

An RNN based DD Channel Estimator for OTFS with Embedded Pilots

Sandesh Rao Mattu and A. Chockalingam

Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore

Abstract—In this paper, we propose a learning based architecture for estimating the delay-Doppler (DD) channel in orthogonal time frequency space (OTFS) systems with embedded pilots. The proposed learning network, called DDNet, is based on a multi-layered recurrent neural network (RNN) framework with a novel training methodology that works seamlessly for both exclusive pilot frames as well as embedded pilot frames. This generalization is attributed to the training methodology, wherein multiple frame realizations with different guard band sizes are used to train the network. Simulation results demonstrate that the proposed DDNet achieves better mean square error and bit error performance compared to impulse based and threshold based DD channel estimation schemes.

Index Terms—OTFS modulation, deep learning, DD channel estimation, recurrent neural network, embedded pilots.

I. INTRODUCTION

Orthogonal time frequency space (OTFS) modulation is a recently introduced modulation scheme suited for high-mobility channels [1]. While contemporary multicarrier modulation schemes such as orthogonal frequency division multiplexing (OFDM) suffer from inter-carrier interference caused by high Doppler spreads in high mobility channels, OTFS has been shown to be robust to high Doppler spreads [1]-[3]. In OTFS, information symbols are multiplexed in the delay-Doppler (DD) domain and the channel is also viewed in the DD domain. A key benefit of viewing the channel in the DD domain is that a rapidly time-varying channel appears as almost a time-invariant channel in the DD domain. Also, the channel matrix is sparse in the DD domain representation. These attributes of the channel in the DD domain makes the channel estimation task less complex.

Accurate estimation of the DD domain channel is essential for reliable OTFS signal detection at the receiver [4],[5]. Several approaches to DD channel estimation for OTFS has been reported in the literature [6]-[12]. Some of these approaches include use of DD domain impulses and PN sequences as pilots [6],[2], threshold based estimation [7], and compressed sensing/sparse Bayesian learning based estimation [11],[12] to name a few. Recently, deep learning for wireless PHY layer designs are increasingly being explored in the literature (e.g., for signal detection, channel estimation, phase noise compensation), and the reported results are promising [13]-[16]. Deep neural networks for channel estimation in OFDM systems have been reported in [14]-[16]. Our interest in this

paper is the investigation of deep learning approach for DD channel estimation in OTFS, which remains to be explored. In particular, we propose a novel learning based DD channel estimation approach for OTFS with embedded pilot frames.

In the embedded pilot scheme in [7], a noise variance based threshold is employed to estimate the delay-Doppler indices of the different paths and the corresponding channel coefficients. This threshold based scheme is simple and works well for high pilot SNRs. However, its performance gets compromised at low-to-moderate pilot SNRs as the estimation of the delay-Doppler indices becomes more erroneous at these SNRs. To overcome this, [8] proposed a maximum likelihood scheme for estimating the Doppler indices of different paths and minimum mean square error scheme for estimating the channel coefficients. A limitation in [8], however, is that it has assumed perfect knowledge of the delay indices of the paths.

In the present work, we overcome the above limitations by using a learning network, which we call as DDNet, that estimates all the three-tuples (i.e., delay indices, Doppler indices, and DD channel coefficients) for all paths and achieves much better performance compared to the thresholding scheme in [7]. The proposed learning network is based on a multi-layered recurrent neural network (RNN) architecture. A novel ingredient in the proposed learning approach is a generalized training methodology that makes the network generalize for OTFS frames with different pilot arrangements. In other words, once trained, the network can seamlessly work for OTFS frames with different guard band sizes. Simulation results for Vehicular A channel model with 220 km/h speed demonstrate that the proposed DDNet achieves significantly better mean square error and bit error performance compared to threshold based DD channel estimation scheme.

The rest of the paper is organized as follows. The OTFS system model is presented in Sec. II. The proposed DDNet architecture and training methodology are presented in Sec. III. Simulation results and discussions are presented in Sec. IV. Conclusions are presented in Sec. V.

Notations: t_j represents the j th entry in the vector \mathbf{t} . $\lceil x \rceil$ and $\lfloor x \rfloor$ represent the ceiling and flooring operation on x , respectively. $(x)_M$ denotes the modulo M operation on x . $\mathbb{E}[\cdot]$ denotes the expectation operator and $\|\mathbf{A}\|_F$ denotes the Frobenius norm of matrix \mathbf{A} .

