

# MuShAug: Boosting Sequence Signal Classification via Multishape Augmentation

Jongseok Kim<sup>ID</sup>, *Student Member, IEEE*, and Ohyun Jo<sup>ID</sup>, *Member, IEEE*

**Abstract**—The utilization of sequence signals in real-world mobile communications plays a crucial role in the design and optimization of communication methods. Through our own performance evaluation, we have confirmed that conventional augmentation techniques, mainly designed for image or photograph data, are unsuitable for sequence signal applications due to inherent differences in data characteristics. To address this practical limitation, multishape augmentation (MuShAug) employs sequence signal-to-image (SSI) to represent sequence signals in image format, enabling the extraction of diverse signal features. To evaluate the practical applicability of our proposed method, we conduct experiments using real-world sequence signals collected from operational Fifth Generation (5G) mobile communication systems. In experimental trials, MuShAug consistently demonstrates robust generalization performance, achieving high levels of classification accuracy. Furthermore, through the incorporation of random phase transformation (RPT), our method achieves further enhanced performance within advanced data augmentation techniques.

**Index Terms**—Data augmentation, Fifth Generation (5G), inception module, multishape augmentation (MuShAug), random phase transformation (RPT), sequence signal-to-image (SSI), signal classification.

## I. INTRODUCTION

DATA augmentation technique has garnered significant success across various fields, notably in computer vision, by augmenting both the quantity and diversity of data essential for training deep learning models [1], [2], [3]. By employing data augmentation, models are safeguarded against overfitting while also improving their ability to generalize across diverse environments [4], [5], [6]. In research areas, such as signal processing and signal classification [7], [8], where addressing various environmental effects is crucial [9], data augmentation can play a vital role. For instance, signal classification across different devices facilitates environmental perception in communication [10], [11], [12]. In recent communication

Manuscript received 3 April 2024; revised 28 May 2024; accepted 19 June 2024. Date of publication 24 June 2024; date of current version 9 October 2024. This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant funded by the Korea Government (MSIT, Development of 5G+ Intelligent Basestation Software Modem) under Grant 2021-0-00165, and in part by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government (MSIT) under Grant 2021R1A2C2095289. This article was presented in part at the Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24) conference proceedings [DOI: 10.1609/aaai.v38i21.30464]. (Corresponding author: Ohyun Jo.)

The authors are with the Department of Computer Science, Chungbuk National University, Chenogju 28644, South Korea (e-mail: kjeok@chungbuk.ac.kr; ohyunjo@chungbuk.ac.kr).

Digital Object Identifier 10.1109/JIOT.2024.3418019

technologies such as Fifth Generation Technology Standard (5G), signal classification plays a crucial role in designing and optimizing communication methods [13], [14], [15], [16].

To accurately classify data influenced by various environmental factors, such as signals, a significant volume of signal data is essential. However, obtaining signal data poses challenges, taking into account both cost and the need for diverse sources [17]. Furthermore, the use of conventional image augmentation techniques [18], [19], [20], [21], well known for addressing limited data, is not suitable for sequence signals. Therefore, research focused on sequence signals necessitates methods for data augmentation [22], [23]. Various methods, including signal mixing and transformation, are available for signal augmentation [24], [25], [26]. Signal mixing combines multiple signals to generate a new signal [27], whereas signal transformation alters signal characteristics like phase and amplitude to create a new signal [28]. However, while these methods may increase the quantity and diversity of training data, they often struggle to preserve the original signal and can distort signal characteristics [29], [30]. Additionally, in real-world scenarios, factors beyond existing noise can introduce additional noise, thereby impacting signal classification performance.

To address these challenges, this article proposes multishape augmentation (MuShAug). MuShAug utilizes the sequence signal-to-image (SSI) method, maintaining the original values of the signal while diversifying its form. SSI enhances the correlation between elements of in-phase and quadrature-phase (IQ) data while converting traditional signal data into images, thereby facilitating smoother learning through 2D-Convolution. Additionally, it combines the Inception Module [31], facilitating smoother feature extraction of signals even with a limited number of data. Additionally, the random phase transformation (RPT) stage is integrated to enhance the characteristics of signals and generate new signals. The RPT stage selects a random phase to strengthen the features of signals and improve the model's generalization [32]. Moreover, by utilizing real collected 5G data, this research demonstrates strong adaptability in diverse real-world environments, showcasing its potential for application in actual systems.

The pivotal contributions of the proposed technology, MuShAug, outlined in this article are as follows.

- 1) MuShAug leverages SSI to seamlessly preserve the authentic values and inherent characteristics of signals. Moreover, this study experimentally validates the performance degradation issue in sequence signal classification commonly encountered when applying

- conventional image augmentation techniques, and proficiently alleviates it using MuShAug.
- 2) MuShAug applies a transformative process to signals, diversifying their forms across each input layer. This enables the extraction of distinct features at every layer, thereby improving signal feature extraction and contributing positively to the classification features of the model.
  - 3) By employing MuShAug, it becomes possible to achieve performance comparable to the original quantity using a reduced amount of data during training. This not only shortens the model's training time but also underscores its applicability to real systems, showcasing its efficiency in handling limited data sets.
  - 4) MuShAug further enhances the overall performance of the model through additional integration with RPT. This combination increases the model's generalization performance and accuracy, demonstrating its improved robustness.
  - 5) Validating performance in real-world environments involves utilizing authentic 5G data. Training the model with real 5G data ensures that experiments closely mimic real-system scenarios. This approach enhances the model's adaptability and robustness, as it is exposed to the complexities present in actual operational environments.

## II. RELATED WORK

### A. Data Augmentation

Data augmentation is an efficient method to enhance model generalization and data diversity. Especially for training on data containing various noise, such as signals, or data that is challenging to collect, data augmentation is crucial. Huang et al. [28] examined various instances to evaluate the impact of different signal augmentations on the classification performance of signals. Tang et al. [7] introduced a data augmentation method using generative adversarial networks for automatic modulation classification. However, these studies lack a detailed analysis of a crucial aspect of augmentation, namely, the model's generalization performance. Furthermore, an analysis of low-level feature extraction, particularly focusing on the closest relationship with the original signal, is not included in the feature extraction domain. Many existing data augmentation methods are not tailored specifically for sequence signal data; instead, they are designed for general photograph image data. Considering the characteristics of signals, we conduct more experiments and analyses to validate our proposed method.

### B. Inception Module

Data augmentation, which allows for the extraction of numerous features from diverse data and enhances performance, is closely related to the Inception module. The Inception module employs kernels of various sizes to extract features of varying sizes and combines them, enabling the network to learn diverse features. Zhang et al. [33] proposed an end-to-end network based on the Inception

structure for accurate recognition of radar modulation modes. Xue et al. [34] suggested a network that combines an improved Inception module with convolutional neural network (CNN) for automatic modulation classification. Mumtaz et al. [35] analyzed the practicality of the Inception module by proposing a network with an Inception module for autonomously classifying modulations from IQ data received in the channel. However, such studies do not encompass scenarios where there is a scarcity of sequence signal. In real-world systems, there may be a shortage of learnable signals, and various elements such as noise can be present in the signals. To address this, we propose a method that combines the Inception module, providing a robust signal classification network in diverse environments.

### C. Signal Feature Extraction

Signals typically have a sequential data format, leading to limitations in the design of Convolution layers for feature extraction. Moreover, validated deep learning models used for signal classification are models trained on data with a format similar to images. To address these issues, research is underway to transform signals into an image format for unrestricted feature extraction. Lee et al. [36] transformed various signal features into 2-D images to train and analyze the classification performance for the modulation types of signals. Liu et al. [37] proposed a framework that preprocesses wireless signals and represents their features in a 2-D time–frequency image. These studies aim to effectively extract signal features by transforming signals into images and training deep learning models. However, commonly, the transformation into images relies on utilizing the features or properties of input signals, without preserving the original values. Additionally, transforming signals into images incurs a certain level of computational cost. Considering the practical application in real systems, we propose a method that maintains the original values of signals while transforming them into a format similar to images for training.

## III. METHODS

In this section, we first define the Signal Model and then proceed to explain the concept of the proposed MuShAug and RPT technique aimed at maximizing the functionality of Augmentation, which allows the addition of layers to the model.

