Learning the unknown: Improving modulation classification performance in unseen scenarios

Erma Perenda*, Sreeraj Rajendran*, Gerome Bovet†, Sofie Pollin* and Mariya Zheleva‡
{erma.perenda, sreeraj.rajendran, sofie.pollin}@esat.kuleuven.be, gerome.bovet@armasuisse.ch, mzheleva@albany.edu
* WaveCore, ESAT, KU Leuven, †Cyber-Defence Campus, armasuisse Science&Technology,
‡Department of Computer Science, University at Albany - SUNY

IEEE INFOCOM 2021 - IEEE Conference on Computer Communications | 978-1-6654-0325-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/INFOCOM42981.2021.9488835

***Abstract*—Automatic Modulation Classification (AMC) is significant for the practical support of a plethora of emerging spectrum applications, such as Dynamic Spectrum Access (DSA) in 5G and beyond, resource allocation, jammer identification, intruder detection, and in general, automated interference analysis. Although a well-known problem, most of the existing AMC work has been done under the assumption that the classifier has prior knowledge about the signal and channel parameters. This paper shows that unknown signal and channel parameters significantly degrade the performance of two of the most popular research streams in modulation classification: expert feature-based and data-driven. By understanding why and where those methods fail, in such unknown scenarios, we propose two possible directions to make AMC more robust to signal shape transformations introduced by unknown signal and channel parameters. We show that Spatial Transformer Networks (STN) and Transfer Learning (TL) embedded into a light ResNeXt-based classifier can improve average classification accuracy up to 10-30% for specific unseen scenarios with only 5% labeled data for a large dataset of 20 complex higher-order modulations.**

***Index terms* —** Modulation Classification, Spatial Transformer Network, Transfer Learning, Spectrum Sensing.

I. Introduction

Although AMC has received considerable research interest for more than 40 years, most developed methods have been designed under the assumption of prior knowledge of transmitter technology (e.g., symbol duration, bandwidth, upsampling and signal shaping) and channel conditions. This prior knowledge plays a critical role in the spectrum sensor's configuration and the model training used for transmitter fingerprinting. Unless a sensor is configured in accordance with the transmitter properties, the respective spectrum scans will not cater to the transmitter fingerprinting task. Similarly, training a model in certain channel conditions/transmitter properties while applying it to different channel conditions/properties also leads to performance deterioration.

Emerging spectrum sensing applications, however, cannot readily rely on the availability of prior channel and transmitter information. Consider crowd-sourced spectrum enforcement as one example application [1]–[3], whose goal is to automatically detect and counteract rogue radio transmissions. To this end, a wide targeted band from 30MHz to 6GHz is continuously monitored by low-cost heterogeneous sensors [1,4]. This target band is (i) orders of magnitude larger than the instantaneous bandwidth of commodity sensors (i.e., 100 [5] to 1000 [6] times larger) and (ii) extremely dynamic and heterogeneous as it houses the majority of the commercial, scientific and federal radio technologies. Thus, it is unrealistic to assume that a sensing system would have detailed prior knowledge about a transmitter to custom-configure the scan. It is, thus, essential to develop novel AMC approaches that can (i) fingerprint a large variety of transmitter technologies while (ii) lacking prior knowledge of the transmitter characteristics and (iii) relying on low-cost and intermittent spectrum scans.

In this paper, we consider three major oversights in the existing literature: (i) limited set of target modulations, (ii) stringent assumptions of signal shaping parameters, and (iii) prior knowledge of channel conditions. We begin by evaluating the performance deterioration of baseline methods from the State-of-the-Art (SotA) when the above assumptions are relaxed. We then propose a novel approach to address these performance gaps by enhancing the learned model architectures and employing agile model training with limited supervision.

Oversight 1: Limited set of target modulations. Most existing AMC methods were evaluated on low-order modulation datasets (e.g., BPSK-8PSK or QAM16), with the most challenging being QAM-64 [7]–[12]. However, emerging wireless systems utilize complex and higher-order modulations to improve spectral efficiency and reliability, such as QAM-1024 in WiFi-6 and 5G and APSK in satellite systems. The classification of such modulations is challenging even at a high Signal-Noise Ratio (SNR). To the best of our knowledge, only [13] considers a complex dataset of 24 modulations, including high-order QAM and APSK. While [13] makes an important headway towards high-order AMC, it still requires prior knowledge of signal shaping and channel conditions.

Oversight 2: Knowledge of signal shaping parameters. As wireless communication systems operate in band-limited channels and mostly employ sampling frequencies much higher than Nyquist's, it is necessary to shape a transmitted signal to limit its bandwidth. Different techniques can be applied to produce a signal with the target bandwidth, e.g., upsampling of the signal with factor 2 results in the same bandwidth as the shaping of the signal with Raised Cosine (RC) filter with a roll-off factor of 0.5. Since those techniques are unknown at the receiving sensor, they introduce uncertainty about the symbol duration of the received signal.

Oversight 3: Knowledge of the channel conditions. AMC models are often trained under certain channel assumptions (e.g., Additive White Gaussian Noise (AWGN) [7], Rayleigh [11]–[14], or Rician [15]). However, if the channel conditions upon runtime modulation classification differ

from those upon training, prior work suffers a significant performance deterioration [16]. Recent efforts focus on the transferability of AWGN training to runtime classification with Rician channel conditions [16], using raw In-phase/Quadrature (I/Q) data and High-Order Cumulants (HOC) as input features. While [16] makes an important first step towards understanding model transferability across channel conditions, there are multiple directions that remain unexplored: (1) consideration of complex datasets; (2) employing SotA deeper networks; and (3) evaluation across more complex and realistic channel conditions such as variable path Rayleigh channels. Blind channel estimators such as constant modulus [17], expectation-maximization [18], and High-Order Moments (HOM) and HOC based [19] algorithms are not directly applicable in our setting. [17,18] are plagued with high computational and sensing costs. HOM and HOC [19] have a lower complexity; however, they are highly-sensitive to signal shaping and channel conditions, as we show later.

Following a detailed exploration of these oversights, we show that existing AMC approaches suffer substantial performance deterioration with complex datasets, unknown signal shaping, and unknown channel conditions. Signal shaping leads to modulation variations, which significantly deteriorate classification performance if not captured in training. Similarly, unknown channel conditions trigger learned classifiers inapplicable out of context, in which they were trained. To address these issues, we explore two avenues for enhanced AMC using Deep Learning (DL). We begin by enhancing the SotA [20] with STN in search of resilient feature extraction. In addition, we also employ transfer learning, which improves classification models by incorporating limited training (e.g., in cases with rarely-seen channel conditions).

This paper makes the following contributions:

- We identify key criteria such as signal shaping, channel conditions, and dataset complexity that hamper the DL AMC methods' practical applicability to emerging spectrum applications. We employ a rigorous empirical evaluation to pinpoint performance drawbacks under realistic circumstances.
- We explore the performance deterioration caused by the lack of channel-aware training and develop a principled approach for DL model training across channel conditions to ensure model applicability to unknown channels.
- We are the first to conceptualize and demonstrate the adverse effects of unknown signal shaping parameters on the data-driven and expert feature-based AMC methods. Accordingly, we develop a principled approach for model training that ensures model robustness with unknown shaping parameters.
- We show that STN and TL can substantially improve the robustness of DL-based approaches to unknown signal properties.

II. Related work

A vast literature exists on AMC with three main methodological themes: Likelihood-Based (LB), Feature-Based (FB), and Deep Learning (DL). Due to their high computational cost and the required prior knowledge about the signal model, LB approaches have limited practical applicability. On the other hand, FB and DL approaches have received considerable research attention. Below we summarize the pros and cons of each approach.

