



CoDERT: Distilling Encoder Representations with Co-learning for Transducer-based Speech Recognition

Rupak Vignesh Swaminathan, Brian King, Grant P. Strimel, Jasha Droppo, Athanasios Mouchtaris

Alexa Machine Learning, Amazon.com, USA

{swarupak,bbking,gsstrime,drojasha,mouchta}@amazon.com

Abstract

We propose a simple yet effective method to compress an RNN-Transducer (RNN-T) through the well-known knowledge distillation paradigm. We show that the transducer’s encoder outputs naturally have a high entropy and contain rich information about acoustically similar word-piece confusions. This rich information is suppressed when combined with the lower entropy decoder outputs to produce the joint network logits. Consequently, we introduce an auxiliary loss to distill the encoder logits from a teacher transducer’s encoder, and explore training strategies where this encoder distillation works effectively. We find that tandem training of teacher and student encoders with an in-place encoder distillation outperforms the use of a pre-trained and static teacher transducer. We also report an interesting phenomenon we refer to as implicit distillation, that occurs when the teacher and student encoders share the same decoder. Our experiments show 5.37-8.4% relative word error rate reductions (WERR) on in-house test sets, and 5.05-6.18% relative WERRs on LibriSpeech test sets.

Index Terms: knowledge distillation, RNN-Transducer, speech recognition, on-device machine learning.

1. Introduction

As neural network architectures evolve and become more complex, there is a growing need to reduce their footprint, not only for faster runtime and reducing server-side costs but also for on-device deployment. Following the general trend of on-device machine learning models, streaming automatic speech recognition (ASR) systems on edge devices have become popular in recent years [1, 2]. This has contributed to a paradigm shift in ASR systems, moving away from the traditional factored acoustic and language models to fully end-to-end (E2E) neural-based recognizers, thereby saving significantly on memory and latency [3, 4]. Recent studies have shown that such E2E models can even surpass the performance of conventional server-sized, factored models while being significantly smaller in size [5, 6].

In order to facilitate streaming speech recognition on-device, which enables better latency, privacy and reliability, compression and quantization techniques [7, 8, 9] have been developed to reduce model sizes and to accelerate inference. Low-rank matrix factorization has been applied successfully [7, 8] on acoustic models compressing them by a factor of at least 3 times. Pruning methods [10, 11, 12] exploiting sparsity in model weights have also been shown to be promising for ASR footprint reduction. Light recurrent replacements for Long Short-term Memory (LSTM) [13] cells such as Gated Recurrent Units [14], Coupled Input-Forget Gate [15] LSTMs, and Simple Recurrent Units [16] have shown negligible degradation in the performance for E2E ASR as well [1].

The Knowledge Distillation (KD) paradigm [17], commonly referred to as teacher-student training, has become one

of the sought-after techniques for model compression thanks to its architecture-agnostic nature and the ability to take advantage of unsupervised data [18]. It has been proven to be complementary to other compression approaches [19], yielding additional gains. In this work, we focus on distilling the RNN-T [20] encoder into a smaller, compact encoder. The RNN-T encoder is the most cumbersome part of network, and usually is several folds larger than the decoder. Furthermore, the encoder network processes audio frames that are much longer compared to the label sequence processed by the decoder. Motivated by this, we present CoDERT, a methodology for distilling RNN-T encoder representations, which has not been previously explored. Our contributions in this work are as follows: (1) we show empirically that the RNN-T encoder’s high-level representation has rich information about acoustically similar word-pieces, (2) we introduce distillation as an auxiliary task for the RNN-T encoder, distilling from a larger teacher RNN-T encoder, and (3) we demonstrate through experiments that for the encoder distillation to work effectively, the student and the teacher RNN-Ts need to be *co-learned* (trained in tandem) while sharing a common decoder.