### A. Preliminaries

The Zadoff–Chu sequence [38], [39] is a mathematical sequence composed of complex numbers with a fixed length. This sequence finds application in user detection and channel measurement in 5G mobile systems. When applied to a signal for communication, it generates a new signal with a constant amplitude and IQ data, allowing for representation in an orthogonal coordinate system. A single sequence encompasses both IQ components, and an element of the sequence can be expressed by the following formula:

$$z = \begin{bmatrix} I \\ Q \end{bmatrix} = \begin{bmatrix} \text{Acos}(\omega T + \theta) \\ \text{Asin}(\omega T + \theta) \end{bmatrix} \quad (1)$$

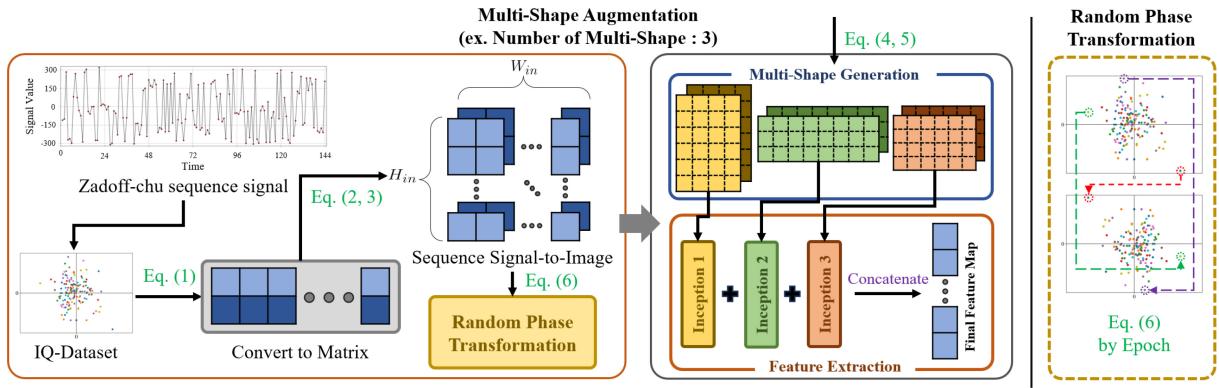


Fig. 1. Framework of MuShAug takes input in the form of images. When the number of Multishape Generation is set to 3, the number of Inception modules is also set to 3.

where  $A$  represents the amplitude,  $\omega$  denotes the angular frequency,  $T$  represents the time of the signal, and  $\theta$  represents the phase. The sequence signal without RPT is initialized with  $\theta$  set to 0. The variable  $z$  represents a complex number, and we denote the length of  $z$  as  $N$ . Assuming the IQ as a single real number, it can be represented as an  $R^{2 \times N}$  matrix. In this article, we utilize  $\theta$  to further enhance the performance of MuShAug. Additionally, to use IQ data as input for CNN, MuShAug transforms the shape of the data. The transformed signal data is converted into a form resembling a real image composed of pixels, which is referred as SSI in this work [13], [40].

### B. MuShAug: Multishape Augmentation

Sequence signals typically appear in a 1-D form, which can limit the effectiveness of convolution filters for feature extraction. Additionally, conventional CNN-based models, well validated for extracting features from image data, may face challenges in extracting features from sequence signals. To address this issue, MuShAug incorporates the SSI, transforming the shape of IQ sequence signals not only as  $R^{2 \times N}$  but also into a format resembling images. SSI is a technique that transforms the form of the original signal into a format resembling an image, allowing for diverse feature extraction from the signal [13]. MuShAug initially relocates the shape of the signal into a 3-D array resembling an image. Moreover, if we assume that  $W_{\text{in}}$  and  $H_{\text{in}}$  represent the width and height of the relocated signal, respectively, and  $W_{\text{in}} \times H_{\text{in}} = N$ , the relocation function can be expressed as follows:

$$r(W_{\text{in}}, H_{\text{in}}) : R^{2 \times N} \rightarrow R^{W_{\text{in}} \times H_{\text{in}} \times 2} \quad (2)$$

where the IQ each take the form of  $W_{\text{in}} \times H_{\text{in}}$ . Subsequently, they are input into individual channels, resulting in an array of shape  $R^{W_{\text{in}} \times H_{\text{in}} \times 2}$ . The structure of the relocated sequence signal can be represented as follows:

$$X_{i,j} = \begin{bmatrix} X_{i,j}^{(1)} & X_{i,j}^{(2)} \end{bmatrix} = \begin{bmatrix} R_{i,j,1} & R_{i,j,2} \end{bmatrix} \quad (3)$$

where  $X^{(1)}$  represents the first channel, and  $X^{(2)}$  represents the second channel, each containing data for IQ, respectively. Furthermore, each matrix has positions denoted by  $(i, j)$ , and the values corresponding to the positions in each channel are

represented as  $R_{i,j,1}$  and  $R_{i,j,2}$ . Here,  $(i, j)$  can be expressed as follows:

$$(i, j) \in \mathbb{Z}^2 \cap [0, W_{\text{in}}) \times [0, H_{\text{in}}) \quad (4)$$

where  $(i, j)$  represents a 2-D integer vector, each ranging from 1 to  $W_{\text{in}}$  and  $H_{\text{in}}$ , respectively. The detailed representation of the matrix based on these channels is as follows:

$$X_{i,j}^{(k)} = \begin{bmatrix} R_{0,0,k} & \dots & R_{0,H_{\text{in}}-1,k} \\ R_{1,0,k} & \dots & R_{1,H_{\text{in}}-1,k} \\ \vdots & \ddots & \vdots \\ R_{W_{\text{in}}-1,0,k} & \dots & R_{W_{\text{in}}-1,H_{\text{in}}-1,k} \end{bmatrix} \quad (5)$$

where  $k$  represents the channel, and IQ data uses the IQ as channels, so  $k \in \{1, 2\}$ . The IQ data, following the aforementioned process, can create various shapes based on  $W_{\text{in}}$  and  $H_{\text{in}}$ . In case  $W_{\text{in}} \times H_{\text{in}} = N$  is not satisfied, techniques, such as signal interpolation or sequence repeating, can be employed to fulfill the requirement.

When using MuShAug as a layer, it enables not only augmentation but also more diverse feature extraction. Fig. 1 illustrates the overall framework through MuShAug.

MuShAug preserves the original signal values while transforming their forms. Particularly, through the step of converting into image form, it can extract more diverse shapes and features. The Multishape Generation step involves specifying the number of shapes, given that various forms arise depending on  $W_{\text{in}}$  and  $H_{\text{in}}$ . Fig. 1 illustrates a structure with the number of Multishape Generation set to 3. MuShAug extracts individual features from the output through the Inception Module. This implies that signals with arrays of different shapes extract features diversely, particularly potent in the Conv2D layer. MuShAug is designed to enhance generalization while enabling the learning of robust models. To further maximize the advantages of MuShAug, we use RPT in conjunction. RPT can alter phases, leading to improved generalization performance and inducing augmentation effects. A detailed explanation of RPT is provided in the following section.

### C. MuShAug+RPT: Random Phase Transformation

In this section, we present a method to enhance the robust augmentation method by combining MuShAug and RPT. In

**Algorithm 1** MuShAug+RPT Feature Extraction

---

**Input:** Sequence signal  $z$ , Number of epoch  $t$

**Initialization:** sequence length  $N$ , number of Multi-Shape Generation  $s$ , Generate combinations of  $W_{in}$  and  $H_{in}$  for  $s$

**Output:** Final Feature Map

- 1:  $z$  : To transform  $(I + iQ)$  into a 2-D matrix. → Eq. (1)
- 2: Transform  $z$  into a 3-Dimensional matrix such as an image (Sequence Signal-to-Image) → Eq. (2), Eq. (3)
- 3: Conducting RPT (Random Phase Transform) for  $\theta$  transformation. → Eq. (6)
- 4: **while**  $0, \dots, s - 1$  **do**
- 5:   **if**  $s=1$  **then**
- 6:     Training with a Single Shape
- 7:   **end if**
- 8:   Relocation Function:  $r(W_{in}, H_{in})$  → Eq. (4), Eq. (5)
- 9:   Inputs enter each Inception Module
- 10:   Perform Feature Extraction and Flatten()
- 11: **end while**
- 12: Final Feature Map : Concatenate Flatten Feature Maps

---

the RPT stage, the phase is set randomly, and the timing of augmentation is set at each epoch. Changing it at each epoch does not significantly affect the overall training time since it does not increase the number of data. If we were to increase the number of data by the possible  $\theta$  at each epoch and then proceed with training, it could not only affect the overall training time but also degrade the generalization performance. The random phase for IQ data at each epoch can be expressed as follows:

$$f(t, \theta) = z^{(t)}, \theta^{(t)} \in \left\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\right\} \quad (6)$$

where  $t$  represents the epoch number, and  $\theta^{(t)}$  denotes the randomly changing  $\theta$  at each epoch. The function  $f$  represents the transformation function for (1) on an epoch basis. This approach allows us to train the network at each epoch with varying phase values, including the original form ( $\theta = 0$ ). We set  $\theta$  considering complete phase transformation as multiples of  $90^\circ$ , addressing issues such as phase invariance [41], [42] for sequence signals. Furthermore, by altering the training data at each epoch, it contributes to training a more robust model. Algorithm 1 describes the combination of MuShAug and RPT to extract features from the sequence signal at the  $t$ th epoch. It transforms the sequence signal into a form resembling an image and conducts RPT to be used as input for MuShAug (lines 1–3). Subsequently, through MuShAug, when feature maps are produced in each inception module, each of them is flattened and concatenated. In the case where  $s = 1$ , it is considered as a Single Shape (lines 4–12). The reason for preflattening is to ensure that the shape of the feature map sufficiently captures the influence of  $W_{in}$  and  $H_{in}$ . Through this process, MuShAug+RPT can extract more features without constraints during Feature Extraction.

