UniEnc-CASSNAT: An Encoder-Only Non-Autoregressive ASR for Speech SSL Models

Ruchao Fan, *Graduate Student Member, IEEE*, Natarajan Balaji Shankar, *Graduate Student Member, IEEE*, and Abeer Alwan, *Fellow, IEEE*

***bstract*—Non-autoregressive automatic speech recognition NASR) models have gained attention due to their parallelism and fast inference. The encoder-based NASR, e.g. connectionist temporal classification CTC), can be initialized from the speech foundation models SFM) but does not account for any dependencies among intermediate tokens. The encoder-decoder-based NASR, like CTC alignment-based single-step non-autoregressive transformer CASS-NAT), can mitigate the dependency problem but is not able to efficiently integrate SFM. Inspired by the success of recent work of speech-text joint pre-training with a shared transformer encoder, we propose a new encoder-based NASR, UniEnc-CASSNAT, to combine the advantages of CTC and CASS-NAT. UniEnc-CASSNAT consists of only an encoder as the major module, which can be the SFM. The encoder plays the role of both the CASS-NAT encoder and decoder by two forward passes. The first pass of the encoder accepts the speech signal as input, while the concatenation of the speech signal and the token-level acoustic embedding is used as the input for the second pass. Examined on the Librispeech 100 h, MyST, and Aishell1 datasets, the proposed UniEnc-CASSNAT achieves state-of-the-art NASR results and is better or comparable to CASS-NAT with only an encoder and hence, fewer model parameters.**

***Index Terms*—Non-autoregressive ASR, E2E ASR, self-supervised learning, speech foundation model.**

I. Introduction

IN RECENT years, self-supervised learning (SSL) has become popular in speech [1], [2], [3] and natural language [4], [5] processing. The SSL models learn prior knowledge from a large amount of unannotated data and are called pre-trained or foundation models. Widely-used speech foundation models include APC [6] that predicts future frames from their histories, and Wav2vec2.0 [7], HuBERT [8], and WavLM [9] that reconstruct or predict pseudo labels via the masked portions of the speech signal. The speech foundation models are proven effective in improving low-resource tasks by fine-tuning [10].

Concurrently, non-autoregressive automatic speech recognition (NASR) has attracted considerable interest due to its fast inference [11], [12], [13]. Although it is not naturally designed for streaming ASR, NASR can greatly improve the inference efficiency for offline applications. As the earliest end-to-end ASR framework, connectionist temporal classification (CTC) [14], [15] can be regarded as an encoder-based NASR model when using greedy decoding. However, the performance of CTC is always constrained by the output independence assumption. On the other hand, most NASR models are proposed based on the encoder-decoder framework where the decoder can mitigate the output independence problem. For example, the decoder of Mask-CTC [16] is a masked language model to correct the low confidence tokens in CTC output. Align-Refine [17] uses the decoder to refine the CTC alignment iteratively. LASO [18], CASS-NAT [19], and Paraformer [20] extract acoustic embedding as the decoder input for token-level contextual representation learning. However, the encoder-decoder framework does not perfectly fit the current foundation models, which are pre-trained with the transformer encoder structure. Although previous work developed pre-trained models [21], [22] for the encoder-decoder framework, it is specifically designed for autoregressive transformers. Additionally, [23] trains the transformer decoder from scratch with the encoder initialized from the speech foundation model. The work in [24] and [25] introduce BERT to the NASR model for better output dependency modeling. However, these methods may contain unnecessary model parameters.

