electronics

Article

Automatic Modulation Classification with Deep Neural Networks

Clayton A. Harper *, Mitchell A. Thornton and Eric C. Larson

Computer Science, Lyle School of Engineering, Southern Methodist University, Dallas, TX 75205, USA; mitch@smu.edu (M.A.T.); eclarson@smu.edu (E.C.L.)
* Correspondence: caharper@smu.edu

Abstract: Automatic modulation classification is an important component in many modern aeronautical communication systems to achieve efficient spectrum usage in congested wireless environments and other communications systems applications. In recent years, numerous convolutional deep learning architectures have been proposed for automatically classifying the modulation used on observed signal bursts. However, a comprehensive analysis of these differing architectures and the importance of each design element has not been carried out. Thus, it is unclear what trade-offs the differing designs of these convolutional neural networks might have. In this research, we investigate numerous architectures for automatic modulation classification and perform a comprehensive ablation study to investigate the impacts of varying hyperparameters and design elements on automatic modulation classification accuracy. We show that a new state-of-the-art accuracy can be achieved using a subset of the studied design elements, particularly as applied to modulation classification over intercepted bursts of varying time duration. In particular, we show that a combination of dilated convolutions, statistics pooling, and squeeze-and-excitation units results in the strongest performing classifier achieving 98.9% peak accuracy and 63.7% overall accuracy on the RadioML 2018.01A dataset. We further investigate this best performer according to various other criteria, including short signal bursts of varying length, common misclassifications, and performance across differing modulation categories and modes.

Keywords: automatic modulation classification; machine learning; convolutional neural networks

Citation: Harper, C.A.; Thornton, M.A.; Larson, E.C. Automatic Modulation Classification with Deep Neural Networks. *Electronics* **2023**, *12*, 3962. https://doi.org/10.3390/electronics12183962

Academic Editors: Yichuang Sun, Haeyoung Lee and Oluyomi Simpson

Received: 14 August 2023
Revised: 15 September 2023
Accepted: 18 September 2023
Published: 20 September 2023

1. Introduction

Automatic modulation classification (AMC) holds particular significance in aerospace applications, specifically in radio frequency (RF) signal analysis and modern software-defined radios. It serves a multitude of crucial tasks including "spectrum interference monitoring, radio fault detection, dynamic spectrum access, opportunistic mesh networking, and numerous regulatory and defense applications" [1]. Upon detection of an RF signal with unknown characteristics, AMC is a crucial initial procedure in order to demodulate the signal for receivers supporting a variety of standard and non-standard modulation schemes. Efficient AMC allows for maximal usage of transmission mediums and can enhance resilience in modern cognitive radios. Systems capable of adaptive modulation schemes can monitor current channel conditions with AMC and adjust exercised modulation schemes to maximize usage across the transmission medium.

Moreover, for receivers that have a versatile demodulation capability, AMC is a requisite task. The correct demodulation scheme must be applied, as a first step, to recover the modulated message within a detected signal. Aerospace communication systems, such as those employed in satellites, unmanned aerial vehicles (UAVs), and aircraft often operate in dynamic and congested environments [2]. AMC is critical in these applications to ensure efficient spectrum utilization. In systems where the modulation scheme is unknown a priori, AMC allows for efficient prediction of the employed modulation scheme.

Higher performing AMC can increase the throughput and accuracy of these systems; therefore, AMC is currently an important research topic in the fields of machine learning and communication systems, specifically for software-defined radios.

Common benchmarks are formulated with the underlying assumption that the AMC model needs to perform classification for both the modulation mode (e.g., QAM) and the specific variant within that mode (e.g., 32QAM as opposed to 64QAM). While many architectures have proven to be effective at high signal-to-noise ratios (SNRs), performance degrades significantly at lower SNRs that often occur in real-world applications. Other works have investigated increasing classification performance at lower SNR levels through the use of SNR-specific modulation classifiers [3] and clustering based on SNR ranges [4]. For the purpose of classification, various signal characteristics have been explored. Traditionally, AMC has made use of statistical moments and higher-order cumulants derived from the received signal [5,6]. Recently, direct employment of the raw in-phase (I) and quadrature (Q) components in the time domain have been embraced [1,7–9]. Additionally, alternative studies have investigated supplementary attributes, including I/Q constellation plots [10–13].

Upon the selection of signal input features, the subsequent step involves the utilization of machine learning models to discern statistical patterns within the data for classification. Classifiers such as support vector machines, decision trees, K-nearest neighbors, and neural networks are commonly used for this application [1,4,7–10,14–17]. Residual neural networks (ResNets), along with convolutional neural networks (CNNs), have been shown to achieve high classification performance for AMC [1,4,7–10,18–21]. Thus, deep learning-based methods in AMC have become more prevalent due to their promising performance and their ability to generalize to large, complex datasets comprising a variety of standard and non-standard modulation schemes.

While other works have contributed to increased AMC performance, the importance of many design elements for AMC remains unclear and a number of architectural elements have yet to be investigated. Therefore, in this work, we aim to formalize the impact of a variety of architectural changes and model design decisions on AMC performance. Numerous modifications to architectures from previous works, including our own [7], and novel combinations of elements applied to AMC are considered. After an initial investigation, we provide a comprehensive ablation study in this work to investigate the performance impact of various architectural modifications. Additionally, we achieve new state-of-the-art classification performance on the RadioML 2018.01A dataset [22] that benefits from the results of the ablation study. Using the best-performing model, we provide additional analyses that characterize its performance across modulation modes and signals with variable duration bursts.

2. Related Work

The area of AMC has been investigated by several research groups. We provide a summary of recent results in AMC to provide context and motivation for our contributions to AMC and the corresponding ablation study described in this paper. The results of the ablation study are then used to determine a new AMC architecture that demonstrates increased performance.

Corgan et al. demonstrate that deep convolutional neural networks exhibit notable classification efficacy, particularly under low SNRs, evidenced by their study on a dataset encompassing 11 distinct modulation types [8]. It was found that CNNs exceeded performance over expertly crafted features. Comparing results with architectures in [1,8], Liu et al. improved AMC performance utilizing self-supervised contrastive learning [23]. First, an encoder is pre-trained in a self-supervised manner through creating contrastive pairs with data augmentation. By creating different views of the input data through augmentation, contrastive loss is used to maximize the cosine similarity between positive pairs (augmented views of the same input). Once converged, the encoder is frozen (i.e., the weights are set to fixed values) and two fully-connected layers are added following the encoder to form the classifier. The classifier is trained using supervised learning to predict the 11 different modulation schemes. Chen et al. applied a novel architecture to the same dataset where the input signal is sliced and transformed into a square matrix and applies a residual network to predict the modulation schemes [24]. A multidimensional CNN-LSTM architecture was utilized in [25], where the CNN performed feature extraction that would later be processed by LSTM (long short-term memory) [26] and classification layers. Other work has investigated empirical and variational mode decomposition to improve few-shot learning for AMC [27]. In our work, we utilize a larger, more complex dataset consisting of 24 modulation schemes, as well as modeling improvements.

Spectrograms and I/Q constellation plots in [28] were found to be effective input features to a traditional CNN achieving nearly equivalent performance as the baseline CNN network in [1], which used raw I/Q signals. Furthermore, Refs. [10–12] employed I/Q constellations as input features in their machine learning models, focusing on a more constrained context involving four or eight modulation types. Additionally, other approaches have been explored for AMC. For instance, Refs. [29,30] utilized statistical features in conjunction with support vector machines, while [31,32] integrated fusion methodologies into CNN classifiers. Mao et al. utilized various constellation diagrams at varying symbol timings, alleviating symbol timing synchronization concerns [33]. A squeeze-and-excitation-inspired [34] architecture was used as an attention mechanism to focus on the most important diagrams.

Although spectrograms and constellation plots have shown promise, they require additional processing overhead and have had comparable performance to raw I/Q signals. In addition, models that use raw I/Q signals could be more adept at handling varying-length signals than constellation plots because they are not limited by periodicity constraints for short-duration signals (i.e., burst transmissions). Consequently, we utilize raw I/Q signals in our work.

Expanding upon these investigations, Tridgell's dissertation [35] explores the application of these architectures within the context of resource-limited Field Programmable Gate Arrays (FPGAs). His research underscores the significance of parameter reduction for modulation classifiers, given their typical deployment in embedded systems characterized by resource constraints. Addressing this concern, Mendis et al. proposed the use of multiplierless deep belief networks that map directly to binary circuits [36].

In [1], Oshea et al. created a dataset with 24 different types of modulation, known as RadioML 2018.01A, and achieved high classification performance using convolutional neural networks, specifically using residual connections (see Figure 1) within the network (ResNet). A total of six residual stacks were used in the architecture. A residual stack is defined as a series of a convolutional layers, residual units, and a max pooling operation as shown in Figure 1. The ResNet employed by [1] attained approximately 95% classification accuracy at high SNR values. Wang et al. also made use of residual connections along with depthwise separable convolutions for feature extraction [37]. This architecture was able to achieve a maximum performance of 97% accuracy and an average of 53.85% accuracy across all signal-to-noise ratios while greatly reducing model complexity.

Harper et al. proposed the use of X-Vectors [38] to increase classification performance using CNNs [7]. X-Vectors are traditionally used in speaker recognition and verification systems making use of aggregate statistics. X-Vectors utilize statistical moments, specifically the mean and variance, computed over convolutional filter outputs. It can be postulated that computing the mean and variance of the embedding layer contributes to the removal of signal-specific details, leaving broader modulation-specific characteristics. Figure 2 illustrates the X-Vector architecture, where statistics are computed over the activations from a convolutional layer producing a fixed-length vector.

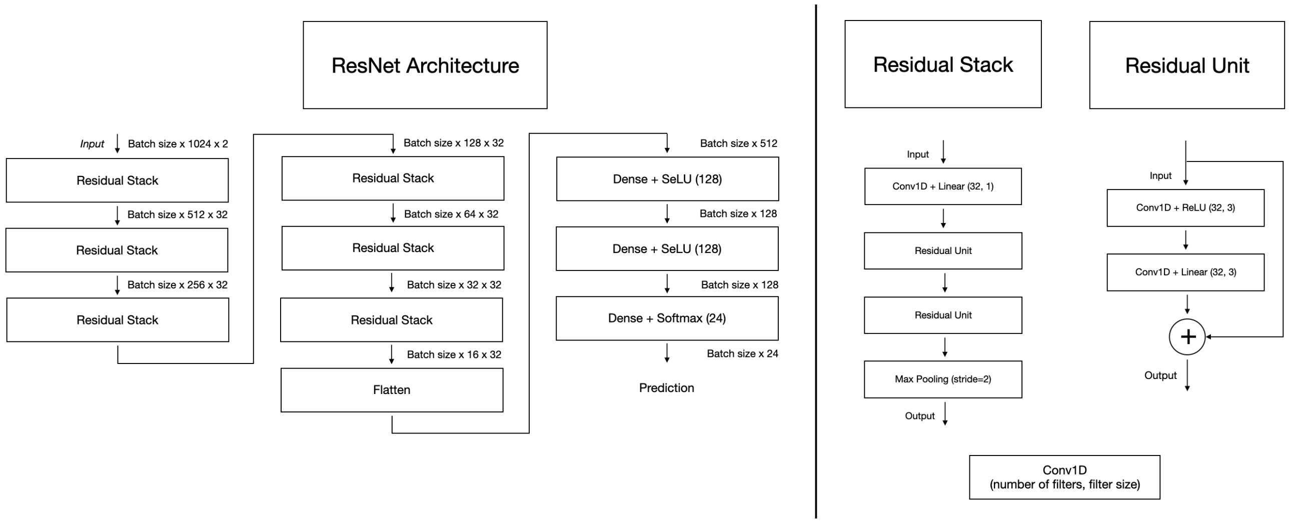


Figure 1. ResNet architecture used in [1]. Each block represents a unit in the network, which may be composed of several layers and connections as shown on the right of the figure. Dimensions of the tensors on the output of each block are also shown where appropriate.

