

Oracle Teacher: Leveraging Target Information for Better Knowledge Distillation of CTC Models

Ji Won Yoon, *Student Member, IEEE*, Hyung Yong Kim, *Student Member, IEEE*, Hyeonseung Lee, *Student Member, IEEE*, Sunghwan Ahn, *Student Member, IEEE*, and Nam Soo Kim, *Senior Member, IEEE*

Abstract—Knowledge distillation (KD), best known as an effective method for model compression, aims at transferring the knowledge of a bigger network (teacher) to a much smaller network (student). Conventional KD methods usually employ the teacher model trained in a supervised manner, where output labels are treated only as targets. Extending this supervised scheme further, we introduce a new type of teacher model for connectionist temporal classification (CTC)-based sequence models, namely Oracle Teacher, that leverages both the source inputs and the output labels as the teacher model's input. Since the Oracle Teacher learns a more accurate CTC alignment by referring to the target information, it can provide the student with more optimal guidance. One potential risk for the proposed approach is a trivial solution that the model's output directly copies the target input. Based on a many-to-one mapping property of the CTC algorithm, we present a training strategy that can effectively prevent the trivial solution and thus enables utilizing both source and target inputs for model training. Extensive experiments are conducted on two sequence learning tasks: speech recognition and scene text recognition. From the experimental results, we empirically show that the proposed model improves the students across these tasks while achieving a considerable speed-up in the teacher model's training time.

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

I. INTRODUCTION

AS deep neural networks bring a significant improvement in various fields such as speech recognition, computer vision, and natural language processing, they also become wider and deeper. However, as models grow in size and complexity, high-performing neural network models become either computationally expensive or consume a large amount of memory, hindering their wide deployment in resource-limited scenarios. To mitigate this computational burden, several techniques such as model pruning [1], [2], quantization [3], and knowledge distillation [4], [5] have been suggested. Among these approaches, knowledge distillation (KD) is a popular compression scheme, which is the process of transferring knowledge from a deep and complex model (teacher) to a shallower and simpler model (student).

Conventional KD methods typically share a common feature; they require a teacher model with high capacity that has been trained in a supervised manner, where the ground-truth

labels are required as a target. However, training the teacher from scratch can be costly since many of the current state-of-the-art (SOTA) models suffer from excessive training time and difficult hyper-parameters tuning. Thus, some existing approaches rely heavily on the pre-trained model, provided by other prior research, as the teacher to save the training time and resource cost. Even though making full use of the provided pre-trained models is one important motivation of KD, this dependency might limit the flexibility of our consideration. If we can train a better teacher model with fewer resources and training time, KD from various teachers will be possible on different tasks or databases.

We revisit the teacher model in KD from a different perspective. In a conventional KD scenario, there is no guarantee that the teacher can find the correct solution for every complicated problem in an optimal way, implying that the teacher model may provide suboptimal guidance for the student. The key idea of our framework is to derive a more accurate problem-solving process by referring to the existing solutions so that the teacher can provide better guidance to the student. On this basis, we introduce a new type of teacher for Connectionist Temporal Classification (CTC) [6]-based sequence models, namely Oracle Teacher. The conventional teacher is typically built in a supervised manner whose goal is to predict the target output for a given source input data. In contrast, the proposed teacher model utilizes not only the source input but also the target value to estimate better CTC alignment.

However, it may be somewhat confusing to understand what it means to train a model using both the source inputs and the output labels as the model's input. Specifically, the Oracle Teacher is likely to heavily rely on the target input, i.e., the output label, while ignoring the embedding from the source input. To overcome this problem, we propose a training scheme that uses the many-to-one mapping property of the CTC algorithm. Since the relationship between the CTC alignment and the original target is many-to-one, we can prevent a trivial solution that the model's output directly copies the target input. To the best of our knowledge, this is the first attempt of using the target input to improve the ability of the teacher model. Utilizing the target input for training the teacher model brings several benefits for KD. Firstly, the proposed teacher model produces a more accurate CTC alignment by referring to the target information so that its knowledge can provide more optimal guidance to the student. Secondly, the representation of the proposed teacher contains target-related embedding that can be supportive for student training. For example, the Oracle Teacher for automatic speech

The authors are with the Department of Electrical and Computer Engineering and the Institute of New Media and Communications, Seoul National University, Seoul, Korea (e-mail: jwyoona@hi.snu.ac.kr, hykim@hi.snu.ac.kr, hsllee@hi.snu.ac.kr, shahn@hi.snu.ac.kr, nkim@snu.ac.kr) (Corresponding author: Nam Soo Kim).

recognition (ASR) is trained to use both speech and text as the model's input during training. Different from the typical ASR teachers that take only acoustic features into consideration, the Oracle Teacher performs a fusion of both acoustic (speech) and linguistic (text) features when generating the prediction. Since unifying acoustic and linguistic representation learning generally enhances the performance of the speech processing [7]–[11], the Oracle Teacher's representation, which considers not only the acoustic but also linguistic information, can be more effective for the ASR student. Also, the Oracle Teacher can boost up the speed of the training since the target input is used as the guidance to reduce the candidate scope of the prediction. Compared to the conventional teacher models that require tremendous time and GPU resources, our framework dramatically reduces the computational cost required to train the teacher model.

Extensive experiments are conducted on two different sequence learning tasks: ASR and scene text recognition (STR). Empirically, we verify that the student distilled from the Oracle Teacher achieves better performance compared to the case when it is distilled from the other pre-trained models, which yield the high performance for each task. Apart from performance, we also measure the computational cost for training teacher model and shows that a powerful teacher can be trained with a reduced computational burden via the proposed scheme.

Our **main contributions** are summarized as follows:

- 1) Our paper introduces a new type of teacher for CTC-based sequence models, namely Oracle Teacher, that utilizes the output labels as an additional input for model training. The proposed teacher model can estimate a more accurate CTC alignment, providing more optimal guidance to the student. To the best of our knowledge, this is the first attempt of using the target input to improve the performance of a teacher model.
- 2) Through extensive experiments on two sequence learning tasks, including ASR and STR, we verify the superiority of the Oracle Teacher compared to the conventional teacher models. Moreover, our framework dramatically reduces the computational cost of the teacher model in terms of the training time and required GPU resources.
- 3) In a detailed case study and analysis, we validate why the proposed method can result in better KD performance than the conventional teacher and check if the Oracle Teacher is correctly trained while preventing the trivial solution.

