Sampling-Based Pruned Knowledge Distillation for Training Lightweight RNN-T

Sungsoo Kim, Dongjune Lee, Ju Yeon Kang, Myeonghun Jeong, *Graduate Student Member, IEEE*, and Nam Soo Kim, *Senior Member, IEEE*

***Abstract*—We present a novel training method for small-scale RNN-T models, widely used in real-world speech recognition applications. Despite efforts to scale down models for edge devices, the demand for even smaller and more compact speech recognition models persists to accommodate a broader range of devices. In this letter, we propose Sampling-based Pruned Knowledge Distillation (SP-KD) for training lightweight RNN-T models. In contrast to the conventional knowledge distillation techniques, the proposed method enables student models to distill knowledge from the distribution of teacher models, which is estimated by considering not only the best paths but also less likely paths. Additionally, we leverage pruning the output lattice of RNN-T to comprehensively transfer knowledge from teacher models to student models. Experimental results demonstrate that our proposed method outperforms the baseline in training tiny RNN-T models.**

***Index Terms*—Knowledge distillation, RNN-T, speech recognition.**

I. Introduction

In RECENT years, the proliferation of end-to-end (E2E) models in speech recognition has led to remarkable advances across numerous applications. While on-device speech recognition models based on Recurrent Neural Network Transducer (RNN-T) [1] running on mobile devices are approaching the performance of server-side models [2], their deployment in resource-constrained environments remains challenging. The advent of tiny models offers a promising solution, facilitating computational efficiency and reduced memory footprint.

In real-world applications, it is crucial to output recognition results in a streaming manner. As stated in [3], RNN-T has advantages over typical encoder-decoder architectures [4], [5] in streaming applications since RNN-T specifically allows "no-predict" to be decoded as one of the output tokens, as mentioned in [6]. One of the smaller and more efficient RNN-T models, Convolution-augmented Transformer (Conformer) [7] outperforms previous ASR models by integrating elements of both Convolutional Neural Networks (CNNs) [8] and Transformers [9]. Another optimized model, Zipformer [10], which employs a U-Net-like architecture [11], has been studied to enhance memory and computational efficiency during training. However, the performance of tiny models often lags behind their larger counterparts.

One of the promising techniques to reduce this performance gap is Knowledge Distillation (KD) [12], which has been widely used for transferring knowledge from teacher models to student models. Extensive research on RNN-T model compression [13], [14], [15] has been performed based on response-based KD [16], which uses the logits obtained from the last output layer of the teacher models. This response-based KD is known as soft targets [16], which are distributions estimated by a softmax function at the frame level. However, for speech recognition, it is more desirable for KD to distill knowledge from the distributions not only at the frame level but also across entire sequence level. As a solution, [17], inspired by sequence-level KD [18], efficiently approximated the distribution using the teacher's best paths. Although this method performs better than learning from scratch, the approximation is inherently limited because the distribution does not fully represent the sequence distribution across all possible paths. Therefore, there remains a need to transfer knowledge from a more generalized sequence distribution.

However, estimating the sequence distribution of the teacher models seems to be a challenging task. To address this challenge, we propose Sampling-based Pruned Knowledge Distillation (SP-KD), which utilizes a Monte Carlo sampling technique to estimate the sequence distributions of the teacher models, taking into account both the best paths and less likely paths. Another issue in KD for RNN-T models is a requirement for a large amount of memory, since the loss of RNN-T models is proportional to input sequence length, target sequence length, and output size. Although [15] compressed the output tokens into three categories: correct, blank, and the rest, we instead propose to prune the output lattice of RNN-T to extensively distill knowledge from the teacher models. To the best of our knowledge, we are the first to propose this method for RNN-T and demonstrate its utility through experiments.

