



Enhancing the Unified Streaming and Non-streaming Model with Contrastive Learning

Yuting Yang, Yuke Li*, Binbin Du

NetEase Yidun AI Lab, Hangzhou, China

{yangyuting04, liyuke, dubinbin}@corp.netease.com

Abstract

The unified streaming and non-streaming speech recognition model has achieved great success due to its comprehensive capabilities. In this paper, we propose to improve the accuracy of the unified model by bridging the inherent representation gap between the streaming and non-streaming modes with a contrastive objective. Specifically, the top-layer hidden representation at the same frame of the streaming and non-streaming modes are regarded as a positive pair, encouraging the representation of the streaming mode close to its non-streaming counterpart. The multiple negative samples are randomly selected from the rest frames of the same sample under the non-streaming mode. Experimental results demonstrate that the proposed method achieves consistent improvements toward the unified model in both streaming and non-streaming modes. Our method achieves CER of 4.66% in the streaming mode and CER of 4.31% in the non-streaming mode, which sets a new state-of-the-art on the AISHELL-1 benchmark.

Index Terms: unified model, streaming speech recognition, contrastive learning

1. Introduction

Recently, the rapid advancements in deep neural networks have made end-to-end (E2E) automatic speech recognition (ASR) systems a mainstream focus of ASR research. Compared with the traditional speech recognition system, end-to-end ASR systems have the advantages of simpler system composition, training process, and decoding procedure. Various E2E ASR models have been explored in the literature, which can be roughly categorized into three main approaches: Connectionist Temporal Classification (CTC) models [1], Transducer models [2, 3], and attention-based encoder-decoder (AED) systems [4–6].

Streaming scenarios pose a challenge to state-of-the-art E2E ASR systems like Transformer [5] and Conformer [7]. The self-attention mechanism and the source-target multi-head attention require the full context of the sequences, resulting in significant latency, which in turn limits their usability in streaming scenarios. Previously, many efforts have been made to overcome these limitations. Chunk-based [8, 9] and look-ahead [10] methods are proposed to control the latency of the encoder module. Triggered attention (TA) [10, 11] and monotonic chunkwise attention (MoChA) [9, 12] are used to restrict the context of the attention mechanism. In addition, two-pass models have been developed to improve the final performance by integrating supplementary modules that reevaluate the hypotheses generated by the first-pass streaming model [13, 14].

The unified models [15–20] provide a simple and efficient solution to the ASR tasks by learning and optimizing a single neural network for both streaming and non-streaming scenarios. During the inference process, the capabilities for different scenarios are enabled by configuring different decoding settings. In addition, previous literature [20] demonstrated that the performance of streaming mode benefits from the unified way with weight sharing and joint training. During the training process, knowledge distillation is employed to make the performance of the streaming mode benefit from the knowledge derived from the non-streaming mode [19, 20]. However, knowledge distillation improves the performance of streaming mode while the performance improvement of non-streaming mode is negligible. Enhancing the unified model’s performance even further remains a challenge.

In this work, we focus on improving the performance bottleneck of unified ASR models from the neural representation perspective. We expect the result of the streaming mode to be similar to that in the non-streaming mode. From the aspect of neural representation, when the encoded representation of streaming mode is close to its non-streaming counterpart, the content of their speech recognition results should be similar. Inspired by the recent progress of contrastive learning frameworks in computer vision [21, 22] and speech processing [23, 24], we designed a simple yet effective method for unified ASR models to learn the representations that meet the above-mentioned condition explicitly. On the one hand, the contrastive task bridges the gap between the representations in streaming and non-streaming modes. On the other hand, the unified model benefits from the uniform distribution, which is a significant property in contrastive learning [25, 26]. Our contributions are as follows.

- We develop a novel contrastive learning method to improve the performance of the unified streaming and non-streaming model, built upon the joint training framework.
- Our experiments on the AISHELL-1 benchmark demonstrated that the proposed method achieves 4.66% in the streaming mode and 4.31% in the non-streaming mode, outperforming the previous state-of-the-art models.
- We show that the proposed method indeed learns unified representations for two modes and performs better to improve the accuracy of the unified model.