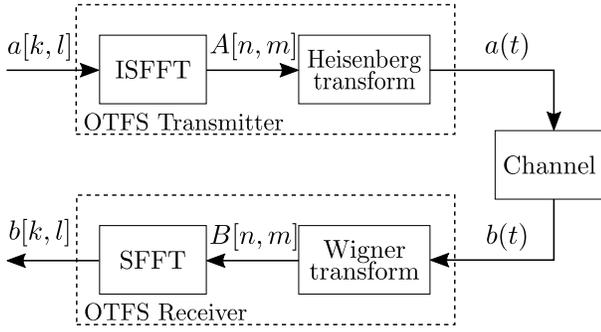


Fig. 1: OTFS modulation scheme.

II. OTFS SYSTEM MODEL

Figure 1 shows the block diagram of the OTFS modulation scheme. At the transmitter, information symbols are placed in the DD domain. They are mapped to time-frequency (TF) domain using inverse symplectic finite Fourier transform (ISFFT). This is followed by conversion to time domain using Heisenberg transform. The time domain signal is transmitted through the channel. At the receiver, the received time domain signal is converted to TF domain using Wigner transform. This is followed by conversion back to DD domain using symplectic finite Fourier transform (SFFT) for detection.

MN information symbols, denoted by $a[k, l]$ s, each belonging to a modulation alphabet \mathbb{A} are placed in an $M \times N$ DD grid $\left\{ \left(\frac{l}{M\Delta f}, \frac{k}{NT} \right), l = 0, \dots, M-1, k = 0, \dots, N-1 \right\}$, where N is the number of Doppler bins, M is the number of delay bins, Δf is the subcarrier spacing, and $T = 1/\Delta f$. The quantities $1/NT$ and $1/M\Delta f$ represent the bin sizes in the Doppler and delay domains, respectively. $a[k, l]$ s in the DD domain are converted to TF domain symbols $A[n, m]$ using the ISFFT operation, given by

$$A[n, m] = \frac{1}{\sqrt{MN}} \sum_{k=0}^{N-1} \sum_{l=0}^{M-1} a[k, l] e^{j2\pi \left(\frac{nk}{N} - \frac{ml}{M} \right)}, \quad (1)$$

for $n = 0, \dots, N-1$ and $m = 0, \dots, M-1$. To obtain the time domain signal $a(t)$, Heisenberg transform of the TF signal $A[n, m]$ is computed. Using a transmit pulse $p_{tx}(t)$, this operation is defined as

$$a(t) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} A[n, m] p_{tx}(t - nT) e^{j2\pi m \Delta f (t - nT)}. \quad (2)$$

The time domain signal $a(t)$ is transmitted through the channel. The channel has the complex baseband channel response in the DD domain, denoted by $g(\tau, \nu)$, given by [3]

$$g(\tau, \nu) = \sum_{i=0}^{L-1} g_i \delta(\tau - \tau_i) \delta(\nu - \nu_i), \quad (3)$$

where L is the number of channel paths in the DD domain, δ is the Kronecker delta function, and g_i , τ_i , and ν_i represent the channel gain, delay, and Doppler shift, respectively, corresponding to the i th path.

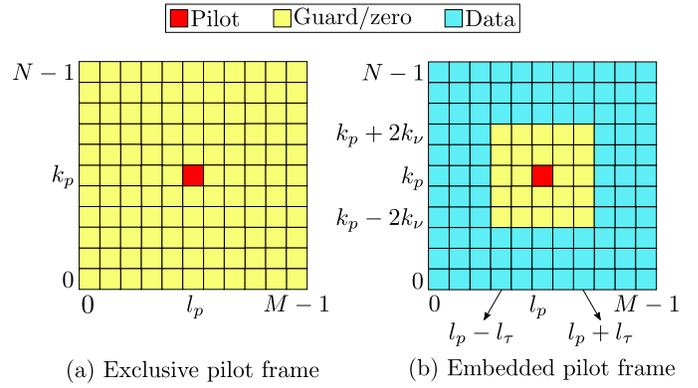


Fig. 2: Pilot, guard, and data symbol placements in exclusive and embedded pilot frames.

The received time domain signal $b(t)$ at the OTFS receiver is given by

$$b(t) = \int_{\nu} \int_{\tau} g(\tau, \nu) a(t - \tau) e^{j2\pi \nu (t - \tau)} d\tau d\nu + w(t), \quad (4)$$

where $w(t)$ represents the additive noise. At the receiver, a match filtering operation is carried out on the received signal $b(t)$ with a receive pulse $p_{rx}(t)$ yielding a TF domain cross-ambiguity function, denoted by $F_{p_{rx}, b}(t, f)$, and given by

$$F_{p_{rx}, b}(t, f) = \int_{t'} p_{rx}^*(t' - t) b(t') e^{-j2\pi f (t' - t)} dt', \quad (5)$$