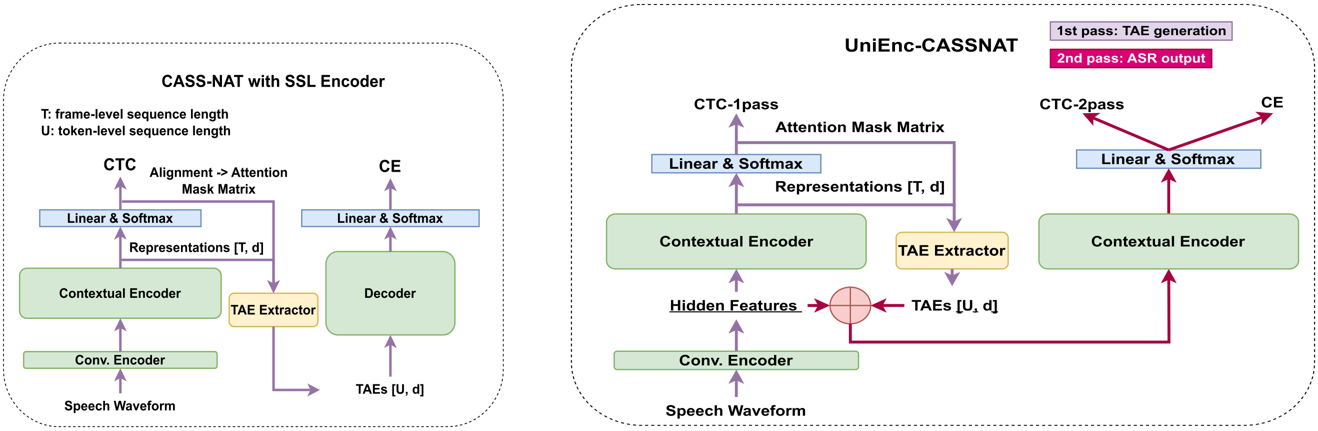
In this work, based on previous method (CASS-NAT) [26], we present a new encoder-only NASR (UniEnc-CASSNAT) that can function in a way that is similar to CASS-NAT encoder and decoder. Like CTC, UniEnc-CASSNAT can be initialized from speech foundation models (HuBERT base model [8] is used). To behave as both the CASS-NAT encoder and decoder, UniEnc-CASSNAT has two forward passes and accepts two types of input for each. In the first pass, speech features (output of HuBERT Conv. encoder) are fed into the contextual encoder to generate token-level acoustic embeddings (TAEs). In the second pass, the concatenation of speech features and the TAEs (along the time dimension) are used as the contextual encoder inputs. The TAE corresponding outputs are selected for ASR loss computation. The outputs in the second pass can generate better quality TAEs than those in the first pass and hence lead to better ASR performance. We, therefore, further propose a multi-pass CTC (MP-CTC) training and iterative decoding method to improve the WER performance. Experiments on Librispeech

Manuscript received 23 October 2023; revised 29 January 2024; accepted 3 February 2024. Date of publication 12 February 2024; date of current version 5 March 2024. This work was supported by the NSF. The associate editor coordinating the review of this manuscript and approving it for publication was Ville M. Hautamaki. *Corresponding author: Ruchao Fan.)*

The authors are with the Department of ECE, University of California Los Angeles, Los Angeles, CA 90095 USA (e-mail: fanruchao@g.ucla.edu; balaji1312@ucla.edu; alwan@ee.ucla.edu).

Our codes are publicly available: https://github.com/Diamondfan/cassnat_asr

Digital Object Identifier 10.1109/LSP.2024.3365036



(a) CASS-NAT (b) UniEnc-CASSNAT

Fig. 1. (a) Diagram of CASS-NAT. (b) Proposed UniEnc-CASSNAT. HuBERT conv. and contextual encoders are used. The TAE extractor is a self-attention module that transforms the acoustic representations with length T to TAEs with length U. The generation of TAEs and second pass forward computation are repeated during iterative decoding.

100-hour [27], MyST [28], and Aishell1 [29] datasets show that the proposed methods can achieve better or comparable WERs to CASS-NAT, and contain fewer parameters. The framework can be applied to other encoder-decoder-based NASR.

The remainder of this paper is organized as follows. Section II introduces the framework of UniEnc-CASSNAT and iterative decoding process. Experimental setups are described in Section III. Results are shown and discussed in Section IV. We conclude the paper in Section V.

