Zero-Shot Dynamic Neural Network Adaptation in Tactical Wireless Systems

Shahriar Rifat*, Jonathan Ashdown‡, Kurt Turck‡ and Francesco Restuccia*
‡ Air Force Research Laboratory, United States
* Institute for the Wireless Internet of Things, Northeastern University, United States
Corresponding author e-mail: rifat.s@northeastern.edu

***Abstract*—Models based on deep neural networks have attracted interest in a myriad of applications in the wireless landscape, including spectrum intelligence, modulation recognition, and radio fingerprinting, among others. One key challenge that is currently inhibiting the application of such models in real-world tactical scenarios is that their performance radically changes with continuously changing dynamic wireless channel conditions. Existing work tackle this by performing efficient fine-tuning or adaptation using as few samples as possible. However, some prior knowledge about the new conditions is involved in the process. In this paper, we propose for the first time zero-shot dynamic neural network adaptation (zDNA) (i.e., without any additional training samples) of wireless classification models after deployment in unseen conditions. Specifically, we show that by changing only the affine transformation parameters and normalizing the learned features in different layers of the neural network in an online manner without any labeled samples, we can achieve superior performance in dynamic conditions. Our proposed approach is evaluated on the publicly available RadioML 2018.01A dataset, to test its adaptability to dynamically changing signal-to-noise ratio (SNR) conditions. Performance improvement consistency in all unseen test scenarios (up to 24% in low SNR regime) demonstrates the applicability of our framework in real-world contexts.**

I. Introduction

Although the interest in application of neural networks in the physical layer traces back to the 90's [1], it has been mostly grounded upon mathematical modeling and information theory. Over the past few years, the significant success of Deep Neural Networks (DNNs) in fields such as computer vision and natural language processing, joint with the stringent Quality of Service (QOS) requirements of Fifth Generation (5G) and beyond cellular systems and the release of Radio Frequency Machine Learning Systems (RFMLS) [2] program by DARPA, have spurred researcher to apply data-driven models in the wireless domain.

Although the advantages of using DNNs in the wireless domain have been widely demonstrated, **existing work has shown that the non-stationary, dynamic and unpredictable effect of the wireless channel, as well as hardware-level transceiver impairments, may cause the classification accuracy to plummet** [3]. Among other factors, the performance loss is mainly due to transceiver-related impairments (e.g., phase noise, I/Q imbalance), which are highly dependent from time-varying temperature and voltage oscillations [4], as well as channel-related impairments, which are strongly tied to the ongoing propagation environment and can hardly be predicted in advance [5]. For example, radio fingerprinting accuracy may decrease by 30% when tested with data collected days after it was trained [6], while beam detection accuracy may decrease by up to 65% when a different antenna and/or receiver orientation are considered [7]. **The key issue is that Deep Learning (DL) models are trained using labeled data collected in specific channel conditions and noise**. To address the problem, data augmentation and few-shot learning based approaches have been proposed. Data Augmentation based approaches try to increase the model's robustness during training by showing the modes synthetically perturbed data so that the model is not confused when similar samples are processed during inference time [8, 9]. Few-shot learning in wireless refers to the ability of a model to learn from a small number of labeled examples and generalize to new, unseen channel conditions, usually done through leveraging transfer learning, meta learning or self-supervised losses [10, 11]. Although such approaches are demonstrated to work in some conditions, **performance of them was not tested without any prior knowledge in completely unseen channel conditions where no labelled data could be collected**. Our proposed framework aims to take a step toward that unexplored territory by introducing zDNAs in wireless DNNs. Zero-shot adaptation refers to the ability of a DNN to adapt and perform in a new environment without any prior knowledge and supervised retraining. Our approach is to modulate the activations of different layers of our trained model with the estimated statistics in the newly-encountered channel through the Batch Normalization (BN) layer, and shift and scale those activations with entropy statistics. The idea of feature modulation through BN statistics is an approach that has been successfully applied in a range of tasks from unsupervised domain adaptation [12], robotic kitting [13] and adaptation against image corruption [14]. However, to the best of our knowledge, has not been applied and tested in wireless classification task for adapting models in unseen channel condition. Our key finding is that **by changing only the affine transformation parameters and normalizing the learned features in different layers of the neural network in an online manner without any labeled samples, we can achieve superior performance in dynamically changing conditions**. We tested our approach on the publicly available RadioML 2018.01A dataset [15], to test its adaptability to

Approved for Public Release; Distribution Unlimited: AFRL-2023-3238.

