LEARNABLE STATISTICAL MOMENTS POOLING FOR AUTOMATIC MODULATION CLASSIFICATION

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

Southern Methodist University

ABSTRACT

We introduce a differentiable statistical moment aggregation layer, enabling networks to learn the optimal method of statistical moment pooling for automatic modulation classification. Statistical pooling, a cornerstone of convolutional networks, consolidates activations into fixed-length representations. Traditionally, this entails mean, variance, and higher-ordered statistics pooling defined as fixed hyperparameters. By enabling the statistics layer to become differentiable, networks are able to optimize the method of statistical aggregations, transcending predefined hyperparameters. With our approach, the statistical moment order is differentiable. Our results demonstrate learned statistical moments are able to outperform fixed-moments—improving modulation classification performance of a time-domain signal.[1]

Index Terms— Statistical moments, automatic modulation classification, deep learning, learnable pooling

1. INTRODUCTION

Neural networks specializing in automatic modulation classification (AMC) have become increasingly prevalent in recent years. Significant attention has focused on improving performance through employing various neural network architectures. Given the inherent variability of radio transmission data, often received in transient bursts, many approaches have proposed architectures that naturally handle variable-length sequences without retraining, cropping, or padding [1–4].

Addressing the intricacies of variable-length data, the use of global statistical moments are often employed. Moments capture distribution characteristics including location, spread, and shape. Three forms of statistical moments are raw, central, and standardized moments. In convolutional neural networks (CNNs), global average pooling, the first-order raw moment, and variance pooling, the second-order central moment, are often utilized. Transformer based approaches often invoke global average pooling and max pooling on output sequences [5, 6].

Using aggregate statistics has a few fundamental purposes. First, these statistics distill activations into a fixed-length representation amenable to processing by dense or fully-connected layers. Because global aggregation collapses the time dimension, the output shape is only dependent on the number of channels enabling predictions without network alterations. Second, the statistics aim to characterize the distribution of intermediate activations that aid classification performance. Aggregation has the potential to amalgamate inherent signal details, ultimately emphasizing characteristics specific to a particular class. Third, feature aggregation can increase generalization performance and reduce computational requirements. Aggregation collapses the input features into single statistics describing the distribution of the input features. This possibly introduces a form of regularization that reduces the chances of overfitting by removing extraneous activations. Additionally, aggregation decreases the number of values the network has to maintain in memory, leading to a more computationally efficient model.

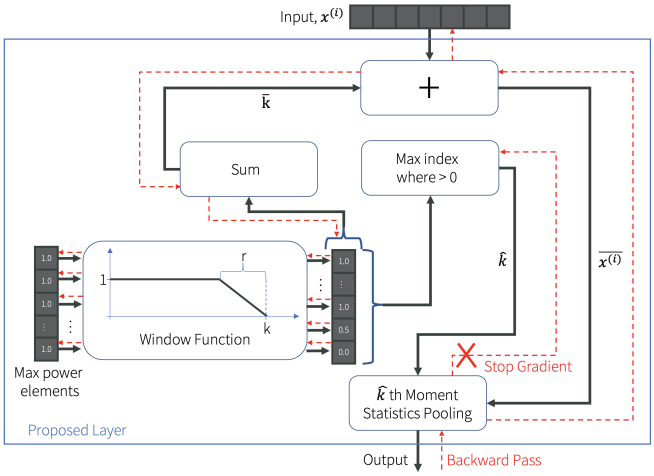


Fig. 1: Overview of the utilized method enabling differentiable statistical moment pooling for central and standardized moments.

In deep learning, mean and variance are often utilized such as d-vector [7] and x-vector [8] architectures because they are computationally efficient. Other works in speaker recognition have investigated the use of higher-ordered moments [9, 10], namely skewness and kurtosis, to provide additional measures of aggregation that may help characterize the distribution of intermediate activations. Skewness, the third order standardized moment, and kurtosis the fourth order standardized moment, estimate the asymmetry and peakedness of the distribution respectively. However, most existing architectures use fixed-statistical moments as aggregation methods. These moments are essentially treated as hyperparameters that do not evolve during training. Therefore, these statistics may not be optimal measures of latent statistical aggregations for improved AMC performance.

