INTER-KD: INTERMEDIATE KNOWLEDGE DISTILLATION FOR CTC-BASED AUTOMATIC SPEECH RECOGNITION

Ji Won Yoon[1], Beom Jun Woo[1], Sunghwan Ahn[1], Hyeonseung Lee[1], and Nam Soo Kim[1]

[1]Department of ECE and INMC, Seoul National University, Seoul, Republic of Korea

ABSTRACT

Recently, the advance in deep learning has brought a considerable improvement in the end-to-end speech recognition field, simplifying the traditional pipeline while producing promising results. Among the end-to-end models, the connectionist temporal classification (CTC)-based model has attracted research interest due to its non-autoregressive nature. However, such CTC models require a heavy computational cost to achieve outstanding performance. To mitigate the computational burden, we propose a simple yet effective knowledge distillation (KD) for the CTC framework, namely Inter-KD, that additionally transfers the teacher's knowledge to the intermediate CTC layers of the student network. From the experimental results on the LibriSpeech, we verify that the Inter-KD shows better achievements compared to the conventional KD methods. Without using any language model (LM) and data augmentation, Inter-KD improves the word error rate (WER) performance from 8.85 % to 6.30 % on the test-clean.

Index Terms— Speech recognition, connectionist temporal classification, teacher-student learning, knowledge distillation

1. INTRODUCTION

In recent years, there has been remarkable progress in end-to-end speech recognition that directly converts an input speech into the corresponding text without any prior alignment information. Compared with the traditional deep neural network (DNN)-hidden Markov model (HMM) hybrid systems, the end-to-end framework simplifies the overall pipeline while achieving better performance.

Among the various types of end-to-end models for speech recognition, connectionist temporal classification (CTC) [1] has attracted increasing interest due to its non-autoregressive (NAR) nature. The NAR model requires $M(\ll N)$ iterations when producing an N-length target sequence. On the other hand, the autoregressive (AR) model costs N iterations, indicating that the NAR framework enables a significant inference speedup over the AR one.

However, existing CTC-based models require high computational cost and long training time to achieve promising results. For their practical deployment in resource-limited set-

(a) Previous KD approach

(b) Proposed KD approach

Fig. 1: Conceptual diagram of Inter-KD compared to the conventional KD. The orange line represents KD with the teacher's output-level knowledge. Different from the conventional KD, the proposed method uses multiple intermediate CTC layers for KD.

tings, there have been continuous efforts to apply knowledge distillation (KD), an effective technique for model compression, to CTC models. The main idea of KD is to transfer the knowledge from a large and powerful teacher model to a small student model. In the speech recognition task, conventional KD methods typically consider the teacher network's output-level knowledge (sentence prediction, softmax prediction, etc.) for distillation. By minimizing the distance between the predictions of the teacher and the student, the distilled student can achieve better performance than the baseline trained only with the target label.

In order to train the student more effectively, in this paper, we propose a novel KD framework for CTC models,

namely Inter-KD. As depicted in Fig. 1, we design the student model's architecture using multiple intermediate CTC layers. Each CTC layer is trained with the output-level knowledge of the teacher in conjunction with the CTC loss. The proposed architecture of the student is similar to a deeply supervised net [2] and an intermediate CTC [3], where both methods add supervision to the hidden layers for performance improvement. The main difference with previous deeply supervised schemes is that Inter-KD trains the intermediate CTC layers with KD instead of using only ground-truth labels. Since the auxiliary CTC layers can be ignored during the inference, a small additional computational load is required only for training.

From the experimental results on the LibriSpeech [4] dataset, it is confirmed that Inter-KD shows better performance than the conventional KD methods. For test-clean dataset, Inter-KD improves the word error rate (WER) performance of the student from 8.85 % to 6.30 % without using any language model (LM) and data augmentation, achieving relative error rate reduction (RERR) 28.81 %. We conduct additional analysis to further check the effect of KD with the intermediate CTC layers.

Our main contributions can be summarized as follows:

- We introduce a new KD framework for CTC models, namely Inter-KD. In the proposed scheme, we newly design the architecture of the student by attaching multiple intermediate CTC layers in the middle of the network. The student can be trained more effectively by transferring the teacher's knowledge to the intermediate CTC layers.

