

Adaptive Neural Network for Eigen-Decomposition of Multi-dimensional Channel Kernels

Iresha Amarasekara, Zhibin Zou and Aavek Dutta
 Department of Electrical and Computer Engineering
 University at Albany SUNY, Albany, NY 12222 USA
 {iamarasekara, zzou2, adutta}@albany.edu

Abstract—Eigenfunctions are widely used to characterize kernels in many data-driven analyses. In machine learning, eigenfunction decomposition is primarily based on Mercer’s theorem, which requires the kernel to be symmetric. This is difficult to satisfy in communication systems as the channel kernel is usually asymmetric due to the different downlink and uplink propagation environments. High Order Generalized Mercer’s Theorem (HOGMT) provides a principled way to decompose any multi-dimensional asymmetric kernel into eigenfunctions. To manage the complexity of the eigen-decomposition, we propose an equivalent Neural Network (NN) for decomposing a general channel kernel. This is further improved by applying the Augmented Lagrangian Method (ALM) to reduce the training time and parameter tuning, which avoids additional tuning rounds when the size of the kernel or the number of eigen-components change depending on the wireless environment. We validate the adaptability of the proposed NN and its accuracy using simulations in PyTorch. The code is available at <https://github.com/ZBZou/HOGMT-ALM/tree/main>.

Keywords—Eigen-decomposition, Adaptive Neural Networks.

I. INTRODUCTION

Eigen-decomposition is at the core of many communication system problems such as equalization, channel characterization and Channel State Information (CSI) feedback. In the Multiple-Input Multiple-Output (MIMO) systems, Singular Value Decomposition (SVD)-based precoding decomposes the spatial channel matrix into eigenvectors that are used as parallel channels to achieve sum rate capacity [1]. However, spatial channel matrices fail to capture joint interference over multiple dimensions¹. Therefore, in Linear Time-Varying (LTV) channels *designing waveforms using eigenfunctions of the channel is known to be optimal* [2]. However, eigen-decomposition by Mercer’s theorem is only limited to the symmetric kernels², which is not always satisfied as the wireless channel kernel is a random operator [5].

Multi-dimensional kernels and its decomposition: The general wireless channel is represented as an *asymmetric kernel* from first principles [6], [7], which reflects the mapping relation from the transmitter and the receiver. Recently, High Order Generalized Mercer’s Theorem (HOGMT) [8] has been proposed as a mathematical technique to decompose *multi-dimensional asymmetric kernel* into jointly orthogonal

¹In the context of the channel representation, multi-dimension includes time, space, frequency, delay-Doppler domains, etc.

²A kernel satisfying $K(x, y) = K(y, x)$ is considered as symmetric. Otherwise, it is an asymmetric kernel [3], [4].

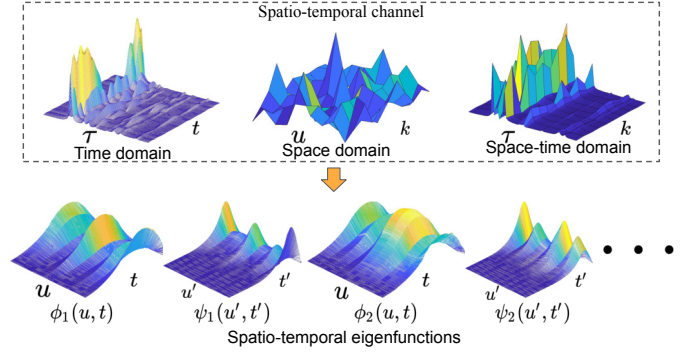


Figure 1: Partial representations of the spatio-temporal channel, and the decomposed spatio-temporal eigenfunctions.

eigenfunctions. This enables precoding and modulation using eigenfunctions to transmit symbols over independent orthogonal subchannels in the eigen-domain, avoiding interference from other symbols across all the DoF. Figure 1 shows an example of eigen-decomposition by HOGMT, where “time-domain” shows the time-delay profile, the space-domain shows the channel gain between MIMO antennas and the space-time-domain shows the delay profile across antennas. Unlike 1-dimensional eigenvectors in the MIMO (spatial) channel matrix, the spatio-temporal eigenfunctions are represented by 2-dimensional waveforms to achieve joint orthogonality in the time-space-domain. In contrast to Fourier bases, commonly used in OFDM and OTFS, eigenfunctions are not pre-defined, resulting in the following open problem [5, Chapter 2.5]: “Eigenfunctions are strictly unstructured, which makes them computationally expensive for practical purposes.”

Neural Network (NN) offers a distinct edge over linear methods as it inherently captures non-linear relationships among hidden variables with practical computational complexity. This encourages us to adopt NNs for the implementing eigen-decomposition by HOGMT. Current NN-based eigen-decomposition methods are based on Mercer’s theorem or SVD, which provide eigenfunctions of symmetric kernels or the eigenvectors of a spatial channel matrix [9]. Consequently, these methods are incapable of decomposing multi-dimensional asymmetric channel kernels, as seen in rapidly time-varying MIMO channel kernels, MIMO-OFDM channel kernels, MIMO-OTFS channel kernels [10]. Meanwhile, existing NNs often treat constraints of a given optimization problem as a fixed penalty, which suffer from exhaustive tuning when

the size of the channel kernel and the number of eigenfunctions change. In this work, we adopt the Augmented Lagrangian Method (ALM) [11] to automatically adapt the Lagrange multiplier during the training process, making the NN adapt to various wireless scenarios. Therefore, the contributions of this paper are as follows:

- We design a baseline NN (HOGMT-Base) to decompose multi-dimensional channel kernels based on HOGMT.
- We improve the baseline model to HOGMT-ALM by incorporating ALM to adapt to different kernel types and number of eigenfunctions.
- We provide a complexity comparison with the SoTA to show the efficiency of HOGMT-ALM.
- We define a metric, *soft orthogonality* to measure the degree of the orthogonality of the eigenfunctions. This is used to evaluate the accuracy of the models in PyTorch.