II. RELATED WORK

A. Knowledge Distillation

There has been a long line of research on KD, which aims at distilling knowledge from a big teacher model to a small student model. Bucila *et al.* [4] proposed a method to compress an ensemble of models into a single model without significant accuracy loss. Later, Ba and Caruana [12] extended it to deep learning by using the logits of the teacher model. Hinton *et al.* [5] revived this idea under the name of KD

that distills class probability by minimizing the Kullback-Leibler (KL)-divergence between the softmax outputs of the teacher and student. In the case of the ASR task, the most frequently employed KD approach is to train a student with the teacher's prediction as a target, in conjunction with the ground truth. For the conventional deep neural network (DNN)-hidden Markov model (HMM) hybrid systems, Li *et al.* [13] first attempted to apply the teacher-student learning to a speech recognition task, and Wong *et al.* [14] applied sequence-level KD to the acoustic model. Several researchers applied KD to improve the performance by minimizing the frame-level cross-entropy loss between the output distributions of the teacher and student [15]–[19]. For end-to-end speech recognition, KD has been successfully applied to CTC models [20]–[25] and attention-based encoder-decoder models [25]–[28]. However, as reported in previous KD studies [20]–[22], [25], simply applying the frame-level CE to the CTC-based model can worsen the performance compared to the baseline. To cover this problem, Kurata and Audhkhasi [23], [24] proposed KD approaches, where the CTC-based student can be trained using the frame-wise alignment of the teacher. Takashima *et al.* [21], [22] explored sequence-level KD methods for training CTC models. Yoon *et al.* [25] suggested that l_2 loss is more suitable than the conventional KL-divergence to distill frame-level posterior in the CTC framework.

The hidden representation from the teacher also has been proven to hold additional knowledge that can contribute to improving the student's performance. Recently, some KD methods [29]–[36], particularly in computer vision, were proposed to minimize the mean squared error (MSE) between the representation-level knowledge of the two models. They address how to extract a better knowledge from the teacher model and transfer it to the student. Yoon *et al.* [25] first attempted to transfer the the hidden representation across different structured neural networks for end-to-end speech recognition while using frame weighting that reflects which frames are important for KD. Recently, several KD approaches [37]–[39] suggested using the hidden representation-level knowledge to improve the self-supervised speech representation learning-based models, like Hidden-Unit BERT (HuBERT).

B. Connectionist Temporal Classification

Generally, an end-to-end sequence model directly converts a sequence of input features $x_{1:T}$ into a sequence of target labels $y_{1:L}$ where $y_l \in \mathcal{I}$ with \mathcal{I} being the set of labels. T and L are respectively the length of $x = x_{1:T}$ and $y = y_{1:L}$. To cope with the mapping problem when the two sequences have different lengths, the Connectionist Temporal Classification (CTC) framework [6] introduces “blank” label and allows the repetition of each label to force the output and input sequences to have the same length. A CTC alignment $\pi_{1:T}$ is a sequence of initial output labels, as every input x_t is mapped to a certain label $\pi_t \in \mathcal{I}'$ where $\mathcal{I}' = \mathcal{I} \cup \{\text{blank}\}$. A mapping function \mathcal{B} , which is defined as $y = \mathcal{B}(\pi)$, maps the alignment sequence π into the final output sequence y after merging consecutive repeated characters and removing blank labels. The conditional

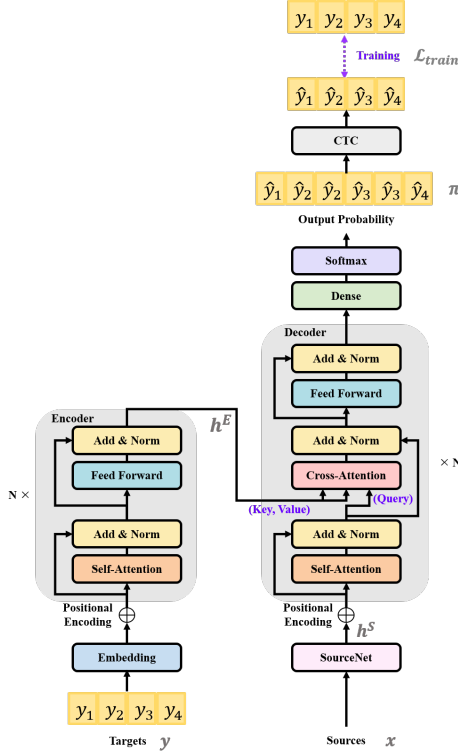


Fig. 1. Overview of the Oracle Teacher. The proposed teacher model mainly consists of three components: the SourceNet, the encoder, and the decoder. Different from the conventional teacher, the target y is used as the additional input to the model. Note that the Oracle Teacher is a non-autoregressive model where the look-ahead mask is not included in the decoder. The architecture selection of the SourceNet depends on the task we are interested in. When the main task is ASR, the SourceNet corresponds to an acoustic model part of the conventional ASR model. In our experiment for ASR, the SourceNet is based on the architecture of Japser [40]. For STR, we apply the CRNN [41] as the SourceNet.

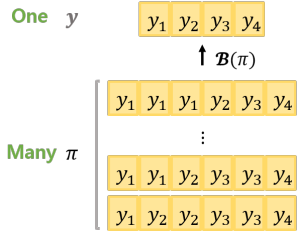


Fig. 2. The relationship between the CTC alignment π and the target input y . A many-to-one mapping function \mathcal{B} converts the alignment sequence π into the final output sequence y .

probability of the target sequence y given the input sequence x is defined as

$$P(y|x) = \sum_{\pi \in \mathcal{B}^{-1}(y)} P(\pi|x). \quad (1)$$

where \mathcal{B}^{-1} denotes the inverse mapping and returns all possible alignment sequences compatible with y . Given a target label sequence y , the loss function \mathcal{L}_{CTC} is defined as:

$$\mathcal{L}_{CTC} = -\log P(y|x). \quad (2)$$

III. ORACLE TEACHER

This section introduces how to design the Oracle teacher that utilizes the output labels as an additional input. As shown in Fig. 1, we let the Oracle Teacher model learn a function from the source x and the target y inputs to the CTC alignment π .

A. Oracle Teacher Training

Let $x = x_{1:T} = \{x_1, \dots, x_T\}$ be an input sequence of length T , and $y = y_{1:L} = \{y_1, \dots, y_L\}$ be a target sequence of length L . As mentioned in Section II-B, the CTC algorithm employs the intermediate CTC alignment $\pi = \pi_{1:T} = \{\pi_1, \dots, \pi_T\}$ to align variable-length input and output sequences. As shown in Fig. 2, the relationship between the alignment π and the target input y is many-to-one via the mapping \mathcal{B} , and this many-to-one setting is the key to training the Oracle Teacher. Intuitively, predicting the CTC alignment π (many) from the target input y (one) should be difficult since many possible paths are compatible with y . Therefore, the proposed model can be trained to use the embeddings of both x and y while preventing a trivial solution where the model's output simply copies the input y . The Oracle Teacher can effectively estimate the most probable CTC alignment π since the target input y is used as guidance to reduce the candidate scope of π . The Oracle Teacher learns the parameters θ to minimize the following training loss:

$$\mathcal{L}_{train} = -\log \sum_{\pi \in \mathcal{B}^{-1}(y)} P(\pi|x, y; \theta) \quad (3)$$

where \mathcal{B} is the many-to-one mapping function in (1) that maps the latent alignment $\pi_{1:T}$ into the target $y_{1:L}$.