1) LB methods: They are optimal in the Bayesian sense, as they minimize the probability of wrong classification [21]. The AMC task is defined as a multiple composite hypothesis-testing problem, where the number of hypotheses is equal to the number of target modulations. The major limitation of conventional LB methods is the careful design and selection of signal and noise models [22], which requires prior knowledge about all signal and channel parameters. In the literature, novel LB methods that blindly estimate unknown parameters have also been proposed [17]. Thus, their performance and computational cost depend on the accuracy and complexity of the employed estimation algorithms.

2) FB methods: These methods extract discriminative features from underlying raw data (e.g., I/Q or Power Spectral Density (PSD)) to classify individual modulations, which requires a substantial system design-time knowledge. In contrast to LB methods, FB methods [23]–[26] are sub-optimal in the Bayesian sense but have lower complexity. Classification features such as instantaneous time-domain features, transformation and spectral features, and statistical features (HOC and HOM) have been widely adopted in the FB methods. Novel features that are more discriminative for certain modulation formats have also been investigated [7]–[9]. As a classifier, FB methods have mostly employed Support Vector Machines (SVM), Decision Tree Classification (DTC), K-Nearest Neighbour (KNN), and shallow Neural Network (NN). Some FB methods have combined those classifiers to boost performance [9].

3) DL methods: The input features for those methods are raw I/Q or PSD data, avoiding the need for system design-time knowledge required for the heavy feature extraction in FB methods. DL methods are orthogonal to FB methods, and can be separated into two major directions: Recurrent Neural Network (RNN) [12] and Convolutional Neural Network (CNN) [10,11,13]. RNNs are suited for temporal feature extraction, whereas CNNs learn spatial features. RNNs tend to have higher running time, as they are difficult to parallelize due to non-linear sequence dependencies [12]. At the same time, CNNs, albeit faster, are more sensitive to noise compared to RNNs. Recently, inspired by RNN, new CNN-based models such are Residual Neural Network (ResNet) [27] and Aggregated Residual Transformations for Deep Neural Networks (ResNeXt) [20] have been proposed. Those models outperform SotA CNN models, as shown for the ResNet-based AMC model in [13]. Building on ResNet, ResNeXt is a multi-branch architecture that follows a split-transform-merge strategy. In [20], the authors show that stacking more parallel layers with the same hyper-parameters (filter size and depth) keeps the network design simple and leads to better results than going deeper or wider. To date, there has not been

any AMC model based on ResNeXt. Furthermore, there have been a few proposals that combine two or more Deep Neural Networks (DNNs) connected in multiple serial or parallel branches to improve performance [10,14,28,29]. The success of DL methods heavily depends on the size and diversity of the available labeled dataset. Semi-supervised DL techniques (mostly Generative Adversarial Network (GAN)) have been employed to enhance classification performance by generating additional high-quality labeled data from a small amount of seed data [30]–[32]. Such generated data are very similar to labeled seed data as they come from the same distribution, while data distributions from unseen scenarios differ. Thus, all DL models perform poorly on the unseen signal shapes. Every change of a signal or channel parameter would affect the signal shape and introduce uncertainty in DNNs performance. This paper investigates which signal and channel parameters lead to performance degradation and give possible directions to make DL methods more robust to unseen signal shapes.

III. Signal model

In this section, we introduce the modulated signal as input to the classifier. Wireless communication systems must be designed under bandwidth and power constraints determined either by law or by technical requirements. We assume that the wireless channel has an available bandwidth equal to B centered around f_c. At the transmitter, there are different operations (e.g., upsampling, pulse shaping, and upconversion) that can be performed to produce a signal with a target bandwidth B. Those operations might introduce some uncertainty about the symbol rate at the receiving sensor since the sensor that is sampling at a fixed frequency $f_s^{(r)}$ does not know the bandwidth, B of the signal, and which operations are performed at the transmitter to produce the signal with that bandwidth. Thus, there might be an oversampling factor $K = f_s^{(r)}/B$ at the receiving sensor. The baseband representation of the transmitted signal is given as $s_{bb}(t) = \sum_{n=-\infty}^{+\infty} a_n g(t - n/f_s)$, where a_n is the n-th complex symbol of the input data, f_s is sampling frequency at the transmitter and $g(t)$ is an impulse response of the pulse shaping filter. This signal has non-zero spectral power over the entire $[-f_s/2, f_s/2]$. If $f_s > B$, then upsampling and interpolation of the sequence a_n is necessary to narrow its spectral width. An upsampling factor, L is chosen such that $L = f_s/B$. The upsampler inserts $L - 1$ zeros between every two input samples, and must be followed by a low-pass filter to remove multiple copies of the upsampled spectrum. The most popular pulse shaping filter used in wireless communications is the RC filter [33], whose passing bandwidth is defined by the roll-off factor α which ranges between 0 and 1, and rarely exceeds 0.5. The upsampled and filtered transmitted signal can be expressed as

$$s(t) = \sum_{k=-\infty}^{+\infty} b_k g(t - \frac{kL}{f_s}), \tag{1}$$

where $b_k = a_n$ for $k = nL$ and $b_k = 0$ otherwise. The signal, $s(t)$, is sent over a dynamic wireless fading channel with an impulse response h_c. Assuming one antenna at the sensor, the distorted and noise-corrupted received signal, $r(t)$ is given as

$$r(t) = e^{-j2\pi\Delta f t} s(t - \tau) \circledast h_c(t) + v(t), \tag{2}$$

where τ is the timing offset, Δf is the frequency offset, and $v(t)$ is AWGN with mean 0 and variance $2\sigma_v^2$. We assume that the receiving sensor is working at the same frequency as the transmitter, $f_s^{(r)} = f_s$, and also the carrier frequency f_c is known. Thus, the oversampling ratio, K, at the receiving sensor is equal to the upsampling factor, L. This signal is fed to the input of AMC classifier that does not know α or L. The AMC classifier's task is to correctly select a modulation format from a pool of known N_{mod} candidate modulations by examining the received signal, $r(t)$.

IV. Datasets and baseline methods

In this section, we introduce the datasets and selected SotA AMC methods used to examine the discussed oversights.

A. Synthetic datasets

We generated several synthetic datasets, as specified in Table I, by varying L, α, f_s, channel models and instance size. An instance denotes a vector of I/Q samples. We consider two sets of modulations:

- *Simple set*, containing 11 low-order modulations typically used in the literature: BPSK, QPSK, 8-PSK, 16/64-QAM, PAM4, GFSK, CPFSK, BFM, DSB-AM and SSB-AM;
- *Complex set*, containing the simple ones and 9 more modulations: OQPSK, 32/128/256-QAM, 16/32/64/128/256-APSK.

We generated 1000 instances with a certain size for each combination (modulation type, α, L, f_s, SNR, channel type), resulting in $91,520,000$ and $166,400,000$ instances for the simple set and complex set, respectively.

B. Baseline AMC models

We employ six baselines from the literature: five based on DL and one FB that uses HOC. The 5 DL methods are: LSTM [12], PF-CNN [14], 2D-CNN [11], ResNet [13], 1D-CNN [13], and the FB method is HOC [7,25,26]. The first three models were optimized for a simple dataset which is an easy classification task at high SNR, with a few DNNs layers (e.g., two layers in LSTM and four layers in 2D-CNN) and very short signal observations (e.g., instance size of 128). However, such shallow DNNs cannot capture higher-order modulations' inherent properties even with extended signal observations. 1D-CNN and ResNet employ a deeper network structure, and to the best of our knowledge, they are the only two models evaluated on a complex dataset. We employ 1D-CNN with seven Conv1D layers with 64 filters of size 3 and stride 1 (note that these parameters were not specified in the original paper [13]). ResNet and 1D-CNN accurately distinguish complex and higher-order modulations, but their resilience to unknown signal and channel parameters is unknown.

