IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 October 2023, accepted 9 December 2023, date of publication 18 December 2023,
date of current version 27 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3343738

RESEARCH ARTICLE

Speech Enhancement Using Dynamic Learning in Knowledge Distillation via Reinforcement Learning

SHIH-CHUAN CHU, CHUNG-HSIEN WU, (Senior Member, IEEE), AND TSAI-WEI SU
Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan 70101, Taiwan

Corresponding author: Chung-Hsien Wu (chunghsienwu@gmail.com)

This work was supported by the National Science and Technology Council of Taiwan under Contract 111-2221-E-006-150-MY3.

ABSTRACT In recent years, most of the research on speech enhancement (SE) has applied different strategies to improve performance through deep neural network models. However, as the performance improves, the memory resources and computational requirements of the model also increase, making it difficult to directly apply them to edge computing. Therefore, various model compression and acceleration techniques are desired. This paper proposes a learning method that dynamically uses Knowledge Distillation (KD) to teach a small student model from a large teacher model by considering the learning ratio from the teacher's output and the real target based on reinforcement learning (RL). During the KD learning process, RL is adopted to estimate the learning ratio by considering the reward favoring the hard target (clean speech) or the soft target (the output of the teacher model) during the training of KD. The proposed method results in a more stable training process for the resulting smaller SE model and yields improved performance. In the experiment, we used the TIMIT and CSTR VCTK datasets and evaluated two representative SE models that employ different loss functions. On the TIMIT dataset, when we reduced the number of parameters in the Wave-U-Net student model from 10.3 million to 2.6 million, our method performed better than non-KD models with improvements of 0.05 in PESQ, 0.1 in STOI, and 0.47 in the scale-invariant signal-to-distortion ratio. Moreover, by utilizing prior knowledge from the pre-trained teacher model, our method effectively guided the learning process of the student model, achieving excellent performance even under low SNR conditions. Furthermore, we use Conv-Tasnet to further validate our proposed method. Finally, for ease of comparison, we conducted a comparison on the VCTK dataset as well.

INDEX TERMS Deep learning, speech enhancement, knowledge distillation, reinforcement learning.

I. INTRODUCTION

Digital speech signals can be seen everywhere in our daily lives. With the accelerated pace of modern life, the mode of human-computer interaction is gradually shifting from the traditional keyboard to the touch panel. It is believed that in the near future, speech control will emerge as the primary method of human-computer interaction. However, noise permeates real-life environments, which not only hampers interpersonal communication but also significantly affects the accuracy of electronic products that rely on speech as an input modality. Fortunately, speech enhancement (SE) algorithms can enhance these speech signals, improving recognition rates and clarity. As a result, the significance of SE and noise reduction in speech-based applications has been increasingly recognized in recent years.

In terms of SE, many deep learning models have been proposed with various architectures. These models can be roughly divided into two categories: mask-based [1], [2], [3], [43], [44] and direct mapping [4], [5], [6], [45], [46]. The former employs a mask-based approach, typically transforming the time-domain signal into a frequency-domain signal using STFT (Short Time Fourier Transform). After element-wise product with the estimated mask, the signal is restored to

The associate editor coordinating the review of this manuscript and approving it for publication was Lin Wang.

the time-domain through iSTFT (Inverse Short Time Fourier Transform) or enhanced in an end-to-end manner, like Conv-TasNet [1]. The latter utilizes a direct mapping technique to estimate the spectrum or time-domain signal. To further enhance the SE performance, various loss functions have been proposed, ranging from common mean square error (MSE) and scale-invariant signal-to-distortion ratio (SISDR) [7] to loss calculation through acoustic models [2], [8], [9], [10], and even multi-task training incorporating multiple deep features [11], [12]. However, with the progress of the previously proposed methods, the amount of calculations and memory size required is getting larger and larger, and how to effectively use the trained model becomes an important issue. Knowledge Distillation (KD) is a method for compressing deep learning models, commonly applied in image classification, which sets it apart from other compression techniques like quantization [13], [14], [15] and network pruning [16], [17], [18]. KD leverages large and complex pre-trained models to facilitate the training of smaller and lighter models. The goal is to achieve better performance with the small model compared to non-KD methods or to match and potentially surpass the performance of the large model, while maintaining the same model size. Hinton et al. [19] introduced the concept of using a large model as a teacher model to guide the learning process of a student model. The output of the teacher model serves as one of the learning objectives for the student, referred to as a soft goal. This soft goal is combined with the original hard goal and incorporated into the student's loss function. As a result, the knowledge from the teacher model can be refined and transferred to the student model, improving the overall learning direction. This approach is known as the teacher-student (T-S) structure.

However, experiments conducted in [20] revealed that the teacher model can only provide assistance once the knowledge has been refined, regardless of whether the teacher model has more or fewer parameters than the student model. In recent years, there have been studies exploring the application of KD in the field of SE [21], [22], [23], [24], [25], [26], [27], [28]. While KD has shown great success in image classification tasks, several variants, such as feature mimic [23], [24] and self-KD [25], [26], have been derived and proven effective. However, when it comes to the SE task, the improvements have not been as significant. In our initial experiments, we attempted to apply KD-based SE methods inspired by feature mimic and self-KD. Unfortunately, these methods did not yield satisfactory results for speech signals with low signal-to-noise ratios (SNR) or certain types of noise. This led us to realize that distilling the SE task layer by layer, as done in self-KD and feature mimic for image classification, poses significant challenges. Previous studies [20], [29], [30] have indicated that KD should not learn from well-trained teachers, and contrarily teachers which do not fully converge could achieve a better performance. This problem is called the Teacher Identity Problem (TIP). However, most of the studies using KD in SE tasks mentioned above are quite difficult to reproduce the results due to TIP, or to compare them directly. So we put forward a new hypothesis: the learning process of the student model is the same as the human learning process, and both require step-by-step teaching materials. We apply this concept to KD and attempt to make incremental changes to the student's training objective to address the above difficulties. Accordingly, this study directly learns the teacher's output with reference to the T-S architecture except that we use the teacher's output as the student's target. Considering that saving too many checkpoints in the process of training a teacher is resource-intensive and difficult to fit other large pre-trained models, we use a dynamic learning ratio on the trained teacher output to simulate the underperforming teacher.

Finally, we have decided to adopt the T-S architecture to directly learn the time domain enhanced output from the teacher. Unlike most of the previously mentioned KD methods, where the learning ratio between soft and hard targets remains fixed, our approach incorporates a sample-based dynamic learning ratio. This dynamic learning ratio takes into account the SNR and noise type of each input speech sample, implicitly contributing to the enhancement of SE performance. In this study, we propose a dynamic knowledge distillation method based on reinforcement learning (RL) [9], [31], [32], [33]. By considering the output of both the teacher and student models in each sample-by-sample training step as a state, the RL model selects the corresponding action, which determines the learning ratio for that specific training sample. Through the designed reward function, the KD process can dynamically determine the learning ratio between soft and hard targets for each data sample during the training of the SE model. This dynamic approach ensures that the SE model achieves the most appropriate learning target for the data, considering different SNRs and noise types.

The two main contributions of this article are as follows:

- We propose a KD method that can be applied to a wide range of models. The first contribution is aimed at avoiding constraints on the SE architecture and reducing model size without the need for additional training data or large pre-trained model conditions.
- We provide dynamic teaching materials for KD training to address the issue of TIP.

The method we propose is versatile and can be applied to most existing SE methods. The experiments in this paper demonstrate its effectiveness in significantly reducing model size with minimal performance degradation or even improvements in some cases.

II. RELATED WORK

In recent research on the SE task, several methods have been proposed with impressive results using large models for noise reduction. These methods often utilize features that involve converting speech into images and employ complex training approaches. Examples of such methods include DCUNet [34], DB-AIAT [35], and CMGAN [36].

The Unet, Transformer, and Conformer architectures are widely recognized as effective methods or models, but they require considerable computational resources. Other studies have focused on adjusting the objective functions commonly used in current models, such as PFPL [2], PFPL-AE [11], and MetricGAN+ [10]. In PFPL and PFPL-AE, high-dimensional features generated by wav2vec 2.0 [37] are used as training indicators to improve the quality of noise reduction. PFPL-AE is a variant that incorporates an ensemble method by adding more pre-trained models (including acoustic, emotion, speaker, etc.) to calculate the loss and further enhance SE performance. MetricGAN+ utilizes Quality-Net [38] to directly estimate the PESQ score, replacing the role of the discriminator in Generative Adversarial Networks (GAN). From the aforementioned methods, it is evident that in addition to directly improving model performance, many studies leverage pre-trained models as additional knowledge extraction tools to further enhance performance. Although there are numerous pre-trained acoustic and SE models available for noise reduction assistance, it is known that out-of-training noise can still significantly impact model performance. Therefore, to effectively utilize existing knowledge extraction tools and SE models that cover different types of noise, end-to-end KD techniques adapted from image classification have started to gain attention.

