Over the Air Deep Learning Based Radio Signal Classification

Article *in* IEEE Journal of Selected Topics in Signal Processing · December 2017

DOI: 10.1109/JSTSP.2018.2797022

3 authors:

Tim O'Shea
Virginia Tech (Virginia Polytechnic Institute and State University)
37 PUBLICATIONS **3,858** CITATIONS
SEE PROFILE

Tamoghna Roy
DeepSig Inc
17 PUBLICATIONS **1,220** CITATIONS
SEE PROFILE

T. Charles Clancy
Virginia Tech (Virginia Polytechnic Institute and State University)
162 PUBLICATIONS **7,717** CITATIONS
SEE PROFILE

Over the Air Deep Learning Based Radio Signal Classification

Tim O'Shea, *Senior Member, IEEE,* Tamoghna Roy, *Member, IEEE*
and T. Charles Clancy, *Senior Member, IEEE*

***Abstract*—We conduct an in depth study on the performance of deep learning based radio signal classification for radio communications signals. We consider a rigorous baseline method using higher order moments and strong boosted gradient tree classification and compare performance between the two approaches across a range of configurations and channel impairments. We consider the effects of carrier frequency offset, symbol rate, and multi-path fading in simulation and conduct over-the-air measurement of radio classification performance in the lab using software radios and compare performance and training strategies for both. Finally we conclude with a discussion of remaining problems, and design considerations for using such techniques.**

I. Introduction

Rapidly understanding and labeling of the radio spectrum in an autonomous way is a key enabler for spectrum interference monitoring, radio fault detection, dynamic spectrum access, opportunistic mesh networking, and numerous regulatory and defense applications. Boiling down a complex high-data rate flood of RF information to precise and accurate labels which can be acted on and conveyed compactly is a critical component today in numerous radio sensing and communications systems. For many years, radio signal classification and modulation recognition have been accomplished by carefully handcrafting specialized feature extractors for specific signal types and properties and by and deriving compact decision bounds from them using either analytically derived decision boundaries or statistical learned boundaries within low-dimensional feature spaces.

In the past five years, we have seen rapid disruption occurring based on the improved neural network architectures, algorithms and optimization techniques collectively known as deep learning (DL) [26]. DL has recently replaced the machine learning (ML) state of the art in computer vision, voice and natural language processing; in both of these fields, feature engineering and pre-processing were once critically important topics, allowing cleverly designed feature extractors and transforms to extract pertinent information into a manageable reduced dimension representation from which labels or decisions could be readily learned with tools like support vector machines or decision trees. Among these widely used front-end features were the scale-invariant feature transform (SIFT) [9], the bag of words [8], Mel-frequency Cepstral coefficients (MFCC) [1] and others which were widely relied upon only a few years ago, but are no longer needed for state of the art performance today.

DL greatly increased the capacity for feature learning directly on raw high dimensional input data based on high level supervised objectives due to the new found capacity for learning of very large neural network models with high numbers of free parameters. This was made possible by the combination of strong regularization techniques [18], [21], greatly improved methods for stochastic gradient descent (SGD) [15], [16], low cost high performance graphics card processing power, and combining of key neural network architecture innovations such as convolutional neural networks [5], and rectified linear units [13]. It was not until Alexnet [14] that many of these techniques were used together to realize an increase of several orders of magnitude in the practical model size, parameter count, and target dataset and task complexity which made feature learning directly from imagery state of the art. At this point, the trend in ML has been relentless towards the replacement of rigid simplified analytic features and models with approximate models with much more accurate high degrees of freedom (DOF) models derived from data using end-to-end feature learning. This trend has been demonstrated in vision, text processing, and voice, but has yet to be widely applied or fully realized on radio time series data sets until recently.

We showed in [30], [32] that these methods can be readily applied to simulated radio time series sample data in order to classify emitter types with excellent performance, obtaining equivalent accuracies several times more sensitive than existing best practice methods using feature based classifiers on higher order moments. In this work we provide a more extensive dataset of additional radio signal types, a more realistic simulation of the wireless propagation environment, over the air measurement of the new dataset (i.e. real propagation effects), new methods for signal classification which drastically outperform those we initially introduced, and an in depth analysis of many practical engineering design and system parameters impacting the performance and accuracy of the radio signal classifier.

II. Background

A. Baseline Classification Approach

1) Statistical Modulation Features: For digital modulation techniques, higher order statistics and cyclo-stationary moments [2], [3], [10], [23], [33] are among the most widely used features to compactly sense and detect signals with strong periodic components such as are created by the structure of the

Authors are with the Bradley Department of Electrical and Computer Engineering, Virginia Tech and DeepSig, Arlington, VA e-mail: (oshea,tamoghna,tcc)@vt.edu.

carrier, symbol timing, and symbol structure for certain modulations. By incorporating precise knowledge of this structure, expected values of peaks in auto-correlation function (ACF) and spectral correlation function (SCF) surfaces have been used successfully to provide robust classification for signals with unknown or purely random data. For analog modulation where symbol timing does not produce these artifacts, other statistical features are useful in performing signal classification.

For our baseline features in this work, we leverage a number of compact higher order statistics (HOSs). To obtain these we compute the higher order moments (HOMs) using the expression given below:

$$M(p,q) = E[x^{p-q}(x^*)^q] \tag{1}$$

From these HOMs we can derive a number of higher order cumulantss (HOCs) which have been shown to be effective discriminators for many modulation types [23]. HOCs can be computed combinatorially using HOMs, each expression varying slightly; below we show one example such expression for the $C(4,0)$ HOM.

$$C(4,0) = \sqrt{M(4,0) - 3 \times M\left(2,0\right)^2} \tag{2}$$

Additionally we consider a number of analog features which capture other statistical behaviors which can be useful, these include mean, standard deviation and kurtosis of the normalized centered amplitude, the centered phase, instantaneous frequency, absolute normalized instantaneous frequency, and several others which have shown to be useful in prior work. [6].

2) Decision Criterion: When mapping our baseline features to a class label, a number of compact machine learning or analytic decision processes can be used. Probabilistically derived decision trees on expert modulation features were among the first to be used in this field, but for many years such decision processes have also been trained directly on datasets represented in their feature space. Popular methods here include support vector machines (SVMs), decision trees (DTrees), neural networks (NNs) and ensembling methods which combine collections of classifiers to improve performance. Among these ensembling methods are Boosting, Bagging [4], and Gradient tree boosting [7]. In particular, XGBoost [24] has proven to be an extremely effective implementation of gradient tree boosting which has been used successfully by winners of numerous Kaggle data science competitions [12]. In this work we opt to use the XGBoost approach for our feature classifier as it outperforms any single decision tree, SVM, or other method evaluated consistently as was the case in [32].

B. Radio Channel Models

When modeling a wireless channel there are many compact stochastic models for propagation effects which can be used [11]. Primary impairments seen in any wireless channel include:

- carrier frequency offset (CFO): carrier phase and frequency offset due to disparate local oscillators (LOs) and motion (Doppler).
- symbol rate offset (SRO): symbol clock offset and time dilation due to disparate clock sources and motion.
- Delay Spread: non-impulsive delay spread due to delayed reflection, diffraction and diffusion of emissions on multiple paths.
- Thermal Noise: additive white-noise impairment at the receiver due to physical device sensitivity.

Each of these effects can be compactly modeled well and is present in some form on any wireless propagation medium. There are numerous additional propagation effects which can also be modeled synthetically beyond the scope of our exploration here.

C. Deep Learning Classification Approach

DL relies today on SGD to optimize large parametric neural network models. Since Alexnet [14] and the techniques described in section I, there have been numerous architectural advances within computer vision leading to significant performance improvements. However, the core approach remains largely unchanged. Neural networks are comprised of a series of layers which map each layer input h_0 to output h_1 using parametric dense matrix operations followed by non-linearities. This can be expressed simply as follows, where weights, W, have the dimension $|h_0 \times h_1|$, bias, b, has the dimension $|h_1|$ (both constituting θ), and max is applied element-wise per-output $|h_1|$ (applying rectified linear unit (ReLU) activation functions).

$$h_1 = max(0, h_0 W + b) \tag{3}$$

Convolutional layers can be formed by assigning a shape to inputs and outputs and forming W from the replication of filter tap variables at regular strides across the input (to reduce parameter count and enforce translation invariance).