Our final baseline is a FB classifier that uses HOC features. To perform well on the complex dataset, HOCs require longer

Table I: Datasets specifications

Name	f_s $[MHz]$	L	RC α	Channel Model
DS_AWGN	$[0.2, 0.6, 1, 1.5, 2]$	$[2, 3.2, 4, 6.4, 8, 10.67, 16, 32]$	$[0.15, 0.25, 0.35, 0.45]$	AWGN with SNR in range $[-6, 18]$ dB
DS_Rayleigh	$[0.2, 0.6, 1, 1.5, 2]$	$[2, 3.2, 4, 6.4, 8, 10.67, 16, 32]$	$[0.15, 0.25, 0.35, 0.45]$	Rayleigh with a path profile: Delays: $[0, 4.5, 8.5]$ μs; Gains: $[0, -1, -5]$ dB, and AWGN with SNR in range $[-6, 18]$ dB. Maximum Doppler shift is 4 Hz.
DS_Rician	$[0.2, 0.6, 1, 1.5, 2]$	$[2, 3.2, 4, 6.4, 8, 10.67, 16, 32]$	$[0.15, 0.25, 0.35, 0.45]$	Rician with K factor of 4, a path profile with: Delays: $[0, 0.25, 3, 8]$ μs; Gains: $[0, -2, -10, -3]$ dB, and AWGN with SNR in range $[-6, 18]$ dB. Maximum Doppler shift is 4 Hz.
DS_Rayleigh_2	$[0.2, 0.6, 1, 1.5, 2]$	$[2, 3.2, 4, 6.4, 8, 10.67, 16, 32]$	$[0.15, 0.25, 0.35, 0.45]$	Rayleigh with a path profile: Delays: $[0, 0.2, 3, 9]$ μs; Gains: $[0, -7, -2, -1]$ dB, and AWGN with SNR in range $[-6, 18]$ dB. Maximum Doppler shift is 8 Hz.

signal observations. Thus, we set out instance size to 1024 I/Q samples and use C_{20}, C_{21}, C_{40}, C_{41}, C_{42}, C_{60}, C_{61}, C_{62}, C_{63}, C_{80}, C_{84} [26] employed with a simple linear SVM classifier.

C. Training and testing settings

A seed is used to generate random mutually exclusive instance indices, which are then used to split the data in each dataset mentioned in Table I into three sets, training, validation and testing using order 80:10:10 respectively. Each evaluated AMC method is implemented using TensorFlow [34]. As an optimizer, we opted for Adam [35], with a learning rate of 0.001. This learning rate is a reasonable trade-off between slow convergence at lower rates and inaccurate results at higher rates. Training is done through 80 epochs and a batch size of 256. The models are trained and tested on a GPU server with eight Nvidia RTX 2080Ti cards. All presented classification accuracies are averaged over the whole SNR range of $[-6, 18]$ dB.

V. Limitations of Existing Work

We now evaluate the baselines' limitations (§IV-B) with complex datasets, unknown signal shaping and channels.

A. Sensitivity to complex datasets

Most prior AMC methods have been evaluated on simple low-order modulation datasets. In this section, we examine whether these models maintain robust performance with a complex dataset. The analysis is done on the complex DS_AWGN dataset with $L = 4$, $\alpha = 0.35$ and $f_s = 200$ kHz. We keep the same model configurations while varying the instance size.

Fig. 1 shows that shallow DNN models such as 2D-CNN have reduced performance for complex datasets, even when the instance size is increased. PF-CNN, 1D-CNN, and ResNet are deeper structures, and their accuracy increases up to 16% with a higher instance size. Those models are CNN based and much more computationally and memory efficient than Long-Short Term Memory (LSTM). We employ LSTM with default training parameters and no hyper-parameter tuning. LSTM fails to converge for the instance sizes of 512 and 1024 on the complex dataset. For further examination, we selected the ResNet and 1D-CNN with the instance size of 1024 as they performed the best for complex datasets.

B. Sensitivity to signal shaping

According to the signal model described in Section III, there are three signal shaping parameters (L, RC α, f_s), the settings of which might impact the AMC performance. All SotA

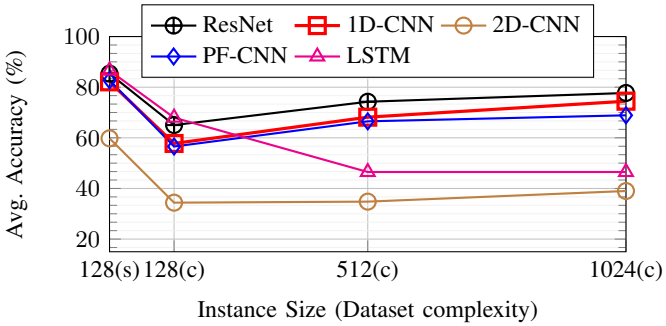


Figure 1: SotA AMC models evaluated on DS_AWGN ($L = 4$, $f_s = 200$ kHz, $\alpha = 0.35$). "s" denotes a simple and "c" denotes a complex dataset.

AMC models assume that those parameters are known prior to modulation classification and consequently consider only a limited subset of all possible practical system realizations. What remains unclear is how the lack of prior knowledge of these parameters at the sensor will affect the classification performance. In what follows, we tackle this question.

1) Upsampling factor variations (L): Baselines are trained for $L = 8$, $\alpha = 0.35$ and $f_s = 200$ kHz in AWGN, Rayleigh+AWGN and Rician+AWGN channel conditions. Testing is done for other L values while channel conditions, RC α and f_s remain the same as for training.

Fig. 2 shows that 1D-CNN and ResNet have similar sensitivity to unknown Ls in all channels. Both have a small accuracy drop of c. 10% for $L = 6.4$, and a large accuracy drop of c. 60% for $L = 2$. HOC-SVM achieves the lowest accuracy for known $L = 8$, but it is also the least sensitive to unknown Ls with an accuracy drop of 25% for $L = 2$ with Rician+AWGN channel. These results illustrate that a lack of prior knowledge of L is critical for AMC performance.

2) RC roll-off factor variations (α): Baselines are trained for $L = 8$, $\alpha = 0.35$ and $f_s = 200$ kHz in AWGN, Rayleigh+AWGN and Rician+AWGN channels. Testing is done for unknown RC α values, while channel conditions, L and f_s remain the same as for training. Fig. 3 shows that prior methods' performance is consistent across all roll-off values and channel conditions, with a peak accuracy drop of 8% for 1D-CNN with AWGN channel. These results indicate that unknown filter settings might not harm classification accuracy.

3) Sampling frequency variations (f_s): Finally, we evaluate the resilience of AMC performance to the sampling frequency f_s. Baselines are trained for $L = 8$, $\alpha = 0.35$ and $f_s = 600$ kHz in AWGN, Rayleigh+AWGN and Rician+AWGN channels. Testing is done for unknown f_s values, while channel conditions, L and RC α remain the same as for training.

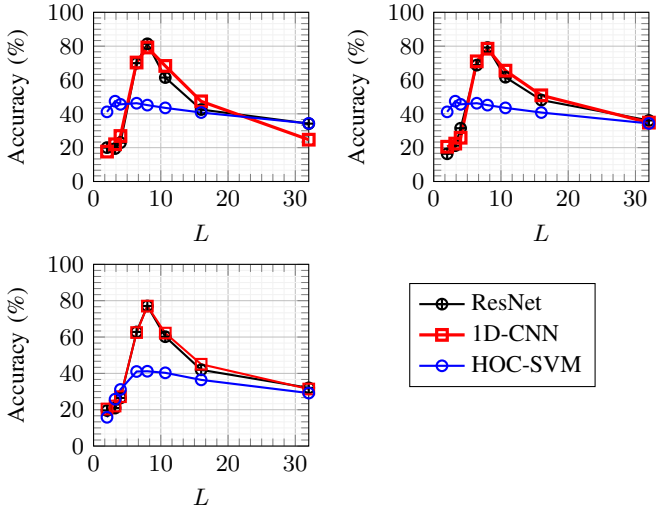


Figure 2: Sensitivity to upsampling factor L in: AWGN (top-left), Rayleigh+AWGN (top-right), and Rician+AWGN (bottom) (trained for $f_s = 200$ kHz, $\alpha = 0.35$, $L = 8$).