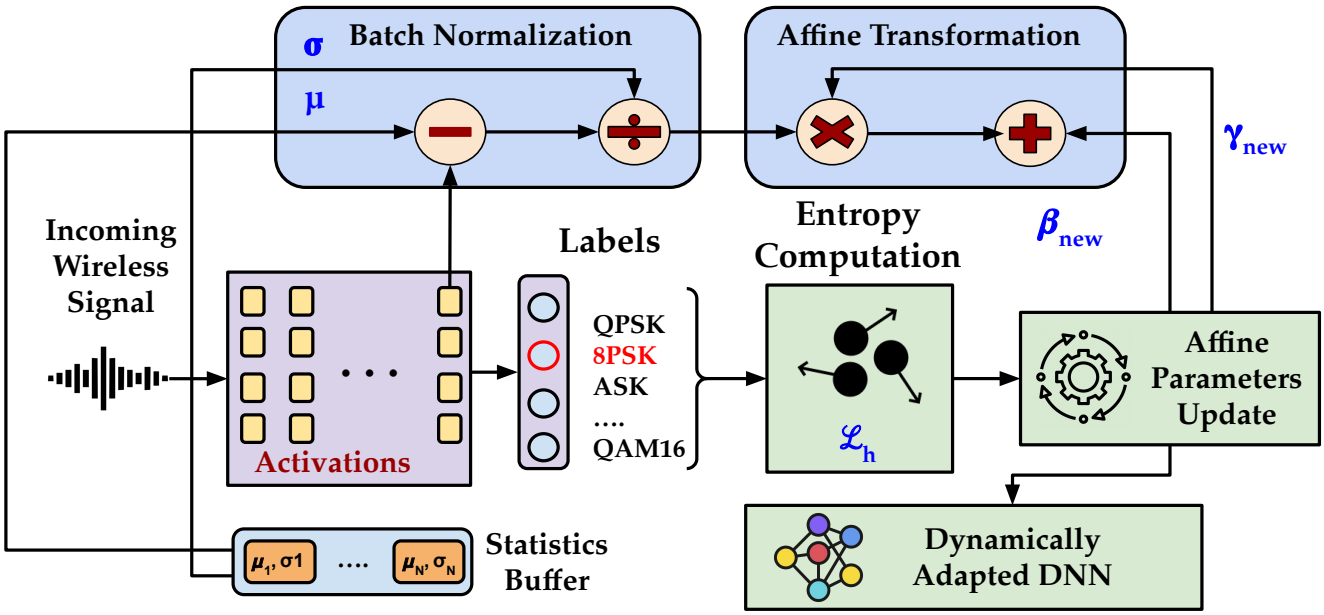


Fig. 1: Overview of our zero-shot dynamic neural network adaptation process.

dynamically changing SNR conditions. Performance improvement consistency in all unseen test scenarios (up to 24% in low SNR regime) demonstrates the applicability of the framework in real-world contexts.

The rest of the paper is structured as follows. Section II discusses earlier attempts to improve the performance of DNN-based wireless classification networks. In Section III, we describe the common assumptions in channel-adaptive DNN models, their flaws, and the mathematical rationale for our assumptions. Section IV explains in detail approach. In Section V, we describe our experimental setup and the related results. This is followed by a summary of the paper and concluding remarks in Section VI.

II. Related Work

Over the last few years, deep neural networks (DNN) have enjoyed tremendous success in the networking and wireless research communities [16, 17]. DNNs have the unique capability of addressing classification and/or optimization problems that have intractable complexity and/or closed-form solutions are difficult to obtain. Convolutional neural networks, in particular, can operate on unprocessed I/Q samples without using feature extraction and/or selection algorithms [18]. **Although the effectiveness of DNNs is proven, a plethora of prior work has exposed the vulnerability of data-driven approaches to changing and/or dynamic environmental conditions**, both related to the hardware circuitry of the wireless platform and the different propagation environments that the DNNs need to operate on at test time. Among other application scenarios, the issue has been unveiled for problems such as radio fingerprinting [3, 6], and beam fingerprinting [7]. Indeed, it is inevitable that tactical systems will be subjected to environment dynamics (*i.e.*, noise/interference) and performance metrics (*e.g.*, link or network throughput) that will drastically change over time. We can also expect that the objective function of DNNs will also change over time – for example, a tactical platform may prefer wireless performance at the beginning of its lifetime and then prefer energy consumption reduction when its batteries are exhausted.