TABLE I  
HYPERPARAMETER SETTINGS FOR THE BASE MODEL AND MUSHAUG

Parameters	
<b>Base Model</b>	VGGNet [43] - Light Weight
<b>Number of Multi-Shape</b>	3 shape
<b>Random Phase</b>	$0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$
<b>Multi-Shape Size</b> ( $N=144$ )	$W_{in} : 24, H_{in} : 6$ (1-shape) $W_{in} : 16, H_{in} : 9$ (2-shape) $W_{in} : 12, H_{in} : 12$ (3-shape)
<b>Epoch</b>	100

## IV. EXPERIMENTS

To substantiate the effectiveness of MuShAug in real-world scenarios, we conduct evaluations using authentic signals collected from operational commercial 5G systems. Furthermore, we assess our methods by applying scenarios from a data perspective that could address potential data scarcity issues when training models in real systems.

## A. Experimental Setting

To validate the effectiveness of our proposed method in a real system, we collected 5G signals, specifically the demodulation reference signals (DMRSs), using two universal software radio peripheral (USRP) devices [39], [44]. DMRS is a crucial signal in mobile wireless communication, commonly used in standards, such as LTE and 5G networks. Additionally, one crucial step in communication systems is the classification of the unique indices present in DMRS. This process is referred to as DMRS index classification. Further explanations and detailed information about the data set can be found in our Appendix for reference in this article. For DMRS index classification, we applied MuShAug by training the model on a reduced data set to demonstrate improved performance compared to conventional methods, even with limited data. During the data reduction process, we systematically removed a certain percentage of data for each index. We experimented with four different reduction ratios, where, for instance, R-0.75 indicates that 75% of the collected data was removed. Consequently, the Data By SNR becomes 40 000, and Data By Index becomes 5000. To ensure a rigorous experiment, we used 20% of the entire data set as test data and an additional 20% from the remaining data as validation data to more accurately evaluate the model's performance.

Additionally, for a more accurate validation of MuShAug, we employ the Inception Module of MuShAug by lightweighting the previously validated model. Detailed information regarding the hyperparameter settings of MuShAug can be found in Table I. We employ a lightweight version of the VGGNet [43] as the base model and utilize this lightweight model for feature extraction with an Inception module [45]. In this case, to ensure fair feature extraction for each input layer and for experimentation purposes, all Inception modules have the same structure. Detailed information about the model

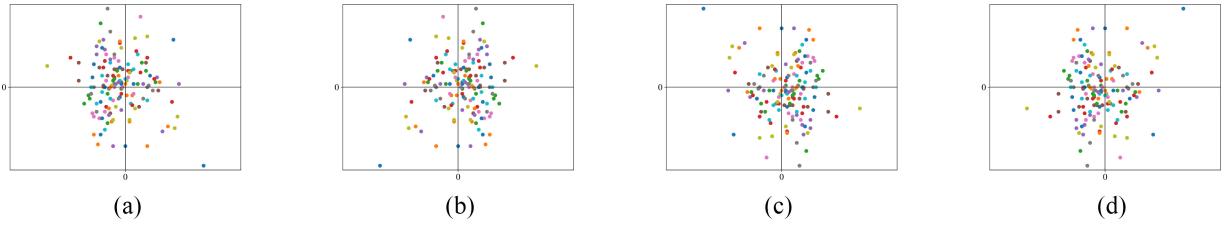
Fig. 2. Transformed the phase sequence signal as follows: (a) 0, (b)  $(\pi/2)$ , (c)  $\pi$ , and (d)  $(3\pi/2)$ .

TABLE II  
PERFORMANCE COMPARISON BASED ON RPT APPLICATION TIMING FOR EACH DATA SET

Dataset	Augmentation	Number of data	SNR(dB) .Acc(%)								Train Time		
			Timing	(Train Data)	-2.74	-2.81	-2.99	-3.13	-3.42	-3.7	Average		
R-0.0	None	102,400			96.29	97.14	95.34	94.47	90.40	82.23	65.21	88.72	216.93
	Pre-Processing	409,600			97.40	97.39	95.95	94.60	91.38	81.42	62.78	88.70	720.30
	Epoch	102,400			98.05 (0.65↑)	98.00 (0.61↑)	97.99 (2.04↑)	96.93 (2.33↑)	94.56 (3.18↑)	89.81 (8.39↑)	71.90 (9.12↑)	92.46 (3.76↑)	210.81 ( <b>509.49↓</b> )
R-0.2	Pre-Processing	327,680			95.77	96.98	95.49	94.18	89.86	80.00	59.84	87.45	577.72
	Epoch	81,920			97.17 (1.4↑)	97.89 (0.91↑)	96.76 (1.27↑)	96.07 (1.89↑)	94.23 (4.37↑)	87.54 (7.54↑)	73.87 (14.03↑)	91.93 (4.48↑)	167.70 ( <b>-410.02↓</b> )
R-0.5	Pre-Processing	204,800			94.41	95.79	92.52	91.43	85.09	72.47	42.84	82.08	379.21
	Epoch	51,200			96.34 (1.93↑)	96.74 (0.95↑)	94.70 (2.18↑)	94.38 (2.95↑)	91.27 (6.18↑)	80.61 (8.14↑)	62.11 (19.27↑)	88.02 (5.94↑)	116.59 ( <b>262.62↓</b> )
R-0.75	Pre-Processing	102,400			90.99	91.68	87.99	82.99	76.10	52.86	23.13	72.25	197.98
	Epoch	25,600			94.06 (3.07↑)	92.66 (0.98↑)	89.83 (1.84↑)	85.51 (2.52↑)	82.06 (5.96↑)	73.04 (20.18↑)	30.59 (7.46↑)	78.25 (6.00↑)	63.59 ( <b>134.39↓</b> )

architecture can be found in our Appendix. Additionally, in MuShAug, we specify the number of inputs as three to verify its applicability with a reduced number of input layers. Furthermore, to precisely validate the advanced stage of MuShAug, the RPT Stage, the random phase is set to be multiples of  $\pi/2$ . The input shape for MuShAug is also specified; however, this can vary significantly depending on the form and structure of the data, and MuShAug is a technique capable of accommodating these variations.

### B. Signal Classification With RPT Timing

We first validate the applicability of RPT. Since conventional augmentation techniques may not be suitable for all types of data, verification of RPT is essential. The results of applying traditional image augmentation, not RPT, to IQ data can be found in the Appendix. We output IQ coordinate systems before and after applying RPT, and Fig. 2 represents the IQ data for RPT. When utilizing IQ data trained with RPT-applied data, we can verify phase invariance. Additionally, we can examine the performance variations depending on the timing of RPT application. Table II demonstrates the performance based on the application timing of RPT for each data set. Essentially, a comparison is made between Preprocessing Timing and Epoch Timing within the same data set. The differences for each result are indicated in parentheses next to the Epoch Timing results. Additionally, the actual number of training samples and the required training time for learning are provided in the results. In Table II, for R-0.0, data without augmentation is labeled with RPT Timing set to None. In the context of R-0.0, “None Timing” refers to Fig. 2(a). Preprocessing Timing involves generating training data using all four phase values, and Epoch Timing involves randomly selecting one phase Fig. 2(a)–(d) for each epoch

during training, with no change in the number of data samples. Typically, as the training data decreases, the performance of classification models tends to decline. However, this is a common challenge that can be addressed through RPT techniques, and the performance varies depending on the RPT Timing. Preprocessing Timing refers to performing augmentation before training the sequence signal, allowing the model to learn from transformed data. Epoch Timing, on the other hand, involves applying random augmentation at each epoch during training to maintain the data quantity while facilitating learning.

