IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 8 August 2024, accepted 8 September 2024, date of publication 11 September 2024,
date of current version 23 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3457859

RESEARCH ARTICLE

Cons-KD: Dropout-Robust Knowledge Distillation for CTC-Based Automatic Speech Recognition

JI WON YOON[1], HYEONSEUNG LEE[2], JU YEON KANG[3,4], (Student Member, IEEE), AND NAM SOO KIM[3,4], (Senior Member, IEEE)

[1]Department of Artificial Intelligence, Chung-Ang University, Seoul 06974, South Korea
[2]XL8 Inc., Seoul 06764, South Korea
[3]Department of Electrical and Computer Engineering, Seoul National University, Seoul 08826, South Korea
[4]INMC, Seoul National University, Seoul 08826, South Korea

Corresponding author: Nam Soo Kim (nkim@snu.ac.kr)

This research was supported by the Chung-Ang University Research Grants in 2024.

This work was supported by Samsung Electronics Co., Ltd. (IO201211-08075-01).

ABSTRACT In recent years, there has been a growing interest in applying knowledge distillation (KD) techniques to the connectionist temporal classification (CTC) framework for training more efficient speech recognition models. Although conventional KD approaches have successfully reduced computational burden, they have limitations in dealing with the inconsistency problem caused by dropout regularization, particularly the gap between the training and inference stages. In the context of KD, this inconsistency may hinder the performance improvement of the student model. To overcome this issue, we propose a novel approach, namely Cons-KD, that combines KD and consistency regularization, where the former trains the student model to benefit from the knowledge of the teacher model, and the latter trains the student model to be more robust to the dropout-induced inconsistency. By directly mitigating the inconsistency problem, our KD framework can further improve the student's performance compared to the vanilla KD. Experimental results on the LibriSpeech dataset demonstrate that Cons-KD significantly outperforms previous KD methods, improving the word error rate (WER) from 5.10 % to 4.13 % on the test-clean subset and from 12.87 % to 10.32 % on the test-other subset, respectively. These improvements correspond to relative error rate reduction (RERR) of 19.02 % and 19.81 %, respectively, implying notable advancements beyond conventional KD methods. Additionally, we conduct an in-depth analysis to verify the effect of each proposed objective.

INDEX TERMS Speech recognition, knowledge distillation, teacher-student learning, consistency regularization, consistency training, connectionist temporal classification.

I. INTRODUCTION

Recently, there have been significant advancements in the field of end-to-end speech recognition, aiming to directly map a speech signal to the corresponding text. In comparison to the traditional deep neural network (DNN)-hidden Markov model (HMM) hybrid systems, end-to-end speech recognition models not only simplify the overall training pipeline but also achieve superior performance on the learning task.

With the growing interest in model efficiency, the connectionist temporal classification (CTC) [1] model has gained attention among the end-to-end models due to its non-autoregressive (NAR) nature. In contrast to autoregressive (AR) models [2], [3], [4] that require M decoding steps to generate M tokens during inference, the CTC model can produce the output sequence in parallel, regardless of the length of the target tokens. This NAR property offers a significant advantage in real-world applications where fast inference is desirable.

However, existing CTC models [5], [6], [7], [8] often require substantial computational costs, such as large model size, lengthy training time, and extensive GPU resources, to achieve promising performance. To alleviate this burden, several approaches have been proposed to apply knowledge

The associate editor coordinating the review of this manuscript and approving it for publication was Hasan S. Mir.

distillation (KD) [9] to the CTC framework [10], [11], [12], [13], [14], [15], [16]. KD is a widely-used technique that aims to transfer knowledge from a deep and complex teacher model to a student model with a reduced structure. In the context of KD for CTC, the student model is typically trained to simultaneously predict both the ground-truth labels and the soft labels generated by the teacher model. These soft labels can include sentence predictions or frame-wise softmax outputs. By leveraging the teacher's knowledge, the distilled student can perform better than its baseline trained solely based on the ground-truth labels.

Despite their effectiveness, conventional KD approaches have certain limitations in dealing with the inconsistency problem caused by dropout regularization [17]. Recent research on consistency regularization has shed light on the side effects of dropout, particularly the inconsistency between the training and inference stages [18], [19], [20]. During training, dropout randomly drops out a fraction of neurons, creating a *sub-model* at each iteration. In contrast, during inference, a *full model* is employed without dropout randomness. This significant gap between the sub-model (during training) and the full model (during inference) often leads to performance degradation because the full model may not perform optimally; training based on the dropout technique considers only the configuration of the sub-model. In the context of KD, existing methods have primarily focused on how to effectively transfer knowledge from the teacher model to the student model but have not directly addressed this inconsistency issue, which could impede the performance of the student model. To minimize the performance degradation caused by dropout, it is crucial to design a new KD framework that can mitigate dropout-induced inconsistency, enabling more effective training of the student model.

In this paper, we propose a novel dropout-robust KD framework, referred to Cons-KD. Cons-KD combines KD and consistency regularization, where the former guides the student to mimic the behavior of the teacher, while the latter encourages the student to produce consistent outputs. The core idea is to use dropout not merely as a regularization technique but as a mechanism to generate diverse sub-models within the student model. Through multiple forward passes with varied dropout masks, it is possible to generate a range of student sub-model outputs from a single input, without adding extra model parameters. In contrast to the previous KD frameworks, Cons-KD presents a novel aspect of randomly sampling different student sub-models and encouraging them to produce consistent outputs. Thus, Cons-KD guarantees that the outputs of the student sub-models are consistent and reliable, despite the internal variations introduced by dropout. It not only enhances the effectiveness of KD but also increases the robustness of the distilled model, thereby improving the overall quality of the student model.

When training the student sub-models, the proposed framework involves three training objectives: (1) the original CTC objective function with the ground-truth labels as the target, (2) the KD loss to minimize the distance between the sub-models' average prediction and the teacher's prediction, and (3) the consistency regularization objective to reduce the variance among the predictions of the sub-models. By integrating these three objectives, Cons-KD aims to train a student model that is more robust to dropout, leading to further performance improvements compared to the original KD.

From experimental results on LibriSpeech, it is verified that the proposed method achieves better performance than other previous KD methods. Specifically, on test-clean and test-other datasets, Cons-KD improves the student's word-error rate (WER) from 5.10 % to 4.13 % and from 12.87 % to 10.32 % with greedy decoding, respectively. These improvements correspond to relative error rate reduction (RERR) of 19.02 % and 19.81 %, respectively, implying notable advancements beyond conventional KD methods. Additionally, we conduct further analyses to verify the effect of each objective function on the student model's performance. Our findings highlight the effectiveness of combining KD with consistency regularization, setting a new benchmark for KD applications in speech recognition and offering a robust solution to model inconsistency challenges.