B. Knowledge Distillation with Oracle Teacher

We can interpret KD framework from a different perspective by applying the additional target input. Given a source x , the student model learns the parameter ϕ to maximize the following conditional probability:

$$\begin{aligned} \log P(y|x; \phi) &= \log \int_{\pi} P(y, \pi|x; \phi) d\pi \\ &= \log \int_{\pi} P(y, \pi|x; \phi) \frac{P(\pi|x, y; \theta)}{P(\pi|x, y; \theta)} d\pi \\ &= \log \int_{\pi} \delta(y - \mathcal{B}(\pi)) P(\pi|x; \phi) \frac{P(\pi|x, y; \theta)}{P(\pi|x, y; \theta)} d\pi \\ &= \log \int_{\pi \in \mathcal{B}^{-1}(y)} P(\pi|x, y; \theta) \frac{P(\pi|x; \phi)}{P(\pi|x, y; \theta)} d\pi \\ &\geq \int_{\pi \in \mathcal{B}^{-1}(y)} P(\pi|x, y; \theta) \log \frac{P(\pi|x; \phi)}{P(\pi|x, y; \theta)} d\pi \\ &= -D_{KL}(\underbrace{P(\pi|x, y; \theta)}_{\text{Oracle Teacher}} \parallel \underbrace{P(\pi|x; \phi)}_{\text{Student}}) \end{aligned} \quad (4)$$

where the inequality follows from Jensen's inequality, \mathcal{B} represents the mapping function in (3), and D_{KL} denotes the KL-divergence. In our framework, $P(\pi|x, y; \theta)$ and $P(\pi|x; \phi)$ correspond to the alignment probability derived from the Oracle Teacher and the student, respectively. By minimizing the

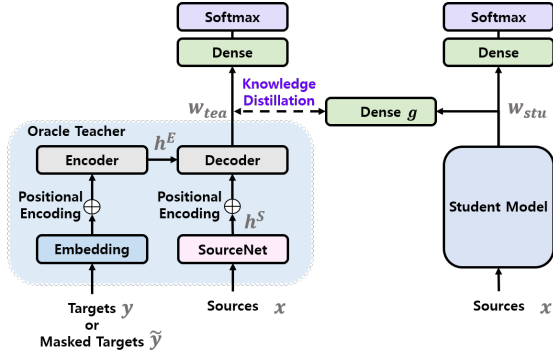


Fig. 3. KD procedure with the Oracle Teacher.

KL-divergence between the Oracle Teacher and the student, we can maximize the conditional probability of the student model $P(y|x; \phi)$.

Directly optimizing the KL-divergence in (4) is intractable because the KL divergence involves the integral that is difficult to calculate. To sidestep this problem, we can minimize the CE between the softmax outputs of the Oracle Teacher and the student. However, as reported in previous KD studies [20]–[22], simply applying the frame-level CE to the CTC-based model can worsen the performance compared to the baseline trained only with the ground truth. Instead, we adopt FitNets [29] as the basic KD technique, which considers the hidden representation for distillation, and there are two reasons for this choice: (1) In the CTC framework, transferring the hidden representation is much more effective than the softmax-level KD approach [25]; (2) Recent KD approaches for sequence learning [25], [37], [39] are based on the Fitnets.

As depicted in Fig. 3, w_{tea} and w_{stu} respectively denote the hidden representation obtained from the last layer of the teacher and student models. Since usually the hidden layer dimensions of w_{tea} and w_{stu} are different, we apply a fully connected layer g to bridge the dimension mismatch. The process of KD initializes the student by minimizing the distance between hidden representation of the teacher model w_{tea} and student model w_{stu} . The objective for KD is given by

$$\mathcal{L}_{KD}(w_{stu}, w_{tea}) = \|w_{tea} - g(w_{stu})\|_2^2. \quad (5)$$

C. Model Structure

The Oracle Teacher mainly consists of three components: 1) an encoder to encode the target input, 2) a SourceNet to extract the features from the source input data, and 3) a decoder to predict the CTC alignment. Its architecture follows the encoder-decoder structure of the Transformer [42], which allows the model to attend to related target information when making a prediction. Note that the Oracle Teacher is a non-autoregressive model where the look-ahead mask is not included in the decoder. In Fig. 1, we illustrate the schematics of the Oracle Teacher model which can also be summarized

as follows:

$$h^S = \text{SourceNet}(x; \theta_S), \quad (6)$$

$$h^E = \text{Encoder}(y; \theta_E), \quad (7)$$

$$P(\pi|x, y) \sim \text{Decoder}(h^E, h^S; \theta_D). \quad (8)$$

Compared to the vanilla Transformer, the main architectural difference lies in the cross attention. The encoder takes inputs from the whole target y , and its resulting vectors h^E are treated as key-value pairs of the cross attention. In addition, we do not employ a look-ahead mask, which is used in the vanilla Transformer to mask the future tokens, in the multi-head attention layer of the decoder.

1) *SourceNet*: The SourceNet converts the source input x into high-level representations h^S . Since h^S serves as query for the decoder, the length of the decoder output has the same length as h^S . The architecture of the SourceNet depends on the task we are interested in. When the main task is ASR, the SourceNet corresponds to an acoustic model part of the conventional ASR model. In our experiments, we apply CTC-based Jasper [40] architecture as the SourceNet, and consequently $|h^S| \geq |y|$. In the case of the STR task, we adopt the architecture of the CRNN [41] for the SourceNet.

2) *Encoder*: In the encoder, we adopt the same structure as the encoder of the original Transformer. The self-attention captures dependencies between different positions in y and outputs intermediate representations h^E .

3) *Decoder*: The representations h^S and h^E are fed into the decoder, which follows the architecture of the conventional Transformer decoder. The self-attention layer, the first attention layer of the decoder, takes the representations h^S as the input. Then, the output serves as queries for the cross attention, whose key-value pairs are the representations h^E . The cross attention allows the decoder to look into the relevant target information when producing the prediction. Note that the look-ahead mask is not included in the decoder. Different from the autoregressive model that only uses the past output tokens in producing the results, the Oracle Teacher can utilize more global output features when predicting the output.

In the proposed framework, the representation of the source x corresponds to the queries for the cross-attention. This is because, for KD, the length of the decoder output (= the length of the query h^S) should have the same length (T) as the student's output, which is determined by the source x . If the decoder output has a different length from that of the student's output, the Oracle Teacher cannot transfer the knowledge to the student.

