Classification-Based Transfer Learning for Blind Adaptive Receiver Beamforming

Michael Wentz*†, Jack Capper†, Binoy Kurien†, Keith Forsythe†, Kaushik Chowdhury*
*Institute for the Wireless Internet of Things, Northeastern University, Boston, MA, USA
†MIT Lincoln Laboratory, Lexington, MA, USA
wentz.m@northeastern.edu, {jack.capper, bkurien, forsythe}@ll.mit.edu, krc@ece.neu.edu

2024 IEEE 21st Consumer Communications & Networking Conference (CCNC) | 979-8-3503-0457-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/CCNC51664.2024.10454656

***Abstract*—Adaptive receiver beamforming processors typically require expert design and can be limited by their convergence rate in data-starved applications. In this paper, we present a new type of machine learning beamformer using classification-based transfer learning (CBTL) to alleviate these limitations. The architecture consists of a pre-trained signal classifier, in our case a convolutional neural network, prepended by a beamforming layer. Narrowband beamforming weights are optimized by minimizing the classification loss, in turn nulling interference and amplifying a signal of interest (SOI). There are no requirements for calibration of the array, synchronization to the SOI, or training data modulated by the SOI. We describe the CBTL beamformer and demonstrate its effectiveness using several modulated signals. Simulated performance was compared to two well-established methods for blind source separation, and we achieved average signal-to-interference-plus-noise ratio gains of 3 to 9 dB when fewer than 100 samples were available from a 4-element array. The technique shows promise for applications where there is limited prior knowledge and few samples are available for beamformer estimation.**

***Index Terms*—Array processing, adaptive beamforming, blind beamforming, transfer learning, signal classification**

I. Introduction

The advent of miniature and affordable radio frequency (RF) transceivers has led to a continually growing number of wireless devices and crowded spectral environments. Receivers must be capable of mitigating interference from uncooperative transmitters, and one approach has been to utilize an antenna array and blind adaptive beamforming (ABF) processor to null interference and amplify a signal of interest (SOI). However, the initial convergence of typical techniques, such as those based on blind source separation (BSS), can require thousands of waveform samples [1]. This presents a challenge when such data cannot be acquired, for example, due to a rapidly changing channel. Further, ABF processors typically require expert-level algorithm design and considerable time spent tuning the performance. In this paper, we propose classification-based transfer learning (CBTL) to overcome these limitations and improve the performance of narrowband blind ABF.

Research was sponsored by the United States Air Force Research Laboratory and the Department of the Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. 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 copyright notation herein.

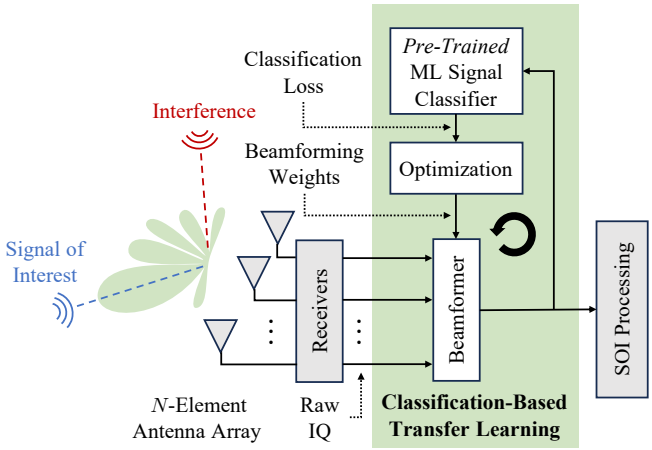

Fig. 1. Block diagram of the proposed system using CBTL for blind ABF. The beamformer is optimized by minimizing the classification loss for the SOI, which nulls interference and provides gain prior to downstream processing.

Interest has been growing in machine learning (ML) techniques for RF physical layer processing, including interference suppression [2], [3]. Motivating factors include their versatility and ability to generalize and learn imperfections not captured by simple models. ML techniques excel at pattern recognition and have demonstrated highly accurate RF signal classification using short observations [4]. Soft decisions from such a classifier can be useful feedback for the processing necessary to extract a SOI from co-channel interference. For example, confidence in the presence of an SOI can be used to adapt a beamformer to preserve features that the classifier has been trained to recognize. We explore this concept of using a pre-trained ML signal classifier and transfer learning for narrowband blind ABF. Simulated performance is compared to two well-established methods for BSS, and we achieve higher average signal-to-interference-plus-noise ratios (SINRs) when limited data is available.