2. Related work

2.1. Unifying streaming and non-streaming ASR

Recently, there has been growing interest in the unified model since it not only reduces the resource consumption of model training and deployment but also improves the performance of

* Corresponding author

streaming speech recognition. There are several unified models based on the Transducer network [18, 20, 27]. For instance, dual-mode [20] developed a unified framework with shared weights and joint training mechanisms, and utilized knowledge distillation to further strengthen the performance of the streaming mode. Cascaded model [18] built a single E2E ASR model consisting of streaming and non-streaming encoders. The decoder module generates outputs by utilizing the output produced by either the streaming or non-streaming encoder.

In addition, some solutions adopt attention-based approaches, such as Universal ASR [17], U2 [15], and U2++ [16] models. Both U2 [15] and U2++ [16] models utilized the hybrid CTC/Attention architecture and designed a dynamic chunks strategy to meet diverse latency requirements during inference. For decoding, U2 [15] proposed a two-pass decoding method that involves a CTC-Decoder to generate initial hypotheses followed by an Attention-Decoder to finalize the results with full context attention at the end of the input. Additionally, U2++ [16] further enhanced the model with a bidirectional attention decoder and a spectral augmentation method named SpecSub.

2.2. Contrastive learning

Contrastive learning is proposed to learn general data representations from unlabeled data, which aims at maximizing the similarity between anchors and their positive examples while keeping randomly sampled negative examples far away from it in the embedding space. Contrastive learning is extensively used in unsupervised or self-supervised domains, especially in the field of computer vision [21, 22]. Since no labels are available, negative pairs are formed by the anchor and randomly chosen samples from the minibatch, and the positive pair often consists of the author and its variants via data augmentation operators such as crop, color distortion, and Gaussian blur. Contrastive learning is widely used in the natural language processing tasks, such as sentence representation learning [28] and machine translation [29]. Contrastive learning is also applicable to the area of speech processing and achieves promising results. For instance, wav2vec [24] learns general representations from raw audio via a contrastive task that requires distinguishing a true future audio sample from negatives, and then the representations serve as an input to a speech recognition system. In addition, wav2vec 2.0 [23] is a self-supervised framework that learns to identify the correct masked-out quantized representation with a contextualized representation from a Transformer model.

Inspired by the contrastive learning framework, we propose to introduce a contrastive task into the unified model, where the positive pairs consist of the representations computed on chunk-based context (streaming) mode and full context (non-streaming) mode, encouraging the neural representation pairs generated in two modes near together.

3. Method

3.1. Model architecture

The model architecture used in this paper is shown in Figure 1. We adopt the same hybrid CTC/Attention [30] architecture as U2 [15] and U2++ [16] models. It mainly contains shared encoder and decoder modules. The former learns the context of the acoustic features in both full and chunk-based manners, and the latter predicts the next token given the high-level acoustic representations and previous token embeddings. We adopt the Conformer [7] encoder and the Transformer [5] decoder. An encoder block of the Conformer is composed of four mod-

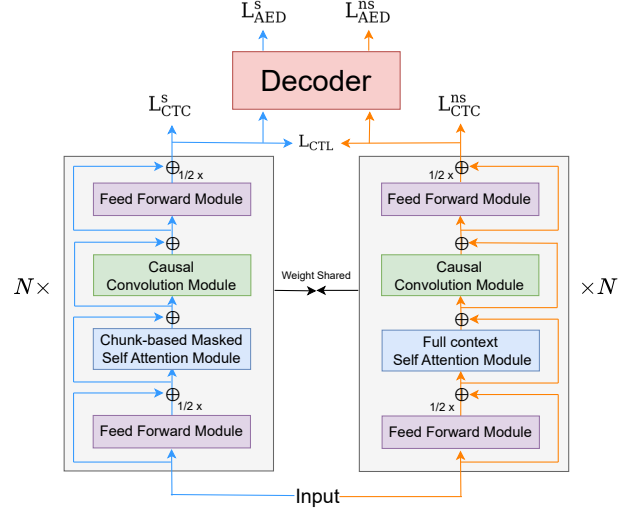


Figure 1: *The framework of our method.* ($\mathcal{L}_{CTC}^s, \mathcal{L}_{AED}^s$) denote the CTC loss and AED loss from the streaming branch (blue line). By contrast, ($\mathcal{L}_{CTC}^{ns}, \mathcal{L}_{AED}^{ns}$) denote the CTC loss and AED loss from the non-streaming branch (yellow line). \mathcal{L}_{CTL} is the contrastive loss term. Layer normalization, the front convolution module, and the token embedding module are omitted.