2. Related Work

The importance of KD in ASR model compression has been well established for both hybrid acoustic models and E2E models [21, 22, 23]. In [24], the authors performed KD from a bi-directional teacher RNN-T into a uni-directional student RNN-T by computing KL-divergence between the outputs of the joint networks of the teacher and student RNN-Ts. This technique can be considered as lattice distillation since the distillation takes place at the end of the joint network’s output lattice. The approach does not scale to large batch sizes and output vocabulary sizes, since the distillation loss computation and memory usage become prohibitively expensive [25]. The authors of [25] propose a work-around approximation by conglomerating the probabilities of output tokens apart from the blank symbol (ϕ) and ground-truth symbol (y_{u+1}) into a “garbage bin” and then applying KL-divergence to compute the distillation loss. In [26], the authors propose an auxiliary task for the RNN-T encoder to predict context-dependent graphemic units used in the acoustic model of a traditional hybrid ASR system. This can be viewed as acoustic distillation, which we explore in this work, as opposed to the lattice distillation discussed earlier [24, 25].

Tandem training of teacher and student models are gaining traction recently [27, 28, 29]. In [29], a novel RNN-T architecture, which has an encoder working in dual-mode (full-context and streaming) with unique weight sharing strategies, uses [25] as an *inplace* distillation [28] method to distill knowledge from the full-context mode onto the streaming mode. Motivated by [29], we jointly train the student and teacher encoders while sharing a common decoder, which we refer to as *co-learning* in

this work. However, compared to [29], our approach focuses only on streaming encoders for both teacher and student models, and more importantly, our distillation loss remains distinct.

3. Technical Approach

In this section, we review the RNN-T architecture and provide an explanation of our rationale motivating our focus on the RNN-T encoder. We then present the formulation and elucidate our encoder distillation approach in detail, and finally discuss the benefits of the proposed approach.

3.1. RNN-Transducer

The RNN-T architecture was introduced in [20] as an extension to the connectionist temporal classification (CTC) framework [30] to overcome the conditional independence assumption of CTC. RNN-T consists of an encoder \mathcal{F} , which takes audio frame x_t as input and produces a high-level representation h_t^{enc} ; a prediction network \mathcal{G} (also referred to as the decoder), which outputs a high-level representation h_u^{dec} given the previous non-blank symbol y_u^{prev} as input; and a joint network \mathcal{J} which combines the encoder and decoder representations to produce the token logits. The posterior probability $P_{t,u}^k$ for the output token k at the lattice location (t, u) , is computed by applying softmax on the joint network output $h_{t,u}^k$.

The ground truth label sequence is used to train the decoder (teacher forcing) in the training phase, and the previous non-blank output of the transducer is used as the input to decoder during inference. The joint network strategy in our work is a simple addition followed by a tanh non-linearity. The RNN-T optimization is then formulated as maximizing the sum of all possible alignments $P(y^*|x)$, or equivalently minimizing the negative log-likelihood of $P(y^*|x)$, which is efficiently computed using a forward-backward algorithm as shown in [20].

3.2. Motivation: Why focus on the encoder?

First, the RNN-T encoder is the largest and most complex sub-network of the model. Unlike the decoder which performs a simple next step prediction task, the encoder’s objective is to model speech signals that are highly dynamic with acoustic variability coming from multiple speakers, noises and reverberant conditions. In order to model such variability, the encoder transformation $\mathcal{F}(x)$ requires an extremely high modeling capacity. However, due to the constraints involved in on-device deployment, one typically is required to trade-off accuracy for memory and latency. During inference, the decoder outputs can be cached and the search tree nodes which are expanded can be grouped together for batched computation on the decoder. Meanwhile the encoder, which is orders of magnitude larger must process the audio sequence frame-by-frame which is typically much longer than the label sequence for decoder. The encoder’s latency can be improved by techniques such as time reduction [31] or bifocal networks [32] without affecting accuracy, but naively reducing the number of layers and/or hidden units to downsize the encoder can severely impair its modeling capacity [1].

Next, we show a few examples of the rich representations learnt by the RNN-T encoder. Since our joint network strategy is an add operation followed by tanh non-linearity, the encoder and decoder output projections are the same dimensions as the number of output tokens. This property allows us to visualize the higher-level representations at the output of encoder in a meaningful fashion; computing $\text{softmax}(h_t^{enc})$ will produce

a probability distribution which can be directly attributed to the output tokens, and an *argmax* of the result can be considered the output token at time t . This is presented in Table 1, showcasing the ground truth tokens (word-pieces) and their related acoustically similar word-pieces found in the higher-level representations of the encoder of a fully-trained RNN-T model.