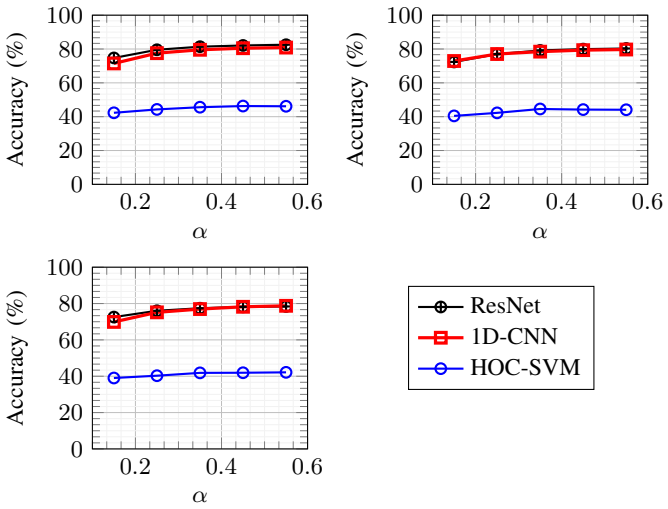


Figure 3: Sensitivity to RC roll-off factor α in: AWGN (top-left), Rayeleigh + AWGN (top-right), and Rician+ AWGN (bottom) (trained for $f_s = 200$ kHz, $\alpha = 0.35$ and $L = 8$).

Sampling frequencies higher than the Nyquist are applied in practice to improve the performance of A/D and D/A converters. The choice of f_s higher than Nyquist's is not harmful to AMC performance in AWGN, as shown in Fig. 4 (top left). In contrast, unknown f_s highly deteriorates AMC performance in fading channels for all baselines. Fig. 4 (bottom) shows that f_s variations introduce a large accuracy drop of 68% for 1D-CNN and ResNet and 35% for HOC-SVM.

C. Sensitivity to channel models

The wireless channel is inherently dynamic, and thus, the characteristic of the same signal may significantly differ over time. Thus, it is crucial to understand how differences between the channel conditions in training and runtime classification affect the AMC performance. In this section, we set out to quantify these performance issues and determine which channel conditions are most conducive to

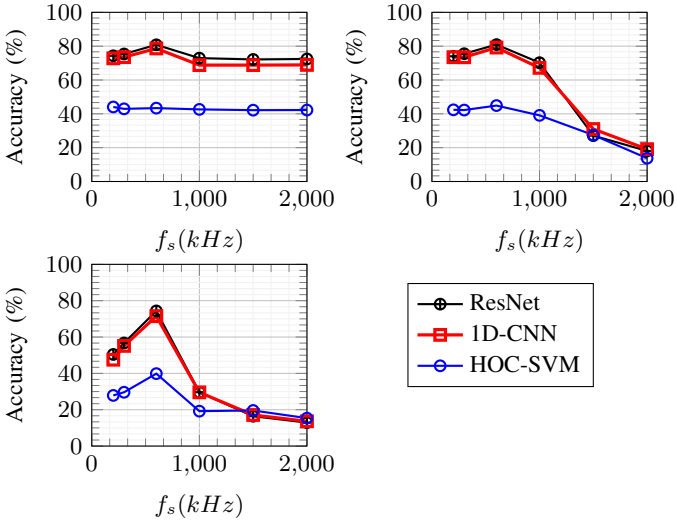


Figure 4: Sensitivity to sampling frequency f_s in: AWGN (top-left), Rayleigh+AWGN (top-right), and Rician+AWGN (bottom) (trained for $f_s = 600$ kHz, $\alpha = 0.35$ and $L = 8$).

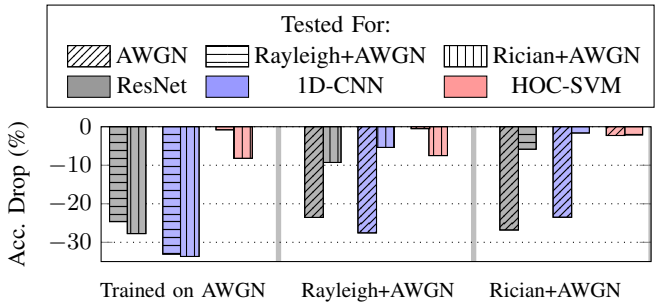


Figure 5: Classification accuracy drops due to the channel model variations.

universal training. To this end, we performed three experiments: (1) train for AWGN, test on Rayleigh+AWGN and Rician+AWGN; (2) train for Rayleigh+AWGN, test on AWGN and Rician+AWGN; (3) train for Rician+AWGN, test on AWGN and Rayleigh+AWGN. Training signal parameters $L = 8$, $\alpha = 0.35$ and $f_s = 200$ kHz were used for all runs.

Fig. 5 shows that all baselines have the highest accuracy drop (ResNet: 27.74%, 1D-CNN: 32.25%, HOC-SVM: 8.21%) when they are trained for AWGN. ResNet achieves the best results when trained for Rayleigh+AWGN, while 1D-CNN and HOC-SVM achieve a bit better results when trained for Rician+AWGN. Also, HOC-SVM is much less sensitive to channel model variations than 1D-CNN and ResNet. We further evaluate the effect of changing channel conditions while the channel model remains the same, but with a different profile. To this end, the model trained for DS_Rayleigh we tested on DS_Rayleigh_2. Even within the same channel model, the small changes in the channel profile introduce an accuracy drop of 6%, as shown in Fig. 6. Based on accuracy drops and overall average accuracy, training either for Rayleigh+AWGN or Rician+AWGN channel would be a conducive choice. Since Rayleigh has a lower number of hyper-parameters, it would be a preferred choice.

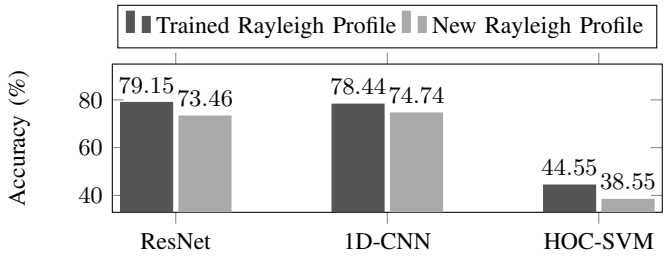


Figure 6: Sensitivity to different Rayleigh channels.

D. Learned representations and sensitivity analysis

The previous subsections show that DL AMC methods suffer from generalization issues with signal and channel parameter variations. In contrast, FB AMC methods are much less sensitive to these unknown parameters at the cost of lower classification accuracy. In what follows, we analyze the features learned by our baselines and their sensitivity to parameter selection in detail.

1) DNN models: Since all DNN AMC models have been treated as "black boxes", it is unclear how does DNN work with the input data, what features does it learn, and how does it arrive at the final prediction output. Gradient weighted Class Activation Map (Grad-CAM) [36] is an efficient representation of the importance of input features to CNN models. As both 1D-CNN and ResNet have similar performance, we pursue Grad-CAM analysis for the 1D-CNN model to understand how do the input properties affect the signal shape, and how are features learned. The Grad-CAM output is a heatmap, which captures the importance of each input data point for a given class. This heatmap for I/Q samples of one instance is shown in Table II. The color of each point represents the importance of that point for the model. A line is drawn between two adjacent I/Q points in time, if both have weights higher than 0.8, to capture the model's transition learning capabilities.