In terms of model compression, studies [24], [25] have shown a common practice in KD, where multiple teachers are trained using sub-band and SNR information, enabling students to select the appropriate learning targets based on the input during the training process. Additionally, a low-latency SE method [26], [27] has been derived to achieve faster processing speed. The online version of this method utilizes a shorter input time scale, often employing recurrent modules to address the input time scale issue. Reference [26] employs the online model as a student to learn from the offline teacher through KD training, while [27] combines two model compression techniques: first purifying and fine-tuning the weight matrix of the SE model, followed by quantization of the weight matrix to further accelerate operations. However, it should be noted that many KD methods in the TIP field are challenging to reproduce or exhibit instability during training, leading to significant variations in results depending on the selected learning targets.

III. PROPOSED METHOD

To address the issues raised above, we propose a method called Knowledge Distillation via Reinforcement Learning (KDRL), which is a KD-based learning structure for SE tasks. The training process of KDRL is divided into two parts. First, the noisy speech is fed to the student model after KD learning, which is carried out based on the learning ratio obtained from the policy network. Since there is no specified adjustment method for the learning ratio of the policy network, we introduce two "reference models" with the same student architecture and use the parameters before KD learning as their initial weights. During KD training, both the reference models and the student model perform the same tasks. The difference is that the reference models are updated with extreme values (0 and 1) of the KD learning ratio, respectively. The speech quality of the student and reference models is compared to serve as the basis for adjustment. Second, the mixture is re-passed to the student and reference models after their respective updates, and the policy network adjusts the learning ratio based on the new results from each model. A designed reward function is used to evaluate the quality difference before and after the update, which is directly used as the loss for adjusting the learning ratio. This enables schedule adjustment of the KD learning objective based on the speech samples under different noise conditions. In the following subsections, we elaborate on the three main modules of KDRL: speech enhancement, KD learning ratio estimation, and reinforcement learning-based KD.

A. SPEECH ENHANCEMENT MODEL

Let us denote the time-domain monaural noisy speech as X, the clean speech as Y, and the noise signal as N. The mixture of clean speech and noise can be expressed by (1). The goal of the SE model is to predict the clean speech Y based on the input noisy speech X. In this study, we evaluate two end-to-end architectures for speech enhancement: Wave-U-Net [6] and Conv-TasNet. The loss function used in Wave-U-Net is the mean squared error (MSE), while Conv-TasNet uses the scale-invariant signal-to-distortion ratio (SI-SDR). To enhance the speech enhancement capabilities of the student model, we utilize a well-trained teacher model to guide the learning process. The student architecture is designed to reduce computational requirements by reducing the number of layers, blocks, or kernels compared to the teacher model. During the training phase, the teacher model is used for KD learning. In the testing phase, only the student model is used to reduce the noise in the input speech. By leveraging the knowledge and expertise of the teacher model, the student model can benefit from improved speech enhancement performance. The use of KD allows the student model to learn from the teacher model's outputs and optimize its own performance.

$$X = Y + N \tag{1}$$

B. KD LEARNING RATIO ESTIMATION MODEL

The outputs of the teacher and student models are denoted by Y_S and Y_T, respectively, and the clean speech source is Y_C. For KD learning ratio estimation, a policy network P_{net} is constructed. The input state $\mathbb{Z}$ of the P_{net} is the combination of the difference between the student and the target and the difference between the teacher and the target, as shown in (2)

$$\mathbb{Z} = (Y_C - Y_S) \oplus (Y_T - Y_S) \tag{2}$$

where $\oplus$ means concatenation. The architecture of P_{net} is shown in Figure 1. The input first goes through two convolutional layers. After the result is flattened, four fully connected layers are connected. Finally, we use the sigmoid function

TABLE 1. Layer parameters for the applied policy network.

Model Structure		Input Dimension (L×C)	Kernal Stride/Padding	Kernal Dimension (L×C)	Output Dimension (L×C)
Input		16009×2			
CNN Layer 1	1-D Conv	16009×2	1/0	6×32	16004×32
	1-D BN	16004×32			16004×32
	1-D MaxPool	16004×32	2/0		4001×32
	Relu	4001×32			4001×32
CNN Layer 2	1-D Conv	4001×32	1/0	6×32	3996×4
	1-D BN	3996×4			3996×4
	1-D MaxPool	3996×4	2/0		999×4
	Relu	999×4			999×4
Fully Connected 1		999×4			1024
Fully Connected 2		1024			128
Fully Connected 3		128			32
Fully Connected 4 & Sigmoid		32			1

FIGURE 1. Structure of the policy network.

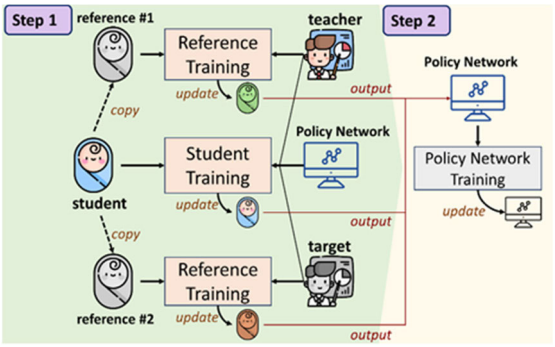


FIGURE 2. Conceptual framework of the proposed SE method.

to limit the output range from 0 to 1 and get the learning ratio α for the SE model training as shown in (3), and the layer parameters for policy network application as shown in Table 1.

$$\alpha = \sigma\left(P_{net}\left(\mathbb{Z}\right)\right) \tag{3}$$

C. REINFORCEMENT LEARNING-BASED KD FOR SE

The training process of KDRL is divided into two steps: SE model training and policy network training, as depicted in Figure 2. The system comprises a teacher model, a student model, and two reference models. The teacher model is pre-trained, and the student model is the one we aim to train. In the first step, which is the SE model training, we employ the policy network to estimate the KD learning ratio. This ratio is then used to weight the model loss, resulting in the SE loss for student training, as illustrated in Figure 3. Simultaneously, the

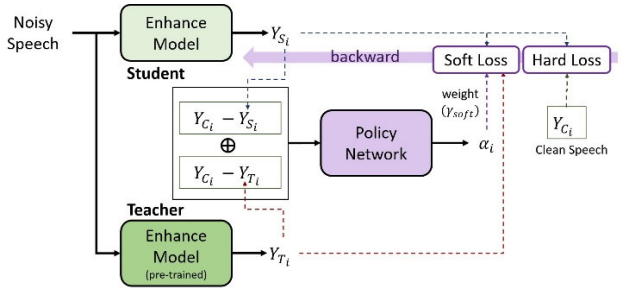


FIGURE 3. The block diagram for backward propagation of the SE loss.

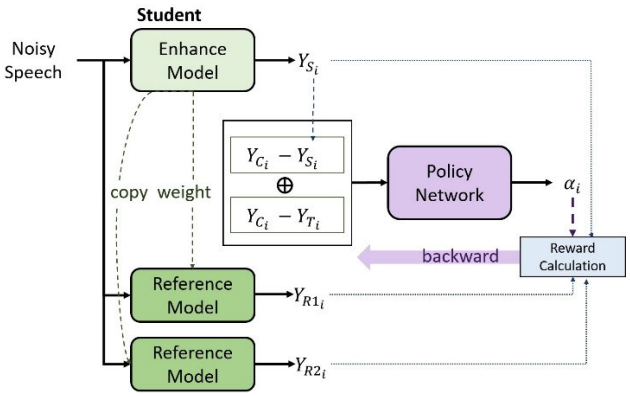


FIGURE 4. The block diagram for backward propagation of the loss by the policy network.

mixture is also passed to the two reference models. Similar to the student training process, the reference models are trained using either soft or hard targets as their training objectives, respectively, rather than a proportional mixture of both.