To summarize, the main contributions of this paper are as follows:

- We introduce a novel KD method, namely Cons-KD, designed to improve the student model's robustness against inconsistencies induced by dropout. The proposed framework effectively minimizes the inconsistency issue between the training and inference stages when distilling the knowledge, thereby improving the student's performance. As far as we know, this is the first attempt to explore the combination of consistency regularization and KD specifically for a speech recognition task.
- The proposed approach employs multiple forward passes, each utilizing unique dropout masks, to generate diverse student sub-model outputs from a single input. This sampling method does not require any additional model parameters, thus maintaining the lightweight nature of the student model.
- A novel tripartite training objective framework is presented within Cons-KD, comprising the original CTC loss, the KD objective based on the averaged outputs of student sub-models, and the consistency regularization objective. This strategy ensures effective knowledge transfer and reduces prediction variance, thereby enhancing model stability and performance.
- Experimental results on the LibriSpeech dataset demonstrate that our proposed method notably outperforms conventional KD methods that do not address the inconsistency problem of the student. By minimizing the performance degradation caused by the inconsistency,

Cons-KD significantly improves the student's performance.

The rest of the paper is organized as follows: Section II describes the previous work related to our research. In Section III, we define the inconsistency problem induced by the dropout and introduce our proposed KD approach, namely Cons-KD. Section IV presents the experimental setup and results. Section V presents the paper's conclusions.

II. RELATED WORK

A. CONNECTIONIST TEMPORAL CLASSIFICATION

An end-to-end speech recognition framework aims to convert a sequence of input acoustic features, represented by $x_{1:T} = \{x_1, \ldots, x_T\}$, into a sequence of textual labels $y_{1:N} = \{y_1, \ldots, y_N\}$, where each y_n is a part of a predefined set of labels $\mathcal{Y}$. Here, T and N respectively denote the total number of acoustic frames and the sequence length of the target labels. Given that the source sequence $x_{1:T} = \{x_1, \ldots, x_T\}$ and the target sequence $y_{1:N} = \{y_1, \ldots, y_N\}$ typically have unequal lengths, addressing the challenge of mapping these sequences of varying lengths is essential. To tackle this challenge, the Connectionist Temporal Classification (CTC) [1] employs a "blank" label and allows label repetition across frames. This approach generates a CTC alignment $\pi_{1:T} = \{\pi_1, \ldots, \pi_T\}$, mapping each frame x_t to a label π_t from the expanded set $\mathcal{Y}' = \mathcal{Y} \cup \{\text{blank}\}$. A mapping function $\mathcal{B}$ processes the sequence π into the output sequence y by merging consecutive identical labels and removing blanks. For instance, CTC alignment for the word"speech" could include sequences like $\{\varepsilon, \varepsilon, s, s, \varepsilon, p, e, \varepsilon, e, \varepsilon, c, h, \varepsilon\}$ and $\{s, \varepsilon, p, \varepsilon, e, e, \varepsilon, c, \varepsilon, h\}$ (using 'ε' to denote blank label). After being processed by the mapping function $\mathcal{B}$, both sequences are transformed into the same sequence $\{s, p, e, e, c, h\}$. Explicit alignment between the source sequence x and the target sequence y is not required in the training phase for CTC. The conditional probability of the target sequence y given the source sequence x can be computed as

$$p(y|x) = \sum_{\pi \in \mathcal{B}^{-1}(y)} p(\pi|x), \tag{1}$$

where $\mathcal{B}^{-1}$ represents the inverse of the mapping function, returning all possible alignment sequences that map to y. Under the conditional independence assumption, the probability of the CTC alignment $\pi = \{\pi_1, \ldots, \pi_T\}$ is given by

$$p(\pi|x) = \prod_{t=1}^{T} p(\pi_t|x). \tag{2}$$

Given the target y and the input x, the CTC loss function $\mathcal{L}_{ctc}$ is formulated as

$$\mathcal{L}_{ctc} = -\sum_{(x,y)\in Z} \ln p(y|x), \tag{3}$$

where Z is a set of training data pairs.

B. KNOWLEDGE DISTILLATION

KD is one of the most effective methods for model compression. Hinton et al. [9] initially introduced the concept of KD, which is aimed at transferring knowledge from a deep and powerful teacher model to a shallow and efficient student model. By minimizing the Kullback-Leibler (KL) divergence between the softmax outputs of the two models, the distilled student achieves better performance than its baseline trained solely on the target ground truth. As lightweight models have received significant attention, there have been extensive efforts to integrate KD into the speech recognition framework. In the context of the DNN-HMM hybrid framework, earlier KD research has mainly involved training the student model by reducing the frame-level cross-entropy (CE) loss between the posterior probabilities of the teacher and student models [21], [22], [23], [24], [25], [26]. However, it has been shown that using the frame-level CE objective for training the CTC-based speech recognition model is not suitable, given the alignment-free characteristic of the CTC model and its tendency to produce peaky softmax values. According to previous research [14], [15], [16], this approach can result in performance degradation compared to the case when the model is trained solely with the target label. To deal with this problem, Takashima et al. explored the application of sequence-level KD to the CTC model, inspired by the method proposed by Kim and Rush [27]. Initially, they leveraged N-best hypotheses generated by the teacher model instead of using frame-level softmax outputs [14]. Following this, they introduced a lattice-based sequence-level KD method to enhance computational efficiency [15]. Kurata and Audhkhasi [13] suggested the KD framework designed to train a low-latency student model using knowledge from a high-latency teacher model. Additionally, they introduced Guided CTC training [12], a method for distilling CTC spike timings from the teacher model. Yoon et al. proposed softmax-level KD (SKD) [10], a framework that employs an l_2 loss to measure differences between the softmax outputs of two models, providing an effective alternative to the Kullback-Leibler (KL) divergence objective. Additionally, they pioneered the application of the Fitnets concept, originally proposed by Romero et al. [28], to speech recognition systems. This approach involves transferring the hidden representations from the teacher model to improve the training of speech recognition models. Recently, Yoon et al. introduced Inter-KD [11] that additionally transfers the teacher's knowledge to the intermediate CTC layers [29] of the student network.