Table 1: Randomly picked ground truth word pieces corresponding to the words "Alexis", "flurry", and "can you" and their acoustically similar labels present in the encoder output representation. Underscore indicates beginning of a word.

Ground-truth word-pieces	Acoustically similar encoder representation
._alexis	._alexa
._flu, r, r, y	._fairy
._can, _you	._commute

Last, we visualize the entropy of the output of each of the components in the RNN-T, and demonstrate that the encoder naturally has a much higher entropy while the decoder’s output are highly predictable. Figure 1 shows the density of entropy at the output of joint, decoder, and encoder components, computed over a randomly chosen training batch. Based on the joint network entropy, one concludes that when encoder outputs are combined with that of the decoder, much of the valuable information about the relative probabilities between the output tokens of the encoder is lost. The relative distance of how close or far two sounds are to each other, and a high output entropy, are important characteristics for an effective distillation [17]; the RNN-T encoder output naturally contains such information as shown in Table 1 and Figure 1, respectively.

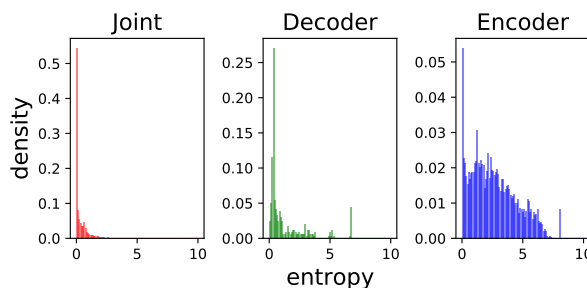


Figure 1: Entropy density of softmax outputs corresponding to the joint (red), decoder (green), and encoder (blue) components of a fully-trained RNN-T with 4K word-pieces.

3.3. Formulation: Distilling the high-level representations

Our encoder distillation formulation is simple yet powerful and easily reproducible. During training, we introduce an auxiliary task for the student encoder to match the logits of the teacher encoder’s logits. We experimented with different loss functions (L_2 , softmax cross-entropy and cosine-loss) and found L_2 loss to yield better results.

$$\mathcal{L}_{distill} = \|h_{1:t}^{enc}(S) - h_{1:t}^{enc}(T)\|^2 \quad (1)$$

where (S) and (T) denote components belonging to student and teacher, respectively.

As discussed in Section 5 in detail, we find that in order for the distillation to work effectively, the student and teacher encoders need to share the same decoder (see Figure 2). Unlike traditional distillation, where the teacher is pre-trained and its parameters are frozen, we update the teacher parameters and perform the distillation *inplace* [28, 29]. The final loss is expressed as,

$$\mathcal{L} = \mathcal{L}_{rnt}(S) + \mathcal{L}_{rnt}(T) + \lambda \mathcal{L}_{distill} \quad (2)$$

where λ is the distillation weight, chosen through hyper-parameter optimization on a development set.

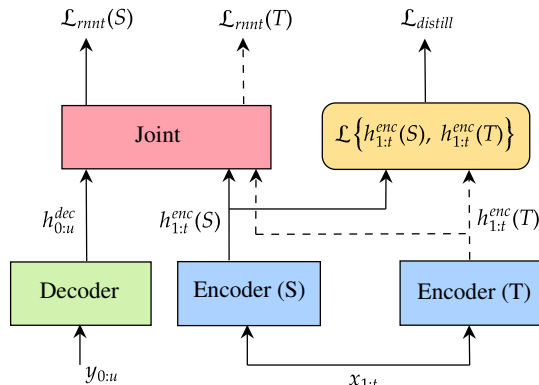


Figure 2: CoDERT: Proposed method for distilling encoder representations. The student (S) and teacher (T) encoders share a common decoder, and are updated through the sum of the RNN-T losses (co-learned). Additionally, the student encoder is updated through the distillation loss presented in Equation 1.