The second step involves training the policy model. The reward for a given learning ratio α is determined based on the performances of the student and reference models. The policy network is then adjusted based on the reward. As a result, the reinforcement learning model can determine whether α should approach 1 or 0 based on the improvement in speech quality, as shown in Figure 4. The overall block diagram of

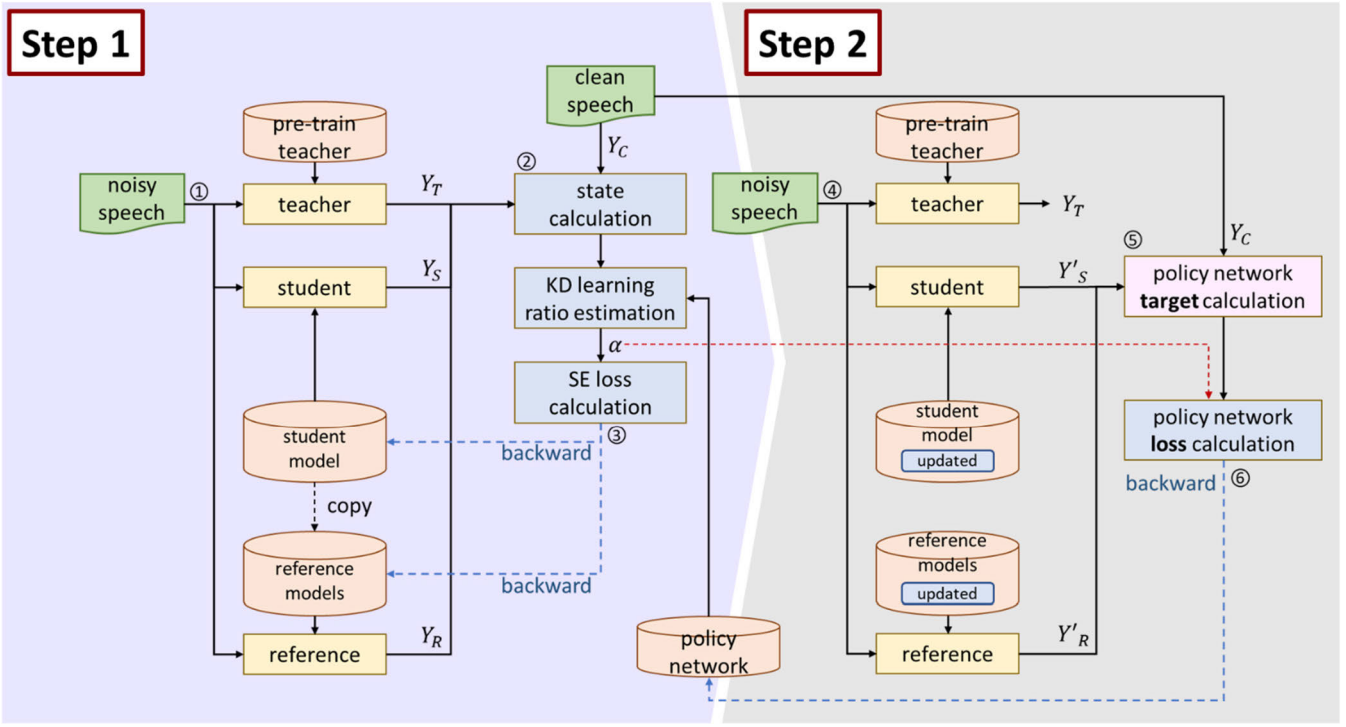


FIGURE 5. The block diagram for system training.

the system is presented in Figure 5. Detailed training procedures for each component will be discussed in the subsequent subsections.

1) SE MODEL TRAINING

First, the SE model training process involves copying the parameters of the student model at the beginning of each iteration and passing them to two reference models with the same architecture. This action ensures that the two models start from the same initial point. The learning process then proceeds with different proportions of goals. Finally, this method compares the results with the student model output based on the current updated results, as depicted in the left half of Figure 5.

Next, we utilize the training set to train the model. In order to verify that the proposed method can be applied to different loss functions, two SE models with representative losses are selected and expressed as (4). Subsequently, we employ the noisy input to obtain the enhanced result from each model: the teacher model output Y_T, the student model output Y_S, and the reference model output Y_R. At this point, we have the soft target Y_T and the hard target Y_C. The two reference models can then calculate their respective losses. One of the reference models is trained entirely using the soft target, with its loss function denoted as L_{R1} in (5), while the other reference model is trained entirely using the hard target, with the loss function denoted as L_{R2} in (6).

$$D = \begin{cases} MSE, & for\ Wave-U-Net \\ -SISDR, & for\ Conv-TasNet \end{cases} \tag{4}$$

$$L_{R1_i} = \frac{1}{B}\sum_{i=1}^{B}\left[\gamma_{hard}\cdot D\left(Y_{C_i}, Y_{R1_i}\right)\right] + \gamma_{soft}\cdot D\left(Y_{T_i}, Y_{R1_i}\right), \tag{5}$$

$$L_{R2_i} = \frac{1}{B}\sum_{i=1}^{B}\left[\gamma_{hard}\cdot D\left(Y_{C_i}, Y_{R2_i}\right)\right] + \gamma_{soft}\cdot D\left(Y_{T_i}, Y_{R2_i}\right), \tag{6}$$

where γ_{hard} and γ_{soft} respectively represent the ratios of two terms in the loss function. For L_{R1_i}, γ_{hard} is set to 0 and γ_{soft} is set to 1. For L_{R2_i}, γ_{hard} is set to 1 and γ_{soft} is set to 0. B is the batch size.

Because the most suitable proportional relationship between these two terms in the loss function for each data sample is unknown, the third step involves using the policy network as a tutor to obtain the KD (Knowledge Distillation) learning ratio. After α is estimated, the SE loss for the student model is defined as (7).

$$L_{S_i} = \frac{1}{B}\sum_{i=1}^{B}\left[\gamma_{hard}\cdot D(Y_{C_i}, Y_{S_i})\right] + \gamma_{soft}\cdot D(Y_{T_i}, Y_{S_i}),$$
$$\begin{cases} Method\ A: \gamma_{hard} = 1-\alpha_i, & \gamma_{soft} = \alpha_i, \\ Method\ B: \gamma_{hard} = \alpha_i, & \gamma_{soft} = 1, \\ Method\ C: \gamma_{hard} = 1, & \gamma_{soft} = \alpha_i \end{cases} \tag{7}$$

Then, based on the value of α, three weighting methods for KD learning are defined: Methods A, B, and C. Method A corresponds to the traditional KD method known as the quantitative transfer method. This approach involves a quantitative loss calculation, where the sum of the weights assigned

to the soft target γ_{soft} and the hard target γ_{hard} is equal to 1. Methods B and C represent directional approaches that focus on specific learning directions. For instance, Method B primarily relies on the soft target, while Method C is biased towards the hard target. By leveraging the respective directions, KD is employed to adjust the training direction and facilitate the transfer of dark knowledge from the targets. These methods offer an advantage over the traditional approach. Among the three methods, Method A exhibits significant fluctuations in the learning curve throughout the training process, making it challenging to achieve convergence on the SE task. Conversely, Methods B and C demonstrate smoother learning curves in their respective training directions, resulting in a more stable overall training process.

2) POLICY NETWORK TRAINING

In the SE training process, the success of the entire KD learning relies heavily on the appropriateness of the estimated α from the policy network. To train the policy network, we calculate the reward for α by assessing the performance of both the student and reference models. It is important to note that the objective of the policy network is to maximize the reward or penalize by taking the value of α as the KD learning ratio in this training process.

The reward is computed by evaluating the quality of the output from the student and reference models using the distance function D in equation (4). The increase or decrease in the evaluation score directly corresponds to an increase or decrease in the value of α. Hence, the reward R is defined as shown in equation (8).

$$R_i = \begin{cases} D_{R1_i} - D_{S_i}, & if\ D_{R1_i} < D_{R2_i} and\ D_{S_i}, \\ D_{S_i} - D_{R2_i}, & if\ D_{R2_i} < D_{R1_i} and\ D_{S_i}, \\ 0, & if\ D_{S_i} < D_{R1_i} and\ D_{R2_i}. \end{cases} \tag{8}$$

As the reward R may exhibit significant fluctuations in the later stages of training, we introduce a constraint value δ. This value is defined as the amount of progress, representing the percentage of the original loss value R_i, as depicted in equation (9). To ensure that the value remains within the range of 0 to 1, we impose a limit on δ_i, as shown in equation (10). The loss of the policy model is then expressed as equation (11), where $\|\bullet\|_{sg}$ denotes the stop gradient operation.

$$\delta_i = \frac{R_i}{D\left(Y_{Ci}, Y_{S_i}\right)} \cdot \varepsilon, where\ \varepsilon = \frac{1}{number\ of\ epochs}, \tag{9}$$

$$\delta_{ci} = \min\left(\max\left(\|\alpha_i\|_{sg} + \delta_i, 1\right), 0\right) \tag{10}$$

$$Loss_i = \left(\alpha_i - \delta_{ci}\right)^2. \tag{11}$$

IV. EXPERIMENTS

A. DATASETS

This study utilized two datasets for model training and evaluation, with audio files uniformly sampled at 16 kHz in all experiments. The first dataset is a combination of DARPA-TIMIT [39] and NoiseX-92 [40]. The clean speech source is derived from the TIMIT dataset, which includes 630 speakers, each speaking ten sentences. The NoiseX-92 dataset was used as the source for noise, consisting of 15 audio files, each containing a different type of noise. In this study, the training data comprised the first 250 sentences from the DARPA-TIMIT training data subset. Five noise types from NoiseX-92 were selected, namely babble, destroyerops, f16, pink, and volvo. Five SNR levels were used: −10dB, −5dB, 0dB, 5dB, and 10dB. For the test data, the first 25 sentences from the TIMIT test subset were combined with nine noises, including leopard, white, machinegun, hfchannel, destroyerengine, factory1, factory2, buccaneer1, and buccaneer2, at four SNR levels (−7.5dB, −2.5dB, 2.5dB, and 7.5dB). A total of 250 audio test data samples were obtained. It is important to note that the SNR, noise type, and voice settings of the test set and training set are entirely different.