II. PRELIMINARIES AND BACKGROUND

A. General Channel Kernel

The time and space-domain continuous input-output relations for LTV channel [12] is given by,

$$r(t) = \int g(t, \tau) s(t - \tau) d\tau; r(u) = \int h(u, k) s(u - k) dk \quad (1)$$

where $g(t, \tau)$ and $h(u, k)$ are transfer functions in the time and space-domains. By Fourier transform, the time-domain can be transferred to the frequency-domain resulting in the transfer function for OFDM. However, the general form of the input-output relation of LTV channels is expressed as,

$$r(Z) = \int H(Z; \Gamma) s(Z - \Gamma) d\Gamma = \int K(Z; Z') s(Z') dZ' \quad (2)$$

where $H(Z; \Gamma)$ is the multi-dimensional transfer function (e.g., in the time-varying MIMO channel: $(Z; \Gamma) = (u, t; k, \tau)$, while in the MIMO-OFDM system, $(Z; \Gamma) = (u, f; k, \nu)$). $K(Z, Z') = H(Z, Z - Z')$ is the general channel kernel as defined in [6], [12]. Note that $K(Z, Z')$ is considered asymmetric for the general channel, i.e., $K(X, Y) \neq K(Y, X)$.

B. High Order Generalized Mercer's Theorem

Kernel decomposition Mercer's theorem is formulated as,

$$K(t, t') = \sum_{n=1}^{\infty} \lambda_n \phi_n(t) \phi_n(t') \quad (3)$$

where, ϕ_n denotes the eigenfunction and λ_n is the corresponding eigenvalue. HOGMT generalizes Mercer's Theorem to multi-dimensional asymmetric kernels in (2), into low-dimension, jointly orthogonal eigenfunctions expressed as,

$$K(Z; Z') = \sum_{n=1}^{\infty} \sigma_n \psi_n(Z) \phi_n(Z') \quad (4)$$

where $\mathbb{E}\{\sigma_n \sigma_{n'}\} = \lambda_n \delta_{nn'}$. λ_n is the n -th eigenvalue and $\psi_n(Z)$ and $\phi_n(Z')$ are orthonormal eigenfunctions, i.e.,

$$\int \phi_n(Z') \phi_{n'}^*(Z') dZ' = \delta_{nn'}; \int \psi_n(Z) \psi_{n'}^*(Z) dZ = \delta_{nn'} \quad (5)$$

These eigenfunctions are referred as *dual eigenfunctions* that exhibit the important *duality* property,

$$\int K(Z; Z') \phi_n^*(Z') dZ' = \sigma_n \psi_n(Z) \quad (6)$$

This property indicates that the eigenfunctions are transferred to their dual eigenfunction scaled only by the channel gains when transmitting over the channel.

C. Augmented Lagrangian Method

ALM [11] is an adaptive technique to solve an equality-constrained optimization problem. Consider an optimization problem under M constraints

$$\text{minimize } F(x) \quad \text{s.t. : } c_i(x) = 0, \forall i = 1, 2, \dots, M \quad (7)$$

where $x \in \mathbb{R}^n$. The corresponding Lagrangian function is,

$$L(x, \alpha) \triangleq F(x) + \sum_{i=1}^M \alpha_i c_i(x) \quad (8)$$

where $\alpha \triangleq [\alpha_1, \alpha_2, \dots, \alpha_M]^T$ are the Lagrange multipliers. ALM modifies (7) in to the following problem,

$$\begin{aligned} &\text{minimize } F(x) + \frac{\mu}{2} \|c(x)\|^2 \\ &\text{subject to: } c_i(x) = 0, \forall i = 1, 2, \dots, M \end{aligned} \quad (9)$$

where, μ is the penalty parameter and $c(x) \triangleq [c_1(x), c_2(x), \dots, c_M(x)]^T \in \mathbb{R}^M$. Therefore, the Lagrangian function of (9) is given by,

$$L_\mu(x, \alpha) \triangleq F(x) + \frac{\mu}{2} \|c(x)\|^2 + \sum_{i=1}^M \alpha_i c_i(x) \quad (10)$$

(10) is called the augmented Lagrange function. It can be shown that, both problems (7) and (9) share the same optimum solution x^* and the optimum Lagrange multipliers α^* [11]. In all, ALM transforms the constrained optimization in (7) into an unconstrained problem in (10).