where $(\cdot)^*$ represents the complex conjugation operation. The transmit and receive pulse are chosen such that they satisfy the biorthogonality condition, i.e., $F_{h_{rx}, h_{tx}}(t, f)|_{t=nT, f=m\Delta f} = \delta(n)\delta(m)$. Sampling (5) at $t = nT$ and $f = m\Delta f$ gives

$$B[n, m] = F_{p_{rx}, b}(t, f)|_{t=nT, f=m\Delta f}. \quad (6)$$

The TF domain signal $B[n, m]$ is then mapped back to the DD domain through SFFT operation to obtain $b[k, l]$ as

$$b[k, l] = \frac{1}{\sqrt{MN}} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} B[n, m] e^{-j2\pi \left(\frac{nk}{N} - \frac{ml}{M} \right)}. \quad (7)$$

Using (1)-(7), the input-output relation of the OTFS modulation scheme in the DD domain can be written as [3]

$$b[k, l] = \sum_{i=0}^{L-1} g'_i a[(k - \beta_i)_N, (l - \alpha_i)_M] + w[k, l], \quad (8)$$

where $g'_i = g_i e^{-j2\pi \tau_i \nu_i}$, α_i is the integer corresponding to the index of delay tap and β_i is the integer corresponding to the Doppler frequency associated with τ_i and ν_i , respectively. Therefore, $\tau_i = \frac{\alpha_i}{M\Delta f}$ and $\nu_i = \frac{\beta_i}{NT}$. Further, (8) can be written in a vectorized form as

$$\mathbf{b} = \mathbf{G}\mathbf{a} + \mathbf{w}, \quad (9)$$

where $\mathbf{b}, \mathbf{a}, \mathbf{w} \in \mathbb{C}^{M \times N}$ and $\mathbf{G} \in \mathbb{C}^{MN \times MN}$ and the $(kM + l)$ th entry of \mathbf{a} , $a_{kM+l} = a[k, l]$ for $k = 0, \dots, N-1, l = 0, \dots, M-1$ and $a[k, l] \in \mathbb{A}$. Likewise, $b_{kM+l} = b[k, l]$ and $w_{kM+l} = w[k, l]$ for $k = 0, \dots, N-1, l = 0, \dots, M-1$. g_i s are assumed to be i.i.d. and are distributed as $\mathcal{CN}(0, 1/L)$.



Fig. 3: Proposed RNN based DDNet channel estimation scheme.

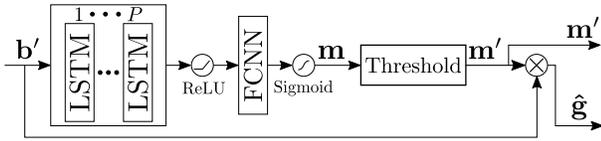


Fig. 4: Proposed RNN based DDNet architecture.

Parameter	Value
Number of LSTM layers	P
LSTM Hidden size (h)	100
LSTM input dimensions	$(c, s, 2)$
LSTM output dimensions	$(c, s, 100)$
FCNN input neurons	100
FCNN output neurons	1

TABLE I: Parameters of the DDNet architecture.

A. Pilot placement schemes

To estimate the DD domain channel matrix \mathbf{G} , known symbols called pilots are placed in the DD domain and transmitted. At the receiver, the received symbols corresponding to the transmitted pilots are used to estimate the channel in the DD domain, which is then used to construct the matrix \mathbf{G} . We consider two types of pilot placement schemes. Figure 2 shows the OTFS frame structure for the two pilot placement schemes. The first scheme is the *exclusive pilot frame* scheme used in [6], wherein the entire DD domain grid consists of a single pilot symbol and zeros elsewhere (see Fig. 2a), i.e.,

$$\mathbf{a} = \begin{cases} 0, & \text{if } k \neq k_p, l \neq l_p \\ a_p, & \text{if } k = k_p, l = l_p, \end{cases} \quad (10)$$

for $k = 0, \dots, N-1$ and $l = 0, \dots, M-1$. A predetermined pilot symbol a_p is placed in the DD grid indexed by k_p and l_p , the Doppler and delay bin indices, respectively. The second scheme is the *embedded pilot frame* scheme used in [7], shown in Fig. 2b, where the DD grid consists of a pilot symbol (marked in red), guard symbols (marked in yellow), and data symbols (marked in blue), which can be represented as

$$\mathbf{a} = \begin{cases} 0, & \text{if } k = k_g, l = l_g \\ a_p, & \text{if } k = k_p, l = l_p \\ a_d, & \text{elsewhere,} \end{cases} \quad (11)$$

