人工知能学会第二種研究会資料
AIチャレンジ研究会
SIG-Challenge-063-03

End-to-end integration of online and offline encoders using auxiliary losses for automatic speech recognition

Muhammad Shakeel[1*] Yui Sudo[1] Yifan Peng[2] Shinji Watanabe[2]

[1] (株) ホンダ・リサーチ・インスティチュート・ジャパン
[1] Honda Research Institute Japan Co., Ltd.
[2] カーネギーメロン大学
[2] Carnegie Mellon University

Abstract: End-to-end (E2E) automatic speech recognition (ASR) models have two desirable properties: online and offline modes. The online ASR mode, which operates under strict latency constraints, processes speech frames in real-time to provide transcription. Conversely, the offline ASR mode waits for the complete utterance of speech frames before generating a transcription. Recently, the integration of online and offline ASR for recurrent neural network transducers (RNN-T) can be achieved through the joint training of online and offline encoders with a shared decoder. However, this integration comes at the cost of performance degradation in the offline ASR mode, as the shared decoder must handle features of varying contexts. Namely, with E2E integration framework of online and offline encoders, we explore two approaches to enhance the performance of both the ASR modes. First, we introduce separate RNN-T decoders for each ASR mode while maintaining shared encoders, thereby effectively managing features of different contexts. Second, we explore multiple auxiliary loss criteria to introduce additional regularization, thereby enhancing the overall stability and performance of the framework. Overall, evaluation results show 1.8%-2.5% relative character error rate reductions (CERR) on corpora of spontaneous Japanese (CSJ) for online ASR, and 4.4%-6.3% relative CERRs for offline ASR within a single model compared to separate online and offline models.

1 Introduction

End-to-end automatic speech recognition (E2E-ASR) systems [1] strive to achieve low latency and high performance for a variety of tasks, including online [2] and offline [3] ASR. However, the creation of distinct architectures for each task is neither scalable nor flexible. Therefore, a single E2E-ASR framework, capable of handling multiple tasks with high accuracy and adaptability, is preferable. One potential solution is the joint training of online and offline E2E-ASR tasks using shared weights. However, most of the existing methods suffer from negative transfer, a phenomenon where the performance of one task interferes with another and degrades the performance of either of them. For example, in the case of shared weights between online and offline E2E-ASR, the limited contextual features of online encoder could lead to a conflict with full-context offline encoder. This is because the model might not be able to effectively differentiate between the features relevant to each encoder, leading to the inclusion of irrelevant negative samples. To address this issue, we propose an E2E-ASR framework that integrates online and offline ASR and combines the unique capabilities of both the ASR models.

In this study, our primary objective is to optimize both online and offline ASR modes. This framework employs multiple encoders, with one designated for online ASR and another for offline ASR, and separate decoders for each. The framework extracts the hidden states of the online encoder and uses them as input for the offline encoder. This method allows us to integrate the functionalities of both encoders while maintaining separate decoders for online and offline modes. As a general approach rather than relying only on the cascaded integration [4] for performance optimization, our method introduces sepa-

*連絡先：(株) ホンダ・リサーチ・インスティチュート・ジャパン
〒 351-0188 埼玉県和光市本町 8-1
E-mail: shakeel.muhammad@jp.honda-ri.com

rate online and offline recurrent neural network transducer (RNN-T) [5] decoders to leverage varying contextual information from both online and offline encoders. Additionally, we employ connectionist temporal classification (CTC) [6], attention mechanism [7], and masked language model (MLM) [8] based auxiliary losses to bring more regularization and refinement to the E2E-ASR framework. The CTC loss helps in aligning the input features with the output labels, the attention mechanism provides a dynamic alignment between the input and output sequences, and the MLM loss aids in predicting the masked input features.

Through extensive experimentation, we have been able to demonstrate the effectiveness of each auxiliary loss in improving the performance of the offline ASR model. Our results indicate a significant improvement in the accuracy and efficiency of both online and offline ASR models, validating the effectiveness of our integrated framework and auxiliary loss techniques.

2 Related work

One of the current challenges in E2E-ASR is to develop a unified model [9,10] that can handle both online and offline scenarios. Online ASR is suitable for applications that require low latency and real-time feedback, such as voice assistants and online meetings. However, online ASR can only use limited context information from the past and present frames, which may limit its accuracy and robustness. On the other hand, offline ASR is suitable for applications that do not have strict latency constraints, such as offline transcription and speech analysis. Offline ASR can exploit full context information from the whole utterance, which may improve its performance and generalization. Therefore, different context information may require different acoustic features and network architectures, making it difficult to jointly optimize a single model for both scenarios.