III. KERNEL DECOMPOSITION WITH NN

A. Approximating Kernel with Finite Eigenfunctions

Theoretically, both Mercer's Theorem and HOGMT decompose the kernel into infinite eigenfunctions. However, we can only operate a finite number of eigenfunctions in reality. Therefore, we convert the kernel decomposition problem to the approximation problem by N eigenfunctions as

$$\arg \min_{\hat{K}} \mathbb{E}\{\|K(Z; Z') - \hat{K}(Z; Z')\|^2\} \quad (11)$$

where $\hat{K}(Z; Z')$ is the approximated kernel as

$$\hat{K}(Z; Z') = \sum_{n=1}^N \sigma_n \hat{\psi}_n(Z) \hat{\phi}_n(Z') \quad (12)$$

with the orthogonal constraints

$$\int \hat{\phi}_n(Z') \hat{\phi}_{n'}^*(Z') dZ' = \delta_{nn'}; \int \hat{\psi}_n(Z) \hat{\psi}_{n'}^*(Z) dZ = \delta_{nn'} \quad (13)$$

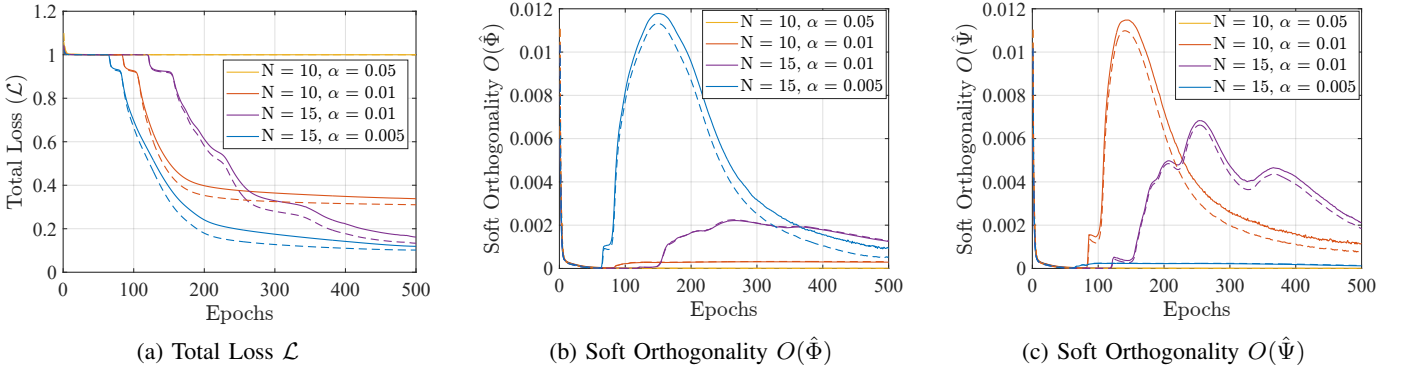


Figure 2: Performance of HOGMT-Base changes over number of eigenfunctions N and penalty α . Dashed and solid lines indicate the convergence of the NN during training and testing respectively.

In this work, we focus on asymmetric kernels unless otherwise specified because the symmetric kernel is just a special case of that. Unlike Fourier methods that provide complex exponentials as subcarriers, which are independent in time-invariant channels only, eigenfunctions of the channel kernel remain independent in all types of channels. However, since the eigenfunctions are not pre-defined, computing these require high computational complexity, which limits their practical implementation [13]. We postulate that *the output of a neural network that solves the MMSE problem in (11) converges to eigenfunctions, making it practical.*

B. HOGMT-Base: Implementing HOGMT with NN

We design an equivalent NN, HOGMT-Base with the equality-constrained objective function (14a) to solve (11), whose outputs converge to the eigenfunctions decomposed by HOGMT from the universal approximation theorem [15], [16].

$$\mathcal{L} = \mathcal{J} + \sum_{i=1}^2 \alpha_i \Omega_i \quad (14a)$$

where α_i is the penalty for the constraint Ω_i and $\mathcal{J}$ is the MSE optimization in (11)

$$\mathcal{J} = \frac{1}{B} \sum_{b=1}^B \frac{\|\mathbf{K}_b - \sum_{n=1}^N \hat{\sigma}_{n,b} \hat{\Psi}_{n,b} \otimes \hat{\Phi}_{n,b}\|^2}{\|\mathbf{K}_b\|^2} \quad (14b)$$

where $\mathbf{K}_b$ is the input kernel. B is the batch size. Ω_1 and Ω_2 are the regularizations for the orthogonality constraints in (13),

$$\Omega_1 = \frac{1}{B} \sum_{b=1}^B \sum_{n=1}^N \sum_{n' \neq n}^N \|\langle \hat{\Phi}_{n,b}, \hat{\Phi}_{n',b} \rangle\| \quad (14c)$$

$$\Omega_2 = \frac{1}{B} \sum_{b=1}^B \sum_{n=1}^N \sum_{n' \neq n}^N \|\langle \hat{\Psi}_{n,b}, \hat{\Psi}_{n',b} \rangle\| \quad (14d)$$

Comparing (14a) and (8), $\mathcal{L}$ is a Lagrange function with α_i as multipliers.