3.4. Benefits: An information-preserving distillation

The output of the joint network is a 4-dimensional tensor of shape $B \times T \times U \times V$, where B , T , U , and V correspond to batch size, maximum length of audio frames, maximum length of labels, and the vocabulary size, respectively. The traditional KL-divergence loss between teacher and student RNN-T joint outputs becomes infeasible for production quality training choices of B , T , U , and V , since the loss computation and memory bandwidth to maintain such large multi-dimensional tensors become highly impractical on today’s machines. An attempted work-around proposed in [25] is to consider only three probabilities $P_{t,u}^{y+1}$, $P_{t,u}^{\phi}$ and $1 - P_{t,u}^{y+1} - P_{t,u}^{\phi}$ (shortened as $P_{t,u}^{rest}$), which are then used in the KL-divergence based distillation loss computation. However, in bucketing the remaining probabilities to compute $P_{t,u}^{rest}$, information about the relative probabilities at the output of joint network is lost. Furthermore, the relative probability of encoder output tokens is not preserved at the joint output as shown in Section 3.2. In contrast, the proposed method leverages our findings that most of the information is in the encoder, and distills over the entire $B \times T \times V$ space, thereby circumventing the scaling barrier without having to compromise on the relative probabilities. Finally, our method does not preclude one from additionally applying the lattice distillation proposed in [25], so it can be considered a complementary approach.

4. Experiments

In this section, we discuss briefly about the experimental setup for training and evaluating our encoder distillation approach.

4.1. Model architecture

Our teacher encoder consists of 5 LSTM layers with 1024 hidden units each, followed by a fully-connected output projection layer of 4001 dimensions. The student encoder consists of 4 LSTM layers with 640 units each making it 35% of the teacher encoder’s size. The encoders use Log-Mel filterbank energies (LFBE) of 64-dimensions, with a window size of 25ms and hop size of 10ms and downsampled by 3 to have a frame rate of 30ms. The encoders also have a time reduction component [31] after the second layer which reduces the number of audio frames by a factor of 2. The decoder size is kept constant for both student and teacher models to ensure comparable analysis in our experiments. The decoder network consists of an input embedding of 512-dimensions, a 2-layer LSTM with 1024 hidden units and an output vocab projection of 4001-dimensions. The joint network consists of an addition operation on the encoder and decoder outputs followed by a tanh non-linearity. The output tokens modeled by the transducer are 4001 word pieces which include the blank symbol ϕ .

4.2. In-house data experimental setup

We use a 10K-hour in-house collection of de-identified Alexa English data to investigate our distillation method. We use 60 hours of audio as the development set for hyper-parameter optimization and selecting final model checkpoints, and a 60-hour test set to evaluate our models. Additionally, we also use a long-form ASR utterances test set (*long*) and a tail distribution low-frequency utterances test set (*tail*) in our evaluation. Our learning rate schedule policy has a warm up stage where the learning rate linearly increases from $1e-7$ to $5e-4$ (*warm up*) for 3K steps, stays constant at $5e-4$ for 35K steps (*hold*), and then exponentially decays to final learning rate of $1e-5$ at 75K steps. In these experiments, the convergence is achieved usually in between 85K - 100K steps. We report the relative word error rate reduction (WERR) over the baseline model in Table 2 trained without distillation, throughout all experiments with in-house data. We use a beam size of 6 during inference.

4.3. LibriSpeech experimental setup

For LibriSpeech [33] experiments, we remove the time reduction component in the encoders and introduce a dropout rate of 30% for the decoder. The vocabulary size is 4096 (+1 blank symbol), hence the fully-connected output projection from the encoder/decoder becomes 4097. The learning rate is also modified accordingly; the *warm up* and *hold* stages are shortened to 1K steps and 20K steps, respectively. The models usually converge around 80-95K steps in these experiments. A beam size of 16 is used during decoding.