for $k = 0, \dots, N-1$ and $l = 0, \dots, M-1$. In (11), k_g s and l_g s denote the indices of guard bands around the pilot symbol a_p , and the remaining indices are occupied by data symbols $a_d \in \mathbb{A}$. The pilot symbol is surrounded by guard symbols to alleviate interference from data symbols. The number of guard symbols are adjusted to accommodate l_τ and k_ν , the delay and Doppler taps corresponding to the largest delay τ_{\max} and Doppler ν_{\max} , respectively [7]. Note that the embedded pilot scheme in (11) becomes exclusive pilot scheme in (10) when $k_g = 0, 1, \dots, k_p - 1, k_p + 1, \dots, N-1$ and $l_g = 0, 1, \dots, l_p - 1, l_p + 1, \dots, M-1$.

III. DDNET - PROPOSED RNN BASED DD CHANNEL ESTIMATOR

In this section, we present the proposed DDNet, an RNN based architecture for DD channel estimation, and the train-

ing methodology. Figure 3 shows the block diagram of the proposed DDNet. The information symbols $a[k, l]$ s in the DD domain are converted to a time domain signal $a(t)$ at the OTFS transmitter and transmitted through a doubly-selective fading channel. At the OTFS receiver, the received signal $b(t)$ is converted back to DD symbols $b[k, l]$ s, $k = 0, \dots, N-1$, $l = 0, \dots, M-1$, given by (8). Let $b'[k', l']$ s, a subset of $b[k, l]$ s, denote the received DD symbols corresponding to the pilot and guard bins, where $k' = k_p - k_\nu, k_p - k_\nu + 1, \dots, k_p + k_\nu$ and $l' = l_p, l_p + 1, \dots, l_p + l_\tau$ for the scheme in (11) and $k' = 0, 1, \dots, N-1$, $l' = 0, 1, \dots, M-1$ for the scheme in (10). The number of symbols in $b'[k', l']$ s is $(2k_\nu + 1)(l_\tau + 1)$ for pilot scheme in (11) and MN for pilot scheme in (10), respectively. These $b'[k', l']$ symbols are converted to a vector \mathbf{b}' of length $(2k_\nu + 1)(l_\tau + 1)$ for scheme in (11) and of length MN for the scheme in (10). The input to the DDNet is the vector \mathbf{b}' . The output of the DDNet is a vector \mathbf{m} , called mask, of length $(2k_\nu + 1)(l_\tau + 1)$ for scheme in (11) and MN for scheme in (10). Entries in \mathbf{m} are values between 0 and 1. These entries are thresholded such that values above 0.5 are replaced by 1 and those below are replaced by 0. In the thresholded \mathbf{m} vector, denoted by \mathbf{m}' , the indices corresponding to location of 1s denote presence of valid channel paths at those locations in the DD grid. These locations are used to obtain the estimates of the integers corresponding to delay taps (i.e., $\hat{\alpha}_i$ s) and Doppler frequencies ($\hat{\beta}_i$ s) (see (8)) in the DD grid. Finally, the vector \mathbf{m}' is element-wise multiplied with the input vector \mathbf{b}' and the non-zero values from the resulting vector are returned as DD domain channel coefficient vector $\hat{\mathbf{g}}$. Using the estimates $\hat{\mathbf{g}}$, $\hat{\alpha}$, and $\hat{\beta}$, the estimated DD domain channel matrix $\hat{\mathbf{G}}$ is obtained. This matrix is used for detection of data symbols.

A. Architecture

The architecture of the proposed DDNet block is shown in Fig. 4. The architecture consists of P layers of long short-term memory (LSTM) [17], a variant of RNN. The output of the LSTM layers is passed through a ReLU activation function, given by $\text{ReLU}(x) = \max(0, x)$, $\forall x \in (-\infty, \infty)$. This is then passed on to a fully connected neural network (FCNN) with one layer. The FCNN is employed to reduce the dimension of the output of the LSTM network to the required

Parameter	Value
Epochs	20000
Optimizer	Adam
Learning rate	0.001, divide by 2 every 4000 epochs
Batch size	1000
Mini-batch size	64

TABLE II: Hyper-parameters used for training the DDNet.

dimension. This is then followed by a sigmoid activation function, given by $\text{sigmoid}(x) = \frac{1}{1+e^{-x}}$, $\forall x \in (-\infty, \infty)$. The purpose of using a sigmoid function is to restrict the output between 0 and 1 and to determine if a particular DD bin contains a valid path. To achieve this, as mentioned above, the mask \mathbf{m} at the output of sigmoid function is first thresholded to obtain the vector \mathbf{m}' , followed by element-wise multiplication with the input. The non-zero entries in the resulting vector constitute the estimated DD channel coefficients, denoted by $\hat{\mathbf{g}}$. The other parameters of the DDNet architecture are given in Table I. The variable c refers to the batch size and s is the sequence length, which is set to be MN for the pilot scheme in (10) and $(2k_\nu + 1)(l_\tau + 1)$ for pilot scheme in (11).