- According to the experimental results on the LibriSpeech dataset, we verify the effectiveness of the Inter-KD. When transferring the output-level knowledge of the teacher, Inter-KD yields better performance than other previous KD methods.

2. RELATED WORK

2.1. Connectionist temporal classification

For given source input $x_{1:T}$ and target label $y_{1:N}$, the CTC framework [1] can directly convert x into y by using the additional token "blank". Unlike the traditional hybrid system, the predefined alignment knowledge is not required. CTC considers all possible alignments compatible with y to compute the conditional probability of y. When training the CTC model, we minimize the following objective:

$$- \log p(y|x) = - \log \sum_{a \in \beta^{-1}(y)} p(a|x) \qquad (1)$$

where β denotes a many-to-one mapping function for CTC, and a represents the intermediate alignments, which include the blank token. $\beta^{-1}(y)$ returns the possible set of alignments.

Even though the CTC framework provides efficient decoding, there is a strong conditional independence assumption between the output tokens, resulting in relatively poor performance compared to the AR models. Recently, there have been some attempts to close the gap between the CTC and AR models. Chan *et al.* [5] and Higuchi *et al.* [6] used the additional network to refine the initial output from the CTC. Majumdar *et al.* [7] proposed an improved CTC-based architecture that combines a QuartzNet [8] with the squeeze and excitation [9]. Lee and Watanabe [3] introduced an intermediate CTC loss, which uses the intermediate layer in the encoder network and its corresponding CTC loss to improve the performance of the model.

2.2. Knowledge distillation

Hinton *et al.* [10] first proposed the concept of KD, which transfers knowledge by minimizing Kullback-Leibler (KL)-divergence between the predictions of the teacher and the student. Since the large and powerful teacher model is utilized to guide the training of the small student model, the student can produce better performance compared with the case when it is solely trained with the ground-truth labels. With steadily increasing interest in on-device speech recognition, there have been several efforts to develop KD for speech recognition models. For the DNN-HMM hybrid system, previous KD studies typically trained the student by minimizing cross-entropy (CE) loss between the posterior probability of the teacher and the student [11, 12, 13, 14, 15, 16]. However, it is challenging to train CTC-based speech recognition models with the same CE criteria. According to the previous studies [17, 18, 19], applying the CE-based KD technique to the CTC-based models can worsen the performance compared to the baseline model trained only with the target label. To cover this issue, Takashima *et al.* [18] attempted to apply sequence-level KD [20] to the CTC model. Kurata and Audhkhasi [21, 22] suggested the KD framework that can train the low-latency student model with the knowledge of the high-latency teacher model and also proposed guided CTC training that distills the spike timings from the teacher. Yoon *et al.* [23] introduced softmax-level KD (SKD) that uses l_2 loss instead of KL divergence when distilling frame-level posterior of the CTC-based teacher model.

3. PROPOSED METHOD

3.1. Student model architecture

As shown in Fig. 2, we newly design the architecture of the student. Different from the conventional student model, multiple CTC layers are added to the intermediate layers of the student. In our experiments, additional CTC layers are attached to three layers: 18th, 24th, and 30th layers of the student, whose network is composed of 33 depthwise separable convolutional layers. For the convenience of notation, we let

Fig. 2: An overview of Inter-KD for CTC models.

"intermediate CTC layer" denotes the auxiliary CTC layer in the middle of the student, and "original CTC layer" denotes the last CTC layer of the student. Each CTC layer is combined with a fully-connected layer and the softmax function. Note that the fully-connected layers in intermediate CTC are not shared with the original CTC's fully-connected layer. The proposed architecture of the student is similar to some deep supervision-based networks [2, 3] that add supervision to the hidden layers. The main difference is that the intermediate CTC layers are trained with KD instead of using only ground-truth labels.