In recent studies [11], authors have explored the unification of online and offline encoders by using the same decoder for different input features, or by using the output of the online encoder as the input of the offline encoder [12]. However, integrating online and offline encoders often face challenges in offline scenarios, particularly when the online mode is prioritized during optimization. A common approach involves using a single shared decoder based on RNN-T to handle features of varying contexts. However, the shared

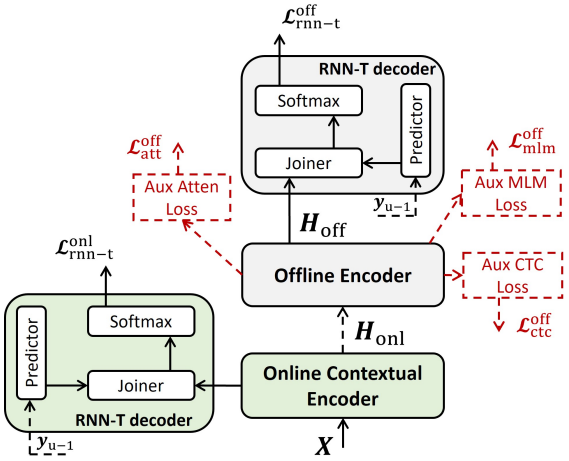

図 1: An end-to-end architecture of online and offline encoders integration with auxiliary losses: In black, the original losses. In red the new auxiliary losses.

decoder may not effectively differentiate between the limited and full context features, leading to the inclusion of more negative samples; thus, degrading the ASR performance. While multitask learning strives to enhance the performance of multiple tasks concurrently, the selective use of auxiliary losses [13] can support the primary task and improve the generalization ability of the framework. Therefore, we propose an end-to-end integration of online and offline ASR using separate RNN-T decoders and auxiliary losses, such as CTC, attention, and MLM loss, each of which brings its own strength in a single model. We expect that our proposed model can achieve better performance than the existing methods.

3 Framework

This section introduces the proposed end-to-end integration of online and offline ASR framework, followed by detailed descriptions of each of our design modules.

3.1 Online and offline encoder

We propose a joint architecture that combines an online encoder and an offline encoder, each with its own RNN-T decoder, to handle both online and offline application scenarios. The online encoder is based on block processing [2], which preserves the previous context information using context embeddings. The offline encoder is based on conformer [3], which captures the full context information from the whole utterance. The RNN-T decoders are based on recurrent

neural network transducer, which models the sequential nature of speech and text. Figure 1 shows the overview of our proposed architecture. The online encoder consists of M encoder layers, each of which has a context inheritance mechanism. The context inheritance mechanism computes a context embedding for each block at each sublayer, and passes it to the next sublayer. The context embedding encodes the past and present context information of the block. The block size and hop length are denoted by L_{block} and L_{hop}, respectively. The b-th block of the input audio feature sequence $\boldsymbol{X_b}$ is defined as:

$$\boldsymbol{X}^b = (\boldsymbol{X}_t | t = (b-1)L_{\text{hop}} + 1, ..., (b-1)L_{\text{hop}} + L_{\text{block}} + 1) \quad (1)$$

The hidden state for each block, labeled as the b-th block, is encoded whereas each block contains a series of hidden states of L_{block}- *length*, i.e., $\boldsymbol{H}^b = (\boldsymbol{h}^b_1, \ldots, \boldsymbol{h}^b_{L_{block}})$. The encoding process is carried out sequentially, resulting in a series of hidden states with a length of T. These features are then input into the offline encoder, where they are transformed into a subsampled sequence of hidden states, also of length T, as given in the Eq.(3).

$$\boldsymbol{H}_{\text{onl}} = \text{OnlineEncoder}(\boldsymbol{X}). \quad (2)$$

$$\boldsymbol{H}_{\text{off}} = \text{OfflineEncoder}(\boldsymbol{H}_{\text{onl}}). \quad (3)$$

In this study, we have the online contextual conformer encoder functioning as an independent online encoder-decoder module. This is linked to an offline encoder-decoder module via an output derived from the online encoder.