The total loss is a general evaluation metric for NNs, which depends on the sum of all eigenvalues for eigen-decomposition [17]. However, in the communication systems, orthogonality is the most critical metric for interference cancellation. However, any NN-based method cannot achieve

strict orthogonality as this hard decision will limit the freedom of the output space and lead to over-fitting. Therefore, we define a new metric, *soft orthogonality* as follows,

Definition 1. *Soft orthogonality of the eigenfunction set $\{\hat{\Phi}\}$,*

$$O(\hat{\Phi}) = \frac{1}{N(N-1)} \sum_{n=1}^N \sum_{n' \neq n}^N \|\langle \hat{\Phi}_n, \hat{\Phi}_{n'} \rangle\| \quad (15)$$

where $O(\hat{\Phi})=0$ indicates that the eigenfunctions obtained from the NN, $\{\hat{\Phi}\}$ are strictly orthogonal. Similarly, $O(\hat{\Psi})$ denotes the soft orthogonality for eigenfunctions $\{\hat{\Psi}\}$.

We validate this baseline model, using spatio-temporal channel kernels, which is detailed in Section V. Figure 2 shows the performance of HOGMT-Base for $N=10$ and $N=15$ cases, with two penalty values for each. Although it shows convergence with respect to both the total loss and the soft orthogonality with appropriate penalties, it has two limitations:

- The choice of the penalty (α_i) greatly affects the performance of HOGMT-Base even for the same value of N . For example, in Figure 2a, for $N=10$, the total loss for $\alpha=0.01$ converges but that for $\alpha=0.05$, does not.
- If N changes, the same penalty cannot ensure convergence. For example, in Figure 2a, 2b and 2c we find that the total loss for $N=10$ and $N=15$ are not comparable for the same $\alpha=0.01$. This will require tuning of the penalty to $\alpha=0.005$ for similar performance.

These limitations motivate us to design an adaptive version of HOGMT-Base, which ensures the convergence of the eigenfunctions without unnecessary tuning for different scenarios.

IV. ADAPTIVE NEURAL NETWORK BASED ON ALM

A. HOGMT-ALM: ALM Based NN for HOGMT

In practice, both the required number of eigenfunctions and the channel environment (such as users) may vary, leading to different kernel sizes and N , which require changes in the NN architecture and tuning of the Lagrange multiplier (penalty), α . The fundamental reason is that any α , is no longer optimal for the Lagrangian function when its optimal solution changes for a different input. Therefore, training HOGMT-Base with a

Table I: Complexity of eigen-decomposition methods. HOGMT-ALM and HOGMT-Base both have the least complexity

Channel Kernel Type	Method	Complexity	Parameters
Spatial channel matrix $H \in \mathbb{C}^{N_t \times N_r}$	SVD	$\mathcal{O}(\min(N_t N_r^2, N_r^2 N_t))$	N : Number of eigen-components N_d : The order of dimensions N_L : Number of layers
	SVD-DNN [9]	$\mathcal{O}(\max(2N^2 N_t N_r, 2N^2 N_t^2, 2N^2 N_r^2))$	
Spatio-temporal channel tensor $K \in \mathbb{C}^{L_u \times L_t \times L_{u'} \times L_{t'}}$	HOSVD [14]	$\mathcal{O}(N_d \max(L_u L_t L_{u'}, L_u L_t L_{t'}, L_t L_{u'} L_{t'})^3)$	
	HOGMT [8]	$\mathcal{O}(\min(L_u L_t (L_{u'} L_{t'})^2, (L_u L_t)^2 L_{u'} L_{t'}))$	
	HOGMT-Base	$\mathcal{O}(2NN_L L_u L_t L_{u'} L_{t'})$	
	HOGMT-ALM	$\mathcal{O}(2NN_L L_u L_t L_{u'} L_{t'})$	

fixed penalty for different kernel types does not guarantee convergence, which requires dynamic adaptation of the Lagrange multipliers during the training. Therefore, we incorporate the ALM method and modify the objective of HOGMT-Base as,

$$\mathcal{L} = \mathcal{J} + \sum_{i=1}^2 A_i^\top \Omega'_i + \frac{\mu}{2} \sum_{i=1}^2 \|\Omega'_i\|^2 \quad (16)$$

where, μ is the penalty parameter,

$$A_i \triangleq [\alpha_{i,1}, \alpha_{i,2}, \dots, \alpha_{i,K}]^\top \in \mathbb{R}^{2K} \quad (17)$$

is the vector containing the Lagrange multipliers and

$$\begin{aligned} \Omega'_1 &\triangleq \frac{1}{B} \sum_{b=1}^B [\Re(\tilde{\Phi}_{1,b}), \Im(\tilde{\Phi}_{1,b}), \dots, \Re(\tilde{\Phi}_{K,b}), \Im(\tilde{\Phi}_{K,b})]^\top \\ \Omega'_2 &\triangleq \frac{1}{B} \sum_{b=1}^B [\Re(\tilde{\Psi}_{1,b}), \Im(\tilde{\Psi}_{1,b}), \dots, \Re(\tilde{\Psi}_{K,b}), \Im(\tilde{\Psi}_{K,b})]^\top \end{aligned} \quad (18)$$

where, $\Re(\cdot)$ and $\Im(\cdot)$ are the real and imaginary parts respectively. $\tilde{\Phi}_{k,b}$ and $\tilde{\Psi}_{k,b}$ are the inner products of one pair of $(\hat{\Phi}_{n,b}, \hat{\Phi}_{n',b})$ and $(\hat{\Psi}_{n,b}, \hat{\Psi}_{n',b})$ for $n \neq n'$ respectively. There are $K=N(N-1)$ pairs for such eigenfunction set.