ules, i.e., a feed-forward network, a self-attention module, a convolution module, and a second feed-forward network. In addition, the decoder layer of the Transformer is composed of three modules, a self-attention module, a multi-head attention module, and a feed-forward module. During training, the self-attention operation is restricted to the chunk-based feature context in the streaming mode via a predefined mask matrix. In addition, causal convolution [31] layers are used where the convolution window is shifted to the left to avoid additional latency.

3.2. Joint training

we adopt a joint training strategy [20] for optimization. Specifically, the total loss function is a combination of the streaming ASR loss, the non-streaming ASR loss, and the contrastive loss. Formally,

$$\mathcal{L} = \mathcal{L}_{ASR}^s + \mathcal{L}_{ASR}^{ns} + \mathcal{L}_{CTL}, \quad (1)$$

where

$$\begin{aligned} \mathcal{L}_{ASR}^s &= \lambda \mathcal{L}_{CTC}^s + (1 - \lambda) \mathcal{L}_{AED}^s, \\ \mathcal{L}_{ASR}^{ns} &= \lambda \mathcal{L}_{CTC}^{ns} + (1 - \lambda) \mathcal{L}_{AED}^{ns}. \end{aligned} \quad (2)$$

The first two terms in Eq.1 are streaming ASR loss \mathcal{L}_{ASR}^s and non-streaming ASR loss \mathcal{L}_{ASR}^{ns} built from the streaming and non-streaming branches, respectively. Specifically, as shown in Eq.2, the \mathcal{L}_{ASR}^s term is a combination of CTC loss \mathcal{L}_{CTC}^s and AED loss \mathcal{L}_{AED}^s , and the \mathcal{L}_{ASR}^{ns} term is a combination of CTC loss \mathcal{L}_{CTC}^{ns} and AED loss \mathcal{L}_{AED}^{ns} as well. For the streaming branch, we use dynamic chunk size for different batches, which is sampled from 1 to 25 [15]. In addition, the last element introduces a contrastive loss term \mathcal{L}_{CTL} (see Section 3.3 for details).

3.3. Contrastive loss for unified ASR models

We apply the encoder module to extract contextual features and product high-level hidden representations. We define that h^s

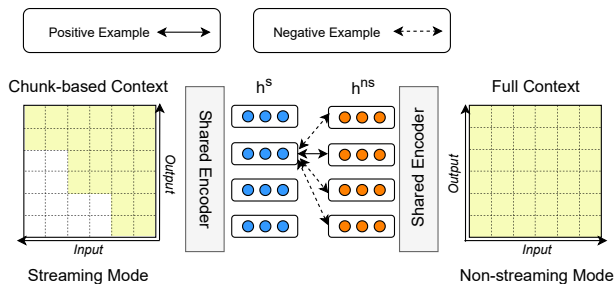


Figure 2: An illustration of the proposed method with contrastive learning.

and h^{ns} denote the output of the encoder module generated by the streaming and the non-streaming modes, respectively. Figure 2 illustrates the proposed method with contrastive learning. The main idea of our method is to introduce a loss that brings positive pairs consisting of streaming representations h^s and corresponding non-streaming representations h^{ns} near together while pushing irrelevant pairs (negative pairs) far apart from each other. Specifically, the frame-level contrastive loss for (h_i^s, h_i^{ns}) is defined as

$$\mathcal{L}_{fctl}(h_i^s, h_i^{ns}) = -\log \frac{\exp(\text{sim}(h_i^s, h_i^{ns})/\tau)}{\sum_{k_j \in A} \exp(\text{sim}(h_i^s, k_j)/\tau)}. \quad (3)$$