3.2 Online encoder-decoder loss

The acoustic features, denoted as $\boldsymbol{X} = (x_1, \ldots, x_T)$ are initially processed by the online module. This module employs a contextual block conformer as an encoder and the RNN-T as an online decoder, as described in [2] . The online RNN-T decoder computes the marginal likelihood of the output y over all possible alignments, as shown in Eq.(4):

$$P_{\text{onl}}(\boldsymbol{y} \mid \boldsymbol{X}) = \sum_{\boldsymbol{o} \in \mathcal{O}(\boldsymbol{y})} P_{\text{onl}}(\boldsymbol{o}|\boldsymbol{X}) = \sum_{\boldsymbol{o} \in \mathcal{O}(\boldsymbol{y})} \left[\prod_{i=1}^{T+S} P\left(\boldsymbol{o}_i \mid \boldsymbol{h}_{t_i}, g_{s_i}\right) \right], \quad (4)$$

The RNN-T model optimizes its parameters by minimizing the negative log-likelihood, as defined in Eq. (5):

$$\mathcal{L}^{\text{rnn-t}}_{\text{onl}} = - \sum_{(\boldsymbol{X} \rightarrow \boldsymbol{H}_{\text{onl}}, \boldsymbol{y})} \log P_{\text{onl}}(\boldsymbol{y} \mid \boldsymbol{H}_{\text{onl}}). \quad (5)$$

This loss ensures that the model is continually optimized to learn the online contextual features during training.

3.3 Offline encoder-decoder loss

The offline encoder-decoder modules receives the processed hidden sequences from the online encoder and employs full-context conformer as an encoder and separate RNN-T as an offline decoder. Here, the offline RNN-T decoder computes the marginal likelihood of the output y over all possible alignments, as shown in Eq.(6):

$$P_{\text{off}}(\boldsymbol{y} \mid \boldsymbol{H}_{\text{onl}}) = \sum_{\boldsymbol{a} \in \mathcal{A}(\boldsymbol{y})} P_{\text{off}}(\boldsymbol{a}|\boldsymbol{H}_{\text{onl}}) = \sum_{\boldsymbol{a} \in \mathcal{A}(\boldsymbol{y})} \left[\prod_{i=1}^{T+S} P\left(\boldsymbol{a}_i \mid \boldsymbol{h}_{t_i}, g_{s_i}\right) \right], \quad (6)$$

where the offline RNN-T loss refines the model parameters by minimizing the negative log-likelihood, as shown below:

$$\mathcal{L}^{\text{rnn-t}}_{\text{off}} = - \sum_{(\boldsymbol{H}_{\text{onl}} \rightarrow \boldsymbol{H}_{\text{off}}, \boldsymbol{y})} \log P_{\text{off}}(\boldsymbol{y} \mid \boldsymbol{H}_{\text{off}}), \quad (7)$$

3.4 Auxiliary losses

In this study, we enhance the performance of the offline ASR by optimizing the offline encoder-decoder module. This optimization is achieved through a multi-task approach, where a full-context Conformer encoder is shared with the CTC, attention, and MLM-based auxiliary losses. The offline encoder, which relies on the limited context hidden state features processed by the online encoder, often experiences performance degradation due to the restricted context information. Our approach mitigates this issue and enhances the robustness of the framework by incorporating additional auxiliary losses into the offline RNN-T decoder. The inclusion of these losses serves a dual purpose. Firstly, they contribute to the regularization of the framework, ensuring stability during the learning process. Secondly, they aid in the optimization of the model by providing additional signals for error correction during training.

3.4.1 CTC loss

The likelihood of the CTC is given in Eq.(8):

$$P_{\text{ctc}}(\boldsymbol{y} \mid \boldsymbol{H}_{\text{off}}) = \sum_{\boldsymbol{z} \in \mathcal{Z}(\boldsymbol{y})} P(\boldsymbol{z} \mid \boldsymbol{H}_{\text{off}}) = \sum_{\boldsymbol{z} \in \mathcal{Z}(\boldsymbol{y})} \left[\prod_{t=1}^{T} P\left(\boldsymbol{z}_t \mid \boldsymbol{h}_t\right) \right], \quad (8)$$

where it serves as an auxiliary loss to refine the model parameters by minimizing the negative log-likelihood, as shown in the Eq.(9) below:

$$L_{\text{ctc}} = - \sum_{(\boldsymbol{H}_{\text{onl}} \rightarrow \boldsymbol{H}_{\text{off}}, \boldsymbol{y})} \log P_{\text{ctc}}(\boldsymbol{y} \mid \boldsymbol{H}_{\text{off}}), \quad (9)$$