Training typically leverages a loss function ($\mathscr{L}$), in this case (for supervised classification) categorical cross-entropy, between one-hot known class labels y_i (a zero vector, with a one value at the class index i of the correct class) and predicted class values $\hat{y_i}$.

$$\mathscr{L}(y,\hat{y}) = \frac{-1}{N}\sum_{i=0}^{N}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right] \tag{4}$$

Back propagation of loss gradients can be used to iteratively update network weights (θ) for each epoch n within the network ($f(x,\theta)$) until validation loss is no longer decreasing. We use the Adam optimizer [16], whose form roughly follows the conventional SGD expression below, except for a more complex time varying expression for learning rate (η) beyond the scope of this work.

$$\theta_{n+1} = \theta_n - \eta\frac{\partial \mathscr{L}(y, f(x,\theta_n))}{\partial \theta_n} \tag{5}$$

TABLE I. RANDOM VARIABLE INITIALIZATION

Random Variable	Distribution
α	$U(0.1, 0.4)$
Δ_t	$U(0, 16)$
Δf_s	$N(0, \sigma_{clk})$
θ_c	$U(0, 2\pi)$
Δf_c	$N(0, \sigma_{clk})$
H	$\Sigma_i \delta(t - \text{Rayleigh}_i(\tau))$

To reduce over fitting to training data, regularization is used. We use batch normalization [21] for regularization of convolutional layers and Alpha Dropout [31] for regularization of fully connected layers. Detail descriptions of additional layers used including SoftMax, Max-Pooling, etc are beyond the scope of this work and are described fully in [26].

III. DATASET GENERATION APPROACH

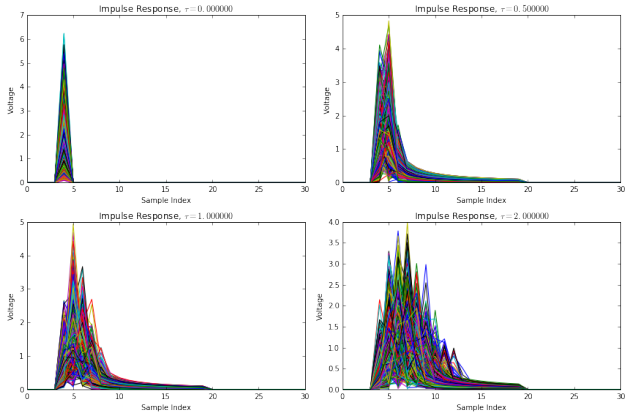

Fig. 1. Fading Power Delay Profile Examples

We generate new datasets for this investigation by building upon an improved version of the tools described in [29]. 24 different analog and digital modulators are used which cover a wide range of single carrier modulation schemes. We consider several different propagation scenarios in the context of this work, first are several simulated wireless channels generated from the model shown in figure 2, and second we consider over the air (OTA) transmission channel of clean signals as shown in figures 3 and 4 with no synthetic channel impairments. Digital signals are shaped with a root-raised cosine pulse shaping filter [36] with a range of roll-off values (α).

For each example in the synthetic data sets, we independently draw a random value for each of the variables shown below in table I. This results in a new and uncorrelated random channel initialization for each example.

Figure 1 illustrates several random values for H, the channel impulse response envelope, for different delay spreads, $\tau = [0, 0.5, 1.0, 2.0]$, relating to different levels of multi-path fading in increasingly more difficult Rayleigh fading environments. Figure 22 illustrate examples from the training set when using a simulated channel at low SNR (0 dB E_s/N_0).

We consider two different compositions of the dataset, first a "Normal" dataset, which consists of 11 classes which are all relatively low information density and are commonly seen in impaired environments. These 11 signals represent a relatively simple classification task at high SNR in most cases, somewhat comparable to the canonical MNIST digits. Second, we introduce a "Difficult" dataset, which contains all 24 modulations. These include a number of high order modulations (QAM256 and APSK256), which are used in the real world in very high-SNR low-fading channel environments such as on impulsive satellite links [25] (e.g. DVB-S2X). We however, apply impairments which are beyond that which you would expect to see in such a scenario and consider only relatively short-time observation windows for classification, where the number of samples (ℓ) is $= 1024$. Short time classification is a hard problem since decision processes can not wait and acquire more data to increase certainty. This is the case in many real world systems when dealing with short observations (such as when rapidly scanning a receiver) or short signal bursts in the environment. Under these effects, with low SNR examples (from -20 dB to +30 dB E_s/N_0), one would not expect to be able to achieve anywhere near 100% classification rates on the full dataset, making it a good benchmark for comparison and future research comparison.

The specific modulations considered within each of these two dataset types are as follows:

- Normal Classes: OOK, 4ASK, BPSK, QPSK, 8PSK, 16QAM, AM-SSB-SC, AM-DSB-SC, FM, GMSK, OQPSK
- Difficult Classes: OOK, 4ASK, 8ASK, BPSK, QPSK, 8PSK, 16PSK, 32PSK, 16APSK, 32APSK, 64APSK, 128APSK, 16QAM, 32QAM, 64QAM, 128QAM, 256QAM, AM-SSB-WC, AM-SSB-SC, AM-DSB-WC, AM-DSB-SC, FM, GMSK, OQPSK

The raw datasets will be made available on the RadioML website [1] shortly after publication.

A. Over the air data capture

In additional to simulating wireless channel impairments, we also implement an OTA test-bed in which we modulate and transmit signals using a universal software radio peripheral (USRP) [19] B210 software defined radio (SDR). We use a second B210 (with a separate free-running LO) to receive these transmissions in the lab, over a relatively benign indoor wireless channel on the 900MHz ISM band. These radios use the Analog Devices AD9361 [35] radio frequency integrated circuit (RFIC) as their radio front-end and have an LO that provides a frequency (and clock) stability of around 2 parts per million (PPM). We off-tune our signal by around 1 MHz to avoid DC signal impairment associated with direct conversion, but store signals at base-band (offset only by LO error). Received test emissions are stored off unmodified along with ground truth labels for the modulation from the emitter.

IV. SIGNAL CLASSIFICATION MODELS

In this section we explore the radio signal classification methods in more detail which we will use for the remainder of this paper.

[1] https://radioml.org

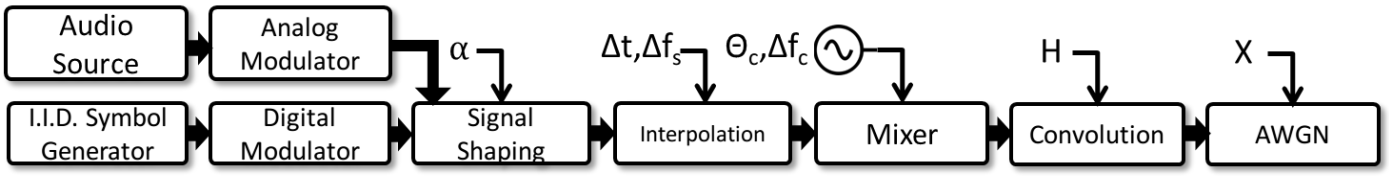

Fig. 2. System for dataset signal generation and synthetic channel impairment modeling

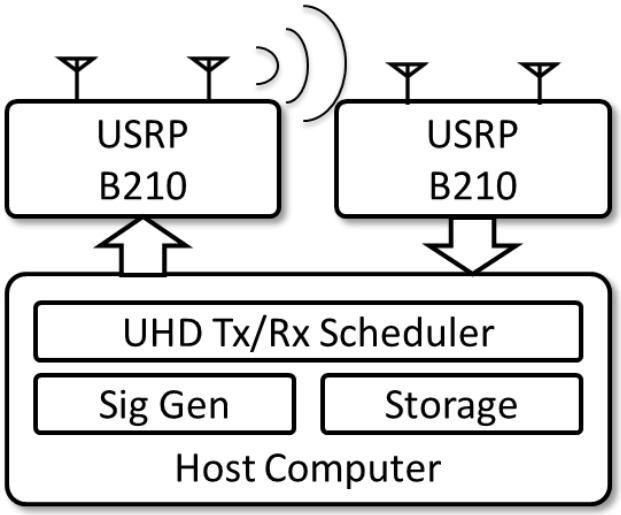

Fig. 3. Over the Air Test Configuration

TABLE II. FEATURES USED

Feature Name
M(2,0), M(2,1)
M(4,0), M(4,1), M(4,2), M(4,3)
M(6,0), M(6,1), M(6,2), M(6,3)
C(2,0), C(2,1)
C(4,0), C(4,1), C(4,2),
C(6,0), C(6,1), C(6,2), C(6,3)
Additional analog II-A

Fig. 4. Configuration for Over the Air Transmission of Signals

A. Baseline Method

Our baseline method leverages the list of higher order moments and other aggregate signal behavior statistics given in table II. Here we can compute each of these statistics over each 1024 sample example, and translate the example into feature space, a set of real values associated with each statistic for the example. This new representation has reduced the dimension of each example from $\mathbb{R}^{1024*2}$ to $\mathbb{R}^{28}$, making the classification task much simpler but also discarding the vast majority of the data. We use an ensemble model of gradient boosted trees (XGBoost) [24] to classify modulations from these features, which outperforms a single decision tree or support vector machine (SVM) significantly on the task.

