

Blind Beamforming via Deep Learning-Based Signal Classification and Transfer Learning

Michael Wentz¹, Jack Capper², Binoy Kurien², Keith Forsythe², and Kaushik Chowdhury³, *Fellow, IEEE*

Abstract—We propose a new cognitive technique for blind adaptive beamforming which uses a pre-trained deep learning-based signal classifier to protect a signal of interest (SOI) from interference. The method is an application of transfer learning since the features learned from noisy single-antenna captures are transferred to the task of beamforming using an antenna array. The proposed technique, called the classification-based transfer learning (CBTL) beamformer, forms a beam towards the SOI and nulls towards interference by maximizing a measure of classification confidence for the SOI. It does not require synchronization to the SOI or knowledge of the array manifold. We describe how the beamformer is optimized and the desired characteristics of the classifier. Extensive simulations show that the CBTL beamformer can improve the array signal-to-interference-plus-noise ratio by 8 dB or more compared to state-of-the-art techniques when using 64 samples from a 4-element array, which can be useful for dealing with dynamic environments. We validate this result using real-world data from an open-access radio testbed. The novelty of the proposed approach is in its ability to repurpose a classifier for a task that it was not trained for, eliminate the need for training data from an antenna array, and automate the development of beamformers without relying on expert-designed features.

Index Terms—Array processing, blind beamforming, interference suppression, signal classification, transfer learning.

I. INTRODUCTION

THE proliferation of wireless devices and increased demands for bandwidth have led to crowded radio frequency (RF) environments and the need for receivers to reject interference rather than simply avoiding it. Uncooperative transmissions from co-channel users can degrade communications and sensing performance and are typically difficult to identify and resolve. Receivers with a large field of view, such as those used with aircrafts, satellites, and skywave propagation, can be especially prone to interference. Recent examples with worldwide attention include the effects of 5G cellular networks in C-band on essential services such as radar altimeters and satellite Earth stations [2].

24 June 2025. This work was supported in part by the United States Air Force under Contract FA8702-15-D-0001 or FA8702-25-D-B002, in part by the Department of the Air Force Artificial Intelligence Accelerator under Cooperative Agreement FA8750-19-2-1000, and in part by the National Science Foundation under Grant ECCS-2516080. An earlier version of this paper was presented in part at the 2024 IEEE Consumer Communications and Networking Conference (CCNC) [1]. (Corresponding author: Michael Wentz)

Michael Wentz is with the Institute for the Wireless Internet of Things, Northeastern University, Boston, MA 02115 USA, and with MIT Lincoln Laboratory, Lexington, MA 02421 USA (e-mail: wentz.m@northeastern.edu).

Jack Capper, Binoy Kurien, and Keith Forsythe are with MIT Lincoln Laboratory, Lexington, MA 02421 USA (e-mail: jack.capper@ll.mit.edu; bkurien@ll.mit.edu; forsythe@ll.mit.edu).

Kaushik Chowdhury is with the Chandra Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712 USA (e-mail: kaushik@utexas.edu).

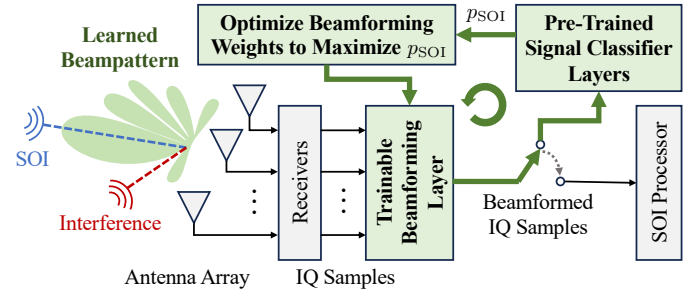


Fig. 1. Block diagram of a system using the proposed CBTL beamformer, which optimizes its weights to put gain on the SOI and null out interference based on the confidence measure p_{SOI} from the pre-trained signal classifier.

Adaptive antenna array processing can protect a signal of interest (SOI) from interference that arrives from a different direction by applying a spatial filter at the receiver. However, forming beams or nulls in specified directions often requires a model of the array manifold and frequent calibration to correct time-varying errors [3]. Conversely, *blind* beamformers do not require knowing the directions of arrival or the array manifold. They are insensitive to calibration errors and can help boost transmission efficiency since they typically do not depend on overhead (e.g., a preamble) to train the beamformer. One class of blind techniques relies on developing an expert-crafted feature for a SOI and optimizing the beamformer to enhance that feature [4], [5]. These require detailed knowledge of the waveforms and have a limited ability to generalize to new SOIs. Another class is based on blind source separation (BSS), which is an unsupervised learning approach that does not require prior knowledge of the signals or how they interfere [6], [7], [8]. This is useful for developing *automated* beamformers for new SOIs without a human in the loop. However, they typically require thousands of stationary samples for good performance [1], [9], which hinders their application to short bursts of data or fast fading channels.

To overcome these limitations, we propose a new machine learning (ML)-driven process wherein the features of a SOI are automatically learned from a dataset of raw noisy in-phase and quadrature (IQ) samples and then exploited for blind beamforming. As shown in Fig. 1, we utilize inferences from a deep learning (DL)-based signal classifier to learn a beampattern that protects the SOI from interference. The classifier has been pre-trained to output a measure of confidence that the SOI is present, which can be accomplished using noisy captures from a single antenna. It is then deployed to process the output of a trainable linear beamforming layer and guide optimization of the beamforming weights to maximize

this confidence measure for the received array data. We refer to this technique as the classification-based transfer learning (CBTL) beamformer since it exploits the features previously learned by the classifier for the new task of beamforming. The proposed technique aims to automate the development of blind beamformers for new SOIs and improve the performance by utilizing supervised learning. We demonstrate the CBTL beamformer using a convolutional neural network (CNN) as the signal classification model in this paper. However, the concept easily extends to other ML models.

Several challenges needed to be addressed to realize the CBTL beamformer. First, the measure of classification confidence must be a proxy for the signal-to-interference-plus-noise ratio (SINR) of the SOI, so that maximizing the confidence also maximizes the SINR. We enabled this through a simple calibration of the pre-trained CNN. Second, learning the beamforming weights should converge quickly. We utilized a spatial whitening pre-processor and pruned the parameters of the CNN to facilitate this. Finally, the beamformer should be able to extract a SOI even if the interference is from the *same* class as the SOI. We instantiate multiple beamformers and force their outputs to be decorrelated to separate the signals in this case. The resulting technique is simple to integrate and can help automate the electronic protection of receivers that deal with many or novel SOIs, such as communications gateways and spectrum monitoring systems.

A. Related Works

The established techniques for blind beamforming are based on restoring a known property of a SOI, leveraging differences between signals, or exploiting the statistical independence of their sources. Known properties include cyclostationarity [10], constant modulus [11], or the modulation alphabet [12]. These typically depend on an expert-crafted feature that has a limited or no ability to automatically generalize to other SOIs. Techniques which exploit differences between signals, such as in their power levels [13], burst edges [4], or spectral shapes [4], [7], are degraded without these differences. Approaches based on BSS typically require thousands of stationary samples for good performance [1], [9] and assume mutually independent and non-Gaussian signals [6], [8], which hinders their application to Gaussian-like modulations such as orthogonal frequency division multiplexing (OFDM) [14].

Despite their limitations, the approaches based on BSS reduce the need for expert design and can help automate the processing for new SOIs without a human in the loop. We briefly discuss three of the well-known techniques. The first two use independent component analysis (ICA) but differ in their implementation. The joint approximation diagonalization of eigen-matrices (JADE) algorithm [6] exploits fourth-order cumulants based on independent vectors having maximal autocumulants and null cross-cumulants. JADE is deterministic and has no parameters to tune. Complex fast ICA (cFastICA) [8] is based on a popular algorithm for ICA and uses an iterative procedure to maximize a measure of non-Gaussianity. It randomly initializes the beamforming weights and proposes several measures to aid in their adaptation. The third technique is the second-order blind identification (SOBI) algorithm

[7], which leverages spectral differences and can separate Gaussian-like signals if they are not spectrally matched. It is based on joint diagonalization of a user-specified number of time-lagged spatial covariance matrices.

Utilizing additional information that may be available for the SOI can improve the separation performance relative to BSS [15]. This is known as *informed* source separation and has received recent attention for target speech extraction (TSE) using microphone arrays and DL-based classifiers [16]. Although DL-based classifiers have been used to optimize temporal filters to compensate for RF channels [17], their application to beamforming for RF signals has yet to be studied in the literature. This concept poses new challenges, since the techniques used for TSE typically rely on the signals being sparse and disjoint in the time-frequency domain. RF signals can have considerable overlap in this domain (e.g., in multi-user communications systems) and more intricate features must be learned to discriminate between them. Recent work has shown that such features can be learned automatically from raw IQ data using DL, and their performance can exceed classical techniques [18], [19], [20]. Our line of research leverages this approach to learn the features of signals and protect them from interference using a beamformer.

ML techniques for RF interference suppression and beamforming have also gained significant interest in recent years [21], [22]. However, the existing techniques for blind beamforming require training data from (or generated by a model of) an antenna array, and none appear to have been evaluated using real-world data [23], [24], [25]. Transfer learning has been studied and shown to improve the generalization of models for beamforming [26], [27], but these works used networks that were explicitly designed and pre-trained for beamforming. Alternatively, we propose optimizing the beamformer based on the learnable structure of its output, which does *not* require training data from an antenna array. This avoids the restrictions of prior art and eliminates dependencies on the array manifold. The proposed form of ML-driven beamforming for RF signals does not appear to have been studied in the literature.

B. Motivation and Contributions

Motivated by the limitations of existing blind beamformers and the need for more generic forms of interference rejection, we propose a new technique wherein features used to classify a SOI are learned *automatically* from a noisy single antenna dataset and then leveraged for beamforming. This avoids time-consuming expert design and tedious performance tuning while capitalizing on the versatility and success of DL-based RF signal classifiers. Instead of training a beamforming network from scratch, we apply transfer learning to train a simple linear beamforming layer using knowledge previously learned by the classifier. This saves time and cost, avoids the requirement for a comprehensive training dataset containing various mixtures of signals arriving at a particular antenna array, and is simple to integrate with existing receivers.

In our preliminary work [1], we introduced the concept behind the CBTL beamformer and used simulations to analyze its performance. However, the beamformer converged slowly,

could only form a single beam, and its evaluation was limited. In this paper, we build upon and extend our prior work by making the following contributions:

- **Novel ML-driven beamformer:** we propose the CBTL beamformer, a new technique which optimizes its weights using a pre-trained DL-based signal classifier and transfer learning. The classifier is pre-trained using raw IQ data to automatically learn the features of a SOI and reduce the need for expert design. To the best of our knowledge, this is the first ML-driven beamformer that does not require training data derived from an antenna array.
- **Classifier calibration:** we demonstrate the utility of temperature scaling [28] for increasing the sensitivity between a pre-trained classifier's confidence and the SINR. This simple technique enables the CBTL beamformer to be significantly more reliable and performant.
- **Improved learning algorithm:** options for spatial dimensionality reduction and Gram-Schmidt orthonormalization were incorporated to improve the performance and enable forming multiple beams to extract additional signals. We enhanced the convergence rate about tenfold by adapting the learning rate and refining the stopping criteria.
- **Extensive performance analysis:** the CBTL beamformer is thoroughly evaluated using simulations and compared to several state-of-the-art techniques. We study its performance under a variety of conditions, including in a multi-user communications scenario where all signals are from the same class. It typically achieves superior performance compared to state-of-the-art blind beamformers, especially when the sample support is small.
- **Real-world evaluation:** we validated the performance of the CBTL beamformer using over-the-air data collected using the open-access research testbed for next-generation wireless networks (ORBIT) [29]. It performed close to expectations despite real-world impairments that were not included in the classifier's training data. To the best of our knowledge, this was the first evaluation of a ML-driven blind beamformer using over-the-air data.