3.4.2 Auxiliary attention loss

The likelihood of an attention mechanism is described as in Eq.(10):

$$P_{\text{att}}(\boldsymbol{y} \mid \boldsymbol{H}_{\text{onl}}) = \prod_{s=1}^{S} P\left(y_s \mid \boldsymbol{y_{s-1}}, \boldsymbol{H}_{\text{off}}\right), \quad (10)$$

The auxiliary attention loss provides an alignment between the input and output sequences and brings model regularization by optimizing the corresponding negative log-likelihood, as shown below:

$$L_{\text{att}} = - \sum_{(\boldsymbol{H}_{\text{onl}} \rightarrow \boldsymbol{H}_{\text{off}}, \boldsymbol{y})} \log P_{\text{att}}(\boldsymbol{y} \mid \boldsymbol{H}_{\text{off}}). \quad (11)$$

3.4.3 Auxiliary MLM loss

The MLM auxiliary loss in our framework estimates the token sequence using the full sequence given by $\boldsymbol{H}_{\text{off}}$ as shown in Eq.(3), similar to the attention mechanism. However, during the training phase, MLM distinguishes itself from attention by masking randomly selected tokens, denoted as y_{mask}, with a special token.

Subsequently, y_{mask} is predicted based on the remaining unmasked tokens, y_{obs}, as $P_{\text{mlm}}(y_{\text{mask}}|y_{\text{obs}}, \boldsymbol{H}_{\text{off}})$. Here, the MLM refines the model parameters by minimizing the negative log-likelihood as outlined in the equation below:

$$\mathcal{L}_{\text{off}}^{\text{mlm}} = - \sum_{(\boldsymbol{H}_{\text{onl}} \rightarrow \boldsymbol{H}_{\text{off}}, \boldsymbol{y})} \log P_{\text{mlm}}(y_{\text{mask}}|y_{\text{obs}}, \boldsymbol{H}_{\text{off}}). \quad (12)$$

This approach allows the MLM to contribute to the refinement of the model parameters, enhancing the overall performance of the system.

Finally, the offline loss ($\mathcal{L}_{\text{off}}$) is computed using the weighted sum of individual loss objectives as defined in Eqs.(7), (9), (11) and (12):

$$\mathcal{L}_{\text{off}} = \lambda_{\text{ctc}}\mathcal{L}_{\text{off}}^{\text{ctc}} + \lambda_{\text{rnnt}}\mathcal{L}_{\text{off}}^{\text{rnnt}} + \lambda_{\text{att}}\mathcal{L}_{\text{off}}^{\text{att}} + \lambda_{\text{mlm}}\mathcal{L}_{\text{off}}^{\text{mlm}}, \quad (13)$$

where $\lambda_{\text{ctc}}, \lambda_{\text{rnnt}}, \lambda_{\text{att}}$ and λ_{mlm} are tunable hyperparameters and are determined experimentally. However, for this work we used the hyperparameters as reported in [14] and obtained optimal results.

Finally, we define the total multi-task learning objective for end-to-end integration of online and offline encoders as:

$$\mathcal{L}_{\text{mtl}} = \lambda_{\text{onl}}\mathcal{L}_{\text{rnn-t}}^{\text{onl}} + \lambda_{\text{off}}\mathcal{L}_{\text{off}} \quad (14)$$

where λ_{onl} and λ_{off} are the weighting terms and $\mathcal{L}_{\text{onl}}$ is the online loss obtained from Eq.(5) and $\mathcal{L}_{\text{off}}$ is the offline loss obtained from Eq.(13).

4 Experiments

4.1 Dataset

In this study, we primarily focus on a specific subset, referred to as subset A, of the Corpus of Spontaneous Japanese (CSJ) [15]. This subset consists of academic lecture-based ASR tasks and comprises 236 hours of speech data.

For evaluation purposes, we have divided the dataset into three distinct tasks: eval 1, eval 2, and eval 3. These tasks contain 1.9 hours, 2.0 hours, and 1.3 hours of speech data respectively, providing a comprehensive and diverse set of data for our analysis.