IV. EXPERIMENTAL SETTINGS

A. Datasets and Baselines

1) *Speech Recognition*: For ASR, we evaluated the performance of the models on LibriSpeech [43]. In the training phase, “train-clean-100”, “train-clean-360”, and “train-other-500” were used. For evaluation, “dev-clean”, “dev-other”, “test-clean”, and “test-other” were applied. We adopted the current high-performing models for the conventional teacher in each task. In the case of ASR, we applied pre-trained Jasper Dense Residual (Jasper DR) [40], which

consists of 54 convolutional layers. Recent ASR studies [25], [44]–[46] utilized Jasper DR as the baseline. According to the previous study [40], Jasper DR produces word error rate (WER) 2.62 % with strong Transformer-XL [47] LM, still SOTA performance on LibriSpeech. As for the ASR student model, we used Jasper Mini, composed of 33 depthwise separable 1D convolutional layers. The Oracle Teacher had the SourceNet with 8 convolutional layers, and both the encoder and decoder consisted of 2 layers.

2) *Scene Text Recognition*: We evaluated STR models on seven benchmark datasets¹: Street View Text (SVT) [49], SVT Perspective (SVTP) [50], IIIT5K-Words (IIIT) [51], CUTE80 (CT) [52], ICDAR03 (IC03) [53], ICDAR13 (IC13) [54], and ICDAR15 (IC15) [55]. For validation, IC13, IC15, IIIT, and SVT were applied. As training datasets, we used the two most popular datasets: MJSynth [56] and SynthText [57]. We adopted Rosetta [58] and STAR-Net [59], considered as the benchmarking SOTA models in recent researches [60], [61]. In the case of the student model, CRNN [41] was adopted with a thin-plate spline (TPS), which normalizes curved and perspective texts into a standardized view. The SourceNet followed the TPS-CRNN structure, and both the encoder and decoder used 1 layer.

B. Implementation Details

1) *Speech Recognition*: For the LibriSpeech dataset, We used the OpenSeq2Seq [62] toolkit for the implementation. ASR models were based on the character-level CTC models. The character set had 29 labels. In the case of Jasper DR, we used the pre-trained model provided by the OpenSeq2Seq toolkit. The student model was run on three Titan V GPUs, each with 12GB of memory. We used a NovoGrad optimizer [63] whose initial learning rate started from 0.02 with a weight decay of 0.001. For KD, the student was initially trained with FitNets [29] loss for 5 epochs. After initialization, 50 epochs were spent on CTC training. In addition, we trained the Oracle Teacher for 30 epochs on a single Titan V GPU using Noam learning rate scheduler with 4000 steps of warmup and a learning rate of 1.5. When applying beam-search decoding with language model (LM), we used KenLM [64] for 4-gram LM, where the LM weight, the word insertion weight, and the beam width were experimentally set to 2.0, 1.5, and 512, respectively. For the Mandarin ASR dataset, the character set had a total of 5207 labels. Pre-trained Jasper DR, which was used as the conventional teacher, was provided by the NeMo [65] toolkit. The student was trained in an identical way to LibriSpeech, but the initial learning rate was set to 0.005. Instead of WER, we measured the character error rate (CER) since a single character often represents a word for Mandarin.

2) *Scene Text Recognition*: When training the STR models, our experiments were conducted using the official implementation provided by Baek *et al.*² [48]. STR models were based on the character-level CTC models. The character set had a total of 37 labels. All STR models, including the Oracle

Teacher, were trained for 300k iterations on a single Titan V GPU (12GB) in the CTC framework. We employed the AdaDelta optimizer [66] with a decay rate of 0.95, and the initial learning rate was 1.0. In FitNets [29] training, we trained 300k iterations for the student.

V. EXPERIMENTAL RESULTS

In the subsequent part of this paper, $A \rightarrow B$ means that teacher model A transfers knowledge to the student model B . As mentioned in Section III-B, we employed FitNets [29] as the basic KD technique.

A. Main Results: Performance Comparison

Since the Oracle Teacher is the teacher model for KD, not the baseline model performing the learning task, the evaluation results of the Oracle Teacher itself are not described in this section. Note that the model size and the performance of the Oracle Teacher will be additionally discussed in Section V-D and V-E3.

1) *Speech Recognition*: The results for LibriSpeech are shown in Table I. We measured WER to quantify the performance. The best performance was achieved when training the student with the Oracle Teacher. In addition, to further check the effectiveness of the target input y , which is used as the additional input of the Oracle Teacher, we applied an incomplete Oracle Teacher model, called Oracle Teacher w/o target. In Oracle Teacher w/o target, zero arrays were treated as the additional input instead of the target input y during training and inference phases. Since the Oracle Teacher w/o target only consumed the source input, its architecture was similar to that of the conventional CTC model. When applying the Oracle Teacher w/o target, the distilled student achieved improvement over the baseline student, which indicates that the knowledge of the SourceNet contributed to improving the performance of the student. However, their performances were worse than the Oracle Teacher \rightarrow Jasper Mini in all configurations, implying that the oracle guidance helped the Oracle Teacher extract a more supportive knowledge for the student.

As presented in Table II, we can confirm that the Oracle Teacher still works well with KD on the Mandarin dataset. Interestingly, when the Oracle Teacher was applied, the distilled student (CER: 9.74 % on dev-iOS) performed similarly to the pre-trained Jasper DR (CER: 9.69 % on dev-iOS), notwithstanding its smaller parameter size (14 M parameters) than Jasper DR (333 M parameters). In some cases, including test-iOS and test-Android, the student distilled from the Oracle teacher outperformed the Jasper DR teacher. When transferring the knowledge from the Oracle Teacher w/o target, the results show that Oracle Teacher w/o target \rightarrow Jasper Mini performed better than Jasper DR \rightarrow Jasper Mini. It indicates that, even without the additional target information, the student can benefit from the knowledge of the SourceNet. However, our best performance was achieved when applying the Oracle Teacher as the teacher model.

¹We applied the datasets used in the comparative study conducted by Baek *et al.* [48].

²<https://github.com/clovaai/deep-text-recognition-benchmark>

TABLE I
WER (%) PERFORMANCE COMPARISON ACROSS CTC-BASED ASR MODELS ON LIBRISPEECH. THE BEST RESULT OF THE STUDENT IS IN BOLD.

ASR baseline model	Params.	w/o LM				w/ LM			
		dev		test		dev		test	
		clean	other	clean	other	clean	other	clean	other
Jasper DR [40]	333 M	3.61	11.37	3.77	11.08	2.99	9.40	3.62	9.33
Jasper Mini	8 M	8.66	23.28	8.85	24.26	4.78	15.14	5.15	15.77
Student	Teacher	w/o LM				w/ LM			
		dev		test		dev		test	
		clean	other	clean	other	clean	other	clean	other
Jasper Mini	None	8.66	23.28	8.85	24.26	4.78	15.14	5.15	15.77
	Jasper DR [40]	7.05	19.41	7.03	20.41	4.80	14.32	5.00	14.99
	Oracle Teacher (ours)	6.64	18.91	6.67	19.82	4.65	14.31	4.90	14.65
	Oracle Teacher w/o target	7.22	20.39	7.32	21.10	4.72	14.67	4.91	15.15

TABLE II
CER (%) ON AISHELL-2 WHEN GREEDY DECODING WAS APPLIED. THE BEST RESULT OF THE STUDENT IS IN BOLD.