As shown in Fig. 1, the proposed blind ABF system consists of an N-element antenna array, N receivers that convert RF to baseband, and a beamformer that operates on raw in-phase and quadrature (IQ) samples at the Nyquist rate. The beamformer's output is fed into a pre-trained ML signal classifier, and the beamforming weights are optimized by asserting the received data contains the SOI and minimizing the classification loss

without changing any of the parameters within the classifier (i.e., they are frozen). This procedure can achieve good beamforming performance if the SOI has a learnable structure, and if the classifier can differentiate between the SOI and any interference or noise. The technique is blind since it does not require calibration of the array, synchronization to the SOI, or training data modulated by the SOI. We discuss the adaptation of an existing ML signal classification structure and optimization procedure for the CBTL beamformer.

Existing blind ABF techniques typically lack the array manifold but assume some level of knowledge for the SOI or interference waveforms. For example, the widely used constant modulus algorithm assumes a constant envelope modulation. We desire a more general method, and therefore consider BSS since it does not rely on prior waveform knowledge. Well-established techniques include joint approximation diagonalization of eigen-matrices (JADE) and complex fast independent component analysis (cFastICA) [5], [6]. However, these typically assume non-Gaussian signals, which hinders their application to multi-carrier modulations. Further, they generally require detecting the number of signals and later determining which is the SOI. We propose a stand-alone solution by using ML to exploit differences between the observable structures of the SOI and interference, even when both are highly Gaussian.

ML-based beamformers have gained significant attention in recent years [7]. Like traditional ABF algorithms, neural network-based beamformers and their blind variants often learn from an estimate of the spatial covariance matrix of the input data [8]–[10]. Instead, we propose to guide learning based on the structure of the beamformer's output. While transfer learning has been shown to improve the generality of beamforming networks [11], the concept proposed herein, where the underlying network has been trained for a task other than beamforming, does not appear to have been studied in the literature. A related approach was used to optimize temporal filters for improved classification accuracy in [12]. Our work appears to be the first to consider the application to spatial filtering and interference suppression.

The main contributions of this paper are the development and demonstration of the CBTL beamformer, a novel technique for narrowband blind ABF that does not require expert design and can achieve good performance with limited data. Four SOIs were considered to demonstrate its versatility: quadrature phase shift keying (QPSK), 16-quadrature amplitude modulation (16-QAM), and orthogonal frequency division multiplexing (OFDM) communications signals, and frequency modulated continuous wave (FMCW) radar signals. Simulated performance was compared to JADE and cFastICA, and we observed average SINR gains between 3 and 9 dB when fewer than 100 samples are available from a 4-element array.

II. Received Signal Model

Consider the baseband equivalent model for two signals arriving at an N-element antenna array at time instant t

$$\mathbf{z}(t) = \mathbf{v}_1 y_1(t) + \mathbf{v}_2 y_2(t) + \mathbf{n} \tag{1}$$

where $y_1(t)$ is the SOI with steering vector $\mathbf{v}_1$, $y_2(t)$ is the interference with steering vector $\mathbf{v}_2$, and $\mathbf{n}$ is the noise from the N receivers. The steering vectors are modeled for a uniform linear array (ULA) with half-wavelength spacing

$$\mathbf{v}_i = \exp\left(j\pi\big(n - \frac{N-1}{2}\big)\cos(\theta_i)\right) \quad \text{for} \quad i \in \{1, 2\} \tag{2}$$

where $n = [0, 1, \ldots, N-1]$ and the angles of arrival are separated by a scalar η of the null-to-null beamwidth

$$\theta_2 = \theta_1 + 2\eta \sin^{-1}\left(\frac{2}{N}\right). \tag{3}$$

As is typically assumed without synchronization between the transmitter and receiver, the SOI arrives with unknown channel gain α_1, timing offset τ_1, phase offset ϕ_1, and frequency offset f_1,

$$y_1(t) = \alpha_1 x_1(t - \tau_1) e^{j(2\pi f_1 t + \phi_1)} \tag{4}$$

where $x_1(t)$ is the SOI transmitted by the source. As such, this is a narrowband flat fading model. The interference is assumed to be independent and identically distributed (i.i.d.) complex additive white Gaussian noise (AWGN) $\mathcal{CN}(0, \sigma_2^2)$. Similarly, the receiver noises are modeled by N i.i.d. complex AWGN processes $\mathcal{CN}(0, \sigma_n^2)$. The signal power level is $\sigma_1^2 = E[|y_1|^2]$, and the signal-to-noise ratio (SNR) $= \sigma_1^2/\sigma_n^2$ and interference-to-noise ratio (INR) $= \sigma_2^2/\sigma_n^2$ for the data received by a single element of the array.