II. Recurrent Neural Network Transducer

Consider an input sequence $\mathbf{x} = x_{1:T} = \{x_1, x_2, \ldots, x_T\}$ with length T and an output label sequence $\mathbf{y} = y_{1:U} =$

Received 31 October 2024; revised 30 December 2024; accepted 31 December 2024. Date of publication 13 January 2025; date of current version 21 January 2025. The associate editor coordinating the review of this article and approving it for publication was Dr. Jonas Borgstrom. *(Corresponding author: Nam Soo Kim.)*

The authors are with the Department of Electrical and Computer Engineering, Seoul National University, Seoul 08826, South Korea, and also with the Institute of New Media and Communications, Seoul National University, Seoul 08826, South Korea (e-mail: sskim@hi.snu.ac.kr; djlee@hi.snu.ac.kr; jykang@hi.snu.ac.kr; mhjeong@hi.snu.ac.kr; nkim@snu.ac.kr).

Digital Object Identifier 10.1109/LSP.2025.3528364

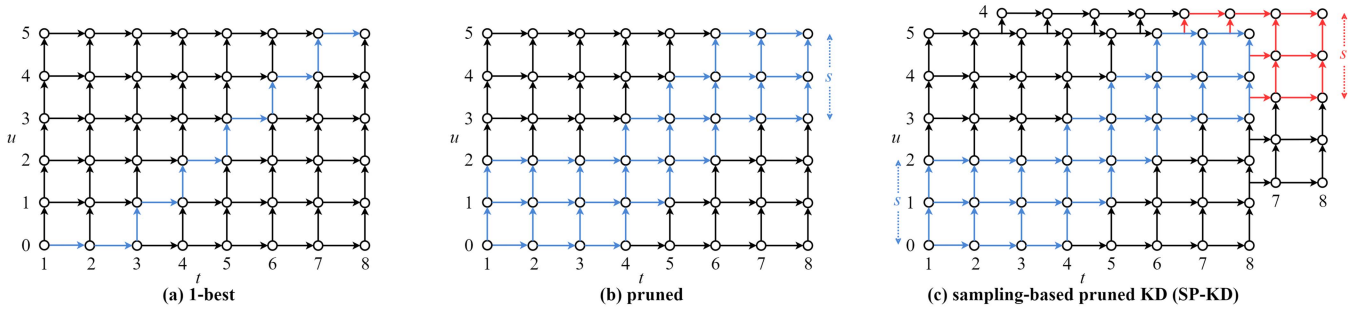

Fig. 1. Visualization of the output lattice of RNN-T. Blue indicates the output lattice from the ground truth, whereas red indicates the output lattice from the sample label.

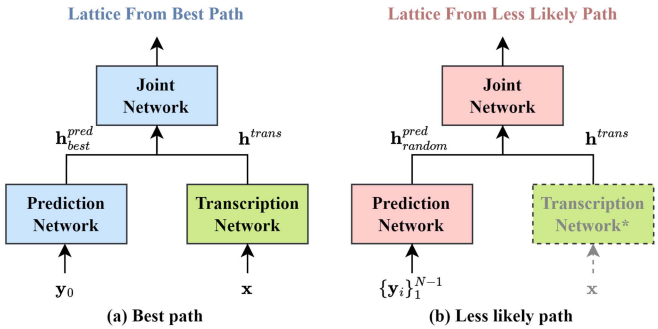

Fig. 2. Overview of our proposed sampling-based method. "Transcription Network*" indicates that the output of the transcription network is reused.

$\{y_1, y_2, \ldots, y_U\}$ with length U, where $y_u \in V$ and V represents the set of labels. RNN-T introduces "blank" ϕ into an RNN-T alignment $\mathbf{a} = a_{1:T+U} = \{a_1, a_2, \ldots, a_{T+U}\}$ to align variable-length input and output sequences [1]. The output label sequence $\mathbf{y}$ is mapped to a certain alignment $\mathbf{a}$, where $a_{t+u} \in V' = V \cup \{\phi\}$. Let $\mathcal{B}$ denote a function that removes the blank symbols from the alignment $\mathbf{a}$. Given $\mathbf{x}$ and $\mathbf{y}$, the conditional marginal probability of $\mathbf{y}$ is computed under the RNN-T framework as follows:

$$P(\mathbf{y} \mid \mathbf{x}) = \sum_{\mathbf{a} \in \mathcal{B}^{-1}(\mathbf{y})} P(\mathbf{a} \mid \mathbf{x}). \tag{1}$$

In (1), $P(\mathbf{y}|\mathbf{x})$ is computed via a $T \times U$ output probability lattice [15] as shown in Fig. 1.