3.2. Intermediate CTC layer

Intermediate CTC layers are located in the middle of the student network, as depicted in Fig. 2. These layers are only used in the training procedure and can be removed during the inference, so a small additional computational load is required only for training. When training the student, the intermediate CTC layers are trained via KD in conjunction with CTC loss. The CTC loss function of ith intermediate CTC layer is given as

$$\mathcal{L}^i_{\text{InterCTC}} = CTC_{\text{loss}}(y, g_i(x)) \qquad (2)$$

where x, y, $g_i(x)$, and CTC_{loss} represent the source input, the corresponding label, the softmax output of the ith intermediate CTC layer, and the CTC loss, respectively. Regardless of the layer position, the conventional deep supervised learning [2, 3] added supervision to the hidden layers with the same target. Since we followed the conventional framework, we used the same target labels for each intermediate layer.

The second loss source is the KD loss function. As for the KD technique, we adopt the SKD [23] due to its effective

improvement for CTC models. The SKD applies l_2 loss for transferring the knowledge. Therefore, the KD loss function for i^{th} intermediate CTC layer can be computed as

$$\mathcal{L}^i_{\text{InterKD}} = ||f_{tea}(x) - g_i(x)||^2_2 \qquad (3)$$

where $f_{tea}(x)$ denotes the softmax prediction of the teacher.

3.3. Original CTC layer

The original CTC layer is attached to the last layer of the student model. We use both KD and CTC losses for training the original CTC layer. Firstly, the CTC loss function of the original CTC layer is as follows:

$$\mathcal{L}_{\text{OrigCTC}} = CTC_{\text{loss}}(y, f_{stu}(x)) \qquad (4)$$

where $f_{stu}(x)$ denotes the softmax value of the original CTC layer. $\mathcal{L}_{\text{OrigCTC}}$ loss is exactly same as the vanilla CTC training. The KD loss for the original CTC layer is formulated as

$$\mathcal{L}_{\text{OrigKD}} = ||f_{tea}(x) - f_{stu}(x)||^2_2. \qquad (5)$$

3.4. Training

In the Inter-KD framework, there are two kinds of training losses to improve the performance of the student.

- Loss 1: CTC loss from target labels. The supervision of CTC is added not only to the original CTC layer, but also to the intermediate CTC layers.

- Loss 2: KD loss using the teacher model's softmax prediction. All CTC layers are trained with the knowledge of the teacher.

Table 1: Comparison of WER (%) and RERR (%) on LibriSpeech using greedy decoding. "Ours" denotes Inter-KD with $K = 3$. The best result is in bold.

Model	Params.	WER (%) w/o LM				RERR (%) w/o LM			
		dev		test		dev		test	
		clean	other	clean	other	clean	other	clean	other
Teacher: Jasper DR	332.63 M	3.61	11.37	3.77	11.08	-	-	-	-
Student: Jasper Mini		8.66	23.28	8.85	24.26	-	-	-	-
+ Sequence-level KD [18]		8.96	23.73	9.10	24.81	-3.46	-1.93	-2.82	-2.27
+ Guided CTC training [22]	8.19 M	7.81	21.93	8.29	22.49	9.82	5.80	6.33	7.30
+ SKD [23]		7.63	21.36	7.81	22.41	11.89	8.25	11.75	7.63
+ Ours		**6.24**	**18.82**	**6.30**	**19.49**	**27.94**	**19.16**	**28.81**	**19.66**

When there are K intermediate CTC layers in the student model, the CTC loss function $\mathcal{L}_{\text{CTC}}$ for Inter-KD is given as

$$\mathcal{L}_{\text{CTC}} = \mathcal{L}_{\text{OrigCTC}} + \sum_{i=1}^{K} \mathcal{L}_{\text{InterCTC}}^{i}. \qquad (6)$$

The KD loss between the student and the teacher is as follows:

$$\mathcal{L}_{\text{KD}} = \mathcal{L}_{\text{OrigKD}} + \sum_{i=1}^{K} \mathcal{L}_{\text{InterKD}}^{i}. \qquad (7)$$

Thus, the final objective function $\mathcal{L}_{\text{Total}}$ for Inter-KD is given as

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{CTC}} + \lambda \cdot \mathcal{L}_{\text{KD}} \qquad (8)$$

where λ is a tunable parameter to balance $\mathcal{L}_{\text{CTC}}$ and $\mathcal{L}_{\text{KD}}$.

3.5. Inference

The intermediate CTC layers are unaffected during the whole inference procedure. Only the original CTC layer is used to generate the final prediction of the student.