B. Convolutional Neural Network

Since [5] and [14] the use of convolutional neural network (CNN) layers to impart translation invariance in the input, followed by fully connected layers (FC) in classifiers, has been used in the computer vision problems. In [17], the question of how to structure such networks is explored, and several basic design principals for "VGG" networks are introduced (e.g. filter size is minimized at 3x3, smallest size pooling operations are used at 2x2). Following this approach has generally led to straight forward way to construct CNNs with good performance. We adapt the VGG architecture principals to a 1D CNN, improving upon the similar networks in [30], [32]. This represents a simple DL CNN design approach which can be readily trained and deployed to effectively accomplish many small radio signal classification tasks.

Of significant note here, is that the features into this CNN are the raw I/Q samples of each radio signal example which have been normalized to unit variance. We do not perform any expert feature extraction or other pre-processing on the raw radio signal, instead allowing the network to learn raw time-

TABLE III. CNN NETWORK LAYOUT

Layer	Output dimensions
Input	2×1024
Conv	64×1024
Max Pool	64×512
Conv	64×512
Max Pool	64×256
Conv	64×256
Max Pool	64×128
Conv	64×128
Max Pool	64×64
Conv	64×64
Max Pool	64×32
Conv	64×32
Max Pool	64×16
Conv	64×16
Max Pool	64×8
FC/Selu	128
FC/Selu	128
FC/Softmax	24

series features directly on the high dimension data. Real valued networks are used, as complex valued auto-differentiation is not yet mature enough for practical use.

C. Residual Neural Network

As network algorithms and architectures have improved since Alexnet, they have made the effective training of deeper networks using more and wider layers possible, and leading to improved performance. In our original work [30] we employ only a small convolutional neural network with several layers to improve over the prior state of the art. However in the computer vision space, the idea of deep residual networks has become increasingly effective [27]. In a deep residual network, as is shown in figure 5, the notion of skip or bypass connections is used heavily, allowing for features to operate at multiple scales and depths through the network. This has led to significant improvements in computer vision performance, and has also been used effectively on time-series audio data [28]. In [34], the use of residual networks for time-series radio classification is investigated, and seen to train in fewer epochs, but not to provide significant performance improvements in terms of classification accuracy. We revisit the problem of modulation recognition with a modified residual network and obtain improved performance when compared to the CNN on this dataset. The basic residual unit and stack of residual units is shown in figure 5, while the network architecture for our best architecture for ($\ell = 1024$) is shown in table IV. We also employ self-normalizing neural networks [31] in the fully connected region of the network, employing the scaled exponential linear unit (SELU) activation function, mean-response scaled initialization (MRSA) [20], and Alpha Dropout [31], which provides a slight improvement over conventional ReLU performance.

For the two network layouts shown, with $\ell = 1024$ and $L = 5$, The ResNet has 236,344 trainable parameters, while the CNN/VGG network has a comparable 257,099 trainable parameters.

V. SENSING PERFORMANCE ANALYSIS

There are numerous design, deployment, training, and data considerations which can significantly effect the performance

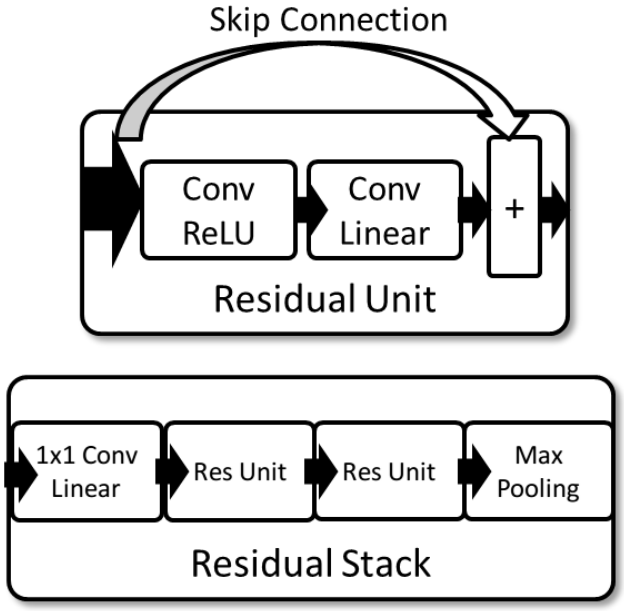

Fig. 5. Hierarchical Layers Used in Network

TABLE IV. RESNET NETWORK LAYOUT

Layer	Output dimensions
Input	2×1024
Residual Stack	32×512
Residual Stack	32×256
Residual Stack	32×128
Residual Stack	32×64
Residual Stack	32×32
Residual Stack	32×16
FC/SeLU	128
FC/SeLU	128
FC/Softmax	24

of a DL based approach to radio signal classification which must be carefully considered when designing a solution. In this section we explore several of the most common design parameters which impact classification accuracy including radio propagation effects, model size/depth, data set sizes, observation size, and signal modulation type.

A. Classification on Low Order Modulations

We first compare performance on the lower difficulty dataset on lower order modulation types. Training on a dataset of 1 million example, each 1024 samples long, we obtain excellent performance at high SNR for both the VGG CNN and the ResNet (RN) CNN.

In this case, the ResNet achieves roughly 5 dB higher sensitivity for equivalent classification accuracy than the baseline, and at high SNR a maximum classification accuracy rate of 99.8% is achieved by the ResNet, while the VGG network achieves 98.3% and the baseline method achieves a 94.6% accuracy. At lower SNRs, performance between VGG and ResNet networks are virtually identical, but at high-SNR performance improves considerably using the ResNet and obtaining almost perfect classification accuracy.

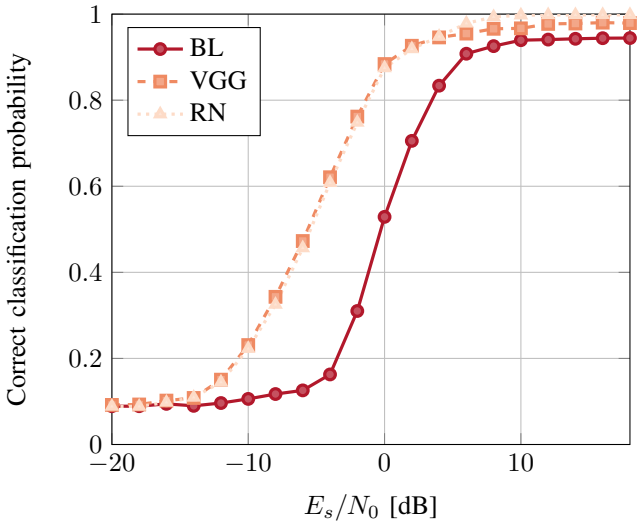

Fig. 6. 11-modulation AWGN dataset performance comparison (N=1M)

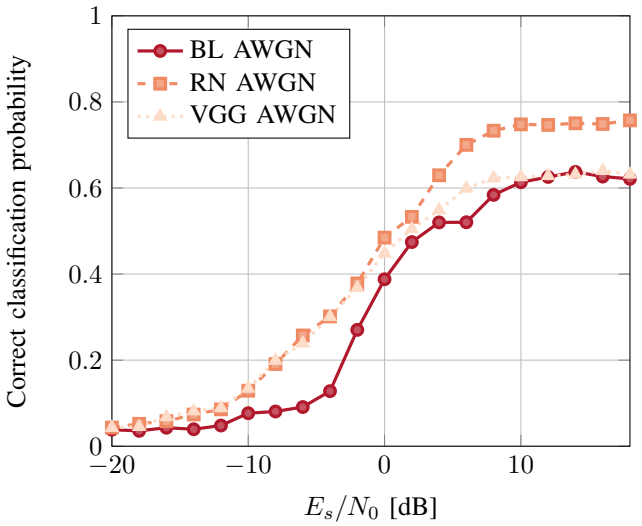

Fig. 7. Comparison models under AWGN (N=240k)

For the remainder of the paper, we will consider the much harder task of 24 class high order modulations containing higher information rates and much more easily confused classes between multiple high order PSKs, APSKs and QAMs.