As far as possible solutions are concerned, to make DNNs perform well in an environment with impulsive noise authors in [19] have proposed an auto correlation function based on hyperbolic tangent cyclic spectrum. To make DNN classification model robust against noise, in [20] the authors proposed to project the raw I/Q samples into a grid like image with polar transformation and assign color to them based on temporal accumulation probability. The work in [8] has shown the application of different techniques of imaging time series data to transform I/Q samples with images and reported superior performance with ResNet-18 architectures for images. Although working on the transformed input space of wireless signals have been agreed upon to boost the performance, none of them have reported results on how such transformation performs in unseen conditions, which is the main technical target of this paper.

Shang et al. [21] proposed a knowledge transfer mechanism to transfer the knowledge from a trained DNN classifier to a U-Net based signal reconstruction module to use latent embeddings learned by the network for mapping low SNR signal to a proper format. A dynamic distribution adaptation process was proposed in [22], which learns a shared latent subspace of signals across source and target domain and transfers knowledge to target domain signal efficiently by using fewer samples. In such transfer learning-based approaches, some labeled samples from target domain are available. But in our proposed setting, target domain is considered

completely unseen which resembles closely with practical scenarios. Recent work [23] leverages an autoencoder for end to end communication that adapts itself with the change in wireless channel by learning only from a few labeled samples (few-shot). Such an approach is feasible in scenarios where collecting a few labeled samples in a changed environment is inexpensive in terms of time and effort, which is not the setting we are considering in this paper.

III. Problem Statement and Motivation

We mathematically model the wireless channel (*i.e.*, medium of propagation, imperfection of transceivers) as a transfer function that transform an input $\mathbf{x} \in \mathbb{R}^d$ into an output $\mathbf{z} \in \mathbb{R}^d$. We assume that there exists a transfer function that represents the channel which is stochastic and non-linear in nature and can be written as $\mathbf{z} = \mathbf{h}_{\theta_\mathbf{c}}(\mathbf{x}, \mathbf{r})$, where θ_c denotes the parameters of the channel models and all the random aspects (*e.g.*, frequency and phase offset, hardware imperfections,noise) are being captured by $\mathbf{r}$. In a traditional wireless classification problem, a dataset is provided, i.e., $\mathcal{D} = \{\mathcal{Z}, \mathcal{Y}\}$, where $y \in \mathcal{Y} := \{1, ..., k\}$ are the class labels of k-classes of $z \in \mathcal{Z}$, which are actually associated with the unaltered input data $x \in \mathcal{X}$ to train a DNN model $f_\theta(.)$ that gives us the probability distribution $P_\theta(y \mid x)$.

The overall objective can be modeled as follows:

$$\hat{y}(z) = \arg\max_{y \in \mathcal{Y}} P_\theta(y \mid z) = f_\theta\left(h_{\theta_c}(x, r)\right) \quad \mathbf{s.t.} \tag{1}$$

$$\max_{\theta} \quad \mathbb{E}_{(x,y)}[\mathbf{1}(\hat{y} == y)] \tag{2}$$

In Equation 2, $\mathbf{1}(c)$ is an indicator function that takes value 1 or 0 conditioned on c being true or false. In a real-world communication scenario, both the channel parameters $\theta_\mathbf{c}$ and the random factor $\mathbf{r}$ are non-stationary and changes over time according to hardware imperfections and the mobility of users. In other words, $\mathbf{h}^{\mathbf{t}}_{\theta_\mathbf{c}}(\mathbf{x}, \mathbf{r}) \neq \mathbf{h}^{\mathbf{t}'}_{\theta_\mathbf{c}}(\mathbf{x}, \mathbf{r})$. From a machine learning perspective, this scenario can be defined as the standard covariate shift assumption [24], i.e., $P^t(z|\mathbf{x}) \neq P^{t'}(z|\mathbf{x})$ and $P^t(y|\mathbf{x}) = P^{t'}(y|\mathbf{x})$ where $P^t(.)$ denotes the distribution of data at time t. Some of the previous works address this issue by transforming the input to some different mode (*e.g.* image, grid constellation) [20, 25] so that the *distribution shift* is slightly less in that transformed domain. Another popular line of approach in the literature [21, 26] is to learn some transformation function $\mathcal{T}_{\theta_p}(.)$ that is typically another Neural Network (NN) that transforms z in such a way that effect in the change of channel parameters is compensated $P^t\left(\mathcal{T}_{\theta_p}(z)\,|\mathbf{x}\right) \sim P^{t'}\left(\mathcal{T}_{\theta_p}(z)\,|\mathbf{x}\right)$. However, to learn such a transformation function that performs well across changing wireless condition, $\mathcal{T}_{\theta_p}(.)$ needs to be trained on a huge number of scenarios in a supervised manner. This approach ultimately needs the collection of labelled data set in all possible channel conditions, which is not feasible in real-world highly-dynamic settings.