B. Training methodology

Training data is obtained by generating multiple DD domain OTFS frames using pilot schemes in (10) and (11). Further, various guard band realizations are also added to the training data for the scheme in (11). These frames are converted to time domain and transmitted through a doubly-selective channel and the received signal is converted back to DD domain. From the received DD symbols, depending on the pilot scheme employed, s -length vector \mathbf{b}' (see Fig. 4) is obtained. The real and imaginary parts of this vector are concatenated before being fed to the DDNet. For training the DDNet, the ground truth is obtained by generating s -length true mask, denoted by \mathbf{z} , whose i th entry is defined as

$$z_i = \begin{cases} 1, & \text{if DD bin corresponding to } i \text{ is a valid path} \\ 0, & \text{else.} \end{cases} \quad (12)$$

During training, the weights of the DDNet are updated such that the value of the binary cross entropy (BCE) loss function between \mathbf{z} and the output of the DDNet, \mathbf{m} , is minimized. The BCE loss function for the i th index is given by

$$L(z_i, m_i) = -z_i \log(m_i) - (1 - z_i) \log(1 - m_i), \quad (13)$$

where $0 \leq m_i \leq 1$ is the output of the DDNet and $z_i \in \{0, 1\}$ is the ground truth. The other hyper-parameters used in the training of the DDNet are presented in Table II. Note that this training needs to be carried out offline, only once. Subsequently, the network weights are frozen. New channel estimates are obtained from pilots in each OTFS frame using the same trained network.

C. Inference from DDNet

Once the DDNet is trained, the weights are frozen. During the inference (testing) phase, channel estimates, $\hat{\mathbf{g}}$, in the DD domain are obtained through element-wise multiplication of

the input (\mathbf{b}') with the thresholded mask (\mathbf{m}'), as shown in Fig. 4. To obtain the estimates of α and β , denoted by $\hat{\alpha}$ and $\hat{\beta}$, respectively, the following steps are followed for the pilot scheme in (10). Let \mathcal{J} denote the set of indices where the thresholded mask \mathbf{m}' is 1, i.e., $\mathcal{J} = \{j : m'_j = 1, j = 0, 1, \dots, s-1\}$. Then, for the i th path index,

$$\hat{\alpha}_i = (\mathcal{J}_i)_M - l_p, \quad (14)$$

$$\hat{\beta}_i = \left\lfloor \frac{\mathcal{J}_i}{M} \right\rfloor - k_p. \quad (15)$$

For the pilot system in (11) where $s = (2k_\nu + 1)(l_\tau + 1)$, the thresholded mask \mathbf{m}' is reshaped into a matrix of shape $(2k_\nu + 1) \times (l_\tau + 1)$. Vectors \mathbf{u} and \mathbf{v} are defined as $\mathbf{u} = [k_p - k_\nu, k_p - k_\nu + 1, \dots, k_p + k_\nu]$ and $\mathbf{v} = [l_p, l_p + 1, \dots, l_p + l_\tau]$. Then, an index set, \mathcal{I} , is defined as $\mathcal{I} = \{u_i \times M + v_j : m'_{ij} = 1, i = 0, 1, \dots, 2k_\nu, j = 0, 1, \dots, l_\tau\}$. For the i th path index,

$$\hat{\alpha}_i = (\mathcal{I}_i)_M - l_p, \quad (16)$$

$$\hat{\beta}_i = \left\lfloor \frac{\mathcal{I}_i}{M} \right\rfloor - k_p. \quad (17)$$

We have carried out the simulations for performance evaluation using PyTorch machine learning library [18] on RTX 3090 GPU platform.

IV. RESULTS AND DISCUSSIONS

In this section, we present the performance of the proposed DDNet for DD channel estimation in OTFS. A carrier frequency of $f_c = 4$ GHz and a subcarrier spacing of $\Delta f = 15$ kHz are considered. We consider the Vehicular A (VehA) channel model defined by ITU-R [19] with $L = 6$ paths and a maximum speed of 220 km/h. This speed at 4 GHz carrier frequency corresponds to a maximum Doppler shift, ν_{\max} , of 815 Hz. Each path has a Doppler shift generated using Jakes model $\nu_i = \nu_{\max} \cos \theta_i$, where θ_i is assumed to be uniformly distributed between $[-\pi, \pi]$. We fix the number of Doppler bins (N) and delay bins (M) to be 12 and 64, respectively. A BPSK symbol +1 is used as the pilot symbol and data symbols are chosen from 4-QAM alphabet. To train the network, the batch size (c) is chosen to be 1100 of which 1000 OTFS frames are used for training and 100 frames are used for validating the training. This training data is refreshed every 20 epochs, wherein the pilot schemes in (10) and (11) are chosen randomly with equal probability.