B. Classification under AWGN conditions

Signal classification under additive white gaussian noise (AWGN) is the canonical problem which has been explored for many years in communications literature. It is a simple starting point, and it is the condition under which analytic feature extractors should generally perform their best (since they were derived under these conditions). In figure 7 we compare the performance of the ResNet (RN), VGG network, and the baseline (BL) method on our full dataset for $\ell = 1024$ samples, $N = 239,616$ examples, and $L = 6$ residual stacks. Here, the residual network provides the best performance at both high and low SNRs on the difficult dataset by a margin of 2-6 dB in improved sensitivity for equivalent classification accuracy.

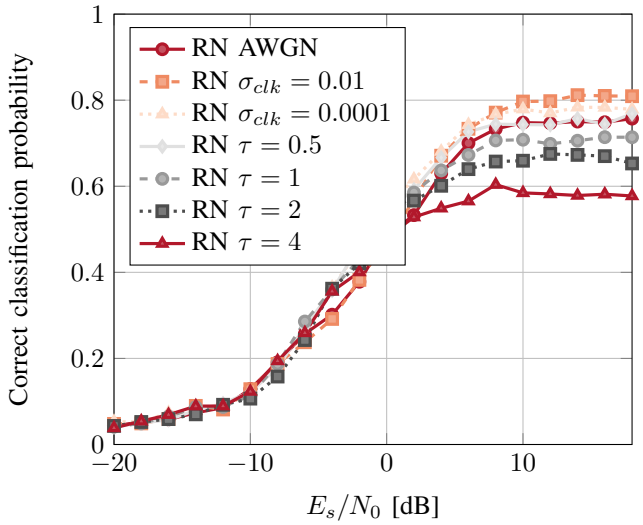

Fig. 8. Resnet performance under various channel impairments (N=240k)

C. Classification under Impairments

In any real world scenario, wireless signals are impaired by a number of effects. While AWGN is widely used in simulation and modeling, the effects described above are present almost universally. It is interesting to inspect how well learned classifiers perform under such impairments and compare their rate of degradation under these impairments with that of more traditional approaches to signal classification.

In figure 8 we plot the performance of the residual network based classifier under each considered impairment model. This includes AWGN, $\sigma_{clk} = 0.0001$ - minor LO offset, $\sigma_{clk} = 0.01$ - moderate LO offset, and several fading models ranging from $\tau = 0.5$ to $\tau = 4.0$. Under the fading models, moderate LO offset is assumed as well. Interestingly in this plot, ResNet performance improves under LO offset rather than degrading. Additional LO offset which results in spinning or dilated versions of the original signal, appears to have a positive regularizing effect on the learning process which provides quite a noticeable improvement in performance. At high SNR performance ranges from around 80% in the best case down to about 59% in the worst case.

In figure 9 we show the degradation of the baseline classifier under impairments. In this case, LO offset never helps, but the performance instead degrades with both LO offset and fading effects, in the best case at high SNR this method obtains about 61% accuracy while in the worst case it degrades to around 45% accuracy.

Directly comparing the performance of each model under moderate LO impairment effects, in figure 10 we show that

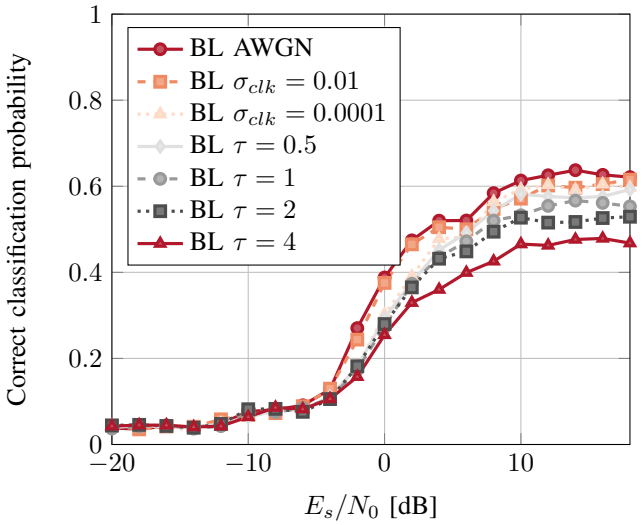

Fig. 9. Baseline performance under channel impairments (N=240k)

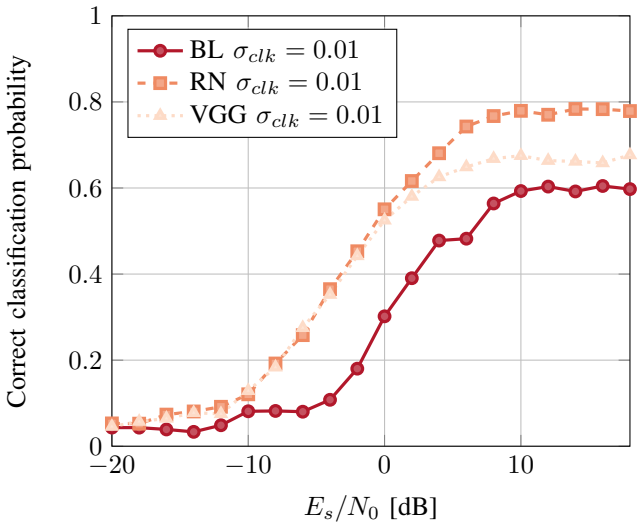

Fig. 10. Comparison models under LO impairment

for many real world systems with unsynchronized LOs and Doppler frequency offset there is nearly a 6dB performance advantage of the ResNet approach vs the baseline, and a 20% accuracy increase at high SNR. In this section, all models are trained using $N = 239,616$ and $\ell = 1024$ for this comparison.

D. Classifier performance by depth

Model size can have a significant impact on the ability of large neural network models to accurately represent complex features. In computer vision, convolutional layer based DL models for the ImageNet dataset started around 10 layers deep, but modern state of the art networks on ImageNet are often over 100 layers deep [22], and more recently even over 200 layers. Initial investigations of deeper networks in [34] did not

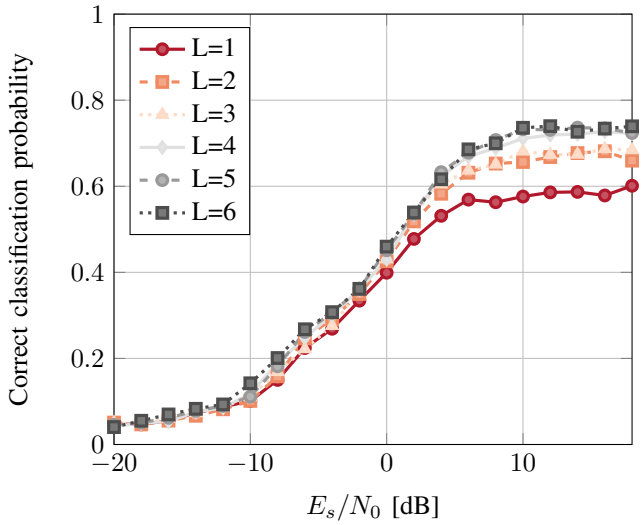

Fig. 11. ResNet performance vs depth (L = number of residual stacks)

show significant gains from such large architectures, but with use of deep residual networks on this larger dataset, we begin to see quite a benefit to additional depth. This is likely due to the significantly larger number of examples and classes used. In figure 11 we show the increasing validation accuracy of deep residual networks as we introduce more residual stack units within the network architecture (i.e. making the network deeper). We see that performance steadily increases with depth in this case with diminishing returns as we approach around 6 layers. When considering all of the primitive layers within this network, when $L = 6$ we the ResNet has 121 layers and 229k trainable parameters, when $L = 0$ it has 25 layers and 2.1M trainable parameters. Results are shown for $N = 239,616$ and $\ell = 1024$.