Observing Preprocessing Timing and Epoch Timing for R-0.2 in Table II, it is evident that despite increasing the training data, Preprocessing Timing exhibits lower performance. However, Epoch Timing, even with a reduced data quantity, outperforms Preprocessing Timing due to the effects of augmentation. Particularly, at  $-3.42$  dB with the lowest data quantity in R-0.75, it demonstrates approximately 20% higher accuracy, showcasing good performance even at low SNR. Moreover, it consistently achieves higher classification accuracy across all SNRs compared to using the original data. Given the lower data quantity, it also results in reduced training time compared to the conventional approach. For R-0.5, Epoch Timing shows a training time approximately 410 s faster than Preprocessing Timing. This indicates that applying augmentation during Epoch Timing not only reduces model training time but also improves classification accuracy. The performance differences in RPT Timing for other data sets can be examined through the parentheses in the epoch results. Additionally, the Epoch Timing for R-0.5, where the data quantity is reduced by 50%, demonstrates comparable performance to the original data without augmentation, with a negligible 0.7% difference in average classification accuracy. This highlights that RPT Timing can achieve similar

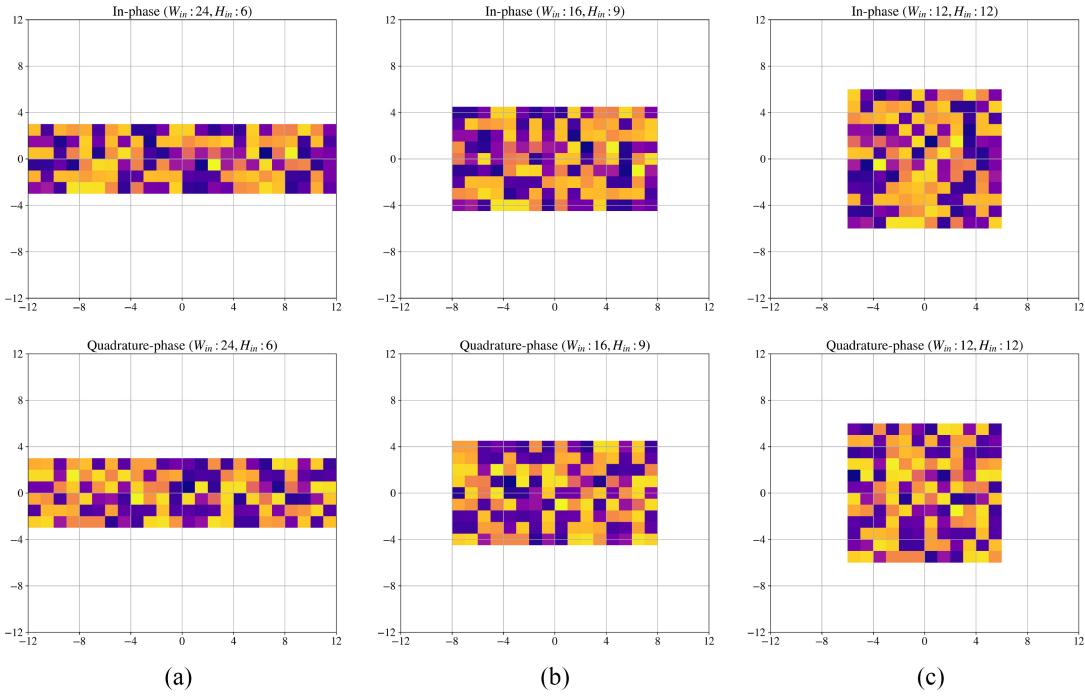


Fig. 3. This figure illustrates the results obtained after applying MuShAug. The shape varies depending on the values of  $W_{in}$  and  $H_{in}$ . (a) Rectangle with  $W_{in}$  set to 24 and  $H_{in}$  set to 6, while (b) shows a rectangle with  $W_{in}$  set to 16 and  $H_{in}$  set to 9. The default shape used in Single Shape is represented by (c). Through MuShAug, both the In-phase and Quadrature-phase serve as a single channel. This can be verified through (3) and (5).

performance with half the data. Furthermore, when comparing timing across the entire data set, all Epoch Timing consistently outperforms Preprocessing Timing, indicating the superiority of Epoch Timing, which is then adopted in the RPT Stage for its excellent performance.

### C. MuShAug Performance Evaluation

MuShAug is a method of transforming signals into various forms, resembling images and beyond. By efficiently utilizing IQ data while preserving the original features of the signal, this technique enables the model to achieve robust generalization and high classification accuracy. Fig. 3 represents the signal after the application of MuShAug.

**Classification Accuracy:** Accurate validation of MuShAug requires performance comparison in a state where RPT is not applied. To achieve this, we conduct experiments using a data set without RPT. Furthermore, we utilize data sets up to R-0.5, which demonstrated performance closest to the original data in Table II. Fig. 4 presents the performance of MuShAug. Single Shape refers to the use of the original sequence signal without applying MuShAug, with both  $W_{in}$  and  $H_{in}$  set to 12. When utilizing the same data set, training with MuShAug demonstrates higher performance across all SNRs compared to training with the conventional Single Shape. This difference is particularly notable as the SNR decreases. At  $-4.11$  dB, employing MuShAug leads to approximately 4% improvement in performance. Similarly, for the R-0.2 data set, a difference of about 1% in performance is observed at  $-2.74$  dB. While performance tended to decrease with lower SNRs, this trend can be overcome by incorporating RPT in subsequent steps.

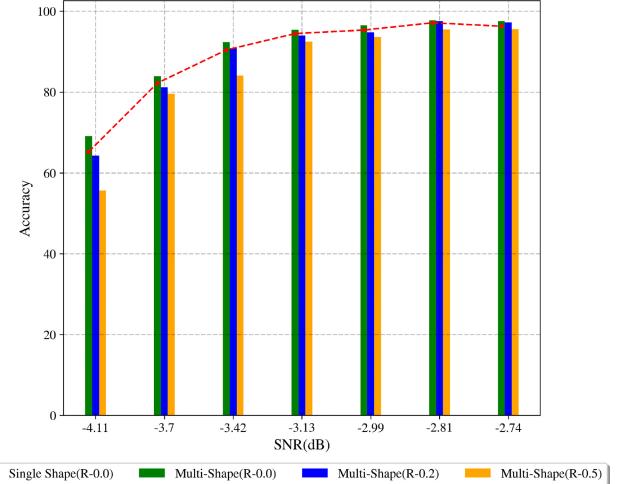


Fig. 4. This graph displays the classification accuracy of Single Shape (R-0.0) and MuShAug for various SNR environments. The plot graph represents the results for Single Shape, while the bar graph illustrates the outcomes for MuShAug.

**Generalization:** We additionally considered the scenario of using a more extensive set of conventional sequence signal to evaluate the model's generalization performance. Generalization is a crucial concept in deep learning, as it assesses a model's ability to effectively predict and classify not only on the training data but also on previously unseen new data. To evaluate this, we monitor the changes in the Loss values for both training and validation data at each epoch, providing insights into the model's generalization capability. Furthermore, we assess generalization performance using cleaner sequence signal than the proposed data in Table IV

**TABLE III**  
PERFORMANCE ANALYSIS OF MUShAug WITH INTEGRATED RPT ON THE INPUT LAYER FOR R-0.75 DATA SET

	Model	SNR(dB) .Acc(%)							
		-2.74	-2.81	-2.99	-3.13	-3.42	-3.7	-4.11	Average
w/o Random Phase Transformation	ResNet18 [46]	67.34	71.39	74.75	69.40	65.20	50.39	23.34	60.26
	InceptionV3[47]	93.16	91.83	86.24	86.84	78.98	45.10	22.80	72.13
	InceptionResNetV2 [48]	91.45	91.84	86.49	84.36	76.56	47.46	17.89	70.86
	<b>MuShAug(w/o RPT)</b>	90.49	91.09	86.03	82.91	78.58	54.01	33.05	73.74
with Random Phase Transformation	ResNet18 [46]	74.56	72.58	68.28	63.64	59.74	12.60	12.93	52.04
	InceptionV3 [47]	91.29	93.73	84.06	82.00	85.75	69.24	14.14	74.31
	InceptionResNetV2 [48]	90.15	94.60	93.60	76.20	86.76	63.68	13.28	74.04
	<b>MuShAug+RPT</b>	95.14( <b>4.65↑</b> )	94.54( <b>3.45↑</b> )	93.26( <b>7.23↑</b> )	91.85( <b>8.94↑</b> )	87.96( <b>9.38↑</b> )	76.86( <b>22.85↑</b> )	41.41( <b>8.36↑</b> )	83.00( <b>9.26↑</b> )

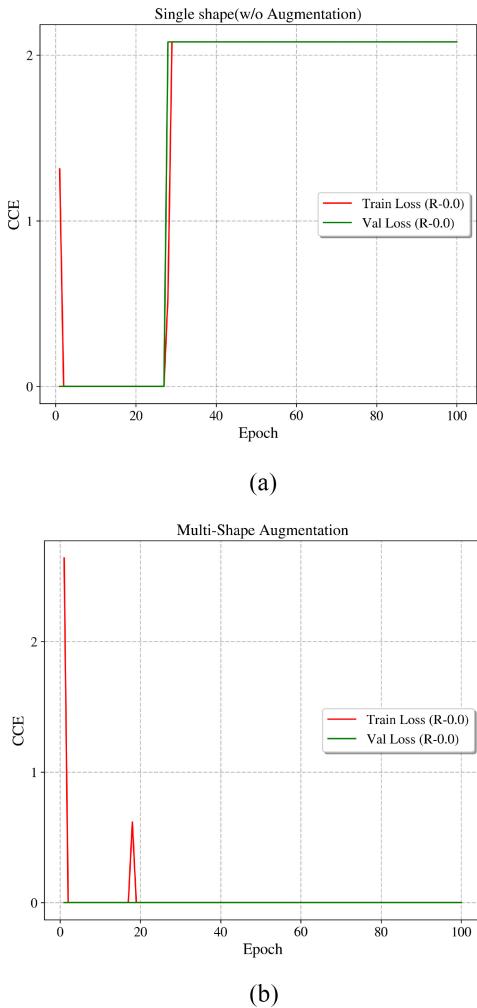


Fig. 5. This graph depicts the variation in loss over epochs, demonstrating the generalization performance of MuShAug on the R-0.0 data set. (a) Represents Single Shape, while (b) corresponds to MuShAug.

in our Appendix. For this purpose, cross-categorical entropy (CCE) is employed in this article. CCE serves as a metric to evaluate how well the model classifies in multiclass classification problems. Fig. 5 illustrates the generalization performance for R-0.0 concerning Single Shape and MuShAug.