Given a positive example of such a representation pair (h_i^s, h_i^{ns}) , we randomly pick a set of N representations from the rest frames of h^{ns} as negative examples. In Eq.3, A includes h_i^{ns} and N distractors, and τ is the temperature hyperparameter. We compute the cosine similarity $\text{sim}(a, b) = \frac{a^T b}{\|a\| \|b\|}$ between the streaming representation and the non-streaming representation. The overall contrastive loss can be formulated as

$$\mathcal{L}_{CTL} = \frac{1}{n} \sum_{i=1}^n \mathcal{L}_{fctl}(h_i^s, h_i^{ns}), \quad (4)$$

where n is the number of speech frames.

4. Experiments

We conduct experiments on the AISHELL-1 [32] corpus, containing over 170 hours of Mandarin speech data from 400 speakers. The input speech features are 80-dimensional filterbank (FBank) features computed on 25ms windows with 10ms shifts. The vocabulary includes 4233 characters.

4.1. Experimental setups

We conduct experiments using the Wenet [33] toolkit. We adopt the proposed method on two baseline models: U2 [15] and U2++ [16]. Two convolution downsampling layers with kernel size 3×3 and stride 2 in the front of the encoder are adopted. For U2 [15], it adopts a 12-layer Conformer encoder structure and a 6-layer Transformer decoder structure. The dimension of the attention layer is 256 with 4 split heads, and the dimension of the feed-forward layer is 2048. The kernel size is 15 for the Conformer encoder module. The U2 model has $\sim 46M$ parameters. For U2++ [16], The decoder module is a bidirectional structure with a 3-layer left-to-right decoder and a 3-layer right-to-left decoder. And the kernel size is 8 for Conformer. The other structure settings are the same as in U2. The U2++ model has $\sim 48M$ parameters.

Table 1: Character error rate (CER) on the AISHELL-1 test set. 1st pass and 2st pass stand for CTC prefix beam search and attention rescoring, respectively. * is our reproduction.

Methods	Decoding mode	Decoding chunk size			
		Full	16	8	4
U2 [15]	1st pass	5.51	6.23	6.57	6.92
	2st pass	5.05	5.48	5.66	5.85
U2++ [16]	1st pass	5.19	5.81	-	-
	2st pass	4.63	5.05	-	-
U2++*	1st pass	5.23	5.87	6.15	6.37
	2st pass	4.66	5.05	5.23	5.39
Our method with					
U2	1st pass	5.12	5.84	6.18	6.53
	2st pass	4.76	5.21	5.43	5.63
U2++	1st pass	5.03	5.65	5.87	6.16
	2st pass	4.52	4.90	5.09	5.18

We apply speed perturbation [34] and SpecAugment [35] to the training data. U2++ based models additionally adopt the SpecsSub [16] method. For training, we keep almost the same training settings as in the AISHELL-1 recipe, including regularization, optimizer, learning rate schedule, and data augmentation. Please refer to the open-sourced recipe¹ for more details. We train the model for 240 epochs and 360 epochs on 8 V100 GPUs with U2 models and U2++ models respectively, and a final model is obtained by averaging the top 30 models with the best validation loss. The λ in Eq.2 is set to 0.3 according to the recipe. Additionally, we choose the temperature hyperparameter in contrastive loss ranging from 0.05 to 0.8, and we set $\tau = 0.4$ for the U2 model and $\tau = 0.6$ for the U2++ model that achieve better performance than other options. Following [23], we use $N = 100$ distractors in contrastive loss. For decoding, we present the first-pass (CTC prefix beam search [36]) results and the second-pass (attention rescoring [15]) results, respectively. In Table 2 and Table 3, a 3-gram language model (LM) was trained on the transcripts used for shallow fusion.

4.2. Results

4.2.1. Comparison with baseline models

Table 1 compares the (CER) results of our method on the AISHELL-1 test set with baseline models [15, 16]. We can see from Table 1 that the proposed method outperforms the baseline models while using exactly the same architectures, demonstrating that contrastive learning gives a consistent improvement over the baseline model in both the streaming and non-streaming modes. The performance of the non-streaming mode is also improved by our framework with contrastive loss. Our explanation is that the contrastive loss makes the positive pair closer while pushing the irrelevant pairs farther. Therefore, the non-streaming mode benefits from the uniform distribution.