Table II presents the heatmaps for two simple modulations, BPSK and QPSK, at $SNR = 18$ dB. Correct predictions are shaded in green, while wrong predictions are shaded in red. From the cells in green, we note that the 1D-CNN learns spatial information of constellation points giving high importance to constellation point transitions. Upsampling and channel fading lead to profound changes in the constellation diagrams and result in miss-classification. An upsampling factor L introduces $L-1$ constellation points between two original constellation points; thus, a higher L results in higher dispersion of the constellation diagrams. Rayleigh fading introduces constellation rotation, while Rician fading adds more constellations in one diagram depending on the number of Line-Of-Sight (LOS) direct discrete paths (e.g., the last column in Table II shows two over-imposed constellations in both modulation classes). These results illustrate that the signal shape critically depends on the input conditions and explain the poor transferability of learned models to unseen scenarios.

2) HOC-based FB models: Although we expected that HOC's discriminative power would persist across signal and channel models, the above-presented analysis shows the opposite. In what follows, we quantify the sensitivity of HOC.

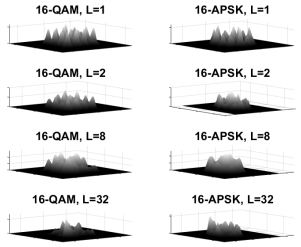


Figure 7: Effects of L on constellation shape for 16-QAM and 16-APSK in AWGN (SNR=18 dB).

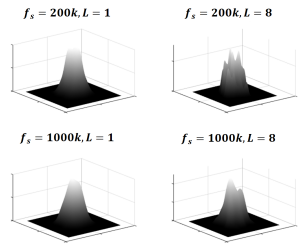
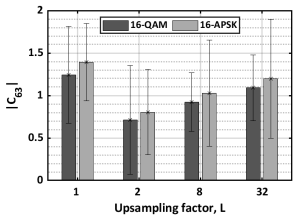
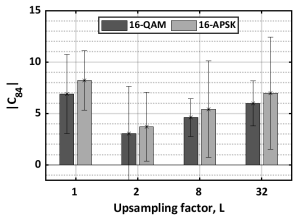


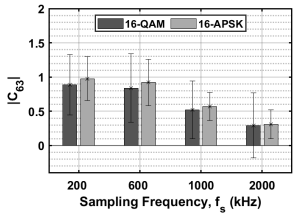
Figure 8: Effects of L and f_s on constellation shape for 16-QAM in Rician+AWGN (SNR=18 dB).



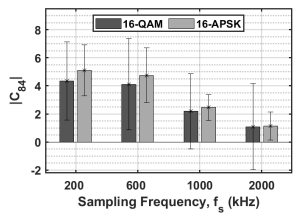
(a) C_{63} (Sensitivity to L)



(b) C_{84} (Sensitivity to L)



(c) C_{63} (Sensitivity to f_s)



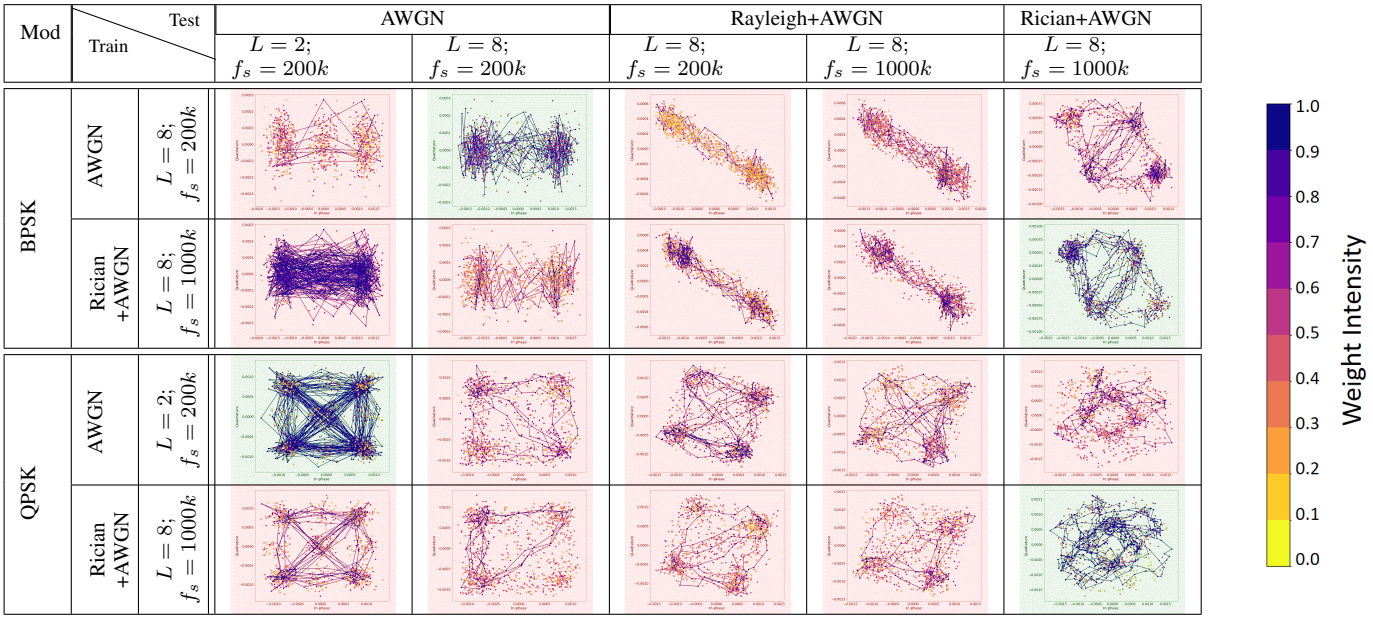
(d) C_{84} (Sensitivity to f_s)

Figure 9: Sensitivity of mean and standard deviation of C_{63} and C_{84} to L in AWGN (top) and f_s in Rician+AWGN (bottom). (All averaged across SNR range $[0, 18]$, for each SNR value is generated 100 instances).

Upsampling factor sensitivity. With $L > 1$, the shape of the modulation is distorted, and this distortion is larger for higher L. Fig. 7 illustrates the effect of upsampling for 16-QAM and 16-APSK in AWGN at $SNR = 18$ dB. Without upsampling, the constellations for both 16-QAM and 16-APSK are ideal with clear sample clusters. With upsampling, the clusters become distorted making it hard to recover the original constellation. Consequently, the HOC values are unstable, which increases with L, as shown in Figs. 9(a) and 9(b). The discriminative power of C_{63} and C_{84} is maintained for each L, however, their values change with L, leading to classifier confusion when applied to datasets with unknown L.

Sampling frequency sensitivity. The sampling frequency affects the classification performance only in fading channels. Rician and Rayleigh channels are modeled as slow flat for $f_s = 200$ kHz, while for higher f_s, these models act as a slow frequency selective fading channel. In flat fading, the channel's multipath structure is such that the transmitted signal's spectral characteristics are preserved at the receiver. In contrast, the frequency selective channel introduces Inter-Symbol Interference (ISI) since different frequency components of the signal are affected independently. As time varies, the channel varies in gain and phase across the signal spectrum, and it is highly unlikely that all parts of the signal will be simultaneously affected by a deep fade [33]. Consequently, the discriminative

Table II: Grad-CAM visualisation of the importance of input features to 1D-CNN trained for different values of signal and channel parameters at $SNR = 18$ dB. Correct classification predictions are shaded in green, while wrong ones are shaded in red. Higher weight intensity means higher importance.



power of the cumulants features decreases with increasing f_s, as shown in Figs. 9(c) and 9(d) for C_{63} and C_{84} in two representative modulations. The higher f_s, the deep fade takes a more severe impact on the signal.