The main idea behind HOGMT-ALM is to get the optimal solution for the constrained optimization problem (11) by unconstrained optimization of the augmented Lagrangian function (16). This is achieved by iteratively updating the NN model parameters and A_i based on the gradient of $\mathcal{L}$. During training, the model is updated using well-known optimizers like Stochastic Gradient Descent (SGD) or Adam, while A_i is updated as follows,

$$A_i^{k+1} = A_i^k + \mu^k \Omega'_i \quad (19)$$

where μ^k is used to control the learning rate of the model.

The ALM theory requires dynamic update of μ in order to adapt to the current constraint. Otherwise, it leads to either too-small or too-large update rate for A_i . [11] provides a dynamic update criteria for μ . This update step depends on the current conditions of the constraint. Predefined parameters $\beta > 1$ and $\gamma < 1$ ensure that μ increases when the constraints Ω'_1 and Ω'_2 does not decrease over the iterations. Therefore, by coupling the NN training with the update of the Lagrange multipliers in (19) and the parameter μ , HOGMT-ALM can ensure that the Lagrange multipliers are always optimized towards the optimal NN model, which solves the problem of fixed penalty in HOGMT-Base mentioned above.

Algorithm 1 is used for training HOGMT-ALM with an

initialization of the inputs, A_1 , A_2 , μ and γ . Lines 3-9 are steps for kernel processing and parameter calculation in each batch according to (16)-(18). Lines 10-11 updates the Lagrange multipliers according to the constraints and μ , where μ is further updated in lines 12-16.

Algorithm 1 HOGMT-ALM Training

```

1: Inputs  $A_1^{[0]}, A_2^{[0]}, \mu^{[0]}, \gamma$ ;
2: for  $i \leftarrow 0$  to training epochs do
3:   for mini batches  $\leftarrow 0$  to data size/batch size do
4:      $x_b \leftarrow$  split real and imaginary parts of  $X_b$ 
5:      $Y_b = NN\_Model(x_b)$ 
6:     Derive  $\hat{\sigma}_{n,b}$ ,  $\hat{\Phi}_{n,b}$  and  $\hat{\Psi}_{n,b}$  for each eigenfunction
7:     Calculate  $\mathcal{J}$  according to (14b)
8:     Calculate  $\Omega'_1$  and  $\Omega'_2$  according to (18)
9:   end for
10:   $A_1^{[i+1]} \leftarrow A_1^{[i]} + \mu^{[i]} \Omega'_1$ 
11:   $A_2^{[i+1]} \leftarrow A_2^{[i]} + \mu^{[i]} \Omega'_2$ 
12:  if  $\|\Omega'_1\|^{[i]} + \|\Omega'_2\|^{[i]} > \gamma(\|\Omega'_1\|^{[i-1]} + \|\Omega'_2\|^{[i-1]})$  then
13:     $\mu^{[i+1]} \leftarrow \beta \mu^{[i]}$ 
14:  else
15:     $\mu^{[i+1]} \leftarrow \mu^{[i]}$ 
16:  end if
17: end for
    
```

B. Complexity Analysis

The computational complexity for eigenvector or eigenfunction decomposition is shown in Table I. Both SVD and DNN-SVD are designed to decompose 2-D matrices only. While High-Order SVD (HOSVD), HOGMT and proposed methods decompose multi-dimensional tensors. The relationship between SVD, HOSVD and HOGMT can be found in Lemma 3 of [8]. Since both HOGMT-Base and HOGMT-ALM use fully connected architectures (Section V), their time complexities are the same, which depends on the number of layers (N_L), the size of the input ($2L_u L_t L_{u'} L_{t'}$ for a complex-valued NN) and the size of the output (N). It's clear that proposed NNs have less complexity than HOSVD. Meanwhile, for a fixed N and N_L , the complexity of both HOGMT-Base and HOGMT-ALM increase at a much slower rate than HOGMT with increasing size of the input tensor.

V. HOGMT-ALM IMPLEMENTATION

NN architecture: To decompose a spatio-temporal channel kernel, $K(u, t; u', t') \in \mathbb{C}^{L_u \times L_t \times L_{u'} \times L_{t'}}$ with $L \triangleq L_u L_t L_{u'} L_{t'}$

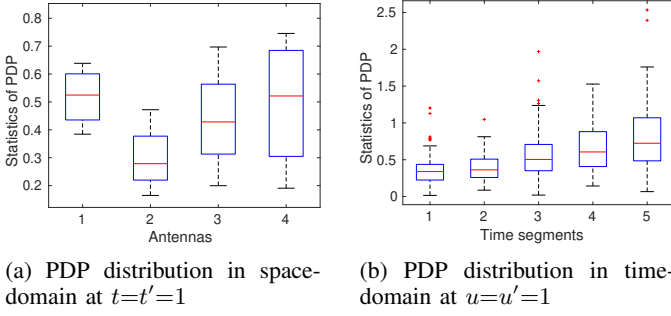


Figure 3: Statistics of PDP change over space and time

elements, the HOGMT-Base is designed with 4 fully connected, feed-forward layers. However, this model requires real inputs and outputs only. As a result, the L elements of the kernel are split into real and imaginary parts at the input. The dimension of each layer is as follows: input and output layers have $2L$ and $\tilde{N} \triangleq N(1+L_u L_t + L_{u'} L_{t'})$ nodes, respectively, while the two hidden layers have $2L$ and $\tilde{L} \triangleq L + \tilde{N}$ nodes. The *LeakyRelu* activation function with a negative slope of 0.1 is used for the hidden linear layers. This is a tuned slope and is fixed for all the NNs in this work. The architectures of HOGMT-ALM and HOGMT-Base are kept the same for correct evaluation and comparison.