For further evaluation, another dataset, the Voice Bank Corpus (CSTR VCTK), was utilized. This dataset consists of a training set and a test set, both pre-mixed with the DEMAND noise dataset. The noisy training set comprises 56 speakers, with clean speech mixed with 10 types of noise (including 2 types of artificial noise and 8 types of noise from the Demand database [41]) at 4 SNR conditions (15, 10, 5, and 0 dB). Each person contributed 10 different sentences per condition. The test set consists of 2 speakers, with 5 types of noise and 4 SNR conditions (17.5, 12.5, 7.5, and 2.5 dB). For each condition, 20 different sentences from each person were used.

B. EXPERIMENTAL RESULTS

1) ANALYSIS ON THE SIZE OF THE WAVE-U-NET

Since the mixed TIMIT dataset exhibits a wide range of signal-to-noise ratios (SNR), we chose to utilize it for our analysis. Initially, we examined the impact of model parameters on speech quality. To compress the model size, we reduced the number of kernels in each convolutional layer. Our model architecture consisted of 12-layer downsampling blocks as the encoder and 12-layer upsampling blocks as the decoder. The number of layers remained the same before and after the reduction.

During training, we employed the Adam optimizer with a learning rate of 0.001 and decay rates of $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The batch size was set to 16. In the 100% size model frame we completely follow the settings in the original Wave-U-Net article. The encoder consists of conv1d layers and downsampling blocks, which is responsible for compressing and resampling the feature size. The decoder, on the other hand, includes corresponding upsampling blocks and deconv1d layers. To determine the compression rate of the student model, we gradually reduce the original model size from 100% to 75%, 50%, 25%, and 12%, as shown in Table 2. For comparison, we reduce the hidden size of the encoder and decoder. The layer number settings are presented in Table 3.

TABLE 2. The score of different model sizes for the Wave-U-Net.

model size (param.) / metric	noisy	100% (10.3M)	75% (7.9M)	50% (5.1M)	25% (2.6M)	12% (1.1M)
STOI	0.69	0.77	**0.78**	0.76	0.76	0.73
PESQ	1.65	**2.17**	2.16	2.15	2.13	2.04
SISDR	0.01	8.65	**8.92**	8.42	8.65	7.51

TABLE 3. The layer setting and the number of initial filters for each model size.

		100%	75%	50%	25%
	w DS/US block	The number of initial filters			
DS block 1, 2, … DS block 7	○	24	21	12	8
Conv1d layer 1, 2, ... Conv1d layer 5	×				
deConv1d layer 1, 2, … deConv1d layer 5	×				
US block 1, 2, … US block 7	○				

TABLE 4. The score of different KD learning ratios.

α / metric	1	0.75	0.5	0.25
STOI	**0.76**	**0.76**	0.75	0.75
PESQ	2.15	**2.17**	2.14	2.11
SISDR	8.24	8.38	**8.53**	7.87

The left side of the table is stacked in order according to the layer order of the model, which contains Downsample (DS) and Upsample (US) blocks, convolution and deconvolution layers. The number of filters in different model sizes doubles layer by layer in encoding stage. The kernel size is set to 5 in all layers.

It was observed that when the model size was further reduced from 25% to 12%, the performance significantly declined. This can be attributed to the fact that with a reduction from 100% to 25% of the model size, the parameters were still sufficient to handle the task. However, when the number of model parameters becomes insufficient, the performance experiences a significant degradation. Based on this observation, for subsequent experiments, we utilized 100% of the model parameters for the teacher model and 25% for the student model.

2) FIXED KD LEARNING RATIO ANALYSIS

Next, we conducted experiments to evaluate the effects of different KD learning ratios. The KD learning ratio α remained fixed throughout the training process, and different values of α emphasized different points in the training. Typically, in KD learning, α is set to 0.5, indicating an equal contribution from the soft target and the hard target. In this experiment, both the teacher and student models used the same loss function, and we set α values to 1, 0.75, 0.5, and 0.25, respectively. A value of 1 meant that the soft target (teacher output) was fully utilized, while a decrease in α indicated a shift in the training direction towards the clean target. The results are presented in Table 4.

Based on the experimental results, it can be observed that under the same number of parameters, the gap between the maximum and minimum PESQ scores was 0.06, while SISDR reached 0.7. This demonstrates that the KD learning ratio has a significant impact on model training. Furthermore, the model performed best when α was set to 0.75. Therefore, in subsequent experiments, we used the fixed ratio KD model with α set to 0.75 for comparison.

TABLE 5. Comparison of PESQ score in different SNRs.

SNR / model	-7.5	-2.5	2.5	7.5	avg.
noisy	1.17	1.46	1.81	2.17	1.65
teacher (100%)	1.58	2.01	**2.39**	**2.71**	**2.18**
non-KD (25%)	1.59	1.98	2.33	2.63	2.13
fixed α (25%)	**1.63**	2.01	2.37	2.66	2.17
method A (25%)	**1.63**	**2.02**	2.37	2.62	2.17
method B (25%)	1.62	2.00	2.35	2.65	2.16
method C (25%)	1.62	**2.02**	2.38	2.68	**2.18**

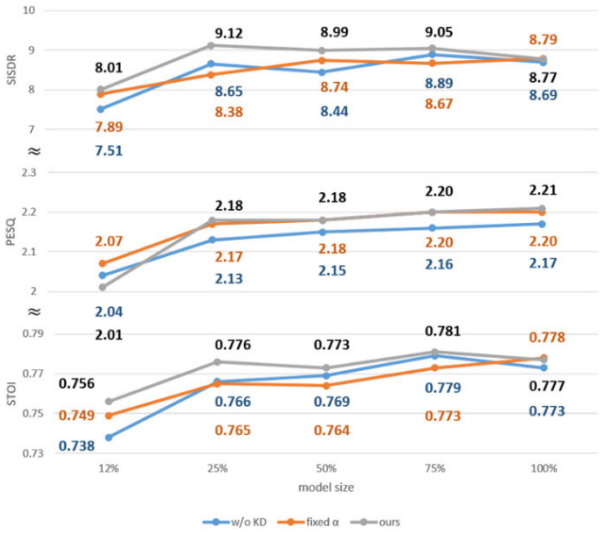


FIGURE 6. The scores for method C with different model sizes.

3) SE WITH RL-BASED KD ANALYSIS

To evaluate the performance of SE, we trained three student models using Methods A, B, and C as proposed in (6) and compared them with the teacher model, non-KD students, and the best model with a fixed KD learning ratio ($\alpha = 0.75$). The results are presented in Table 5. It can be observed that under a 25% reduction in the teacher model size, Method C exhibited the best and most stable performance. We further tested the SE systems trained using non-KD, fixed ratio KD, and Method C, and the results are illustrated in Figure 6.

From the experimental results, it was found that the teacher model had an average Perceptual Evaluation of Speech Quality (PESQ) score of 2.18, and when the size was reduced to

TABLE 6. Comparison of the PESQ scores for the speech data in different noise types.