Due to such limitations, we postulate that the DNN $f_\theta(.)$ should dynamically update itself without any supervision to facilitate the utilization of such DNN classifier in actual real-world scenarios. Our proposed zDNA approach does not assume any prior knowledge about new channel condition and update itself only from the statistics and inference results of a few unlabelled samples $\mathcal{D}^t = \{\mathcal{Z}^t\}$ without the associated labels $\mathcal{Y}^t$.

IV. Proposed Zero-Shot Adaptation Approach

We describe below our zDNA, which is also summarized in Figure 1.

A. Batch Normalization

Batch normalization (BN) is one of the most important tools for training a DNN [27]. As data is passed through the layers of a DNN, activation values in a number of layers becomes large. Thus, BN helps stabilizing the training by making the optimization landscape smoother [28]. Let $A^l \in \mathbb{R}^{B \times C_l \times N_l}$ is a batch of activation tensors of l^{th} convolutional layer, where B corresponds to the batch size, C_l denotes the number of channels in l^{th} layer and N_l is the dimension of activations in each channel. A BN layer first calculates $\mu_c = \frac{1}{|B||N_l|}\sum_{b \in B,\, n \in N_l} A_l$ and $\sigma_c = \sqrt{\frac{1}{|B|}\sum_{b \in B}(A_l - \mu_c)^2}$ and subtracts μ_c from all input activation's in channel c. Subsequently, BN divides the centered activation by the standard deviation σ_c. The following normalization is applied for each channel for all batches:

$$BN\left(A^l_{b,c,n\,l}\right) \leftarrow \gamma_c \times \frac{A^l_{b,c,n\,l} - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}} + \beta_c \quad \forall\, b, c, n_l \tag{3}$$

Here, γ_c and β_c are the affine scaling and shifting parameters followed by normalization, while $(\epsilon > 0)$ is a small constant added for numerical stability. The normalized and affine transformed outputs are passed to the next $(l+1)^{th}$ layer while the normalized output is kept to the l^{th} layer. BN also keeps track of the estimate of running mean and variance to use during the inference phase as a global estimate of normalization statistics, and γ_c and β_c are optimized with other model parameters through back propagation.

B. Entropy as a Proxy to Sense Dynamic Change

When there is a change in channel condition, which would be viewed as distribution shift in data, the model will be less confident on its prediction. Specifically, during inference time, we can measure the entropy using the following equation.

$$H(\hat{y}) = -\sum_c p(\hat{y}_c) \log p(\hat{y}_c) \tag{4}$$

where, $\hat{y}_c$ is the logit of the prediction of the model. Optimizing over a single sample prediction would assign all probability to the most probable class even if the prediction is incorrect. To prevent such a problem, we propose to consider both the mean and variance of entropy over a batch of predictions to optimize our desired parameters. The unsupervised

zero shot loss across last s samples can be calculated using the following equation:

$$\mathcal{L}_h = \frac{1}{|s|}\sum_s H\left(\hat{y}\right) + \left(H\left(\hat{y}\right) - \left(\frac{1}{|s|}\sum_s H\left(\hat{y}\right)\right)\right)^2 \quad (5)$$

From Figure 2, it can be observed that, for a randomly sampled batch of data, the mean of entropy is higher when the DNN model is tested in unseen noise conditions although there are some inconsistencies (e.g., SNR = 2 dB). The rationale behind such inconsistencies can be attributed to the randomness of incoming data samples as the model may predict erroneously with high confidence for some samples in unseen scenario. So, Unlike [14], taking also the variance of entropy over the samples, such inconsistencies can be prevented, as in unseen noise conditions the model's predictions should be more erratic. It should also be noticed that our objective is unsupervised as it does not need any annotations but only predictions.