To evaluate the accuracy of the channel estimates provided by the DDNet, we evaluate the normalized mean square error (NSME) for the DD domain channel matrix. The value of NMSE is computed as follows. The estimates of the channel coefficients, delay taps, and Doppler taps are obtained from the DDNet as described in Sec. III-C. Using these values, an estimate for the matrix \mathbf{G} (see (9)), denoted by $\hat{\mathbf{G}}$, is obtained. The NMSE is computed as $\text{NMSE} = \mathbb{E} \left[\frac{\|\mathbf{G} - \hat{\mathbf{G}}\|_F^2}{\|\mathbf{G}\|_F^2} \right]$. For evaluating the bit error rate (BER) performance, the message passing (MP) detector in [3] is used.

Effect of number of LSTM layers: Figure 5 shows the NMSE performance of the DDNet as a function of pilot SNR for three

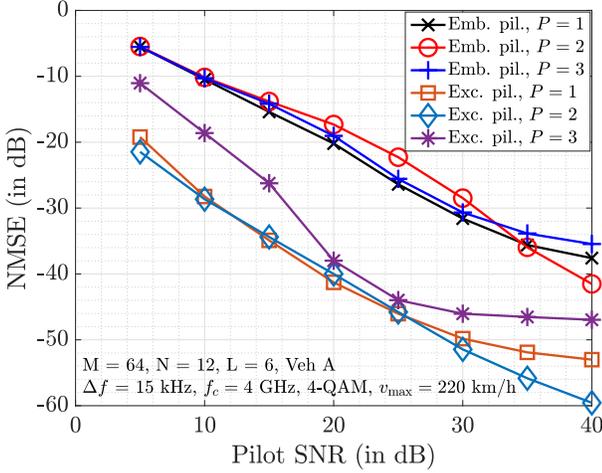


Fig. 5: Effect of number of LSTM layers, P , on the NMSE performance of the proposed DDNet.

different values of the number of LSTM layers, $P = 1, 2, 3$. Other than the value of P , the same parameters in Table I and the training hyper-parameters in Table II are used for all the values of P . The number of parameters for P layers can be computed as $N_P = 4h^2(2P - 1) + 4hi_d + 8Ph + 101$, where h is the hidden size (see Table I), $i_d = 2$ is the input dimension, and 101 is the number of parameters in the FCNN layer. Therefore, the number of parameters to be learnt are 41701, 122501, and 203301 for $P = 1, 2$, and 3, respectively. Performance of embedded as well as exclusive pilot frames are shown. From Fig. 5, it can be seen that while the NMSE performance is comparable for different values of P , the performance for $P = 3$ is slightly worse. This can be attributed to the steep increase in the number of parameters that need to be learnt for $P = 3$, resulting in difficulty in training the network. As a good balance between training complexity and achieved performance, we fix $P = 2$ for the rest of the performance evaluation experiments. During the inference (testing) stage, only 301 floating point operations (FLOPs) are required to compute the mask from the DDNet. In contrast, the approach in [7] does not involve an offline training phase. Further, the number of FLOPs required is $5(2k_v + 1)(l_\tau + 1)$.

NMSE vs spectral efficiency: Figure 6 shows the effect of number of guard symbols on the NMSE performance of the DDNet. It shows NMSE as a function of spectral efficiency η , where η is defined as $\eta = 1 - \frac{N_g}{MN}$, and N_g is the number of guard symbols in the frame. The pilot SNRs considered are 20 dB and 30 dB. The performance of the threshold based scheme in [7] is also plotted for comparison. It can be seen that the DDNet achieves significantly better NMSE performance compared to the threshold based scheme in [7]. Also, while the NMSE of DDNet improves as the number of guard symbols N_g is increased (i.e., smaller values of η), the NMSE of threshold based scheme does not improve because of the inherent limitation in using a fixed threshold, for a given pilot SNR. Whereas, the DDNet is able to generalize for