ASR baseline model	Params.	dev			test		
		iOS	Android	Mic	iOS	Android	Mic
Jasper DR [40]	338 M	9.69	11.48	12.23	9.37	10.84	11.84
Jasper Mini	14 M	11.77	14.23	15.03	11.38	12.71	14.27
Student	Teacher	dev			test		
		iOS	Android	Mic	iOS	Android	Mic
Jasper Mini	None	11.77	14.23	15.03	11.38	12.71	14.27
	Jasper DR [40]	10.70	12.78	13.66	10.12	11.31	12.60
	Oracle Teacher (ours)	9.74	11.49	12.31	9.27	10.36	11.99
	Oracle Teacher w/o target	10.45	12.42	13.13	9.76	10.92	12.19

TABLE III
PERFORMANCE OF CTC-BASED STR MODELS. THE BEST RESULT OF THE STUDENT IS IN BOLD.

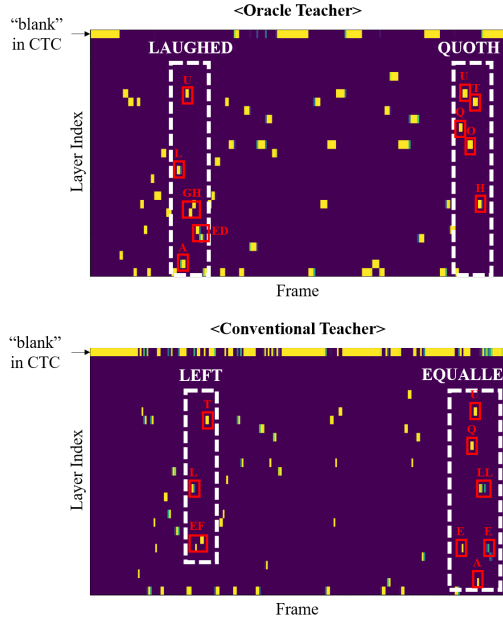
STR baseline model	Params.	IIIT 3000	SVT	IC03 860	IC13 857	IC13 1015	IC15 1811	IC15 2077	SVTP	CT	Total accuracy
Rosetta [58]	46 M	85.53	84.85	94.19	91.95	90.74	73.22	70.55	76.12	68.99	82.45
Star-Net [59]	49 M	85.50	85.47	93.84	92.77	91.92	72.50	69.77	73.80	70.38	82.24
CRNN [41]	10 M	83.87	80.37	93.02	90.43	89.46	70.07	67.53	72.09	65.51	80.10
Student	Teacher	IIIT 3000	SVT	IC03 860	IC13 857	IC13 1015	IC15 1811	IC15 2077	SVTP	CT	Total accuracy
CRNN [41]	None	83.87	80.37	93.02	90.43	89.46	70.07	67.53	72.09	65.51	80.10
	Rosetta [58]	84.70	83.46	92.91	91.02	90.15	71.89	69.20	71.16	65.85	81.04
	Star-Net [59]	85.20	84.39	93.49	91.60	90.74	72.45	69.77	72.25	70.04	81.77
	Oracle Teacher (ours)	85.77	84.54	93.61	91.48	90.54	73.11	70.40	74.26	70.38	82.21
	Oracle Teacher w/o target	85.40	82.84	93.02	90.78	89.75	71.73	69.04	72.71	68.99	81.30

2) *Scene Text Recognition*: For the STR task, we used accuracy, the success rate of word predictions per image, as a performance metric. As reported in Table III, the student distilled from the Oracle Teacher showed better performance than those distilled from other teachers, and its total accuracy (82.21 %) was almost similar to that of the conventional teacher Star-Net (82.24 %) while having much fewer parameters (10 M parameters). On IC13 datasets, the performances of Star-Net → CRNN were slightly better than those of Oracle Teacher → CRNN. However, the differences were negligible since Oracle Teacher → CRNN showed better improvements in most cases, including the total accuracy. Oracle Teacher w/o target → CRNN performed better than Rosetta → CRNN in some cases. It means that, even without using the additional target input, the student can benefit from the knowledge of the SourceNet. However, the distilled student from the Oracle Teacher w/o target had worse achievements than Star-Net → CRNN and Oracle Teacher → CRNN, indicating that the target input played an important role in the effectiveness of the Oracle Teacher.

B. Case Study: The Effect of Target Input

To validate why the proposed method could result in better KD performance than the conventional teacher, we conducted a case study for ASR on LibriSpeech test-other dataset. By comparing predictions between the conventional teacher and the Oracle Teacher, we verified the effect of using the target information and the behaviour of the Oracle Teacher.

In Fig. 4, we visualized the softmax prediction (CTC alignment) of the conventional teacher and the Oracle Teacher. The x-axis refers to acoustic frames, and the y-axis refers to the character labels. As displayed in Fig. 4, the conventional teacher converted a given speech into “but the king left him to scorn thou a sword equalle” and made erroneous predictions with “left” and “equalle”. When conditioning on the speech voice only, it is hard to distinguish “left”/“laughed” and “equalle”/“he quoth”. However, the Oracle Teacher gave accurate CTC alignment by utilizing the additional target (text) information, implying that a more optimal problem-solving could be derived by referring to both source (speech) and target (text) information.



Target (y)	but the king laughed him to scorn thou a sword he quoth
Conventional Teacher	but the king left him to scorn thou a sword equalle
Oracle Teacher	but the king laughed him to scorn thou a sword he quoth

Fig. 4. Frame-wise label probability example on LibriSpeech test-clean dataset. Conventional teacher denotes the Jasper DR model. The x-axis refers to acoustic frames, and the y-axis refers to the character labels. The last label index represents the “blank” label in the CTC framework.

Target	you owe me some bills gov'nor	no wait another half hour
Conventional Teacher (w/ greedy decoding)	yuowe me some bills gov'nor	no aight another a half hour
Conventional Teacher (w/ KenLM)	you owe me some bills gov'nor	no wait another half hour
Oracle Teacher (w/ greedy decoding)	you owe me some bills gov'nor	no wait another half hour

Fig. 5. Recognition example on LibriSpeech test-clean dataset.

Interestingly, as shown in Fig. 4, most frames of the conventional teacher had the highest probability for the “blank” token. In contrast, the Oracle Teacher had fewer frames that were predicted as “blank” token. This is because the target input y was used as guidance to reduce the candidate scope of the CTC alignment π . By using the additional target (text) information, multiple frames of the Oracle Teacher were more likely to be predicted as non-blank character labels rather than the “blank” during the active speech periods, implying that the representation of the Oracle Teacher could contain much more information about non-blank labels corresponding to the characters (text).