As depicted in Fig. 2, the basic RNN-T architecture comprises three main components: a transcription network (encoder), a prediction network (decoder), and a joint network. Let $\mathbf{h}^{trans} = h^{trans}_{1:T}$ and $\mathbf{h}^{pred} = h^{pred}_{0:U}$ stand for the hidden vectors extracted from the transcription network and the prediction network of RNN-T, respectively. At each lattice node (t, u), where $t \in \mathbb{N}$ and $1 \le t \le T$, and $u \in \mathbb{N}_0$ and $0 \le u \le U$, the transition probability $P_v(t, u)$ of RNN-T can be calculated as

$$\begin{aligned} z_{t,u} &= \text{Joint}(h^{trans}_t, h^{pred}_u), \\ a_{1:n-1} &= \mathcal{B}^{-1}(y_{1:u-1}) \ \text{ where n = t+u}, \\ P(a_n = v \,|\, x_{1:T}, a_{1:n-1}) &= P(a_n = v \,|\, z_{t,u}) \\ &= \frac{\exp(z^v_{t,u})}{\sum_{v' \in V'} \exp(z^{v'}_{t,u})}, \\ P_v(t,u) &\equiv P(a_n = v \,|\, x_{1:T}, a_{1:n-1}), \end{aligned} \tag{2}$$

where the hidden vectors h^{trans}_t and h^{pred}_u are mapped to the output vector $z_{t,u}$ of dimension $|V'|$ through the joint network (Joint), and $z^v_{t,u}$ denotes the v-th element of $z_{t,u}$. Although $P(\mathbf{y} \mid \mathbf{x})$ in (1) can be efficiently computed using a forward-backward algorithm based on dynamic programming [1], as detailed in [15], training RNN-T models demands a memory allocation proportional to $2 \times T \times U \times |V'|$ per utterance. For further details on RNN-T, the reader is referred to [1].

III. Sequence-Level Knowledge Distillation

Let $\tilde{P}(\mathbf{y} \mid \mathbf{x})$ and $P(\mathbf{y} \mid \mathbf{x})$ denote the sequence distributions of a teacher model and the corresponding student model, respectively. Then, objective function for the sequence-level KD is given by

$$\mathcal{L}_{seq} = \sum_{\mathbf{y}} \tilde{P}(\mathbf{y} \mid \mathbf{x}) \ln \frac{\tilde{P}(\mathbf{y} \mid \mathbf{x})}{P(\mathbf{y} \mid \mathbf{x})}. \tag{3}$$

In (3), since $\mathcal{L}_{seq}$ accounts for all the possible label sequences $\mathbf{y}$, the sequence-level KD enables the teacher model to transfer a broader range of knowledge to the student model [18]. However, as pointed out in [18], taking all the possible sequence $\{\mathbf{y}\}$ into account is intractable. One promising way of approximation is

$$\begin{aligned} \tilde{P}(\mathbf{y}|\mathbf{x}) &\approx \delta(\mathbf{y} - \mathcal{M}(\mathbf{x})) \\ \text{where } \mathcal{M}(\mathbf{x}) &= \text{argmax}_{\mathbf{y}} \tilde{P}(\mathbf{y} \mid \mathbf{x}). \end{aligned} \tag{4}$$

As a result, the sequence-level KD can be approximated by

$$\mathcal{L}_{seq} \approx \tilde{P}(\mathcal{M}(\mathbf{x}) \mid \mathbf{x}) \ln \frac{\tilde{P}(\mathcal{M}(\mathbf{x}) \mid \mathbf{x})}{P(\mathcal{M}(\mathbf{x}) \mid \mathbf{x})}. \tag{5}$$