III. PROPOSED METHOD

A. PROBLEM FORMULATION

An end-to-end speech recognition model is trained to directly map a speech input x to a text sequence y. When training the model, dropout is commonly employed as a regularization technique to prevent overfitting and improve generalization. With the dropout technique, the model's parameters, denoted

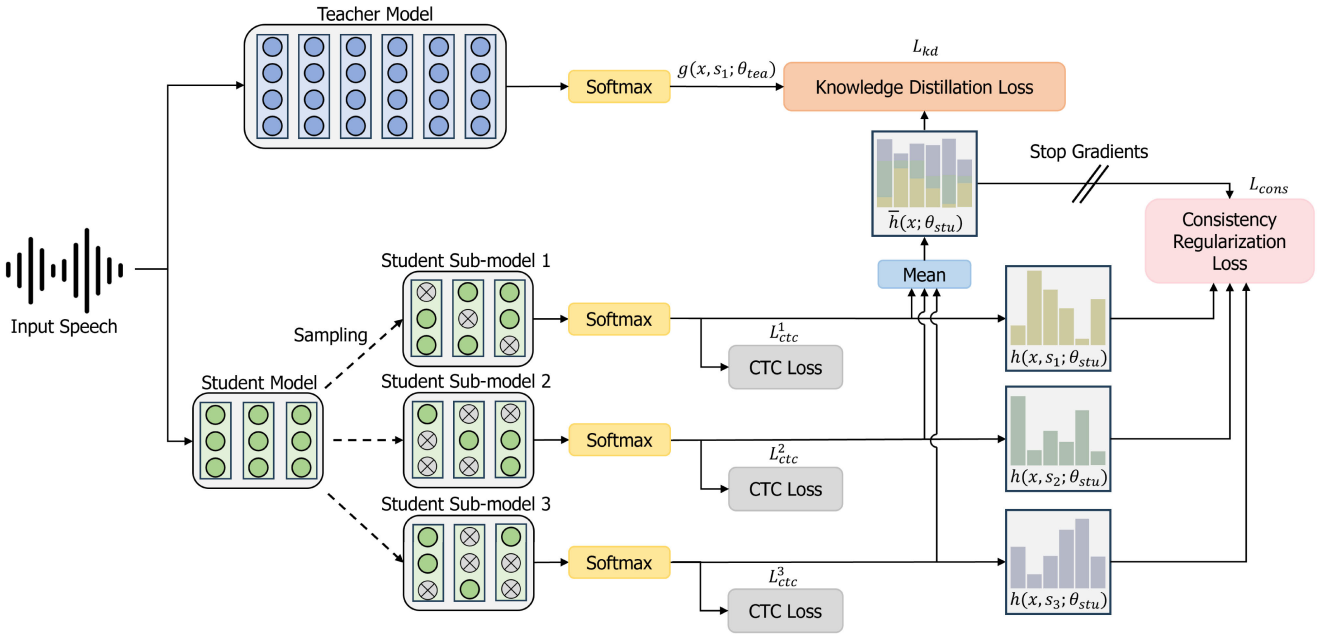


FIGURE 1. An overview of Cons-KD when there are $K = 3$ student sub-models. In the student-sub-model, a crossed-out circle represents dropped subsets of neurons due to the dropout. The proposed framework randomly samples multiple student sub-models and incorporates three training objectives: (1) original CTC training to train the sub-models with the ground-truth labels, (2) knowledge distillation loss to transfer the knowledge from the teacher to the student sub-models, and (3) consistency regularization objective to minimize the variance among the predictions of the sub-models. During the consistency regularization process, the gradients of the average prediction are stopped. This ensures that the consistency objective solely promotes consistency among the sub-models without directly influencing the training of the student model. It is important to note that the proposed framework does not require additional parameters for student sub-model sampling; all student sub-models share the same set of parameters.

as ϕ, are trained to maximize the following log-likelihood:

$$\phi^* = \arg\min_{\phi} \mathrm{E}_s[-\sum_{i=1}^{N} \log p(y_i|x_i, s_i; \phi)] \tag{4}$$

where N represents the total number of training samples, and s_i is the dropout mask for the i-th training sample. Since the dropout involves randomly sampling dropout masks s_i for each training example, it allows the model to sample different sub-models for every training iteration. This stochastic mask sampling procedure introduces diversity and prevents the model from heavily relying on specific units or patterns, resulting in enhanced generalizability.

However, despite its effectiveness in model training, the dropout technique can lead to inconsistencies between the training and inference stages. During training, a *sub-model* is randomly sampled with dropout, while we use a *full model* without dropout during inference. Consequently, the model's behavior may differ between the two stages. Although conventional KD methods show promising performance improvements, they have certain limitations in effectively addressing this inconsistency issue.

B. CONS-KD

Inspired by the above gap between training and inference, we propose a novel distillation method called Cons-KD, which aims to mitigate the inconsistency problem of the student model during knowledge transfer.

As depicted in Figure 1, we employ K multiple forward passes of the student model to generate a set of student sub-models. Each forward pass is constructed by applying a distinct dropout mask, resulting in sub-models inherently different from each other. This sampling technique enables us to obtain diverse output variants from a single input without requiring any additional model parameters. It is important to note that all student sub-models share the same set of parameters. To simplify notation, we denote the softmax output of the k^{th} student sub-model with the dropout mask s_k as $h(x, s_k; \theta_{stu}) \in R^{T \times C}$, where T represents the total number of frames, and C denotes the number of target labels.

The proposed KD framework uses three objectives to train the student sub-models.

1) ORIGINAL CTC OBJECTIVE FUNCTION

Firstly, we train the student sub-models using the original CTC loss. The CTC loss for the k^{th} sub-model is as follows:

$$\mathcal{L}^k_{ctc} = CTC(y, h(x, s_k; \theta_{stu}))/K \tag{5}$$

where y represents the ground-truth target sequence.

2) KD WITH AVERAGE PREDICTION OF STUDENT SUB-MODELS

Our KD objective aims to minimize the difference between the average prediction of the student sub-models and the prediction of the teacher model. Unlike conventional KD methods that solely rely on a single sub-model's output,

we leverage the average prediction derived from multiple sub-models. By employing the average prediction during knowledge transfer, the proposed KD objective not only enables the transfer of the teacher's knowledge to the student model but also improves the overall consistency of the student model's predictions. In the proposed framework, the KD loss can be computed as

$$\mathcal{L}_{kd} = \lambda_{kd} \cdot \|g(x; \theta_{tea}) - \bar{h}(x; \theta_{stu})\|_2^2 \tag{6}$$

where $g(x; \theta_{tea})$ denotes the softmax output of the teacher model, and $\bar{h}(x; \theta_{stu}) = \frac{1}{K}\sum_{k=1}^{K} h(x, s_k; \theta_{stu})$ represents the mean of student sub-models' outputs. We employ the l_2 loss for knowledge transfer instead of the Kullback-Leibler (KL) divergence-based distillation loss, as the latter often fails to converge for the CTC model [10], [14], [15]. In our experiments, we experimentally set the tunable parameter λ_{kd} to 0.25.

3) CONSISTENCY REGULARIZATION FOR STUDENT SUB-MODEL

Additionally, we minimize the l_2 loss between the average prediction $\bar{h}(x; \theta_{stu})$ and each student sub-model's output $h(x, s_k; \theta_{stu})$, similar to computing the sample variance. This objective further focuses on reducing the variance among the predictions. The proposed consistency objective for each sub-model is as follows:

$$\mathcal{L}_{cons}^{k} = \lambda_{cons} \cdot \|h(x, s_k; \theta_{stu}) - \text{sg}[\bar{h}(x; \theta_{stu})]\|_2^2 \tag{7}$$

where sg[·] represents the stop-gradient operation. The stop-gradient operation is used to prevent the gradients from propagating through the average prediction of the student sub-models $\bar{h}(x; \theta_{stu})$, treating it as a constant during the backpropagation process. This ensures that the consistency objective only encourages the student sub-models to generate consistent predictions among themselves without directly influencing the training of the student model. In our experimental setting, we set the parameter λ_{cons} to a value of 0.25.