E. Classification performance by modulation type

In figure 12 we show the performance of the classifier for individual modulation types. Detection performance of each modulation type varies drastically over about 18dB of signal to noise ratio (SNR). Some signals with lower information rates and vastly different structure such as AM and FM analog modulations are much more readily identified at low SNR, while high-order modulations require higher SNRs for robust performance and never reach perfect classification rates. However, all modulation types reach rates above 80% accuracy by around 10dB SNR. In figure 13 we show a confusion matrix for the classifier across all 24 classes for AWGN validation examples where SNR is greater than or equal to zero. We can see again here that the largest sources of error are between high order phase shift keying (PSK) (16/32-PSK), between high order quadrature amplitude modulation (QAM) (64/128/256-QAM), as well as between AM modes (confusing with-carrier (WC) and suppressed-carrier (SC)). This is largely to be expected as for short time observations, and under noisy observations, high order QAM and PSK can be extremely difficult to tell apart through any approach.

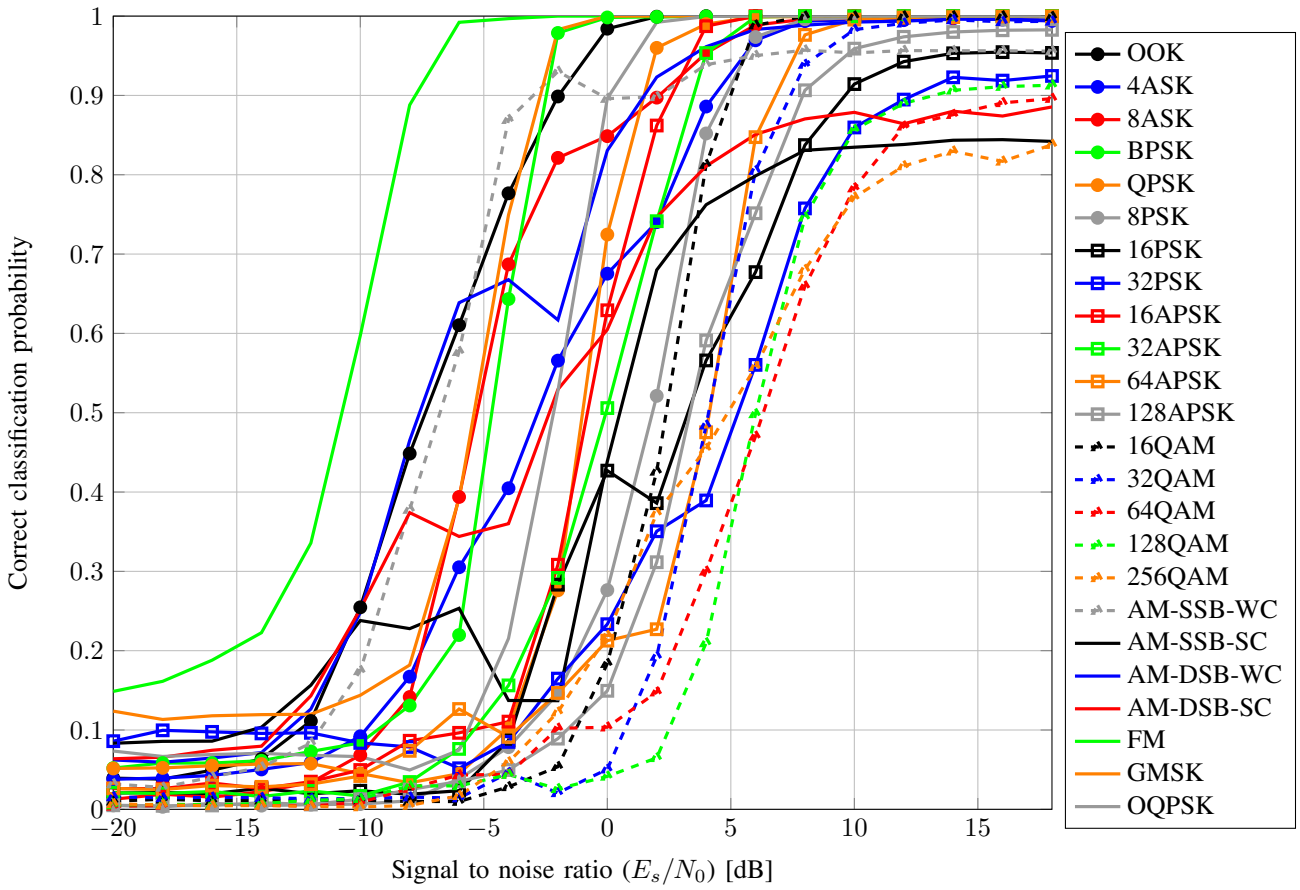

Fig. 12. Modrec performance vs modulation type (Resnet on synthetic data with N=1M, σ_{clk}=0.0001)

F. Classifier Training Size Requirements

When using data-centric machine learning methods, the dataset often has an enormous impact on the quality of the model learned. We consider the influence of the number of example signals in the training set, N, as well as the time-length of each individual example in number of samples, ℓ.

In figure 14 we show how performance of the resulting model changes based on the total number of training examples used. Here we see that dataset size has a dramatic impact on model training, high SNR classification accuracy is near random until 4-8k examples and improves 5-20% with each doubling until around 1M. These results illustrate that having sufficient training data is critical for performance. For the largest case, with 2 million examples, training on a single state of the art Nvidia V100 graphics processing unit (GPU) (with approximately 125 tera-floating point operations per second (FLOPS)) takes around 16 hours to reach a stopping point, making significant experimentation at these dataset sizes cumbersome. We do not see significant improvement going from 1M to 2M examples, indicating a point of diminishing returns for number of examples around 1M with this configuration. With either 1M or 2M examples we obtain roughly 95% test set accuracy at high SNR. The class-confusion matrix for the best performing mode with ℓ=1024 and N=1M is shown in figure 15 for test examples at or above 0dB SNR, in all instances here we use the $\sigma_{clk} = 0.0001$ dataset, which yields slightly better performance than AWGN.

Figure 16 shows how the model performance varies by window size, or the number of time-samples per example used for a single classification. Here we obtain approximately a 3% accuracy improvement for each doubling of the input size (with N=240k), with significant diminishing returns once we reach $\ell = 512$ or $\ell = 1024$. We find that CNNs scale very well up to this 512-1024 size, but may need additional scaling strategies thereafter for larger input windows simply due to memory requirements, training time requirements, and dataset requirements.

G. Over the air performance

We generate 1.44M examples of the 24 modulation dataset over the air using the USRP setup described above. Using a partition of 80% training and 20% test, we can directly train a ResNet for classification. Doing so on an Nvidia V100 in around 14 hours, we obtain a 95.6% test set accuracy on the over the air dataset, where all examples are roughly 10dB SNR.