In the case of RNN-T, when considering all the possible alignments $\mathbf{a}$, (5) can be formulated as

$$\mathcal{L}_{\text{RNN-T}} = \sum_{\mathbf{a} \in \mathcal{B}^{-1}(\mathcal{M}(\mathbf{x}))} \tilde{P}(\mathbf{a} \mid \mathbf{x}) \ln \frac{\tilde{P}(\mathbf{a} \mid \mathbf{x})}{P(\mathbf{a} \mid \mathbf{x})}, \tag{6}$$

where $\tilde{P}(\mathbf{a} \mid \mathbf{x})$ and $P(\mathbf{a} \mid \mathbf{x})$ are the alignment distributions of the teacher model and the student model, respectively. Let $\mathcal{I}$ be a function that returns node indices from the RNN-T alignment. By using soft targets instead of hard targets, the objective function of the original RNN-T KD [15] is given by

$$\mathcal{L}_{\text{original}} = \sum_{(t,u) \in \mathcal{I}(\mathcal{B}^{-1}(\mathcal{M}(\mathbf{x})))} \sum_{v \in V'} \tilde{P}_v(t,u) \ln \frac{\tilde{P}_v(t,u)}{P_v(t,u)}. \tag{7}$$

However, as detailed in [15], performing KD on RNN-T models using (7) requires a large amount of memory due to the memory

complexity $O(T \times U \times |V'|)$. In [15], instead of distilling the transition probability distributions for the $|V'|$-dimensional logits, KD is performed after reducing them to three dimensions: target y, blank ϕ, and the remainder r. This reduction can be represented as

$$\mathcal{L}_{\text{3-dims}} = \sum_{(t,u)\in\mathcal{I}(\mathcal{B}^{-1}(\mathcal{M}(\mathbf{x})))} \sum_{v\in\{y,\phi,r\}} \tilde{P}_v(t,u) \ln \frac{\tilde{P}_v(t,u)}{P_v(t,u)}. \quad (8)$$

Consequently, (8) reduces the memory complexity to $O(T \times U \times 3)$.

Despite the reduction achieved by (8), as demonstrated in [17], this approximation overlooks the correlation across different output tokens. To address this issue, [17] proposed to distill knowledge only over the best path in the output lattice of all possible alignments. This approach is based on the assumption presented in [19] that the sequence distribution of RNN-T models is concentrated in a specific region. Using an approximation similar to that of (4), the sequence-level KD in [17] can be approximated by

$$\tilde{P}(\mathbf{a} \mid \mathbf{x}) = \delta(\mathbf{a} - \mathcal{A}(\mathbf{x})) \text{ where } \mathcal{A}(\mathbf{x}) = \text{argmax}_{\mathbf{a}} \tilde{P}(\mathbf{a} \mid \mathbf{x}),$$

$$\mathcal{L}_{\text{RNN-T}} \approx \mathcal{L}_{\text{1-best}} = \sum_{(t,u)\in\mathcal{I}(\mathcal{A}(\mathbf{x}))} \sum_{v\in V'} \tilde{P}_v(t,u) \ln \frac{\tilde{P}_v(t,u)}{P_v(t,u)}. \quad (9)$$

The memory complexity of the formulation in (9) is $O((T+U) \times |V'|)$ [17]. In addition, the output lattice of (9) can be visualized, as illustrated in Fig. 1(a).