While the distillation term $\mathcal{L}_{kd}$ also contributes to the consistency of the student model, the consistency regularization term $\mathcal{L}_{cons}^{k}$ specifically targets the reduction of variance among the predictions. We empirically find that combining both $\mathcal{L}_{kd}$ and $\mathcal{L}_{cons}^{k}$ yields better performance improvements compared to the case when using only the distillation term $\mathcal{L}_{kd}$, which will be additionally described in Session IV-D.

4) TOTAL TRAINING OBJECTIVE

When there are K sub-models, the final objective function for Cons-KD is formulated as follows:

$$\mathcal{L}_{total} = \sum_{k=1}^{K}(\mathcal{L}_{ctc}^{k} + \mathcal{L}_{cons}^{k}) + \mathcal{L}_{kd}. \tag{8}$$

Algorithm 1 summarizes the proposed KD strategy.

Algorithm 1 Cons-KD Training Procedure

Input: Dataset (X, Y)
Parameters: teacher model parameters θ_{tea}, student model parameters θ_{stu}, number of student sub-models K, set of dropout masks $\{s_k\}$, learning rate η, weight for KD loss λ_{kd}, weight for consistency loss λ_{cons}
Output: Model parameters θ_{stu}

for (x, y) in (X, Y) **do**
  Initialize $\bar{h} \leftarrow 0$, $\mathcal{L}_{\text{total}} \leftarrow 0$
  Calculate average prediction of student sub-model predictions $h_k \leftarrow h(x, s_k; \theta_{\text{stu}})$
  $\bar{h} \leftarrow \frac{1}{K}\sum_{k=1}^{K} h_k$
  for $k = 1 to K$ **do**
    Compute CTC loss $\mathcal{L}_{\text{ctc}}^{k} \leftarrow \text{CTC}(y, h_k)/K$
    Compute consistency loss $\mathcal{L}_{\text{cons}}^{k} \leftarrow \lambda_{\text{cons}} \cdot \|h_k - \bar{h}\|_2^2$
    Accumulate total loss $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{total}} + \mathcal{L}_{\text{ctc}}^{k} + \mathcal{L}_{\text{cons}}^{k}$
  end for
  Compute KD loss $\mathcal{L}_{\text{kd}}$ for $\bar{h}$ and teacher prediction $\mathcal{L}_{\text{kd}} \leftarrow \lambda_{\text{kd}} \cdot \|g(x; \theta_{\text{tea}}) - \bar{h}\|_2^2$
  Update total loss $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{total}} + \mathcal{L}_{\text{kd}}$
  Update model parameters $\theta_{\text{stu}} \leftarrow \theta_{\text{stu}} - \eta \cdot \nabla_{\theta_{\text{stu}}} \mathcal{L}_{\text{total}}$
end for
return θ_{stu}

IV. EXPERIMENTS

A. EXPERIMENTAL SETTING

1) DATASET

We mainly conducted our experiments on the LibriSpeech [30] dataset, which is widely used as a benchmark for the speech recognition task. This dataset consists of around 1000 hours of English speech readings, recorded at a sampling rate of 16 kHz. During training, we utilized the following subsets: "train-clean-100", "train-clean-360", and "train-other-500". For evaluation, we applied the subsets "dev-clean", "dev-other", "test-clean", and "test-other."

2) PERFORMANCE METRICS

To compare the performance, we measured word error rate (WER) and relative error rate reduction (RERR). WER is a widely used metric to evaluate speech recognition performance. It is calculated by identifying the number of errors in the form of substitutions, insertions, and deletions within the recognition result. These errors are then summed up and divided by the total number of words in the reference sentence. RERR measures the proportional reduction in WER relative to the baseline performance, indicating improvements in speech recognition accuracy.

3) MODEL CONFIGURATION

a: LIBRISPEECH

Both the teacher and student models were built based on the Conformer-CTC architecture, which is the CTC-based variant of the Conformer [8] model. Specifically, Conformer-CTC employs the same encoder architecture as the original

Conformer but utilizes the CTC decoding instead of the Transducer approach, making it a NAR model. Additionally, it replaces the long short-term memory (LSTM) decoder with a linear decoder on the top of the encoder. The Conformer model integrates self-attention and convolution modules, using self-attention layers to capture global interactions and convolutions to effectively identify local correlations. In our experiments, the student model consisted of 16 Conformer blocks, each having 144 dimensions, and utilized a multi-head attention mechanism with 4 heads. This configuration resulted in approximately 9 M parameters. The teacher model comprised 18 Conformer blocks, with each block having 512 dimensions, and employed a multi-head attention mechanism with 8 heads, leading to a total of about 122 M parameters. For the language model (LM), we employed the Transformer [3]-based LM as a neural rescoring tool, designed to rescore the top candidates predicted by the speech recognition model. This LM was trained using the LibriSpeech text corpus. In particular, the top candidates were produced via beam search decoding and subsequently presented to the Transformer-based LM for ranking.

b: COMMON VOICE

We also conducted our experiments on the Common Voice 7.0 [31] Spanish dataset, evaluating the model on both the dev and test sets. For both the teacher and student models, we adopted the hybrid Transducer-CTC architecture [32], a recent ASR model that utilizes a shared encoder with both CTC and Transducer decoders to enhance accuracy and reduce computation. The student baseline's encoder consisted of 15 FastConformer [33] layers with 128 dimensions and 4 heads. The prediction and joint networks had 320 dimensions, and the student model had approximately 8 M parameters. The pre-trained teacher model's encoder comprised 17 FastConformer layers with 512 dimensions and 8 heads. Its prediction and joint networks had 640 dimensions, resulting in approximately 114 M parameters.

4) IMPLEMENTATION DETAILS

In our experiments, we used the NeMo [34] toolkit for implementing the ASR models. For data augmentation during training, we employed SpecAugment [35]. During the training phase of the student model, we configured the augmentation parameters by setting the number of frequency masks to 2 and the number of time masks to 5. Additionally, the frequency and time masks were configured with widths of 27 and 0.05, respectively.

a: LIBRISPEECH

The student model's training was performed on four Quadro RTX 8000 GPUs, each with 48 GB of memory, with a total batch size of 192. We trained the student with 200 epochs, and the AdamW [36] algorithm was adopted as the optimizer, with an initial learning rate of 5.0. The NoamAnnealing scheduler [3] was utilized to regulate the learning rate during the training period, with warmup steps set to 10,000. Furthermore, a minimum learning rate of 1e-6 was specified. As for the teacher model, we utilized a pre-trained checkpoint provided by the NeMo toolkit. Similarly, the checkpoint of the Transformer-based LM was obtained from NeMo. For both teacher and student models, a byte-pair encoding (BPE) [37] vocab size of 128 was used. When applying beam-search decoding with the Transformer-LM, the beam width was experimentally set to 256.

b: COMMON VOICE

The student model was trained using four Quadro RTX 8000 GPUs with a total batch size of 256. Training was conducted over 50 epochs, utilizing the AdamW algorithm as the optimizer with an initial learning rate of 1.0. The NoamAnnealing scheduler settings matched those used in the LibriSpeech experiments. For both the teacher and student models, a BPE vocabulary size of 1024 was employed.