When using a single shape, the loss remains consistently low, almost converging to zero until approximately 27 epochs. However, from that point onward, the loss sharply increases,

indicating poor generalization performance. This suggests that the model may struggle with accurate classification of new signals in a real system. Additionally, it demonstrates that using an excessive amount of data to train the model can lead to a decline in generalization performance. On the other hand, employing MuShAug at R-0.0 shows an improvement in generalization performance. Although there is a slightly higher loss in the early epochs of training, stability is achieved from the second epoch onward, maintaining a loss close to zero. The observed increase in loss around the 17th epoch is common, considering the presence of noise that may interfere with the training of signals extracted from real-world environments. Through this analysis, it becomes evident that using MuShAug enhances both classification and generalization performance compared to the conventional approach of using a single shape. Furthermore, it provides insights into the potential for MuShAug in real-world systems, where a variety of factors can occur, can also be seen.

#### D. MuShAug+RPT

**MuShAug+RPT:** To further enhance the performance of the proposed MuShAug, we integrate RPT into the input layer and conduct performance analysis. For accurate validation, we compare existing models, considering that Inception Module is used in MuShAug. We select models with both a regular CNN and an Inception Module, aiming to confirm the augmentation performance with a minimal amount of data in the case of R-0.75. Table III demonstrates the performance of existing models and MuShAug+RPT through R-0.75. Performance comparison before and after combining MuShAug with RPT can be observed within the parentheses next to the MuShAug+RPT results. If we train the model with data where RPT is not applied at the input layer, the existing Inception models show slightly higher accuracy, especially at relatively high SNRs. However, as SNR decreases, the classification accuracy of MuShAug becomes higher. Particularly at -3.43 dB, a performance improvement of about 7% compared to InceptionResNetV2 can be observed, and the average accuracy is also highest when using MuShAug, with a difference of over 1% compared to InceptionV3. When RPT is applied at the input layer, except for ResNet18, all models show improved average accuracy. In particular, the MuSh, combining MuShAug and RPT, demonstrated an

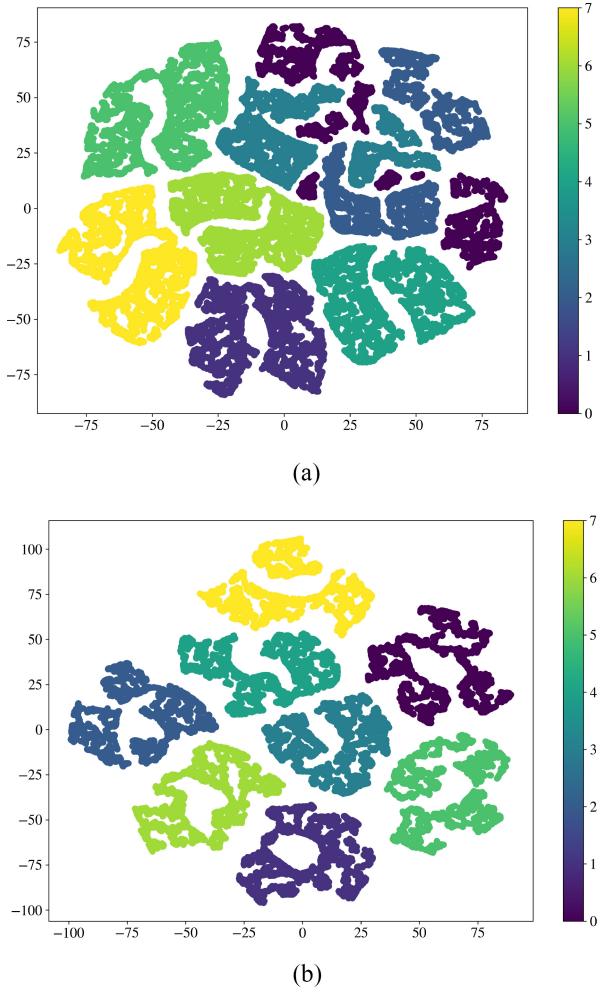


Fig. 6. This graph visualizes features extracted at the low level. (a) Shows the results for the Single Shape without the application of RPT, while (b) shows the results for MuShAug+RPT in R-0.5.

accuracy improvement of approximately 9%, reaching 83% accuracy. Enhanced performance was consistently observed across all SNRs, with around 23% improvement at  $-3.42$  dB. Comparing MuShAug+RPT with existing models, it shows slightly lower accuracy than InceptionResNetV2 at  $-2.74$  and  $-2.81$  dB, but this is a very marginal difference of 0.4%. Except for this, it exhibits very high accuracy performance across all other SNRs. The overall results, including other data sets, can be found in our Appendix.

**Low-Level Feature Visualization:** Considering that features at low levels play a significant role in classifying signals, we extract feature maps at intermediate layers and evaluate performance using t-distributed stochastic neighbor embedding (t-SNE). For this purpose, we compared R-0.0, where no augmentation was applied, with RPT-applied MuShAug using R-0.5. To enhance the reliability of evaluation metrics, we utilized sufficiently clean signals used to validate the loss of MuShAug. Fig. 6 shows the visualization results of features extracted at low levels. Regarding the t-SNE results, we evaluate the performance based on the degree of clustering without considering the values on the x-axis and y-axis. R-0.0 without augmentation shows a reasonable level of clustering, but some

classes have slight distances between them. On the other hand, MuShAug+RPT demonstrates precise clustering for each class. This indicates that using MuShAug+RPT yields better performance in classifying real signals. Additionally, based on the superior performance at the low level of features, it is evident that even with fewer layers, MuShAug+RPT can achieve good performance. Further experiments on the output of MuShAug+RPT at the low level are available in our Appendix.

## V. CONCLUSION

Our results demonstrate that leveraging MuShAug in the model architecture effectively addresses the limitations of conventional image augmentation in sequence signal classification. Furthermore, incorporating RPT has been shown to further enhance the final classification accuracy. Particularly, when compared to previously high-performing models, our approach consistently achieved high accuracy, showcasing precise clustering in the low-level feature extraction area. Additionally, our study exhibits strong adaptability in various real-world 5G environments, indicating its potential for practical system applications. To extend our research, we plan to explore various methods for further enhancing our approach. If a more diverse set of phases is appropriately utilized in the RPT stage, the model can become more robust. Moreover, streamlining the inception module within MuShAug may render it more applicable to real-world systems. Looking ahead, we aim to expand our research into models capable of mitigating signal noise along with MuShAug.

## APPENDIX

### A. Sequence Signal in Real World: Details of Demodulation Reference Signal of 5G Mobile Communication (Section IV-A)

DMRS serves the following purposes to enhance the performance of communications systems.

- 1) It is used in MIMO systems to estimate the channel between multiple antennas and demodulate data. This contributes to the efficiency of data transmission in various transmission and reception methods using multiple antennas.
- 2) DMRS is employed for estimating channel state information. By receiving DMRS, the receiving end can grasp the current wireless channel state, allowing accurate modeling of channel characteristics.
- 3) Ultimately, in wireless communication, DMRS plays a crucial role in compensating for distortions caused by the time-varying frequency response of the channel, thereby improving the quality of the received signal.

To fulfill these roles effectively, DMRS index classification is essential. Through collected DMRS signals in various environments, we validate the effectiveness of MuShAug. The details of the collected DMRS are provided in Table IV.

We collected signals in seven different negative SNR environments. The collected signals are all in the form of IQ data, and each signal consists of 144 complex numbers. Additionally, the default number of collected data is 160 000

TABLE IV  
DETAILS ON DIRECTLY COLLECTED DMRS DATA AND EXPERIMENT  
DATA CONSTRUCTION

	Details
<b>SNR</b>	-2.74 dB, -2.81 dB, -2.99 dB, -3.13 dB, -3.42 dB, -3.7 dB, -4.11 dB
<b>Component of Sequence</b>	$N = 144$ ( $W_{in} \times H_{in} = N$ )
<b>Complex Elements of Data Samples</b>	144 (In-phase), 144 (Quadrature-phase)
<b>Index</b>	8 Integer Index
<b>Number of Sequence Data per Index</b>	20,000 (R-0.0)
<b>Number of Data per SNR Level</b>	20,000 (Sequence) $\times$ 8 (Index) = 160,000 (R-0.0)
<b>Reduction Rate</b>	R-0.0, R-0.2, R-0.5, R-0.75
<b>Train : Validation : Test</b>	6.4 : 1.6 : 2

for each SNR, with eight indices, resulting in 20 000 data points per index.