IV. Proposed Method

Although (9) is a simple way to do KD at sequence level, considering only the single best path is too limited to represent the true sequence distribution. In order to address this issue, we propose a more sophisticated method to distill the sequence distribution of the teacher model. Let us assume that an input sequence $\mathbf{x}$ is included in a mini-batch $\mathcal{D}_{\text{mini-batch}} = \{(\mathbf{x}^m, \mathbf{y}^m) \mid m \in \mathbb{Z}_M = \{0, 1, \ldots, M-1\}\}$. If $\mathbf{x} = \mathbf{x}^m$, then we let

$$\mathbf{y}^{-m} = \{\mathbf{y}^n \mid n \neq m, n \in \mathbb{Z}_M\}. \quad (10)$$

In the mini-batch, $\mathbf{x}^m$ and $\mathbf{y}^{-m}$ are uncorrelated, allowing the student model to learn less likely paths of the teacher model. We propose a sampling distribution $Q(\mathbf{y} \mid \mathbf{x})$ based on a mini-batch $\mathcal{D}_{\text{mini-batch}}$. Suppose that we sample N label sequences $\mathbf{y}_0, \ldots, \mathbf{y}_{N-1}$ given an input $\mathbf{x} = \mathbf{x}^m$.

Then, the proposed sampling distribution $Q(\mathbf{y} \mid \mathbf{x})$ is given by

$$\begin{cases} Q(\mathbf{y}_0 \mid \mathbf{x} = \mathbf{x}^m) = \frac{\alpha}{N}, & \text{for } \mathbf{y}_0 = \mathbf{y}^m = \mathcal{M}(\mathbf{x}^m) \\ Q(\mathbf{y}_i \mid \mathbf{x} = \mathbf{x}^m) = \frac{N-\alpha}{N(N-1)}, & \text{for } \mathbf{y}_i \sim \text{Uniform}(\mathbf{y}^{-m}) \end{cases}$$

$$\text{where } i = 1, \ldots, N-1 \text{ and } \alpha \in (0, N). \quad (11)$$

Given $\mathbf{x}$ and sampled label sequences $\{\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_{N-1}\}$, our proposed KD method is based on the Monte Carlo approximation as follows:

$$\mathcal{L}_{\text{proposed}} = \sum_{\mathbf{y}} Q(\mathbf{y} \mid \mathbf{x}) \frac{\tilde{P}(\mathbf{y} \mid \mathbf{x})}{Q(\mathbf{y} \mid \mathbf{x})} \ln \frac{\tilde{P}(\mathbf{y} \mid \mathbf{x})}{P(\mathbf{y} \mid \mathbf{x})}$$

$$\approx \frac{1}{N} \sum_{i=0}^{N-1} \frac{\tilde{P}(\mathbf{y}_i \mid \mathbf{x})}{Q(\mathbf{y}_i \mid \mathbf{x})} \ln \frac{\tilde{P}(\mathbf{y}_i \mid \mathbf{x})}{P(\mathbf{y}_i \mid \mathbf{x})} \text{ with } \mathbf{y}_i \sim Q(\mathbf{y} \mid \mathbf{x}). \quad (12)$$

The formulation in (12) can be further decomposed into

$$\begin{aligned} \mathcal{L}_{\text{decomposed}} &= \tilde{P}(\mathbf{y}_0 \mid \mathbf{x}) \ln \frac{\tilde{P}(\mathbf{y}_0 \mid \mathbf{x})}{P(\mathbf{y}_0 \mid \mathbf{x})} \\ &\quad + \frac{\alpha(N-1)}{N-\alpha} \sum_{i=1}^{N-1} \tilde{P}(\mathbf{y}_i \mid \mathbf{x}) \ln \frac{\tilde{P}(\mathbf{y}_i \mid \mathbf{x})}{P(\mathbf{y}_i \mid \mathbf{x})} \\ &= \mathcal{L}^{best} + \lambda \mathcal{L}^{less}, \end{aligned} \quad (13)$$

where $\mathcal{L}^{best}$ and $\mathcal{L}^{less}$ denote the objective functions of the sequence-level KD for the best paths and less likely paths, respectively, and $\lambda = \frac{\alpha(N-1)}{N-\alpha}$ stands for a hyperparameter to weight the KD loss for less likely paths. Given $\mathbf{x}$ and the sampled label sequences $\{\mathbf{y}_0, \ldots, \mathbf{y}_{N-1}\}$, the RNN-T output lattice can be obtained as depicted in Fig. 2. Notably, for $\{\mathbf{y}_i\}_{i=1}^{N-1} = \{\mathbf{y}_1, \ldots, \mathbf{y}_{N-1}\}$, the hidden vector $\mathbf{h}^{pred}_{random}$ is re-extracted from the prediction network while keeping $\mathbf{h}^{trans}$ fixed, as shown in Fig. 2(b). Accordingly, the proposed sequence-level KD in (12) not only considers the single output label sequence but also accounts for all possible diverse label sequences from the teacher model.