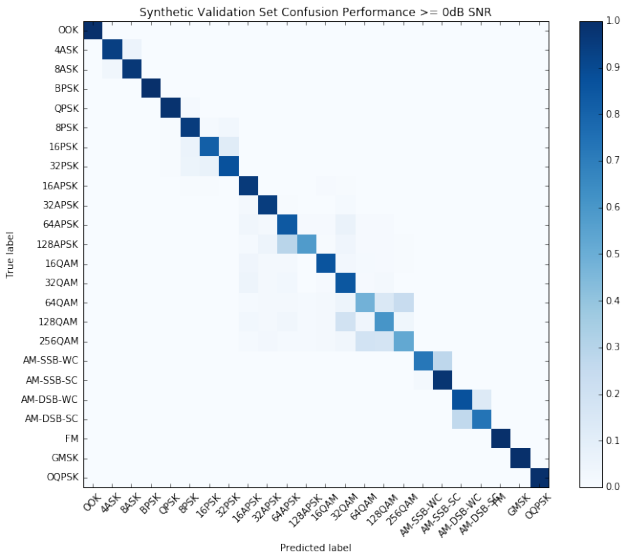

Fig. 13. 24-modulation confusion matrix for ResNet trained and tested on synthetic dataset with N=1M and AWGN

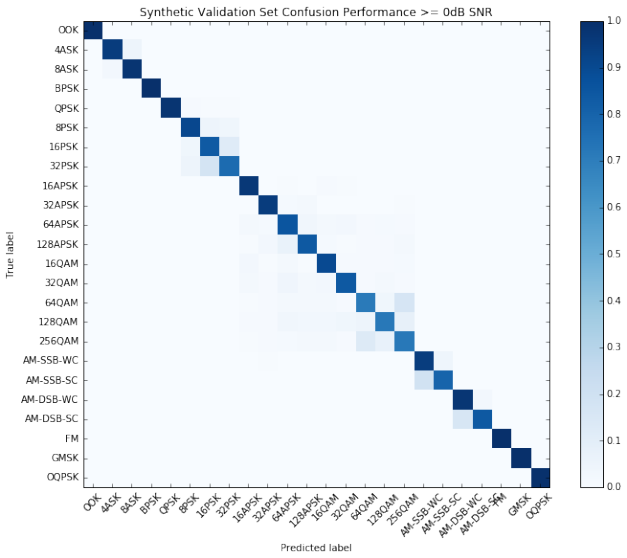

Fig. 15. 24-modulation confusion matrix for ResNet trained and tested on synthetic dataset with N=1M and $\sigma_{clk} = 0.0001$

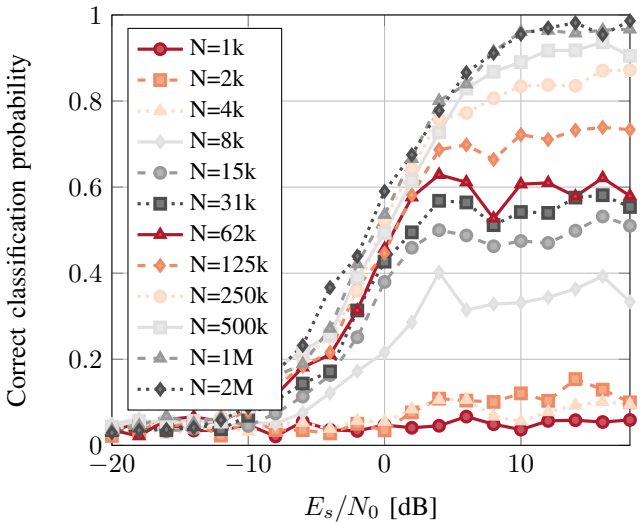

Fig. 14. Performance vs training set size (N) with ℓ = 1024

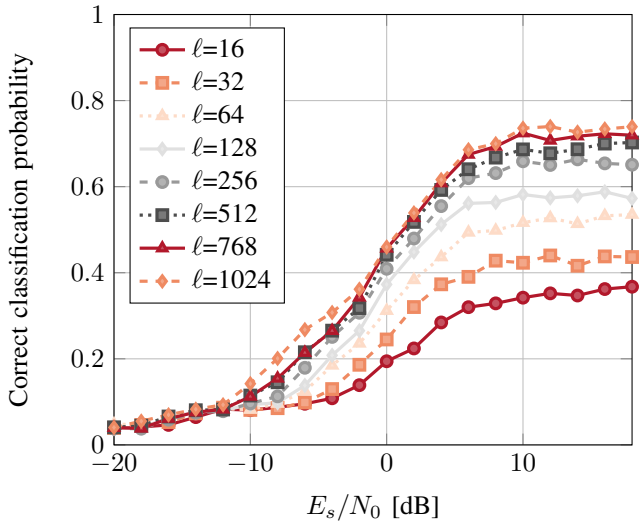

Fig. 16. Performance vs example length in samples (ℓ)

A confusion matrix for this OTA test set performance based on direct training is shown in figure 17.

H. Transfer learning over-the-air performance

We also consider over the air signal classification as a transfer learning problem, where the model is trained on synthetic data and then only evaluated and/or fine-tuned on OTA data. Because full model training can take hours on a high end GPU and typically requires a large dataset to be effective, transfer learning is a convenient alternative for leveraging existing models and updating them on smaller computational platforms and target datasets. We consider transfer learning, where we freeze network parameter weights for all layers except the last several fully connected layers (last three layers from table IV) in our network when while updating. This is commonly done today with computer vision models where it is common start by using pre-trained VGG or other model weights for ImageNet or similar datasets and perform transfer learning using another dataset or set of classes. In this case, many low-level features work well for different classes or datasets, and do not need to change during fine tuning. In our case, we consider several cases where we start with models trained on simulated wireless impairment models using residual networks and then evaluate them on OTA examples. The accuracies of our initial models (trained with N=1M) on synthetic data shown in figure 8, and these ranged from 84% to 96% on the hard 24-class

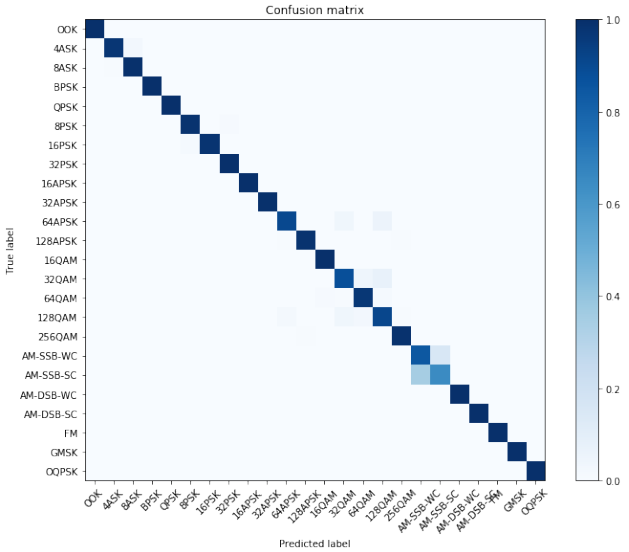

Fig. 17. 24-modulation confusion matrix for ResNet trained and tested on OTA examples with SNR $\sim$ 10 dB

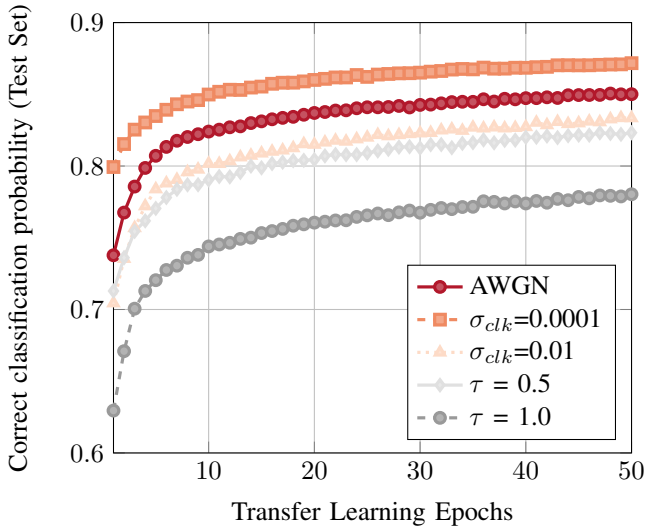

Fig. 18. RESNET Transfer Learning OTA Performance (N=120k)