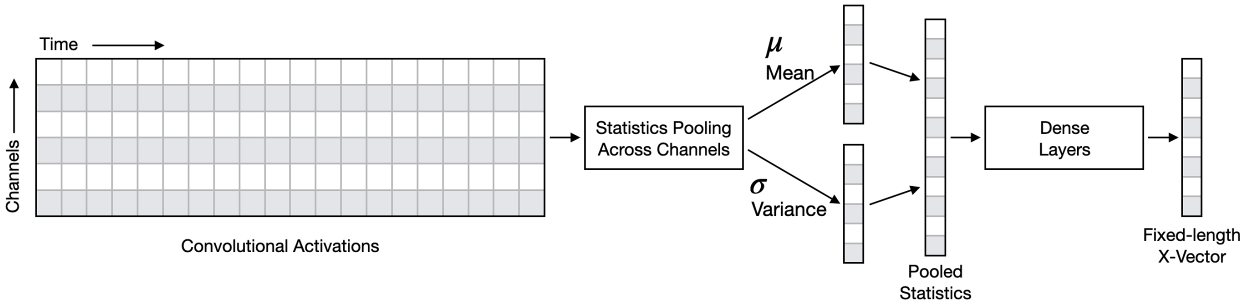


Figure 2. X-Vector architecture overview. The convolutional activations immediately before pooling are shown. These activations are fed into two statistical pooling layers that collapse the activations over time, creating a fixed-length tensor that can be further processed by fully connected dense layers.

Additionally, this architecture upholds a completely convolutional framework, enabling adaptability to inputs of varying sizes within the network. The utilization of statistical aggregations capitalizes on this characteristic. With statistical aggregations, the input to the initial dense layer becomes contingent upon the quantity of filters in the final convolutional layer. The number of filters is a hyperparameter that remains distinct from the temporal length of the input signal fed into the neural network.

In the absence of statistical aggregations, input signals for a conventional CNN or ResNet would require resampling, cropping, or padding to attain a consistent temporal length for the subsequent dense layers. While the dataset used in this work has uniformly sized signals in terms of duration, (1024 × 2), this is an architectural advantage in our deployment, as received signals may vary in duration. Instead of modifying the inputs to the network via sampling, cropping, padding, etc., the X-Vector architecture can directly operate with variable-length inputs without modifications to the network or input signal. Work by Li et al. [39] utilizes LSTMs while highlighting this desirable characteristic.

Figure 3 outlines the employed X-Vector architecture in [7] where $F = [f_1, f_2, ..., f_7] = 64$ and $K = [k_1, k_2, ..., k_7] = 3$. Mean and variance pooling are performed on the final convolutional outputs, concatenated, and fed through a series of dense layers creating the fixed-length X-Vector. A maximum of 98% accuracy was achieved at high SNR levels.

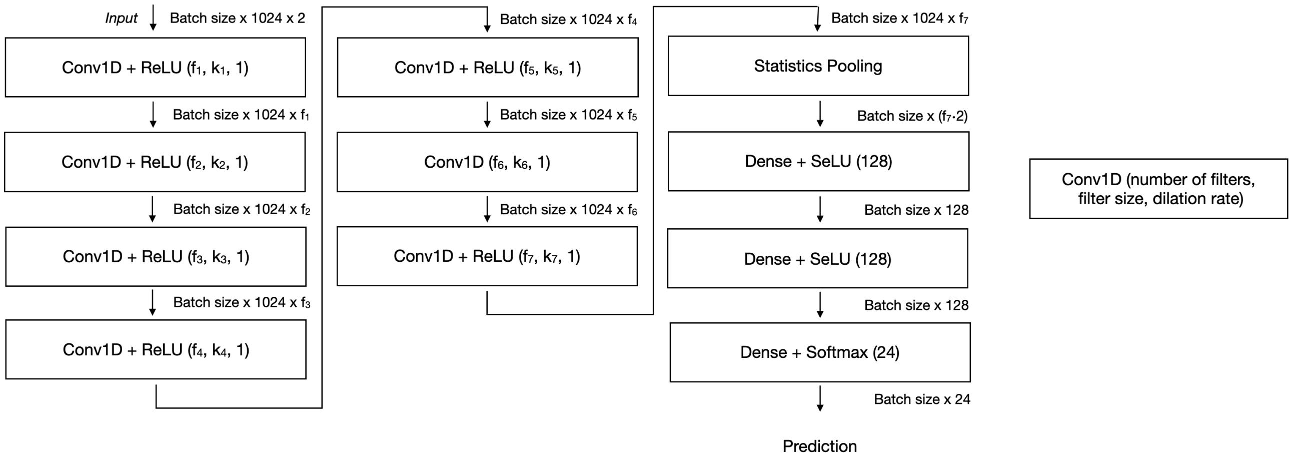


Figure 3. Proposed CNN Architecture in [7]. This is the first work to employ an X-Vector-inspired architecture for AMC showing strong performance. This architecture is used as a baseline for the modifications investigated in this paper. The f and k variables shown designate the number of kernels and size of each kernel, respectively, in each layer. These parameters are investigated for optimal sizing in our initial investigation.

The work of [7] replicated the ResNet architecture from [1] and compared the results with the X-Vector architectures as seen in Figure 4. Harper et al. [7] were able to reproduce this architecture, achieving a maximum of 93.7% accuracy. The authors attribute the difference in performance to differences in the train and test set separation that they used, since these parameters were unavailable.

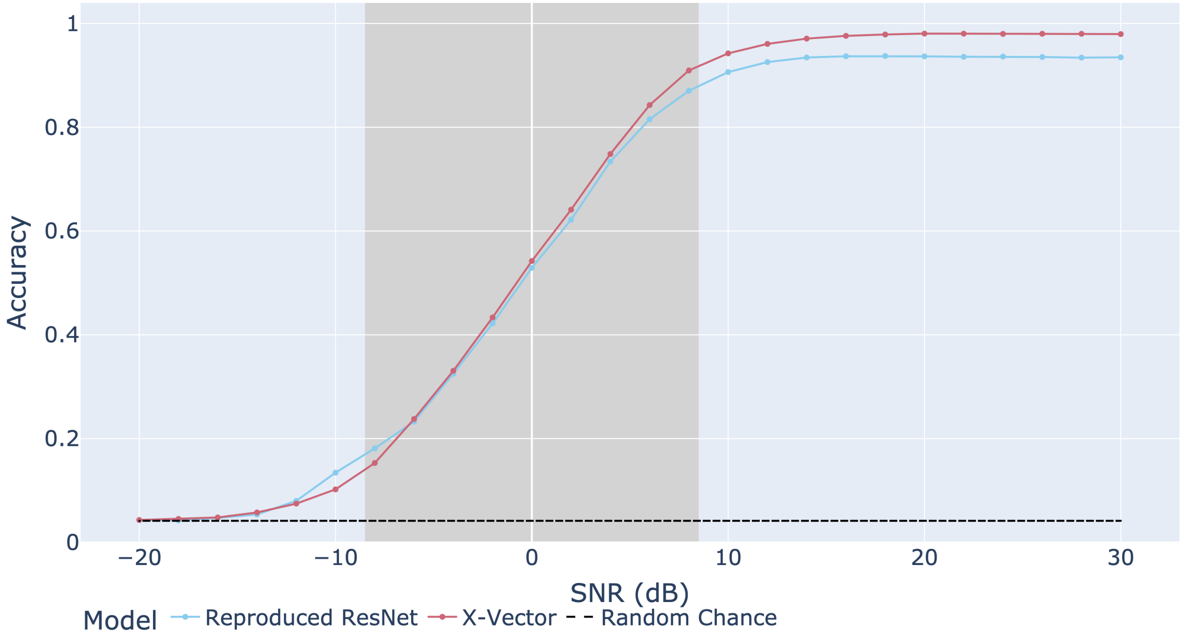


Figure 4. Accuracy comparison of the ResNet reproduced in [1] and the X-Vector-inspired model from [7] over varying SNRs. This accuracy comparison shows the superior performance of the X-Vector architecture, especially at higher SNRs, and supports using this architecture as a baseline for the improvements investigated in this paper.

As expected, the classifiers perform with a higher accuracy as the SNR value increases. At low SNR values, the classification task becomes more difficult due to the increased presence of noise. High SNR values are not invariably guaranteed in software-defined radios. However, notable enhancements are evident when compared to random chance, even under conditions of diminished SNR. In time-critical classification scenarios, this factor gains heightened significance, potentially leading to a pivotal advantage, as fewer demodulation schemes would require trial-and-error application to ascertain the correct scheme, thus streamlining the process.

One challenge of AMC is that it is desirable for performance to work well across a large range of SNRs. For instance, Figure 4 illustrates that modulation classification performance reached a plateau beyond +8 dB SNR, and approached chance-level classification performance when the SNR dipped below −8 dB on the RadioML 2018.01A dataset. This range is denoted by the shaded region. Harper et al. investigated methods to improve classification performance in this range by employing an SNR regression model to aid separate modulation classifiers (MCs). While other works have trained models to be robust across diverse SNR scenarios, Harper et al. employed SNR-specific MCs [3].

Six MCs were created by discretizing the SNR range to ameliorate performance between −8 dB and +8 dB SNR. These groupings were chosen in order to provide sufficient training data to avoid overfitting the MCs and provide enough resolution, so that combining MCs provided more value than a single classifier.

Firstly, by predicting the SNR of the received signal with a regression model, an SNR-specific MC that was trained on signals with the predicted SNR is applied to make the final prediction. While the dataset's SNR values are discretized, the SNR is measured on a continuous scale in practical deployment scenarios, subject to temporal fluctuations. Consequently, a regression approach is adopted instead of classification. By employing this methodology, various classifiers can tailor their feature processing to accommodate distinct SNR ranges. Each MC in this approach uses the same architecture as that proposed in [7]; however, each MC is trained with signals within each MC's SNR training range (see Table 1).

Table 1. SNR-specific modulation classifiers (MCs) groupings during training and inference phases, adapted from [3].

AMC Model	Training Range (dB)	Employed during Inference (dB)
MC 1	$[-20, -8]$	$(-\infty, -8)$
MC 2	$[-8, -4]$	$[-8, -4)$
MC 3	$[-4, 0]$	$[-4, 0)$
MC 4	$[0, 4]$	$[0, 4)$
MC 5	$[4, 8]$	$[4, 8)$
MC 6	$[8, 30]$	$[8, \infty)$

Illustrating enhancements across diverse SNR levels, Figure 5 presents the performance improvement (expressed as percentage accuracy) achieved through the employment of the SNR-informed architecture, contrasted with the baseline classification architecture detailed in [7]. While a marginal decline in performance was evident at −8 dB and a more substantial reduction at −2 dB, discernible enhancement is observable across most SNR conditions, with a pronounced emphasis on the desired range, spanning from −8 dB to +8 dB.

Declined performance at specific SNRs could be attributed to the optimization of a specific modulation classifier (MC), which led to an enhanced performance for a specific SNR grouping at the cost of lower performance for an individual value within the same group. To elaborate, the MC designed for the $[-4, 0)$ dB range may have bolstered the overall performance by accurately classifying signals at −4 dB and 0 dB, potentially at the expense of −2 dB accuracy. Given the substantial size of the testing dataset, these marginal percentage gains hold significance, as they result in thousands of additional correct classifications. Importantly, all outcomes achieve statistical significance according to McNemar's test [40], consequently achieving new state-of-the-art performance at the time.