However, as discussed in Section III, a naive computation of (13) consumes significant memory resources. To deal with this issue, inspired by pruned RNN-T [19] as shown in Fig. 1(b), we propose Sampling-based Pruned Knowledge Distillation (SP-KD), which limits the range of the output lattice extracted from RNN-T models at each step from U to a hyperparameter S as illustrated in Fig. 1(c). Let π_S, for which more detailed information can be found in [19], denote a function that prunes the output lattice within the range S, and $\tilde{R}_v(t,u)$ and $R_v(t,u)$ represent the transition probabilities from the teacher model and the student model, respectively, when given $\mathbf{x}$ and $\{\mathbf{y}_i\}_{i=1}^{N-1}$. Finally, our SP-KD is given by

$$\begin{aligned} \mathcal{L}_{\text{SP-KD}} &= \mathcal{L}^{best}_{\text{pruned}} + \lambda \mathcal{L}^{less}_{\text{pruned}} \\ &= \sum_{(t,s)\in\mathcal{I}(\pi_S(\mathcal{B}^{-1}(\mathbf{y}_0)))} \sum_{v\in V'} \tilde{P}_v(t,s) \ln \frac{\tilde{P}_v(t,s)}{P_v(t,s)} \\ &\quad + \lambda \sum_{(t,s)\in\mathcal{I}(\pi_S(\mathcal{B}^{-1}(\{\mathbf{y}_i\}_{i=1}^{N-1})))} \sum_{v\in V'} \tilde{R}_v(t,s) \ln \frac{\tilde{R}_v(t,s)}{R_v(t,s)}, \end{aligned} \quad (14)$$

where $\mathcal{L}^{best}_{\text{pruned}}$ and $\mathcal{L}^{less}_{\text{pruned}}$ are the pruned KD losses for the best paths and less likely paths, respectively. Consequently, our proposed SP-KD facilitates accounting for various paths, not only the single best path from the output lattice of the teacher model but also other diverse paths. Moreover, the complexity of (14) for each KD loss is reduced from $O(T \times U \times |V'|)$ to $O(T \times S \times |V'|)$, where $U \gg S$.

V. Experiments

A. Experimental Setup

1) Data preparation: We used the LibriSpeech corpus [20] and the AISHELL-1 corpus [21] for evaluating the performance

TABLE I
DETAILED CONFIGURATION OF TRANSCRIPTION NETWORK

Configuration	Teacher	Student
layer-numbers	2,4,3,2,4	1,1,1,1,1
encoder-dim.	384,384,384,384,384	196,196,196,196,196
number of head	8,8,8,8,8	4,4,4,4,4
feedforward-dim.	1024,1024,2048,2048,1024	256,256,512,512,256

of the proposed technique. As the speech feature, we used an 80-dimensional log mel filterbank energy, which was extracted using a 25 ms Hanning window with a stride of 10 ms. The output tokens consisted of 500 B-pair encodings (BPEs) [22] in the case of LibriSpeech and 4136 characters in the case of AISHELL-1.

2) Implementation details: The toolkits k2[1] and icefall[2] were utilized for training and evaluation in all experiments, which were run on two NVIDIA RTX 2080Ti and two RTX 3090 GPUs. We applied SpecAugment [23] and mixing Musan [24] to the acoustic features exclusively for the transcription network input. We trained our models for 30 epochs to ensure convergence. The initial learning rate was set to 0.035 with the Adam optimizer [25]. For the batch size, we fixed the maximum duration at 70, except for the LibriSpeech 960 hrs experiment, which was set to 200. The range S was 5 for LibriSpeech, and 10 for AISHELL-1. For the SP-KD experiments, we sampled only one label sequence $\mathbf{y}_1$ corresponding to each $\mathbf{y}_0$. To obtain the word error rate (WER) and the character error rate (CER), we used a modified beam search algorithm that limits the maximum number of symbols emitted per frame to one under the RNN-T framework.