Statistical moments have long been found to be discriminating factors for AMC. Normalized standard deviation has been shown to be effective in distinguishing constant amplitude signals like FM, FSK, and BPSK, while skewness and kurtosis have been found to be useful features for classifying broader modulation schemes [11]. In particular, kurtosis and joint moments are adept at classifying PSK

[1] https://github.com/caharper/learnable_moments_pooling

and QAM schemes [12–14]. Although features are abstracted into high-dimensional latent spaces in deep learning, we argue that exploring additional statistical moments can likewise increase predictive performance of deep AMC models.

Conventional pooling approaches pose a limitation of applying the same aggregation to all channels. To capture channel-specific traits effectively, applying different statistical moment aggregations to different channels might be beneficial. However, optimizing such an approach would involve an onerous hyperparameter search, especially in high-dimensional architectures as statistical moment order is not directly differentiable.

In this work, we present a novel approach enhancing AMC performance using differentiable statistical moment aggregations allowing the statistical moment order to be learned. Additionally, we explore channel-specific statistical moment aggregation, providing a more nuanced latent representation of the employed modulation scheme. Our method demonstrates promising results for enhancing the expressive performance of statistical moment aggregations in machine learning architectures and improves AMC performance.

2. STATISTICAL MOMENTS

Statistical moments are functions describing distribution characteristics of an input sequence. Consider a single-channel sequence for channel i as $\mathbf{x}^{(i)} = [x_1, x_2, ..., x_N]$ where $i \in \mathbb{R}^C$, C is the number of channels, and N is the total number of elements. The expected value is given by Equation (1).

$$\mathbb{E}[\mathbf{x}^{(i)}] = \mu_X^{(i)} = \frac{1}{N}\sum_{j=1}^{N} x_j^{(i)} \tag{1}$$

Statistical moments materialize in three forms—raw, central, and standardized, each delineating distinct aspects of the distribution. To optimize neural networks with gradient descent, it is necessary for the loss space to be differentiable. However, the statistical moment order, k, is not directly differentiable. Traditional fixed-aggregation methods do not face gradient issues as k is static and has no corresponding gradient. However, to enable learnable k, it is necessary to address these gradient calculation challenges.

2.1. Raw Moments

The kth raw moment is shown in Equation (2). When $k = 1$, the location, or mean, of a distribution is defined. The corresponding gradient with respect to k can be seen in Equation (3).

$$r_k(\mathbf{x}^{(i)}) = \mathbb{E}[(\mathbf{x}^{(i)})^k] = \frac{1}{N}\sum_{j=1}^{N}(x_j^{(i)})^k \tag{2}$$

$$\frac{\partial r_k(\mathbf{x}^{(i)})}{\partial k} = \frac{1}{N}\sum_{j=1}^{N}(x_j^{(i)})^k \ln(x_j^{(i)}) \tag{3}$$

Given $\ln(\cdot)$ is only defined for values greater than 0, $\{x_j^{(i)}\}_{j\in N} > 0$ must hold. To satisfy this constraint, we utilize the rectified linear unit (ReLU) [15] such that the learned raw kth moment is computed over $ReLU(x^{(i)}) + \epsilon$ where ϵ is a small, positive constant. In our work, we clip the value of k such that $k \in [\epsilon_k, 6.0]$ where ϵ_k is also a small, positive constant. This preserves the non-negativity requisite for statistical moments. Notably, $k = 0$ yields a constant output regardless of the input. 6.0 is chosen as the maximum value to avoid the exploding gradient problem. We found that higher-ordered moments are rarely found to be optimal during training. Thus, this is a reasonable upper bound for AMC.