II. Proposed Framework: UniEnc-CASSNAT

A. Encoder-Decoder CASS-NAT

CASS-NAT [19] consists of an encoder, a token-level embedding extractor (TAEE), and a decoder as plotted in Fig. 1(a). The connectionist temporal classification (CTC) [14] loss is added to learn the alignment between the acoustic and token sequences. The alignment can provide segmentation information for each token. TAEE extracts an embedding for each token from encoder outputs (with a shape of $[T \; d]$, where T is the frame length and d is the hidden dimension) using the segmentation information. The extracted token-level acoustic embeddings (TAEs) (with a shape of $[U \; d]$, where U is the token sequence length) are fed into the decoder, which models the relationship between tokens. Suppose the input sequence is $X = \{x_1 \; \ldots \; x_t \; \ldots \; x_T\}$, the ground truth is $Y = \{y_1 \; \ldots . y_u \; \ldots \; y_U\}$ and the CTC alignment is Z, then the objective function on the decoder side can be written as:

$$\begin{aligned} L_{\text{dec}} &= \log P \;\, Y|X) \\ &\geq \;\; {}_{Z|X}[\log P \;\, Y|Z \;\; X)] \\ &\approx \max_{Z} \log \prod_{u=1}^{U} P \;\, y_u|z_{t \; \_ \quad 1:t} \quad x_{1:T}) \end{aligned} \tag{1}$$

where $t_{u-1}+1 : t_u$ represents the acoustical boundary for token u provided by the alignment Z. We use a maximum approximation for the expectation (Viterbi-alignment during training). CASS-NAT is then trained by jointly maximizing the decoder loss in (1) and the CTC loss on the encoder side with a task ratio .

$$L_{\text{joint}} = L_{\text{dec}} + \quad \log \sum_{Z \in q} \prod_{i=1}^{T} P \;\, z_i|X) \tag{2}$$

where q is all the alignments that can be mapped to the label Y by removing blank tokens in CTC and repetitions.

During decoding, Viterbi-alignment is not available. We therefore use error-based sampled alignments (ESA) (see details in [19]), where the multiple alignments Z are sampled based on the CTC greedy search output with low confidence scores. The TAEs computed from the sampled alignments are fed into the decoder to obtain multiple ASR outputs. The autoregressive transformer provides a ranking score for each ASR output (one ASR output corresponds to one alignment).

B. Encoder-Only CASS-NAT: UniEnc-CASSNAT

Speech foundation models are proven to be useful in downstream ASR tasks. The encoder of CASS-NAT can be inherited from a speech foundation model and extracts better acoustic representations [23]. However, the CASS-NAT decoder has to be trained from scratch. Inspired by the success of recent work of speech-text joint pre-training [30], [31] with a shared encoder, we rethought the necessity of CASS-NAT decoder and propose an encoder-only CASS-NAT, denoted as UniEnc-CASSNAT to fit the size of the speech foundation models.

The UniEnc-CASSNAT is shown in Fig. 1(b) with two forward passes. In the first pass, the hidden features extracted from the conv. encoder are fed into the contextual encoder for CTC modeling and the token-level acoustic embeddings (TAEs) are extracted using the alignment information from CTC outputs. In the second pass, the extracted TAEs ($[U \; d]$) are concatenated with the hidden features ($[T \; d]$) (along the time dimension) to be the input to the contextual encoder. The self-attention layer in the contextual encoder enables frame-frame, frame-token, and token-token interactions between hidden features and TAEs. Note that the goal of the first pass is to obtain TAEs, whose quality is highly related to the ASR performance. The better the speech foundation model, the better the quality of the TAEs extracted by UniEnc-CASSNAT. The second pass is similar to the role of the CASS-NAT decoder for modeling the relationships between TAEs and frame-level hidden features. We investigate whether the encoder is capable of both frame-level acoustic representation learning and contextual modeling between tokens.

C. MP-CTC Training and Iterative Decoding

The output of the second pass is a sequence of $T + U$ vectors, where the first T vectors correspond to hidden features, and the U vectors correspond to TAEs. Since the quality of TAEs is essential to the performance of the CASS-NAT decoder, we propose to add another CTC loss to the first T outputs of the second pass and formulate a multi-pass CTC (MP-CTC) training. With the CE loss used on the U outputs, the final objective function of UniEnc-CASSNAT can be written as:

$$L_{\text{unienc-cassnat}} = L_{\text{dec}} + \quad {}_1 \quad L_{CTC-1pass} + \quad {}_2 \quad L_{CTC-2pass} \tag{3}$$