Data Generation: We generated 5120 time-varying MIMO channel kernels. Each kernel represents $L_u \times L_{u'}$ antennas for $L_t = L_{t'}$ moments or time indexes, where $L_u = L_{u'} = 4$ and $L_t = L_{t'} = 5$. The delay taps are randomly chosen between 3 and 5. The distribution of the Power Delay Profile (PDP) at each moment changes over space and time as shown in Figure 3a and Figure 3b, respectively. A training-test split of 80%-20% is used to evaluate the model and we do not restrict the dimensions and size of the kernels to show the generality of the eigen-decomposition process.

Training: Figure 4 shows how various training-related parameters are updated in every *batch* and *epoch* while training HOGMT-ALM. The training is done as mini-batches where the chosen mini-batch size is $B=16$. The Adam optimizer with a learning rate of 1×10^{-5} is used for training. Figure 4 shows that there are some adaptive parameters involved in the training, such as A_1 , A_2 and μ , that depend on Ω'_1 and Ω'_2 . These parameters are updated during each epoch, relative to the Ω'_1 and Ω'_2 of the last mini-batch of the previous epoch using (19) for A_1 and A_2 and Algorithm 1 lines 12-16 for μ with $\gamma = 0.75$ and $\beta = 1.01$ and initial value for μ as 0. The initial values of A_i are set to 0 as it provides flexibility in choosing the rest of the parameters.

VI. EVALUATION AND RESULTS

Adapting to varying number of eigenfunctions: We validate the HOGMT-ALM design using the above dataset for $N=10$, $N=15$ and $N=20$ cases with the same initialization of A_1 , A_2 , μ and γ . Figure 5 shows that both total loss and soft orthogonality converge without tuning the penalty, which addresses the limitation (a) above. Further, the convergence

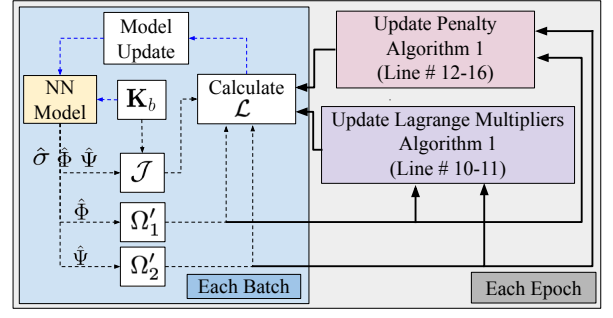


Figure 4: Adaptive Training of HOGMT-ALM, which has two loops: (1) The outer loop (“Each Epoch”) updates the Lagrange multipliers A_i and the penalty parameter μ , and (2) The inner loop (“Each Batch”) updates the NN parameters.

remain unchanged with different N as well, which removes limitation (b). It is also observed that unlike HOGMT-Base, HOGMT-ALM begins to converge around the same time (number of epochs) for all N , which makes the convergent behavior more predictable. From a practical system implementation viewpoint, this property provides a reference for completion time (in epochs) for the training, which eliminates waiting for *uncertain* epochs for the model to converge when N changes. Figure 5a shows that larger N results in smaller values of total loss. This is because, more eigenfunctions can approximate the kernel with higher accuracy [17]. Figure 5b and 5c show that the two soft orthogonality constraints also converge to less than 0.001 within 500 epochs, which cannot be otherwise guaranteed by the baseline model, HOGMT-Base.

Adapting to degenerate cases: The above results show the performance of HOGMT-ALM for 4-D kernels. To compare with the SoTA, SVD-DNN [9], we generate 16×16 spatial channel matrices as the dataset (size = 50000), since SVD is applicable 2-D matrices only. We use the same hyper-parameters for HOGMT-ALM as in the 4-D channel kernel case. The penalty is set at $\alpha=0.01$ for both HOGMT-Base and SVD-DNN, which is empirically the best penalty as shown in Figure 2 for $N=10$ case. In Figure 6, HOGMT-Base achieves similar performance as HOGMT-ALM in this degenerate (low-dimension channel kernel) case with a proper penalty. However, HOGMT-ALM outperforms SVD-DNN indicating that HOGMT-ALM is adaptive to different channel types without tuning the penalty. Further, Figure 6b and Fig 6c show that for SVD-DNN, $O(\hat{\Phi})$ is around 0.0275 and $O(\hat{\Psi})$ is around 0.003 which further increases over epochs. It means SVD-DNN is unable to maintain the orthogonality constraint, which is critical for eigenfunction based waveforms. Specifically, the large soft orthogonality will induce interference as data symbols are not transmitted over independent subchannels. More comparison for larger kernels is shown in the Appendix [18].