2.2. Central and Standardized Moments

Central moments, or moments about the mean, μ, describe the shape of the distribution independent of translation (Equations (4) and (5) show the function and gradient).

$$c_k(\mathbf{x}^{(i)}) = \mathbb{E}\left[(\mathbf{x}^{(i)} - \mu_X^{(i)})^k\right] = \frac{1}{N}\sum_{j=1}^{N}(x_j^{(i)} - \mu_X^{(i)})^k \tag{4}$$

$$\frac{\partial c_k(\mathbf{x}^{(i)})}{\partial k} = \frac{1}{N}\sum_{j=1}^{N}(x_j^{(i)} - \mu_X^{(i)})^k \cdot \ln\left(x_j^{(i)} - \mu_X^{(i)}\right) \tag{5}$$

Standardized moments (see Equations (6) and (7)) are normalized by the standard deviation, σ, allowing the measure to be scale invariant describing the tailedness of distributions.

$$s_k(\mathbf{x}^{(i)}) = \mathbb{E}\left[\left(\frac{\mathbf{x}^{(i)} - \mu_X^{(i)}}{\sigma_X^{(i)}}\right)^k\right] = \frac{1}{N}\sum_{j=1}^{N}\left(\frac{x_j^{(i)} - \mu_X^{(i)}}{\sigma_X^{(i)}}\right)^k \tag{6}$$

$$\frac{\partial s_k(\mathbf{x}^{(i)})}{\partial k} = \frac{1}{N}\sum_{j=1}^{N}\left[\left(\frac{x_j^{(i)} - \mu_X^{(i)}}{\sigma_X^{(i)}}\right)^k \cdot \ln\left(\frac{x_j^{(i)} - \mu_X^{(i)}}{\sigma_X^{(i)}}\right)\right] \tag{7}$$

where $\sigma_X^{(i)} = |\sqrt{c_2(\mathbf{x}^{(i)})}|$.

Several challenges must be overcome to allow k to be differentiable for central and standardized moments. $\sigma_X^{(i)}$ must be greater than 0 to avoid a division by zero for standardized moments. Because the variance, and therefore standard deviation, are non-negative, adding a small, positive perturbation factor, ϵ, avoids this problem. Given $k \in \mathbb{R}$, for both central and standardized moments to have strictly real outputs, one would have to ensure the centering operation $(x_j^{(i)} - \mu_X^{(i)}) \geq 0$. For $k \in \mathbb{Z}$, this is not required; however, sequences of integers are not differentiable. Even if complex values could be utilized (*e.g.*, $k \in \mathbb{R}$), $\ln(\cdot)$ is only defined for values greater than 0. ReLU activation could be utilized similarly to raw moments. However, unlike raw moments, this would need to be performed within the the computation of the moments, not prior to the computation. This would produce a value that is not a genuine statistical moment.

To overcome these challenges, we propose an architecture utilizing the *stop gradient* operator [16]. Our approach simulates fixed-moments while enabling a gradient-enabled k. A differentiable window function akin to [17,18] proposed in [19] facilitates this process. A smoothness factor, r, takes the value of the window function down from 1 to 0 in r steps. In our work, we set $r = 1$. A two-branch mechanism, as portrayed in Figure 1, encapsulates our proposed architecture for central and standardized moments. The first branch, enabled by the stop gradient operation, mirrors fixed-moment utilization with the applied value of k, $\hat{k}$, to overcome gradient instability posed by centering operations. Additionally, because $\hat{k} \in \mathbb{Z}$, the network avoids complex values if $(x_j^{(i)} - \mu_X^{(i)}) < 0$. This is accomplished by finding the maximum index where the result of the window function is greater than 0. In practice, this implements the