3) Model architecture: Our RNN-T model utilized the Zipformer architecture [10] for the transcription network, as shown in Table I. For the prediction network, we employed a 512-dimensional stateless prediction network [26]. Finally, we adopted a joint network that produces a softmax output corresponding to the number of the output tokens V'. The total number of parameters for the models is 70.4 M and 6.17 M for the teacher and student models, respectively, on the LibriSpeech dataset, and 79.4 M and 13.6 M for those on AISHELL-1.

4) Baseline: We trained the student model without teacher models as our baseline. Additionally, we measured the performance of the student model trained using the original RNN-T KD loss in (7), considering it as a resource-unlimited scenario. To evaluate the effectiveness of our proposed method, we compared it with those of the previous studies [15], [17]. Specifically, we examined the performance with respect to the N-best KD loss. Furthermore, we conducted experiments using the LibriSpeech and AISHELL-1 datasets to verify data dependencies.

B. Results and Analysis

We present the results of our SP-KD on LibriSpeech in Table II, where "pruned" indicates that only $\mathcal{L}_{\text{pruned}}^{best}$ is used for the KD loss with λ to 0. As shown in Table II, our proposed method demonstrates superior performance on LibriSpeech compared to the previous methods, which indicates the efficacy of the proposed method in distilling knowledge from the distribution of teacher models. Additionally, the results in Table III confirm that our proposed method works effectively on the Mandarin dataset, AISHELL-1. In particular, our SP-KD method significantly outperforms the previous approaches, even including the "original" KD method. Furthermore, in the case of "pruned" where $\lambda = 0$, our method still exhibits better performance on both LibriSpeech and AISHELL-1 when compared to those of "3-dims" and "1-best". The performance of "pruned" is notably superior to that of "N-best". We speculate that the proposed "pruned" method, which considers all possible alignments within the pruned lattice, is more effective since the alignments from the N-best decoding results do not significantly differ from that of the 1-best. However, given that the SP-KD method achieves the best performance in our experiments, we conjecture that our proposed SP-KD more effectively approximates the sequence probability of the teacher model using the Monte Carlo method. Consequently, the results in Table II and III demonstrate that our SP-KD method is not only applicable across languages but also accurately estimates the sequence path distribution by considering both the best paths and less likely paths.

TABLE II
WER (%) PERFORMANCE ON LIBRISPEECH

Condition (Param.)	100 hrs		960 hrs	
	test-clean	test-other	test-clean	test-other
Teacher (70.4M)	7.09	18.76	2.66	6.29
Student (6.17M)	9.56	24.39	4.60	11.08
+ KD in original	8.45	22.49	4.45	**10.75**
+ KD in 3-dims [15]	9.16	23.07	4.53	11.05
+ KD in 1-best [17]	8.88	22.97	4.36	11.05
+ KD in N-best (N=2)	8.82	22.78	4.48	11.19
proposed				
+ KD in pruned (λ=0.00, S=5)	8.51	22.51	4.48	10.94
+ KD in SP-KD (λ=0.50, S=5)	**8.39**	**21.79**	**4.34**	10.96
+ KD in SP-KD (λ=1.00, S=5)	8.40	22.26	4.41	10.99

TABLE III
CER (%) PERFORMANCE ON AISHELL-1

Condition (Param.)	test	dev
Teacher (79.4M)	6.64	5.99
Student (13.6M)	9.21	8.61
+ KD in original	7.07	6.43
+ KD in 3-dims [15]	8.23	7.82
+ KD in 1-best [17]	7.90	7.26
+ KD in N-best (N=3)	7.75	7.20
proposed		
+ KD in pruned (λ=0.00, S=10)	7.11	6.51
+ KD in SP-KD (λ=0.75, S=10)	**6.83**	**6.23**
+ KD in SP-KD (λ=1.00, S=10)	6.93	6.33

VI. CONCLUSION

We have proposed a novel training method for lightweight RNN-T models using Sampling-based Pruned KD (SP-KD). The experimental results verify the effectiveness of our proposed methodology. For future work, we plan to improve distilling with more extensive databases and extend our proposed method to other architectures and tasks.