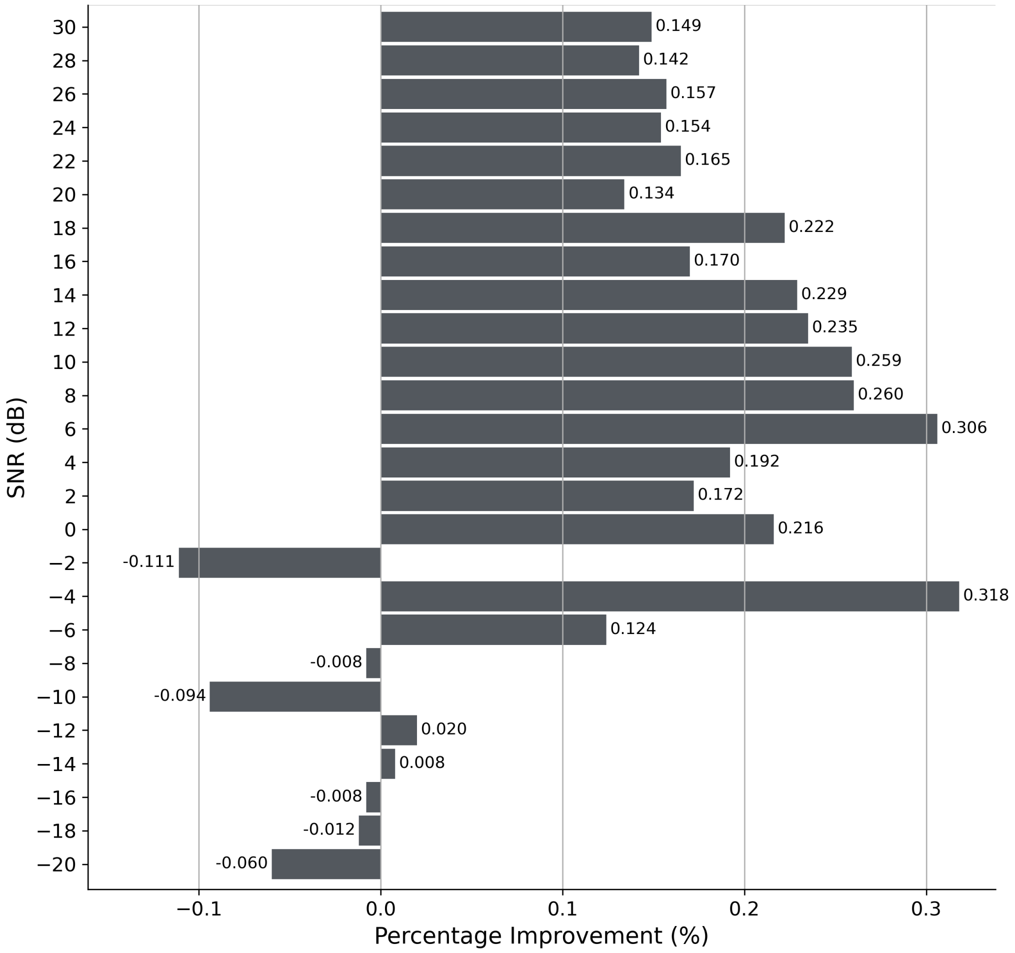


Figure 5. Summary of percentage improvement in accuracy over [7] seen in [3]. This work showed how the baseline architecture could be tuned to specific SNR ranges. Positive improvement is observed for most SNR ranges.

Soltani et al. found that SNR regions of $[-10, -2]$ dB, $[0, 8]$ dB, and $[10, 30]$ dB had similar classification patterns [4]. Instead of predicting exact modulation variants, the authors grouped commonly confused variants into a more generic, coarse-grained label. This grouping increases the performance of AMC by combining modulation variants that are commonly confused. However, it also decreases the sensitivity of the model to the numerous possible variants.

Cai et al. utilized a transformer-based architecture to aid performance at low SNR levels with relatively few training parameters (approximately 265,000 parameters) [41]. Ren et al. proposed *ResSwinT-SwinT*, making use of transformers to denoise signals under low SNR conditions prior to classification [17]. A multi-scale network along with center loss [42] was used in [43]. It was found that larger kernel sizes improved AMC performance. We further explore kernel size performance impacts in this work. Zhang et al. proposed a high-order attention mechanism using the covariance matrix achieving a maximum accuracy of 95.49% [44].

Although many discussed works use the same RadioML 2018.01A dataset, there is a lack of a uniform dataset split to establish a benchmark for papers to report performance. In an effort to make AMC more reproducible and comparable across publications, we have made our dataset split and accompanying code available on GitHub (https://github.com/caharper/Automatic-Modulation-Classification-with-Deep-Neural-Networks).

While numerous works have investigated architectural improvements, we aim to improve upon these works by introducing additional modifications, as well as a comprehensive ablation study that illustrates the improvement of each modification. With the new modifications, we achieve new state-of-the-art AMC performance.

3. Dataset

In order to assess different machine learning architectures, we employ the RadioML 2018.01A dataset, which encompasses a collection of 24 distinct modulation types [1,22]. Due to the complexity and variety of modulation schemes in the dataset, it is fairly representative of typically encountered modulation schemes. Moreover, this variety increases the likelihood that AMC models will generalize to more exotic or non-existing modulation schemes in the training data that are derived from these traditional variants.

There are a total of 2.56 million labeled signals, $S(T)$, each consisting of 1024 time domain digitized intermediate frequency (IF) samples of in-phase (I) and quadrature (Q) signal components where $S(T) = I(T) + jQ(T)$. The data were collected at a 900 MHz IF with an assumed sampling rate of 1MS/sec, such that each 1024 time domain digitized I/Q sample is 1.024 ms [8]. The 24 modulation types and the representative groups that we chose for each are listed as follows:

- **Amplitude**: OOK, 4ASK, 8ASK, AM-SSB-SC, AM-SSB-WC, AM-DSB-WC, and AM-DSB-SC.
- **Phase**: BPSK, QPSK, 8PSK, 16PSK, 32PSK, and OQPSK.
- **Amplitude and Phase**: 16APSK, 32APSK, 64APSK, 128APSK, 16QAM, 32QAM, 64QAM, 128QAM, and 256QAM.
- **Frequency**: FM and GMSK.

Each modulation type has a total of 106,496 observations ranging from -20 dB to $+30$ dB SNR in 2 dB increments. In total, there are 26 different SNR values. The SNR is assumed to be consistent over the same window length as the I/Q sample window.

The dataset was partitioned into 1 million training observations and 1.5 million testing observations through a random shuffle split, as carried out in [3,7]. This division was performed in a stratified manner, taking into account modulation type and the SNR. As a result of this balanced approach, the anticipated performance for a classifier employing random chance is 1/24 or approximately 4.2%. Considering the dataset's incorporation of diverse SNR levels, it is reasonable to expect that the classifier's accuracy would increase with the SNR. For consistency, each model investigated in this work was trained and evaluated on the same train and test set splits.

4. Initial Investigation

In this work, we use the architecture described in [7] as the baseline architecture. We note that [3] improved upon the baseline; however, each individual MC used the baseline architecture, except each is trained on specific SNR ranges. Therefore, the base architectural elements were similar to [7], but separated for different SNRs. In this work, our focus is to improve upon the employed CNN architecture for an individual MC rather than the use of several MCs. Therefore, we use the architecture from [7] as our baseline.

Before exploring an ablation study, we make a few notable changes from the baseline architecture in an effort to increase AMC performance. This initial exploration is for clarity as it reserves the ablation study that follows from requiring an inordinate number of models. It also introduces the general training procedures that assist and orient the reader in following the ablation study—the ablation study mirrors these procedures. We first provide an initial investigation exploring these notable changes.

We train each model using the Adam optimizer [45] with an initial learning rate lr = 0.0001, and a decay factor of 0.1, if the validation loss does not decrease for 12 epochs, and a minimum learning rate of 1×10^{-7}. If the validation loss does not decrease after 20 epochs, training is terminated and the models are deemed converged. For all experiments, mini-batches of size 32 are used. As has been established in most programming packages for neural networks, we refer to fully connected neural network layers as *dense* layers, which are typically followed by an activation function.

4.1. Architectural Changes

A common property of neural networks is using fewer but larger kernels in the early layers of the network, and an increased number of smaller kernels is used in the later layers, compared to the baseline architecture. This is commonly referred to as the information distillation pipeline [46]. By utilizing a smaller number of large kernels in early layers, we are able to increase the temporal context of the convolutional features without dramatically increasing the number of trainable parameters. Numerous, but smaller kernels are used in later convolutional layers to create more abstract features. Configuring the network in this manner is especially popular in image classification, where later layers represent more abstract, class-specific features.

We investigate this modification in three stages, using the baseline architecture described in Figure 3 [7]. We denote the number of filters in the network and the filter sizes as $F = [f_1, f_2, ..., f_7]$ and $K = [k_1, k_2, ...k_7]$ in Figure 3. The baseline architecture used $f = 64$ (for all layers) and $k = 3$ (consistent kernel size for all layers). Our first modification to the baseline architecture is $F = [32, 48, 64, 72, 84, 96, 108]$, but keeping $k = 3$ for all layers. Second, we use the baseline architecture, but change the size of filters in the network where $f = 64$ (the same as the baseline) and $K = [7, 5, 7, 5, 3, 3, 3]$. Third, we make both modifications and compare the result to the baseline model where $F = [32, 48, 64, 72, 84, 96, 108]$ and $K = [7, 5, 7, 5, 3, 3, 3]$. These modifications are not exhaustive searches; rather, these modifications are meant to guide future changes to the network by understanding the influence of filter quantity and filter size in a limited context.

4.2. Initial Investigation Results

As shown in Table 2, increasing the size of the filters in earlier layers increases both average and maximum test accuracy over [7], but at the cost of additional parameters. A possible explanation for the increase in performance is the increase in temporal context due to the larger kernel sizes. Increasing the number of filters without increasing temporal context decreases performance. This is possibly because it increases the complexity of the model without adding additional signal context.

Table 2. Initial investigation performance overview. All architectures employ the baseline with varying numbers of kernels and kernel sizes.

Notes	# Params	Avg. Accuracy	Max Accuracy
Reproduced ResNet [1]	165,144	59.2%	93.7%
X-Vector [7]	110,680	61.3%	98.0%
More Filters (Same Filter Sizes)	149,168	61.0%	96.1%
Larger Filter Sizes (Same # Filters)	143,960	62.6%	98.2%
Combined	174,000	62.9%	98.6%

Figure 6 illustrates the change in accuracy with varying SNR. The combined model, utilizing various kernel sizes and numbers of filters, consistently outperforms the other architectures across changing SNR conditions.

Although increasing the number of filters decreases performance alone, combining the approach with larger kernel sizes yields the best performance in our initial investigation. Increasing the temporal context may have allowed additional filters to better characterize the input signal. Because increased temporal context improves AMC performance, we are inspired to investigate additional methods, such as squeeze-and-excitation blocks and dilated convolutions, that can increase global and local context [34,47].