In addition, we compared the ASR predictions of the conventional teacher and the Oracle Teacher, as shown in Fig. 5. In Fig. 5, the conventional teacher made erroneous predictions with “yuowe”, “aight”, and “a” using the greedy decoding. When considering the acoustic (speech) feature only, it is challenging to distinguish some words, such as “you owe”/“yuowe” and “wait”/“aight”. The conventional teacher generated an accurate prediction when decoding with the external KenLM that provided additional linguistic information.

TABLE IV
WER (%) PERFORMANCE COMPARISON ACROSS CTC-BASED ASR MODELS ON LIBRISPEECH. THE BEST RESULT OF THE STUDENT IS IN BOLD.

Teacher	Student	KD method	dev		test	
			clean	other	clean	other
None	Jasper Mini	None	8.66	23.28	8.85	24.26
Jasper DR		RKD [25]	6.74	19.27	6.77	19.78
Oracle Teacher			6.44	18.36	6.43	18.97
Jasper DR		SKD [25]	7.64	21.36	7.81	22.41
Oracle Teacher			7.57	21.20	7.71	21.71
Jasper DR		FitNets [29]	7.05	19.41	7.03	20.41
Oracle Teacher			6.64	18.91	6.67	19.82

However, the proposed Oracle Teacher could produce correct ASR prediction without using the external LM. This is because the Oracle Teacher leveraged both the source input (speech) and the output label (text) as the teacher model’s input. Unlike the conventional teacher that only considered acoustic features, the Oracle Teacher performed a fusion of acoustic (speech) and linguistic (text) features when generating the prediction. Since unifying acoustic and linguistic representation learning generally enhances the performance of the speech processing [7]–[11], the Oracle Teacher that considered linguistic information could estimate better CTC prediction, and also its representation was a more supportive knowledge for the ASR student.

C. Performance Comparison with Other KD Methods

In the previous results, we applied FitNets [29] as the basic KD method. To further validate the effectiveness of the Oracle Teacher, we used other KD methods for performance comparison.

Firstly, we applied RKD [25] as the KD method, a recent KD approach in ASR task. It transfers the representation-level knowledge by considering a frame weighting, reflecting which frames were important for KD. In addition to the RKD, we adopted SKD [25] for KD, which effectively transfers the softmax-level knowledge in the CTC framework. From the results in Table IV, it is confirmed that Oracle Teacher was more supportive than the conventional Jasper DR teacher in all configurations. Also, we verified that RKD achieved better improvements than other KD methods, including FitNets and SKD. The best performance was achieved when using the Oracle Teacher with the RKD. We can observe the consistent performance gain of the Oracle Teacher over the conventional teacher for various KD methods.

D. Computational Cost Comparison

In addition to the previous experiments, we proceeded to verify the computational efficiency of the proposed teacher model. Computational resource consumption compared to the conventional teacher models are shown in Table V.

1) *Speech Recognition*: Since it is difficult to reproduce the reported WER results of Jasper DR [40] without a large number of resources, we used the checkpoint for LibriSpeech, provided by the OpenSeq2Seq [62] toolkit. For LibriSpeech, the pre-trained Jasper DR model required eight 32GB GPUs

TABLE V
COMPUTATIONAL RESOURCE CONSUMPTION COMPARISON ACROSS TEACHER MODELS.

Task	Training dataset	Teacher model	Params.	GPU	Batch	Times	Epochs
ASR	LibriSpeech	Jasper DR [40]	333 M	8 * 32GB	256	-	400 epochs
		Oracle Teacher (ours)	33 M	1 * 12GB	64	22 h	30 epochs
ASR	AISHELL-2	Jasper DR [40]	338 M	8 * 32GB	128	-	50 epochs
		Oracle Teacher (ours)	34 M	1 * 12GB	64	118 h	30 epochs
STR	MJSynth + SynthText	Star-Net [59]	49 M	1 * 12GB	192	27 h	300k iter.
		Rosetta [58]	46 M	1 * 12GB	192	27 h	300k iter.
		Oracle Teacher (ours)	12 M	1 * 12GB	192	10 h	300k iter.

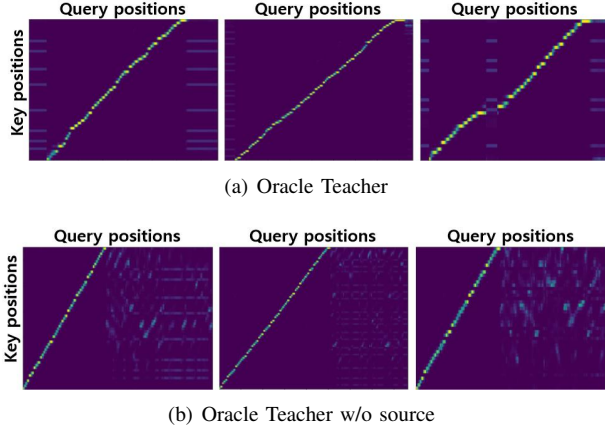


Fig. 6. Visualization of the attention weights of the Oracle Teacher with and without the source input

for 400 epochs with a batch size of 256. Its training time had not been reported previously, either in the paper of Li *et al.* [40] or the toolkit. In the case of the Oracle Teacher, we trained the model for 30 epochs on a single 12GB GPU, which took about 22 hours (≈ 1 day) to finish the training. Considering that the reported training of the Quartznet [67], which is more computationally efficient than Jasper DR, for 400 epochs took 122 hours (≈ 5 days) with eight 32GB GPUs with a batch size of 256, the Oracle Teacher dramatically reduced the computational cost of the teacher model.

2) *Scene Text Recognition*: As presented in Table V, the training of Star-Net [59] took about 27 hours on a single Titan V GPU (12GB) with a total batch size of 192, and the training of Rosetta [58] required about 27 hours. Compared to the two conventional models, the training of Oracle Teacher consumed much less training time (10 hours) with the same computational resource.

E. Analysis

From the previous experimental results, we validate the superiority of the proposed teacher model. Therefore, it is necessary to test if the model has been correctly trained.

1) *Visualization of Cross Attention*: We trained another incomplete Oracle Teacher, called Oracle Teacher w/o source. The zero arrays, which had the same size of x , were treated as the input instead of the source input x during the training. Then, the Oracle Teacher w/o source only considered the target input y when making a prediction, similar to the aforementioned trivial solution. In Fig. 6, we visualize the cross attention scores of the decoder for the ASR task, where

TABLE VI
PERFORMANCE AND PARAMETER COMPARISON BETWEEN THE ORACLE TEACHER AND THE CONVENTIONAL TEACHER.