4.2.2. Comparison with published streaming models

Table 2 shows the comparisons with recently published streaming systems and unified systems with streaming mode on the AISHELL-1 test set. Specifically, our method achieves CER of 4.9% without external LM and CER of 4.66% with a 3-gram LM for shallow fusion on the AISHELL-1 test set, surpassing recently published streaming models. To our best knowledge, this is the state-of-the-art performance for streaming ASR systems on the AISHELL-1 task.

¹<https://github.com/wenet-e2e/wenet/tree/main/examples/aishell/s0>

Table 2: Comparison with recently published streaming E2E systems on the AISHELL-1 test set. Δ denotes the additional latency introduced by rescoring the first-pass hypotheses.

Methods	Latency (ms)	Test set
HS-DACS Transformer [37]	1280	6.80
WNARS (w/ LM) [38]	640	5.22
U2 [15]	640 + Δ	5.42
U2++ [16]	640 + Δ	5.05
CUSIDE (w/ LM) [39]	400 + 2	5.47
+NNLM rescoring	400 + 2 + Δ	4.79
Our method with		
U2	640 + Δ	5.21
+ w/ a 3-gram LM	640 + Δ	4.87
U2++	640 + Δ	4.90
+ w/ a 3-gram LM	640 + Δ	4.66

Table 3: Comparison of our unified model with pure non-streaming E2E systems on AISHELL-1. * denotes our reproduction using the Conformer backbone with 46M parameters.

Models	Dev set	Test set
Pure non-streaming *	4.29	4.67
Pure non-streaming [40]	4.4	4.7
Our method with		
U2	4.36	4.76
+ w/ a 3-gram LM	4.14	4.51
U2++	4.23	4.52
+ w/ a 3-gram LM	4.08	4.31

4.2.3. Comparison with pure non-streaming models

Table 3 further compares the accuracy of our unified model with pure non-streaming models on the AISHELL-1 dataset. Specifically, our unified method achieves CER of 4.23%/4.52% without external LM and CER of 4.08%/4.31% with a 3-gram LM for shallow fusion on the AISHELL-1 dev/test sets under the non-streaming mode. These results demonstrate that our unified models obtain comparable results to pure offline models while having the benefit of comprehensive capabilities on streaming and non-streaming scenarios.

4.3. Discussions

4.3.1. Is contrastive loss more effective?

We attempt to introduce the contrastive loss term to close the distance between streaming and non-streaming representations, which are obtained by restricted context and full context of input sequences, respectively. However, there are other options to achieve this goal from the idea of knowledge distillation, such as L2 loss. It bridges the gap by directly reducing the Euclidean distance between the representations of streaming and non-streaming modes, and no negative samples were introduced in this process.

We study the effects of different loss terms based on the U2 model architecture. The results are shown in Table 4. Both the proposed method and L2 loss improve the accuracy of the unified model, and the proposed method performs better to this goal. These results in Table 4 suggest the effectiveness of incorporating the contrastive loss term in enhancing the performance of the unified model.

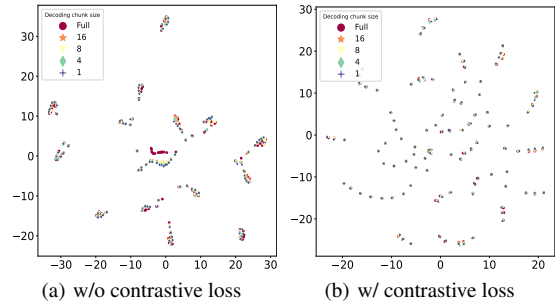


Figure 3: Visualization of the hidden representation of streaming and non-streaming modes. We utilize T-SNE to reduce the dimension into 2D. The red dots are the encoded representations from non-streaming mode. In contrast, the other points stand for the representations from the streaming mode with decoding chunk size {16, 8, 4, 1}, respectively. The utterance is randomly selected from the test set.