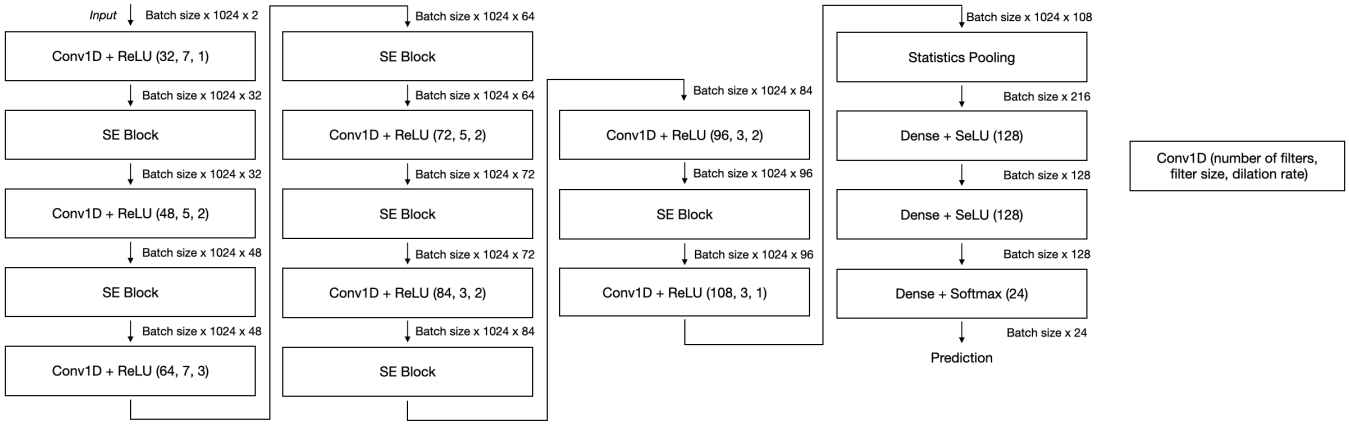


Fig. 2: Overview of the base architecture.

floor operation, $\lfloor k \rfloor$. The second branch facilitates backward gradient flow, making k learnable. It aggregates window outputs to scalar $\bar{k}$, which is then added to input $x_j^{(i)}$ before statistical moment evaluation. Because the same value, $\bar{k}$, is added to each $x_j^{(i)}$, the resulting moments are unaffected due to the centering operation. That is, $c_{\hat{k}}(\mathbf{x}^{(i)}) = c_{\hat{k}}(\mathbf{x}^{(i)} + \bar{k})$ and $s_{\hat{k}}(\mathbf{x}^{(i)}) = s_{\hat{k}}(\mathbf{x}^{(i)} + \bar{k})$. To avoid feature collapse to constant values and exploding gradient, we clip k such that $k \in [2.0, 5.0]$ and $k \in [3.0, 6.0]$ for central and standardized moments respectively.

3. DATASET

To investigate the utility of these methods for AMC, we use the RadioML 2018.01A dataset that is comprised of 24 different modulation types with a total of 2.56 million labeled signals $S(T)$ sampled at 900MHz [20, 21]. Each signal contains 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 ranges between -20dB to $+30$dB signal to noise ratio (SNR) with 106,496 observations per modulation scheme. We use the same train and test splits as [3] with 1 million training observations and 1.5 million testing observations.

4. EXPERIMENTAL DESIGN

Our architecture is based on previous work [3] using 7 convolutional layers, each followed by squeeze-and-excitation (SE) blocks [22] as depicted in Figure 2. Each SE block is defined as temporal global average pooling across convolutional filters followed by two dense, or fully-connected, layers. We use a dimensionality reduction ratio of 2 for each SE block adhering to the approach in [3]. Pooling strategies are denoted by the components in the statistics pooling vector. To signify fixed-moments, we employ lowercase notation, including mean (μ), variance (σ^2), skewness (γ), and kurtosis (κ). Uppercase letters denote learned-moments, encompassing raw (U), central (Σ), and standardized (Γ) moments. Statistics vectors containing both fixed and learned-moments are denoted *mixed*-moments. Central moments are uniformly initialized to $k = 2$ and standardized moments are uniformly initialized to $k = 3$.