5. Results and Discussion

First, using the experimental setup explained in Section 4.2, we demonstrate that the encoder distillation is ineffective when the student and teacher models are not co-learned, and do not share a common decoder. In other words, the teacher and student are two separate RNN-Ts, and the encoder representation of a pre-trained, static teacher RNN-T is used in distillation. The teacher RNN-T is trained on the same data and its parameters are frozen during the student RNN-T training. From the results in Table 2, we notice only minimal improvements in the range of 0.9-1.61% relative WERR.

Next, we show that in order for the encoder distillation to work effectively, the decoder must be shared across the teacher and student encoders as seen in Figure 2. We use Equation 2

to compute the total loss, and co-learn both the teacher and student parameters. From the results in Table 2, we can notice that consistent improvements across all test sets are obtained even before the distillation loss is introduced (i.e., $\lambda=0.0$). Upon introducing the distillation loss (i.e., $\lambda=1.0$), there is additional improvements in the range of 1.58-3.47%. The long-form ASR test set (*long*) benefits the most (8.4% improvement), suggesting the distilled encoder is able to retain long-range acoustic contexts. It is worth noting that the co-learned teacher ($\text{Teacher}_{\text{Co}}$) performs at the same level of the standalone-trained teacher except on the tail distribution test set.

Table 2: *On in-house test sets: relative WERR (%) reported over the Baseline model with smaller encoder trained without distillation. Vanilla student model (row 2) distilled from a pre-trained, static teacher shows minimal improvements, whereas the CoDERT trained student ($\lambda=1.0$) shows significant improvement on all test sets.*

Model	Encoder Params	test(%)	long(%)	tail(%)
Baseline	16M	-	-	-
Student	16M	1.61	1.73	0.9
Teacher	46M	20.16	21.15	16.45
CoDERT				
$\lambda=0.0$	16M	4.03	4.93	3.79
$\lambda=1.0$	16M	6.45	8.4	5.37
$\text{Teacher}_{\text{Co}}$	46M	20.16	21.15	15.82

Next, we study the effect of distilling top-k logits of highest magnitudes. Preserving only top-k logits for distillation is a common practice in hybrid acoustic model distillation [34], to remove unwanted noise in the logits. However, as shown in Table 3, it is noticeable that our encoder distillation works best, especially on *long* and *tail* test sets, when all 4001 logits are used in distillation. Our findings corroborate with the earlier discussions on KD taking advantage of high entropy at the encoder output and advantage of distilling over the entire $B \times T \times V$ space instead of preserving only the top-k logits.

Table 3: *Results corresponding to the CoDERT student ($\lambda=1.0$): relative WERR numbers for distilling using only the top-k logits of the encoder.*

top-k value	test (%)	long (%)	tail (%)
5	4.04	4.93	3.16
50	4.83	5.79	4.43
500	6.45	7.53	4.43
4001 (all)	6.45	8.4	5.37

Implicit Distillation: It is evident from results reported in Table 2 that using shared decoder and co-learning increases the distillation efficiency. This is because the decoder learns its own unique representations and greatly influences the final joint network outputs (see Figure 1), and in learning the alignment [35]. Hence, the encoder distillation is nearly futile without sharing the decoder with the teacher. Interestingly, we discover a phenomenon we call *implicit distillation* (Figure 3, center plot), where the output between the teacher and student encoders are

implicitly minimized without introducing the distillation loss ($\lambda = 0.0$) when the decoder is shared. The teacher-student error is further driven down by introducing the distillation loss.

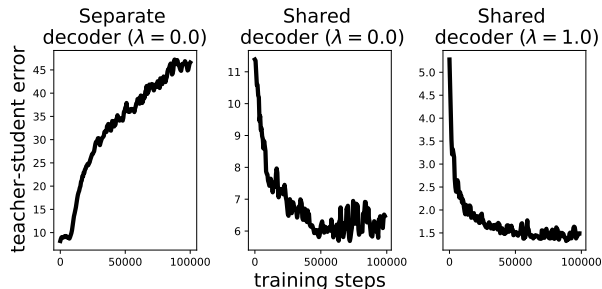


Figure 3: *Error between teacher and student encoder logits in (a) separate decoder without distillation (left), (b) shared decoder without distillation (center), and (c) shared decoder with distillation.*

Finally, we select the best settings from in-house data experiments (shared decoder, co-learning, and full-logit distillation instead of only top-k logits) moving forward with LibriSpeech data. The results presented in Table 4 once again indicate the competency of CoDERT across all dev and test sets on LibriSpeech. The co-learned model without distillation shows no improvement on test-clean, and a modest improvement on test-other (2.29%), but upon introducing the distillation loss, one can notice significant WER reductions on both test-clean (6.18%) and test-other (5.05%).