5) CONVENTIONAL KD APPROACHES FOR COMPARISON

We compared Cons-KD with conventional KD techniques, including Guided-CTC [12], SKD [10], and Inter-KD [11]. All KD methods share the same training settings (e.g., learning rate, GPU usage, BPE configuration, etc.) as those of the student baseline. In the Guided CTC framework, the approach focuses on distilling the spike timings from the teacher model. It is achieved by generating a mask that emphasizes the output label with the highest probability at each frame. By applying this mask during the student training, the student model is guided to follow the teacher's spike timing for each prediction. The SKD is a promising KD method in the field of speech recognition, aimed at transferring the frame-level posterior probabilities from the teacher model to the student model. It employs the l_2 loss to measure differences between the softmax outputs of the teacher and student models, providing an effective alternative to the Kullback-Leibler (KL) divergence objective. Inter-KD additionally transfers the teacher's knowledge not only to the final output layer but also to the intermediate CTC layers [29] of the student network. In the original configuration of Inter-KD, the intermediate CTC layers were attached to the 18^{th}, 24^{th}, and 30^{th} layers of the Jasper Mini model, which consists of a total of 33 1D convolutional layers. These intermediate CTC layers were located at higher layers of the model. However, we found that placing the CTC layers in the middle position was more effective for the Conformer model. Therefore, we applied different configurations for performance comparison. Specifically, we used $L = 6$, $L = 8$, $L = 10$, $L = 12$, and $L = 8, 10$, where $L = 6$ indicates that the intermediate CTC layer was added to the 6^{th} layer of the Conformer-CTC model.

TABLE 1. Comparison of WER (%) and RERR (%) on LibriSpeech with greedy decoding. Bold represents superior results.

Model	# of Params.	KD Method		WER (%)				RERR (%)			
				dev		test		dev		test	
				clean	other	clean	other	clean	other	clean	other
Teacher	122 M	None		2.48	6.03	2.78	6.18	-	-	-	-
Student	9 M	None		4.85	12.71	5.10	12.87	-	-	-	-
		Guided CTC [12]		4.62	12.39	4.87	12.10	4.74	2.52	4.51	5.98
		SKD [10]		4.34	11.22	4.48	11.38	10.52	11.72	12.16	11.58
		Inter-KD [11]	L=6	4.25	11.49	4.34	11.40	12.37	9.60	14.90	11.42
			L=8	4.02	11.20	4.35	11.07	17.11	11.88	14.71	13.99
			L=10	4.23	11.08	4.33	11.21	12.78	12.82	15.10	12.90
			L=12	4.28	11.66	4.52	11.48	11.75	8.26	11.37	10.80
			L=8,10	4.24	11.39	4.50	11.37	12.58	10.39	11.76	11.66
		Cons-KD (Ours)		**3.92**	**10.56**	**4.13**	**10.32**	**19.18**	**16.92**	**19.02**	**19.81**

TABLE 2. Comparison of WER (%) and RERR (%) on LibriSpeech when applying beam-search decoding with neural rescorer. Bold represents superior results.

Model	# of Params.	KD Method		WER (%)				RERR (%)			
				dev		test		dev		test	
				clean	other	clean	other	clean	other	clean	other
Teacher	122 M	None		1.97	4.67	2.05	4.85	-	-	-	-
Student	9 M	None		3.02	9.80	3.21	9.26	-	-	-	-
		Guided CTC [12]		2.95	9.02	3.10	8.89	2.32	7.96	3.43	4.00
		SKD [10]		2.86	8.33	2.99	8.32	5.30	15.00	6.85	10.15
		Inter-KD [11]	L=6	2.92	8.85	3.05	8.58	3.31	9.69	4.98	7.34
			L=8	2.80	8.85	3.03	8.24	7.28	9.69	5.61	11.02
			L=10	2.84	8.59	2.91	8.46	5.96	12.35	9.35	8.64
			L=12	2.92	8.88	3.14	8.67	3.31	9.39	2.18	6.37
			L=8,10	2.96	8.86	3.01	8.76	1.99	9.59	6.23	5.40
		Cons-KD (Ours)		**2.69**	**8.18**	**2.89**	**7.85**	**10.93**	**16.53**	**9.97**	**15.23**

B. MAIN RESULTS

1) LIBRISPEECH

Table 1 shows the WER and RERR results on the LibriSpeech dataset with greedy decoding, comparing Cons-KD with other competing KD methods. The key difference between Cons-KD and other methods lied in its ability to address the issue of inconsistency during the KD process. It is important to note that the proposed framework did not require additional parameters for student sub-model sampling; all student sub-models share the same set of parameters. From the results, it is verified that the student with Cons-KD yielded the best performance for all configurations. Specifically, the distilled student model achieved WER 3.92 % on the dev-clean dataset and WER 4.13 % on the test-clean dataset, corresponding to RERR 19.18 % and RERR 19.02 %, respectively. On the more challenging 'other' datasets, the student model using Cons-KD achieved WER 10.56 % on the dev-other and WER 10.32 % on the test-other, with RERRs of 16.92 % and 19.81 %, respectively. In the context of Inter-KD, various configurations for the intermediate CTC layer were examined. It is found that adding the intermediate CTC layer to the 8^{th} layer of the Conformer-CTC model ($L = 8$) resulted in better performance than the other layer configurations, including $L = 6$, $L = 10$, $L = 12$, and a combined $L = 8, 10$ setup. Although Inter-KD with $L = 8$ produced promising results, Cons-KD outperformed it by a significant margin. This indicates that the dropout-robust student using Cons-KD could greatly improve its overall performance by mitigating inconsistency caused by dropout. Compared to the distilled student using Inter-KD ($L = 8$), which showed RERR 11.76 % and RERR 11.66 % on test-clean and test-other datasets, the student model using Cons-KD achieved notably higher RERRs of 19.03 % on the test-clean dataset and 19.81 % on the test-other dataset. Unlike other methods, Cons-KD ensured that the outputs of the student sub-models remained consistent and reliable, even with the internal variations introduced by dropout. The results confirmed that the proposed approach not only enhanced the effectiveness of KD but also boosted the robustness of the distilled model compared to other KD methods, thereby improving the quality of the student model. The ability of Cons-KD to enhance the student model's robustness and overall performance contributed to its superiority and highlighted its potential as the effective approach for KD.