Method	PESQ								
	leopard	factory1	factory2	machinegun	buccaneer1	buccaneer2	destroyer engine	white	hf channel
Noisy	2.01	1.53	1.77	2.12	1.44	1.52	1.58	1.54	1.38
Teacher (100%)	2.83	2.20	2.51	2.55	2.16	1.89	1.94	1.78	1.71
non-KD (25%)	2.67	2.17	2.45	**2.56**	2.12	1.86	1.92	1.82	1.63
alpha = 0.75	2.71	2.17	2.47	2.40	2.13	1.78	1.96	1.67	1.69
Method A	2.70	2.18	2.48	2.51	2.15	**1.92**	1.98	**1.90**	**1.73**
Method B	**2.77**	**2.20**	**2.51**	2.41	2.17	1.82	**2.00**	1.81	1.69
Method C	2.76	**2.20**	2.49	2.54	**2.18**	**1.92**	1.96	1.89	1.65

TABLE 7. Comparison of the SISDR scores for the speech data in different noise types.

Method	SISDR								
	leopard	factory1	factory2	machinegun	buccaneer1	buccaneer2	destroyer engine	white	hf channel
Noisy	-3e-4	0.02	0.01	0.02	4.1e-4	-3e-4	-3e-3	0.01	3.6e-4
Teacher (100%)	14.77	**9.49**	**12.37**	**14.56**	**8.74**	3.96	6.81	2.84	6.04
non-KD (25%)	14.33	9.30	12.11	14.18	8.55	4.81	6.36	3.94	5.29
alpha = 0.75	13.93	8.84	11.62	12.89	8.18	2.33	6.00	1.98	5.01
Method A	14.31	9.24	12.10	14.24	8.53	4.74	6.78	4.79	6.02
Method B	14.63	9.37	12.23	13.63	8.71	3.85	**6.89**	3.68	**6.08**
Method C	**14.79**	9.33	12.28	14.24	8.71	**5.78**	6.58	**5.54**	5.35

TABLE 8. Comparison of the scores for the speech data in different SNRs.

Method	-7.5 dB		-2.5 dB		2.5 dB		7.5 dB		total	
	PESQ	SISDR	PESQ	SISDR	PESQ	SISDR	PESQ	SISDR	PESQ	SISDR
Noisy	1.17	-7.49	1.46	-2.48	1.81	2.51	2.17	7.50	1.65	0.001
Teacher (100%)	1.58	2.46	2.01	7.25	2.39	11.01	2.71	13.89	2.18	8.65
non-KD (25%)	1.59	2.68	1.98	7.25	2.33	11.01	2.63	14.17	2.13	8.77
alpha = 0.75	**1.63**	2.03	2.01	6.94	2.37	10.76	2.66	13.75	2.17	8.38
Method A	**1.63**	2.98	**2.02**	7.58	2.37	11.19	2.62	14.17	2.17	8.98
Method B	1.62	2.76	2.00	7.32	2.35	8.50	2.65	14.08	2.16	8.79
Method C	1.62	**3.14**	**2.02**	**7.68**	**2.38**	**11.33**	**2.68**	**14.32**	**2.18**	**9.12**

TABLE 9. Comparison of PESQ scores for speech data with different noise types similar to the training set.

Method	PESQ				
	pink	babble	destroyerops	f16	volvo
Noisy	1.52	1.58	1.61	1.59	2.93
Teacher (100%)	**2.32**	**2.17**	**2.40**	**2.29**	**3.39**
non-KD (25%)	2.28	2.15	2.36	2.25	3.37
alpha = 0.75	2.23	2.13	2.34	2.21	3.33
Method C	2.30	2.16	2.38	2.27	3.33

25%, the performance declined by 0.05, which corresponds to a 9.43% decrease in performance. Through knowledge distillation, both the fixed ratio KD and the proposed methods proved beneficial in improving performance under a 25% model size. Method C, in particular, showed no performance degradation. To further analyze the differences between the

TABLE 10. Comparison of PESQ scores for speech data with different noise types similar to the training set.

Method	SISDR				
	pink	babble	destroyerops	f16	volvo
Noisy	-0.01	-0.02	1e-3	-0.01	-0.01
Teacher (100%)	7.41	6.60	8.37	7.06	13.32
non-KD (25%)	7.27	6.33	8.13	6.79	12.44
alpha = 0.75	7.12	6.35	8.07	6.69	12.54
Method C	**7.80**	**6.90**	**8.76**	**7.37**	**13.91**

TABLE 11. Comparison of mixture noise reduction similar to the training set under different SNRs.

Method	-7.5 dB		-2.5 dB		2.5 dB		7.5 dB	
	PESQ	SISDR	PESQ	SISDR	PESQ	SISDR	PESQ	SISDR
Noisy	1.31	-7.54	1.66	-2.51	2.02	2.50	2.39	7.50
Teacher (100%)	1.97	4.46	**2.37**	7.97	2.66	10.16	2.97	11.62
non-KD (25%)	**1.99**	4.23	2.37	7.67	2.70	9.78	**2.99**	11.08
alpha = 0.75	1.91	4.25	2.33	7.60	2.65	9.69	2.91	11.08
Method C	1.96	**4.63**	2.36	**8.25**	2.68	**10.63**	2.95	**13.91**

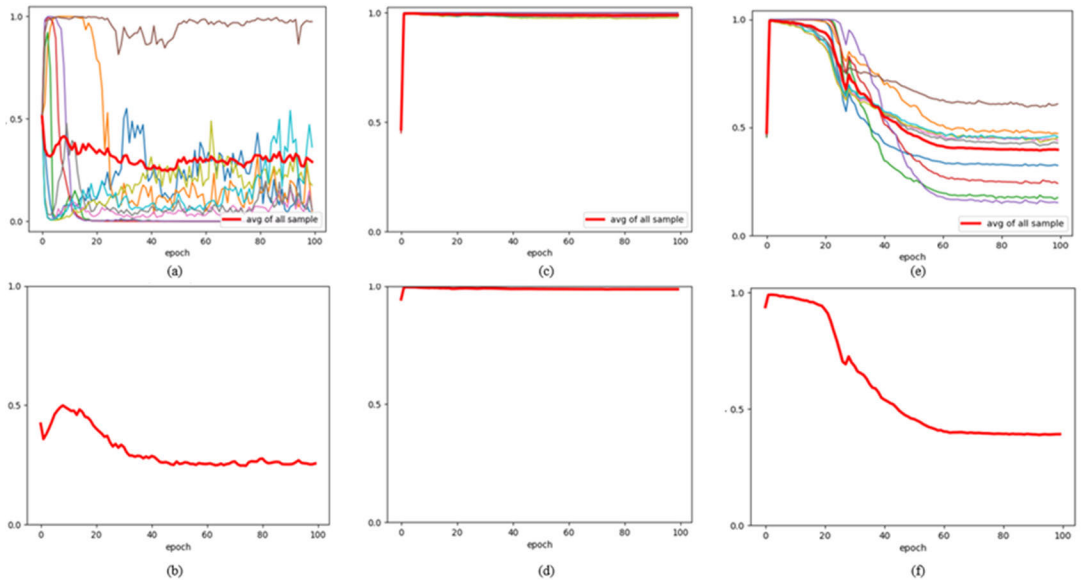


FIGURE 7. α value as a function of epoch for (a) 10 samples with Method A (b) average α value for all samples with Method A (c) 10 samples with Method B (d) average α value for all samples with Method B, and (e) 10 samples with Method C (f) average α value for all samples with Method C.

proposed KD learning method, the fixed ratio KD, and the student model, we provided scores for each SNR and noise type in Table 6, Table 7, and Table 8. It can be observed that the fixed KD learning ratio resulted in a decline in SISDR performance, whereas Methods A, B, and C all exhibited improvements to some extent. Among them, Method C was the most stable and effective.

Method C demonstrated outstanding performance, particularly at low SNR levels. From Table 8, it can be seen that at an SNR of −2.5 dB, Method C achieved improvements of 1.81% and 4.41% in PESQ and SISDR scores, respectively, compared to the teacher model. Similarly, at an SNR of −7.5 dB, Method C showed enhancements of 9.75% and 6.83% in PESQ and SISDR scores, respectively. However, in high SNR conditions, the PESQ score slightly decreased. This can be attributed to the training stability decay method employed in the reward, which was relatively rough and caused the training objective for speech with low-level noise to deviate slightly.