Table 4: *On LibriSpeech: Absolute (abs) WER and relative (rel) WERR numbers on LibriSpeech data for baseline model trained without distillation and models trained through CoDERT.*

Model	dev-clean		dev-other		test-clean		test-other	
	abs.	rel.	abs.	rel.	abs.	rel.	abs.	rel.
Baseline	9.6	-	21.1	-	9.7	-	21.8	-
CoDERT								
$\lambda=0.0$	9.1	5.2	20.5	2.84	9.7	0.0	21.3	2.29
$\lambda=1.0$	8.9	7.29	19.8	6.16	9.1	6.18	20.7	5.05
$\text{Teacher}_{\text{Co}}$	7.9	17.7	18.8	10.9	8.3	14.43	19.5	10.55

6. Conclusion

In this work, we explored distilling the high-level representations of the RNN-T encoder and investigated its effectiveness in building a RNN-T model with a compact encoder. We studied training strategies for encoder distillation and learned that the method works at its best when the student RNN-T shares its decoder with a teacher model that is trained in tandem. We reported an interesting implicit distillation behaviour that occurs during the tandem training. Our results showed significant improvements across various test sets of both real-world and open-source data, demonstrating the efficacy of the proposed technique.

7. Acknowledgements

We thank Ariya Rastrow, Gautam Tiwari, Harish Arsikere and Maurizio Omologo for helpful discussions and feedback.