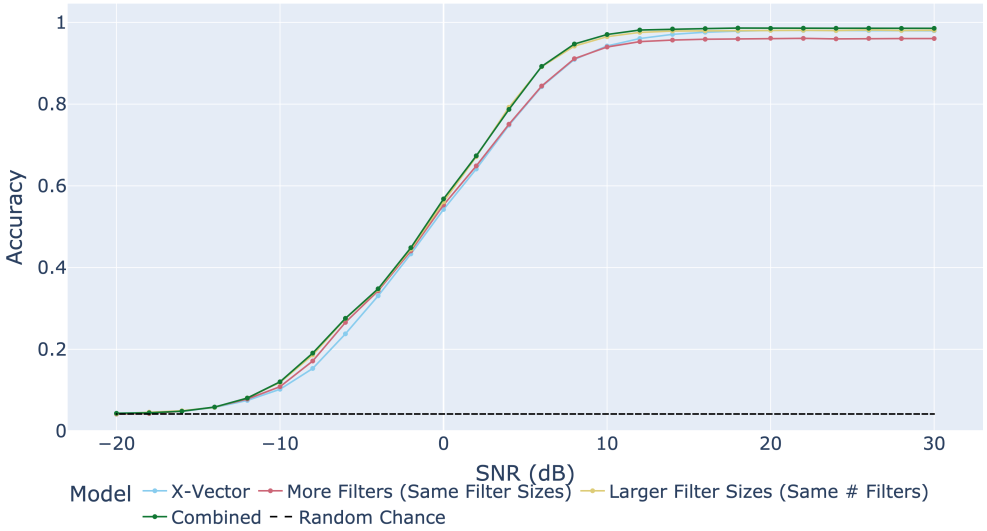


Figure 6. SNR vs. accuracy comparison of the initial investigation using the X-Vector baseline architecture [7]. Noticeable improvements can be observed across all SNRs.

5. Ablation Study Architecture Background

Building upon our findings from our initial investigation, we make additional modifications to the baseline architecture. For the MCs, we introduce dilated convolutions, squeeze-and-excitation blocks, self-attention, and other architectural changes. We also investigate various kernel sizes and the quantity of kernels employed from the initial investigation. Our goal is to improve upon existing architectures while investigating the impact of each modification on classification accuracy through an ablation study. In this section, we describe each modification performed.

5.1. Squeeze-and-Excitation Networks

Squeeze-and-excitation (SE) blocks introduce a channel-wise attention mechanism, first proposed in [34]. Due to the limited receptive field of each convolutional filter, SE blocks propose a recalibration step based on global statistics across channels (average pooling) to provide global context. Although initially utilized for image classification tasks [34,48,49], we argue the use of SE blocks can provide meaningful global context to the convolutional network used for AMC over the time domain.

Figure 7 depicts an SE block. The squeeze operation is defined as temporal global average pooling across convolutional filters. For an individual channel, c, the squeeze operation is defined as:

$$z_c = F_{sq}(x_c) = \frac{1}{T}\sum_{i=1}^{T} x_{i,c} \tag{1}$$

where $X \in \mathbb{R}^{T\times C} = [x_1, x_2, ..., x_C]$, $Z \in \mathbb{R}^{1\times C} = [z_1, z_2, ..., z_C]$, T is the number of samples in time, and C is the total number of channels. To model nonlinear interactions between channel-wise statistics, Z is fed into a series of dense layers followed by nonlinear activation functions:

$$s = F_{ex}(z, W) = \sigma(g(z, W)) = \sigma(W_2\delta(W_1 z)) \tag{2}$$

where δ is the rectified linear (ReLU) activation function, $W_1 \in \mathbb{R}^{\frac{C}{r}\times C}$, $W_2 \in \mathbb{R}^{C\times\frac{C}{r}}$, r is a dimensionality reduction ratio, and σ is the sigmoid activation function. The sigmoid function is chosen, as opposed to the softmax function, so that multiple channels can be accentuated and are not mutually exclusive. That is, the normalization term in the softmax can cause dependencies among channels, so the sigmoid activation is preferred. W_1 imposes a bottleneck to improve generalization performance and reduce parameter counts, while W_2 increases the dimensionality back to the original number of channels for the recalibration operation. In our work, we use $r = 2$ for all SE blocks to ensure a reasonable number of trainable parameters without over-squashing the embedding size.

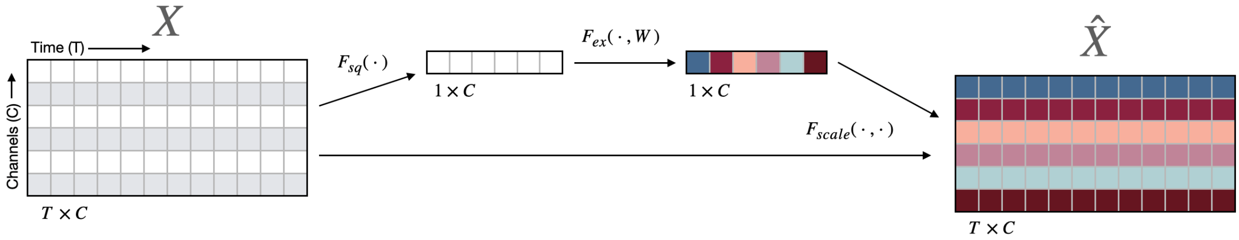


Figure 7. Squeeze-and-excitation block proposed in [34]. One SE block is shown applied to a single layer convolutional output activation. Two paths are shown: a scaling path and an identity path. The scaling vector is applied across channels to the identity path of the activations.

The final operation in the SE block, scaling or recalibration, is obtained by scaling the the input X by s:

$$\hat{x_c} = F_{scale}(x_c, s_c) = s_c x_c \tag{3}$$

where $\hat{X} \in \mathbb{R}^{T \times C} = [\hat{x_1}, \hat{x_2}, ..., \hat{x_C}]$.

5.2. Dilated Convolutions

As proposed in [47], Figure 8 depicts dilated convolutions, where the convolutional kernels are denoted by the colored components. In a traditional convolution, the dilation rate is equal to 1. Dilated convolutions build temporal context by increasing the receptive field of the convolutional kernels without increasing parameter counts, as the number of entries in the kernel remains the same.

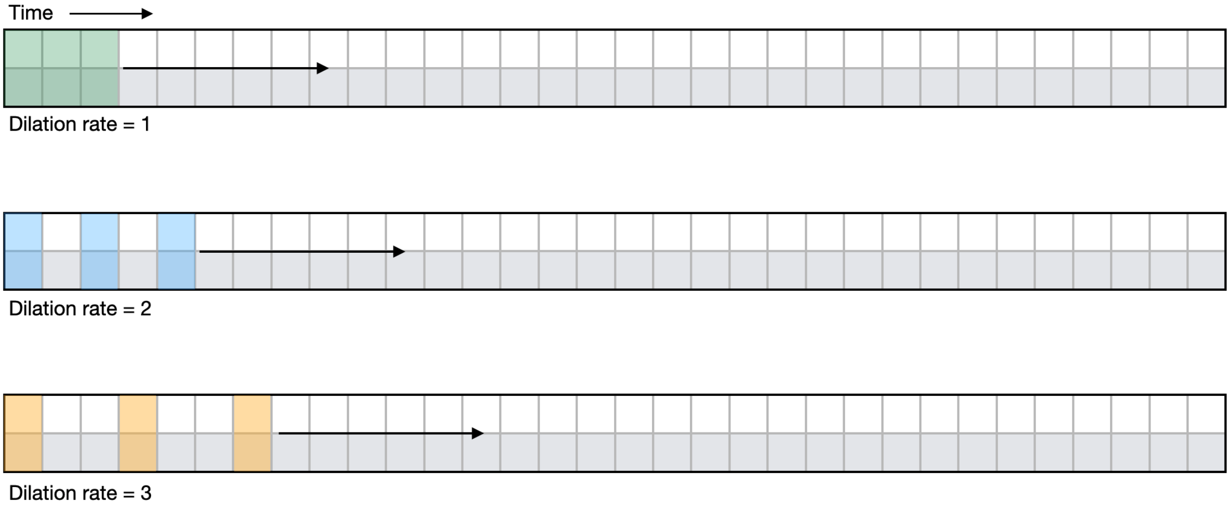


Figure 8. Dilated convolutions diagram. The top shows a traditional kernel applied to sequential time series points. The middle and bottom diagrams illustrate dilation rates of two and three, respectively. These dilations serve to increase the receptive field of the filter without increasing the number of trainable variables in the kernel.

Also, dilated convolutions do not downsample the signals like strided convolutions. Instead, the output of a dilated convolution can be the exact size of the input after properly handling edge effects at the beginning and end of the signal.

5.3. Final Convolutional Activation

We also investigate the impact of using an activation function (ReLU) after the last convolutional layer, just before statistics pooling. Because ReLU transforms the input sequence to be non-negative, the distribution characterized by the pooling statistics may become skewed. In [3,7], no activation was applied after the final convolutional layer, as shown in Figure 3. We investigate if this transformation impacts classification performance.

5.4. Self-Attention

Self-attention allows the convolutional outputs to interact with one another, enabling the network to learn to focus on important outputs. Self-attention before statistics pooling essentially creates a weighted summation over the convolutional outputs, weighting their importance similarly to [50–52].

We use the attention mechanism described by Vaswani et al. in [53], where each output element is a weighted sum of the linearly transformed input, where the dimensionality of K is d_k, as seen in Equation (4).

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{|\sqrt{d_k}|}\right)V \tag{4}$$

In the case of self-attention, Q, K, and V are equal. A scaling factor of $\frac{1}{|\sqrt{d_k}|}$ is applied to counteract vanishing gradients in the softmax output when d_k is large.

6. Ablation Study Architecture

Applying the specified modifications to the architecture in [7], Figure 9 illustrates the proposed architecture with every modification included in the graphic. Each colored block represents an optional change to the architecture that will be investigated in the ablation study. That is, each combination of network modifications is analyzed to aid understanding of each modification's impact on the network.

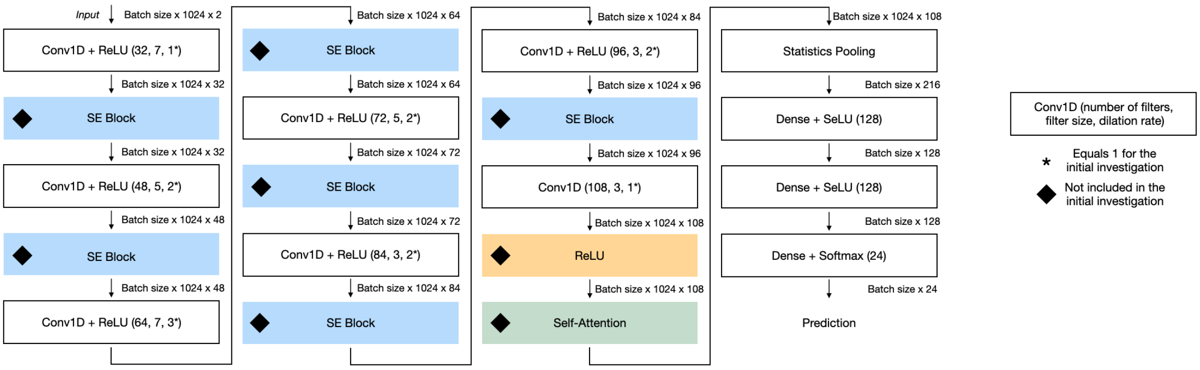


Figure 9. Proposed architecture with modifications including SENets, dilated convolutions, optional ReLU activation before statistics pooling, and self-attention. The output tensor sizes are also shown for each unit in the diagram. * denotes where the sizes differ from the baseline architecture.

Each convolutional layer has the following parameters: number of filters, kernel size, and dilation rate. The asterisk next to each dilation rate represents the changing of dilation rates in the ablation study. If dilated convolutions are used, then the dilation rate value in the graphic is used. If dilated convolutions are not used, each dilation rate is set to 1. That is, a traditional convolution is applied. All convolutions use a stride of 1, and the same training procedure from the initial investigation is used.