4. EXPERIMENTAL SETTINGS

4.1. Dataset

We evaluated the word error rate (WER) performance on LibriSpeech [4] dataset. In the training phase, "train-clean-100", "train-clean-360", and "train-other-500" were applied. We used "dev-clean", "dev-other", "test-clean", and "test-other" for evaluation.

4.2. Performance metrics

For the performance comparison, we measured WER and relative error rate reduction (RERR). WER is a widely-used metric to quantify the performance of the speech recognition model and RERR is a standard metric to measure the WER improvement compared to the baseline.

4.3. Model configurations

In our experiments, we adopted Jasper DR [24] and Jasper Mini as the teacher and the student models, respectively. Both CTC models had the same label set, which included a total of 29 character labels. For model training and inference, We utilized the OpenSeq2Seq toolkit [25]. For the Jasper DR teacher, we used the pre-trained model checkpoint provided by the OpenSeq2Seq. The Jasper Mini student model consists of 33 depthwise separable 1D convolutional layers.

4.4. Implementation details

We trained the student with 50 epochs for CTC training with KD, and three Titan V GPUs (each with 12GB memory) were used for training. NovoGrad optimizer [26] was adopted for training the student, where the initial learning rate was set to 0.02. As aforementioned in Section 3, we added intermediate CTC layers to three layers: 18th, 24th, and 30th layers of the Jasper Mini. The tunable parameter in the Equation 8 was set to 0.25. When decoding with LM, we used 4-gram KenLM [27], where the beam width was 256.

4.5. Conventional distillation methods for comparison

Since the proposed framework transferred the softmax prediction of the teacher, we mainly compared the Inter-KD with the conventional KD methods that considered the teacher network's output-level knowledge (sentence prediction, softmax prediction, etc.). For performance comparison, we applied three conventional KD techniques for CTC models, including sequence-level KD [18], guided CTC training [22], and SKD [23].

5. EXPERIMENTAL RESULTS

5.1. Performance comparison

Table 1 shows the WER and RERR results with greedy decoding, comparing the performance improvement of the Inter-

Table 2: Comparison of WER (%) and RERR (%) using the 4-gram LM. "Ours" denotes Inter-KD with $K = 3$. The best result is in bold.

| Model | Params. | WER (%) w/ LM | | | | RERR (%) w/ LM | | | |
| | | dev | | test | | dev | | test | |
		clean	other	clean	other	clean	other	clean	other
Teacher: Jasper DR	332.63 M	3.04	9.52	3.69	9.38	-	-	-	-
Student: Jasper Mini		4.83	15.53	5.24	16.40	-	-	-	-
+ Sequence-level KD [18]		5.16	15.54	5.48	16.91	-6.83	-0.06	-4.58	-3.11
+ Guided CTC training [22]	8.19 M	5.17	15.94	5.58	16.85	-7.04	-2.64	-6.49	-2.74
+ SKD [23]		4.77	15.01	5.26	15.96	1.24	3.35	-0.38	2.68
+ Ours		**4.61**	**14.55**	**4.99**	**15.19**	**4.55**	**6.31**	**4.77**	**7.38**

KD with conventional KD techniques. From the results, it is verified that the proposed KD method considerably improved the WER performance of the student compared to other KD methods. The distilled student using Inter-KD achieved WER 6.24 % and WER 6.30 % on dev-clean and test-clean, corresponding to RERR 27.94 % and RERR 28.81 %. In the case of test-other, Inter-KD gave WER 19.49 % and RERR 19.66 %. Among the conventional KD approaches, SKD yielded better achievements than sequence-level KD and guided CTC training. Still, the best WER performance was obtained when applying the Inter-KD to the student model.

Also, we conducted experiments based on the LM decoding. As presented in Table 2, applying LM was more challenging than greedy decoding. Sequence-level KD and guided CTC training had a negative RERR value for all configurations, and SKD had a little improvements compared to the case when utilizing greedy decoding. Compared to the previous results in Table 1, the conventional KD techniques did not perform well with the LM decoding. However, it is confirmed that Inter-KD gave significant improvements in all configurations, even when decoding with LM. The proposed KD achieved WER 4.99 % and WER 15.19 % on test-clean and test-other, providing RERR 4.77 % and RERR 7.38 %, respectively.