The remainder of this paper is organized as follows. Sec. II defines the received signal model and the blind beamforming problem. Sec. III presents the proposed CBTL beamformer and the details of our implementation. Its performance is analyzed using simulation experiments in Sec. IV and validated using over-the-air data in Sec. V. We conclude and discuss future work in Sec. VI. We plan to release our code and datasets to the public upon publication of this paper.

II. RECEIVED SIGNAL MODEL AND PROBLEM STATEMENT

A complex baseband-equivalent model for M signals received at an N -element antenna array at time t is

$$\mathbf{z}(t) = \sum_{i=0}^{M-1} \mathbf{a}_i y_i(t) + \mathbf{n}(t), \quad (1)$$

where $\mathbf{a}_i$ is an $N \times 1$ array response vector, $y_i(t)$ is the signal received from source i , and $\mathbf{n}(t) \sim \mathcal{CN}(\mathbf{0}, \sigma_n^2 \mathbf{I})$ are the receiver noises with powers σ_n^2 . The signals are assumed to use similar carrier frequencies which causes co-channel

interference. We assume that $1 \leq M \leq N$ with the exact value of M unknown. We designate $y_0(t)$ as the SOI and $y_{i \neq 0}(t)$ as interference. The array response vectors are modeled by $\mathbf{a}_i = \Psi \odot \mathbf{v}_i$, where $[\Psi]_n = e^{j\psi_n}$ and $\psi_n \sim \mathcal{U}[0, 2\pi]$ for $n = \{0, 1, \dots, N-1\}$ are the unknown receiver calibration errors and $\odot$ denotes the Hadamard product with the steering vector $\mathbf{v}_i$. We assume the signal sources are distant and the array elements are homogeneous, thus amplitude variations are not considered in $\mathbf{a}_i$.

The steering vectors depend on the relative positions of the antenna elements and sources. We consider a uniform linear array (ULA) with half-wavelength spacing, where

$$[\mathbf{v}_i(\theta_i)]_n = e^{j(n-(N-1)/2)\pi \cos(\theta_i)} \quad (2)$$

and θ_i is the signal's angle of arrival. The SOI arrives from an unknown angle θ_0 while the interferers arrive from angles

$$\theta_{i \neq 0} = \theta_0 + \eta_i 2 \sin^{-1}(2/N), \quad (3)$$

where η_i scales the null-to-null beamwidth of the ULA [30].

The signals $y_i(t)$ are assumed to be *narrowband*, i.e., having a bandwidth B_i Hz that satisfies $B_i \tau_{\max} \ll 1$ where $\tau_{\max}$ seconds is the maximum difference in arrival times between any two elements of the array. Therefore, the differences in arrival times for $y_i(t)$ can be approximated by the phase shifts in $\mathbf{v}_i$, and the channels are not frequency selective [30]. Each signal is received with an unknown time offset τ_i seconds, frequency offset f_i Hz, and phase ϕ_i rad,

$$y_i(t) = x_i(t + \tau_i) e^{j(2\pi f_i t + \phi_i)}, \quad (4)$$

where $x_i(t)$ is the signal sent by source i . Each element of the array is sampled at a rate of $F_s \geq B_0$ samples/s to produce an $N \times L$ matrix of IQ samples $\mathbf{Z}$ with columns $\mathbf{z}(l/F_s)$ for $l = \{0, 1, \dots, L-1\}$. We assume the channel is stationary for the L samples in $\mathbf{Z}$. The received signal-to-noise ratio (SNR), interference-to-noise ratio (INR), and SINR per element are

$$\text{SNR} = \sigma_0^2 / \sigma_n^2, \quad (5)$$

$$\text{INR} = \sum_{i=1}^{M-1} \sigma_i^2 / \sigma_n^2, \quad \text{and} \quad (6)$$

$$\text{SINR} = \text{SNR} / (1 + \text{INR}), \quad (7)$$

where σ_i^2 is the power of the $1 \times L$ sampled signal vector $\mathbf{y}_i$.

The general problem is to estimate an $N \times 1$ beamforming weight vector $\mathbf{w}$ to enhance the communications and sensing performance by minimizing the interference and noise power received with the SOI y_0 . Formally, we aim to find $\mathbf{w}$ to minimize the mean-square error

$$\mathbf{w}_{\text{OPT}} = \underset{\mathbf{w}}{\text{argmin}} E[(\hat{\mathbf{y}}_0 - \mathbf{y}_0)^2] \quad (8)$$

where $\hat{\mathbf{y}}_0 = \mathbf{w}^H \mathbf{Z}$ is the output of the beamformer. This is optimally accomplished by $\mathbf{w}_{\text{OPT}} \propto \mathbf{R}_n^{-1} \mathbf{a}_0$ [30], where

$$\mathbf{R}_n = \sum_{i=1}^{M-1} \sigma_i^2 \mathbf{a}_i \mathbf{a}_i^H + \sigma_n^2 \mathbf{I} \quad (9)$$

is the spatial covariance matrix of the interference-plus-noise and $\mathbf{a}_0$ is the array response of the SOI. $\mathbf{w}_{\text{OPT}}$ also maximizes

TABLE I
SUMMARY OF THE MAIN SYMBOLS USED IN SEC. III.

Symbol	Definition
N	Number of antenna elements
L	Number of IQ samples per antenna element
$\mathbf{Z}$	$N \times L$ matrix of IQ samples from the antenna array
$\mathbf{w}$	$N \times 1$ vector of beamforming weights
S	Assumed dimensionality of the signal subspace
$\mathbf{Q}$	$S \times N$ whitening and subspace projection matrix (18)
$\mathbf{Z}_Q$	$\mathbf{Z}$ after whitening and dimensionality reduction, $\mathbf{Z}_Q = \mathbf{Q}\mathbf{Z}$
$\mathbf{w}_Q$	$\mathbf{w}$ optimized for $\mathbf{Z}_Q$, $\mathbf{Q}^H \mathbf{w}_Q = \mathbf{w}$
$\hat{\mathbf{y}}_0$	Output of the beamforming layer, $\hat{\mathbf{y}}_0 = \mathbf{w}^H \mathbf{Z}$
p_{SOI}	Probability that $\hat{\mathbf{y}}_0$ belongs to the same class as the SOI
$\mathcal{L}$	Negative log-likelihood loss function
κ	Convolutional filter kernel size within the classifier
T	Temperature scaling of the classifier's outputs

the array signal-to-interference-plus-noise ratio (ASINR) [30], which is typically used to judge the quality of a received signal after beamforming. This is defined by

$$\text{ASINR} = \frac{\mathbf{w}^H \mathbf{R}_s \mathbf{w}}{\mathbf{w}^H \mathbf{R}_n \mathbf{w}}, \quad (10)$$

where $\mathbf{R}_s = \sigma_0^2 \mathbf{a}_0 \mathbf{a}_0^H$ is the spatial covariance matrix of the SOI. The goal in this paper is to use the confidence of a pre-trained DL-based classifier as a proxy for the ASINR and adapt $\mathbf{w}$ to maximize the confidence and hence approximate $\mathbf{w}_{\text{OPT}}$ without any prior knowledge about the waveform structure of $\mathbf{y}_0$. Sec. III describes this in detail and formulates the optimization problem solved in place of (8).

III. THE PROPOSED BEAMFORMER

As shown in Fig. 1, the proposed design prepends a linear beamforming layer with weights $\mathbf{w}$ to a pre-trained DL-based signal classifier and optimizes $\mathbf{w}$ based on knowledge previously learned by the classifier. The classifier, which has been trained to recognize properties of the SOI without knowledge of the antenna array or beamformer, processes the beamformed output $\hat{\mathbf{y}} = \mathbf{w}^H \mathbf{Z}$ with its parameters frozen and predicts the probability p_{SOI} that $\hat{\mathbf{y}}_0$ belongs to the same class as the SOI. Intuitively, p_{SOI} should be large if the beamformer puts gain on the SOI and nulls on interference. We build on this intuition by using the classifier as a feature extractor for the SOI and using transfer learning [31] to train the beamforming layer to maximize p_{SOI} . The following subsections describe the CBTL beamformer and classification model used in this paper. The main symbols used in this section are summarized in Table I.

A. Classification-Based Transfer Learning Beamformer

This section describes how the classifier is used to train the beamforming layer. First, we introduce the general concept and a simple method to increase the sensitivity of a pre-trained classifier to the ASINR. Then, we discuss our design choices, specify an algorithm for training the beamforming layer, and provide the details of our implementation.

1) *General Approach:* We model the classification function as $f(\mathbf{y}) = \mathbf{p}$, where $\mathbf{y}$ is a vector of L complex samples and $\mathbf{p}$ is a vector of likelihoods $p_i = p(\text{class} = i | \mathbf{y})$, i.e., the probabilities of $\mathbf{y}$ belonging to class $i = \{0, 1, \dots, C-1\}$

where C is the number of classes. The classifier's output for the beamformed signal $f(\mathbf{w}^H \mathbf{Z})$ depends on the beamforming weights, hence they can be trained to minimize a loss function $\mathcal{L} := \mathcal{L}(\mathbf{w} | \Theta, \mathbf{Z})$ where Θ are the frozen parameters of the classifier. Denoting class 0 as the SOI, $f_0(\mathbf{w}^H \mathbf{Z}) = p_{\text{SOI}}$ as its classification function, and $\mathcal{L} = -\log(p_{\text{SOI}})$ as the negative log-likelihood loss function, the optimization problem is

$$\mathbf{w}_{\text{CBTL}} = \underset{\mathbf{w}}{\text{argmin}} \mathcal{L}(\mathbf{w} | \Theta, \mathbf{Z}) \quad (11)$$

$$= \underset{\mathbf{w}}{\text{argmax}} f_0(\mathbf{w}^H \mathbf{Z}), \quad (12)$$

which is challenging to solve because $f_0(\cdot)$ is generally nonlinear and Θ typically contains thousands of parameters. However, we can use the chain rule for derivatives and backpropagation to compute the gradient of $\mathcal{L}$ with respect to the classifier's input, $\partial \mathcal{L} / \partial \mathbf{y}^*$. We can then compute the gradient of $\mathcal{L}$ with respect to $\mathbf{w}$,

$$\left(\frac{\partial \mathcal{L}}{\partial \mathbf{w}} \right)^T = \frac{\partial \mathcal{L}}{\partial \mathbf{y}^*} \times \left(\frac{\partial \mathbf{y}}{\partial \mathbf{w}^*} \right)^* \quad (13)$$

$$(\nabla_{\mathbf{w}} \mathcal{L})^T = \nabla_{\mathbf{y}^*} \mathcal{L} \times \mathbf{Z}^H, \quad (14)$$

and use a gradient descent method to update $\mathbf{w}^H$ to minimize $\mathcal{L}$ and solve (11). The process of beamforming, classification, backpropagation, and updating $\mathbf{w}^H$ repeats until convergence. We note the use of the *conjugate* derivatives in (13) [32].