7. Evaluation Metrics

We present several evaluation metrics to compare the different architectures considered in the ablation study. In this section, we will discuss each evaluation technique used in the results section.

Due to the varying levels of SNRs in the employed dataset, we plot classification accuracy over each true SNR value. This allows for a visualization of the trade-off in performance, as noise becomes more or less dominant in the received signals. Additionally, we report average accuracy and maximum accuracy across the entire test set for each model. While we note that average accuracy is not indicative of the model's performance, as accuracy is highly correlated to the SNR of the input signal, we share this result to give other researchers the ability to reproduce and compare works.

As discussed in [35], AMC is often implemented on resource-constrained devices. In these systems, using larger models in terms of parameter counts may not be feasible. We report the number of parameters for each model in the ablation study to examine the trade-off in AMC performance and model size.

Additional analyses are also carried out. However, due to the large number of models investigated in this study, we will select the best-performing model from the ablation study for brevity and analyze the performance of this model in greater detail. For example, confusion matrices for the best-performing model from the ablation study are provided to show common misclassifications for each modulation type. Additionally, there exist several use-cases where relatively short signal bursts are received. For example, a wide-band scanning receiver may only detect a short signal burst. Therefore, signal duration in the time domain versus AMC performance is investigated to determine the robustness of the best-performing model when short signal bursts are received.

8. Ablation Results

8.1. Overall Performance

Table 3 lists the maximum and average accuracy performance for each model in the ablation study. A binary naming convention is used to indicate the various methods used for each architecture. Similarly to the result found in Section 4, increasing the temporal context typically results in increased performance. Models that incorporate dilated convolutions tended to have higher average accuracies than models without dilated convolutions.

The best-performing model, in terms of average accuracy across all SNR conditions, included SE blocks, dilated convolutions, and a ReLU activation, prior to statistics pooling (model 1110), with an average accuracy of approximately 63.7%. This model also achieved the highest maximum accuracy of about 98.9% at a 22 dB level. Both values achieve new state-of-the-art performance on the RadioML 2018.01A dataset. In terms of overall accuracy, model 1110 outperforms the results reported in prior work [1,3,7,37,54] (all between 52.47% and 61.3%) and all other models investigated in this work. In terms of peak accuracy, model 1110 outperforms the methods proposed in [1,3,4,7,37,39,41,43,44,54]—each with a reported peak accuracy between 80% and 98%.

SE blocks did not increase performance compared to model 0000, with the exception of models 1110 and 1111. However, SE blocks were incorporated in the best-performing model, 1110. Self-attention was not found to aid classification performance in general with the proposed architecture. Self-attention introduces a large number of trainable parameters, possibly forming a complex loss space.

Table 4 lists the performances of single modification (from baseline) architectures. Each component of the ablation study, with the exception of dilated convolutions, decreased performance when applied individually. When combined, however, the best-performing model was found. Therefore, we conclude that each component could possibly aid the optimization of each other—and, in general, dilated convolutions tend to have the most dramatic performance increases.

Table 3. Ablation study performance overview.

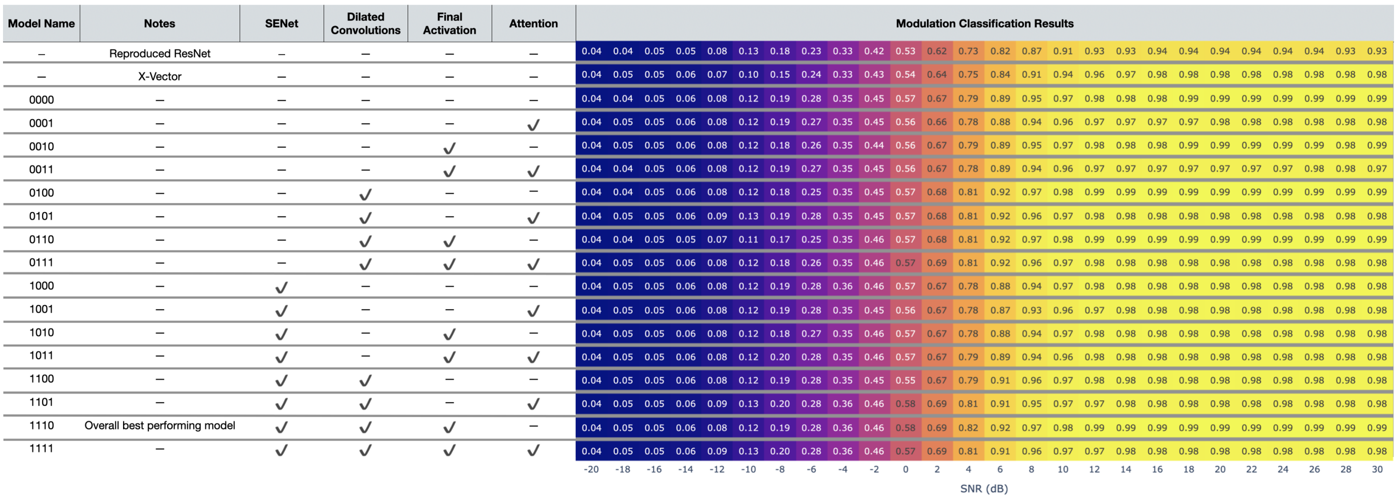
Model Name	Notes	SENet	Dilated Convolutions	Final Activation	Attention	# Params	Avg. Accuracy	Max Accuracy
—	Reproduced ResNet [1]	—	—	—	—	165,144	59.2%	93.7%
—	X-Vector [7]	—	—	—	—	110,680	61.3%	98.0%
0000	Best-performing model from the initial investigation	—	—	—	—	174,000	62.9%	98.6%
0001		—	—	—	✓	221,088	62.3%	97.6%
0010		—	—	✓	—	174,000	62.8%	98.6%
0011		—	—	✓	✓	221,088	62.3%	97.5%
0100		—	✓	—	—	174,000	63.2%	98.9%
0101		—	✓	—	✓	221,088	63.1%	97.9%
0110		—	✓	✓	—	174,000	63.2%	98.9%
0111		—	✓	✓	✓	221,088	63.0%	98.0%
1000		✓	—	—	—	202,880	62.9%	98.5%
1001		✓	—	—	✓	249,968	62.6%	98.2%
1010		✓	—	✓	—	202,880	62.6%	98.3%
1011		✓	—	✓	✓	249,968	62.8%	98.1%
1100		✓	✓	—	—	202,880	62.8%	98.2%
1101		✓	✓	—	✓	249,968	63.0%	97.7%
1110	Overall best performing model	✓	✓	✓	—	202,880	63.7%	98.9%
1111		✓	✓	✓	✓	249,968	63.0%	97.8%

Table 4. Individual network modification performance overview. Entries are repeated from Table 3 for clarity.

Model Name	Notes	SENet	Dilated Convolutions	Final Activation	Attention	# Params	Avg. Accuracy	Max Accuracy
—	X-Vector [7]	—	—	—	—	110,680	61.3%	98.0%
0000		—	—	—	—	174,000	62.9%	98.6%
0001		—	—	—	✓	221,088	62.3%	97.6%
0010		—	—	✓	—	174,000	62.8%	98.6%
0100		—	✓	—	—	174,000	63.2%	98.9%
1000		✓	—	—	—	202,880	62.9%	98.5%
1110	Best-performer	✓	✓	✓	—	202,880	63.7%	98.9%

8.2. Accuracy over Varying SNR

Figure 10 summarizes the ablation study in terms of classification accuracy over varying SNR levels. We add this figure for completeness and reproducibility for other researchers. The accuracy within each SNR band is shown along with the modifications used, similar to Table 3. The coloring in the figure denotes the accuracy in each SNR band. The performance follows a trend similar to that of a sigmoid function, where the rate at which peak classification accuracy is achieved is the most distinguishing feature between the different models. With the improved architectures, a maximum of 99% accuracy is achieved at high SNR levels (starting around 12 dB SNR).



Model Name	Notes	SENet	Dilated Convolutions	Final Activation	Attention	Modulation Classification Results																									
–	Reproduced ResNet	–	–	–	–	0.04	0.04	0.05	0.05	0.08	0.13	0.18	0.23	0.33	0.42	0.53	0.62	0.73	0.82	0.87	0.91	0.93	0.93	0.94	0.94	0.94	0.94	0.94	0.94	0.93	0.93
–	X-Vector	–	–	–	–	0.04	0.05	0.05	0.06	0.07	0.10	0.15	0.24	0.33	0.43	0.54	0.64	0.75	0.84	0.91	0.94	0.96	0.97	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
0000	–	–	–	–	–	0.04	0.04	0.05	0.06	0.08	0.12	0.19	0.28	0.35	0.45	0.57	0.67	0.79	0.89	0.95	0.97	0.98	0.98	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99
0001	–	–	–	–	✓	0.04	0.05	0.05	0.06	0.08	0.12	0.19	0.27	0.35	0.45	0.56	0.66	0.78	0.88	0.94	0.96	0.97	0.97	0.97	0.97	0.98	0.98	0.98	0.98	0.98	0.98
0010	–	–	–	✓	–	0.04	0.05	0.05	0.06	0.08	0.12	0.18	0.26	0.35	0.44	0.56	0.67	0.79	0.89	0.95	0.97	0.98	0.98	0.98	0.99	0.99	0.99	0.99	0.99	0.98	0.99
0011	–	–	–	✓	✓	0.04	0.04	0.05	0.06	0.08	0.12	0.19	0.27	0.35	0.45	0.56	0.67	0.78	0.89	0.94	0.96	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.98	0.98	0.97
0100	–	–	✓	–	–	0.04	0.04	0.05	0.06	0.08	0.12	0.18	0.25	0.35	0.45	0.57	0.68	0.81	0.92	0.97	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
0101	–	–	✓	–	✓	0.04	0.05	0.05	0.06	0.09	0.13	0.19	0.28	0.35	0.45	0.57	0.68	0.81	0.92	0.96	0.97	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
0110	–	–	✓	✓	–	0.04	0.04	0.05	0.05	0.07	0.11	0.17	0.25	0.35	0.46	0.57	0.68	0.81	0.92	0.97	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
0111	–	–	✓	✓	✓	0.04	0.05	0.05	0.06	0.08	0.12	0.18	0.26	0.35	0.46	0.57	0.69	0.81	0.92	0.96	0.97	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
1000	–	✓	–	–	–	0.04	0.05	0.05	0.06	0.08	0.12	0.19	0.28	0.36	0.46	0.57	0.67	0.78	0.88	0.94	0.97	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
1001	–	✓	–	–	✓	0.04	0.05	0.05	0.06	0.08	0.12	0.19	0.28	0.35	0.45	0.56	0.67	0.78	0.87	0.93	0.96	0.97	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
1010	–	✓	–	✓	–	0.04	0.05	0.05	0.06	0.08	0.12	0.18	0.27	0.35	0.46	0.57	0.67	0.78	0.88	0.94	0.97	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
1011	–	✓	–	✓	✓	0.04	0.05	0.05	0.06	0.08	0.12	0.20	0.28	0.35	0.46	0.57	0.67	0.79	0.89	0.94	0.96	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
1100	–	✓	✓	–	–	0.04	0.05	0.05	0.06	0.08	0.12	0.19	0.28	0.35	0.45	0.55	0.67	0.79	0.91	0.96	0.97	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
1101	–	✓	✓	–	✓	0.04	0.05	0.05	0.06	0.09	0.13	0.20	0.28	0.36	0.46	0.58	0.69	0.81	0.91	0.95	0.97	0.97	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
1110	Overall best performing model	✓	✓	✓	–	0.04	0.05	0.05	0.06	0.08	0.12	0.19	0.28	0.36	0.46	0.58	0.69	0.82	0.92	0.97	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
1111	–	✓	✓	✓	✓	0.04	0.05	0.05	0.06	0.09	0.13	0.20	0.28	0.36	0.46	0.57	0.69	0.81	0.91	0.96	0.97	0.97	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98
						-20	-18	-16	-14	-12	-10	-8	-6	-4	-2	0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30

SNR (dB)

Figure 10. Ablation study results in terms of classification accuracy across SNR ranges. The reproduced ResNet [1] and X-Vector baseline [7] architectures are included. The best-performing model is in the second-to-last row and displays strong performance across SNR values.