Additionally, we performed experiments using LM decoding. We employed the Transformer-based LM as a neural rescoring tool, as previously mentioned. Table 2 gives the WER and RERR results when incorporating the LM into the decoding process. For Guided CTC, even though there were performance improvements over the student baseline, the improvements were relatively marginal. Regarding Inter-KD, which achieved promising results among conventional techniques, its effectiveness was not maintained in beam-search decoding with the LM. Although the configuration with $L = 8$ yielded better performance than other configurations, the differences among these other configurations were not

TABLE 3. Comparison of WER (%) on Common Voice Spanish benchmark when applying greedy decoding. Bold represents superior results.

Model	# of Params.	KD Method	WER (%)	
			dev	test
Teacher	114 M	None	8.89	9.86
Student	8 M	None	16.84	17.82
		Guided CTC [12]	16.76	18.02
		SKD [10]	16.94	17.97
		Inter-KD [11]	16.78	17.91
		Cons-KD (Ours)	**15.91**	**16.98**

TABLE 4. Comparison of WER (%) when applying different consistency regularization terms.

Consistency Regularization Term	dev		test	
	clean	other	clean	other
l_2 difference	4.07	11.11	4.37	11.03
$\mathcal{L}^k_{cons}$ in Eq. (7)	**3.92**	**10.56**	**4.13**	**10.21**

substantial, as shown in Table 2. Moreover, while Inter-KD outperformed SKD in the majority of configurations with greedy decoding, it did not maintain this lead in beam-search decoding with the LM. For instance, in the previous experiments with greedy decoding, the distilled student model with Inter-KD ($L = 8$) significantly outperformed the SKD model. However, Inter-KD lagged behind SKD in most configurations when using beam-search decoding with the LM. The distilled students using Inter-KD ($L = 8$) and SKD produced WER 8.85 % and WER 8.33 % respectively on the dev-clean dataset. These results indicate that integrating LM into the decoding process was notably more challenging than employing greedy decoding. Interestingly, we verified that Cons-KD consistently showed the best performance among the conventional KD approaches, even in scenarios involving decoding with the LM. In particular, the distilled student with Cons-KD achieved the lowest WER of 2.69 % on dev-clean and 2.89 % on test-clean, corresponding RERR 10.93 % and RERR 9.97 %, respectively. Furthermore, Cons-KD also resulted in the lowest WER on the more challenging 'other' datasets, recording 8.18 % on dev-other and 7.85 % on test-other, translating to RERR 16.53 % and 15.23 %. These results underscore the proposed method's substantial effectiveness in KD.

2) COMMON VOICE

In order to show the versatility of the proposed method, we also conducted our experiment on Common Voice 7.0 Spanish benchmark. For the teacher model, we used a pre-trained Hybrid Transducer-CTC FastConformer [33], provided by the NeMo toolkit. During the KD process, we transferred the CTC predictions of the teacher model, utilizing the same competing methods as in the previous experiments. Table 3 presents the WER results on Common Voice 7.0, showing that Cons-KD still performed effectively with the Spanish dataset. The results confirm that conventional KD methods achieved minimal performance improvement and, in some cases, performed worse than their student baseline. However, the proposed method improved the student's performance in all configurations, indicating its effectiveness.

TABLE 5. WER (%) results on LibriSpeech with different loss combinations.

Training Objective	dev		test	
	clean	other	clean	other
$\sum_{k=1}^{K}(\mathcal{L}^k_{ctc})$	4.85	12.71	5.10	12.87
$\sum_{k=1}^{K}(\mathcal{L}^k_{ctc}) + \mathcal{L}_{kd}$	4.21	11.19	4.48	11.40
$\sum_{k=1}^{K}(\mathcal{L}^k_{ctc} + \mathcal{L}^k_{cons})$	4.20	11.26	4.39	11.02
$\sum_{k=1}^{K}(\mathcal{L}^k_{ctc} + \mathcal{L}^k_{cons}) + \mathcal{L}_{kd}$	**3.92**	**10.56**	**4.13**	**10.32**

C. COMPARISON OF CONSISTENCY REGULARIZATION OBJECTIVES

The proposed consistency regularization term $\mathcal{L}^k_{cons}$ was formulated to minimize the l_2 loss between the average prediction $\bar{h}(x; \theta_{stu})$ and the output of each student sub-model $h(x, s_k; \theta_{stu})$ while stopping the gradients of the average prediction during backpropagation. This proposed objective aimed to reduce the variance among the predictions. As supported by the analytical findings of Zolna et al. [19], minimizing the l_2 difference between predictions is also equivalent to minimizing the variance in predictions obtained from different independent and identically distributed (i.i.d.) dropout masks. As a result, we compared the proposed consistency objective $\mathcal{L}^k_{cons}$ with the alternative consistency term, which minimized the l_2 difference between the softmax predictions of the sub-models. Table 4 reports the WER results. To ensure a fair comparison, we only modified the consistency regularization term while keeping the other proposed losses unchanged. From the results, we verified that $\mathcal{L}^k_{cons}$, which was inspired by the sample variance, outperformed the alternative approach that measured the l_2 difference between the predictions. Recent studies in self-supervised learning (SSL) have shown that utilizing the stop gradient operation was effective for training siamese networks [38], [39], [40], [41], [42], [43]. Consistent with these findings, our study further highlighted the efficacy of incorporating the stop gradient operation in training the student sub-models.

D. ABLATION STUDY: EFFECT OF EACH TRAINING OBJECTIVE

In the proposed KD framework, there are three training objectives to train the student: the original CTC objective in Eq. (5), the KD loss in Eq. (6), and the consistency regularization term in Eq. (7). In this section, we proceeded to verify the effect of each objective function on the student model's performance. As shown in Table 5, we evaluated WERs for different loss combinations. The ablation study set the number of student sub-models K to 3. From the results, it is verified that the incorporation of the KD loss alongside the CTC loss $\sum_{k=1}^{K}(\mathcal{L}^k_{ctc}) + \mathcal{L}_{kd}$ led to the performance improvement in WER, demonstrating the

TABLE 6. Comparison of WER (%) when applying different K.

# of Student Sub-model Sampling	dev		test	
	clean	other	clean	other
$K = 2$	4.01	10.88	4.18	10.65
$K = 3$	**3.92**	**10.56**	**4.13**	**10.32**

effectiveness of KD in the training of the student model. Different from conventional KD methods that depended solely on the output of a single sub-model, our KD objective was designed to minimize the difference between the averaged predictions of the student sub-models and the prediction of the teacher model. Additionally, except for the dev-other, further improvement was observed when the consistency regularization term was added to the CTC loss $\sum_{k=1}^{K}(\mathcal{L}_{ctc}^{k} + \mathcal{L}_{cons}^{k})$, suggesting that promoting consistency within the model contributes positively to its ability to generalize. The best performance was achieved when all the proposed objective functions were applied together $\sum_{k=1}^{K}(\mathcal{L}_{ctc}^{k} + \mathcal{L}_{cons}^{k}) + \mathcal{L}_{kd}$. While the combination of the CTC and KD losses already yielded satisfactory results, the addition of the consistency regularization term could further improve the student model's performance. This combination achieved a WER of 3.92 % on dev-clean, 10.56 % on dev-other, 4.13 % on test-clean, and 10.32 % on test-other, respectively, underscoring the synergistic effect of these training objectives.