5.2. Analysis

5.2.1. Effect of the number of intermediate CTC layers

In addition to the performance comparison, we set different number of intermediate CTC layers to verify the impact of K, which denotes the number of intermediate CTC layers. Table 3 gives the WER and RERR results on dev-clean by setting different K from 1 to 3. When K was set to 1, we attached one intermediate CTC layer to the 30th layer of the student model. In the case of $K = 2$, the intermediate CTC layers were added to 24th and 30th layers of the student. The intermediate layers are attached to 18th, 24th, and 30th layers of the student when $K = 3$. From the results, we confirmed that,

Table 3: WER (%) and RERR (%) on LibriSpeech dev-clean using greedy decoding. Different number of intermediate CTC layers (=K) were set from 1 to 3. RERR measured the WER improvement compared to the baseline, where the student baseline provided WER 8.66 % on dev-clean.

# of intermediate CTC layers	KD	WER (%)	RERR (%)
K=3	O	**6.24**	**27.94**
	X	6.60	23.79
K=2	O	6.29	27.37
	X	7.37	14.90
K=1	O	6.42	25.87
	X	8.20	5.31

regardless of applying KD, the performance progressively improved as the K increased from 1 to 3.

5.2.2. Performance comparison with deep supervised learning

The conventional deep supervised learning [2], such as intermediate CTC [3], added supervision to the hidden layers. Therefore, training the intermediate CTC layers without KD was similar to the deep supervised scheme. In addition to the previous experimental results, we continued to conduct the performance comparison between Inter-KD and the deep supervised learning. The results in Table 3 show that Inter-KD achieved better improvements compared to the case without KD, where multiple intermediate CTC layers were trained only with the ground-truth. With the setting of $K = 1$, there was relatively little achievement without KD, providing RERR 5.31 %. We observed that the distilled student ($K = 1$) using Inter-KD was improved considerably, yielding RERR 25.87 %. In the case of $K = 2$, the model provided WER 7.37 % without KD, and the performance was further im-

Table 4: WER (%) on LibriSpeech when greedy decoding was applied. K was set to 3.

Intermediate CTC layer index	WER (%)			
	dev		test	
	clean	other	clean	other
1	11.91	28.91	12.03	29.81
2	7.36	21.00	7.39	21.76
3	6.30	18.97	6.37	19.48

proved when applying Inter-KD (WER 6.29 %). We verified that applying KD for each intermediate CTC layer performed well for all configurations, including $K = 1$, $K = 2$, and $K = 3$. Our best improvement was obtained when applying $K = 3$ with Inter-KD.

5.2.3. Performance of each intermediate CTC layer

Since we used multiple CTC intermediate layers that can produce the intermediate ASR prediction, we measured the WER performance of each intermediate CTC layer. Table 4 summarizes the results of each intermediate CTC layer when applying $K = 3$. Layer indexes 1, 2, and 3 indicate each intermediate layer attached to the 18th, 24th, and 30th layers of the student model, respectively. From the results, the intermediate CTC with index 1 achieved WER 12.03 % and WER 29.81 % on test-clean and test-other datasets, indicating a substantial WER degradation compared to other layer indexes. In the case of index 3, we observed a minimal degradation for all configurations, compared with the previous results of Inter-KD in Table 1. Interestingly, for the test-other, the WER performance of layer index 3 (WER 19.48 %) was slightly better than that of the original Inter-KD (WER 19.49 %). This implies that intermediate CTC layers can produce confident ASR predictions via the Inter-KD. If we properly use these intermediate outputs for the early exit framework, the ASR result can be returned early instead of finishing the whole inference procedure. In other words, the Inter-KD gave more possibilities for accelerating the model's inference speed.

6. CONCLUSION

In this paper, we proposed a simple yet effective KD method for CTC models, namely Inter-KD. In the proposed KD framework, we newly designed the architecture of the student by adding multiple intermediate CTC layers in the middle of the student network. These intermediate layers were trained with the teacher's knowledge in conjunction with the ground-truth labels. From the experimental results on LibriSpeech, it is confirmed that Inter-KD can effectively improve the WER performance of the student. Also, the proposed KD achieved

better improvements for all configurations compared to the conventional output-level KD methods. The detailed analysis was performed for each intermediate CTC layer, and we gave the possibility that the proposed KD can be applied to the early exit framework, accelerating the model's inference speed.