VII. DISCUSSION AND APPLICATIONS

The choice of N : As shown in Figure 5a and Table I, N provides a trade-off between the total loss and complexity, where total loss is inversely proportional to the sum of the

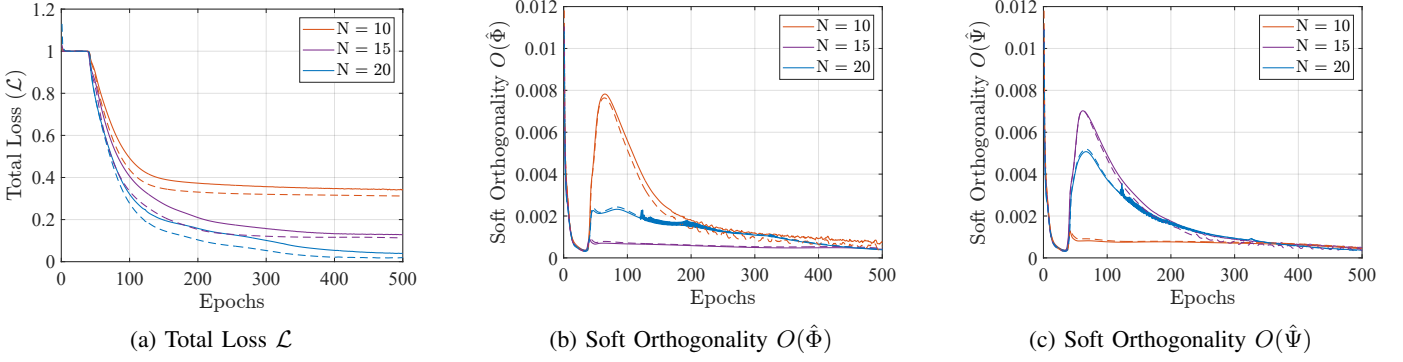


Figure 5: Performance of HOGMT-ALM for 4-D channel kernels with $N=10, 15$, and 20 . Dashed and solid lines indicate the convergence of the NN during training and testing respectively

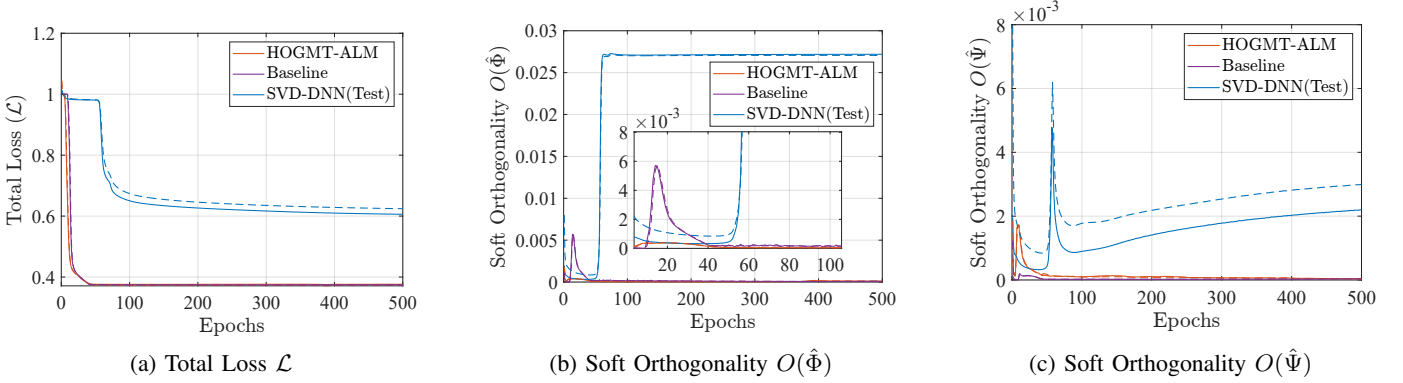


Figure 6: Performance comparison between HOGMT-ALM, HOGMT-Base, and SVD-DNN for 2-D channel kernels with $N=10$. Dashed and solid lines indicate the convergence of the NN during training and testing respectively

eigenvalues according to (11). However, in some applications such as decomposing the channel into orthogonal subchannels and transmitting symbols using those, the gain (given by the eigenvalue) of each subchannel (represented by the eigenfunction) is more important than sum itself. In this case, N can be chosen as the largest integer that satisfies the a specific limit on the eigenvalues, as shown in the example below. The error rate for M-QAM modulated symbol transmitted over the eigenfunction with gain σ_N is³,

$$Pr(M, \gamma, \sigma_N) \approx 4Q\left(\sigma_N \sqrt{3\gamma/(M-1)}\right) \quad (20)$$

where γ is the SNR. Given the desired error probability bound β , then N is chosen as the largest integer satisfy the constraint $Pr(M, \gamma, \sigma_N) < \beta$, i.e.,

$$\sigma_N > Q^{-1}(\beta/4)/\sqrt{3\gamma/(M-1)} \quad (21)$$

which means the error probability of the symbol that is transmitted over the N -th eigenwave should be less than β .