E. ABLATION STUDY: EFFECT OF NUMBER OF STUDENT SUB-MODELS

Table 6 shows the impact of varying the number of student sub-model samplings K on the WER performance. When K is set to 2, the distilled student model's WER performance was 4.01 % for the dev-clean and 10.88 % for the dev-other, respectively. For the test sets, the model with $K = 2$ recorded a WER of 4.18 % for the test-clean and 10.65 % for the test-other. An increase in the number of sub-models to $K = 3$ resulted in WER performance improvement. The distilled student with $K = 3$ achieved WER 3.92 % for the dev-clean and WER 10.56 % for the dev-other. This enhancement was also evident in the test datasets, with the test-clean WER decreasing to 4.13 % and the test-other WER to 10.32 %. These findings imply that employing a greater number of sub-models, as indicated by a higher K value, can lead to improved model performance. This suggests that the model gains from the varied predictions made during the ensemble training process, enhancing its overall predictive accuracy.

F. ANALYSIS: EFFECT OF λ_{KD}

In the proposed KD framework, we considered the tunable parameter λ_{kd} in Eq. (6), which was employed to balance the distillation loss $\mathcal{L}_{kd}$. We explored the impact of λ_{kd} on Cons-KD performance, as depicted in Figure 2. We evaluated the student model on LibriSpeech subsets, including dev-clean, dev-other, test-clean, and test-other, while varying λ_{kd}. The values of λ_{kd} were chosen from the set $\{0.125, 0.25, 0.5, 1.0\}$.

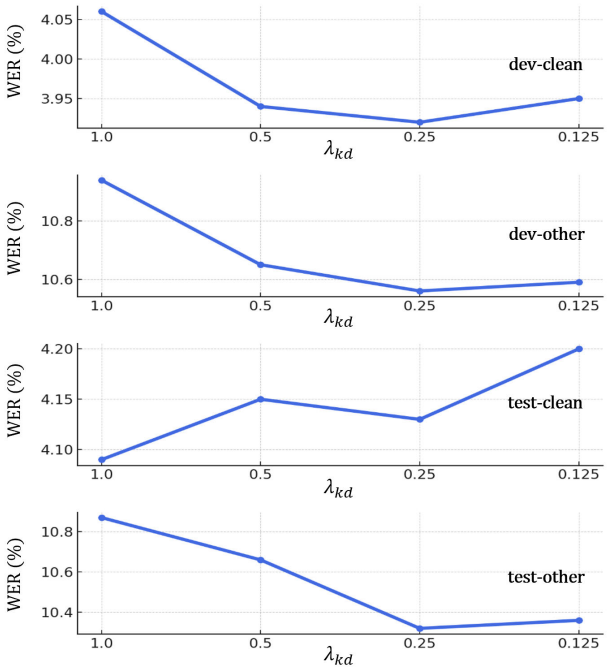


FIGURE 2. WER performance across different values of λ_{kd}.

TABLE 7. Comparison of the relative training times. The baseline trained without KD method was set to 1.

KD method		Relative Training Time
None		x1
Guided CTC [12]		x1.14
SKD [10]		x1.14
Cons-KD(Ours)	K=2	x2.24
	K=3	x2.86

From the results, it is confirmed that $\lambda_{kd} = 0.25$ yielded better performance compared to the other settings in most configurations.

V. LIMITATION

The proposed framework required multiple forward passes of the student model, resulting in a longer training time compared to previous KD methods, as shown in Table 7. Despite the comparatively longer training time, each forward pass for the lightweight student model was not significantly long. Moreover, considering the performance improvement, the multiple forward passes were reasonable. In future work, we aim to explore techniques to reduce the training time without compromising the performance improvements.

VI. CONCLUSION

In this work, we introduced Cons-KD, a novel approach that combines consistency regularization and KD to effectively improve the student's performance. By minimizing the inconsistency between the training and inference stages, Cons-KD could train a more dropout-robust student model, leading to significant performance improvements. From experimental

results on the LibriSpeech dataset, we confirmed that Cons-KD outperformed conventional KD methods that did not consider the inconsistency problem of the student. The success of Cons-KD highlighted the importance of addressing dropout-related inconsistency in KD frameworks. We expect that the proposed approach will provide a broader impact on KD research, benefiting various types of models across different tasks.

REFERENCES

[1] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. 23rd ICML*, 2006, pp. 369–376.

[2] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. ICASSP*, 2016, pp. 4960–4964.

[3] A. Vaswani, "Attention is all you need," in *Proc. Adv. Neural Inform. Process. Syst. (NIPS)*, 2017, pp. 5998–6008.

[4] A. Graves, "Sequence transduction with recurrent neural networks," in *Proc. ICML Workshop Represent. Learn.*, 2012, pp. 1–20.

[5] J. Li, V. Lavrukhin, B. Ginsburg, R. Leary, O. Kuchaiev, J. M. Cohen, H. Nguyen, and R. T. Gadde, "Jasper: An end-to-end convolutional neural acoustic model," 2019, *arXiv:1904.03288*.

[6] S. Majumdar, J. Balam, O. Hrinchuk, V. Lavrukhin, V. Noroozi, and B. Ginsburg, "Citrinet: Closing the gap between non-autoregressive and autoregressive end-to-end models for automatic speech recognition," 2021, *arXiv:2104.01721*.

[7] A. Kalinov, S. Majumdar, J. Balam, and B. Ginsburg, "Carnelinet: Neural mixture model for automatic speech recognition," in *Proc. ASRU*, 2021, pp. 1–11.

[8] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech*, pp. 1–18, Oct. 2020.

[9] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Proc. NIPS Workshop Deep Learn.*, 2014, pp. 1–18.

[10] J. W. Yoon, H. Lee, H. Y. Kim, W. I. Cho, and N. S. Kim, "TutorNet: Towards flexible knowledge distillation for end-to-end speech recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 29, no. 1, pp. 1626–1638, May 2021.

[11] J. W. Yoon, B. J. Woo, S. Ahn, H. Lee, and N. S. Kim, "Inter-KD: Intermediate knowledge distillation for CTC-based automatic speech recognition," in *Proc. IEEE Spoken Language Technology Workshop*, Jul. 2022, pp. 1–27.

[12] G. Kurata and K. Audhkhasi, "Guiding CTC posterior spike timings for improved posterior fusion and knowledge distillation," in *Proc. Interspeech*, Sep. 2019, pp. 1616–1620.

[13] G. Kurata and K. Audhkhasi, "Improved knowledge distillation from bi-directional to uni-directional LSTM CTC for end-to-end speech recognition," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Dec. 2018, pp. 411–417.