We share the final feed-forward layer for the two CTC losses. Theoretically, the second-pass CTC loss would have better performance than the first pass because it accepts additional input information (TAEs). An intuitive idea is to iteratively improve the quality of TAEs by repeating the second pass with newly extracted TAEs. Hence, we propose an iterative decoding method for UniEnc-CASSNAT. Specifically, we define the hidden features as H, and the first pass of UniEnc-CASSNAT encoder as Iter_0. Iter_0 would generate TAE_0. The second pass uses $H + \text{TAE}_0$ as input and generates TAE_1, which we define as Iter_1. Generally, for iteration n, the contextual encoder accepts H and TAE_{n-1} as input and generates TAE_n for the iteration $n + 1$. In each iteration, ESA generates multiple TAEs for the next iteration. We define the number of sampled alignments in each iteration as S_n. The total number of the sampled alignments would be $\;{}^{N-1}_{n=0} S_n$, where N is the number of iterations used in the decoding. We empirically found that two iterations are sufficient for a desirable word error rate (WER).

III. Experimental Setup

A. Data Settings

The experiments were conducted on three datasets: the 100-hour subset of LibriSpeech English corpus [27], the 240-hour (annotated section) My Science Tutor (MyST) children's speech corpus [28], and 170-hour Aishell1 Mandarian corpus [29]. We chose the 100-hour subset of Librispeech to enable comparisons with previous work on NASR. We conducted pre-processing on MyST dataset to get a better baseline compared to [23]. For example, we mapped filling pauses, non-speech events, and truncated words to $\langle$UNK$\rangle$. The $\langle$UNK$\rangle$ is not considered when computing WER.

The sets of output labels consist of 1024 word-pieces for Librispeech 100 h and 500 word-pieces for the MyST, obtained by the SentencePiece method [32]. For Aishell, 4230 characters are used as the vocabulary.

B. Model Settings

A CTC/Attention autoregressive transformer (AT) baseline was first trained with an architecture of a 12-block encoder and a 6-block decoder. Suppose the tuple of a transformer setting is represented by (model dimension, feed-forward layer dimension, number of heads in self-attention), we define three settings: d_{512} for (512, 2048, 8), d_{768} for (768, 3072, 12), and d_{256} for (256, 2048, 4). d_{512} is used for the two English datasets, and d_{256} is used for the Aishell1 dataset. Later on, we follow the same setting as in [23] for CASS-NAT training. For a fair comparison, we also include a CTC baseline as an encoder-only NASR architecture. When training with the speech foundation models, the 12-block encoder was replaced with a HuBERT-base model, either the English[1] version for Librispeech and MyST, or the Chinese version[2] for Aishell1. We also conducted experiments on the TAE extractor in UniEnc-CASSNAT to examine the trade-off between performance and model size.

All models are optimized using a noam scheduler [33] with warmup steps of 15 k (10 k for Librispeech 100 h), a peak learning rate of 5e-5 for the encoder, and 1e-3 (5e-4 for MyST) for uninitialized modules. The models were trained using a batch size of 80 s audio samples (40 s for MyST because it contains longer utterances). The training either stops when the WER of the valid set doesn't improve for 10 epochs or is terminated at 30 epochs. For MP-CTC training, the task ratio of CTC loss in the second pass is set to one.

All results are decoded without the usage of the external language model. For the AT baseline, the beam search decoding is applied with a beam size of 20 for Librispeech and MyST, and 10 for Aishell1. For CASS-NAT, the number of sampled alignments is 50 and the threshold is 0.9. We explore the effects of the number of sampled alignments in two iterations, and the threshold for each iteration is set to 0.9 as well.

IV. Results and Discussion

A. Main Results

The main WER results of UniEnc-CASSNAT on the Librispeech 100 h, MyST, and Aishell1 datasets are shown in Table I. We first train two autoregressive transformer baselines with or without the usage of self-supervised learning. The results in the table again show the effectiveness of the speech foundation models. CASS-NAT achieves close performance to their AT counterpart, which is consistent with previous work. We also present the results of CTC on the three datasets. Due to the output-independent assumption, CTC is worse than the AT baseline and CASS-NAT although it requires fewer parameters. Note that the motivation of UniEnc-CASSNAT is to investigate whether the encoder can jointly model the frame-level and token-level acoustic embedding without the use of the decoder and thus has fewer model parameters. We expect to obtain a model with similar model parameters compared to CTC but close performance to the CASS-NAT. As shown in Table I, the proposed UniEnc-CASSNAT achieves comparable or better results than CASS-NAT, for example, a WER of 11.0 for UniEnc-CASSNAT vs. 11.2 for CASS-NAT on the Librispeech test-other set, but is superior to CASS-NAT in terms of model size (99.3 M vs. 130.5 M). A smaller model size can be helpful for on-device deployment. Compared to CTC, the UniEnc-CASSNAT achieves much better performance than CTC with a similar model size. The additional 3 M parameters compared to CTC (95.7 M) are from the TAE extractor. The limitation of UniEnc-CASSNAT could be