While the proposed changes to the architectures generally improve performance at higher SNR levels, the largest improvements occur between −12 dB and 12 dB, compared to the baseline model in [7]. For example, at 4 dB, the performance increases from 75% up to 82%. Incorporating these modifications to the network may prove to be critical in real-world situations, where noisy signals are likely to be obtained. Improving AMC performance at lower SNR ranges (<−12 dB) is still an open research topic, with accuracies at near-chance level.

A receiver using model 1110 within its demodulator achieves a notable 91% classification accuracy at 6 dB SNR, which is an improvement compared to previous work [1], which achieved a similar accuracy of around 10 dB, and [7], which achieved a similar accuracy of around 8 dB. Wireless communications systems employed in various applications can suffer from poor reception in low SNR environments due to environmental conditions, such as complex channel characteristics, multipath interference, fading, and man-made conditions, such as congested channels, among other factors. Therefore, any improvement that can increase performance at low SNR is desirable. Because AMC can directly impact decision-making algorithms, in situations where reliable communications are essential, such as emergency response systems, military operations, or autonomous vehicle networks, the ability to accurately classify modulations under challenging SNR conditions becomes a pivotal determinant of system effectiveness and safety.

One observation is that the best-performing model can vary with the SNR. In systems that have available memory and processing power, an approach similar to [3] may be used to utilize several models and intelligently choose predictions based on estimated SNR conditions. That is, if the SNR of the signal of interest is known, a model can be tuned to increase performance slightly, as shown in [3]. Using the results presented here, researchers could also choose the architecture differences that perform best for a given SNR range (although performance differences are subtle).

8.3. Parameter Count Trade-Off

An overview of each model's complexity and overall performance across the entire testing set is shown in Table 3. This information is also shown graphically in Figure 11 for the maximum accuracy over SNR and the average accuracy across all SNRs. Whether looking at the maximum or the average measures of performance, the conclusions are similar. The previously described binary model name also appears in the figure. We found a slight correlation between the number of model parameters and overall model performance; however, with the architectures explored, there was a general parameter count where performance peaked. Models with parameter counts between approximately 170 k and 205 k generally performed better than smaller and larger models. We note that the models with more than 205 k parameters included self-attention, which was found to decrease model performance with the proposed architectures. This implies that one possible reason self-attention did not perform as well as other modifications is because of the increase in parameters, resulting in a more difficult loss space, from which to optimize.

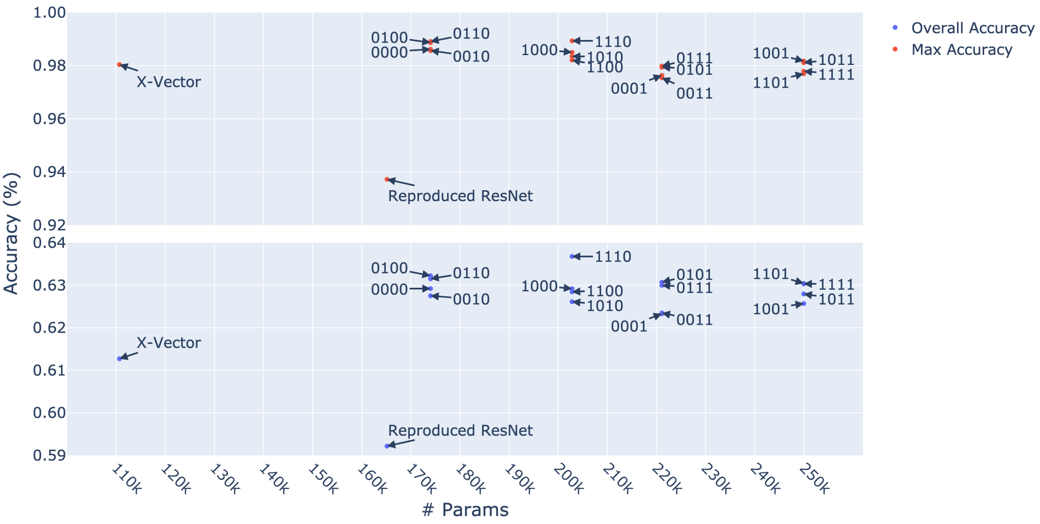


Figure 11. Ablation study parameter count trade-off including the reproduced ResNet [1] and X-Vector baseline [7]. The x-axis shows the number of trainable variables in each model and the y-axis shows max or average accuracy. The callout for each point denotes the model name, as shown in Table 3.

9. Best-Performing Model Investigation

Due to the large volume of models, we focus upon the best-performing model, model 1110, for the remainder of this work. As previously mentioned, this model employs all modifications except self-attention.

9.1. Top-K Accuracy

As discussed, in systems where the modulation schemes must be classified quickly, it is advantageous to apply fewer demodulation schemes in a trial-and-error fashion. This is particularly significant at lower SNR values, where accuracy is mediocre. Top-k accuracy allows an in-depth view of the expected number of trials before finding the correct modulation scheme. Although traditional accuracy (top-1 accuracy) characterizes the performance of the model in terms of classifying the exact variant, top-k accuracy characterizes the percentage of the classifier predicting the correct variant among the top-k predictions (sorted by descending class probabilities). We plot the top-1, top-2, and top-5 classification accuracy over varying SNR conditions for each modulation grouping, as defined in Section 3 in Figure 12.

Although performance decays to approximately random chance for the overall (all modulation schemes) performance curves for each top-k accuracy, it is notable that some modulation group performances drop below random chance. The models are trained to maximize the overall model performance. This could explain why certain modulation groups dip below random chance but the overall performance and other modulation groups remain at or above random chance.

Using the proposed method greatly reduces the correct modulation scheme search space. While high performance in top-1 accuracy is increasingly difficult to achieve with low

SNR signals, top-2 and top-5 accuracies converge to higher values at a much faster rate. This indicates that our proposed method greatly reduces the search space from 24 modulation candidates to fewer candidate types when employing trial-and-error methods to determine the correct modulation scheme. Further, if the group of modulation is known (e.g., FM), one can view a more specific trade-off curve in terms of SNR and top-k accuracy, as given in Figure 12.

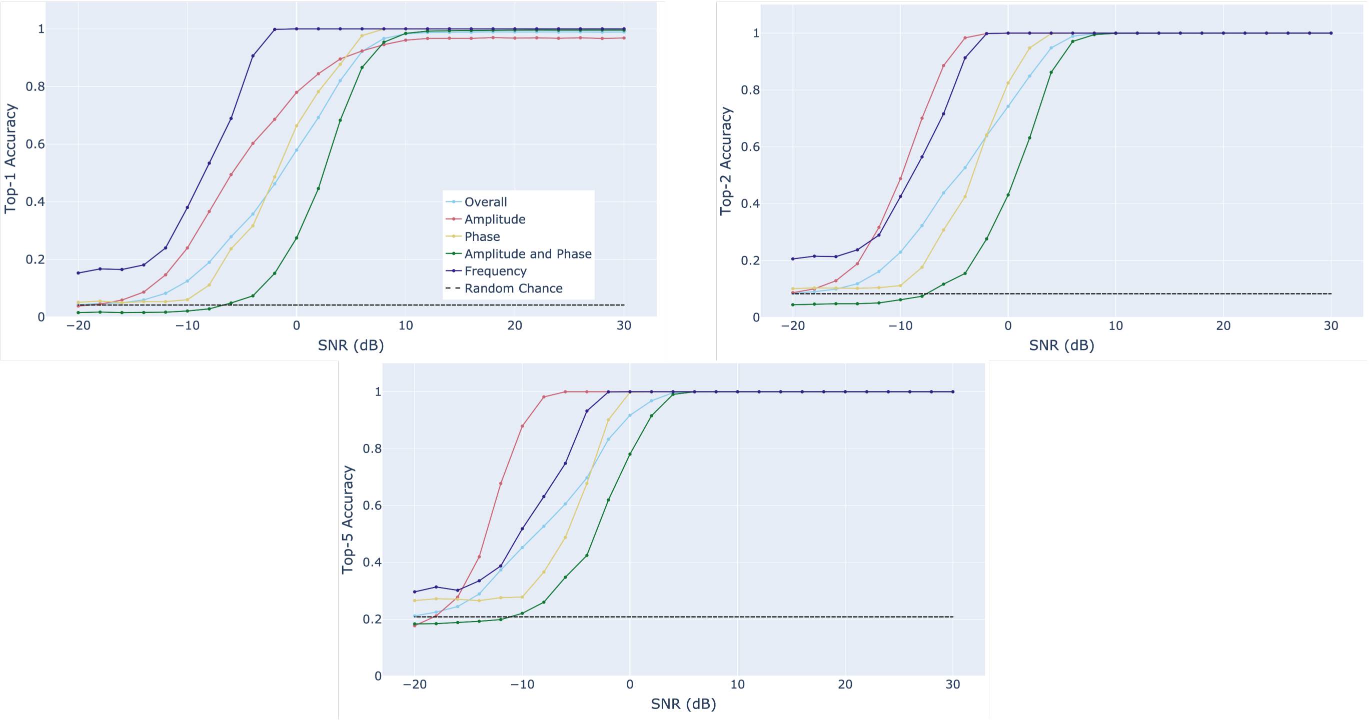


Figure 12. Top-1 (**top left**), top-2 (**top right**), and top-5 (**bottom**) accuracy over varying SNR conditions for model 1110. Random chance for each is defined as 1/24, 2/24, and 5/24, respectively.

9.2. Short-Duration Signal Bursts

Due to the rapid scanning characteristic of some modern software-defined radios, we investigate the performance trade-off of varying signal duration and AMC performance. This analysis is meant to emulate the situation wherein a receiver only detects a short RF signal burst. We investigate signal burst durations of 1.024 ms (full length signal from original dataset), 512 µs, 256 µs, 128 µs, 64 µs, 32 µs, and 16 µs. We assume the same 1 MS/sec sampling rate, as in the previous analyses, such that the 16 µs burst is captured in 16 I/Q samples.

In this section, we use the same test set as our other investigations; however, a uniformly random starting point is determined for each signal such that a contiguous sample of the desired duration, starting at the random point, is chosen. Thus, the chosen segment from a test set sample is randomly assigned.

We also note that, although the sample length for the evaluation is changed, the best-performing model is the same architecture with exactly the same trained weights, because this model uses statistics pooling from the X-Vector-inspired modification. A significant benefit of the X-Vector-inspired architecture is its ability to handle variable-length inputs without the need of padding, retraining, or other network modifications. This is achieved by taking global statistics across convolutional channels, producing a fixed-length vector, regardless of signal duration. Due to this flexibility, the same model (model 1110) weights are used for each duration experiment. This fact also emphasizes the desirability of using X-vector-inspired AMC architectures for receivers that are deployed in an environment where short-burst and variable duration signals are anticipated to be present.