### B. Model Structure (Section IV-A)

The model utilized in this article is a lightweight version of VGGNet. The specifics for each layer of MuShAug are detailed in Table V.

The convolutional layers utilize kernels of size  $3 \times 3$ , with padding set to “same” to maintain the input and output shapes of the layer. Subsequently, MaxPooling is applied to reduce the size of the input shape. Up to the first MaxPooling layer, feature extraction is output using t-SNE in Section IV-D, and additional results for this can be found in Appendix-D. After the final MaxPooling layer concludes the feature extraction process, each module outputs feature maps with varying shapes. To concatenate all the output feature maps and use them as inputs for the Fully Connected Layer, the flatten function is applied to the last layer of each module, transforming them into 1-D matrices. The final feature map is then created by summing all the output features. Considering that the model structure directly influences feature extraction in the feature extraction area, we designed Inception Modules with the same architecture for all modules.

### C. Inadequacy of Conventional Image Augmentation for Sequence Signal Classification (Section IV-B)

To investigate the suitability of conventional augmentation for sequence signals, we applied image augmentation to DMRS data and evaluated its performance in index classification [3], [49]. Among various image augmentation techniques, we utilized Rotation, which plays a role similar to RPT, with rotation angles set as multiples of  $90^\circ$ , consistent with the random phase specified in this article. Fig. 7 illustrates the results of index classification with augmentation applied based on the rotation angle.

TABLE V  
LIGHTWEIGHT VGGNET ARCHITECTURE USED FOR EXPERIMENTS

Input : Sequence Signal( $R^{W_{in} \times H_{in} \times k}$ ), $k$ : Channel, Number of Multi-Shape : 3			Output Shape		
Layer	Filter	Kernel Size	$W_{in} : 24, H_{in} : 6$	$W_{in} : 16, H_{in} : 9$	$W_{in} : 12, H_{in} : 12$
Conv2D	16	$3 \times 3$	(24, 6, 16)	(16, 9, 16)	(12, 12, 16)
Conv2D	64	$3 \times 3$	(24, 6, 64)	(16, 9, 64)	(12, 12, 64)
MaxPool	1	$2 \times 2$	(12, 3, 64)	(8, 5, 64)	(6, 6, 64)
Conv2D	128	$3 \times 3$	(12, 3, 128)	(8, 5, 128)	(6, 6, 128)
Conv2D	128	$3 \times 3$	(12, 3, 128)	(8, 5, 128)	(6, 6, 128)
MaxPool	1	$2 \times 2$	(6, 2, 128)	(4, 3, 128)	(3, 3, 128)

**Output** : Final Feature Map Matrix for Each Layer

Initially, regardless of the data set, it is evident that image augmentation fails to accurately perform index classification. Particularly, results without augmentation show effective index classification based on SNR. This trend, observed in the graph, is commonly attributed to increased noise interference at lower SNRs, making accurate classification challenging. Conversely, in cases where augmentation is applied during Epoch Timing, lower classification performance is observed even at higher SNRs. This indicates an inability to accurately extract features from the signal. When examining the results for each data set, for R-0.0, it is noted that augmentation, regardless of augmentation timing, degrades classification accuracy. Notably, when using the Epoch Timing similar to RPT, the model exhibits around 45% performance at the lowest SNR of  $-4.11$  dB, highlighting the difference between RPT and Image Augmentation. For R-0.5, Epoch Timing demonstrates a slight improvement of approximately 3% at  $-3.42$  dB, but at  $-3.13$  dB, the performance drops by more than 69%, revealing the model’s unsuitability for Epoch Timing. In the case of Preprocessing Timing, despite directly increasing the number of training data, except for a marginal improvement of about 0.6% at  $-3.13$  dB in R-0.75, all other scenarios exhibited lower performance compared to classification without augmentation. Through these experiments, we have verified that conventional image augmentation techniques are not suitable for sequence signals, indicating the necessity for augmentation techniques tailored specifically for sequence signals.

### D. MuShAug+RPT (Section IV-D)

Before analyzing the performance of MuShAug+RPT, we first examine the performance difference between MuShAug and existing models. This allows us to observe the performance variation with RPT for each model and enhances the reliability of the validation of MuShAug+RPT. Fig. 8 presents the performance analysis of MuShAug and existing models on data sets not shown in Section IV-D. For R-0.0, MuShAug exhibited superior performance at all SNRs except for  $-4.11$  and  $-3.7$  dB. At  $-3.7$  dB, a negligible difference of 0.1% was observed, and models utilizing Inception Modules consistently showed high classification performance. In particular, InceptionResNetV2 showed a difference of over 1% at  $-4.11$  dB. In R-0.2, except for ResNet18, all

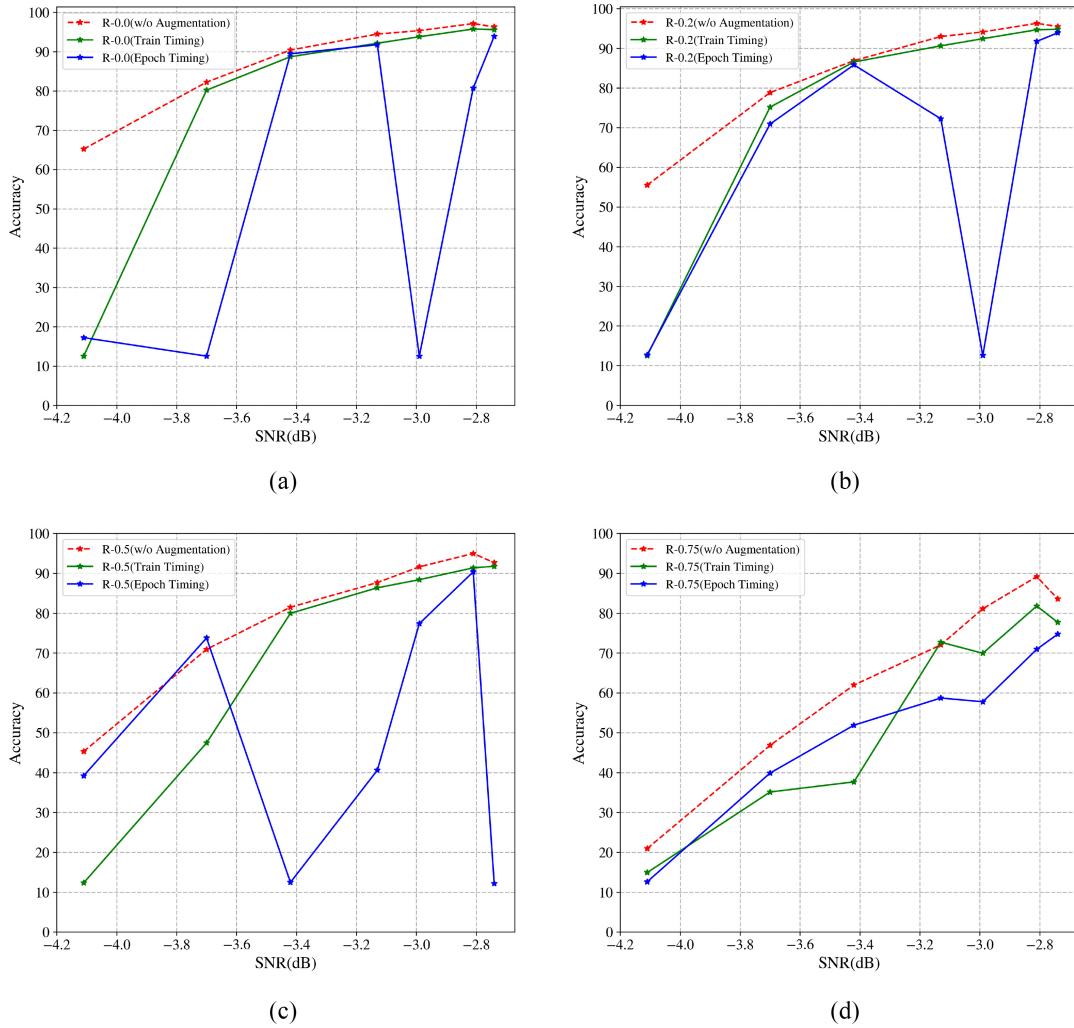


Fig. 7. Performance of conventional image augmentation (rotation) in index classification. (a) R-0.0. (b) R-0.2. (c) R-0.5. (d) R-0.75.