[1] [Online]. Available: https://dl.fbaipublicfiles.com/hubert/hubert_base_ls960.pt

[2] [Online]. Available: https://huggingface.co/TencentGameMate/chinese-hubert-base

TABLE I
WER PERFORMANCE OF UNIENC-CASSNAT AND COMPARISONS TO PREVIOUS METHODS ON LIBRISPEECH-100 H, MYST, AND AISHELL1 DATASETS

Model Type	Model Size	Librispeech-100h					MyST		Model Size	Aishell1	
		dev-clean	dev-other	test-clean	test-other	RTF	dev	test		dev	test
AT-w/o SSL	85.1M	6.6	18.2	6.9	18.2	0.325	13.5	14.9	33.6M	4.6	5.0
AT-w/ SSL	121.6M	4.8	11.0	4.8	10.8	0.486	11.4	13.1	107.3M	4.0	4.3
Non-autoregressive ASR											
Previous SOTA	w/ SSL	4.6	11.3	4.8	11.3[34]	-	16.0	15.6[23]	w/ SSL	3.6	3.8[35]
BERT-CTC [25]	-	7.0	16.3	7.2	16.6	-	-	-	143M	3.9	3.9
CTC	95.7M	6.1	13.8	6.2	13.8	0.005	12.9	14.5	95.7M	4.5	4.9
CASS-NAT	130.5M	**4.7**	11.4	4.9	11.2	0.014	11.9	**13.5**	109.7M	**4.0**	**4.3**
UniEnc-CASSNAT	99.3M	4.9	**11.0**	**4.8**	**11.0**	0.093	**11.8**	**13.5**	102.7M	4.2	4.5

State-of-the-art (SOTA) results with the usage of speech foundation models that are pre-trained with the same amount of unannotated data to ours are reported. The real-time factor (RTF) of each method on Librispeech test-other data is presented for speed comparison. All bold-faced improvements are statistically significant.

its slower inference than CTC and CASSNAT because of the multiple forward computations of the encoder with a longer input sequence (concatenation of frames and tokens). The RTF values in Table I show that the UniEnc-CASSNAT is still 3–5x faster than the AT models although it is 6x slower than CASS-NAT.

The proposed UniEnc-CASSNAT achieves the best-performing NASR results so far in the literature [23], [34] on Librispeech 100 h and MyST. One can find better WER performance on the Librispeech 100 h data, for example, in [21], [36]. However, in that work, the authors either use a larger model trained with Libri60 k hours of data or extra text data. We compare the UniEnc-CASSNAT results to a similar work BERT-CTC [25], which also uses an encoder-only structure. Differently, UniEnc-CASSNAT generates ASR outputs with CE loss instead of CTC loss in BERT-CTC and does not require a pre-trained BERT module (smaller in size than BERT-CTC). In addition, UniEnc-CASSNAT achieves the best performance with two iterations only instead of more than 10 iterations in BERT-CTC (faster inference). Based on the results in Table I, UniEnc-CASSNAT is better on Librispeech-100 h but worse on Aishell1 than BERT-CTC. The reason could be that Aishell1 contains simple sentences where a pre-trained BERT model is more beneficial [25] and the BERT-CTC has 143 M parameters versus 102 M in UniEnc-CASSNAT.