Channel model sensitivity. Although the constellation shapes are lost in fading channels, the HOC values do not change so much in flat fading channels. E.g., in Fig. 8 a 16-QAM constellation appears with a single symbol cluster as opposed to the expected 16 clusters. If we compare C_{63} values for 16-QAM and 16-APSK in AWGN (see the third bar in Fig. 9(a)) and in flat Rician+AWGN (see the first bar in Fig. 9(c)), we observe a small decrease of C_{63} for both, 16-QAM and 16-APSK. The same trend is observed for C_{84} (compare the third bar in Fig. 9(b) and the first bar in Fig. 9(d)). HOC discriminative power is lost in frequency selective fading channels, resulting in poor performance regardless of the adopted classifier.

VI. Using STN and TL to combat sensitivity to unknown signal and channel parameters

In the previous section, we showed that CNNs learn spatial signal information, i.e., constellation shapes. Upsampling, noise, and fading distort the constellation shapes, either by rotation, translation, scaling, or some non-linear transformation, as shown in Table II. Recently, the use of DNNs to learn spatial transformations has become an active research field in DL [37]. STN is introduced in [37] and designed as an independent module that can be easily embedded into any classifier network. There have been a few approaches to embed STN in the AMC classifier [38]–[40]. However, performance evaluation is done on simple datasets (e.g., 3, 8, and 11 in [39], [40], and [38], respectively) with a priori known L, RC α, and channel models. Prior work [38,40] reports marginal improvements in using STN to synchronize the frequency and sample rate offset between a sensor and a transmitter. Our work examines the applicability of STN to combat AMC sensitivity to unknown signal and channel parameters and for a large dataset of 20 complex modulations. We omit receiver impairments such as frequency offset and sample rate offset. Below we present the results achieved by two different approaches: (1) data augmentation with STN and (2) TL with an assumption that small labeled datasets are available for certain unseen scenarios.

A. A novel network model with STN and ResNeXt

Inspired by the advantages of ResNeXt over ResNet [20], we propose a new classifier network in which we add the STN module. Our model, named STN-ResNeXt, is given in Fig. 10. The instances of size 1024 are fed into the model. The model has two parallel branches: (1) ClassNet learns features from the original input; (2) STN+ClassNet learns features from the transformed input. Both learned features are concatenated and sent to the Dense layer of 128 units, followed by the Softmax layer. STN consists of three modules: (1) the Localizer predicts transformation matrix; (2) the Grid Generator implements the transformation; (3) the Sampler implements interpolation. The Localizer network contains two CNN layers followed by three ResNeXt blocks (shown in Fig. 11) and a Global Average Pooling layer. The last layer in the Localizer is a dense layer with 6 units, whose weights are initialized as $[0.7, -0.7, 0.1, 0.3, 0.7, 0.2]$. ClassNet consists of two CNN layers, six ResNeXt blocks, and a Global Average Pooling Layer. Note that ClassNet and the Localizer hyperparameters (number of ResNeXt blocks, block structure, etc.) are optimized through trial-and-error, and found optimal values are given in Figs. 10 and 11. The entire STN-ResNeXt network has 86,212 trainable parameters, which are 2.74 times lower than the number of trainable parameters in ResNet (236,344), and 1.65 times lower than 1D-CNN (142,932).

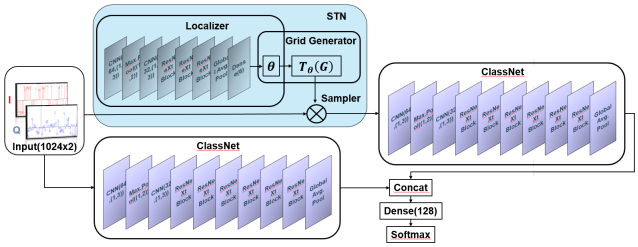


Figure 10: STN-ResNeXt network layout.

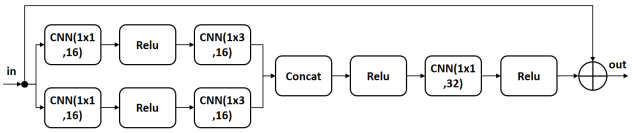


Figure 11: ResNeXt block layout.

B. Performance analysis with STN

We seek to evaluate the benefits of STN in unseen scenarios. We trained STN-ResNeXt for a Rayleigh channel, as it is most conducive to universal training. The remaining training parameters are $L = 8$, $\alpha = 0.35$, and $f_s = 200$ kHz. First, we evaluated the performance when only the ClassNet branch is active, while the STN+ClassNet branch is disabled. The results show that the ClassNet branch alone achieves the same results as ResNet [38]. Second, by enabling the STN+ClassNet branch, we also achieved an average accuracy improvement of up to 6%. However, the sensitivity to unknown signal parameters persists similarly to our already evaluated AMC methods: ResNet and 1D-CNN. Replacing the ClassNet block with the ResNet model [13] only gave an expected slight improvement of 3%, which is in line with the conventional reported improvements of less than 6% [37].

C. Performance analysis with TL

Since labeled data across all scenarios is infeasible to obtain, we propose to use TL to transfer learned features from existing source tasks to improve and expedite learning in related tasks with a limited set of labeled training data [41]. TL has been applied to AMC, where the target domain is a new modulation set [42], unknown receiver impairments (frequency offset or time drift), or different instance sizes [13,42,43]. Those approaches either aim to improve the accuracy only for the target domain ([42]) or to improve the overall accuracy by creating individual models for each domain [43]. While the former might result in very poor performance for the source domain, the latter suffers from high computational complexity. We aim for generalizability by having only one model that achieves the best average accuracy for each unseen scenario with a small labeled data set. In this analysis, we use a baseline the STN-ResNeXt model trained for Rayleigh+AWGN channel, $L = 8$, $f_s = 200$ kHz, and $\alpha = 0.35$.

1) STN-ResNeXt with varying channel models: The baseline STN-ResNeXt model achieves an average classification accuracy of 80.4% in Rayleigh+AWGN, while in unseen channel conditions, it achieves 53.11% for AWGN and 72.34% in Rician+AWGN. We improve the accuracy by retraining only a portion of the baseline using TL with $5, 8$ or 10% labeled data. Specifically, we explore two scenarios: (1) retrain only the last Dense layer; and (2) retrain only the last ResNeXt block in STN. Fig. 12 shows that the transfer of a limited amount of information substantially boosts the performance over the baseline in both modified scenarios. Increasing the percentage of limited training, from 5% to 10%, does not lead to substantial improvement. The best results are achieved when only the last Dense layer is retrained with 10% labeled data. The average accuracy is increased by 20% in AWGN and 3% in Rician+AWGN. Fig. 12 shows that lower layers are indeed in charge of more general features, while higher layers are more sensitive to domain differences as elaborated in [44]. There is a slight decrease in classification accuracy of up to 2% for the source domain, Rayleigh+AWGN, due to the network layers' adaptation with the other two unknown channel models; however, this is justified by the significant performance boost in the other two unknown channel models.

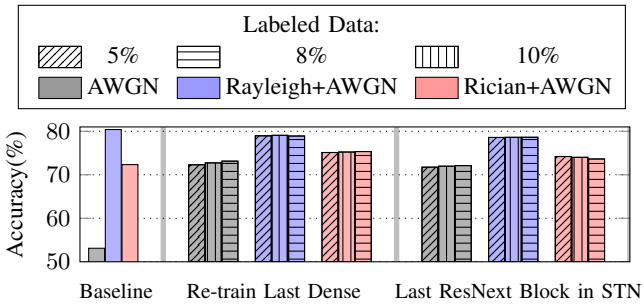


Figure 12: Classification accuracy across channel models with TL.