8. References

- [1] Y. Shangquan, J. Li, Q. Liang, R. Alvarez, and I. McGraw, "Optimizing speech recognition for the edge," *arXiv preprint arXiv:1909.12408*, 2019.
- [2] C.-F. Yeh, J. Mahadeokar, K. Kalgaonkar, Y. Wang, D. Le, M. Jain, K. Schubert, C. Fuegen, and M. L. Seltzer, "Transformer-transducer: End-to-end speech recognition with self-attention," *arXiv preprint arXiv:1910.12977*, 2019.
- [3] J. Li, R. Zhao, H. Hu, and Y. Gong, "Improving rnn transducer modeling for end-to-end speech recognition," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 114–121.
- [4] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and N. Jaitly, "A comparison of sequence-to-sequence models for speech recognition," in *Proc. Interspeech 2017*, 2017, pp. 939–943. [Online]. Available: <http://dx.doi.org/10.21437/Interspeech.2017-233>
- [5] T. N. Sainath, Y. He, B. Li, A. Narayanan, R. Pang, A. Bruguier, S.-y. Chang, W. Li, R. Alvarez, Z. Chen *et al.*, "A streaming on-device end-to-end model surpassing server-side conventional model quality and latency," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6059–6063.
- [6] J. Li, R. Zhao, Z. Meng, Y. Liu, W. Wei, S. Parthasarathy, V. Mazalov, Z. Wang, L. He, S. Zhao *et al.*, "Developing rnn-t models surpassing high-performance hybrid models with customization capability," *arXiv preprint arXiv:2007.15188*, 2020.
- [7] J. Xue, J. Li, and Y. Gong, "Restructuring of deep neural network acoustic models with singular value decomposition," in *Interspeech*, 2013, pp. 2365–2369.
- [8] R. Prabhavalkar, O. Alsharif, A. Bruguier, and L. McGraw, "On the compression of recurrent neural networks with an application to lvcvr acoustic modeling for embedded speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5970–5974.
- [9] H. D. Nguyen, A. Alexandridis, and A. Mouchtaris, "Quantization aware training with absolute-cosine regularization for automatic speech recognition," in *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2020.
- [10] Z. Wu, D. Zhao, Q. Liang, J. Yu, A. Gulati, and R. Pang, "Dynamic sparsity neural networks for automatic speech recognition," *arXiv preprint arXiv:2005.10627*, 2020.
- [11] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," *arXiv preprint arXiv:1803.03635*, 2018.
- [12] K. Zhen, H. D. Nguyen, F.-J. Chang, A. Mouchtaris, and A. Rastrow, "Sparsification via compressed sensing for automatic speech recognition," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.
- [13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [14] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, "Light gated recurrent units for speech recognition," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 2, pp. 92–102, 2018.
- [15] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2222–2232, 2016.
- [16] T. Lei, Y. Zhang, S. I. Wang, H. Dai, and Y. Artzi, "Simple recurrent units for highly parallelizable recurrence," *arXiv preprint arXiv:1709.02755*, 2017.
- [17] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.
- [18] S. H. K. Parthasarathi and N. Strom, "Lessons from building acoustic models with a million hours of speech," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6670–6674.
- [19] A. Mishra and D. Marr, "Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy," *arXiv preprint arXiv:1711.05852*, 2017.
- [20] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.
- [21] Y. Chebotar and A. Waters, "Distilling knowledge from ensembles of neural networks for speech recognition," in *Interspeech*, 2016, pp. 3439–3443.
- [22] G. Kurata and K. Audhkhasi, "Guiding ctc posterior spike timings for improved posterior fusion and knowledge distillation," *arXiv preprint arXiv:1904.08311*, 2019.
- [23] J. Liu, R. V. Swaminathan, S. H. K. Parthasarathi, C. Lyu, A. Mouchtaris, and S. Kunzmann, "Exploiting large-scale teacher-student training for on-device acoustic models," *arXiv preprint arXiv:2106.06126*, 2021.
- [24] G. Kurata and G. Saon, "Knowledge distillation from offline to streaming rnn transducer for end-to-end speech recognition," *Proc. Interspeech 2020*, pp. 2117–2121, 2020.
- [25] S. Panchapagesan, D. S. Park, C.-C. Chiu, Y. Shangquan, Q. Liang, and A. Gruenstein, "Efficient knowledge distillation for rnn-transducer models," *arXiv preprint arXiv:2011.06110*, 2020.
- [26] C. Liu, F. Zhang, D. Le, S. Kim, Y. Saraf, and G. Zweig, "Improving rnn transducer based asr with auxiliary tasks," *arXiv preprint arXiv:2011.03109*, 2020.
- [27] Z. Ke, D. Wang, Q. Yan, J. Ren, and R. W. Lau, "Dual student: Breaking the limits of the teacher in semi-supervised learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6728–6736.
- [28] J. Yu and T. S. Huang, "Universally slimmable networks and improved training techniques," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1803–1811.
- [29] J. Yu, W. Han, A. Gulati, C.-C. Chiu, B. Li, T. N. Sainath, Y. Wu, and R. Pang, "Dual-mode asr: Unify and improve streaming asr with full-context modeling," *Proceedings of ICLR*, 2021.
- [30] A. Graves, S. Fernández, and F. Gomez, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *In Proceedings of the International Conference on Machine Learning, ICML 2006*, 2006, pp. 369–376.
- [31] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang *et al.*, "Streaming end-to-end speech recognition for mobile devices," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6381–6385.
- [32] J. Macoskey, G. P. Strimel, and A. Rastrow, "Bi-focal neural asr: Exploiting keyword spotting for inference optimization," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021.
- [33] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.
- [34] L. Mošner, M. Wu, A. Raju, S. H. K. Parthasarathi, K. Kumatani, S. Sundaram, R. Maas, and B. Hoffmeister, "Improving noise robustness of automatic speech recognition via parallel data and teacher-student learning," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6475–6479.
- [35] M. Ghodsi, X. Liu, J. Apfel, R. Cabrera, and E. Weinstein, "Rnn-transducer with stateless prediction network," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7049–7053.