4) ANALYSIS OF DIFFERENT POLICY NETWORKS

In this section, we will provide a more detailed explanation of Methods A, B, and C, as well as the α value estimated by the policy network, to elucidate the performance differences

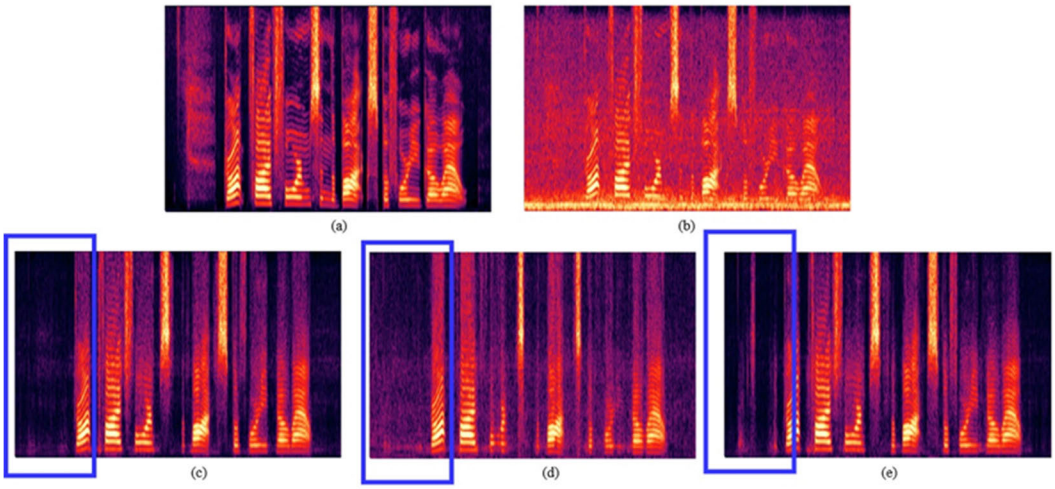


FIGURE 8. Spectrum of: (a) clean (b) speech mixed with leopard noise (c) speech enhanced by teacher model (d) speech enhanced by non-KD method, and (e) speech enhanced by Method C.

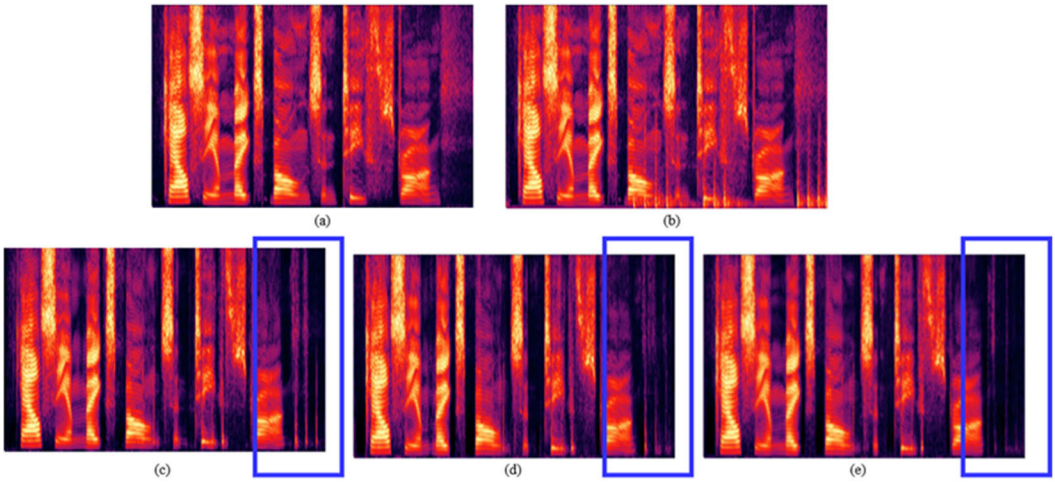


FIGURE 9. Spectrum of: (a) clean (b) speech mixed with leopard noise (c) teacher model (d) non-KD method, and (e) Method C.

TABLE 12. Comparison of score mean and standard deviation for different methods in CSTR VCTK dataset.

method	teacher	RL learning rate	PESQ	SISDR
noisy	-	-	1.967±0.752	8.449±5.614
baseline [42]	-	-	2.563±0.602	18.504±4.06
teacher 100%	-	-	2.601±0.593	18.493±4.084
non-KD 25%	-	-	2.523±0.596	18.116±3.977
fixed KD (α=0.75)	best of all	-	2.589±0.608	18.535±3.949
method C	best of all	1e-5 to 1e-6	2.584±0.616	18.474±4.092
	best of all	1e-6 to 1e-8	**2.640±0.631**	**18.639±3.804**
	best of first 100 epoch	1e-6 to 1e-8	2.638±0.617	18.559±3.806

among the three methods. We examined how the α values of 10 individual samples changed during training and calculated the average α value across all samples. First, the results for Method A are presented in Figure 7 (a) and (b). The red line in (a) signifies the average α value for 10 samples, whereas (b) illustrates the average α value for all samples. In the initial stages of training, the α value quickly converged towards either 0 or 1. As the number of training data increased,

TABLE 13. Comparison of PESQ scores for two tailed t-test results.

method	teacher	RL learning rate	t value	p value
fixed KD (α=0.75)	best of all	-	2.235	0.025
method C	best of all	1e-5 to 1e-6	2.032	0.042
	best of all	1e-6 to 1e-8	**3.866**	**1e-4**
	best of first 100 epoch	1e-6 to 1e-8	3.848	**1e-4**

TABLE 14. Comparison of SISDR scores for two tailed t-test results.

method	teacher	RL learning rate	t value	p value
fixed KD (α=0.75)	best of all	-	2.145	0.032
method C	best of all	1e-5 to 1e-6	1.797	0.072
	best of all	1e-6 to 1e-8	**2.726**	**0.006**
	best of first 100 epoch	1e-6 to 1e-8	2.309	0.021

most α values moved closer to 0, with only a few approaching 1. However, in the later stages of training, α did not converge well and exhibited significant oscillations, with a maximum amplitude of approximately 0.3. Nevertheless, due to algorithmic reasons, the convergence curve fluctuates too much compared with Method C, resulting in unstable speech quality.

Continuing with Method B, it can be observed from the average value in (d) that all α values in the early stages quickly converge to near 1. This suggests that hard targets are more beneficial for model training than soft targets, regardless of subsequent input conditions. Additionally, in (c), it can be noted that the variation in α during the training process for different samples is minimal, resulting in the entire training process not significantly differing from the typical KD training. For Method C, (f) illustrates that α quickly converges to 1 at the beginning of training, once again highlighting the value of soft targets during the early stages of training. As training progressed, distinct convergence curves and points emerged for different samples. As depicted in (e), the convergence process remained stable, and the maximum amplitude during the later stages of training did not exceed 0.05. The final convergence values ranged from approximately 0.3 to 0.6, with an average convergence value of around 0.4 across all samples. The α values obtained from Method C proved to be more reasonable and effective. When compared to Method A, Method C exhibited a more stable training process, resulting in a final α value with only 1/10th the amplitude seen in Method A. Compared to Method B, Method C demonstrated superior sample recognition capability. Furthermore, apart from estimating the α value based on different samples, it can also be adjusted according to different training time points. From (e), it is evident that the α value curve for the 10 samples is quite distinct, demonstrating that the model performs well at every step. Adjusting the learning objectives' proportions also addresses the TIP problem and fully leverages the benefits of knowledge distillation. Consequently, Method C was selected to build the final model.

5) SPECTRUM DIFFERENCE OF KDRL

We examined the speech output represented by the spectrum and compare the difference in output between various methods. The results of two different cases are discussed below.

In the first case, leopard noise was used, as shown in Figure 8. Leopard noise was present throughout the entire speech signal, including the silent parts. From figures (c), (d), and (e), it can be observed that the noise in the silent parts of the speech can be filtered out by the teacher model or the RL-based KD method. However, the baseline model is not able to effectively remove the noise. Moving on to the second case, machinegun noise was used, as shown in Figure 9. This noise is non-stationary in nature. It can be observed that in the second half of the speech signal, neither the teacher model nor the baseline model can effectively eliminate the noise. However, our method demonstrates a significant reduction in noise and even performs better than the teacher model.

These results indicate that our proposed method is effective in reducing different types of noise, including noise present in silent parts and non-stationary noise. It outperforms the baseline model and even achieves better performance than the teacher model in certain scenarios.

6) COMPARISON OF CSTR VCTK DATASET

To facilitate comparison with other methods, we conducted similar experiments on the CSTR VCTK dataset and included the effects of the policy network learning rate and low-performing teacher tests. The results are presented in Table 12. For the baseline we implemented the method according to [42], and achieved results close to the original research in terms of speech clarity. As the evaluation scores were similar when the model size was set to 25%, we performed a two-tailed t-test between the results of fixed KD, Method C, and non-KD to assess their significance, as shown in Table 13 and Table 14.