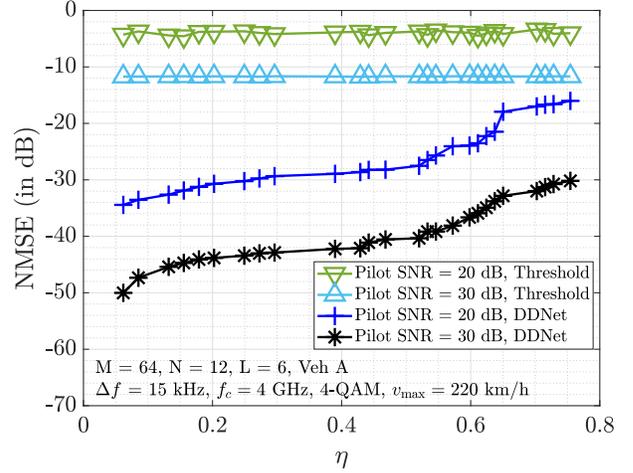


Fig. 6: NMSE vs spectral efficiency at different pilot SNRs.

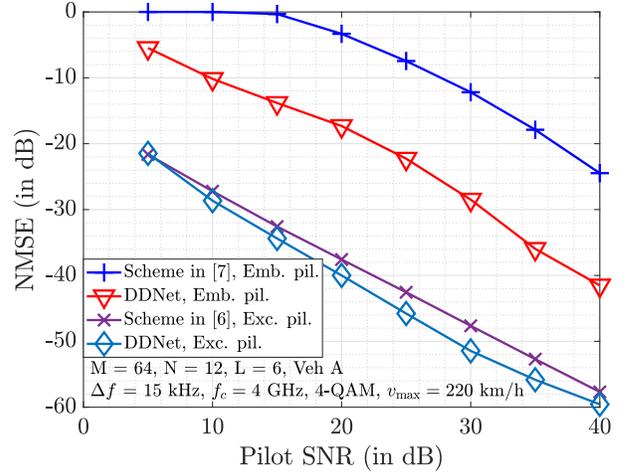


Fig. 7: NMSE performance comparison between the proposed DDNet and the estimation schemes in [6] (exclusive pilot) and [7] (embedded pilot).

varying guard band sizes because of the training methodology employed.

NMSE performance comparison with [6] and [7]: Figure 7 shows the NMSE comparison for the cases of exclusive pilot frame and embedded pilot frame. For the exclusive pilot case, comparison is made between DDNet and the scheme in [6]. For the embedded pilot case, comparison is made between DDNet and the scheme in [7] for $\eta = 0.97$. The following observations can be made from Fig. 7. It is seen that in the exclusive pilot case, DDNet performance is better compared to that of the scheme in [6]. For example, to achieve an NMSE of -50 dB, DDNet requires about 4 dB less pilot SNR. In the embedded pilot case, the DDNet performance is far superior compared to that of the scheme in [7]. The scheme in [7] does not work well up to 15 dB pilot SNR because of the erroneous estimation of the delay and Doppler indices at these SNRs. For pilot SNRs greater than 15 dB, the NMSE is seen to reduce with pilot SNR. Even in this SNR region, there is a significant

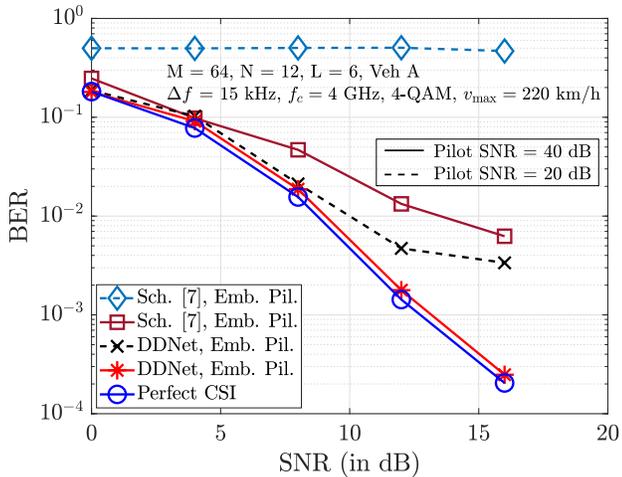


Fig. 8: BER performance comparison between the proposed DDNet and the thresholding scheme in [7] (embedded pilot).

performance advantage for DDNet. For example, to achieve an NMSE of -20 dB, DDNet requires about 14 dB less pilot SNR.