4.2 Experimental setup

In our study, we utilized the ESPnet2 toolkit [16] as a foundation to build our baseline models and the proposed E2E framework. This framework is based on the concept of multi-task learning for ASR task. We integrated additional modules for encoder, decoder, and auxiliary losses into this framework, capitalizing on existing specialized architectures. Our online encoder is composed of twelve layers of 256-dimensional contextual block conformer, each layer having 1024 feed-forward dimensions and 4 attention heads. We apply a dropout rate of 0.1 to each layer. For block-processing [2], we configure the block size to 40 and maintain a look-ahead and hop size of 16 to optimize online streaming performance. Our offline encoder comprises of twelve layers of 256-dimensional

表 1: On Corpus of Spontaneous Japanese (CSJ) : Absolute (abs.) character error rate (CER) and relative (rel.) CERR numbers on CSJ data for i) single baseline models (B1, B3, B5, B6, & B7), ii) a baseline cascaded encoder with shared decoder models (B2 & B4) [12], iii) proposed end-to-end integration of online and offline encoders with separate RNN-T decoders (P1 & P2) and auxiliary losses (P3, P4 & P5). All the results are decoded with a beam size of 10.

Mode	ID	Method	CSJ (SUBSET A)					
			eval1		eval2		eval3	
			abs.↓	rel.(%)↑	abs.↓	rel.(%) ↑	abs.↓	rel.(%)↑
Online	B1	Context-Transducer (baseline)	6.84		4.95		11.72	
	B2	Cascaded [12] (baseline)	6.81	(0.44)	5.07	(-2.42)	11.89	(-1.45)
	P1	Online-Transducer **(ours)**	**6.67**	(2.49)	**4.86**	(1.81)	11.79	(-0.60)
Offline	B3	Conformer-Transducer (baseline)	5.78		4.15		9.94	
	B4	Cascaded [12] (baseline)	5.60	(3.11)	4.04	(2.65)	9.68	(2.61)
	P2	Offline-Transducer **(ours)**	**5.48**	(5.19)	**3.89**	(6.27)	**9.50**	(4.42)
Offline	B5	Conformer-CTC (baseline)	5.50		3.88		9.76	
	P3	Conformer+$\mathcal{L}_{\text{off}}^{\text{ctc}}$ **(ours)**	5.51	(-0.18)	**3.72**	(4.12)	**9.62**	(1.43)
Offline	B6	Conformer-Transformer (baseline)	5.16		3.87		9.90	
	P4	Conformer+$\mathcal{L}_{\text{off}}^{\text{att}}$ **(ours)**	5.29	(-2.46)	**3.71**	(4.13)	**9.55**	(3.54)
Offline	B7	Conformer-MLM (baseline)	5.53		4.00		9.51	
	P5	Conformer+$\mathcal{L}_{\text{off}}^{\text{mlm}}$ **(ours)**	5.58	(-0.90)	**3.81**	(4.75)	9.88	(-3.74)

full-context conformer [3], each layer having 1024 feed-forward dimensions and 4 attention heads. The output from the online contextual block conformer is channeled into the offline conformer block, resulting in a cascaded architecture. For the online encoder, we employ a separate RNN-T decoder with a 256-dimensional embedding prediction network and a 320-dimensional joint network. Conversely, for the offline encoder, we utilized a distinct RNN-T decoder with a 256-dimensional embedding prediction network and a 320-dimensional joint network. To augment the regularization of the offline encoder-decoder, we introduced auxiliary losses based on CTC, attention, and MLM mechanisms. We train this end-to-end architecture for 50 epochs with a learning rate of 0.0015 and warmup steps of 1500. We employ a training weight of 1 to the online encoder to maximize the performance capacity of the online ASR mode. However, for the offline mode and auxiliary losses, we adopt the training weights for $\lambda_{\text{rnnt}}, \lambda_{\text{ctc}}, \lambda_{\text{att}}, \lambda_{\text{mlm}}$ as proposed in [14], i.e., 0.10, 0.15, 0.30, and 0.45 respectively.

4.3 Main results

In this work, we evaluate the performance of our proposed E2E framework with several baseline models. These include the standalone online contextual block conformer transducer (Context-Transducer), the offline full-context conformer transducer (Conformer-Transducer), and a cascaded architecture with a shared RNN-T decoder, as proposed in a previous study [12]. To ensure a fair comparison, we maintained the same number of encoder layers (twelve) for both online and offline modes in all models, including the Context-T and Conformer-T.

For our first primary analysis, we perform an ablation study conducted on the Corpus of Spontaneous Japanese (CSJ) dataset, and is presented in Table 1. In this table, the standalone online and offline transducer baseline models are represented by Context-Transducer (`B1`) and Conformer-Transducer (`B3`), respectively. We also developed a baseline cascaded architecture with shared decoders (`B2` & `B4`) as proposed in the referenced study [12]. These models are compared against our proposed framework (`P1` & `P2`). The

character error rates (CER) listed in Table 1 are obtained using a beam width of 10 for both online and offline RNN-T modules. Our findings indicate that the proposed framework improved the performance of the online ASR path compared to the standalone online model. We observed a relative CER (CERR) improvement ranging from 1.81% to 2.49% across multiple evaluation sets. Moreover, our proposed framework also demonstrates substantial performance improvement for the offline ASR path, with a CERR between 4.4% and 6.3%. These results highlight the effectiveness of our proposed E2E framework in improving the performance of both online and offline ASR modules.

