "Graph neural network assisted efficient signal detection for OTFS systems," *IEEE Commun. Lett.*, vol. 27, no. 8, pp. 2058–2062, Aug. 2023.

[4] A. Kosasih et al., "Bayesian neural network detector for an orthogonal time frequency space modulation," *IEEE Wireless Commun. Lett.*, vol. 11, no. 12, pp. 2570–2574, Dec. 2022.

[5] H. Li et al., "Low complexity receiver via expectation propagation for OTFS modulation," *IEEE Commun. Lett.*, vol. 25, no. 10, pp. 3180–3184, Oct. 2021.

[6] P. Raviteja et al., "Embedded pilot-aided channel estimation for OTFS in delay–Doppler channels," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4906–4917, May 2019.

[7] B. Wang et al., "Data-driven intelligent receiver for OTFS communication in Internet of Vehicles," *IEEE Trans. Veh. Technol.*, vol. 73, no. 5, pp. 6968–6979, May 2024.

[8] X. Wang et al., "Joint Bayesian channel estimation and data detection for OTFS systems in LEO satellite communications," *IEEE Trans. Commun.*, vol. 70, no. 7, pp. 4386–4399, Jul. 2022.

[9] Y. Liang et al., "Joint Bayesian channel estimation and data detection for underwater acoustic communications," *IEEE Trans. Commun.*, vol. 72, no. 9, pp. 5868–5883, Sep. 2024.

[10] J. C. De Luna Ducoing et al., "An assessment of deep learning versus massively parallel, non-linear methods for highly-efficient MIMO detection," *IEEE Access*, vol. 11, pp. 97493–97502, 2023.

[11] E. Balevi et al., "Massive MIMO channel estimation with an untrained deep neural network," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2079–2090, Mar. 2020.

[12] D. Ulyanov et al., "Deep image prior," in *Proc. IEEE/CVF CVPR*, Jun. 2018, pp. 9446–9454.

[13] H. Chang et al., "Untrained neural network based Bayesian detector for OTFS modulation systems," in *Proc. IEEE INFOCOM Workshops*, 2023, pp. 1–6.

[14] H. Chang et al., "Graph-based untrained neural network detector for OTFS systems," in *Proc. IEEE Veh Technol Conf*, 2024, pp. 1–6.

[15] A. Kosasih et al., "A Bayesian receiver with improved complexity-reliability trade-off in massive MIMO systems," *IEEE Trans. Commun.*, vol. 69, no. 9, pp. 6251–6266, Sep. 2021.

[16] P. Raviteja et al., "Practical pulse-shaping waveforms for reduced-cyclic-prefix OTFS," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 957–961, Jan. 2019.

[17] W. Yuan et al., "Data-aided channel estimation for OTFS systems with a superimposed pilot and data transmission scheme," *IEEE Wireless Commun. Lett.*, vol. 10, no. 9, pp. 1954–1958, Sep. 2021.

[18] A. Zhou et al., "On the performance of practical pulse-shaped OTFS with analog receivers," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, 2023, pp. 518–523.

[19] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Mach. Learn. Res.*, vol. 9, 2010, pp. 249–256.

[20] S. Srivastava et al., "Bayesian learning aided sparse channel estimation for orthogonal time frequency space modulated systems," *IEEE Trans. Veh. Technol.*, vol. 70, no. 8, pp. 8343–8348, Aug. 2021.

[21] "Study on channel model for frequencies from 0.5 to 100 GHz (Version 16.1.0 Release 16)," Eur. Telecommun. Stand. Inst. (ETSI), Sophia-Antipolis, France, Rep. TR 38.901, Nov. 2020. [Online]. Available: https://www.etsi.org/