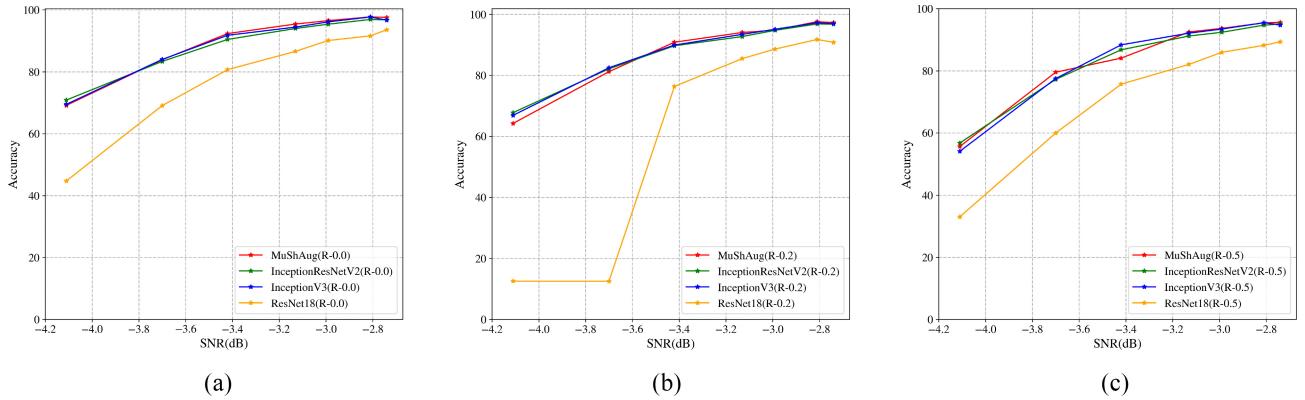


Fig. 8. Performance comparison with existing CNN models without the application of RPT. (a) R-0.0. (b) R-0.2. (c) R-0.5.

models demonstrated classification performance of over 60% at all SNRs. This suggests that using Inception Modules is more effective in improving classification performance than utilizing unique Residual Modules. The same trend was observed in R-0.5. The performance of ResNet18 consistently showed lower classification accuracy compared to models using Inception Modules. For MuShAug, it exhibited higher

accuracy than other Inception Modules at certain SNRs, but this was not consistent across all SNRs. However, this is a challenge that can be addressed by applying RPT. Fig. 9 presents the experimental results combining MuShAug with RPT.

The outstanding classification performance of MuShAug+RPT is evident when examining the overall

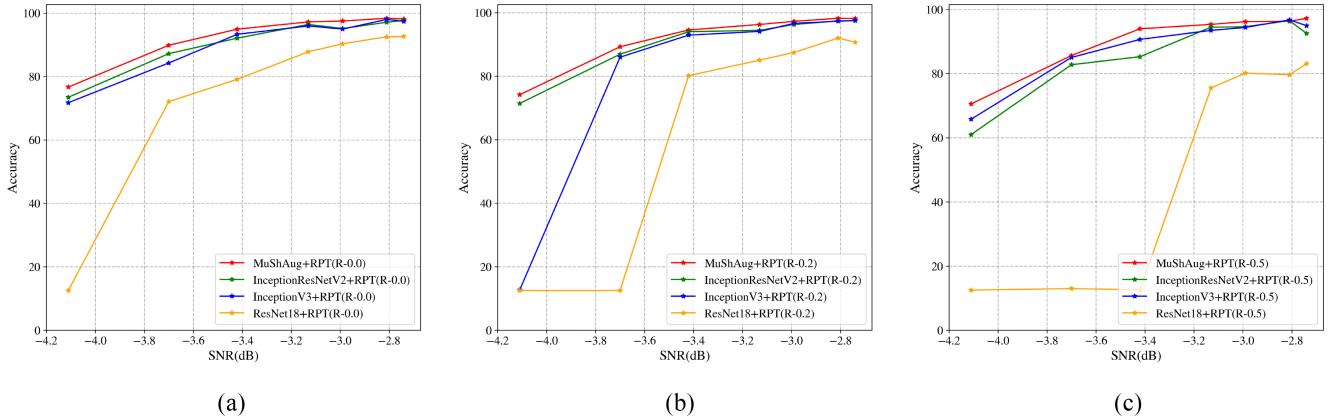


Fig. 9. Performance comparison with existing CNN models applying RPT (Epoch Timing). (a) R-0.0. (b) R-0.2. (c) R-0.5.

TABLE VI  
INDEX CLASSIFICATION PERFORMANCE OF LIGHTWEIGHT MUShAUG+RPT

Dataset	RPT Timing	SNR(dB) .Acc(%)							
		-2.74	-2.81	-2.99	-3.13	-3.42	-3.7	-4.11	Average
R-0.0	None	98.28	98.29	97.48	96.62	93.32	85.97	66.53	90.93
	Pre-Processing	<b>98.68</b>	<b>98.79</b>	<b>98.01</b>	<b>96.68</b>	<b>94.24</b>	87.10	67.34	91.55
	Epoch	98.19	98.59	98.00	96.79	94.80	<b>88.36</b>	<b>74.21</b>	<b>92.71</b>
R-0.2	None	98.43	98.05	96.89	96.09	92.39	84.65	63.54	90.01
	Pre-Processing	<b>98.59</b>	<b>98.43</b>	97.24	96.43	93.03	85.48	64.91	90.59
	Epoch	97.98	98.41	<b>97.34</b>	<b>96.46</b>	<b>93.73</b>	<b>89.51</b>	<b>71.72</b>	<b>92.17</b>
R-0.5	None	90.76	92.76	90.23	88.48	90.24	80.75	57.36	84.37
	Pre-Processing	97.07	<b>97.89</b>	<b>96.84</b>	92.13	85.43	82.24	60.01	87.37
	Epoch	<b>97.71</b>	97.44	96.62	<b>94.81</b>	<b>91.87</b>	<b>85.76</b>	<b>67.03</b>	<b>90.18</b>
R-0.75	None	88.78	89.58	86.03	84.09	78.55	57.43	32.60	73.86
	Pre-Processing	87.65	87.75	86.16	83.16	76.81	58.58	34.29	73.49
	Epoch	<b>96.26</b>	<b>95.18</b>	<b>92.51</b>	<b>92.16</b>	<b>86.89</b>	<b>80.93</b>	<b>56.74</b>	<b>85.81</b>

results. Except for  $-2.81$  dB in R-0.5, MuShAug+RPT consistently exhibits high classification performance across all data sets and SNRs. The only exception was at  $-2.81$  dB in R-0.5, where there was a marginal difference of  $0.5\%$ , and the average accuracy for MuShAug+RPT was the highest for R-0.5. Furthermore, when comparing the performance of R-0.0 MuShAug without RPT in Fig. 8, despite using only half of the data set (R-0.5), MuShAug+RPT surpassed the performance at specific SNRs. At  $-3.7$  and  $-3.42$  dB, it showed approximately a  $1.7\%$  improvement in classification performance. This highlights the robustness and superior performance of MuShAug+RPT, which can achieve high classification accuracy even with a small amount of data.

### E. Lightweight MuShAug+RPT (Section IV-D)

In Section IV-D, we visualized low-level features through t-SNE and verified the performance of MuShAug+RPT in the low-level feature extraction stage. This is a crucial result as it demonstrates the potential for additional lightweighting in MuShAug’s Inception module. To apply the system in real-world scenarios, a more lightweight feature

extraction stage is necessary. For our validated results and our ultimate goal of real-world application, in Section V, we evaluate the performance using a more lightweight MuShAug+RPT. The lightweight MuShAug+RPT extracts features only up to the first Maxpooling layer, as demonstrated in Table V. Table VI presents the classification accuracy performance of lightweight MuShAug+RPT based on RPT Timing.

Setting RPT Timing to Train would directly increase the number of data. Nonetheless, in R-0.75, lightweight MuShAug+RPT(Epoch) consistently exhibited high classification accuracy at all SNRs, showing an average accuracy difference of over 11%. Furthermore, in other data sets, lightweight MuShAug+RPT(Epoch) tends to show slightly lower performance than Preprocessing Timing, with all differences being very minimal, below 1%. However, the average accuracy of lightweight MuShAug+RPT(Epoch) remained higher. This not only confirms the ability to maintain high performance despite lightweighting but also demonstrates the capability to accurately classify signals with a small data set. These results illustrate the potential for lightweighting in MuShAug+RPT and suggest the feasibility of model lightweighting for practical system application.