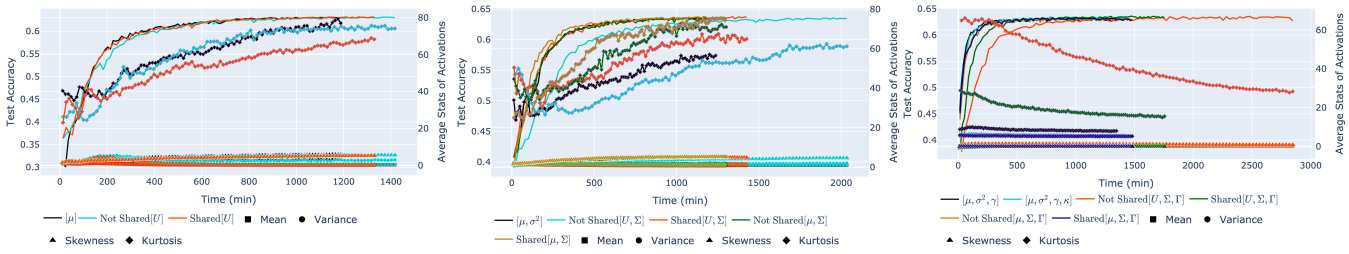
Due to the volume of models, each is trained for total of 100 epochs. We use the Adam optimizer [23] with an initial learning rate lr = 1e-4. A decay factor of 0.1 is applied to the learning rate if training loss does not decrease after 7 epochs with a minimum learning rate of 1e-7.

Table 1: RadioML2018.01A classification results. Best performers for each grouping are shown in bold font. *Std.* stands for standardized moment. Test accuracy is reported across the full SNR range $[-20, 30]$dB with peak accuracy at the best performing SNR value.

Type	Statistics	Shared Weights	Test Accuracy (%)	Peak Accuracy (%)	# Params (K)
Raw	$[\mu]$	✓	62.93	97.45	200.72
	$[U]$	✓	**63.15**	**97.82**	200.721
	$[U]$	–	62.94	97.65	200.828
Central	$[\mu, \sigma^2]$	✓	63.54	**98.80**	214.544
	$[U, \Sigma]$	✓	63.67	98.77	214.548
	$[U, \Sigma]$	–	63.49	98.67	214.762
	$[U, \sigma^2]$	✓	**63.68**	98.75	214.545
	$[U, \sigma^2]$	–	63.51	98.62	214.652
Std.	$[\mu, \sigma^2, \gamma]$	✓	62.96	**98.84**	228.368
	$[\mu, \sigma^2, \gamma, \kappa]$	✓	63.30	98.79	242.192
	$[U, \Sigma, \Gamma]$	✓	**63.35**	98.72	228.373
	$[U, \Sigma, \Gamma]$	–	62.76	97.68	228.694
	$[U, \sigma^2, \gamma, \kappa]$	✓	63.00	98.65	242.193
	$[U, \sigma^2, \gamma, \kappa]$	–	62.93	98.60	242.3

5. RESULTS AND DISCUSSION

Experimental results can be found in Table 1. We were able to replicate similar results to [3] with our x-vector, $[\mu, \sigma^2]$, architecture. However, differences in model performance can be attributed to solely monitoring training loss and limiting the total number of epochs to 100. Due to the scale of the test set, 1.5 million observations, even subtle differences in accuracy can indicate thousands of additional correct classifications. Each grouping found the best performers to contain learned-moments, supporting our hypothesis that



(a) (b) (c)

Fig. 3: Convergence comparison of (a) raw, (b) central, and (c) standardized moment vectors.

networks can benefit from learning their own aggregation method. While learned-moments proved advantageous, learning k for each channel was outperformed by using using shared k weights. Allowing each channel to learn individual aggregation methods may overcomplicate the loss space and disrupt the optimization. Sharing the value of k across channels may provide a useful regularization allowing the network to generalize. While our work focuses on learning a single raw, central, and standardized moment for our statistics pooling layer (*e.g.*, $[U, \Sigma, \Gamma]$), future work could investigate adding multiple moments (*e.g.*, $[U, U, \Sigma, \Sigma, \Gamma, \Gamma]$), perhaps capturing more nuanced signal characteristics.

While standardized moments may contain more statistical measures and parameters, we found that central moments had the highest classification accuracy overall. Because of their increased dimensionality, standardized moment vectors may benefit from additional dense neurons to capture non-linear relationships.