[14] R. Takashima, S. Li, and H. Kawai, "An investigation of a knowledge distillation method for CTC acoustic models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 5809–5813.

[15] R. Takashima, L. Sheng, and H. Kawai, "Investigation of sequence-level knowledge distillation methods for CTC acoustic models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 6156–6160.

[16] A. Senior, H. Sak, F. de Chaumont Quitry, T. Sainath, and K. Rao, "Acoustic modelling with CD-CTC-SMBR LSTM RNNS," in *Proc. IEEE Workshop Autom. Speech Recognit. Understand. (ASRU)*, Dec. 2015, pp. 604–609.

[17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *JMLR*, vol. 15, pp. 1929–1958, Jul. 2014.

[18] X. Ma, Y. Gao, Z. Hu, Y. Yu, Y. Deng, and E. Hovy, "Dropout with expectation-linear regularization," in *Proc. ICLR*, 2017, pp. 1–28.

[19] K. Zolna, D. Arpit, D. Suhubdy, and Y. Bengio, "Fraternal dropout," in *Proc. ICLR*, 2018, pp. 1–10.

[20] X. Liang, L. Wu, J. Li, Y. Wang, Q. Meng, T. Qin, W. Chen, M. Zhang, and T. Liu, "R-drop: Regularized dropout for neural networks," in *Proc. NIPS*, 2021, pp. 1–9.

[21] J. Li, R. Zhao, J.-T. Huang, and Y. Gong, "Learning small-size DNN with output-distribution-based criteria," in *Proc. Interspeech*, Sep. 2014, pp. 1–25.

[22] Y. Chebotar and A. Waters, "Distilling knowledge from ensembles of neural networks for speech recognition," in *Proc. Interspeech*, Sep. 2016, pp. 3439–3443.

[23] S. Watanabe, T. Hori, J. Le Roux, and J. R. Hershey, "Student-teacher network learning with enhanced features," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 5275–5279.

[24] L. Lu, M. Guo, and S. Renals, "Knowledge distillation for small-footprint highway networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 4820–4824.

[25] T. Fukuda, M. Suzuki, G. Kurata, S. Thomas, J. Cui, and B. Ramabhadran, "Efficient knowledge distillation from an ensemble of teachers," in *Proc. Interspeech*, Aug. 2017, pp. 3697–3701.

[26] K. J. Geras, A. Mohamed, R. Caruana, G. Urban, S. Wang, O. Aslan, M. Philipose, M. Richardson, and C. Sutton, "Blending LSTMs into CNNs," in *Proc. ICLR Workshop*, 2016, pp. 1–29.

[27] Y. Kim and A. M. Rush, "Sequence-level knowledge distillation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 1–32.

[28] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," in *Proc. ICLR*, 2015, pp. 1–36.

[29] J. Lee and S. Watanabe, "Intermediate loss regularization for CTC-based speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 6224–6228.

[30] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 5206–5210.

[31] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," in *Proc. LREC*, 2020, pp. 4211–4215.

[32] V. Noroozi, S. Majumdar, A. Kumar, J. Balam, and B. Ginsburg, "Stateful conformer with cache-based inference for streaming automatic speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2024, pp. 12041–12045.

[33] D. Rekesh, N. R. Koluguri, S. Kriman, S. Majumdar, V. Noroozi, H. Huang, O. Hrinchuk, K. Puvvada, A. Kumar, J. Balam, and B. Ginsburg, "Fast conformer with linearly scalable attention for efficient speech recognition," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop (ASRU)*, Dec. 2023, pp. 1–8.

[34] O. Kuchaiev, J. Li, H. Nguyen, O. Hrinchuk, R. Leary, B. Ginsburg, S. Kriman, S. Beliaev, V. Lavrukhin, J. Cook, P. Castonguay, M. Popova, J. Huang, and J. M. Cohen, "NeMo: A toolkit for building AI applications using neural modules," 2019, *arXiv:1909.09577*.

[35] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*, Sep. 2019, pp. 2613–2617.

[36] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. ICLR*, 2019, pp. 1–33.

[37] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 1715–1725.

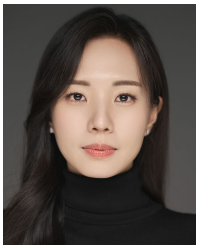
[38] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, "Data2vec: A general framework for self-supervised learning in speech, vision and language," in *Proc. Int. Conf. Mach. Learn. (ICML)*, vol. 162, 2022, pp. 1298–1312.

[39] A. Baevski, A. Babu, W.-N. Hsu, and M. Auli, "Efficient self-supervised learning with contextualized target representations for vision, speech and language," 2022, *arXiv:2212.07525*.

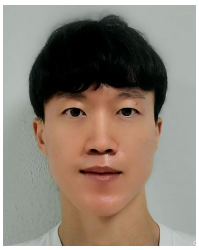
[40] X. Chen and K. He, "Exploring simple Siamese representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15745–15753.

[41] J. B. Grill, F. Strub, F. Altch, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent: A new approach to self-supervised learning," in *Proc. NIPS*, 2020, pp. 1–15.

[42] M. Caron, H. Touvron, I. Misra, H. Jegou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9630–9640.
[43] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Proc. NIPS*, 2017, pp. 1–27.



JI WON YOON received the B.S. degree in electrical engineering from Korea University, Seoul, South Korea, in 2018, and the M.S. and Ph.D. degrees in electrical and computer engineering from Seoul National University (SNU), in 2020 and 2024, respectively. Since 2024, she has been with the Department of Artificial Intelligence, Chung-Ang University, where she is currently a Professor. Her research interests include speech signal processing, speech recognition, speech/audio coding, speech synthesis, adaptive signal processing, and knowledge distillation.



HYEONSEUNG LEE received the B.S. degree in computer engineering and the Ph.D. degree in electrical and computer engineering from Seoul National University (SNU), in 2024. Since 2024, he has been a Research Scientist with XL8 Inc. His research interests include speech recognition, speech enhancement, real-time speech processing, and speech signal processing.

JU YEON KANG (Student Member, IEEE) received the B.S. degree in electrical engineering from Korea University, Seoul, South Korea, in 2023. She is currently pursuing the M.S. degree with the School of Electrical Engineering, Seoul National University (SNU). Her research interests include speech signal processing, speech recognition, speech/audio coding, speech synthesis, adaptive signal processing, and knowledge distillation.

NAM SOO KIM (Senior Member, IEEE) received the B.S. degree in electronics engineering from Seoul National University (SNU), Seoul, South Korea, in 1988, and the M.S. and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology, in 1990 and 1994, respectively. From 1994 to 1998, he was as a Senior Member of Technical Staff with the Samsung Advanced Institute of Technology. Since 1998, he has been with the School of Electrical Engineering, SNU, where he is currently a Professor. His research interests include speech signal processing, speech recognition, speech/audio coding, speech synthesis, adaptive signal processing, machine learning, and mobile communication.