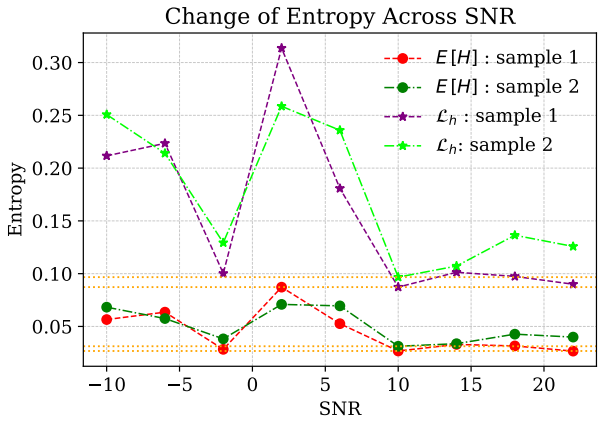


Fig. 2: Entropy of two randomly sampled batches for different SNR when model is trained for SNR=10 dB.

C. Dynamic Update of the DNN

Although updating every DNN parameter in response to a dynamic change would give the best performance in specific scenario, altering θ would inhibit us to reset our model to ideal operating conditions (desirable SNR and channel conditions). Furthermore, $f_\theta(.)$ is highly non-linear and θ is high dimensional for a typical DNN. **Trying to optimize all the parameters of the DNN in new channel condition would make the adaptation very sensitive to changes**. Such an unstable setting can easily make our inference task in new scenario degrade rather than improving. To improve stability and efficiency, and in line with previous work on domain adaptation [14, 29], we instead only update features that are linear (scales and shifts), and low-dimensional (channel-wise).

Specifically, only the affine transformation parameters $\{\gamma_{l,c},\ \beta_{l,c}\}$ of each channel c of each normalization layer l of the model $f_\theta(.)$ are kept in the computational graph that are to be optimized. All other parameters $\theta \setminus \{\gamma_{l,c},\ \beta_{l,c}\}$ are discarded from the computational graph. The global estimate of normalization statistics $\{\bar{\mu}_{l,c}, \bar{\sigma}_{l,c}\}$ are also discarded and stored in a buffer for resetting the model to its initial state. With each incoming sample x^t during inference we update and store the running mean and variance of each BN layer of the model in a buffer using ${\xi_{l,c}}^{new} = (1-m)\tilde{\xi}_{l,c} + m\,\xi^t_{l,c}$ for last s samples, where m is the momentum, $\xi^t_{l,c} = \left\{\mu^t_{l,c}, \sigma^t_{l,c}\right\}$ and $\tilde{\xi}_{l,c}$ are the statistics of current sample and previously stored statistics respectively. We also calculate $H(\hat{y})$ and $\mathcal{L}_h$ of the incoming samples using equation 4 and equation 5 respectively and store them in a buffer.

We update the affine parameters $(\gamma_{l,c},\ \beta_{l,c})$ of all BN layers by a single backward pass using $\mathcal{L}_h$ and replace the previously stored statistics with $\xi^{new}_{l,c} = \left\{\mu^{new}_{l,c}, \sigma^{new}_{l,c}\right\}$ stored in the statistics buffer.

V. Performance Evaluation

A. Description of Dataset

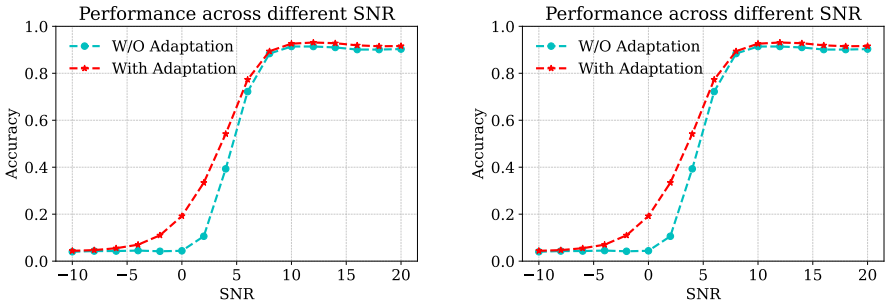
We use the widely popular RadioML 2018.01A dataset [15] of wireless domain classification task and use it to study the performance of our zero-shot adaptation framework. RadioML 2018.01A is a synthetic data set using GNU Radio that includes radio signals of different modulations at varying SNR levels. The dataset consists of 24 commonly used modulations.