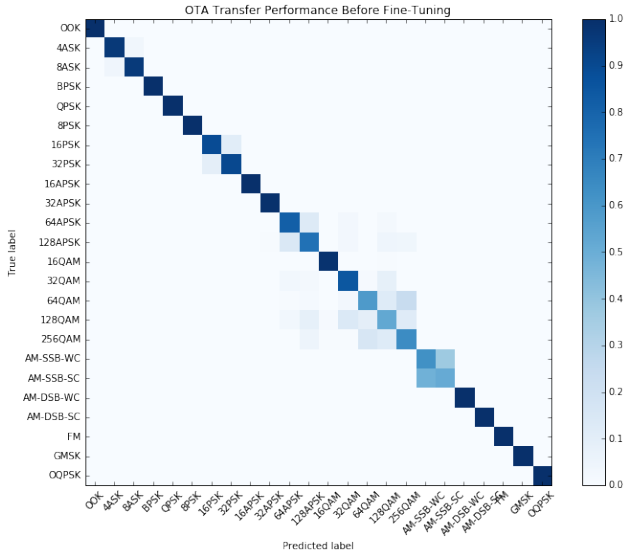

Fig. 19. 24-modulation confusion matrix for ResNet trained on synthetic $\sigma_{clk} = 0.0001$ and tested on OTA examples with SNR $\sim$ 10 dB (prior to fine-tuning)

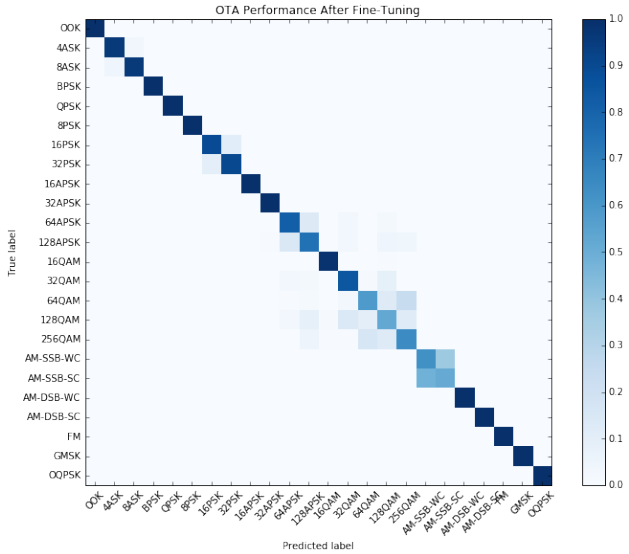

Fig. 20. 24-modulation confusion matrix for ResNet trained on synthetic $\sigma_{clk} = 0.0001$ and tested on OTA examples with SNR $\sim$ 10 dB (after fine-tuning)

dataset. Evaluating performance of these models on OTA data, without any model updates, we obtain classification accuracies between 64% and 80%. By fine-tuning the last two layers of these models on the OTA data using transfer learning, we and can recover approximately 10% of additional accuracy. The validation accuracies are shown for this process in figure 18. These ResNet update epochs on dense layers for 120k examples take roughly 60 seconds on a Titan X card to execute instead of the full $\sim$ 500 seconds on V100 card per epoch when updating model weights.

Ultimately, the model trained on just moderate LO offset (σ_{clk} = 0.0001) performs the best on OTA data. The model obtained 94% accuracy on synthetic data, and drops roughly 7% accuracy when evaluating on OTA data, obtaining an accuracy of 87%. The primary confusion cases prior to training seem to be dealing with suppress or non-suppressed carrier analog signals, as well as the high order QAM and APSK modes.

This seems like it is perhaps the best suited among our models to match the OTA data. Very small LO impairments are present in the data, the radios used had extremely stable oscillators present (GPSDO modules providing high stable $\tilde{7}5$ PPB clocks) over very short example lengths (1024 samples),

and that the two radios were essentially right next to each other, providing a very clean impulsive direct path while any reflections from the surrounding room were likely significantly attenuated in comparison, making for a near impulsive channel. Training on harsher impairments seemed to degrade performance of the OTA data significantly.

We suspect as we evaluate the performance of the model under increasingly harsh real world scenarios, our transfer learning will favor synthetic models which are similarly impaired and most closely match the real wireless conditions (e.g. matching LO distributions, matching fading distributions, etc). In this way, it will be important for this class of systems to train either directly on target signal environments, or on very good impairment simulations of them under which well suited models can be derived. Possible mitigation to this are to include domain-matched attention mechanisms such as the radio transformer network [29] in the network architecture to improve generalization to varying wireless propagation conditions.

VI. Discussion

In this work we have extended prior work on using deep convolutional neural networks for radio signal classification by heavily tuning deep residual networks for the same task. We have also conducted a much more thorough set of performance evaluations on how this type of classifier performs over a wide range of design parameters, channel impairment conditions, and training dataset parameters. This residual network approach achieves state of the art modulation classification performance on a difficult new signal database both synthetically and in over the air performance. Other architectures still hold significant potential, radio transformer networks, recurrent units, and other approaches all still need to be adapted to the domain, tuned and quantitatively benchmarked against the same dataset in the future. Other works have explored these to some degree, but generally not with sufficient hyper-parameter optimization to be meaningful.

We have shown that, contrary to prior work, deep networks do provide significant performance gains for time-series radio signals where the need for such deep feature hierarchies was not apparent, and that residual networks are a highly effective way to build these structures where more traditional CNNs such as VGG struggle to achieve the same performance or make effective use of deep networks. We have also shown that simulated channel effects, especially moderate LO impairments improve the effect of transfer learning to OTA signal evaluation performance, a topic which will require significant future investigation to optimize the synthetic impairment distributions used for training.

VII. Conclusion

DL methods continue to show enormous promise in improving radio signal identification sensitivity and accuracy, especially for short-time observations. We have shown deep networks to be increasingly effective when leveraging deep residual architectures and have shown that synthetically trained deep networks can be effectively transferred to over the air datasets with (in our case) a loss of around 7% accuracy or directly trained effectively on OTA data if enough training data is available. While large well labeled datasets can often be difficult to obtain for such tasks today, and channel models can be difficult to match to real-world deployment conditions, we have quantified the real need to do so when training such systems and helped quantify the performance impact of doing so.

We still have much to learn about how to best curate datasets and training regimes for this class of systems. However, we have demonstrated in this work that our approach provides roughly the same performance on high SNR OTA datasets as it does on the equivalent synthetic datasets, a major step towards real world use. We have demonstrated that transfer learning can be effective, but have not yet been able to achieve equivalent performance to direct training on very large datasets by using transfer learning. As simulation methods become better, and our ability to match synthetic datasets to real world data distributions improves, this gap will close and transfer learning will become and increasingly important tool when real data capture and labeling is difficult. The performance trades shown in this work help shed light on these key parameters in data generation and training, hopefully helping increase understanding and focus future efforts on the optimization of such systems.

References

[1] S. Imai, "Cepstral analysis synthesis on the mel frequency scale," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'83.*, IEEE, vol. 8, 1983, pp. 93–96.

[2] W. A. Gardner and C. M. Spooner, "Signal interception: Performance advantages of cyclic-feature detectors," *IEEE Transactions on Communications*, vol. 40, no. 1, pp. 149–159, 1992.

[3] C. M. Spooner and W. A. Gardner, "Robust feature detection for signal interception," *IEEE transactions on communications*, vol. 42, no. 5, pp. 2165–2173, 1994.

[4] J. R. Quinlan *et al.*, "Bagging, boosting, and c4. 5," in *AAAI/IAAI, Vol. 1*, 1996, pp. 725–730.

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[6] A. K. Nandi and E. E. Azzouz, "Algorithms for automatic modulation recognition of communication signals," *IEEE Transactions on communications*, vol. 46, no. 4, pp. 431–436, 1998.

[7] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[8] M. Vidal-Naquet and S. Ullman, "Object recognition with informative features and linear classification.," in *ICCV*, vol. 3, 2003, p. 281.

[9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[10] A Fehske, J Gaeddert, and J. H. Reed, "A new approach to signal classification using spectral correlation and neural networks," in *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on*, IEEE, 2005, pp. 144–150.

[11] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.

[12] A. Goldbloom, "Data prediction competitions–far more than just a bit of fun," in *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, IEEE, 2010, pp. 1385–1386.

[13] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[15] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.

[16] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ArXiv preprint arXiv:1412.6980*, 2014.

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *ArXiv preprint arXiv:1409.1556*, 2014.

[18] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting.," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[19] M. Ettus and M. Braun, "The universal software radio peripheral (usrp) family of low-cost sdrd," *Opportunistic Spectrum Sharing and White Space Access: The Practical Reality*, pp. 3–23, 2015.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[21] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[22] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[23] A. Abdelmutalab, K. Assaleh, and M. El-Tarhuni, "Automatic modulation classification based on high order cumulants and hierarchical polynomial classifiers," *Physical Communication*, vol. 21, pp. 10–18, 2016.