The results in Table 12 show that Method C consistently performs better than the teacher model at a 25% model size, regardless of whether a less effective teacher is used or not. To confirm that our proposed method is suitable

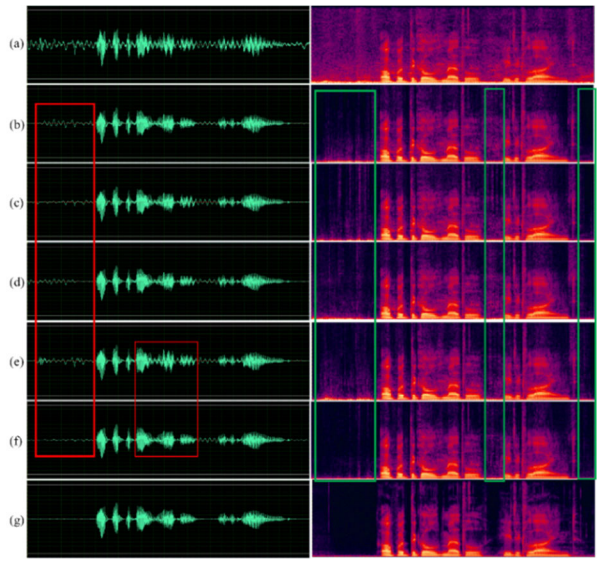


FIGURE 10. Compare silent segments of the following results: (a) noisy (b) teacher 100% (c) teacher 25% (d) fixed ratio KD (e) baseline (f) KDRL and (g) clean, selected from the CSTR VCTK dataset.

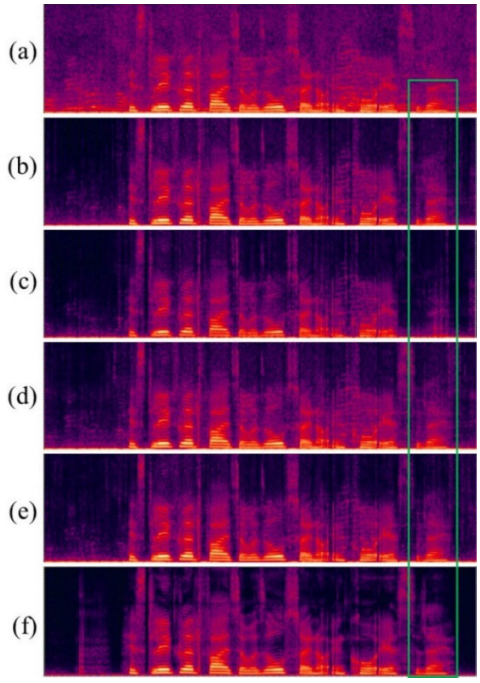


FIGURE 11. Comparing speech end segments from following results: (a) noisy (b) teacher 100% (c) teacher 25% (d) fixed ratio KD (e) KDRL, and (f) clean, selected from the CSTR VCTK dataset.

for various SE models, we selected Wave-U-Net and Conv-TasNet, which utilize different loss functions. We compared our proposed method with recent SE methods using knowledge distillation on the pre-mixed CSTR VCTK dataset. Unlike other methods that require multiple pre-trained teachers, large models for feature extraction, or additional labels and materials, our approach is straightforward, uncomplicated, and efficient. Nevertheless, for direct comparison

TABLE 15. Performance and model size comparison on CSTR VCTK dataset: "-" denotes the results not provided in the original paper; repro. - our reproduction of experiments.

method	param	PESQ	SISDR	training time
noisy	-	1.967	8.449	-
CleanUNet [48]	39.77M	2.88	-	-
DEMUCS(Small) [47]	18.9M	2.93	-	-
baseline (repro.) [42]	4.5M	2.563	18.504	-
Wave-U-Net [6]	10.3M	2.601	18.493	3d 8h
Wave-U-Net + KDRL (ours)	2.6M	2.640	18.639	5d 19h
Conv-TasNet [1]	8.5M	2.482	17.250	3d 3h
Conv-TasNet + KDRL (ours)	2.2M	2.515	18.448	4d 9h

purposes, we have included the data in Table 15. Although there are already excellent compact models available, our method can achieve even further model size compression. The training time listed in the table is an estimate based on the model trained with an Nvidia GeForce GTX 1080 Ti. Because the baseline method consists of multiple modules, with some requiring fine-tuning, we are unable to provide specific training times. When the proposed method KDRL is integrated into Wave-U-Net (Wave-U-Net+KDRL), it can enhance PESQ by 0.039 and SISDR by 0.146 while reducing the model's parameters by 7.7 million. Similarly, when integrated into Conv-TasNet (Conv-TasNet +KDRL), it results in a 0.033 increase in PESQ and a 1.198 improvement in SISDR, all while saving 6.3 million parameters.

Furthermore, we present some differences between our method and others on the CSTR VCTK dataset in Figure 10. In Figure 11 we compare the spectrum of each method of KD, the green rectangle in the spectrogram corresponds to the first red rectangle in the time-domain waveform on the right, highlighting the difference in noise reduction for silent segments. The right side red rectangle indicates that the speech energy in (e) remains relatively intact when the silent segment is cleaner. Additionally, Figure 11 demonstrates that when the teacher model is reduced from 100% to 25%, the speech segment in (c) within the green rectangle has been effectively filtered out, while the speech component can be well preserved by utilizing the knowledge distillation approach. Overall, these results illustrate the superiority of our method compared to others on the CSTR VCTK dataset, both in terms of performance and the ability to preserve speech components in noisy segments.

V. CONCLUSION

This study presents an effective solution to the TIP problem and demonstrates, through experiments, that providing dynamic learning objectives based on different samples can significantly enhance the quality of student learning during their training. By training the policy network to observe student performance and adjust the reward function, we can generate progressive teaching materials and replicate the advantages of ''low-performing teachers'' without the need for extensive teacher selection or storage. Additionally, there is no requirement for additional corpora, annotations, or large pre-trained models. This approach has been validated using

two representative SE models employing different loss functions. As long as the loss function allows for evaluation by the RL model, our KDRL method should be compatible with most SE models.

Furthermore, we conducted experiments on two datasets of varying sizes: TIMIT and CSTR VCTK. The results indicate that the proposed KDRL method effectively eliminates noise while preserving speech components and reducing computational demands. Moreover, sample-level guidance continues to enhance speech quality even after reducing the size of the student model.

REFERENCES

[1] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 8, pp. 1256–1266, Aug. 2019.

[2] T.-A. Hsieh, C. Yu, S.-W. Fu, X. Lu, and Y. Tsao, "Improving perceptual quality by phone-fortified perceptual loss using Wasserstein distance for speech enhancement," 2020, *arXiv:2010.15174*.

[3] A. Sivaraman and M. Kim, "Sparse mixture of local experts for efficient speech enhancement," 2020, *arXiv:2005.08128*.

[4] K. Tan and D. Wang, "Learning complex spectral mapping with gated convolutional recurrent networks for monaural speech enhancement," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, pp. 380–390, 2020.

[5] S. Pascual, A. Bonafonte, and J. Serrà, "SEGAN: Speech enhancement generative adversarial network," 2017, *arXiv:1703.09452*.

[6] D. Stoller, S. Ewert, and S. Dixon, "Wave-U-Net: A multi-scale neural network for end-to-end audio source separation," 2018, *arXiv:1806.03185*.

[7] J. L. Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR—Half-baked or well done?" in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 626–630.

[8] Z. Meng, J. Li, Y. Gong, and B.-H. Juang, "Adversarial feature-mapping for speech enhancement," 2018, *arXiv:1809.02251*.

[9] S.-W. Fu, C.-F. Liao, and Y. Tsao, "Learning with learned loss function: Speech enhancement with quality-net to improve perceptual evaluation of speech quality," *IEEE Signal Process. Lett.*, vol. 27, pp. 26–30, 2020.

[10] S.-W. Fu, C. Yu, T.-A. Hsieh, P. Plantinga, M. Ravanelli, X. Lu, and Y. Tsao, "MetricGAN+: An improved version of MetricGAN for speech enhancement," 2021, *arXiv:2104.03538*.

[11] S. Kataria, J. Villalba, and N. Dehak, "Perceptual loss based speech denoising with an ensemble of audio pattern recognition and self-supervised models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 7118–7122.

[12] X. Tan and X.-L. Zhang, "Speech enhancement aided end-to-end multi-task learning for voice activity detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 6823–6827.

[13] Y. Gong, L. Liu, M. Yang, and L. Bourdev, "Compressing deep convolutional networks using vector quantization," 2014, *arXiv:1412.6115*.

[14] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," 2017, *arXiv:1711.00937*.