[1]https://github.com/k2-fsa/k2
[2]https://github.com/k2-fsa/icefall

REFERENCES

[1] A. Graves, "Sequence transduction with recurrent neural networks," 2012, *arXiv:1211.3711*.

[2] T. N. Sainath et al., "A streaming on-device end-to-end model surpassing server-side conventional model quality and latency," in *Proc. 2020 IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 6059–6063.

[3] K. Rao, H. Sak, and R. Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducer," in *Proc. 2017 IEEE Autom. Speech Recognit. Understanding Workshop*, 2017, pp. 193–199.

[4] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. 2016 IEEE Int. Conf. Acoust., Speech Signal Process.*, 2016, pp. 4960–4964.

[5] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Proc. 2016 IEEE Int. Conf. Acoust., Speech Signal Process.*, 2016, pp. 4945–4949.

[6] C.-C. Chiu et al., "A comparison of end-to-end models for long-form speech recognition," in *Proc. 2019 IEEE Autom. Speech Recognit. Understanding Workshop*, 2019, pp. 889–896.

[7] A. Gulati et al., "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech* 2020, pp. 5036–5040.

[8] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[9] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 6000–6010.

[10] Z. Yao et al., "Zipformer: A faster and better encoder for automatic speech recognition," 2023, *arXiv:2310.11230*.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image Comput. Comput.-Assist. Interv., 18th Int. Conf.*, Munich, Germany, 2015, pp. 234–241.

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

[13] N. Dawalatabad et al., "Two-pass end-to-end ASR model compression," in *Proc. 2021 IEEE Autom. Speech Recognit. Understanding Workshop*, 2021, pp. 403–410.

[14] J. Park et al., "Conformer-based on-device streaming speech recognition with KD compression and two-pass architecture," in *Proc. 2023 IEEE Spoken Lang. Technol. Workshop*, 2023, pp. 92–99.

[15] S. Panchapagesan, D. S. Park, C.-C. Chiu, Y. Shangguan, Q. Liang, and A. Gruenstein, "Efficient knowledge distillation for RNN-transducer models," in *Proc. 2021 IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 5639–5643.

[16] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *Int. J. Comput. Vis.*, vol. 129, no. 6, pp. 1789–1819, 2021.

[17] X. Yang, Q. Li, and P. C. Woodland, "Knowledge distillation for neural transducers from large self-supervised pre-trained models," in *Proc. 2022 IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 8527–8531.

[18] Y. Kim and A. M. Rush, "Sequence-level knowledge distillation," in *Proc. 2016 Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 1317–1327.

[19] F. Kuang et al., "Pruned RNN-T for fast, memory-efficient ASR training," in *Proc. Interspeech*, 2022, pp. 2068–2072.

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

[21] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "AISHELL-1: An open-source Mandarin speech corpus and a speech recognition baseline," in *Proc. 20th Conf. Oriental Chapter Int. Coordinating Committee Speech Databases Speech I/O Syst. Assessment*, 2017, pp. 1–5.

[22] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics (Volume 1: Long Papers)*, 2016, pp. 1715–1725.

[23] D. S. Park et al., "Specaugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*, 2019, Art. no. 2613.

[24] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," 2015, *arXiv:1510.08484v1*.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[26] M. Ghodsi, X. Liu, J. Apfel, R. Cabrera, and E. Weinstein, "RNN-transducer with stateless prediction network," in *Proc. 2020 IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 7049–7053.