B. Experimental Setup

We first split the dataset of each SNR into training and testing dataset with 80% data in the training set and 20% as the testing set. We use three different seeds for generating the random split and all the performance reported are the average across three different runs. Furthermore, we discard data with very low SNR $(< -10dB)$ as the information present in such are not representative of our actual task and also with very high SNR $(> 20dB)$ as the performance increase get saturated around this range. We use the original Residual Network (ResNet) structure proposed in [30] with 18 convolution layers (ResNet-18) as the base model for adaption although this method would work with any Convolutional Neural Network (CNN) that is equipped with BN layer. The number of channels, kernel sizes and strides were kept same as the original ResNet-18 architecture except of two modifications. As we pass the raw I/Q samples directly to the network as a two-channel sequence, the layers in the original ResNet-18 structure intended for processing images were replaced by their one dimensional alternatives (*e.g.* Conv1D, BatchNorm1D). We also had to add an extra Fully Connected (FC) layer after the Global Average Pooling (GAP) layer to stabilize the information passing from the convolutional backbone to the dense classifier. We train our model for 100 epochs with Adam Optimizer (learning rate =0.01 and weight decay =0.01) and cosine annealing learning rate scheduler. In the adaptation phase, we set the learning rate to 0.001.

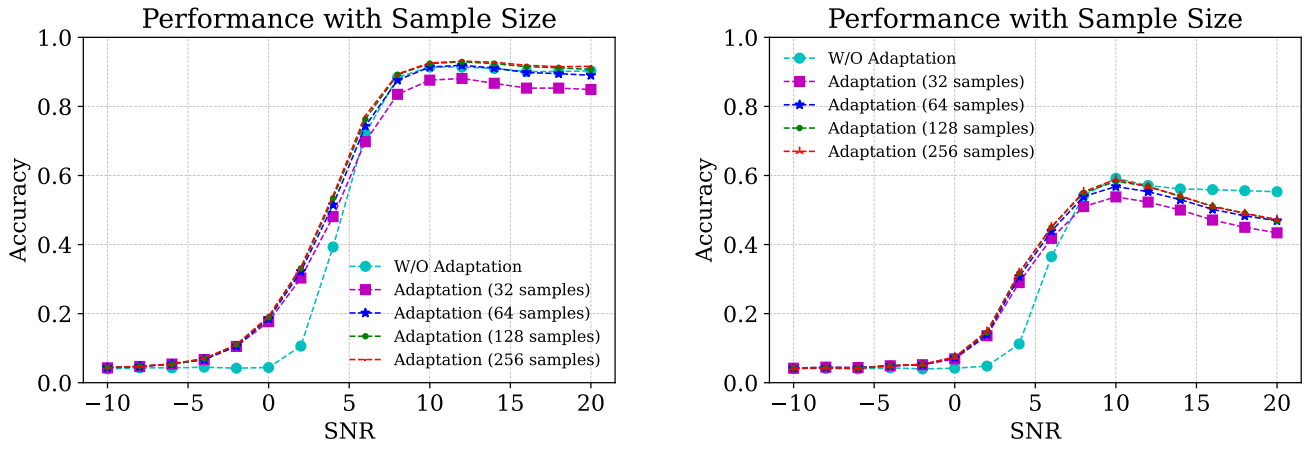
C. Performance Across Different Unseen Noise Condition

For the classification of different modulation types, the random noise and interference do not provide any necessary



(a) Model Trained on only SNR = 10 dB (b) Model Trained on only SNR = 20 dB

Fig. 3: Performance of the Adaptation Framework across different unseen SNR data



(a) ResNet18 Backbone Model (b) VGG Backbone Model in [15]

Fig. 4: Performance of the proposed approach with different sample size; model trained on SNR = 10 dB

information to the DNN, since it has to learn the meaningful original modulation features for generalization. If we trained a DNN with data consisting of high noise and interference, a DNN with sufficient capability would just overfit to some irrelevant features that is not representative of the actual modulation class. From Figure **??**, we can observe that a DNN trained with data at $(SNR = 0\,dB)$ performs much worse than when tested with data from high SNR, which means it did not learn the features of actual modulation class. Thus, adapting the model with a few samples from the high SNR range cannot restore the performance.