2) STN-ResNeXt with varying sampling frequencies: As a baseline model, we use STN-ResNeXt optimized for channel variations. As the sampling frequency is a problem only in fading channels, we need a small set of labeled data with frequency selective channel conditions. We selected a small set from DS_AWGN, DS_Rayleigh, and DS_Rician for $f_s = 1500$ kHz, $L = 8$, and $\alpha = 0.35$. We retrain the last Dense layer with 5% and 10% of labeled data. Fig. 13 shows that by retraining only the last Dense layer with 5% of labeled data, the accuracy for $f_s = 1500$ kHz is increased by 26% in fading channels. The accuracies for $f_s = [1000, 2000]$ kHz see a smaller increase of 10%, and overall they are low even with TL. Thus, we conclude that the sampling frequency difference of 500 kHz is too large to be generalized by DNNs, and small sets of labeled data for both 1000 kHz and 2000 kHz are required to improve their performance. Further investigation is necessary into the minimum sampling frequency offset that DNN can sustain.

3) STN-ResNeXt with varying upsampling factors: Finally, we seek to understand STN-ResNeXt's performance across upsampling factors. As a baseline, we use STN-ResNeXt optimized for channel variations. First, we study the performance of STN-ResNeXt with a fixed upsampling factor $L = 8$ and channel-aware training (i.e., the training uses labeled data for each channel model: 80% of DS_AWGN, 80% of DS_Rayleigh, and 80% of DS_Rician). Fig. 14 (left) illustrates the sensitivity of such trained model to unknown

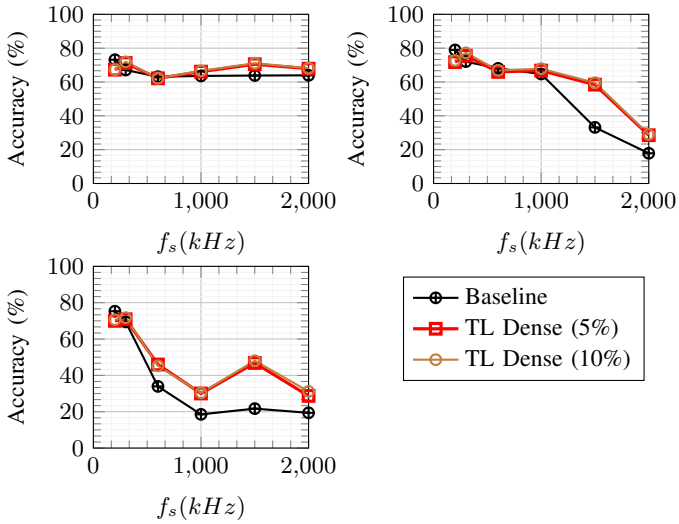


Figure 13: STN-ResNeXt sensitivity to f_s in AWGN (top left), Rayleigh+AWGN (top right), and Rician+AWGN (bottom).

Ls. The maximum accuracy is around 80% for known $L = 8$ in each channel model, while it significantly drops when L is different than 8. We explore TL's performance benefits using a limited set of training data (5%) with $L = [2, 4, 8, 16, 32]$ for each channel. We retrained the last Dense layer in STN-ResNeXt with this data. Fig. 14 (right) shows that the accuracy is increased by 20% for $L = 2$ and by 50% for $L = 32$ in each channel model.

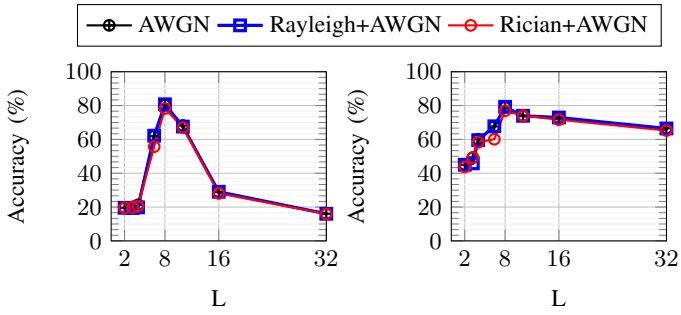


Figure 14: Channel-aware STN-ResNeXt across L. Without TL (left), with TL (right).

Finally, we seek to understand the performance of STN-ResNeXt without channel-aware training and without prior knowledge of Ls (i.e., using our model optimized for channel variations). To this end, we test STN-ResNeXt, optimized for channel variations and $L = 8$, across all three channel conditions (AWGN, Rayleigh+AWGN, and Rician+AWGN) and unknown Ls. Fig. 15 (left) presents accuracy across L without TL. Compared to Fig. 14 (left), we see a 10% increase in accuracy for unknown $L > 8$ in each channel model. Note that TL adjusted STN-ResNeXt was optimized for channel variations with only 10% labeled data of each channel model with $L = 8$. In comparison, channel-aware STN-ResNeXt was trained with 80% labeled data for each channel with $L = 8$. The improvement by 10% is due to the included STN module in our model. To be more restricted with the amount of labeled data, we retrained the last Dense layer with 5% labeled

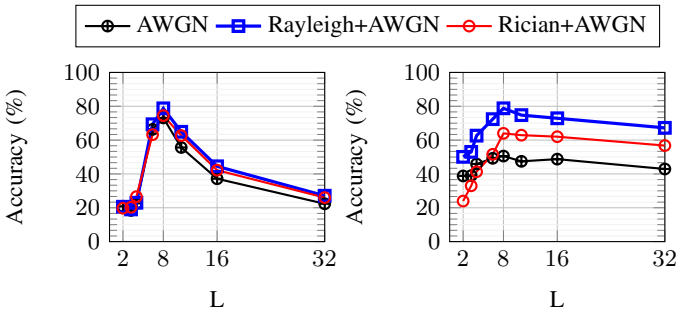


Figure 15: STN-ResNeXt across L. Baseline (left), baseline with TL (right).

data for $L = [2, 4, 8, 16, 32]$, but only for Rayleigh channels. The results after TL (Fig. 15 (right)) show that the accuracy for the Rayleigh+AWGN channel is better than in the case with channel-aware STN-ResNeXt, while for the other two channels, the accuracy is lower. However, compared with the baseline without TL (Fig. 15 (left)), the accuracy increases by 30% and 20% for $L = 32$ in Rician+AWGN and AWGN, respectively. Average accuracy increases across all Ls with TL are 23.07%, 7.5%, and 5.76% for the Rayleigh+AWGN, Rician+AWGN, and AWGN, respectively. These results lead to two important insights: (i) channel-oblivious training suffers a non-negligible performance deterioration; however, (ii) employing transfer learning with limited training substantially improves the modulation classification.

VII. Conclusions

AMC in realistic scenarios is a challenging problem, as knowledge of signal and channel parameters is required for optimal performance. In this paper, we examined how the lack of this knowledge affects the performance of two AMC research streams: data-driven and expert feature-based. We showed that unknown upsampling factors significantly deteriorate classification accuracy in each channel model for both research AMC streams, while unknown RC filter parameters do not harm accuracy. Unknown sampling frequency introduces a substantial accuracy drop in fading channels. Feature-based AMC methods are most sensitive to unknown upsampling since it introduces the most significant deviations in signal shape and, in turn, in features' values. On the other hand, data-driven AMC methods outperform expert feature-based methods in scenarios for which they are trained. As each change of signal and channel parameters impacts the data distribution, data-driven AMC methods that rely on trained data distribution knowledge fail to correctly classify data from different distributions. STN is an active research direction in DL, which aims to improve the robustness of DNNs to distorted data distributions. We showed that the current implementation of STN improves performance by less than 6%. Given a limited labeled data set for a new target domain, we showed that transfer learning improves performance by up to 30% with only 5% of labeled data while retraining just one layer in the overall DNN architecture. However, even with optimal STN and TL, DNNs are still vulnerable to out-of-distribution data, which calls for further investigation into anomaly detection to improve the certainty in DNN's output.