[24] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, ACM, 2016, pp. 785–794.

[25] S. Cioni, G. Colavolpe, V. Mignone, A. Modenini, A. Morello, M. Ricciulli, A. Ugolini, and Y. Zanettini, "Transmission parameters optimization and receiver architectures for dvb-s2x systems," *International Journal of Satellite Communications and Networking*, vol. 34, no. 3, pp. 337–350, 2016.

[26] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[27] 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*, 2016, pp. 770–778.

[28] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *ArXiv preprint arXiv:1609.03499*, 2016.

[29] T. J. O'Shea and N. West, "Radio machine learning dataset generation with gnu radio," in *Proceedings of the GNU Radio Conference*, vol. 1, 2016.

[30] T. J. OShea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *International Conference on Engineering Applications of Neural Networks*, Springer, 2016, pp. 213–226.

[31] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," *ArXiv preprint arXiv:1706.02515*, 2017.

[32] T. OShea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, 2017.

[33] C. M. Spooner, A. N. Mody, J. Chuang, and J. Petersen, "Modulation recognition using second-and higher-order cyclostationarity," in *Dynamic Spectrum Access Networks (DySPAN), 2017 IEEE International Symposium on*, IEEE, 2017, pp. 1–3.

[34] N. E. West and T. J. O'Shea, "Deep architectures for modulation recognition," in *IEEE International Symposium on Dynamic Spectrum Access Networks*, IEEE, 2017.

[35] A. D.-R.A. T. AD9361, "Url: Https://tinyurl.com/hwxym94 (visited on 09/14/08)," *Cited on*, p. 103,

[36] J. G. Proakis, "Digital communications. 1995," *McGraw-Hill, New York*,

APPENDIX

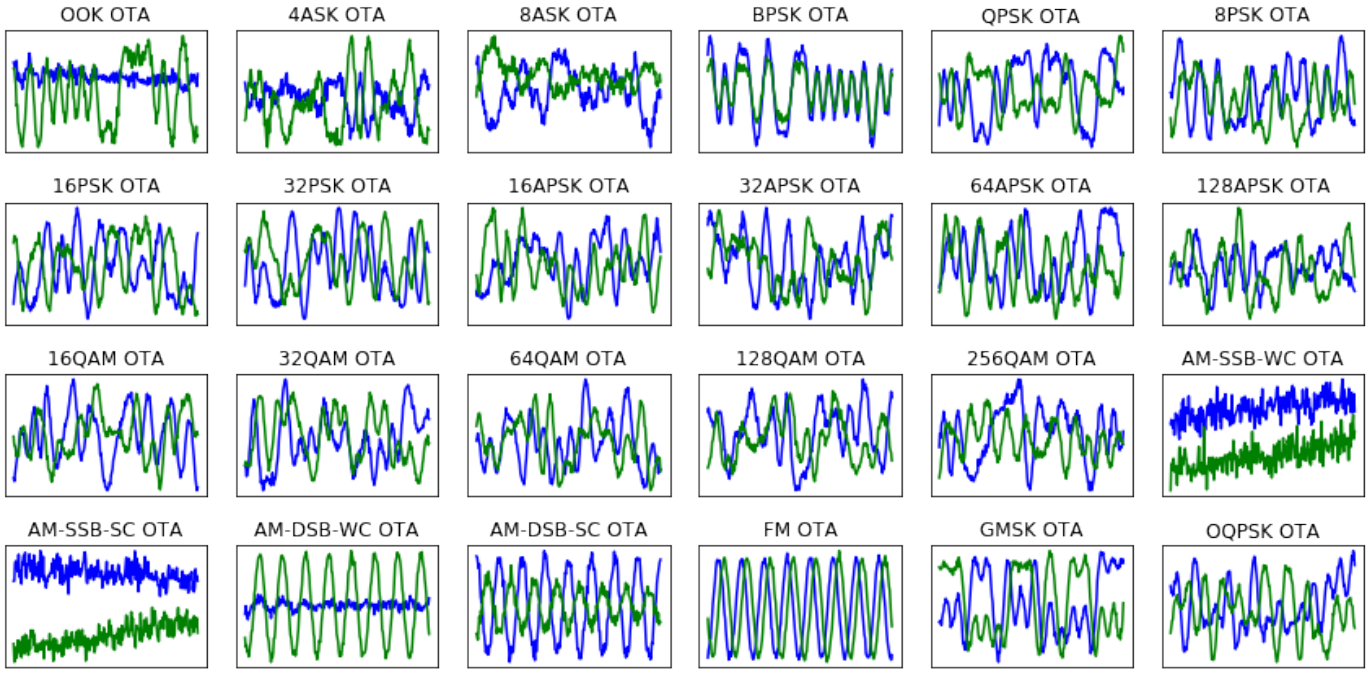

Fig. 21. I/Q time domain examples of 24 modulations over the air at roughly 10 dB E_s/N_0 ($\ell = 256$)

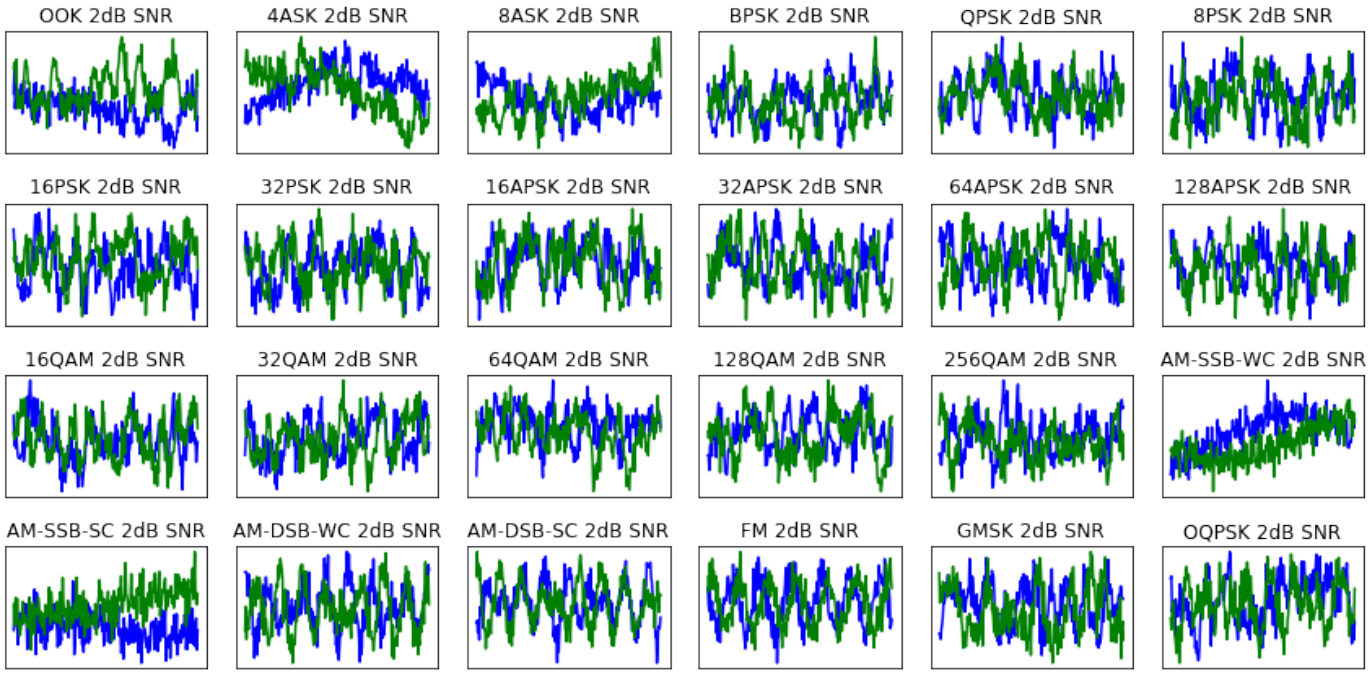

Fig. 22. I/Q time domain examples of 24 modulations from synthetic $\sigma_{clk} = 0.01$ dataset at 2dB E_s/N_0 ($\ell = 256$)