Our approach gives significant performance improvement (up to 24%) when the model is trained on high SNR condition $[10dB, 20dB]$ **but tested in completely unseen SNR condition.** From Figure 3a and 3b, it can be seen that performance in SNR range [-6,6] has been improved by significant amount while slight performance increase can also be observed in higher SNRs. This demonstrates our framework's ability to retrieve performance in low SNR range without any extra SNR prediction and compensation network, that are trained on labeled samples as opposed to previous approaches [21, 26].

One desired capability of the zero-shot framework is that it should adapt with as few unlabeled samples as possible. Also, the adaptation should be agnostic to the DNN architecture. To understand such capabilities of the proposed framework, we have trained the two DNN architectures proposed in [15] and make inference on all other SNR scenarios with our adaptation scheme with sample size, $s = 32, 64, 128, 256$. From Figure 4, we can observe that for both ResNet-18 and VGG backbones, there is consistent performance improvement for SNR $-2dB$ to $10dB$. For ResNet-18 backbone model, accuracy starts to degrade from SNR $\geq 6\,dB$ for sample size of 32 as it can not capture the ideal estimation of normalization parameters with this amount of samples. For the VGG backbone model in Figure 4b, the accuracy improvement is significant (up to 22%) for low SNR range but it starts to degrade after that point even with sample size of 256 with adaptation. We postulate that such behaviour is observed when the model does not have enough representative power for the task (test accuracy $\sim 60\%$), replacing the transformation statistics calculated during training with estimated ones only confuses the already under-fitted model.

VI. Conclusion

In this paper, we propose for the first time *zero-shot adaptation* (i.e., without any additional training samples) of wireless

classification models after deployment in unseen conditions. Conversely from existing work which performs fine-tuning or adaptation using as few samples, we show that by changing only the affine transformation parameters and normalizing the learned features in different layers, we can achieve superior performance in dynamically changing conditions. Experimental results on the RadioML 2018.01A dataset with different SNR conditions show performance improvement up to 24% in low SNR regimes, which demonstrates the applicability of our framework in real-world contexts.

Acknowledgment of Support and Disclaimer

We sincerely thank Dr. Erik Blasch for concept coordination and paper editing. This work is funded in part by the National Science Foundation (NSF) grant CNS-2134973, CNS-2134567, CNS-2120447, ECCS-2146754, OAC-2201536, CCF-2218845, and ECCS-2229472, by the Air Force Office of Scientific Research under contract number FA9550-23-1-0261, by the Office of Naval Research under award number N00014-23-1-2221, and by an effort sponsored by the U.S. Government under Other Transaction number FA8750-21-9-9000 between SOSSEC, Inc. and the Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views expressed are those of the authors and do not reflect the official guidance or position of the United States Government, the Department of Defense, the United States Air Force or the United States Space Force.

References

[1] E. Chesmore, "Neural network architectures for signal detection and demodulation," in *1989 Fifth International Conference on Radio Receivers and Associated Systems*, pp. 1–4, IET, 1990.

[2] T. Rondeau, "Radio frequency machine learning systems (rfmls)," 2017.

[3] A. Al-Shawabka, F. Restuccia, S. D'Oro, T. Jian, B. C. Rendon, N. Soltani, J. Dy, K. Chowdhury, S. Ioannidis, and T. Melodia, "Exposing the Fingerprint: Dissecting the Impact of the Wireless Channel on Radio Fingerprinting," *Proc. of IEEE Conference on Computer Communications (INFOCOM)*, 2020.

[4] L. Samara, M. Mokhtar, Ö. Özdemir, R. Hamila, and T. Khattab, "Residual self-interference analysis for full-duplex ofdm transceivers under phase noise and i/q imbalance," *IEEE Communications Letters*, vol. 21, no. 2, pp. 314–317, 2016.

[5] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep learning-based channel estimation," *IEEE Communications Letters*, vol. 23, no. 4, pp. 652–655, 2019.

[6] F. Restuccia, S. D'Oro, A. Al-Shawabka, M. Belgiovine, L. Angioloni, S. Ioannidis, K. Chowdhury, and T. Melodia, "DeepRadioID: Real-Time Channel-Resilient Optimization of Deep Learning-based Radio Fingerprinting Algorithms," *Proc. of ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2019.