5.1. Convergence and Statistics of Intermediate Activations

We observe that the incorporation of standardized moments can potentially accelerate model convergence rate. This could be due to the additional information that standardized moments provide compared to raw and central moments—increasing network capacity. However, despite the potential benefits of using standardized moments, our experiments showed that centralized moment vectors outperformed other moment types in terms of overall performance.

To investigate this phenomenon, we record the distribution statistics of intermediate activations in the networks during training. Specifically, we record the mean (r_1), variance (c_2), skewness (s_3), and kurtosis (s_4). These statistics are computed just prior to statistics pooling for each model architecture. We use a random sample of 1,000 input signals from the test set and record each statistic at the end of each training epoch. The same randomly sampled dataset is used across all epochs and models to ensure fairness and comparability. To record single datapoints, we take the average of the statistics across all channels and $1,000$ evaluation observations. The results can be seen in Figures 3a to 3c.

Each model is color coded. The left y-axis denotes the test accuracy, shown as solid lines without markers. The right y-axis shows the value of each recorded statistic. Because our learned statistics introduce a small amount of additional parameters and are more computational than fixed-moments, we investigate performance over time instead of epochs to illustrate the viability of the approach.

Mean, variance, and skewness remained relatively stable throughout the training process for all models. Kurtosis, however, behaved differently for standardized moment vectors than raw and central moment vectors. Figures 3a and 3b, raw and central moment vectors, both illustrate increasing kurtosis over training iterations. With the exception of $[U, \Sigma, \Gamma]$ with k learned for each channel, kurtosis remained very stable throughout the training process for standardized moments. Our experiments uncovered a common trend between the kurtosis of the sampled data and accuracy on the test set. Generally, the more stable and consistent the value of kurtosis, the faster the models converged. This trend is particularly evident when comparing Figures 3b and 3c.

Consider $[U, \Sigma]$ using non-shared weights in Figure 3b. This model had the slowest convergence rate and one of the most variable kurtosis values for centralized moment vectors. Of the standardized moment vectors (Figure 3c), the slowest to converge was $[U, \Sigma, \Gamma]$ using non-shared weights. Notably, it had the most variable kurtosis of the standardized moment vectors. $[U, \Sigma, \Gamma]$ using shared weights was the second slowest to converge with the second most variable kurtosis. Models using standardized moments tend to converge faster with more stable kurtosis compared to raw and central counterparts, possibly due to stabilized statistics. Specifically, including standardized moments may ameliorate the covariate shift problem by ensuring that the moments of the activations are consistent across different training epochs. This could facilitate faster generalization to the test set.

Covariate shift has been studied extensively in the field of deep learning [24–28]. Covariate shift occurs when the distribution of intermediate activations of the training set do not match the distribution of the activations of the testing set. Covariate shift can lead to poor performance and reduced convergence rates as the weights of the model are not aligned with the testing set. Including standardized moments may reduce covariate shift in AMC and increase convergence rates. While our proposed method has additional computational overhead and therefore takes more time to reach 100 epochs (ranging from 1.1 to 2.1 times longer), our models containing standardized moments converge faster in terms of time. Nevertheless, our methods are able to outperform their fixed counterparts further justifying this additional cost.

6. CONCLUSION

In this work, we propose a novel method enabling differentiable statistical moment orders. Prior to our work, choosing moment orders was non-differentiable, often relying on heuristics limiting the expressiveness of AMC architectures. Enabling models to learn statistical moment orders showed improved AMC performance over fixed-moments without sacrificing convergence rates. While there is a small additional computational overhead associated with our method, the benefits of improved expressiveness and model performance justify this cost.

7. REFERENCES

[1] Clayton A. Harper, Lauren Lyons, Mitchell A. Thornton, and Eric C. Larson, "Enhanced automatic modulation classification using deep convolutional latent space pooling," in *2020 54th Asilomar Conference on Signals, Systems, and Computers*, 2020, pp. 162–165.

[2] Clayton A. Harper, Avi Sinha, Mitchell A. Thornton, and Eric C. Larson, "SNR-boosted automatic modulation classification," in *2021 55th Asilomar Conference on Signals, Systems, and Computers*, 2021, pp. 372–375.