2) *Calibration of the Classifier:* The proposed algorithm assumes that maximizing p_{SOI} will improve the ASINR. However, it may only reach the *minimum* ASINR required for confident classification, which may be insufficient for other forms of processing (e.g., demodulation). It would be a significant limitation if the classifier were to restrict the ASINR to below what could be theoretically achieved. Overcoming this requires p_{SOI} to be sensitive to, and ideally monotonically increasing with, the ASINR. We were unable to find evidence of this for the DL-based signal classifiers in the literature. However, [19] and [33] show that their accuracy tends to increase with the SINR and saturate once a threshold has been reached. We propose using temperature scaling [28] to mitigate any corresponding saturation in p_{SOI} and increase the classifier's sensitivity to the ASINR. The modified likelihoods are

$$p_i \leftarrow \frac{p_i^{1/T}}{\sum_{j=0}^{C-1} p_j^{1/T}}, \quad (15)$$

where the temperature $T > 0$ controls the model's confidence. Its uncertainty is increased when $T > 1$, which prevents p_{SOI} from saturating and can help the ASINR to continue to grow. We empirically determined a setting for T in Sec. IV-B by testing a pre-trained classifier with a range of SINRs and then choosing T to increase the sensitivity of p_{SOI} to the SINR.

3) *Optimizing the Weights:* We made three design choices to enhance the optimization of $\mathbf{w}$. First, $\mathbf{Z}$ is spatially whitened, a procedure often used to reduce the complexity [34] and increase the convergence rate by reducing the eigenvalue spread of the sample covariance matrix [35]

$$\hat{\mathbf{R}} = (\mathbf{Z} - \bar{\mathbf{Z}})(\mathbf{Z} - \bar{\mathbf{Z}})^H, \quad (16)$$

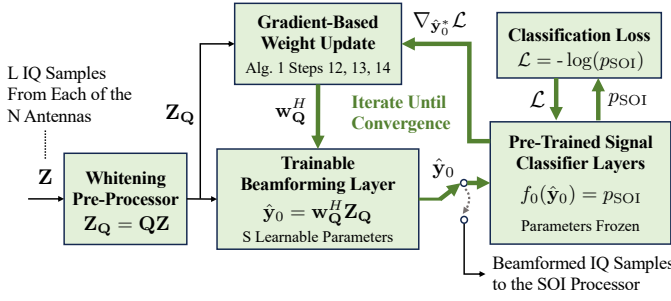


Fig. 2. Block diagram of the CBTL beamformer showing the whitening pre-processor and iterative optimization procedure.

where $\bar{Z}$ is the mean of Z along its rows. We perform principal component analysis (PCA)-based whitening with an option for dimensionality reduction [34], which uses the decomposition $\hat{R} = U\Lambda U^H$ where U is a unitary matrix of eigenvectors and Λ is a diagonal matrix of eigenvalues. The whitened data is

$$Z_Q = (\Lambda_S - I_S \bar{\lambda}_n)^{-1/2} U_S^H Z \quad (17)$$

$$= QZ, \quad (18)$$

where Λ_S is a diagonal matrix of the S largest eigenvalues in Λ , U_S contains their associated eigenvectors, I_S is a $S \times S$ identity matrix, and $\bar{\lambda}_n$ is the mean of the eigenvalues not in Λ_S . This projects Z onto the signal subspace if S is the same as the number of signals M , which reduces its spatial dimension and often reduces the noise. One may assume $S = N$ if M is unknown, in which case $\bar{\lambda}_n = 0$, $Q^H Q = \hat{R}^{-1}$, and the covariance matrix of Z_Q has equalized eigenvalues. In any event, the beamformer now operates in the whitened domain, $\hat{y}_0 = w_Q^H Z_Q$, as do its updates from (14). We note that when $S = N$, the equivalent updates to w are $\hat{R}^{-1} Z (\nabla_{\hat{y}_0} \mathcal{L})^H$. This resembles a typical form of the adaptive weights with $\nabla_{\hat{y}_0} \mathcal{L}$ in place of the unknown SOI y_0 and relates our method to the orthogonalized least mean square algorithm [36]. However, we learn w to minimize $E[(\nabla_{\hat{y}_0} \mathcal{L})^2]$ instead of $E[(\hat{y}_0 - y_0)^2]$, since this can be done without knowledge of y_0 .

Second, we optimize w_Q^H using complex stochastic gradient descent (SGD) with momentum [35] to accelerate the convergence. We note that there is no specific dependence on SGD and other complex gradient-based optimization algorithms may be used instead. Third, we perform Gram-Schmidt orthonormalization [37] after SGD,

$$w_Q \leftarrow w_Q - WW^H w_Q \quad (19)$$

$$w_Q \leftarrow \frac{w_Q}{\|w_Q\|_2}, \quad (20)$$

where W is a $S \times S$ matrix initialized to 0 and then populated with any previous solutions for w_Q on its columns (which are orthonormal as a result of this procedure). This enables the beamformer to extract multiple signals of the same class through sequential runs, since orthonormal sets of weights applied to the whitened data ensure uncorrelated outputs. The resulting algorithm is specified in Alg. 1 and its block diagram is shown in Fig. 2.

4) *Implementation Details:* The performance of Alg. 1 depends on the classifier and several parameters. We typically

Algorithm 1 CBTL Beamformer Weight Optimization

- 1: **Inputs:** array data Z , dimensionality $S \leq N$, learning rates $\mu = [\mu_0, \mu_1, \dots]$ for all iterations, momentum ν , prior weights W .
- 2: **Initialize:** choose weights $w_{Q,0}^H$ and orthonormalize using (19) and (20), $n \leftarrow 0$, $p_{\max} \leftarrow 0$, and $g \leftarrow 0$.
- 3: **Whiten:** $Z_Q \leftarrow QZ$ using (18).
- 4: **while** convergence criteria not satisfied (e.g., (21)) **do**
- 5: Beamform: $\hat{y}_0 \leftarrow w_{Q,n}^H Z_Q$.
- 6: Classify: $p_{SOI} \leftarrow f_0(\hat{y}_0)$.
- 7: Calibrate p_{SOI} using (15).
- 8: **if** $p_{SOI} \geq p_{\max}$ **then**
- 9: $w_{CBTL}^H \leftarrow w_{Q,n}^H$ and $p_{\max} \leftarrow p_{SOI}$, as per (12).
- 10: **end if**
- 11: Compute Loss: $\mathcal{L} \leftarrow -\log(p_{SOI})$.
- 12: Backpropagate: $\nabla_{w_Q} \mathcal{L} \leftarrow Z^* \times (\nabla_{\hat{y}_0} \mathcal{L})^T$.
- 13: Optimize: $w_{Q,n+1}^H \leftarrow w_{Q,n}^H - \mu_n (\nabla_{w_Q} \mathcal{L} + \nu g)$.
- 14: Orthonormalize $w_{Q,n+1}^H$ using (19) and (20).
- 15: $g \leftarrow \nabla_{w_Q} \mathcal{L} + \nu g$ and $n \leftarrow n + 1$.
- 16: **end while**
- 17: **return** w_{CBTL}^H and estimated SOI $\hat{y}_0 \leftarrow w_{CBTL}^H Z_Q$.

assumed $S = N$ and initialized the real and imaginary components of w_Q^H by sampling from $\mathcal{U}[-S^{-1/2}, S^{-1/2}]$ [38]. We found through the simulations in Sec. IV that an initial learning rate of $\mu_0 = 0.2$ and momentum of $\nu = 0.9$ worked well, and that $\mu_n = (1 - p_{SOI})^2$ for $n \geq 1$ accelerated the convergence. We ran a minimum of 25 iterations, maximum of 100 iterations, and declared convergence if

$$\|(|w_{Q,n}^H| - |w_{Q,n+1}^H|)\|_1 \leq 10^{-3}. \quad (21)$$

We implemented Alg. 1 using PyTorch, and now discuss the classifier used in our work.

B. Binary Signal Classification Model

Numerous DL-based RF signal classification models have been studied in the literature and architectures based on CNNs have been widely adopted. The subject of our work is in the application of these models to beamforming and not in the design of the classifier itself. We assume that the pre-trained classifier can discriminate between the SOI and interference to guide beamforming. Based on the scope of our work, we opted to use an existing model with proven performance and made efforts to reduce its complexity for use in Alg. 1.

Our model was a modified version of a baseline CNN that classified 11 types of modulations from raw IQ samples [39]. The baseline model took an input y of L samples, normalized it for unit power, split it into real and imaginary channels, and processed them using a network with six convolutional layers and one fully connected layer. First, we reduced the baseline model's scope from multi-class to one-vs-rest binary classification since it only needs to predict if y came from the same distribution as the SOI. Next, we pruned the baseline model by removing three of the hidden layers and modifying it for a single output. The resulting CNN is shown in Fig. 3 and was implemented using PyTorch. Our modifications reduced

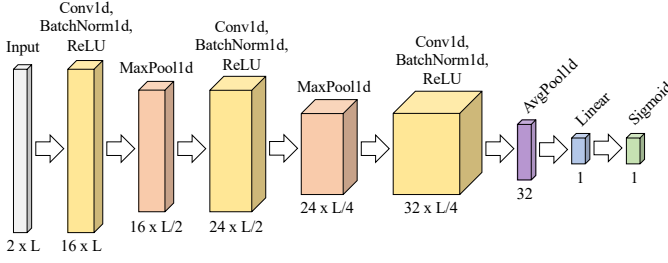


Fig. 3. Structure of the binary classification CNN used in this work. The layer names correspond to the PyTorch library, and their output sizes (channels $\times$ width) are shown below them.

the number of parameters from $11936\kappa + 1034$ to $1184\kappa + 249$, where κ is the size of the convolutional filter kernels, which reduced the number of classifier-related operations in Alg. 1 from about $1984\kappa L$ to $832\kappa L$. Finally, the baseline model used $\kappa = 8$, and we reduced this to $\kappa = 2$ unless noted otherwise.

The classifier was trained using a dataset $\mathbb{D} = [\mathbf{D}_{\text{SOI}}; \mathbf{D}_{\text{NOT}}]$, where $\mathbf{D}_{\text{SOI}}$ and $\mathbf{D}_{\text{NOT}}$ were $K \times L$ matrices containing K training waveforms for the SOI and the interference, and the corresponding target classification outputs $\mathbb{L} = [1, 0]^T$, where the first K entries were 1 (i.e., contained the SOI) and the last K entries were 0 (i.e., did not contain the SOI). The size and contents of $\mathbb{D}$ are discussed in Sec. IV. The parameters of the CNN were optimized following [39]: using mini-batch SGD with mini-batches of 256 randomly selected waveforms, an initial learning rate of $\mu_0 = 0.02$, momentum $\nu = 0.9$, and decaying μ_0 by a factor of 0.1 every 9 epochs. Each epoch used 80% of $\mathbf{D}_{\text{SOI}}$ and $\mathbf{D}_{\text{NOT}}$ for optimization and the remaining 20% for validation. The objective was to minimize the binary cross-entropy loss function

$$\mathcal{L}_v = -\frac{1}{K_v} \sum_{i=0}^{K_v-1} l_i \log(p_{\text{SOI},i}) + (1-l_i) \log(1-p_{\text{SOI},i}), \quad (22)$$

where K_v was the number of validation waveforms and l_i were their corresponding targets from $\mathbb{L}$. We used 5-fold cross-validation and trained the model for 30 epochs each time. The model that minimized $\mathcal{L}_v$ was saved as a pre-trained classifier.