7. REFERENCES

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

[2] C. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Proc. AISTATS*, 2015.

[3] J. Lee and S. Watanabe, "Intermediate loss regularization for ctc-based speech recognition," in *Proc. ICASSP*, 2021.

[4] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Proc. ICASSP*, 2015, pp. 5206–5210.

[5] W. Chan, C. Saharia, G. Hinton, M. Norouzi, and N. Jaitly, "Imputer: sequence modelling via imputation and dynamic programming," in *Proc. ICML*, 2020.

[6] Y. Higuchi et al., "Mask ctc: non-autoregressive end-to-end asr with ctc and mask predict," in *Proc. INTERSPEECH*, 2020.

[7] 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," *arXiv preprint arXiv:2104.01721v1*, 2021.

[8] S. Kriman, K. Beliaev, B. Ginsburg, J. Huang, O. Kuchaiev, V. Lavrukhin, R. Leary, J. Li, and Y. Zhang, "Quartznet: deep automatic speech recognition with 1d time-channel separable convolutions," *arXiv preprint arXiv:1910.10261*, 2019.

[9] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. CVPR*, 2018.

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

[11] J. Li, R. Zhao, T. J. Huang, and Y. Gong, "Learning small-size dnn with output-distribution-based criteria," in *Proc. INTERSPEECH*, 2014.

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

[13] S. Watanabe, T. Hori, J. L. Roux, and J. R. Hershey, "Student-teacher network learning with enhanced features," in *Proc. ICASSP*, 2017, pp. 5275–5279.

[14] L. Lu, M. Guo, and S. Renals, "Knowledge distillation for small-footprint highway networks," in *Proc. ICASSP*, 2017, pp. 4820–4824.

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

[16] K. J. Geras et al., "Blending lstms into cnns," in *Proc. ICLR Workshop*, 2016.

[17] A. Senior, H. Sak, F. C. Quitry, T. Sainath, K. Rao, et al., "Acoustic modelling with cd-ctc-smbr lstm rnns," in *Proc. ASRU*, 2015, pp. 604–609.

[18] R. Takashima, S. Li, and H. Kawai, "An investigation of a knowledge distillation method for ctc acoustic models," in *Proc. ICASSP*, 2018, pp. 5809–5813.

[19] R. Takashima, S. Li, and H. Kawai, "Investigation of sequence-level knowledge distillation methods for ctc acoustic models," in *Proc. ICASSP*, 2019, pp. 6156–6160.

[20] Y. Kim and A. M. Rush, "Sequence-level knowledge distillation," in *Proc. EMNLP*, 2016.

[21] G. Kurata and K. Audhkhasi, "Improved knowledge distillation from bi-directional to uni-directional lstm ctc for end-to-end speech recognition," in *Proc. SLT*, 2018, pp. 411–417.

[22] G. Kurata and K. Audhkhasi, "Guiding ctc posterior spike timings for improved posterior fusion and knowledge distillation," in *Proc. INTERSPEECH*, 2019, pp. 1616–1620.

[23] 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 Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1626–1638, 2021.

[24] 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," *arXiv preprint arXiv:1904.03288*, 2019.

[25] O. Kuchaiev, B. Ginsburg, I. Gitman, V. Lavrukhin, J. Li, H. Nguyen, C. Case, and P. Micikevicius, "Mixed-precision training for nlp and speech recognition with openseq2seq," *arXiv preprint arXiv:1805.10387*, 2018.

[26] B. Ginsburg, P. Castonguay, O. Hrinchuk, O. Kuchaiev, V. Lavrukhin, R. Leary, J. Li, H. Nguyen, and J. M. Cohen, "Stochastic gradient methods with layer-wise adaptive moments for training of deep networks," *arXiv preprint arXiv:1905.11286*, 2019.

[27] K. Heafield, "Kenlm: faster and smaller language model queries," in *Proc. EMNLP*, 2011.