[15] E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, and L. Van Gool, "Soft-to-hard vector quantization for end-to-end learning compressible representations," 2017, *arXiv:1704.00648*.

[16] Z. Huang and N. Wang, "Data-driven sparse structure selection for deep neural networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, vol. 2018, pp. 304–320.

[17] P. Molchanov, A. Mallya, S. Tyree, I. Frosio, and J. Kautz, "Importance estimation for neural network pruning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11256–11264.

[18] Z. Liu, M. Sun, T. Zhou, G. Huang, and T. Darrell, "Rethinking the value of network pruning," 2018, *arXiv:1810.05270*.

[19] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[20] L. Yuan, F. E. Tay, G. Li, T. Wang, and J. Feng, "Revisiting knowledge distillation via label smoothing regularization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3902–3910.

[21] X. Hao, S. Wen, X. Su, Y. Liu, G. Gao, and X. Li, "Sub-band knowledge distillation framework for speech enhancement," 2020, *arXiv:2005.14435*.

[22] S. Nakaoka, L. Li, S. Inoue, and S. Makino, "Teacher–student learning for low-latency online speech enhancement using Wave-U-Net," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 661–665.

[23] K. Xu, L. Rui, Y. Li, and L. Gu, "Feature normalized knowledge distillation for image classification," in *Computer Vision—ECCV 2020*. Cham, Switzerland: Springer, 2020, pp. 664–680.

[24] Y. Liu, L. Sheng, J. Shao, J. Yan, S. Xiang, and C. Pan, "Multi-label image classification via knowledge distillation from weakly-supervised detection," in *Proc. 26th ACM Int. Conf. Multimedia*, Oct. 2018, pp. 700–708.

[25] L. Zhang, J. Song, A. Gao, J. Chen, C. Bao, and K. Ma, "Be your own teacher: Improve the performance of convolutional neural networks via self distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3712–3721.

[26] K. Kim, B. Ji, D. Yoon, and S. Hwang, "Self-knowledge distillation with progressive refinement of targets," 2020, *arXiv:2006.12000*.

[27] K. Tan and D. Wang, "Towards model compression for deep learning based speech enhancement," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 29, pp. 1785–1794, 2021.

[28] S. Wang, K. Li, Z. Huang, S. M. Siniscalchi, and C.-H. Lee, "A transfer learning and progressive stacking approach to reducing deep model sizes with an application to speech enhancement," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 5575–5579.

[29] D. Y. Park, M.-H. Cha, D. Kim, and B. Han, "Learning student-friendly teacher networks for knowledge distillation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 13292–13303.

[30] H. Ma, T. Chen, T.-K. Hu, C. You, X. Xie, and Z. Wang, "Undistillable: Making a nasty teacher that CANNOT teach students," 2021, *arXiv:2105.07381*.

[31] X. Hao, C. Xu, L. Xie, and H. Li, "Optimizing the perceptual quality of time-domain speech enhancement with reinforcement learning," *Tsinghua Sci. Technol.*, vol. 27, no. 6, pp. 939–947, Dec. 2022.

[32] Z. Zhao, S. Elshamy, and T. Fingscheidt, "A perceptual weighting filter loss for DNN training in speech enhancement," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust. (WASPAA)*, Oct. 2019, pp. 229–233.

[33] A. S. Subramanian, X. Wang, M. K. Baskar, S. Watanabe, T. Taniguchi, D. Tran, and Y. Fujita, "Speech enhancement using end-to-end speech recognition objectives," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust. (WASPAA)*, Oct. 2019, pp. 234–238.

[34] S. Zhao, T. H. Nguyen, and B. Ma, "Monaural speech enhancement with complex convolutional block attention module and joint time frequency losses," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 6648–6652.

[35] G. Yu, A. Li, C. Zheng, Y. Guo, Y. Wang, and H. Wang, "Dual-branch attention-in-attention transformer for single-channel speech enhancement," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 7847–7851.

[36] R. Cao, S. Abdulatif, and B. Yang, "CMGAN: Conformer-based metric GAN for speech enhancement," 2022, *arXiv:2203.15149*.

[37] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 12449–12460.

[38] S.-W. Fu, Y. Tsao, H.-T. Hwang, and H.-M. Wang, "Quality-Net: An end-to-end non-intrusive speech quality assessment model based on BLSTM," 2018, *arXiv:1808.05344*.

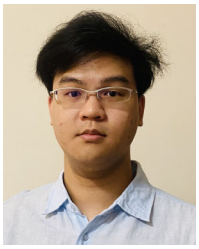
[39] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "DARPA TIMIT acoustic-phonetic continuous speech corpus CD-ROM. NIST speech disc1-1.1," NASA STI/Recon, Tech. Rep. NISTIR4930, 1993.

[40] A. Varga and H. J. M. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Commun.*, vol. 12, no. 3, pp. 247–251, Jul. 1993.

[41] J. Thiemann, N. Ito, and E. Vincent, "The diverse environments multi-channel acoustic noise database (DEMAND): A database of multichannel environmental noise recordings," in *Proc. Meetings Acoust.*, 2013, Art. no. 035081.

[42] M. Thakker, S. E. Eskimez, T. Yoshioka, and H. Wang, "Fast real-time personalized speech enhancement: End-to-end enhancement network (E^3Net) and knowledge distillation," 2022, *arXiv:2204.00771*.

[43] N. Saleem, T. S. Gunawan, M. Kartiwi, B. S. Nugroho, and I. Wijayanto, "NSE-CATNet: Deep neural speech enhancement using convolutional attention transformer network," *IEEE Access*, vol. 11, pp. 66979–66994, 2023.
[44] M. Chen, Q. Zhang, Q. Song, X. Qian, R. Guo, M. Wang, and D. Chen, "Neural-free attention for monaural speech enhancement towards voice user interface for consumer electronics," *IEEE Trans. Consum. Electron.*, early access, Mar. 14, 2023, doi: 10.1109/TCE.2023.3254507.
[45] R. Soleymanpour, M. Soleymanpour, A. J. Brammer, M. T. Johnson, and I. Kim, "Speech enhancement algorithm based on a convolutional neural network reconstruction of the temporal envelope of speech in noisy environments," *IEEE Access*, vol. 11, pp. 5328–5336, 2023.
[46] F. K. Peracha, M. I. Khattak, N. Salem, and N. Saleem, "Causal speech enhancement using dynamical-weighted loss and attention encoder–decoder recurrent neural network," *PLoS ONE*, vol. 18, no. 5, May 2023, Art. no. e0285629.
[47] A. Defossez, G. Synnaeve, and Y. Adi, "Real time speech enhancement in the waveform domain," 2020, *arXiv:2006.12847*.
[48] Z. Kong, W. Ping, A. Dantrey, and B. Catanzaro, "Speech denoising in the waveform domain with self-attention," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 7867–7871.



SHIH-CHUAN CHU received the B.S. and M.S. degrees in electrical engineering from the Southern Taiwan University of Science and Technology, Tainan, Taiwan, in 2017 and 2019, respectively. He is currently pursuing the Ph.D. degree with National Cheng Kung University (NCKU). His research interests include speech signal processing, speech enhancement, and separation.

CHUNG-HSIEN WU (Senior Member, IEEE) received the B.S. degree in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1981, and the M.S. and Ph.D. degrees in electrical engineering from National Cheng Kung University (NCKU), Tainan, Taiwan, in 1987 and 1991, respectively. Since 1991, he has been with the Department of Computer Science and Information Engineering, NCKU. He was a Visiting Scientist with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in Summer 2003. He was the Deputy Dean of the College of Electrical Engineering and Computer Science, NCKU, from 2009 to 2015. He was a Chair Professor, in 2017. His research interests include deep learning, affective computing, speech recognition/synthesis, and spoken language processing. He was the APSIPA BoG Member, from 2019 to 2021. He received the 2018 APSIPA Sadaoki Furui Prize Paper Award, in 2018, and the Outstanding Research Award from the Ministry of Science and Technology, Taiwan, in 2010 and 2016. He was an Associate Editor of IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, from 2010 to 2014; IEEE TRANSACTIONS ON AFFECTIVE COMPUTING, from 2010 to 2014; *ACM Transactions on Asian and Low-Resource Language Information Processing*; and *APSIPA Transactions on Signal and Information Processing*, from 2014 to 2020.

TSAI-WEI SU received the B.S. degree in computer science and information engineering from National Sun Yat-sen University (NSYSU), Kaohsiung, Taiwan, in 2019, and the M.S. degree in computer science and information engineering from National Cheng Kung University (NCKU), Tainan, Taiwan, in 2021.