SVD-based Precoding for MIMO channels: The eigen-decomposition is widely applied in precoding for spatial multiplexing in the form of SVD. In practice, both HOGMT-ALM and SVD-DNN can be employed to produce the same eigenfunctions as SVD. However, eigen-decomposition by neural

³The standard approximate error rate for M-QAM modulated symbol is given by [19] as $4Q(\sqrt{3\gamma/(M-1)})$. It is straightforward to derive it for symbols with the subchannel gain σ_N as in (20)

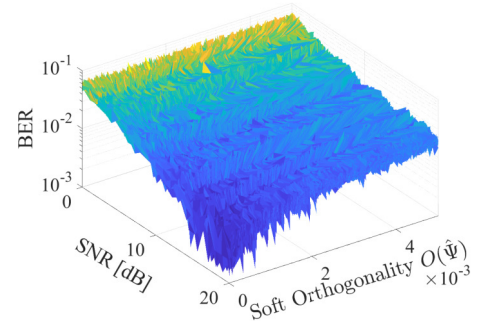


Figure 7: BER of SVD-based Precoding implemented by NNs

networks will lead to a non-zero value of soft orthogonality ($\mathcal{O}(\hat{\Psi})$ in III-B), which varies depending on the objective function enforced by the NN. Figure 7 shows the general behavior of this phenomena for decomposing a 20×20 MIMO channels, where for any given SNR, the BER is lower for smaller values of $\mathcal{O}(\hat{\Psi})$. In Figure 6, we show shows that HOGMT-ALM achieves $\mathcal{O}(\hat{\Psi}) \approx 0$ for decomposing 2-D channel matrices, while SVD-DNN achieves $\mathcal{O}(\hat{\Psi}) \approx 2 \times 10^{-3}$, resulting in much higher BER.

VIII. CONCLUSION

We proposed an equivalent NN called HOGMT-Base for implementing HOGMT, which decomposes the multi-

dimensional asymmetric channel kernel into eigenfunctions. Then we improve it to HOGMT-ALM by incorporating ALM. Both proposed NNs achieve lower approximation error and soft orthogonality than SoTA with lower complexity as shown in Table I, while HOGMT-ALM is adaptive to different kernel types and the number of eigenfunctions. This work solves an important open problem posed in Section I, which creates the opportunity to apply eigenfunctions for real-time, over-the-air communication in fast time-varying wireless channels by employing eigenfunction-based waveforms. The adaptable and computationally efficient NN also paves the way for prototype hardware [19] and experimentation.

IX. ACKNOWLEDGEMENT

This work is funded by the National Science Foundation CAREER Program (Award Number - 2144980).

REFERENCES

- [1] Y. S. Cho, J. Kim, W. Y. Yang, and C. G. Kang, *MIMO-OFDM wireless communications with MATLAB*. John Wiley & Sons, 2010.
- [2] K. Liu, T. Kadous, and A. Sayeed, "Orthogonal time-frequency signaling over doubly dispersive channels," *IEEE Transactions on Information Theory*, vol. 50, no. 11, pp. 2583–2603, 2004.
- [3] J. Mercer, "Functions of Positive and Negative Type, and their Connection with the Theory of Integral Equations," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 209, pp. 415–446, 1909. [Online]. Available: <http://www.jstor.org/stable/91043>
- [4] K. P. Murphy, *Probabilistic machine learning: an introduction*, 2022.
- [5] S. S. Das and R. Prasad, *Orthogonal Time Frequency Space Modulation: OTFS a Waveform for 6G*. CRC Press, 2022.
- [6] P. Bello, "Characterization of Randomly Time-Variant Linear Channels," *IEEE Transactions on Communications Systems*, vol. 11, no. 4, pp. 360–393, 1963.
- [7] L. A. Zadeh, "Frequency analysis of variable networks," *Proceedings of the IRE*, vol. 38, no. 3, pp. 291–299, 1950.
- [8] Z. Zou, M. Careem, A. Dutta, and N. Thawdar, "Joint Spatio-Temporal Precoding for Practical Non-Stationary Wireless Channels," *IEEE Transactions on Communications*, vol. 71, no. 4, pp. 2396–2409, 2023.
- [9] T. Peken, S. Adiga, R. Tandon, and T. Bose, "Deep learning for SVD and hybrid beamforming," *IEEE Transactions on Wireless Communications*, vol. 19, no. 10, pp. 6621–6642, 2020.
- [10] Y. Hong, T. Thaj, and E. Viterbo, *Delay-Doppler Communications: Principles and Applications*. Academic Press, 2022.
- [11] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 2016.
- [12] F. Hlawatsch and G. Matz, *Wireless Communications Over Rapidly Time-Varying Channels*, 1st ed. USA: Academic Press, Inc., 2011.
- [13] K. Liu, T. Kadous, and A. Sayeed, "Orthogonal time-frequency signaling over doubly dispersive channels," *IEEE Transactions on Information Theory*, vol. 50, no. 11, pp. 2583–2603, 2004.
- [14] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.
- [15] K.-I. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Networks*, vol. 2, no. 3, pp. 183–192, 1989. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0893608089900038>
- [16] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.
- [17] Z. Deng, J. Shi, and J. Zhu, "Neuralef: Deconstructing kernels by deep neural networks," in *International Conference on Machine Learning*. PMLR, 2022, pp. 4976–4992.
- [18] I. Amarasekara, Z. Zou and A. Dutta, "Appendix: Adaptive Neural Network for Deconstructing Multi-dimensional Channel Kernels." [Online]. Available: <https://www.dropbox.com/scl/fi/pfblpjehh8fh22mvmuubf/24ICCAppendix.pdf?rlkey=tjcllr4e1v0s6u8ly6u322mjk&dl=0>
- [19] Y. Umuroglu, N. J. Fraser, G. Gambardella, M. Blott, P. Leong, M. Jahre, and K. Vissers, "Finn: A framework for fast, scalable binarized neural network inference," in *Proceedings of the 2017 ACM/SIGDA international symposium on field-programmable gate arrays*, 2017, pp. 65–74.