A snapshot of the received data consists of a matrix of $N \times L$ Nyquist samples from the array. To study the initial convergence of the beamformer, the channel gain and offsets on the SOI are assumed to be static over a snapshot. The value of α_1 is therefore arbitrary since it is captured by the SNR. The phase offset ϕ_1 is modeled as $\mathcal{U}(-\pi, \pi)$ radians. The frequency offset $f_1 = \epsilon \times 10^{-6} f_c$, where f_c is the SOI's nominal RF carrier frequency in Hz and $\epsilon \sim \mathcal{U}(-\epsilon_{\max}, \epsilon_{\max})$ in parts-per-million (ppm). The time delays are modeled by $\tau_1 = D/F_s + \mathcal{U}(0, 1/F_s)$, where D is specified in Sec. IV-A for each SOI, and F_s is the nominal sample rate for the SOI in Hz. The receiver samples at the actual rate of $\bar{F_s} = F_s(1 + \epsilon \times 10^{-6})$ Hz due to the frequency offset.

III. Beamforming Architecture

The proposed CBTL beamforming architecture is shown in Fig. 2 and consists of two main components: (1) a beamformer, which includes a whitening preprocessor, and (2) a pre-trained ML signal classifier, which is a convolutional neural network (CNN) in this work. Both components operate on raw IQ samples to avoid the need for feature development or selection. Transfer learning is applied to adapt the beamforming weights and maximize the output SINR by minimizing the classification loss. The following subsections discuss these components and the training procedure used for CBTL beamforming.

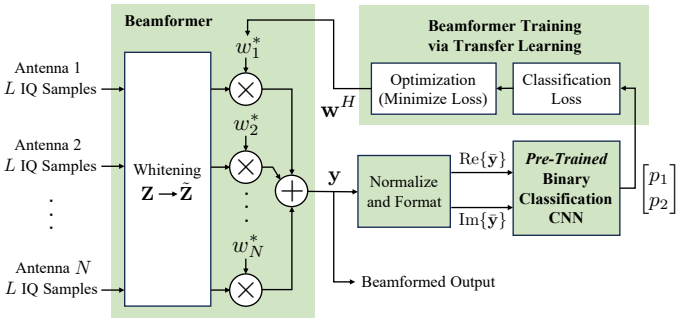

Fig. 2. Block diagram of the CBTL architecture for blind ABF. A pre-trained binary classification CNN is used to optimize the beamforming weights.

TABLE I
BINARY CLASSIFIER CNN LAYOUT

Layer	Output Dimensions
Input	$2 \times L$
Conv2d	$16 \times L/2$
BatchNorm2d	$16 \times L/2$
ReLU	$16 \times L/2$
MaxPool2d	$16 \times L/2$
Conv2d	$24 \times L/4$
BatchNorm2d	$24 \times L/4$
ReLU	$24 \times L/4$
MaxPool2d	$24 \times L/4$
Conv2d	$32 \times L/8$
BatchNorm2d	$32 \times L/8$
ReLU	$32 \times L/8$
AvgPool2d	32×1
Flatten	32×1
Linear	2×1
LogSoftmax	2×1

A. ML Signal Classifier

An example CNN structure from MATLAB's Deep Learning Toolbox [13] was leveraged for our work. We adapted the structure for binary classification, removed hidden layers for computational efficiency, and performed a calibration of the output layer to facilitate beamformer optimization. Our CNN layout is shown in Table I, where the layer names correspond to those from the PyTorch library in which the CBTL beamforming architecture was implemented. We note that the input layer takes two channels, the real and imaginary components of the received waveform samples.

The use of binary classification is motivated by the received signal model (1). The CNN is trained to recognize the presence versus absence of the SOI, with i.i.d. complex AWGN observations in the latter case. This can easily extend to more complicated models using One-vs-Rest classification: binary classifiers can be trained to recognize a specific SOI among other SOIs, interference, and noise. To be useful for beamforming, the classifier must have a low confusion at the operating SNR. It is also desirable for the SOI classification confidences to have a small variance and monotonically increase with the SNR. Classifiers that exhibit a threshold effect wherein the confidence steps from 0 to 1 with a small change in SNR are problematic, since the beamformer may produce the minimum SNR required for confident classification well before achieving the full potential of the array.