IV. SIMULATED PERFORMANCE ANALYSIS

In this section, we use synthetic datasets and Monte Carlo simulation experiments to analyze the performance of the CBTL beamformer. We first describe the baseline scenario, datasets, and pre-training of the classifier. Next, we define the benchmarks and analyze the beamforming performance as functions of the sample support, SNR, and number of antennas. Finally, we demonstrate the versatility of the proposed technique by applying it to two additional scenarios that can challenge other blind beamformers: (i) multi-user communications with spectrally matched interference, and (ii) separation of an OFDM SOI from full-band interference.

A. Baseline Scenario and Datasets

The baseline scenario involves a SOI $y_0(t)$ and interference $y_1(t)$ arriving at an antenna array from unknown but different directions. A 16-quadrature amplitude modulation (QAM)

TABLE II
MAIN PARAMETERS USED IN THE BASELINE SIMULATION EXPERIMENTS.

Component	Parameter	Value
SOI	Carrier Frequency f_c	915 MHz
	Modulation	RRC-filtered 16-QAM
	Symbol Rate	100 Ksymbols/s
	Angle of Arrival θ_0	$\mathcal{U}[0^\circ, 180^\circ]$
	SNR	Varies
Channel	Frequency Error f_0	$\mathcal{U}[-4575, 4575]$ Hz
	Phase Error ϕ_0	$\mathcal{U}[-180^\circ, 180^\circ]$
	Time Delay τ_0	$\mathcal{U}[0, 1]$ symbol
	Sample Support L	Varies
	Modulation	Full-band AWGN
Interference	Angle of Arrival θ_1	$\theta_0 \pm 30^\circ$
	INR	40 dB
	Antenna Array	ULA with various N
Receiver	Sample Rate f_s	200 Ksamples/s

signal is used as the SOI since it is common in practice and modulates both the carrier amplitude and phase. The interference is specified to uniformly cover the frequency band, which is often the most damaging type of interference, and is modeled by additive white Gaussian noise (AWGN),

$$y_1(t) \sim \mathcal{CN}(0, \sigma_i^2). \quad (23)$$

We now describe the noisy single antenna training dataset $\mathbf{D}$ and testing dataset $\mathbf{D}'$ for the SOI. The main parameters of the baseline simulation experiments are shown in Table II.

Training Dataset: We generated $\mathbf{D}$ using randomized 16-QAM symbols and a root-raised cosine (RRC) filter with a roll-off of 1 and span of 12 symbols. The carrier frequency was $f_c = 915$ MHz, the symbol rate was 100 Ksymbols/s, and the sample rate was $f_s = 200$ Ksamples/s (i.e., the Nyquist rate). The received signal was modeled based on (4) with a clock offset of $\epsilon \sim \mathcal{U}[-\epsilon_{\max}, \epsilon_{\max}]$ parts-per-million, phase rotation of $\phi_0 \sim \mathcal{U}[-180^\circ, 180^\circ]$, and time delay of $\tau_0 \sim \mathcal{U}[0, 2/f_s]$ seconds. We used $\epsilon_{\max} = 5$ for errors up to $f_0 = \pm 4575$ Hz in f_c and ± 1 sample/s in f_s . The size of $\mathbf{D}$ was limited 2×10^6 consecutive IQ samples, it did not contain the transient response of the RRC filter, and AWGN was applied such that its SNR was 20 dB.

Testing Dataset: A testing dataset $\mathbf{D}'$ was generated using independent realizations of the SOI not contained in the training data. We generated 10^4 realizations of 16384 IQ samples, where each realization included randomized and filtered 16-QAM symbols with random clock offsets, phase rotations, and time delays following the models and distributions specified for $\mathbf{D}$. Noise and interference were added to the the testing dataset to calibrate the classifier in Sec. IV-B and to generate the received array data in Sec. IV-D through IV-H.

B. Classifier Pre-Training and Calibration

Since $\mathbf{D}$ was relatively small and only represented a single transmitter, we augmented it based on (4) to generate a larger dataset $\mathbf{D}_{\text{SOI}}$ for training the classifier. Each row in $\mathbf{D}_{\text{SOI}}$ contained a waveform that was generated by randomly selecting L consecutive IQ samples from $\mathbf{D}$ and then applying a random clock offset, phase rotation, and time delay distributed as in Sec. IV-A. We generated $K = 10^5$ waveforms to help the classifier learn to be robust to synchronization errors. We

generated $\mathbf{D}_{\text{NOT}}$ using (23) and matched its size to $\mathbf{D}_{\text{SOI}}$. Each waveform in $\mathbf{D}_{\text{SOI}}$ and $\mathbf{D}_{\text{NOT}}$ was normalized for unit power.

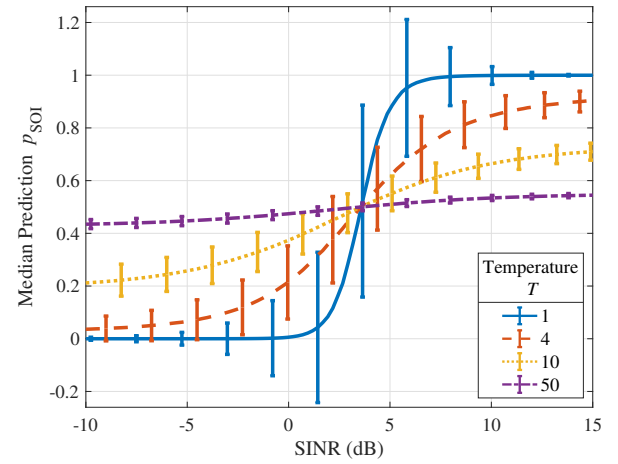
Classifiers with kernel sizes of $\kappa = 2$ were trained for $L = 2^i$ Nyquist samples with $i = \{3, 4, \dots, 14\}$ using the procedure described in Sec. III-B. We used a NVIDIA V100 graphics processing unit (GPU) and the total training time ranged from 8 minutes when $L = 8$ samples to 221 minutes when $L = 16384$ samples. The validation loss converged in all cases and ranged from -3.2 when $L = 8$ samples (validation accuracy of 98.5%) to -9.7 when $L = 16384$ samples (validation accuracy of 100%). Temperature scaling was then applied using (15) to calibrate the trained classifiers. We determined a setting for T by testing the classifiers using $\mathbf{D}'$ and analyzing their predictions p_{SOI} as T was varied. The test inputs to the classifiers consisted of L samples from each waveform in $\mathbf{D}'$ with AWGN applied for a SNR of 20 dB and interference applied for SINRs between -10 dB and 15 dB.

Fig. 4 shows the results of 10^4 tests when $L = 64$ Nyquist samples. The median of p_{SOI} is reported in Fig. 4a since the distributions tended to be skewed. We observed that p_{SOI} typically saturated around a SINR of 10 dB without temperature scaling (i.e., $T = 1$ in (15)). Increasing T helped to prevent this and made the standard deviations of p_{SOI} more uniform across SINR. This is further illustrated in Fig. 4b, which shows the empirical cumulative distribution functions (CDFs) of p_{SOI} at different SINRs. T should ideally be large enough for a high probability of improving the SINR when p_{SOI} is maximized but small enough for p_{SOI} to remain as sensitive as possible to the SINR so that Alg. 1 can converge quickly. We found that $T = 4$ was a good choice for this classifier. Fig. 5 shows that $T = 4$ also worked well for other values of L , and that increasing L reduced the standard deviations of p_{SOI} .

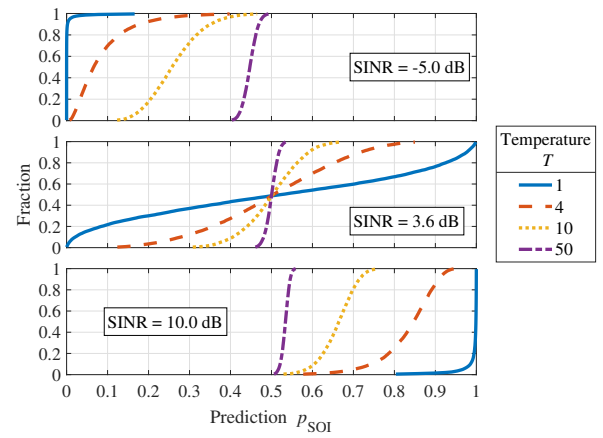
C. Beamforming Performance Metric and Benchmarks

We assessed the performance of a set of beamforming weights $\mathbf{w}$ based on the ASINR as defined in (10). As discussed in Sec. II, the ASINR is maximized by $\mathbf{w}_{\text{OPT}} \propto \mathbf{R}_{\mathbf{n}}^{-1} \mathbf{a}_0$. We use $\mathbf{w}_{\text{OPT}} := (\mathbf{R}_{\mathbf{n}}^{-1} \mathbf{a}_0) / (\mathbf{a}_0^H \mathbf{R}_{\mathbf{n}}^{-1} \mathbf{a}_0)$, which we refer to as the optimal beamformer and its performance serves as the theoretical upper bound. When $\mathbf{w}$ must be estimated using the given array data $\mathbf{Z}$, the ASINR can be maximized by $\mathbf{w}_{\text{LSE}} = (\mathbf{Z}\mathbf{Z}^H)^{-1} \mathbf{Z}\mathbf{y}_0^H$ [30]. We refer to this as the least squares estimator (LSE) beamformer, and it requires knowing the desired output of the beamformer $\mathbf{y}_0$. This is typically a training signal that has been embedded within the SOI and the receiver must synchronize to this signal which presents a challenge in practice. We do not address this challenge and assume perfect synchronization for the LSE beamformer.

Several expert-designed and BSS-based beamformers discussed in Sec. I-A were considered as more practical benchmarks. We refer to these and CBTL as *uninformed* beamformers since they do not rely on an embedded training signal. We included the least squares constant modulus algorithm (LS-CMA) [40], which operated in the whitened domain, had its cost function customized for the average amplitude of 16-QAM [41] learned from the training dataset, and ran up to 100



(a) Median and standard deviation of p_{SOI} versus SINR.



(b) Empirical CDFs of p_{SOI} at different SINRs.

Fig. 4. The effects of temperature scaling on the classifier's predictions.