For each signal duration in the time domain, we plot the overall classification accuracy over varying SNR conditions, as well as the accuracy for each modulation grouping defined in Section 3 in Figure 13, which demonstrates the trade-off for various signal durations, where n is the number of samples from the time domain I/Q signal. The first observation is, as we would expect, that classification performance degrades with decreased signal duration, similarly to [39]. For example, the maximum accuracy begins to degrade at 256 µs and is more noticeable at 128 µs. This is likely a result of using sample statistics that result in unstable or biased estimates for short signal lengths, since the number of received signal data points are insufficient to characterize the sample statistics used during training. Random classification accuracy is approximately 4% and is shown in the black dotted line in Figure 13. Although classification performance decreases with decreased duration, we are still able to achieve significantly higher classification accuracy than random chance, down to 16 µs of signal capture.

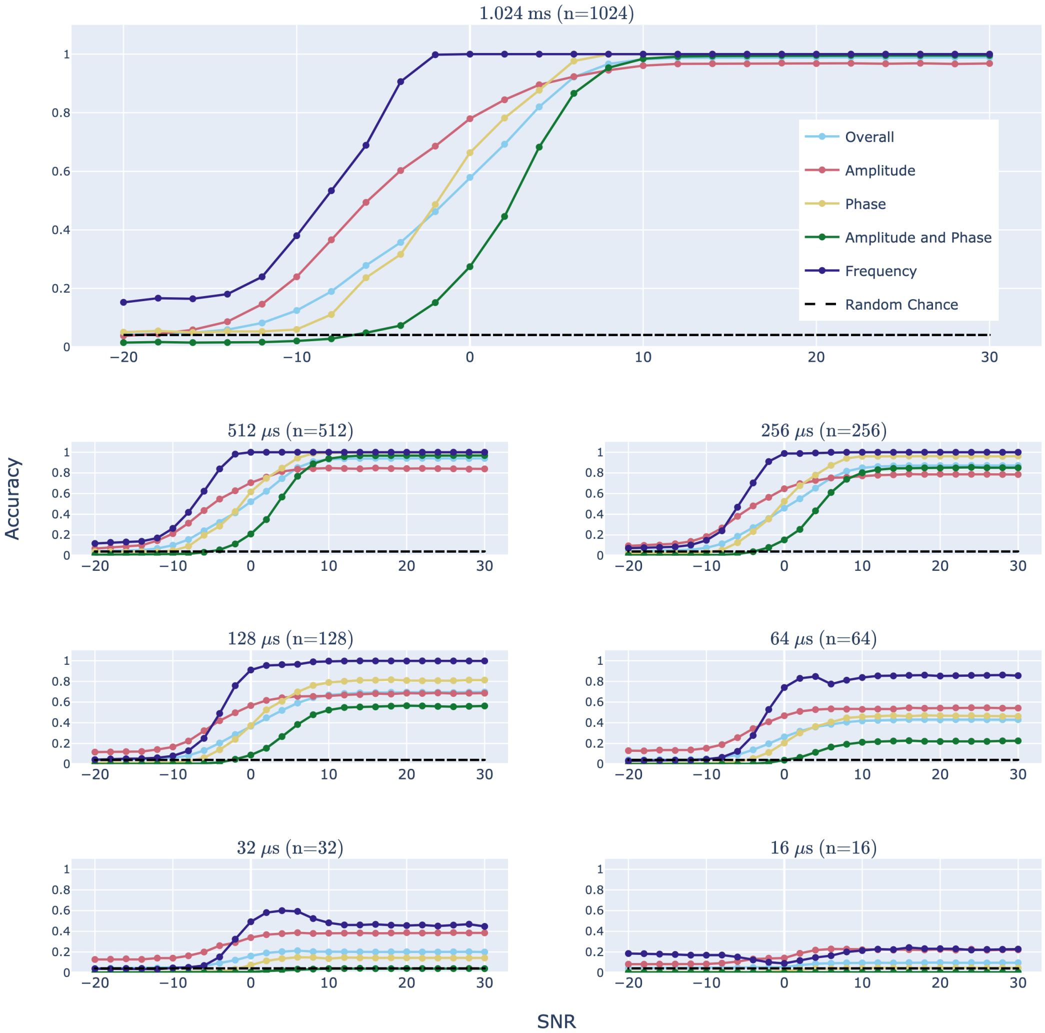


Figure 13. Trade-off in accuracy for various signal lengths across the SNR, grouped by modulation category for the best-performing model, 1110. The top plot shows the baseline performance using the full sequence. Subsequent plots show the same information using increasingly smaller signal lengths for classification.

FM (frequency modulation) signals were typically more resilient to noise interference than AM (amplitude modulation) and AM–PM (amplitude and phase modulation) signals in our AMC. This was observed across all signal burst durations and our top-k accuracy analysis. This behavior indicates that the performance of our AMC for short bursts, in the presence of increasing amounts of noise, is more robust for signals modulated by changes in the carrier frequency and is more sensitive to signals modulated by varying the carrier amplitude. We attribute this behavior to our AMC architecture, the architecture of the receiver, or a combination of both of the AMC and receiver.

9.3. Confusion Matrices

While classification accuracy provides a holistic view of model performance, it lacks the granularity to investigate where misclassifications are occurring. Confusion matrices are used to analyze the distribution of classifications for each given class. For each true label, the proportion of correctly classified samples is calculated along with the proportion of incorrect predictions for each opposing class. In this way, we can see which classes the model is struggling to distinguish from one another. A perfect classifier would be the identity matrix where the diagonal values represent the true class, and which match the predicted class. Each matrix value represents the percentage of classifications for the true label and each row sums to 1 (100%).

Figure 14 illustrates the class confusion matrices for SNR levels greater than or equal to 0 dB for models 1110, the reproduced ResNet architecture from [1], and the baseline X-Vector architecture from [7], respectively. Shown in [7], the X-Vector architecture was able to distinguish PSK and AM-SSB variants to a higher degree and performed better overall than [1]. Both architectures struggled to differentiate QAM variants.

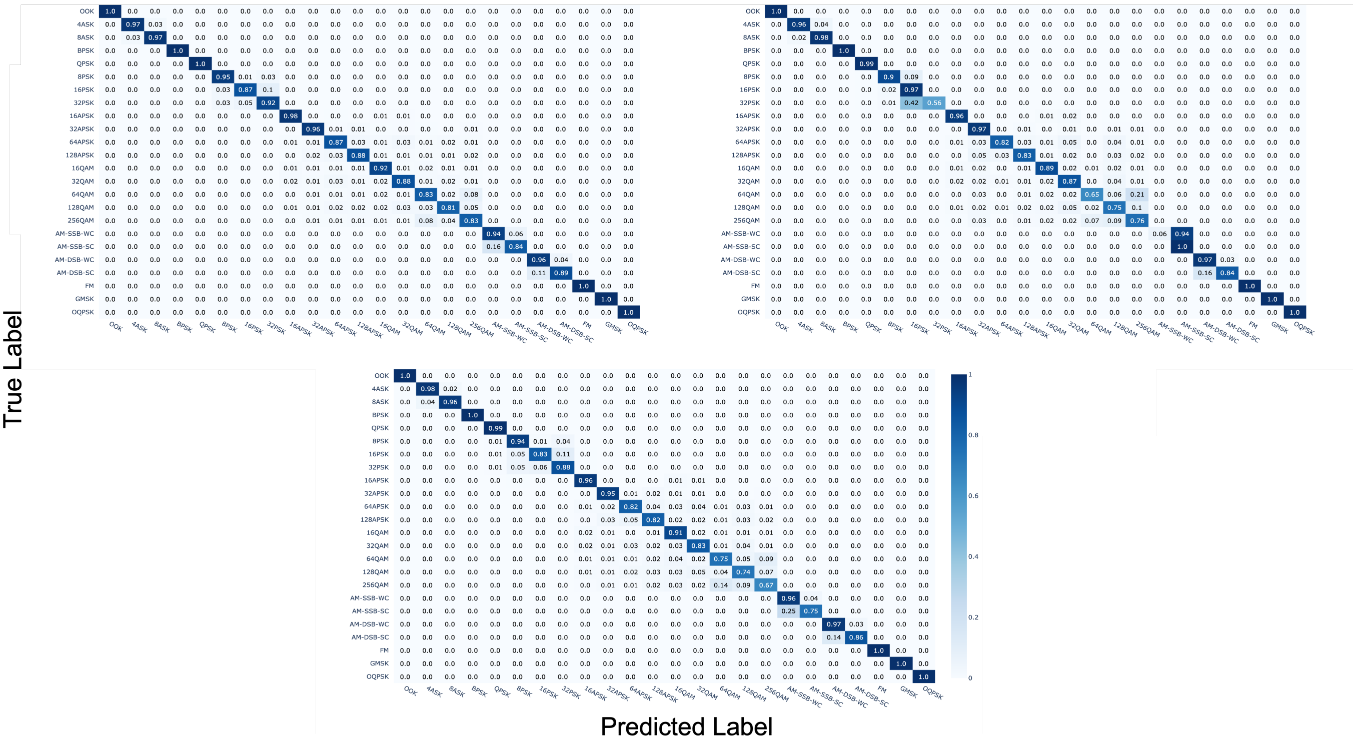


Figure 14. Confusion matrices for model 1110—the best-performing model from this work (**top left**), the reproduced ResNet model from [1] (**top right**), and the X-Vector-inspired model from [7] (**bottom**) with SNR ≥ 0 dB.

Model 1110 improved upon these prior results for QAM signals and, in general, has higher diagonal components than the other architectures. This, again, supports a conclusion that model 1110 achieves a new state of the art in AMC performance.

10. Conclusions

A comprehensive ablation study was carried out with regard to AMC architectural features using the extensive RadioML 2018.01A dataset. This ablation study built upon a strong performance of a new baseline model that was also introduced in the initial investigation of this study. This initial investigation informed the design of a number of AMC architecture modifications—specifically, the use of X-Vectors, dilated convolutions, and SE blocks. With the combined modifications, we achieved a new state of the art in AMC accuracy, improving upon prior work by approximately 2.5% overall accuracy on the RadioML 2018.01A dataset. We also achieve a new state of the art in peak performance with 98.9% accuracy at high SNR values. Among these modifications, dilated convolutions were found to be the most critical architectural feature for model performance. Self-attention was also investigated, but was not found to increase performance—although increased temporal context improved upon prior works. Additionally, the best-performing model was found to be robust against signals of varying duration, down to 128 µs of signal capture.