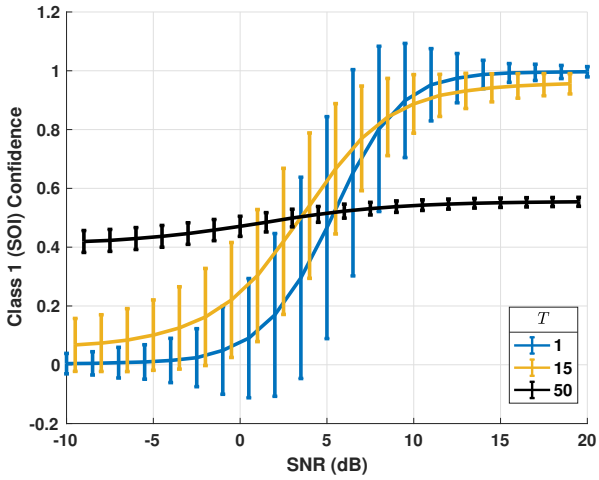

Fig. 3. Mean (solid) and standard deviation (bars) of the classification confidence for RRC filtered QPSK versus the SNR and temperature scaling T. The number of Nyquist samples per frame was $L = 16$.

To ensure the appropriate classifier sensitivity to SNR, we introduced calibration by using temperature scaling [14] on the inputs to the LogSoftmax layer of the CNN. The temperature allows us to control the entropy of the output probability distribution, and the LogSoftmax outputs become

$$p_i = \frac{\exp(l_i/T)}{\exp(l_1/T) + \exp(l_2/T)} \tag{5}$$

where T is the temperature hyperparameter, l_1, l_2 are the outputs of the Linear layer, and p_i is the confidence of class i for $i \in \{1, 2\}$. An example of the effect of temperature scaling is shown in Fig. 3, where the CNNs were trained for root-raised cosine (RRC) filtered QPSK as the SOI (class 1) and i.i.d. complex AWGN as the noise (class 2). We observed that increasing T reduced the standard deviation of the confidence and prevented the classifier from being overly confident at higher SNRs, as desired. However, large values of T caused the classifier to be under confident regardless of the SNR. Using $T = 15$ was found to work well for this CNN and the beamformer optimization described in Sec. III-C, regardless of the type of SOI or number of Nyquist samples L.

For this example, single element training datasets were generated with 10,000 frames per class and $L = 16$ IQ samples per frame. The noise dataset consisted of i.i.d. samples from $\mathcal{CN}(0, \sigma_n^2)$. The model for the SOI dataset followed Sec. II with $N = 1$, $v_2 = 0$, $f_c = 915$ MHz, $F_s = 200$ kHz, $\epsilon_{\max} = 5$ ppm and SNR $= 40$ dB. Each frame consisted of a random sequence of QPSK symbols, and the RRC filter used a roll-off of 0.3, span of 10 symbols, and 2 samples per symbol. A delay of $D = 20$ was used to remove the filter's transient response.

All frames of both datasets were normalized for unit power to ensure that the classifier could not use a feature as simple as amplitude or power level for discrimination.

The Conv2d layers of the CNN were configured with a kernel size of 1×2, based on 2 samples per symbol for the SOI, and the inputs to the LogSoftmax layer were scaled by T^{-1}. Optimization used the negative log-likelihood (NLL) loss function, stochastic gradient descent (SGD) with a learning rate of 0.02, momentum of 0.9, and a learning rate decay factor of 0.1 every 9 epochs. We trained for 30 epochs using a batch size of 256 frames. Separate CNNs were trained for each value of T. Then, the performance of each CNN was evaluated using independently generated test datasets for a range of SNRs.

In this example, $T = 15$ showed our desired behavior and was selected for use in the CBTL beamforming architecture. The LogSoftmax layer was configured with $T = 15$, and new CNNs were trained using the same procedure for different values of L. The results of testing over a range of SNRs are shown in Fig. 4. We observed that the mean confidences were relatively insensitive to L and were constrained such that they do not reach 1 (or drive the loss to 0) while the accuracies still approached 100%. These are desirable characteristics for the CBTL beamforming architecture and offer the potential for good performance with small sample support.