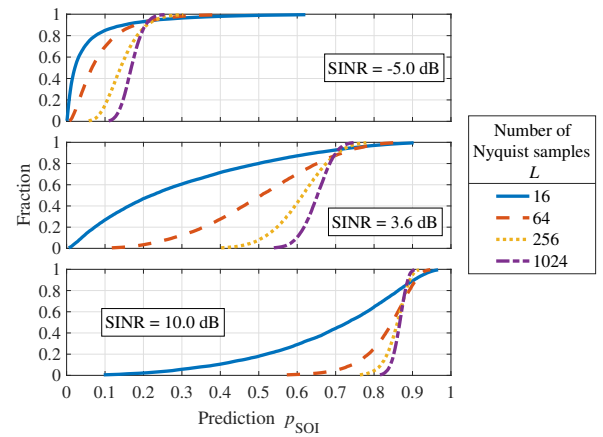


Fig. 5. Empirical CDFs of p_{SOI} with different sample supports and $T = 4$.

iterations or terminated according to (21). We also included the phase-self-coherence restoral (P-SCORE) algorithm [10], which was hand-crafted to exploit the self-coherence of 16-QAM at a multiple of its symbol rate. The BSS-based beamformers included implementations of JADE [42], cFastICA [43], and SOBI [44] with their default parameters.

The LS-CMA, BSS-based, and CBTL beamformers learned weights in the whitened domain, hence the transformation

$\mathbf{w} = \mathbf{Q}^H \mathbf{w}_Q$ with $\mathbf{Q}$ from (17) was applied prior to computing the ASINR. The LSE, SOBI, and CBTL beamformers processed samples directly from the datasets at the Nyquist rate, while the others interpolated the data by a factor of two prior to processing. This was necessary for our configuration of the P-SCORE algorithm and was empirically found to improve the performance of the LS-CMA, JADE, and cFastICA beamformers for 16-QAM.

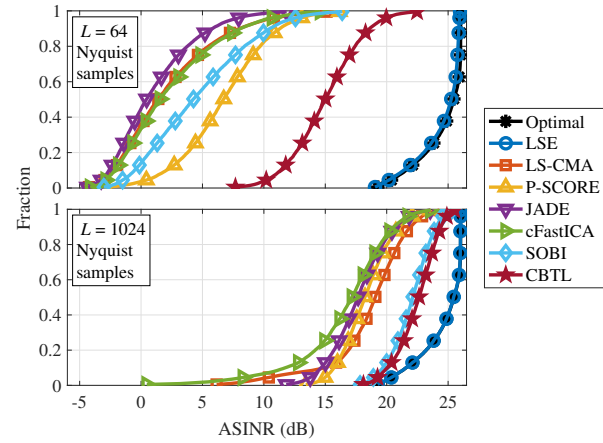
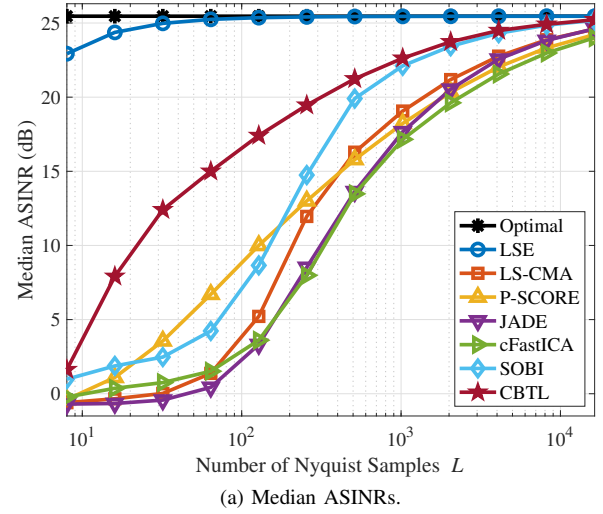
We assumed there was no prior knowledge for $\mathbf{w}$ and estimated it for each realization of $\mathbf{Z}$. Unless stated otherwise, the number of signals was assumed to be unknown. Therefore, we set $S = N$ in (17), used each beamformer to estimate N weight vectors, and selected the one that maximized the ASINR. In practice, the BSS-based beamformers require additional processing to detect the SOI among their N outputs, while the others are designed to restore the desired signal properties (i.e., constant modulus, self-coherence frequency, or signal classification).

D. Performance with Different Sample Supports

The ability to estimate a beamformer using a small amount of data is important for applications involving short data bursts or dynamic RF environments. We now study the performance of the CBTL beamformer as the number of Nyquist samples available for processing are varied. We modeled a $N = 4$ element ULA with the SOI arriving from a random angle $\theta_0 \sim \mathcal{U}[0^\circ, 180^\circ]$ and the interference arriving with a null-to-null beamwidth separation of $\eta_1 = \pm 0.5$ in (3), i.e., $\theta_1 = \theta_0 \pm 30^\circ$ with the sign chosen at random but constrained such that $\theta_1 \in [0^\circ, 180^\circ]$. We generated 10^4 realizations of $\mathbf{Z}$ for testing using the SOI waveforms from $\mathbf{D}'$ and interference from (23) in our model (1). The per-element SNR was 20 dB, and the per-element INR was 40 dB. Each realization contained independent receiver calibration errors and steering vectors. The CBTL beamformer used the pre-trained classifiers from Sec. IV-B with $T = 4$.

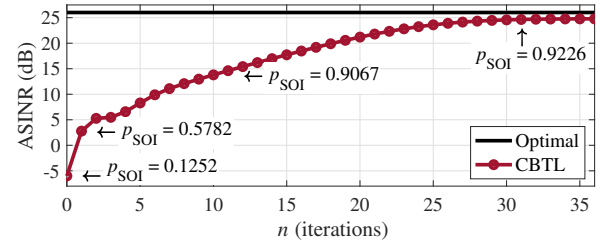
The results of estimating the beamformers as the number of Nyquist samples L was varied are shown in Fig. 6. We found that the CBTL beamformer typically achieved superior ASINRs compared to the other uninformed beamformers. The median ASINRs are reported in Fig. 6a since the distributions tended to be skewed. The typical gains when $L = 64$ samples ranged from 14.6 dB to 8.3 dB. The performance of all non-optimal techniques increased with L , and the LSE beamformer converged to optimal when $L = 64$ samples. The SOBI and CBTL beamformers converged to optimal when $L = 16384$ samples and the others were 0.6 dB to 1.2 dB worse.

We found that the convergence rate of the CBTL beamformer was 100% for $L \geq 512$ samples and was reduced for smaller sample supports: examples include 59%, 83%, and 95% for $L = 8, 16$, and 128 samples. Examples of the convergence and learned beampattern when $L = 1024$ samples are shown in Fig. 7. Fig. 7a shows that p_{SOI} and the ASINR both increased rapidly during the first few iterations and the ASINR improved substantially even as p_{SOI} became close to its maximum. Fig. 7b shows the Array Gain $= |\mathbf{w}^H \mathbf{v}(\theta)|^2$, where $\theta \in [0^\circ, 180^\circ]$, $\|\mathbf{w}\|_2 = 1$, and $\Psi = \mathbf{1}$ (i.e., no receiver calibration errors). Interestingly, the CBTL beampattern appeared

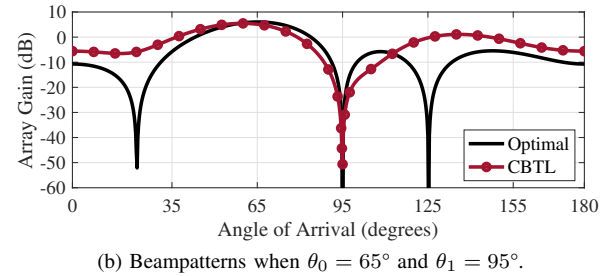


(b) Empirical CDFs of the ASINRs for two sample supports.

Fig. 6. Beamforming performance with different sample supports.



(a) ASINR and p_{SOI} (with $T = 4$) for the iterations of Alg. 1.



(b) Beampatterns when $\theta_0 = 65^\circ$ and $\theta_1 = 95^\circ$.

Fig. 7. Examples of the (a) convergence behavior and (b) learned beampattern.

to primarily utilize only two of the four available degrees of freedom by forming a single null.

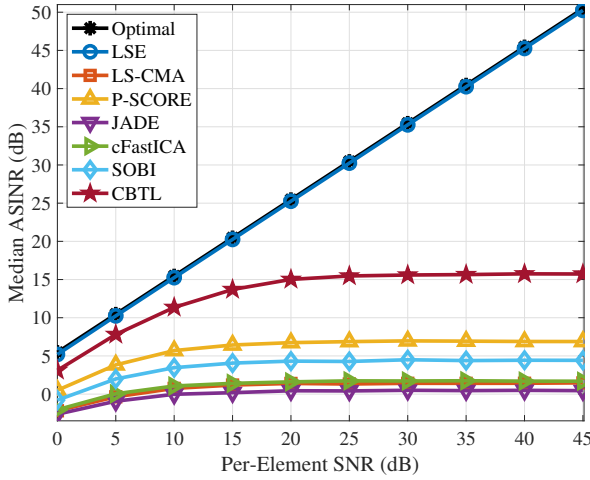


Fig. 8. Median ASINRs with different per-element SNRs.

E. Performance with Different Signal-to-Noise Ratios

Adaptive beamforming is typically challenged by low SNRs. To investigate this, we repeated the experiments from Sec. IV-D and varied the per-element SNR while L was fixed at 64 samples. As shown in Fig. 8, the CBTL beamformer typically outperformed the other uninformed techniques regardless of the SNR. The results also show that all uninformed beamformers were limited by the sample support, since they did not improve once a certain SNR was reached. We observed reduced convergence rates for the CBTL beamformer when the SNR was low, with 31% at 0 dB, 80% at 5 dB, and greater than 90% for SNRs ≥ 10 dB. The convergence rate improved as L increased, for example, to 91% when $L = 256$ samples and the SNR was 5 dB.

F. Performance with Different Numbers of Array Elements

Increasing the number of antenna elements N can enable larger array gains and the ability to null additional sources of interference. However, the beamforming performance can degrade if the sample support does not grow with N . We investigated this degradation by repeating the experiments from Sec. IV-D with L fixed at 256 samples and $N = 2$ to 10 elements in the ULA. The results are shown in Fig. 9. We found that the performance of the CBTL beamformer degraded at a slower rate than the other uninformed beamformers. Its ASINR was typically reduced by 5 dB over this range of N , whereas the ASINRs for the P-SCORE beamformer declined by about 10 dB and the rest declined by nearly 20 dB. This experiment also revealed that the number of iterations required by the CBTL beamformer increased about linearly with N , with 36, 64, and 84 being the mean numbers of iterations when $N = 2, 5$, and 8, respectively. The corresponding convergence rates were 100%, 97%, and 76%.

Computational complexity is of interest for systems that require a low processing latency. We repeated the previous experiment while measuring the runtimes of the uninformed beamformers as N and L were varied to help understand their computational requirements. This was carried out on an Apple MacBook Pro with the CBTL beamformer implemented

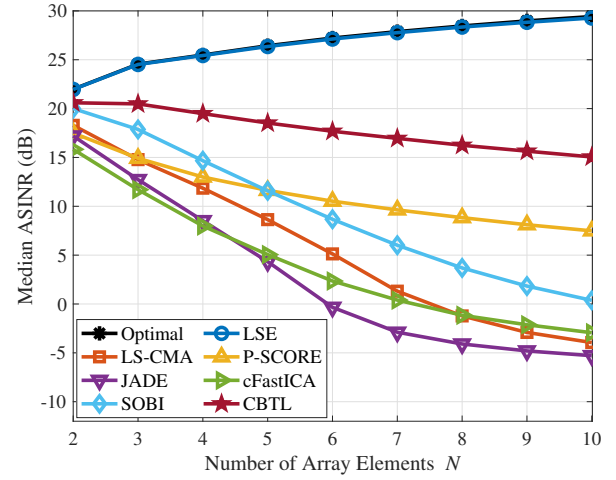


Fig. 9. Median ASINRs for different numbers of antenna elements.

using PyTorch and the others implemented using MATLAB. We ran the CBTL beamformer on the central processing unit (CPU) instead of the GPU for a fair comparison to the other techniques. The experiments did not reduce the dimension of $\mathbf{Z}$ and measured the time each method took to produce beamforming weights for the SOI. The results are shown in Table III. The CBTL beamformer consistently achieved the highest ASINRs but was the slowest until N and L became large. This was due to the large number of operations required for the CNN, which may be alleviated by compressing the model or leveraging the parallel nature of its operations.