[7] M. Polese, F. Restuccia, and T. Melodia, "DeepBeam: Deep Waveform Learning for Coordination-Free Beam Management in mmWave Networks," in *Proceedings of the Twenty-second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, pp. 61–70, 2021.

[8] A. Abbas, V. Pano, G. Mainland, and K. Dandekar, "Radio modulation classification using deep residual neural networks," in *MILCOM 2022-2022 IEEE Military Communications Conference (MILCOM)*, pp. 311–317, IEEE, 2022.

[9] M. Piva, G. Maselli, and F. Restuccia, "The tags are alright: Robust large-scale rfid clone detection through federated data-augmented radio fingerprinting," in *Proceedings of the Twenty-second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, pp. 41–50, 2021.

[10] D. Liu, P. Wang, T. Wang, and T. Abdelzaher, "Self-contrastive learning based semi-supervised radio modulation classification," in *MILCOM 2021-2021 IEEE Military Communications Conference (MILCOM)*, pp. 777–782, IEEE, 2021.

[11] A. Owfi, F. Afghah, and J. Ashdown, "Meta-learning for wireless interference identification," in *2023 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, IEEE, 2023.

[12] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou, "Revisiting batch normalization for practical domain adaptation," *arXiv preprint arXiv:1603.04779*, 2016.

[13] M. Mancini, H. Karaoguz, E. Ricci, P. Jensfelt, and B. Caputo, "Kitting in the wild through online domain adaptation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1103–1109, IEEE, 2018.

[14] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell, "Tent: Fully test-time adaptation by entropy minimization," *arXiv preprint arXiv:2006.10726*, 2020.

[15] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air Deep Learning Based Radio Signal Classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, 2018.

[16] J. Jagannath, N. Polosky, A. Jagannath, F. Restuccia, and T. Melodia, "Machine Learning for Wireless Communications in the Internet of Things: A Comprehensive Survey," *Ad Hoc Networks*, vol. 93, p. 101913, 2019.

[17] C. Zhang, P. Patras, and H. Haddadi, "Deep Learning in Mobile and Wireless Networking: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[18] U. K. Majumder, E. P. Blasch, and D. A. Garren, *Deep learning for radar and communications automatic target recognition*. Artech House, 2020.

[19] J. Ma, M. Hu, T. Wang, Z. Yang, L. Wan, and T. Qiu, "Automatic modulation classification in impulsive noise: Hyperbolic-tangent cyclic spectrum and multibranch attention shuffle network," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–13, 2023.

[20] C.-F. Teng, C.-Y. Chou, C.-H. Chen, and A.-Y. Wu, "Accumulated polar feature-based deep learning for efficient and lightweight automatic modulation classification with channel compensation mechanism," *IEEE transactions on vehicular technology*, vol. 69, no. 12, pp. 15472–15485, 2020.

[21] X. Shang, H. Hu, X. Li, T. Xu, and T. Zhou, "Dive into deep learning based automatic modulation classification: A disentangled approach," *IEEE access*, vol. 8, pp. 113271–113284, 2020.

[22] M. Wang, H. Jiang, Q. Tian, J. Fu, and G. Si, "Terfda: Tensor embedding rf domain adaptation for varying noise interference," *Physical Communication*, vol. 58, p. 102015, 2023.

[23] J. Raghuram, Y. Zeng, D. Garcia, S. Jha, S. Banerjee, J. Widmer, and R. Ruiz, "Fast and sample-efficient domain adaptation for autoencoder-based end-to-end communication," 2021.

[24] J. Quinonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset shift in machine learning*. Mit Press, 2008.

[25] S. Huang, Y. Jiang, Y. Gao, Z. Feng, and P. Zhang, "Automatic modulation classification using contrastive fully convolutional network," *IEEE Wireless Communications Letters*, vol. 8, no. 4, pp. 1044–1047, 2019.

[26] K. Yashashwi, A. Sethi, and P. Chaporkar, "A learnable distortion correction module for modulation recognition," *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 77–80, 2018.

[27] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pp. 448–456, pmlr, 2015.

[28] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How does batch normalization help optimization?," *Advances in neural information processing systems*, vol. 31, 2018.

[29] T. Gong, J. Jeong, T. Kim, Y. Kim, J. Shin, and S.-J. Lee, "Note: Robust continual test-time adaptation against temporal correlation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27253–27266, 2022.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.