B. Beamformer

The beamformer consists of spatial whitening followed by multiplication with the weights $\mathbf{w}^H = [w_1, w_2, \ldots, w_N]^H$. Whitening is a common preprocessing step in BSS [1], [5], [6], and is implemented as

$$\tilde{\mathbf{Z}} = \mathbf{W}\mathbf{Z} \tag{6}$$

$$= \mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{U}^H\mathbf{Z} \tag{7}$$

where $\mathbf{Z}$ is the input $N \times L$ data matrix, $\mathbf{W}$ is the $N \times N$ whitening matrix, and $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^H = \mathbf{Z}\mathbf{Z}^H$ is the eigendecomposition of the sample covariance matrix with $\mathbf{U}$ unitary and $\mathbf{\Lambda}$ diagonal. The L sample output of the beamformer $\mathbf{y} = \mathbf{w}^H\tilde{\mathbf{Z}}$ is normalized for unit power and formatted for processing by the pre-trained CNN. We implemented whitening in a custom layer and beamforming using a complex-valued Linear layer with N input features, 1 output feature, and zero bias. The beamformer has N learnable parameters, the conjugated weights $\mathbf{w}^H$.

C. Beamformer Training via Transfer Learning

We now discuss the procedure for using the pre-trained classifier to learn the beamforming weights. A new network was defined to take a single $N \times L$ input $\mathbf{Z}$ and perform whitening, beamforming, normalization/formatting, and classification. The normalization and formatting were necessary since the beamformer has no constraints on its weights. Therefore, its output must be normalized for unit power $\bar{\mathbf{y}} = \sqrt{L}\mathbf{y}/||\mathbf{y}||$ before splitting into a frame of real and imaginary components $\mathrm{Re}\{\bar{\mathbf{y}}\}, \mathrm{Im}\{\bar{\mathbf{y}}\}$ for the CNN. Since the CNN is pre-trained, i.e., its parameters are frozen, the only learnable parameters in the new network are $\mathbf{w}^H$.

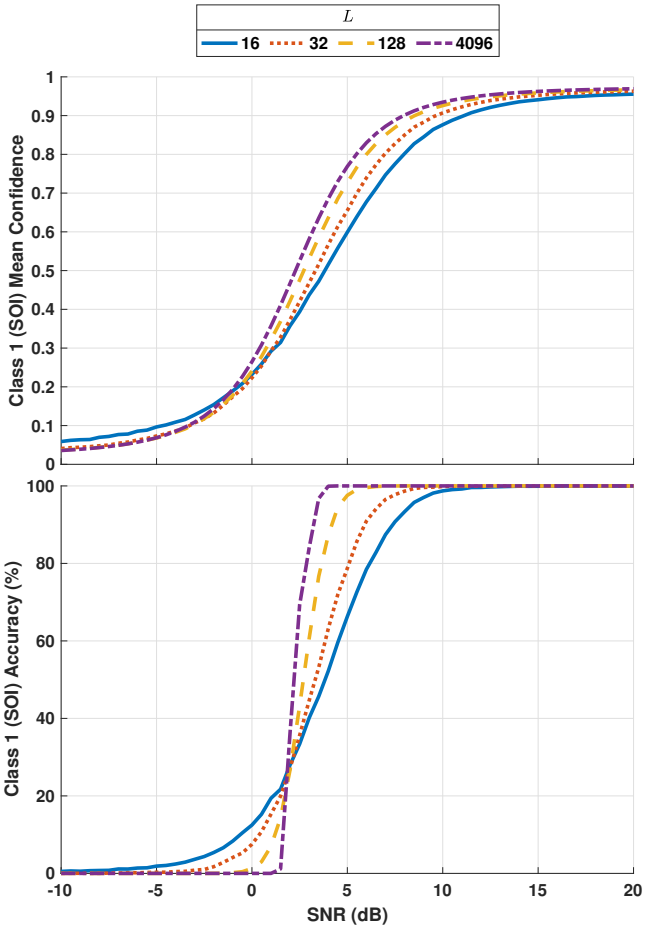

Fig. 4. Mean classification confidence (top) and accuracy (bottom) for RRC filtered QPSK versus the number of Nyquist samples per frame L and SNR. The temperature scaling was $T = 15$.