## REFERENCES

- [1] K. Alomar, H. I. Aysel, and X. Cai, "Data augmentation in classification and segmentation: A survey and new strategies," *J. Imaging*, vol. 9, no. 2, p. 46, 2023.
- [2] B.-B. Jia and M.-L. Zhang, "Multi-dimensional classification via selective feature augmentation," *Mach. Intell. Res.*, vol. 19, no. 1, pp. 38–51, 2022.
- [3] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, 2019.
- [4] R. Shen, S. Bubeck, and S. Gunasekar, "Data augmentation as feature manipulation," in *Proc. 39th Int. Conf. Mach. Learn.*, 2022, pp. 19773–19808.
- [5] J. Li, W. Qiang, C. Zheng, B. Su, and H. Xiong, "MetAug: Contrastive learning via meta feature augmentation," in *Proc. 39th Int. Conf. Mach. Learn.*, 2022, pp. 12964–12978.
- [6] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," 2017, *arXiv:1712.04621*.
- [7] B. Tang, Y. Tu, Z. Zhang, and Y. Lin, "Digital signal modulation classification with data augmentation using generative adversarial nets in cognitive radio networks," *IEEE Access*, vol. 6, pp. 15713–15722, 2018.
- [8] O. A. Dobre, "Signal identification for emerging intelligent radios: Classical problems and new challenges," *IEEE Instrum. Meas. Mag.*, vol. 18, no. 2, pp. 11–18, Apr. 2015.
- [9] J. Nie, Y. Zhang, Z. He, S. Chen, S. Gong, and W. Zhang, "Deep hierarchical network for automatic modulation classification," *IEEE Access*, vol. 7, pp. 94604–94613, 2019.
- [10] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 3, pp. 433–445, Sep. 2018.
- [11] W. Zhang, M. Feng, M. Krantz, and A. H. Y. Abyaneh, "Signal detection and classification in shared spectrum: A deep learning approach," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2021, pp. 1–10.
- [12] K. Tekbiyik, Ö. Akbunar, A. R. Ekti, A. Görçin, G. K. Kurt, and K. A. Qaraqe, "Spectrum sensing and signal identification with deep learning based on spectral correlation function," *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 10514–10527, Oct. 2021.
- [13] J. Kim, S. Kang, and O. Jo, "Lightweight data processing scheme based on machine learning for 5G DMRS index classification," *Asia-Pac. J. Converg. Res. Interchange*, vol. 9, no. 11, pp. 91–101, 2023.
- [14] F.-L. Luo, "Signal processing techniques for 5G: An overview," *ZTE Commun.*, vol. 13, no. 1, pp. 20–27, 2015.
- [15] A. M. Elbir, "DeepMUSIC: Multiple signal classification via deep learning," *IEEE Sensors Lett.*, vol. 4, no. 4, pp. 1–4, Mar. 2020.
- [16] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 168–179, Feb. 2018.
- [17] S. Kang, T. Lee, J. Kim, A.-R.-S. Lee, J. Kim, and O. Jo, "Intelligent index classification method based on machine learning for detection of reference signal in 5G networks," *IEEE Access*, vol. 11, pp. 100810–100822, 2023.
- [18] Y. Liu, S. Yan, L. Leal-Taixé, J. Hays, and D. Ramanan, "Soft augmentation for image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 16241–16250.
- [19] L. Li and A. Li, "A2-Aug: Adaptive automated data augmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 2266–2273.
- [20] C. Gong, D. Wang, M. Li, V. Chandra, and Q. Liu, "KeepAugment: A simple information-preserving data augmentation approach," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1055–1064.
- [21] M. Xu, S. Yoon, A. Fuentes, and D. S. Park, "A comprehensive survey of image augmentation techniques for deep learning," *Pattern Recognit.*, vol. 137, May 2023, Art. no. 109347.
- [22] G. Dong and H. Liu, "Signal augmentations oriented to modulation recognition in the realistic scenarios," *IEEE Trans. Commun.*, vol. 71, no. 3, pp. 1665–1677, Mar. 2023.
- [23] J. Kim and O. Jo, "IncepSeqNet: Advancing signal classification with multi-shape augmentation (student abstract)," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 23542–23543.
- [24] X. Xu et al., "Mixing signals: Data augmentation approach for deep learning based modulation recognition," 2022, *arXiv:2204.03737*.
- [25] Q.-V. Pham, N. T. Nguyen, T. Huynh-The, L. B. Le, K. Lee, and W.-J. Hwang, "Intelligent radio signal processing: A survey," *IEEE Access*, vol. 9, pp. 83818–83850, 2021.
- [26] Z. Chen et al., "SigNet: A novel deep learning framework for radio signal classification," *IEEE Trans. Cogn. Commun. Netw.*, vol. 8, no. 2, pp. 529–541, Jun. 2022.
- [27] F. Wang, T. Shang, C. Hu, and Q. Liu, "Automatic modulation classification using hybrid data augmentation and lightweight neural network," *Sensors*, vol. 23, no. 9, p. 4187, 2023.
- [28] L. Huang, W. Pan, Y. Zhang, L. Qian, N. Gao, and Y. Wu, "Data augmentation for deep learning-based radio modulation classification," *IEEE Access*, vol. 8, pp. 1498–1506, 2020.
- [29] W. Zhu, S. M. Mousavi, and G. C. Beroza, "Seismic signal augmentation to improve generalization of deep neural networks," *Adv. Geophys.*, 2020, vol. 61, pp. 151–177, Jan. 2020.
- [30] J. Zhang et al., "Data augmentation and dense-LSTM for human activity recognition using WiFi signal," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4628–4641, Mar. 2021.
- [31] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.
- [32] I. Kim, Y. Kim, and S. Kim, "Learning loss for test-time augmentation," in *Proc. 34th Adv. Neural Inf. Process. Syst.*, 2020, pp. 4163–4174.
- [33] X. Zhang et al., "Radar signal intrapulse modulation recognition based on a denoising-guided disentangled network," *Remote Sens.*, vol. 14, no. 5, p. 1252, 2022.
- [34] M. L. Xue, M. Huang, J. J. Yang, and J. C. Chen, "MICNet: A hybrid model based on inception network and CNN for automatic modulation classification," in *Proc. Int. Conf. Wireless Commun. Smart Grid (ICWCSG)*, 2021, pp. 331–335.
- [35] M. Z. Mumtaz, M. Khurram, M. Adnan, and A. Fazil, "Autonomous modulation classification using single inception module based convolutional neural network," in *Proc. Int. Bhurban Conf. Appl. Sci. Technol. (IBCAST)*, 2021, pp. 966–973.
- [36] J. H. Lee, K.-Y. Kim, and Y. Shin, "Feature image-based automatic modulation classification method using CNN algorithm," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAICC)*, 2019, pp. 1–4.
- [37] M. Liu, G. Liao, N. Zhao, H. Song, and F. Gong, "Data-driven deep learning for signal classification in industrial cognitive radio networks," *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, pp. 3412–3421, May 2021.
- [38] J. G. Andrews, "A primer on Zadoff Chu sequences," 2023, *arXiv:2211.05702*.
- [39] J. Y. Han, O. Jo, and J. Kim, "Exploitation of channel-learning for enhancing 5G blind beam index detection," *IEEE Trans. Veh. Technol.*, vol. 71, no. 3, pp. 2925–2938, Mar. 2022.
- [40] S. W. Kang and O. Jo, "Multivariate time-series imagination with time embedding in constrained environments (student abstract)," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 23535–23536.
- [41] D. Kumar, D. Sharma, and R. Goecke, "Feature map augmentation to improve rotation invariance in convolutional neural networks," in *Proc. 20th Int. Conf. Adv. Concepts Intell. Vis. Syst.*, 2020, pp. 348–359.
- [42] C. K. Chui, S.-B. Lin, and D.-X. Zhou, "Deep neural networks for rotation-invariance approximation and learning," *Anal. Appl.*, vol. 17, no. 5, pp. 737–772, 2019.
- [43] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015, *arXiv:1409.1556*.
- [44] X. Hou and H. Kayama, *Demodulation Reference Signal Design and Channel Estimation for LTE-Advanced Uplink*. London, U.K.: IntechOpen, 2011.
- [45] U. Muhammad, W. Wang, S. P. Chattha, and S. Ali, "Pre-trained VGGNet architecture for remote-sensing image scene classification," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, 2018, pp. 1622–1627.
- [46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [47] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826.
- [48] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4278–4284.
- [49] M. Elgendi et al., "The effectiveness of image augmentation in deep learning networks for detecting COVID-19: A geometric transformation perspective," *Front. Med.*, vol. 8, Mar. 2021, Art. no. 629134.



**Jongseok Kim** (Student Member, IEEE) received the bachelor's degree in science engineering from Chungbuk National University, Cheongju, South Korea, in 2022, where he is currently pursuing the master's degree in computer science.

During his undergraduate years, he worked as a Research Assistant with Prof. Ohyun Jo's Information System Laboratory, Chungbuk National University. His research interests include signal recognition and estimation, signal generation for denoising, data augmentation optimization, and lightweight model optimization.



**Ohyun Jo** (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2005, 2007, and 2011, respectively.

He is currently an Associate Professor with the Department of Computer Science, Chungbuk National University, Cheongju, South Korea. From April 2011 to February 2016, he was with Samsung Electronics, Suwon, South Korea, in charge of research and development for future wireless communication systems, applications, and services. From March 2016 to July 2017, he was a Senior Researcher with Electronics and Telecommunications Research Institute, Daejeon, and from August 2017 to February 2018, he was an Assistant Professor with the Department of Electrical Engineering, Korea Military Academy, Seoul, South Korea. He has authored and co-authored more than 90 papers, and holds more than 170 registered and filed patents. His research interests include machine learning applications, next-generation mobile communication systems, military communications, Internet of Things, future wireless solutions/applications/services, and embedded communications ASIC design.

Dr. Jo was the recipient of numerous recognitions, including the Gold Prize in Samsung Annual Award, the Most Creative Researcher of the Year Award, the Best Mentoring Award, the Major Research and Development Achievement Award, and the Best Improvement of Organization Culture Award.