[3] Clayton Harper, Mitchell Thornton, and Eric Larson, "Automatic modulation classification with deep neural networks," *arXiv preprint arXiv:2301.11773*, 2023.

[4] Qinghe Zheng, Penghui Zhao, Yang Li, Hongjun Wang, and Yang Yang, "Spectrum interference-based two-level data augmentation method in deep learning for automatic modulation classification," *Neural Computing and Applications*, vol. 33, no. 13, pp. 7723–7745, 2021.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[6] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *ArXiv*, vol. abs/1907.11692, 2019.

[7] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4052–4056.

[8] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[9] Shuai Wang, Yexin Yang, Yanmin Qian, and Kai Yu, "Revisiting the statistics pooling layer in deep speaker embedding learning," in *2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, 2021, pp. 1–5.

[10] Lanhua You, Wu Guo, Li-Rong Dai, and Jun Du, "Multi-Task Learning with High-Order Statistics for x-Vector Based Text-Independent Speaker Verification," in *Proc. Interspeech 2019*, 2019, pp. 1158–1162.

[11] Jackie E. Hipp, "Modulation classification based on statistical moments," in *MILCOM 1986 - IEEE Military Communications Conference: Communications-Computers: Teamed for the 90's*, 1986, vol. 2, pp. 20.2.1–20.2.6.

[12] Wei Dai, Youzheng Wang, and Jing Wang, "Joint power estimation and modulation classification using second- and higher statistics," in *2002 IEEE Wireless Communications and Networking Conference Record. WCNC 2002 (Cat. No.02TH8609)*, 2002, vol. 1, pp. 155–158 vol.1.

[13] C.J. Le Martret and D.M. Boiteau, "Modulation classification by means of different orders statistical moments," in *MILCOM 97 MILCOM 97 Proceedings*, 1997, vol. 3, pp. 1387–1391 vol.3.

[14] S.S. Soliman and S.-Z. Hsue, "Signal classification using statistical moments," *IEEE Transactions on Communications*, vol. 40, no. 5, pp. 908–916, 1992.

[15] Kunihiko Fukushima, "Cognitron: A self-organizing multilayered neural network," *Biological cybernetics*, vol. 20, no. 3-4, pp. 121–136, 1975.

[16] Yoshua Bengio, Nicholas Léonard, and Aaron Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.

[17] Rachid Riad, Olivier Teboul, David Grangier, and Neil Zeghidour, "Learning strides in convolutional neural networks," *ICLR*, 2022.

[18] Clayton Harper, Mitchell Thornton, and Eric Larson, "DCT-diffstride: Differentiable strides with real-valued data," 2023.

[19] Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin, "Adaptive attention span in transformers," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, July 2019, pp. 331–335, Association for Computational Linguistics.

[20] T.J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," in *IEEE Journal of Selected Topics in Signal Processing*. IEEE, 2018, vol. 12:1, pp. 168–179.

[21] DeepSig Incorporated, "RF datasets for machine learning," `https://www.deepsig.ai/datasets`, 2018, Accessed: 2021-04-29.

[22] Jie Hu, Li Shen, and Gang Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[23] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.

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

[25] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge, "Improving robustness against common corruptions by covariate shift adaptation," *Advances in Neural Information Processing Systems*, vol. 33, pp. 11539–11551, 2020.

[26] Felix Ott, David Rügamer, Lucas Heublein, Bernd Bischl, and Christopher Mutschler, "Domain adaptation for time-series classification to mitigate covariate shift," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 5934–5943.

[27] Xiao Sun, Naigang Wang, Chia-Yu Chen, Jiamin Ni, Ankur Agrawal, Xiaodong Cui, Swagath Venkataramani, Kaoutar El Maghraoui, Vijayalakshmi Viji Srinivasan, and Kailash Gopalakrishnan, "Ultra-low precision 4-bit training of deep neural networks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1796–1807, 2020.

[28] Amrutha Machireddy, Ranganath Krishnan, Nilesh Ahuja, and Omesh Tickoo, "Continual active adaptation to evolving distributional shifts," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3444–3450.