When training the new network, the input $\mathbf{Z}$ is asserted to contain the SOI and assigned a class label of 1. The weights $\mathbf{w}^H$ are initialized to random values, and optimization uses the NLL loss function and SGD with a learning rate of 0.02, momentum of 0.9, and a learning rate decay factor of 0.1 every 250 epochs. Training proceeds until the classification confidence for the SOI has converged or a maximum number of epochs κ has been reached. Convergence is declared if

$$\frac{1}{M}\left|\sum_{i=2}^{M}\mathbf{p}_1[i] - \mathbf{p}_1[i-1]\right| < \gamma \tag{8}$$

where $\mathbf{p}_1$ is a vector of the classification confidences for the SOI from the last M epochs and γ is a small value. Setting $\kappa = 1,000$, $M = 500$, and $\gamma = 10^{-8}$ was found to work well.

IV. Simulation Experiments

Monte Carlo simulations were used to evaluate the CBTL beamformer and compare its output SINR to the theoretically optimal and that of the JADE and cFastICA algorithms for four types of SOIs. The following subsections describe the dataset generation, beamformer configuration, and results.

A. Dataset Generation

Datasets for training CNNs and testing beamformers were generated via simulation with 10,000 frames per SOI and $L = 4096$ Nyquist samples per frame. A carrier frequency of $f_c = 915$ MHz and maximum frequency offset of $\epsilon_{\max} = 5$ ppm was used. The SOI-specific parameters were as follows:

- QPSK and 16-QAM: RRC filter with a roll-off of 0.3, span of 10 symbols, and 2 samples per symbol. $F_s = 200$ kHz and $D = 20$ to remove the filter's transient response.
- OFDM: Symbols of 1024 QPSK modulated subcarriers and a cyclic prefix of 72 samples. Symbol rate of 100 Hz, $F_s = 137$ kHz, and $D = \text{round}\big(\mathcal{U}(0, 1370)\big)$ for a random receive time within a symbol.
- FMCW: Repeating linear frequency modulated chirps swept from -50 to 50 kHz relative to f_c in 10 ms. $F_s = 125$ kHz and $D = \text{round}\big(\mathcal{U}(0, 1250)\big)$ for a random receive time within a chirp.

For QPSK, 16-QAM, and OFDM, each frame was created using a random sequence of data symbols. Single element datasets for training the CNNs were generated using $N = 1$, $v_2 = 0$, and SNR $= 40$ dB. Multi-element datasets for testing the beamformers were generated using $N = 4$, $\eta = 0.5$, SNR $= 10$ dB, and INR $= 30$ dB.

B. Beamformer Configuration

CBTL beamformers were implemented by truncating each dataset to $L = 2^i$ samples per frame for $i = [3, 4, \ldots, 12]$ and training for each SOI (a total of 40 CNNs). We used $T = 15$ for all CNNs, and the Conv2d layers were configured with kernel sizes of 1×2 for QPSK and 16-QAM, 1×8 for OFDM, and 1×4 for FMCW. These hyperparameters were chosen based on preliminary experiments to calibrate the classification confidences and maximize the beamformed SINR. After training the CNNs, their parameters were frozen and used to learn the beamforming weights as in Sec. III-C.

The implementations of JADE and cFastICA followed [15] and [16]. The number of signals was assumed unknown, so both algorithms estimated beamformers for all N independent components. Further, because it was unknown which independent component was the SOI due to the inherent permutation ambiguity in these algorithms, beamformed SINRs were computed for all N possibilities and the maximum was saved. This can be a disadvantage in practice compared to the CBTL beamformer, which returns only the weights for the SOI.

C. Results

The datasets were processed by each beamformer and the resulting SINRs were computed using

$$\text{SINR} = \frac{\tilde{\mathbf{w}}^H \mathbf{R}_1 \tilde{\mathbf{w}}}{\tilde{\mathbf{w}}^H \mathbf{R}_2 \tilde{\mathbf{w}}} \tag{9}$$

where

$$\mathbf{R}_1 = \sigma_1^2 \mathbf{v}_1 \mathbf{v}_1^H \tag{10}$$

$$\mathbf{R}_2 = \sigma_2^2 \mathbf{v}_2 \mathbf{v}_2^H + \sigma_n^2 \mathbf{I} \tag{11}$$