Next, we compare our auxiliary tasks $\mathcal{L}_{\text{off}}^{\text{ctc}}$ (P3), $\mathcal{L}_{\text{off}}^{\text{att}}$ (P4) and $\mathcal{L}_{\text{off}}^{\text{mlm}}$ (P5) against the separately trained Conformer-CTC (B5), Conformer-Transformer (B6), and Conformer-MLM (B7) models. Table 1 summarizes our experiments to understand how each task improved or degrades the performance on CSJ evaluation sets compared to the standalone models. Overall, most auxiliary tasks show improved performance on eval2 and eval3 test sets with an exception for test set eval1 which shows performance degradation. This degradation can be attributed to the fact that training weights in this study are optimized to improve the overall performance of the online and offline transducer modes. The performance for each auxiliary task can potentially be improved by assigning more optimal weights to the individual task. It will allow the model to focus more on optimizing the performance of each auxiliary task, rather than prioritizing the overall performance of the transducer modes.

5 Conclusion

In this study, we propose a novel approach to integrate online and offline ASR modules in an end-to-end manner, utilizing auxiliary losses. This framework is designed to optimize the combination of online and offline RNN-T decoders, leveraging the power of multi-task learning. The primary objective is to enhance the learning of contextual representations, thereby offering increased flexibility in the E2E-ASR framework. Our approach demonstrates a significant improvement in CERR for the CSJ corpus, compared to traditional cascaded architectures [12]. This improvement is particularly noticeable with the introduction of auxiliary losses, which provide additional regularization and refinement to the framework.

参考文献

[1] R. Prabhavalkar, T. Hori, T. N. Sainath, R. Schlüter, and S. Watanabe, "End-to-end speech recognition: A survey," 2023.

[2] E. Tsunoo, Y. Kashiwagi, T. Kumakura, and S. Watanabe, "Transformer ASR with contextual block processing," in *Proc. ASRU*, 2019.

[3] A. Gulati, C. Chiu, J. Qin, J. Yu, N. Parmar, R. Pang, S. Wang, W. Han, Y. Wu, Y. Zhang, and Z. Zhang, "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech*, 2020.

[4] B. L. et al., "A better and faster end-to-end model for streaming asr," in *Proc. ICASSP*, 2021.

[5] A. Graves, "Sequence transduction with recurrent neural networks," in *Proc. ICML*, 2012.

[6] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*, 2006.

[7] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. ICASSP*, 2016.

[8] 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.

[9] F. Weninger, M. Gaudesi, M. A. Haidar, N. Ferri, J. Andr'es-Ferrer, and P. Zhan, "Conformer with dual-mode chunked attention for joint online and offline asr," in *Proc. Interspeech*, 2022.

[10] Y. Sudo, M. Shakeel, Y. Peng, and S. Watanabe, "Time-synchronous one-pass beam search for parallel online and offline transducers with dynamic block training," in *Proc. Interspeech*, 2023.

[11] 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," in *Proc. ICLR*, 2021.

[12] A. Narayanan, T. N. Sainath, R. Pang, J. Yu, C.-C. Chiu, R. Prabhavalkar, E. Variani, and T. Strohman, "Cascaded encoders for unifying streaming and non-streaming asr," in *Proc. ICASSP*, 2021.

[13] C. Liu, F. Zhang, D. Le, S. Kim, Y. Saraf, and G. Zweig, "Improving rnn transducer based asr with auxiliary tasks," in *Proc. SLT*, 2021.

[14] Y. Sudo, M. Shakeel, B. Yan, J. Shi, and S. Watanabe, "4D ASR: Joint modeling of CTC, attention, transducer, and mask-predict decoders," in *Proc. Interspeech*, 2023.

[15] K. Maekawa, H. Koiso, S. Furui, and H. Isahara, "Spontaneous speech corpus of japanese," in *Proc. LREC*, 2000.

[16] S. W. et al., "Espnet: End-to-end speech processing toolkit," in *Proc. Interspeech*, 2018.