PCA may be used in practice to estimate the number of signals M and project $\mathbf{Z}$ onto a lower dimensional space before beamforming. This can improve the performance, as discussed in Sec. III-A3. However, it can null the SOI if it is not included in the reduced subspace (e.g., if the eigenvalue associated with the SOI is not among the S largest eigenvalues in (17)). We investigated the performance improvement by repeating the experiments with $N = 8$ elements, $L = 256$ samples, and varying the assumed size of the signal subspace S in (17) when reducing the dimensionality of $\mathbf{Z}$. We used an existing option for this in JADE and implemented similar options in the other uninformed beamformers. Fig. 10 shows that the performance of all techniques improved as S became closer to M , but the CBTL beamformer was still superior and remarkably less dependent on detecting the number of signals arriving at the array.

G. Multi-User Communications Scenario

In multi-user communication systems, spatially diverse transmitters may use the same frequency channel simultaneously to increase the system capacity. Beamforming is often utilized by multi-user receivers to separate co-channel SOIs, which in principle can be done by exploiting the diversity among their array response vectors. Running the CBTL beamformer with the constraint that its weights be orthonormal to any previously estimated weights, as in (19) and (20), ensures extracting uncorrelated signals. However, the classifier must be sensitive to the interference between SOIs to prevent

TABLE III
MEDIAN RUNTIMES (MILLISECONDS) AND ASINRS (DECIBELS) ON AN APPLE M3 MAX CPU.

N	L	LS-CMA		P-SCORE		JADE		cFastICA		SOBI		CBTL	
		Time	ASINR	Time	ASINR	Time	ASINR	Time	ASINR	Time	ASINR	Time	ASINR
4	64	0.7	-5.5	0.1	6.7	1.6	1.2	1.1	0.3	0.7	4.2	25.5	15.1
	256	1.0	11.2	0.1	13.2	5.7	8.2	1.3	8.5	1.4	14.7	24.8	19.6
	1024	0.9	19.0	0.1	18.2	19.7	17.1	2.3	17.8	2.7	22.1	25.4	22.7
8	64	0.6	-10.2	0.1	0.4	5.0	-2.7	6.5	-4.1	5.2	-1.5	39.0	8.6
	256	1.3	-10.2	0.1	8.9	25.8	-1.2	10.6	-4.0	16.4	3.8	37.9	16.3
	1024	2.7	15.2	0.2	15.0	78.5	11.5	22.9	9.5	21.4	19.8	36.2	21.4
16	64	0.7	-13.9	0.2	-12.6	16.3	-6.1	86.5	-6.8	45.5	-4.7	39.8	-4.0
	256	1.4	-13.1	0.3	4.4	79.7	-5.8	157.4	-7.1	218.3	-3.9	42.7	12.0
	1024	10.1	-13.7	0.9	11.6	261.3	-4.0	350.0	-7.1	1044.2	13.8	53.5	19.1

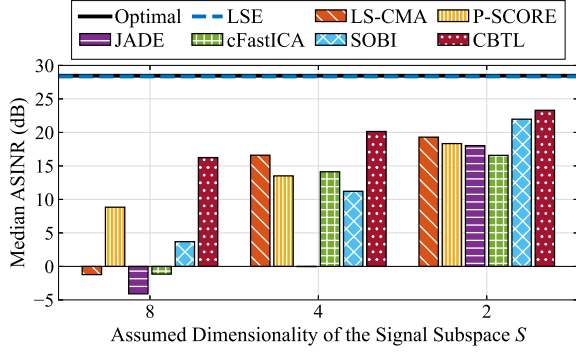


Fig. 10. The effect of dimensionality reduction for 8 antennas and 2 signals.

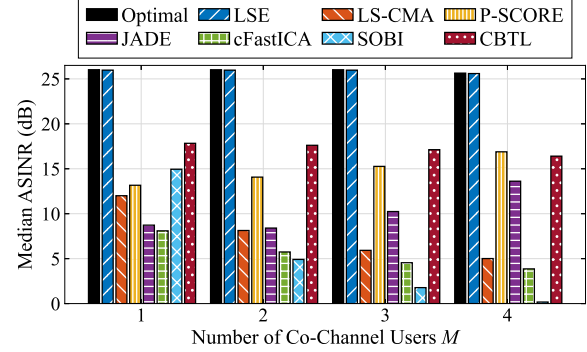


Fig. 11. Median ASINRs for the multi-user communications scenario.

learning beampatterns that have gain towards multiple SOIs simultaneously. In this section, we demonstrate that a simple modification to the pre-training of the classifier can enable good beamforming performance for multi-user systems.

The scenario from Sec. IV-A was modified such that the interference was due to 16-QAM signals from other simulated users. To accommodate this new scenario, we retrained the classifier following Sec. IV-B except with $\mathbf{D}_{\text{NOT}} = \mathbf{D}_{\text{SOI}} + \text{circshift}(\mathbf{D}_{\text{SOI}})$, where $\text{circshift}(\cdot)$ is a circular shift of the rows, to provide the classifier examples of the interference between independent SOIs. Then, we generated 10^4 realizations of $\mathbf{Z}$ for testing with an $N = 4$ element ULA and up to $M = 4$ independent signals from $\mathbf{D}'$ in (1). The signals each had a per-element SNR of 20 dB and arrived from angles based on (3) with $\theta_0 = 65^\circ$ and $\eta_i = 0.5i$ for $i = \{1, 2, 3\}$. Each realization of $\mathbf{Z}$ contained $L = 256$ Nyquist samples and independent receiver calibration errors. The CBTL beamformer used the retrained classifier with the same temperature scaling (i.e., $T = 4$). The orthonormalization constraint from (19) and (20) was used by the LS-CMA beamformer to form multiple beams.

The results for the multi-user scenario are shown in Fig. 11. We found that the CBTL beamformer successfully extracted each of the SOIs and typically outperformed the other uninformed beamformers. It had a typical performance loss of 1.4 dB between one SOI with no interference and three interferers present. The performance of SOBI significantly decreased as the number of co-channel signals increased; this was expected because they did not have appreciable spectral differences. Interestingly, the performance of JADE and P-SCORE increased with the number of signals, and P-SCORE

outperformed the CBTL beamformer by 0.5 dB when $M = 4$.

H. OFDM Scenario

This section investigates another modification to the scenario from Sec. IV-A by changing the modulation of the SOI to OFDM. OFDM is widely used by modern wireless systems and can challenge blind beamformers since it is not constant modulus, uses a long symbol duration, and has a Gaussian-like distribution if the number of subcarriers is large [14].

First, the classifiers were retrained for the new SOI. The OFDM signal used symbols with 1024 subcarriers, all modulated using quadrature phase shift keying symbols, and a cyclic prefix of 72 samples. The training and testing datasets ($\mathbf{D}$ and $\mathbf{D}'$) were generated based on the procedures from Sec. IV-A. However, the OFDM symbol rate was 100 symbols/s, and the sample rate was $f_s = 137$ Ksamples/s (i.e., 25% greater than the Nyquist rate). $\mathbf{D}$ was generated with 1.37×10^6 consecutive IQ samples and the waveform began at a random time offset into the first OFDM symbol, $\tau_0 \sim \mathcal{U}[0, 1370/f_s]$ seconds. We augmented $\mathbf{D}$ to generate $\mathbf{D}_{\text{SOI}}$ with $K = 10^4$ waveforms and trained the classifiers as in Sec. IV-B except with a kernel size of $\kappa = 8$ to minimize the validation loss (22). A temperature scaling of $T = 4$ once again introduced the desired sensitivity between p_{SOI} and the SINR.

The experiments from Sec. IV-D were then repeated using realizations of the OFDM SOI from $\mathbf{D}'$ with a per-element SNR of 10 dB and AWGN interference with a per-element INR of 30 dB. The LS-CMA beamformer was configured based on the average signal amplitude from the training data and the P-SCORE beamformer was designed to detect the

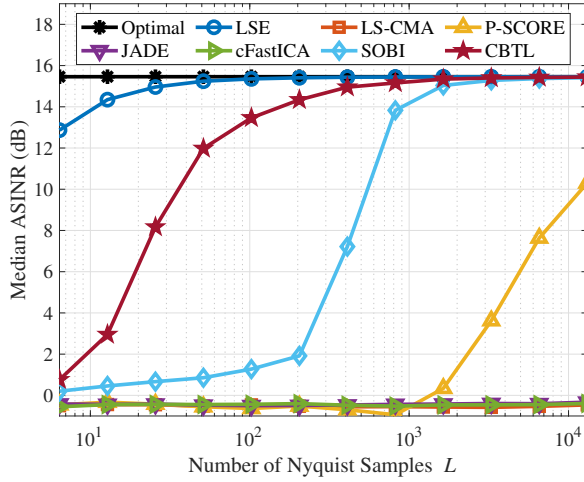


Fig. 12. Median ASINRs for different sample supports with the OFDM SOI.

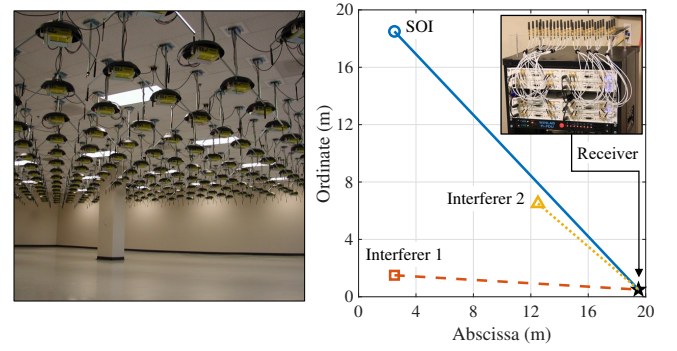
cyclic prefix and self-coherence frequency at a multiple of the OFDM symbol rate. As before, the LSE, SOBI, and CBTL beamformers processed samples directly from the datasets while the other techniques interpolated the data by a factor of two prior to processing. The results are shown in Fig. 12. As expected, the LS-CMA, JADE, and cFastICA beamformers were unable to extract the SOI. The SOBI and P-SCORE beamformers did not have the same limitation; however, due to the large spectral overlap between the SOI and interference and the need to observe multiple OFDM symbols, they required about 16 and 370 times more Nyquist samples than the CBTL beamformer for the typical ASINR to be 10 dB. These results demonstrate that the CBTL beamformer is versatile and can achieve good performance with signals that challenge other techniques.