Table 4: Character error rate (CER) on the AISHELL-1 test set under different loss terms settings with the U2 [15] model. 1st pass and 2st pass decoding mode stand for CTC prefix beam search and attention rescoring, respectively.

Contrastive Loss	L2 Loss	1st pass		2st pass	
		Full	16	full	16
Yes	No	5.12	5.84	4.76	5.21
No	Yes	5.25	5.94	4.87	5.28
No	No	5.32	6.03	4.87	5.33

4.3.2. Analysis on the gap between two modes

Figure 3 visualizes embedding spaces learned by the standard U2 model without contrastive loss and the proposed model with contrastive learning. Figure 3(a) shows that the representations are dissimilar. The gap exists between the two modes without any explicit terms to make them together. In contrast, as shown in Figure 3(b), the proposed model with contrastive learning is able to bring the representations of streaming and non-streaming modes closer. In addition, the representations are more uniformly distributed in the embedding space. These mean that the proposed method indeed encourages the representation pairs generated in two modes near together, thus improving the accuracy of the unified model.

5. Conclusions

In this paper, we propose to enhance the unified streaming and non-streaming ASR model with contrastive learning. Our method aims to bridge the representation gap between the full-context and chunk-based context modes. Major contributions are two-fold: (1) our method introduced a simple yet effective contrastive objective to make the representations at the same frame under two different modes to be similar; (2) extensive experiments demonstrate that our method set a new state-of-the-art on the widely-used AISHELL-1 benchmark. We demonstrated the effectiveness of the proposed method on the attention-based architecture, while our approach can be applied to other E2E ASR models, such as CTC-based and RNN-T models, which will be interesting future work. In addition, we will extend our method to more large-scale datasets in future work.