BER performance comparison with [7]: In Fig. 8, we present a comparison between the BER performance of the proposed DDNet and that of the scheme in [7] with embedded pilot frame with $\eta = 0.97$. The BER performance with perfect channel state information (CSI) is also presented for comparison. It is seen that the DDNet outperforms the scheme in [7] by a large margin. For example, at 40 dB pilot SNR, there is about 4 dB advantage for DDNet at a BER of 10^{-2} . Also, the DDNet performance at this pilot SNR is close to that with perfect CSI. When the pilot SNR is 20 dB, the scheme in [7] fails to perform, whereas the DDNet performs much better. This corroborates with the NMSE performance advantage predicted in Fig. 7, where the scheme in [7] has a high NMSE value at 20 dB pilot SNR.

V. CONCLUSIONS

We proposed DDNet, a multi-layer LSTM based learning network for the purpose of DD channel estimation in OTFS systems. The network was trained using a training methodology that enabled the same trained network to work seamlessly for different guard band sizes. This means the same trained network can work for both exclusive pilot frames and embedded pilot frames. The proposed DDNet was shown to achieve significantly better NMSE and BER performance compared to the thresholding scheme known in the literature. This shows that learning approach is a promising approach for DD channel estimation in OTFS. Further, learning based approach for joint channel estimation and signal detection in OTFS with superimposed pilot frames can be taken up for future work.

REFERENCES

[1] R. Hadani, S. Rakib, M. Tsatsanis, A. Monk, A. J. Goldsmith, A. F. Molisch, and R. Calderbank, "Orthogonal time frequency space modulation," *Proc. IEEE WCNC'2017*, pp. 1-6, Mar. 2017.

[2] K. R. Murali and A. Chockalingam, "On OTFS modulation for high-Doppler fading channels," *Proc. ITA*, pp. 1-10, Feb. 2018.

[3] P. Raviteja, K. T. Phan, Y. Hong, and E. Viterbo, "Interference cancellation and iterative detection for orthogonal time frequency space modulation," *IEEE Trans. Wireless Commun.*, vol. 17, no. 10, pp. 6501-6515, Oct. 2018.

[4] S. S. Das and R. Prasad, *OTFS: Orthogonal Time Frequency Space Modulation A Waveform for 6G*, River Publishers, 2021.

[5] A. Naikoti and A. Chockalingam, "Signal detection and channel estimation in OTFS," *ZTE Commun.*, vol. 19, no. 4, pp. 16-33, Dec. 2021.

[6] M. K. Ramachandran and A. Chockalingam, "MIMO-OTFS in high-doppler fading channels: signal detection and channel estimation," *Proc. IEEE GLOBECOM'2018*, pp. 206-212, Dec. 2018.

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

[8] V. Kumar Singh, M. F. Flanagan, and B. Cardiff, "Maximum likelihood channel path detection and MMSE channel estimation in OTFS systems," *Proc. IEEE VTC'2020*, pp. 1-5, Feb. 2021.

[9] H. B. Mishra, P. Singh, A. K. Prasad, and R. Budhiraja, "OTFS channel estimation and data detection designs with superimposed pilots," in *IEEE Trans. Wireless Commun.*, vol. 21, no. 4, pp. 2258-2274, Apr. 2022.

[10] W. Yuan, S. Li, Z. Wei, J. Yuan, and D. W. K. Ng, "Data-aided channel estimation for OTFS systems with a superimposed pilot and data transmission scheme," in *IEEE Wireless Commun. Lett.*, vol. 10, no. 9, pp. 1954-1958, Sep. 2021.

[11] 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.

[12] S. Srivastava, R. K. Singh, A. K. Jagannatham, and L. Hanzo, "Bayesian learning aided sparse channel estimation for orthogonal time frequency space modulated systems," *IEEE Trans. Veh. Tech.*, vol. 70, no. 8, pp. 8343-8348, Aug. 2021.

[13] N. Shlezinger, Y. C. Eldar, N. Farsad, and A. J. Goldsmith, "ViterbiNet: Symbol detection using a deep learning based Viterbi algorithm," *Proc. IEEE SPAWC'2019*, pp. 1-5, Jul. 2019.

[14] H. Ye, G. Y. Li, and B. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114-117, Feb. 2018.

[15] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep learning-based channel estimation," *IEEE Commun. Lett.*, vol. 23, no.4, pp. 652-655, Apr. 2019.

[16] S. R. Mattu and A. Chockalingam, "Learning based channel estimation and phase noise compensation in doubly-selective channels," *IEEE Commun. Lett.*, doi: 10.1109/LCOMM.2022.3155186 (early access), Mar. 2022.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, Nov. 1997.

[18] A. Paszke *et al.*, "PyTorch: an imperative style high-performance deep learning library," *NeurIPS'2019*, pp. 1-12, Dec. 2019.

[19] ITU-R M.1225, "Guidelines for the evaluation of radio transmission technologies for IMT-2000," International Telecommunication Union Radio communication, 1997.