V. RADIO TESTBED-BASED PERFORMANCE VALIDATION

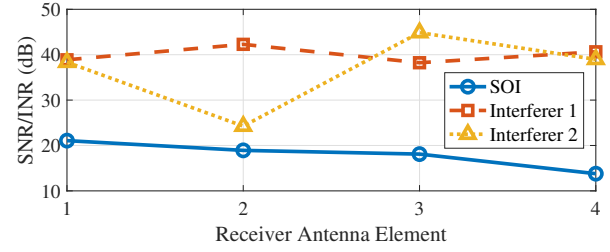
Although the simulated results from the previous section were encouraging, they were based on processing datasets that had been generated using a simple model of the received data. Real-world RF signals experience impairments due to hardware imperfections, multipath interference, and other phenomena. It is intractable to train the classifier for each of these and unrealistic to assume that the training and test data will follow the same distribution. In this section, we validate the performance from Sec. IV-D by processing over-the-air signals collected using the ORBIT testbed [29] *without retraining* the classifiers used by the CBTL beamformer. This represents the practical situation where the classifier's training data is collected in one environment and the beamformer is deployed to another. We now discuss the experimental configuration, datasets collected, processing methodology, and results.

A. Configuration

We used the ORBIT radio grid due to its maturity and accessibility. This testbed is located indoors and contains 400 ceiling-suspended nodes arranged in a 20 m $\times$ 20 m grid. Universal software radio peripheral (USRP) B210-equipped nodes were used as transmitters and one of the USRP X310-based



(a) The ORBIT radio grid (left) and layout of the nodes used in our experiments (right), with images from [45].



(b) Measured per-element SNR and INRs at the receiver.

Fig. 13. The configuration of our experiments on the ORBIT radio grid.

mini-racks [45] was used as a receive array. The layout of the nodes is shown in Fig. 13a. Using the nomenclature from [45], we used node3-19 to transmit the SOI, node3-2 (location 1) or node13-7 (location 2) to transmit interference, and the mini-rack underneath node20-1 to collect raw IQ samples for post-processing. The nominal carrier frequency was $f_c = 5.4$ GHz (wavelength $\lambda_c = 5.55$ cm). Signals were captured using $N = 4$ antennas on top of the mini-rack, specifically the RX2 ports of the UBX daughterboards in devices 24-1 and 24-2, forming a ULA with 10.16 cm ($d = 1.83\lambda_c$) of separation between the antennas. The null-to-null beamwidth [30] of this array was $2 \sin^{-1}(\lambda_c/(Nd)) = 15.7^\circ$ and the interference was separated from the SOI by about $\eta_1 = 2.75$ and $\eta_2 = 0.23$ beamwidths from locations 1 and 2, respectively.

The power levels of the transmitters were calibrated for a SNR of 20 dB and INRs of 40 dB at the first antenna of the receive array. As shown in Fig. 13b, the received power levels varied significantly across the array. This was determined to be due to multipath propagation. We found that the channel was approximately flat over a bandwidth of 625 kHz, thus used this as our signal bandwidth for the narrowband approximation to hold. We note that although multipath was not considered in (1), it typically does not affect the narrowband blind beamforming model [6].

The receive array was not calibrated, but its receivers were synchronized to common pulse-per-second and 10 MHz references provided over equal length cables. This enabled phase coherent down-conversion and synchronous sampling. There was no synchronization between the transmitters and receiver, however, their host computers had their times aligned to within 10s of milliseconds by the network time protocol for the purpose of triggering experiments. The mean carrier frequency offsets between the transmitters and receiver were

measured to be -9.7 kHz, -7.9 kHz, and -7.3 kHz for the SOI, interferer location 1, and interferer location 2.

B. Data Collection

The objective was to collect over-the-air datasets to validate the simulated results shown Sec. IV-D. Due to the lack of synchronization between nodes, we devised a test protocol that consisted of three consecutive segments for the transmitters:

- S1: One second off.
- S2: One second sending tones offset from the nominal f_c , $+75$ kHz for the SOI and -75 kHz for the interference.
- S3: Eight seconds sending known modulated signals for the beamforming experiments.

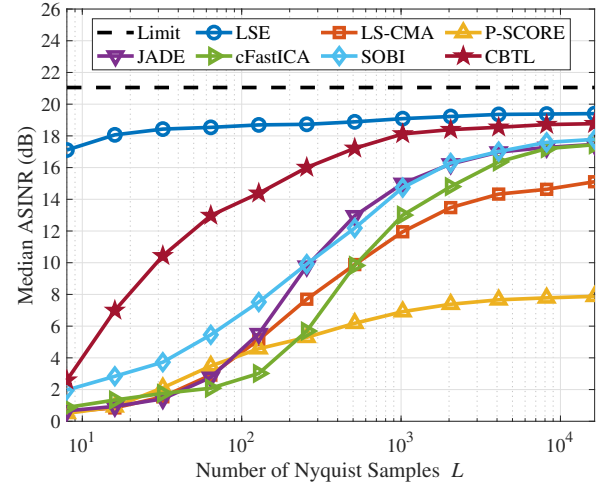
Segments S1 and S2 were for pre-processing and extracting raw IQ samples from segment S3. Experiments were initiated by setting the device time of each USRP to its host computer's time and then commanding the transmit and receive operations to begin at the start of the next minute.

The 16-QAM SOI was generated as in Sec. IV-A at the new Nyquist rate of 625 Ksamples/s and without any noise or synchronization errors. The AWGN interference was generated based on (23). The IQ samples for both signals were normalized to prevent saturation of the transmitters and then saved to files. The GNU Radio software was used on each node to either transmit IQ samples or save the received IQ samples to file at 625 Ksamples/s. We transmitted the interference waveforms from two locations while the SOI transmitted from one, resulting in two datasets with 25 Msamples each.

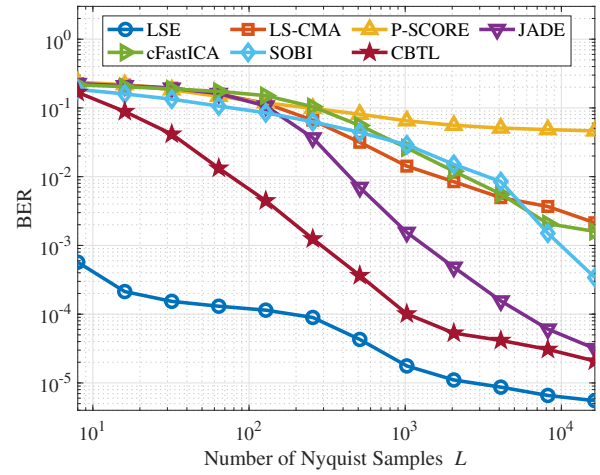
C. Processing Methodology

The datasets were pre-processed to prepare them for the beamforming tests. First, the data from each receiver was made zero mean and the noise powers were estimated from segment S1. Second, the received signal powers, time delays, and frequency offsets were estimated from segment S2. Filters were applied for each tone and detections above the receiver's mean noise power were used to estimate the delays. The power spectral density was estimated and used to estimate the tone frequencies, SNR, and INR. Third, the time delays and frequency offsets were refined using an LSE beamformer and $L = 1000$ Nyquist samples to synchronize to the known signals sent by each transmitter with negligible errors. Candidate delays and frequency shifts were applied to the known signals and the parameters which maximized the ASINRs were determined. Finally, the refined delays were used to extract at least 300 frames with 16384 samples each from segment S3.

We treated each frame as an independent trial and applied the blind beamformers in the same configurations from Sec. IV-D. As previously described, the CBTL beamformer used the pre-trained classifiers from Sec. IV-B with $T = 4$ and *without retraining* them for the larger clock offsets at 5.4 GHz, multipath propagation, or hardware imperfections. Since the true array responses were unknown, (10) could not be used to compute the ASINR. Instead, the beamformed frames were processed by a data-aided 16-QAM demodulator, and the transmitted symbols were subtracted from the demodulated



(a) Median ASINRs versus the sample support.



(b) BERs versus the sample support.

Fig. 14. Beamforming performance for the datasets collected using ORBIT.

symbols to estimate the interference-plus-noise power spectral density and compute the ASINR. Finally, hard decisions were made to produce bitstreams.

D. Results

The results of processing the experimental datasets using different sample supports are shown in Fig. 14. The median ASINRs are shown in Fig. 14a and compared to an upper limit based on the ASINR of the optimal beamformer from Sec. IV-C with a SNR of 20 dB, INR of 40 dB, array responses modeled using the geometry of the radio grid, and receiver phase noise modeled from [46]. We found that the phase noise limited the null depth and reduced the optimal ASINR by about 4 dB. Fig. 14b shows the bit error ratios (BERs), which were computed as the total number of bit errors divided by the total number of bits in the demodulated bitstreams.

The general trends shown in Fig. 14a agreed with the simulated results from Fig. 6a despite differences in the antenna array, angular separations, and impairments of the over-the-air data. We found that even the LSE beamformer was unable to reach the upper limit due to several reasons: non-stationarity of the channel as L increased (due to power

level and clock drift), nonlinearity of the radio hardware, and the channel was not perfectly flat. The first point highlights the need to frequently update the beamforming weights in practice to deal with time-varying channels.

Regardless, we found that the CBTL beamformer was robust to realistic impairments and typically outperformed the other uninformed techniques. When $L = 64$ samples (32 symbols), it improved the typical ASINR by 7.5 dB and attained a BER close to 10^{-2} using about an order of magnitude fewer samples compared to the top-performing BSS technique. Its ASINR converged to within 0.6 dB of the LSE beamformer, and we presume this small gap is due to the mismatch between the classifier's training data and real test data. We found that the P-SCORE algorithm had poor performance due to the presence of correlated multipath components, which necessitates further processing to reconstruct the SOI [10]. These results demonstrate the practical benefit of using the CBTL beamformer to protect a sensitive radio receiver from strong interference.

VI. CONCLUSION

This paper has presented the CBTL beamformer, a novel ML-driven technique that uses a pre-trained signal classifier and transfer learning to train a linear beamforming layer. It operates directly on raw noisy IQ samples without the need for expert-crafted features, synchronization, or knowledge of the array manifold. The proposed approach can help to automate the development of beamformers for new SOIs and protect existing receivers from interference. We extensively evaluated its performance through simulations and validated it using real-world datasets collected using the ORBIT testbed. It improves the typical ASINR compared to state-of-the-art blind beamformers and, unlike several other techniques, can learn to separate spectrally matched and Gaussian-like signals. The improvement is most notable for scenarios where the stationary sample support is limited, which has practical value for blind beamforming in dynamic RF environments. As part of our future work, we plan to develop a low latency implementation, study other pre-trained ML models for beamforming, and extend the concepts to wideband beamforming.

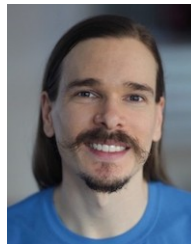
ACKNOWLEDGMENT

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Department of the Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copy-right notation herein.