6. References

- [1] A. Graves, S. Fernández, F. Gomez *et al.*, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.
- [2] E. Battenberg, J. Chen, R. Child *et al.*, “Exploring neural transducers for end-to-end speech recognition,” in *2017 ASRU*. IEEE, 2017, pp. 206–213.
- [3] A. Graves, “Sequence transduction with recurrent neural networks,” *arXiv preprint arXiv:1211.3711*, 2012.
- [4] W. Chan, N. Jaitly, Q. Le *et al.*, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.
- [5] A. Vaswani, N. Shazeer, N. Parmar *et al.*, “Attention is all you need,” in *NIPS*, 2017.
- [6] L. Dong, S. Xu, and B. Xu, “Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition,” in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 5884–5888.
- [7] A. Gulati, J. Qin, C. Chiu *et al.*, “Conformer: Convolution-augmented transformer for speech recognition,” in *INTER-SPEECH*, 2020.
- [8] L. Dong, F. Wang, and B. Xu, “Self-attention aligner: A latency-control end-to-end model for asr using self-attention network and chunk-hopping,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5656–5660.
- [9] E. Tsunoo, Y. Kashiwagi, T. Kumakura *et al.*, “Towards on-line end-to-end transformer automatic speech recognition,” *arXiv preprint arXiv:1910.11871*, 2019.
- [10] N. Moritz, T. Hori, and J. Le, “Streaming automatic speech recognition with the transformer model,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6074–6078.
- [11] N. Moritz, T. Hori, and J. Le Roux, “Triggered attention for end-to-end speech recognition,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5666–5670.
- [12] C.-C. Chiu and C. Raffel, “Monotonic chunkwise attention,” in *ICLR*, 2018.
- [13] T. N. Sainath, R. Pang, D. Rybach *et al.*, “Two-pass end-to-end speech recognition,” in *Interspeech*, 2019.
- [14] K. Hu, T. N. Sainath, R. Pang, and R. Prabhavalkar, “Deliberation model based two-pass end-to-end speech recognition,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7799–7803.
- [15] B. Zhang, D. Wu, Z. Yao *et al.*, “Unified streaming and non-streaming two-pass end-to-end model for speech recognition,” *arXiv preprint arXiv:2012.05481*, 2020.
- [16] D. Wu, B. Zhang, C. Yang *et al.*, “U2++: Unified two-pass bidirectional end-to-end model for speech recognition,” *arXiv preprint arXiv:2106.05642*, 2021.
- [17] Z. Gao, S. Zhang, M. Lei *et al.*, “Universal asr: Unifying streaming and non-streaming asr using a single encoder-decoder model,” *arXiv preprint arXiv:2010.14099*, 2020.
- [18] A. Narayanan, T. N. Sainath, R. Pang *et al.*, “Cascaded encoders for unifying streaming and non-streaming asr,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5629–5633.
- [19] C. Liang, X.-L. Zhang, B. Zhang *et al.*, “Fast-u2++: Fast and accurate end-to-end speech recognition in joint ctc/attention frames,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [20] J. Yu, W. Han, A. Gulati *et al.*, “Dual-mode asr: Unify and improve streaming asr with full-context modeling,” in *ICLR*, 2021.
- [21] T. Chen, S. Kornblith, M. Norouzi *et al.*, “A simple framework for contrastive learning of visual representations,” in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.
- [22] K. He, H. Fan, Y. Wu *et al.*, “Momentum contrast for unsupervised visual representation learning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.
- [23] A. Baevski, Y. Zhou, A. Mohamed *et al.*, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.
- [24] S. Schneider, A. Baevski, R. Collobert *et al.*, “wav2vec: Unsupervised pre-training for speech recognition,” in *Interspeech*, 2019.
- [25] F. Wang and H. Liu, “Understanding the behaviour of contrastive loss,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 2495–2504.
- [26] T. Wang and P. Isola, “Understanding contrastive representation learning through alignment and uniformity on the hypersphere,” in *International Conference on Machine Learning*. PMLR, 2020, pp. 9929–9939.
- [27] A. Tripathi, J. Kim, Q. Zhang *et al.*, “Transformer transducer: One model unifying streaming and non-streaming speech recognition,” *arXiv preprint arXiv:2010.03192*, 2020.
- [28] L. Kong, C. d. M. d’Autume, W. Ling *et al.*, “A mutual information maximization perspective of language representation learning,” in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.
- [29] X. Wei, R. Weng, Y. Hu *et al.*, “On learning universal representations across languages,” in *ICLR*, 2021.
- [30] S. Watanabe, T. Hori, S. Kim *et al.*, “Hybrid ctc/attention architecture for end-to-end speech recognition,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.
- [31] A. v. d. Oord, S. Dieleman, H. Zen *et al.*, “Wavenet: A generative model for raw audio,” in *The 9th ISCA Speech Synthesis Workshop, Sunnyvale, CA, USA, 13-15 September 2016*, 2016.
- [32] H. Bu, J. Du, X. Na *et al.*, “Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline,” in *2017 20th conference of the oriental chapter of the international coordinating committee on speech databases and speech I/O systems and assessment (O-COCOSDA)*, 2017.
- [33] Z. Yao, D. Wu, X. Wang *et al.*, “Wenet: Production oriented streaming and non-streaming end-to-end speech recognition toolkit,” in *Interspeech*, 2021.
- [34] T. Ko, V. Peddinti, D. Povey *et al.*, “Audio augmentation for speech recognition,” in *ISCA*, 2015.
- [35] D. S. Park, W. Chan, Y. Zhang *et al.*, “SpecAugment: A simple data augmentation method for automatic speech recognition,” in *Interspeech*, 2019.
- [36] A. Y. Hannun, A. L. Maas, D. Jurafsky *et al.*, “First-pass large vocabulary continuous speech recognition using bi-directional recurrent dnns,” *arXiv preprint arXiv:1408.2873*, 2014.
- [37] M. Li, C. Zorilă, and R. Doddipatla, “Head-synchronous decoding for transformer-based streaming asr,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5909–5913.
- [38] Z. Wang, W. Yang, P. Zhou *et al.*, “Wnars: Wfst based non-autoregressive streaming end-to-end speech recognition,” *arXiv preprint arXiv:2104.03587*, 2021.
- [39] K. An, H. Zheng, Z. Ou *et al.*, “Cuside: Chunking, simulating future context and decoding for streaming asr,” in *Interspeech*, 2022.
- [40] P. Guo, F. Boyer, X. Chang *et al.*, “Recent developments on espnet toolkit boosted by conformer,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5874–5878.