Task	Model	Param.	WER (%)			
			dev		test	
			clean	other	clean	other
ASR	Jasper DR [40]	333 M	3.61	11.37	3.77	11.08
	Oracle Teacher	33 M	2.87	3.10	4.03	3.29

the x-axis refers to acoustic frames and the y-axis refers to characters. For the Oracle Teacher w/o source, the attention had almost diagonal alignment along with the key position (text) while ignoring the length of query, as shown in Fig. 6(b). In contrast, the Oracle Teacher considered both speech and text alignment in the cross attention, and the attention scores were correctly computed along with the acoustic frames (query), as shown in Fig. 6(a). It means that the Oracle Teacher utilized the source x for model training while preventing the trivial solution. Therefore, we can confirm that the Oracle Teacher, including SourceNet, has been correctly trained.

2) *KD with Oracle Teacher w/o source*: In addition to the previous experiments, we proceeded to train the student model with the knowledge of the Oracle Teacher w/o source. However, as we expected, the distilled student failed to converge. From this additional result, we can verify that the source input x is the necessary factor of the Oracle Teacher, and the proposed Oracle Teacher has been correctly trained.

3) *Performance of Oracle Teacher*: We also evaluated the performance of the Oracle Teacher itself compared to the conventional teachers, as shown in Table VI. If the Oracle Teacher copies the target input y without utilizing the source input x , the performance of the Oracle Teacher should be perfect. We measured WER (%) results on LibriSpeech. The results show that the performance of the Oracle Teacher was more effective than that of the conventional teacher model, which seemed reasonable because the Oracle Teacher was trained with the guidance of y . Meanwhile, the predictions of the Oracle Teacher were not the same as each ground truth. This implies that the Oracle teacher's output did not simply copy the target input y , and the information from a properly-trained SourceNet was utilized to generate the prediction. Compared to the "clean" datasets, the difference of WER was huge in "other" sets. Since the "other" dataset represents a noisy dataset, the conventional ASR model (Jasper DR) showed low performance for the "other" dataset. However, the Oracle Teacher could result in high performance for the noisy dataset since it used text information from the target. This indicates that the Oracle Teacher could be a noisy robust

teacher with small parameters. From the results, we can verify that the Oracle Teacher provided more accurate and better guidance to the student than the conventional ASR teacher model.

VI. DISCUSSION

This work offered a powerful but efficient KD method in the context of CTC-based sequence learning tasks. However, the use of output labels as the additional input for the teacher model could limit its usage on unsupervised data. One possible way to apply the Oracle Teacher in the unsupervised setting is leveraging the prediction as the model's input, like [68]–[70]. However, in Section V-B, it is confirmed that the Oracle Teacher gave accurate CTC alignment by utilizing the ground-truth labels as the additional input. Since the target input was used as guidance to reduce the candidate scope of the alignment, using the erroneous prediction as the additional input might significantly degrade the performance of the Oracle Teacher. Extending the proposed framework further, using the Oracle Teacher on unsupervised data is our future work.

VII. CONCLUSIONS

We introduced a novel teacher for CTC-based sequence models, namely Oracle Teacher, that leverages the output labels as the additional input to the model. Through a number of experiments, we confirmed that the student distilled from the Oracle Teacher performed better compared to the one distilled from the conventional teacher. Furthermore, our framework significantly reduced the computational cost of the teacher model in terms of the training time and required GPU resources. As the effective teacher can be trained with a reduced computational cost, the Oracle Teacher can be a new breakthrough in KD. We expect the application of the Oracle Teacher in various tasks, such as regression, ranking, etc., in the future.

REFERENCES

- [1] S. Han, H. Mao, and W. J. Dally. Deep compression: compressing deep neural network with pruning, trained quantization and huffman coding. In *Proc. ICLR*, 2016.
- [2] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf. Pruning filters for efficient convnets. In *Proc. ICLR*, 2017.
- [3] J. Wu, L. Cong, Y. Wang, Q. Hu, and J. Cheng. Quantized convolutional neural networks for mobile devices. In *Proc. CVPR*, page 4820–4828, 2016.
- [4] C. Bucila, R. Caruana, and A. Niculescu-Mizil. Model compression. In *Proc. ACM SIGKDD*, page 535–541, 2006.
- [5] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *Proc. NIPS Workshop Deep Learn.*, 2014.
- [6] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proc. ICML*, pages 369–376, 2006.
- [7] G. Zheng et al. Wav-bert: cooperative acoustic and linguistic representation learning for low-resource speech recognition. In *Proc. EMNLP*, 2021.
- [8] B. Sharma et al. Leveraging acoustic and linguistic embeddings from pretrained speech and language models for intent classification. In *Proc. ICASSP*, 2021.
- [9] C. Yi, S. Zhou, and B. Xu. Efficiently fusing pretrained acoustic and linguistic encoders for low-resource speech recognition. *IEEE Signal Processing Letters*, 2021.
- [10] F. Yu and K. Chen. Non-autoregressive transformer-based end-to-end asr using bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:1474–1482, 2022.
- [11] G. Winata et al. Adapt-and-adjust: overcoming the long-tail problem of multilingual speech recognition. *arXiv preprint arXiv:2012.01687*, 2020.
- [12] J. Ba and R. Caruana. Do deep nets really need to be deep? In *Proc. NIPS*, pages 2654–2662, 2014.
- [13] J. Li, R. Zhao, T. J. Huang, and Y. Gong. Learning small-size dnn with output-distribution-based criteria. In *Proc. INTERSPEECH*, 2014.
- [14] J. H. M. Wong and M. J. F. Gales. Sequence student-teacher training of deep neural networks. In *Proc. INTERSPEECH*, pages 2761–2765, 2016.
- [15] K. J. Geras et al. Blending lstms into cnns. In *Proc. ICLR Workshop*, 2016.
- [16] Y. Chebotar and A. Waters. Distilling knowledge from ensembles of neural networks for speech recognition. In *Proc. INTERSPEECH*, pages 3439–3443, 2016.
- [17] S. Watanabe, T. Hori, J. L. Roux, and J. R. Hershey. Student-teacher network learning with enhanced features. In *Proc. ICASSP*, pages 5275–5279, 2017.
- [18] L. Lu, M. Guo, and S. Renals. Knowledge distillation for small-footprint highway networks. In *Proc. ICASSP*, pages 4820–4824, 2017.
- [19] T. Fukuda, M. Suzuki, G. Kurata, S. Thomas, J. Cui, and B. Ramabhadran. Efficient knowledge distillation from an ensemble of teachers. In *Proc. INTERSPEECH*, pages 3697–3701, 2017.
- [20] A. Senior, H. Sak, F. C. Quirey, T. Sainath, K. Rao, et al. Acoustic modelling with cd-ctc-smbr lstm rnns. In *Proc. ASRU*, pages 604–609, 2015.
- [21] R. Takashima, S. Li, and H. Kawai. An investigation of a knowledge distillation method for ctc acoustic models. In *Proc. ICASSP*, pages 5809–5813, 2018.
- [22] R. Takashima, S. Li, and H. Kawai. Investigation of sequence-level knowledge distillation methods for ctc acoustic models. In *Proc. ICASSP*, pages 6156–6160, 2019.
- [23] 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*, pages 411–417, 2018.
- [24] G. Kurata and K. Audhkhasi. Guiding ctc posterior spike timings for improved posterior fusion and knowledge distillation. In *Proc. INTERSPEECH*, pages 1616–1620, 2019.
- [25] J. W. Yoon, H. Lee, H. Y. Kim, W. I. Cho, and N. S. Kim. Tutonet: towards flexible knowledge distillation for end-to-end speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:1626–1638, 2021.
- [26] R. Pang, T. Sainath, R. Prabhavalkar, S. Gupta, Y. Wu, S. Zhang, and C. C. Chiu. Compression of end-to-end models. In *Proc. INTERSPEECH*, 2018.
- [27] H. G. Kim, H. Na, H. Lee, J. Lee, T. G. Kang, M. J. Lee, and Y. S. Choi. Knowledge distillation using output errors for self-attention end-to-end models. In *Proc. ICASSP*, 2019.
- [28] K. Kwon, H. Na, H. Lee, and N. S. Kim. Adaptive knowledge distillation based on entropy. In *Proc. ICASSP*, 2020.
- [29] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: hints for thin deep nets. In *Proc. ICLR*, 2015.
- [30] S. Zagoruyko and N. Komodakis. Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer. In *Proc. ICLR*, 2017.
- [31] J. Yim, D. Joo, J. Bae, and J. Kim. A gift from knowledge distillation: fast optimization, network minimization and transfer learning. In *Proc. CVPR*, 2017.
- [32] S. Srinivas and F. Fleuret. Knowledge transfer with jacobian matching. In *Proc. ICML*, 2018.
- [33] J. Kim, S. Park, and N. Kwak. Paraphrasing complex network: network compression via factor transfer. In *Proc. NIPS*, page 2760–2769, 2018.
- [34] B. Heo, M. Lee, S. Yun, and J. Y. Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *Proc. AAAI*, page 3779–3787, 2019.
- [35] W. Park, D. Kim, Yan. Lu, and M. Cho. Relational knowledge distillation. In *Proc. CVPR*, pages 3967–3976, 2019.
- [36] M. Ji, B. Heo, and S. Park. Show, attend and distill: knowledge distillation via attention-based feature matching. In *Proc. AAAI*, pages 7945–7952, 2021.
- [37] H. Chang, S. Yang, and H. Lee. Distilhubert: speech representation learning by layer-wise distillation of hidden-unit bert. In *Proc. ICASSP*, 2022.