REFERENCES

- [1] M. Wentz, J. Capper, B. Kurien, K. Forsythe, and K. Chowdhury, "Classification-based transfer learning for blind adaptive receiver beamforming," in *Proc. IEEE 21st Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2024, pp. 59–64.
- [2] J. Li, S. Liu, H. Lu, and S.-H. Hwang, "Coexistence of 5G communication systems with radar altimeters," *IEEE Access*, vol. 12, pp. 32 554–32 568, Feb. 2024.
- [3] C. Fulton, M. Yeary, D. Thompson, J. Lake, and A. Mitchell, "Digital phased arrays: Challenges and opportunities," *Proc. IEEE*, vol. 104, no. 3, pp. 487–503, Mar. 2016.
- [4] K. W. Forsythe, "Utilizing waveform features for adaptive beamforming and direction finding with narrowband signals," *Lincoln Laboratory Journal*, vol. 10, no. 2, pp. 99–126, 1997.
- [5] L. Godara and D. Ward, "A general framework for blind beamforming," in *Proc. IEEE Region 10 Conf. (TENCON)*, vol. 2, Sep. 1999, pp. 1240–1243.
- [6] J. F. Cardoso and A. Souloumiac, "Blind beamforming for non-gaussian signals," *IEE Proc. F (Radar Signal Process)*, vol. 140, pp. 362–370, Dec. 1993.
- [7] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," *IEEE Trans. Signal Process.*, vol. 45, no. 2, pp. 434–444, Feb. 1997.
- [8] E. Bingham and A. Hyvärinen, "A fast fixed-point algorithm for independent component analysis of complex valued signals," *Int. J. Neural Syst.*, vol. 10, no. 1, pp. 1–8, Feb. 2000.
- [9] J. Eriksson, A.-M. Seppola, and V. Koivunen, "Complex ICA for circular and non-circular sources," in *Proc. 13th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2005, pp. 1–4.
- [10] B. Agee, S. Schell, and W. Gardner, "Spectral self-coherence restoral: A new approach to blind adaptive signal extraction using antenna arrays," *Proc. IEEE*, vol. 78, no. 4, pp. 753–767, Apr. 1990.
- [11] J. Shynk and R. Gooch, "The constant modulus array for cochannel signal copy and direction finding," *IEEE Trans. Signal Process.*, vol. 44, no. 3, pp. 652–660, Mar. 1996.
- [12] A. Swindlehurst, S. Daas, and J. Yang, "Analysis of a decision directed beamformer," *IEEE Trans. Signal Process.*, vol. 43, no. 12, pp. 2920–2927, Dec. 1995.
- [13] S. Choi and D. Yun, "Design of an adaptive antenna array for tracking the source of maximum power and its application to CDMA mobile communications," *IEEE Trans. Antennas Propag.*, vol. 45, no. 9, pp. 1393–1404, Sep. 1997.
- [14] J. G. Proakis and M. Salehi, *Digital Communications*, 5th ed. New York, NY, USA: McGraw-Hill, 2008.
- [15] A. Liutkus, J.-L. Durrieu, L. Daudet, and G. Richard, "An overview of informed audio source separation," in *Proc. IEEE Int. Workshop Image Anal. Multimedia Interactive Serv. (WIAMIS)*, Jul. 2013, pp. 1–4.
- [16] K. Zmolikova, M. Delcroix, T. Ochiai, K. Kinoshita, J. Černocký, and D. Yu, "Neural target speech extraction: An overview," *IEEE Signal Process. Mag.*, vol. 40, no. 3, pp. 8–29, May 2023.
- [17] F. Restuccia, S. D'Oro, A. Al-Shawabka, B. C. Rendon, S. Ioannidis, and T. Melodia, "DeepFIR: Channel-robust physical-layer deep learning through adaptive waveform filtering," *IEEE Trans. Wireless Commun.*, vol. 20, no. 12, pp. 8054–8066, Dec. 2021.
- [18] X. Liu, D. Yang, and A. E. Gamal, "Deep neural network architectures for modulation classification," in *Proc. 51st Asilomar Conf. Signals, Syst., Comput.*, Oct. 2017, pp. 915–919.
- [19] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 168–179, Feb. 2018.
- [20] S. Riyaz, K. Sankhe, S. Ioannidis, and K. Chowdhury, "Deep learning convolutional neural networks for radio identification," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 146–152, Sep. 2018.
- [21] T. Oyedare, V. K. Shah, D. J. Jakubisin, and J. H. Reed, "Interference suppression using deep learning: Current approaches and open challenges," *IEEE Access*, vol. 10, pp. 66 238–66 266, Jun. 2022.
- [22] H. A. Kassir, Z. D. Zaharis, P. I. Lazaridis, N. V. Kantartzis, T. V. Yioultis, and T. D. Xenos, "A review of the state of the art and future challenges of deep learning-based beamforming," *IEEE Access*, vol. 10, pp. 80 869–80 882, Aug. 2022.
- [23] K.-L. Du, K. Cheng, and M. Swamy, "A fast neural beamformer for antenna arrays," in *Proc. IEEE Int. Conf. Commun. (ICC)*, vol. 1, May 2002, pp. 139–144 vol.1.
- [24] P. Ramezanpour and M.-R. Mosavi, "Two-stage beamforming for rejecting interferences using deep neural networks," *IEEE Syst. J.*, vol. 15, no. 3, pp. 4439–4447, Sep. 2021.
- [25] S. Schoenbrod, E. Saba, M. Bazdresch, S. Kelly, T. Besard, and K. Fischer, "Blind adaptive beamforming of narrowband signals using an uncalibrated antenna-array by machine learning," in *Proc. IEEE Int. Symp. Phased Array Syst. Technol. (PAST)*, Oct. 2022, pp. 1–7.
- [26] H. Yang, J. Jee, G. Kwon, and H. Park, "Deep transfer learning-based adaptive beamforming for realistic communication channels," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2020, pp. 1373–1376.

- [27] Y. Yuan, G. Zheng, K.-K. Wong, B. Ottersten, and Z.-Q. Luo, "Transfer learning and meta learning-based fast downlink beamforming adaptation," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1742–1755, Mar. 2021.
- [28] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Aug. 2017, p. 1321–1330.
- [29] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh, "Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols," in *Proc. IEEE Wireless Commun. Netw. Conf (WCNC)*, vol. 3, Mar. 2005, pp. 1664–1669.
- [30] H. Van Trees, *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*. New York, NY, USA: Wiley, 2002.
- [31] J. Wang and Y. Chen, *Introduction to Transfer Learning: Algorithms and Practice*. Singapore: Springer Nature, 2023.
- [32] C. Boeddeker, P. Hanebrink, L. Drude, J. Heymann, and R. Haeb-Umbach, "On the computation of complex-valued gradients with application to statistically optimum beamforming," Feb. 2019, *arXiv:1701.00392v2 [cs.NA]*.
- [33] P. Triantaris, E. Tsimballo, W. H. Chin, and D. Gündüz, "Automatic modulation classification in the presence of interference," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Jun. 2019, pp. 549–553.
- [34] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithms and applications," *Neural Networks*, vol. 13, no. 4, pp. 411–430, Jun. 2000.
- [35] S. Haykin, *Adaptive Filter Theory*, 4th ed. Upper Saddle River, NJ, USA: Prentice Hall, 2002.
- [36] B. Widrow and E. Walach, "On the statistical efficiency of the LMS algorithm with nonstationary inputs," *IEEE Trans. Inf. Theory*, vol. 30, no. 2, pp. 211–221, Mar. 1984.
- [37] G. Strang, *Linear Algebra and its Applications*, 3rd ed. Orlando, FL, USA: Harcourt Brace Jovanovich, 1988.
- [38] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, May 2010, pp. 249–256.
- [39] The MathWorks, Inc. (2019) *Modulation Classification with Deep Learning*. Natick, MA, USA. Accessed: Feb. 2023. [Online]. Available: <https://www.mathworks.com/help/deeplearning/ug/modulation-classification-with-deep-learning.html>
- [40] A. van der Veen, "Algebraic constant modulus algorithms," in *Signal Processing Advances in Wireless and Mobile Communications*. Upper Saddle River, NJ, USA: Prentice Hall, 2000, vol. 2.
- [41] N. Kikuma, K. Takai, K. Nishimori, F. Saito, and N. Inagaki, "Consideration on performance of the CMA adaptive array antenna for 16 QAM signals," in *Proc. Int. Symp. Pers., Indoor Mobile Radio Commun.*, vol. 2, Sep. 1995, pp. 677–681.
- [42] J. F. Cardoso, "Source separation of complex signals with JADE," Jun. 2013, accessed: Aug. 2023. [Online]. Available: <https://www2.iap.fr/users/cardoso/code/Jade/jade.m>
- [43] E. Bingham, "Complex FastICA," Oct. 2003, accessed: Jul. 2023. [Online]. Available: http://users.ics.aalto.fi/ella/publications/cfastica_public.m
- [44] A. Delorme and S. Makeig, "EEGLAB: An open source toolbox for analysis of single-trial EEG dynamics including independent component analysis," *J. Neurosci. Methods*, vol. 134, no. 1, pp. 9–21, Mar. 2004.
- [45] ORBIT Wiki. (2022) *Hardware Domains: Grid*. Accessed: Jan. 2024. [Online]. Available: <https://www.orbit-lab.org/wiki/Hardware/bDomains/aGrid>
- [46] Ettus Research. (2022) *UBX Daughterboard Datasheet*. https://www.ettus.com/wp-content/uploads/2022/06/UBX_Data_Sheet-6.22.pdf. Accessed: Apr. 2024.



Michael Wentz received the M.S. degree in electrical and computer engineering in 2015 and is currently pursuing the Ph.D. degree in electrical engineering, both with Northeastern University. He is a Lincoln Scholar at MIT Lincoln Laboratory in the Tactical Edge Communications Group. His research interests include array signal processing, applications of machine learning to wireless communications, algorithm development, and software-defined radio technology.



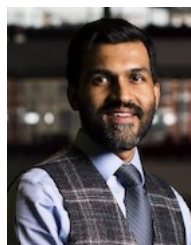
Jack Capper received the B.S. degree in Aerospace Engineering from the Massachusetts Institute of Technology in 2022. He is a Lincoln Scholar at MIT Lincoln Laboratory in the Electronics for Contested Space Group. His research interests include wireless communications, artificial intelligence, and machine learning.



Binoy Kurien received the master's and Ph.D. degrees in engineering sciences from the School of Engineering and Applied Sciences, Harvard University in 2009 and 2016, respectively. He is currently an Assistant Group Leader at MIT Lincoln Laboratory in the Tactical Edge Communications Group. His research interests include dynamic spectrum access, applications of machine learning to wireless communications, array signal processing, and remote sensing.



Keith Forsythe received the B.S. and M.S. degrees in mathematics from the Massachusetts Institute of Technology. He is a Senior Staff Member at MIT Lincoln Laboratory in the Tactical Edge Communications Group, where he has helped develop sensor-array signal processing algorithms that exploit waveform features for beamforming, direction finding, geolocation, and other forms of parameter estimation. His current research interests include applications of machine learning to beamforming.



Kaushik Chowdhury (Fellow, IEEE) received the M.S. degree from the University of Cincinnati in 2006 and the Ph.D. degree from the Georgia Institute of Technology in 2009. He is a professor and holds the Chandra Family Endowed Distinguished Professorship #2 at The University of Texas at Austin. His research involves applied machine learning for spectrum sensing and sharing, systems aspects of networked robotics, and experimental deployment of emerging wireless technologies.