Author Contributions: Conceptualization, C.A.H.; Software, C.A.H.; Supervision, M.A.T. and E.C.L.; Visualization, C.A.H.; Writing—original draft, C.A.H.; Writing—review and editing, C.A.H., M.A.T. and E.C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: https://www.deepsig.ai/datasets (accessed on 1 February 2020).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. O'Shea, T.; Roy, T.; Clancy, T.C. Over-the-Air Deep Learning Based Radio Signal Classification. *IEEE J. Sel. Top. Signal Process.* **2018**, *12*, 168–179. [CrossRef]
2. Jacob, P.; Sirigina, R.P.; Madhukumar, A.S.; Prasad, V.A. Cognitive Radio for Aeronautical Communications: A Survey. *IEEE Access* **2016**, *4*, 3417–3443. [CrossRef]
3. Harper, C.A.; Sinha, A.; Thornton, M.A.; Larson, E.C. SNR-Boosted Automatic Modulation Classification. In Proceedings of the 2021 55th Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 31 October–3 November 2021; pp. 372–375.
4. Soltani, N.; Sankhe, K.; Ioannidis, S.; Jaisinghani, D.; Chowdhury, K. Spectrum awareness at the edge: Modulation classification using smartphones. In Proceedings of the 2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), Newark, NJ, USA, 11–14 November 2019; pp. 1–10.
5. Swami, A.; Sadler, B. Hierarchical Digital Modulation Classification using Cumulants. *IEEE Trans. Commun.* **2000**, *48*, 416–429. [CrossRef]
6. Abdelbar, M.; Tranter, W.H.; Bose, T. Cooperative Cumulants-Based Modulation Classification in Distributed Networks. *IEEE Trans. Cogn. Commun. Netw.* **2018**, *4*, 446–461. [CrossRef]
7. Harper, C.A.; Lyons, L.; Thornton, M.A.; Larson, E.C. Enhanced Automatic Modulation Classification Using Deep Convolutional Latent Space Pooling. In Proceedings of the 2020 54th Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 1–4 November 2020.
8. O'Shea, T.; Corgan, J.; Clancy, T. Convolutional Radio Modulation Recognition Networks. In Proceedings of the International Conference on Engineering Applications of Neural Networks, Aberdeen, UK, 2–5 September 2016; pp. 213–226.
9. Huynh-The, T.; Hua, C.H.; Kim, J.W.; Kim, S.H.; Kim, D.S. Exploiting a Low-Cost CNN with Skip Connection for Robust Automatic Modulation Classification. In Proceedings of the 2020 IEEE Wireless Communications and Networking Conference (WCNC), Seoul, Republic of Korea, 25–28 May 2020; pp. 1–6. [CrossRef]
10. Peng, S.; Jiang, H.; Wang, H.; Alwageed, H.; Yao, Y. Modulation Classification using Convolutional Neural Network Based Deep Learning Model. In Proceedings of the 2017 26th Wireless and Optical Communication Conference (WOCC), Newark, NJ, USA, 7–8 April 2017; pp. 1–5. [CrossRef]
11. Peng, S.; Jiang, H.; Wang, H.; Alwageed, H.; Zhou, Y.; Sebdani, M.M.; Yao, Y. Modulation Classification Based on Signal Constellation Diagrams and Deep Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 718–727. [CrossRef]
12. Wang, F.; Wang, Y.; Chen, X. Graphic Constellations and DBN Based Automatic Modulation Classification. In Proceedings of the 2017 IEEE 85th Vehicular Technology Conference (VTC Spring), Sydney, NSW, Australia, 4–7 June 2017; pp. 1–5. [CrossRef]
13. Dulek, B. A Sparse Approach for Identification of Signal Constellations Over Additive Noise Channels. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 817–822. [CrossRef]
14. Hassan, K.; Dayoub, I.; Hamouda, W.; Nzeza, C.N.; Berbineau, M. Blind Digital Modulation Identification for Spatially-Correlated MIMO Systems. *IEEE Trans. Wirel. Commun.* **2012**, *11*, 683–693. [CrossRef]
15. Abdelbar, M.; Tranter, B.; Bose, T. Cooperative Modulation Classification of Multiple Signals in Cognitive Radio Networks. In Proceedings of the 2014 IEEE International Conference on Communications (ICC), Sydney, NSW, Australia, 10–14 June 2014; pp. 1483–1488. . [CrossRef]

16. Zhang, Q.; Guan, Y.; Li, H.; Song, Z. Distributed Cooperative Automatic Modulation Classification Using DWA-ADMM in Wireless Communication Networks. *Electronics* **2023**, *12*, 3002. [CrossRef]
17. Ren, B.; Teh, K.C.; An, H.; Gunawan, E. Automatic Modulation Recognition of Dual-Component Radar Signals Using ResSwinT-SwinT Network. *IEEE Trans. Aerosp. Electron. Syst.* **2023**, pp. 1–13. [CrossRef]
18. Alzaq-Osmanoglu, H.; Alrehaili, J.; Ustundag, B.B. Low-SNR Modulation Recognition based on Deep Learning on Software Defined Radio. In Proceedings of the 2022 5th International Conference on Advanced Communication Technologies and Networking (CommNet), Marrakech, Morocco, 12–14 December 2022; pp. 1–6.
19. Wang, Y.; Liu, M.; Yang, J.; Gui, G. Data-Driven Deep Learning for Automatic Modulation Recognition in Cognitive Radios. *IEEE Trans. Veh. Technol.* **2019**, *68*, 4074–4077. [CrossRef]
20. Meng, F.; Chen, P.; Wu, L.; Wang, X. Automatic Modulation Classification: A Deep Learning Enabled Approach. *IEEE Trans. Veh. Technol.* **2018**, *67*, 10760–10772. [CrossRef]
21. Zhang, L.; Liu, H.; Yang, X.; Jiang, Y.; Wu, Z. Intelligent Denoising-Aided Deep Learning Modulation Recognition With Cyclic Spectrum Features for Higher Accuracy. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 3749–3757. [CrossRef]
22. DeepSig Incorporated. RF Datasets for Machine Learning. 2018. Available online: https://www.deepsig.ai/datasets (accessed on 29 April 2021).
23. Liu, D.; Wang, P.; Wang, T.; Abdelzaher, T. Self-Contrastive Learning based Semi-Supervised Radio Modulation Classification. In Proceedings of the MILCOM 2021-2021 IEEE Military Communications Conference (MILCOM), San Diego, CA, USA, 29 November–2 December 2021; pp. 777–782.
24. Chen, Z.; Cui, H.; Xiang, J.; Qiu, K.; Huang, L.; Zheng, S.; Chen, S.; Xuan, Q.; Yang, X. SigNet: A Novel Deep Learning Framework for Radio Signal Classification. *IEEE Trans. Cogn. Commun. Netw.* **2022**, *8*, 529–541. [CrossRef]
25. Wang, N.; Liu, Y.; Ma, L.; Yang, Y.; Wang, H. Multidimensional CNN-LSTM network for automatic modulation classification. *Electronics* **2021**, *10*, 1649. [CrossRef]
26. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
27. Chen, T.; Gao, S.; Zheng, S.; Yu, S.; Xuan, Q.; Lou, C.; Yang, X. EMD and VMD Empowered Deep Learning for Radio Modulation Recognition. *IEEE Trans. Cogn. Commun. Netw.* **2022**, *9*, 43–57. [CrossRef]
28. Uppal, A.J.; Hegarty, M.; Haftel, W.; Sallee, P.A.; Cribbs, H.B.; Huang, H.H. High-Performance Deep Learning Classification for Radio Signals. In Proceedings of the 2019 53rd Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 3–6 November 2019.
29. Park, C.; Choi, J.; Nah, S.; Jang, W.; Kim, D.Y. Automatic Modulation Recognition of Digital Signals using Wavelet Features and SVM. In Proceedings of the 2008 10th International Conference on Advanced Communication Technology, Gangwon, Republic of Korea, 17–20 February 2008; Volume 1, pp. 387–390. [CrossRef]
30. Teng, X.; Tian, P.; Yu, H. Modulation Classification Based on Spectral Correlation and SVM. In Proceedings of the 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing, Dalian, China, 12–14 October 2008; pp. 1–4. [CrossRef]
31. Zhang, Z.; Wang, C.; Gan, C.; Sun, S.; Wang, M. Automatic Modulation Classification Using Convolutional Neural Network With Features Fusion of SPWVD and BJD. *IEEE Trans. Signal Inf. Process. Netw.* **2019**, *5*, 469–478. [CrossRef]
32. Zheng, S.; Qi, P.; Chen, S.; Yang, X. Fusion Methods for CNN-Based Automatic Modulation Classification. *IEEE Access* **2019**, *7*, 66496–66504. [CrossRef]
33. Mao, Y.; Dong, Y.Y.; Sun, T.; Rao, X.; Dong, C.X. Attentive Siamese Networks for Automatic Modulation Classification Based on Multitiming Constellation Diagrams. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *34*, 5988–6002. [CrossRef]
34. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
35. Tridgell, S. Low Latency Machine Learning on FPGAs. Ph.D. Thesis, The University of Sydney Australia, Sydney, NSW, Australia, 2019.
36. Mendis, G.J.; Wei-Kocsis, J.; Madanayake, A. Deep Learning Based Radio-Signal Identification With Hardware Design. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *55*, 2516–2531. [CrossRef]
37. Wang, Z.; Sun, D.; Gong, K.; Wang, W.; Sun, P. A Lightweight CNN Architecture for Automatic Modulation Classification. *Electronics* **2021**, *10*, 2679. [CrossRef]
38. Snyder, D.; Garcia-Romero, D.; Sell, G.; Povey, D.; Khudanpur, S. X-vectors: Robust DNN Embeddings for Speaker Recognition. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 5329–5333.
39. Li, L.; Dong, Z.; Zhu, Z.; Jiang, Q. Deep-Learning Hopping Capture Model for Automatic Modulation Classification of Wireless Communication Signals. *IEEE Trans. Aerosp. Electron. Syst.* **2023**, *59*, 772–783. [CrossRef]
40. McNemar, Q. Note on the Sampling Error of the Difference Between Correlated Proportions or Percentages. *Psychometrika* **1947**, *12*, 153–157. [CrossRef]
41. Cai, J.; Gan, F.; Cao, X.; Liu, W. Signal Modulation Classification Based on the Transformer Network. *IEEE Trans. Cogn. Commun. Netw.* **2022**, *8*, 1348–1357. [CrossRef]

42. Wen, Y.; Zhang, K.; Li, Z.; Qiao, Y. A discriminative feature learning approach for deep face recognition. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 499–515.
43. Zhang, H.; Zhou, F.; Wu, Q.; Wu, W.; Hu, R.Q. A Novel Automatic Modulation Classification Scheme Based on Multi-Scale Networks. *IEEE Trans. Cogn. Commun. Netw.* **2022**, *8*, 97–110. [CrossRef]
44. Zhang, D.; Lu, Y.; Li, Y.; Ding, W.; Zhang, B. High-Order Convolutional Attention Networks for Automatic Modulation Classification in Communication. *IEEE Trans. Wirel. Commun.* **2022**, *22*, 4600–4610. [CrossRef]
45. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2015**, arXiv:1412.6980.
46. Chollet, F. *Deep Learning with Python*; Manning Publications: Shelter Island, NY, USA, 2021.
47. Yu, F.; Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions. In Proceedings of the International Conference on Learning Representations (ICLR), San Juan, Puerto Rico, 2–4 May 2016.
48. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6848–6856.
49. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning. PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
50. Okabe, K.; Koshinaka, T.; Shinoda, K. Attentive Statistics Pooling for Deep Speaker Embedding. In Proceedings of the Interspeech 2018, Hyderabad, India, 2–6 September 2018; pp. 2252–2256. [CrossRef]
51. Safari, P.; India, M.; Hernando, J. Self-Attention Encoding and Pooling for Speaker Recognition. In Proceedings of the Interspeech 2020, Shanghai, China, 25–29 October 2020; pp. 941–945. [CrossRef]
52. Sammit, G.; Wu, Z.; Wang, Y.; Wu, Z.; Kamata, A.; Nese, J.; Larson, E.C. Automated prosody classification for oral reading fluency with quadratic kappa loss and attentive x-vectors. In Proceedings of the ICASSP 2022–2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 23–27 May 2022.
53. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
54. Huynh-The, T.; Pham, Q.V.; Nguyen, T.V.; Nguyen, T.T.; Costa, D.B.d.; Kim, D.S. RanNet: Learning Residual-Attention Structure in CNNs for Automatic Modulation Classification. *IEEE Wirel. Commun. Lett.* **2022**, *11*, 1243–1247. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.