- [38] R. Wang and others. Lighthubert: lightweight and configurable speech representation learning with once-for-all hidden-unit bert. In *Proc. INTERSPEECH*, 2022.
- [39] Y. Lee et al. Fithubert: going thinner and deeper for knowledge distillation of speech self-supervised learning. In *Proc. INTERSPEECH*, 2022.
- [40] 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. In *Proc. INTERSPEECH*, 2019.
- [41] B. Shi et al. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:2298–2304, 2017.
- [42] A. Vaswani et al. Attention is all you need. In *Proc. NIPS*, pages 5998–6008, 2017.
- [43] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *Proc. ICASSP*, pages 5206–5210, 2015.
- [44] S. Kim et al. Integer-only zero-shot quantization for efficient speech recognition. In *Proc. ICASSP*, 2022.
- [45] O. Hrinchuk, M. Popova, and B. Ginsburg. Correction of automatic speech recognition with transformer sequence-to-sequence model. In *Proc. ICASSP*, 2020.
- [46] N. Zhang, J. Wang, W. Wei, x. Qu, N. Cheng, and J. Xiao. Cactnet: cube attentional cnn for automatic speech recognition. In *Proc. IJCNN*, 2021.
- [47] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov. Transformer-xl: attentive language models beyond a fixed-length context. In *Proc. ACL*, pages 2978–2988, 2019.
- [48] J. Baek, G. Kim, j. Lee, S. Park, D. Han, S. Yun, S. J. Oh, and H. Lee. What is wrong with scene text recognition model comparisons? In *Proc. ICCV*, 2019.
- [49] K. Wang, B. Babenko, and S. Belongie. End-to-end scenetext recognition. In *Proc. ICCV*, page 1457–1464, 2011.
- [50] T. Q. Phan et al. Recognizing text with perspective distortion in natural scenes. In *Proc. ICCV*, page 569–576, 2013.
- [51] A. Mishra, K. Alahari, and C. Jawahar. Scene text recognition using higher order language priors. In *Proc. BMVC*, 2012.
- [52] A. Risnumawan et al. A robust arbitrary text detection system for natural scene images. *Expert Systems with Applications*, 41:8027–8048, 2014.
- [53] S. M. Lucas et al. Icdar 2003 robust reading competitions. In *Proc. ICDAR*, page 682–687, 2003.
- [54] D. Karatzas et al. Icdar 2013 robust reading competition. In *Proc. ICDAR*, page 1484–1493, 2013.
- [55] D. Karatzas et al. Icdar 2015 competition on robust reading. In *Proc. ICDAR*, page 1156–1160, 2015.
- [56] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. In *Proc. NIPS*, 2014.
- [57] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In *Proc. CVPR*, 2016.
- [58] F. Borisjuk et al. Rosetta: large scale system for text detection and recognition in images. In *Proc. ACM SIGKDD*, page 71–79, 2018.
- [59] W. Liu, C. Chen, K. K. Wong, and Z. Su. Star-net: A spatial attention residue network for scene text recognition. In *Proc. BMVC*, 2016.
- [60] A. Singh, G. Pang, M. Toh, J. Huang, W. Galuba, and T. Hassner. Textocr: towards large-scale end-to-end reasoning for arbitrary-shaped scene text. In *Proc. CVPR*, page 8802–8812, 2021.
- [61] X. Xu, J. Chen, J. Xiao, L. Gao, F. Shen, and H. T. Shen. What machines see is not what they get: fooling scene text recognition models with adversarial text images. In *Proc. CVPR*, pages 12304–12314, 2020.
- [62] 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.
- [63] 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.
- [64] K. Heafield. Kenlm: faster and smaller language model queries. In *Proc. EMNLP*, 2011.
- [65] O. Kuchaiev et al. Nemo: a toolkit for building ai applications using neural modules. *arXiv preprint arXiv:1909.09577*, 2019.
- [66] M. D. Zeiler. Adadelata: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701v1*, 2012.
- [67] 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.
- [68] Y. Xia et al. Deliberation networks: sequence generation beyond one-pass decoding. 2017.
- [69] W. Chan, C. Saharia, G. Hinton, M. Norouzi, and N. Jaitly. Imputer: sequence modelling via imputation and dynamic programming. In *Proc. ICML*, 2020.
- [70] Y. Higuchi, S. Watanabe, N. Chen, T. Ogawa, and T. Kobayashi. Mask ctc: non-autoregressive end-to-end asr with ctc and mask predict. In *Proc. INTERSPEECH*, 2020.