REFERENCES

[1] A. Chakraborty, M. S. Rahman, *et al.*, "Specsense: Crowdsensing for efficient querying of spectrum occupancy," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pp. 1–9, IEEE, 2017.

[2] M. Khaledi, M. Khaledi, *et al.*, "Simultaneous power-based localization of transmitters for crowdsourced spectrum monitoring," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pp. 235–247, 2017.

[3] S. Rajendran, V. Lenders, *et al.*, "Crowdsourced wireless spectrum anomaly detection," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 2, pp. 694–703, 2020.

[4] S. Rajendran, R. Calvo-Palomino, *et al.*, "Electrosense: Open and big spectrum data," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 210–217, 2018.

[5] "Universal Software Radio Peripheral (USRP)." https://www.ettus.com/.

[6] "RTL-SDR." http://www.rtl-sdr.com/.

[7] W. Xiong, P. Bogdanov, and M. Zheleva, "Robust and efficient modulation recognition based on local sequential iq features," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pp. 1612–1620, 2019.

[8] J. Ma and T. Qiu, "Automatic modulation classification using cyclic correntropy spectrum in impulsive noise," *IEEE Wireless Communications Letters*, vol. 8, no. 2, pp. 440–443, 2019.

[9] X. Zhang, J. Sun, and X. Zhang, "Automatic modulation classification based on novel feature extraction algorithms," *IEEE Access*, vol. 8, pp. 16362–16371, 2020.

[10] Y. Wang, M. Liu, *et al.*, "Data-Driven Deep Learning for Automatic Modulation Recognition in Cognitive Radios," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 4074–4077, 2019.

[11] T. J. O'Shea and J. Corgan, "Convolutional radio modulation recognition network," *International Conference on Engineering Applications of Neural Networks*, pp. 213–226, 2016.

[12] S. Rajendran, W. Meert, *et al.*, "Deep Learning Models for Wireless Signal Classification with Distributed Low-Cost Spectrum Sensors," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 3, pp. 433–445, 2018.

[13] T. 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, 2018.

[14] E. Perenda, S. Rajendran, and S. Pollin, "Automatic modulation classification using parallel fusion of convolutional neural networks," in *2019 3rd International Balkan Conference on Communications and Networking. Skopje, North Macedonia*, 2019.

[15] B. Kim, J. Kim, *et al.*, "Deep neural network-based automatic modulation classification technique," in *2016 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 579–582, 2016.

[16] B. Luo, Q. Peng, *et al.*, "Robustness of deep modulation recognition under awgn and rician fading," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, pp. 447–450, 2018.

[17] R. Maoudj, M. Terre, and I. Ahriz, "Constant modulus algorithm based on fourth order moments initialization," in *2015 23rd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pp. 351–355, 2015.

[18] Liang Hong, "Maximum likelihood bpsk and qpsk classifier in fading environment using the em algorithm," in *2006 Proceeding of the Thirty-Eighth Southeastern Symposium on System Theory*, pp. 313–317, 2006.

[19] W. C. Headley, V. G. Chavali, and C. R. C. M. da Silva, "Maximum-likelihood modulation classification with incomplete channel information," in *2013 Information Theory and Applications Workshop (ITA)*, pp. 1–4, 2013.

[20] S. Xie, R. Girshick, *et al.*, "Aggregated residual transformations for deep neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5987–5995, 2017.

[21] J. L. Xu, W. Su, and M. Zhou, "Likelihood-Ratio Approaches to Automatic Modulation Classification," *IEEE Trans. Systems, Man, and Cybernetics Part C: Applications and Reviews*, vol. 41, pp. 455–469, July 2011.

[22] F. Hameed, O. A. Dobre, and D. C. Popescu, "On the likelihood-based approach to modulation classification," *IEEE Transactions on Wireless Communications*, vol. 8, no. 12, pp. 5884–5892, 2009.

[23] S. Majhi, R. Gupta, *et al.*, "Hierarchical Hypothesis and Feature-Based Blind Modulation Classification for Linearly Modulated Signal," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, 2017.

[24] J. Li, Q. Meng, *et al.*, "Automatic Modulation Classification Using Support Vector Machines and Error Correcting Output Codes," *2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference*, 2017.

[25] Y. Zhang, J. Wang, *et al.*, "Wireless signal classification based on high-order cumulants and machine learning," in *2018 IEEE International Conference of Safety Produce Informatization (IICSPI)*, pp. 246–250, 2018.

[26] F. Yang, B. Hao, *et al.*, "A method of high-precision signal recognition based on higher-order cumulants and svm," in *2018 5th International Conference on Systems and Informatics (ICSAI)*, pp. 455–459, 2018.

[27] K. He, X. Zhang, *et al.*, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[28] S. Chen, Y. Zhang, *et al.*, "A novel attention cooperative framework for automatic modulation recognition," *IEEE Access*, vol. 8, pp. 15673–15686, 2020.

[29] H. Zhang, Y. Wang, *et al.*, "Automatic modulation classification using a deep multi-stream neural network," *IEEE Access*, vol. 8, pp. 43888–43897, 2020.

[30] B. Tang, Y. Tu, *et al.*, "Digital signal modulation classification with data augmentation using generative adversarial nets in cognitive radio networks," *IEEE Access*, vol. 6, pp. 15713–15722, 2018.

[31] S. Rajendran and S. Pollin, "Large scale wireless spectrum monitoring: Challenges and solutions based on machine learning," in *Spectrum Sharing: The Next Frontier in Wireless Networks* (D. T. M. S. Tharmalingam Ratnarajah, Constantinos B. Papadias, ed.), ch. 16, Wiley-Blackwell, 2020.

[32] M. Patel, X. Wang, and S. Mao, "Data augmentation with conditional gan for automatic modulation classification," *Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning*, 2020.

[33] T. Rappaport, *Wireless Communications: Principles and Practice*. USA: Prentice Hall PTR, 2nd ed., 2001.

[34] M. Abadi, A. Agarwal, *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[35] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," *San Diego: The International Conference on Learning Representations (ICLR)*, vol. 1, no. 1, 2015.

[36] R. R. Selvaraju, M. Cogswell, *et al.*, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017.

[37] M. Jaderberg, K. Simonyan, *et al.*, "Spatial transformer networks," in *Advances in Neural Information Processing Systems 28* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), pp. 2017–2025, Curran Associates, Inc., 2015.

[38] T. J. O'Shea, L. Pemula, *et al.*, "Radio transformer networks: Attention models for learning to synchronize in wireless systems," in *2016 50th Asilomar Conference on Signals, Systems and Computers*, pp. 662–666, 2016.

[39] M. Mirmohammadsadeghi, S. S. Hanna, and D. Cabric, "Modulation classification using convolutional neural networks and spatial transformer networks," in *2017 51st Asilomar Conference on Signals, Systems, and Computers*, pp. 936–939, 2017.

[40] M. Li, O. Li, *et al.*, "An automatic modulation recognition method with low parameter estimation dependence based on spatial transformer networks," *Applied Sciences*, vol. 9, p. 1010, Mar 2019.

[41] L. Torrey and J. Shavlik, "Transfer learning," in *IGI Global, edited by E. Soria, J. Martin, R. Magdalena, M. Martinez and A. Serrano*, 2009.

[42] F. Meng, P. Chen, *et al.*, "Automatic modulation classification: A deep learning enabled approach," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10760–10772, 2018.

[43] K. Bu, Y. He, *et al.*, "Adversarial transfer learning for deep learning based automatic modulation classification," *IEEE Signal Processing Letters*, vol. 27, pp. 880–884, 2020.

[44] B. Neyshabur, H. Sedghi, and C. Zhang, "What is being transferred in transfer learning?," *ArXiv*, vol. abs/2008.11687, 2020.