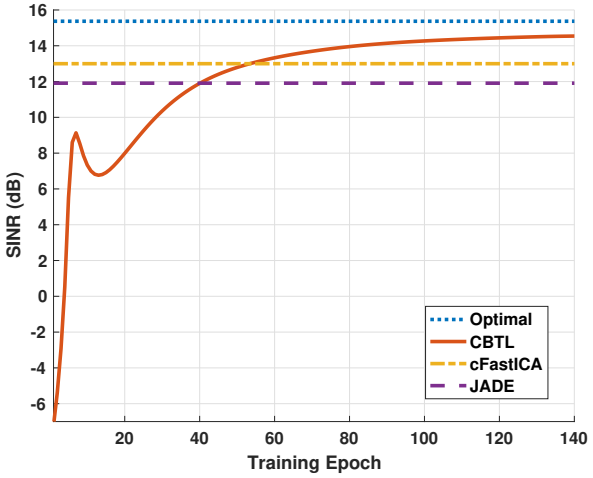

Fig. 5. Example of the output SINR while training the CBTL beamformer for the QPSK SOI in AWGN interference. $N = 4$ element ULA, SNR $= 10$ dB, INR $= 30$ dB, and angle of arrivals separated by $\eta = 0.5$ beamwidths.

are the covariance matrices for the SOI and interference plus noise, $\mathbf{I}$ is the $N \times N$ identity matrix, and $\tilde{\mathbf{w}} = \mathbf{W}^H \mathbf{w}$ are the unwhitened beamforming weights. An example of the CBTL beamformer's convergence for a single frame of the QPSK SOI with $L = 128$ is shown in Fig. 5, where the optimal SINR was computed using the minimum variance distortionless response beamforming weights $\mathbf{w}_{\text{opt}} = \beta \mathbf{R}_2^{-1} \mathbf{v}_1$ [17], which assumes perfect knowledge of $\mathbf{v}_1$ and $\mathbf{R}_2$, and

$$\beta = \frac{N}{\mathbf{v}_1^H \mathbf{R}_2^{-1} \mathbf{v}_1} \tag{12}$$

in our implementation. We observed that the CBTL beamformer achieved an output SINR within 1 dB of optimal after training for 111 epochs and achieved SINR gains of 1.4 and 2.5 dB over cFastICA and JADE, respectfully. The CBTL beamformer continued to improve the SINR by another 0.1 dB until it declared convergence after 939 epochs.

Average SINRs achieved by each beamformer for all datasets are shown versus L in Fig. 6. We observed that the CBTL beamformer consistently outperformed JADE and cFastICA when there was limited sample support, and the performance was just as good for larger values of L. However, we note that this benefit came at the cost of comparatively long compute times, making the current implementation of the CBTL beamformer most applicable to offline processing.

Except for the OFDM case, all beamformers approached optimal performance as L increased. Neither JADE nor cFastICA were expected to perform well for OFDM, since both the SOI and interference were effectively Gaussian distributed and will fail to be separated [5], [6]. Similarly, 16-QAM is more Gaussian-like than QPSK, and the performance of JADE and cFastICA suffer as expected. The results show that CBTL beamformers can achieve good performance if there is structure that can be recognized by the CNN, even when the distribution of the SOI resembles a Gaussian.

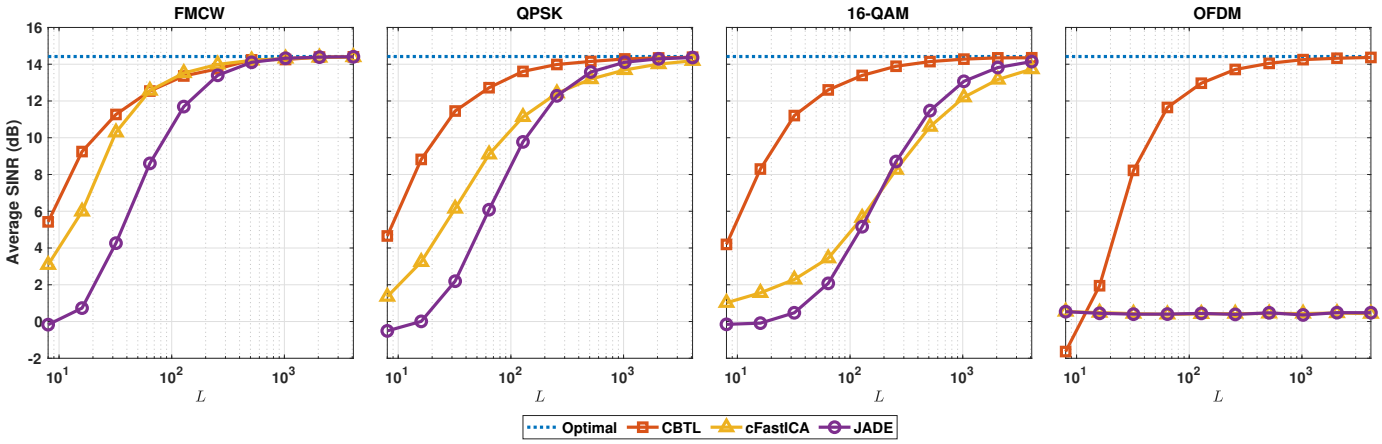

Fig. 6. Average beamformed SINRs versus the number of Nyquist samples per frame L over 10,000 trials for the FMCW, QPSK, 16-QAM, and OFDM (left to right) SOIs in AWGN interference using a $N = 4$ element ULA, SNR $= 10$ dB, INR $= 30$ dB, and angle of arrivals separated by $\eta = 0.5$ beamwidths.

The ability for these results to apply to real-world scenarios depends on how closely the real data matches the received signal model from Sec. II. Although the model captures significant aspects of the wireless channel, differences in real data may arise for several reasons. Since both the model and beamformer are narrowband, strong multipath reflections that cause frequency selectivity will result in a performance loss. Any distortions introduced by RF components, such as an amplifier in saturation, will alter the structure of an SOI and should be included in the training dataset. While this paper considers only a single i.i.d. AWGN interferer, the proposed CBTL beamforming architecture does not preclude other types of interference waveforms or multiple interferers; however, the classifier should be pre-trained to discriminate between any interference and the SOI. As part of our future work, we plan to study the performance of training on synthetic data and testing on real over-the-air data.

V. Conclusions

This paper has introduced a novel architecture for narrowband blind ABF based on transfer learning. The technique leverages the recent success of CNNs for RF signal classification and uses a simple procedure to learn beamforming weights for extracting an SOI from interference. Simulations were used to evaluate the performance for a variety of SOIs, and the proposed technique demonstrated average SINR gains of 3 to 9 dB over well-established methods for BSS when the sample support was small. The CBTL beamformer developed herein easily extends to other SOIs and models without expert feature extraction or algorithm development. Future work will include evaluation using over-the-air datasets and improving the optimization procedure to enable real-time operation.

References

[1] J. Eriksson, A.-M. Seppola, and V. Koivunen, "Complex ICA for circular and non-circular sources," in *Proc. 13th Eur. Signal Process. Conf.*, 2005, pp. 1–4.

[2] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. and Netw.*, vol. 3, no. 4, pp. 563–575, 2017.

[3] 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, 2022.

[4] 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, 2018.

[5] J.-F. Cardoso and A. Souloumiac, "Blind beamforming for non-gaussian signals," in *IEE Proc. F (Radar and Signal Process.)*, vol. 140, no. 6. IET, 1993, pp. 362–370.

[6] 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. 01, pp. 1–8, 2000.

[7] A. M. Elbir, K. V. Mishra, S. A. Vorobyov, and R. W. Heath, "Twenty-five years of advances in beamforming: From convex and nonconvex optimization to learning techniques," *IEEE Signal Process. Mag.*, vol. 40, no. 4, pp. 118–131, 2023.

[8] T. Sallam, A. B. Abdel-Rahman, M. Alghoniemy, Z. Kawasaki, and T. Ushio, "A neural-network-based beamformer for phased array weather radar," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 9, pp. 5095–5104, 2016.

[9] S. Mohammadzadeh, V. H. Nascimento, R. C. de Lamare, and N. Hajarolasvadi, "Robust beamforming based on complex-valued convolutional neural networks for sensor arrays," *IEEE Signal Process. Lett.*, vol. 29, pp. 2108–2112, 2022.

[10] 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. on Phased Array Syst. Technol.*, 2022, pp. 1–7.

[11] 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. Wirel. Commun.*, vol. 20, no. 3, pp. 1742–1755, 2021.

[12] 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. Wirel. Commun.*, vol. 20, no. 12, pp. 8054–8066, 2021.

[13] The MathWorks Inc., "Deep learning toolbox: Modulation classification with deep learning," Natick, MA, USA, 2022.

[14] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2017, pp. 1321–1330.

[15] J. F. Cardoso, "Source separation of complex signals with JADE," 2013. [Online]. Available: https://www2.iap.fr/users/cardoso/code/Jade/jade.m

[16] E. Bingham, "Complex FastICA," 2003. [Online]. Available: http://users.ics.aalto.fi/ella/publications/cfastica_public.m

[17] H. Van Trees, *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*. New York, NY, USA: Wiley, 2002.