B. Ablation Study of UniEnc-CASSNAT

We present more results on the Librispeech 100 h data to show the importance of the proposed MP-CTC training and iterative decoding. First, we set $_2$ in (3) to zero and train a UniEnc-CASSNAT with only first-pass CTC. The results in Table II show that the single-pass CTC (SP-CTC) training has a performance gap compared to the CASS-NAT. Additionally, SP-CTC training is not able to perform iterative decoding because TAE$_n$ $_1$ is not constrained by CTC outputs. MP-CTC training is also worse than the CASS-NAT without iterative decoding (e.g. (50, NA)). When applying iterative decoding, we explore different combinations of the number of sampled alignments S_n in each iteration. The total number of sampled alignments is set to the same as that used in CASS-NAT for a fair comparison. As shown in Table II, iterative decoding with a setting of 25 2) achieves the best WER performance and is better than the WER of CASS-NAT. Most of the combinations of S_n achieve comparable WERs to CASS-NAT. It is also noted that the diversity of sampled alignments in the first iteration is more important than that in the second iteration.

TABLE II
ABLATION STUDY OF MP-CTC TRAINING, THE SIZE OF THE TAE MODULE, AND THE ITERATIVE DECODING

Model Type	(S_1, S_2)	dev-clean	dev-other	test-clean	test-other
CASS-NAT	(50, NA)	4.7	11.4	4.9	11.2
	UniEnc-CASSNAT				
SP-CTC	(50, NA)	4.9	11.9	5.0	11.6
MP-CTC-d_{512}	(50, NA)	5.0	11.7	5.2	11.8
	(50, 1)	5.0	11.1	4.9	11.1
	(25, 2)	**4.9**	**11.0**	**4.8**	**11.0**
	(10, 5)	4.9	11.1	4.9	11.1
	(5, 10)	4.9	11.1	4.9	11.2
	(2, 25)	5.0	11.4	5.1	11.4
	(1, 50)	5.2	11.5	5.3	11.6
MP-CTC-d_{256}	(25, 2)	4.9	11.4	4.9	11.2
MP-CTC-d_{768}	(25, 2)	4.7	11.2	4.8	11.0

d_{256}, d_{512}, and d_{768} are defined in Section III-B and their model sizes (including encoder) are 96.1M, 99.3M, 104.2M, respectively. S_n is the number of sampled alignments in the iteration n.

Finally, since the TAE extractor introduces extra model parameters besides the foundation model, we conduct experiments of UniEnc-CASSNAT with different transformer settings (d_{256}, d_{512}, d_{768}) described in Section III-B). The results are also shown in Table II. We can see from the table that with a bigger TAEE module, the performance tends to be better. However, we select MP-CTC-d_{512} as the final results to show in Table I because MP-CTC-d_{768} did not achieve significant improvements with additional 5 M parameters.

V. CONCLUSION

We present a novel encoder-only non-autoregressive ASR (NASR) model, UniEnc-CASSNAT, which integrates the advantage of CTC and CASS-NAT. The encoder of UniEnc-CASSNAT acts as both the encoder and decoder in CASS-NAT to reduce the model parameters and can be well initialized from the speech foundation models. Furthermore, MP-CTC training and iterative decoding are proposed for UniEnc-CASSNAT to further improve the performance to be better or comparable to CASS-NAT. We examined the effectiveness of the proposed methods on the Librispeech 100 h, MyST, and Aishell1 datasets. To the best of our knowledge, we have achieved the best-performing WER results for NASR on the first two datasets with the same settings as those in the literature. Future work includes model compression and distillation to further reduce the parameters.

References

[1] Y. Zhang et al., "BigSSL: Exploring the frontier of large-scale semi-supervised learning for automatic speech recognition," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 6, pp. 1519–1532, Oct. 2022.

[2] A. Mohamed et al., "Self-supervised speech representation learning: A review," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 6, pp. 1179–1210, Oct. 2022.

[3] X. Chang et al., "An exploration of self-supervised pretrained representations for end-to-end speech recognition," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2021, pp. 228–235.

[4] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 5485–5551, 2020.

[5] J. Devlin, M.-W. C. Kenton, and L. K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Computat. Linguistics-HLT*, 2019, pp. 4171–4186.

[6] Y.-A. Chung and J. R. Glass, "Generative pre-training for speech with autoregressive predictive coding," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 3497–3501.

[7] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 12449–12460.

[8] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 29, pp. 3451–3460, 2021.

[9] S. Chen et al., "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 6, pp. 1505–1518, Oct. 2022.

[10] R. Fan, Y. Zhu, J. Wang, and A. Alwan, "Towards better domain adaptation for self-supervised models: A case study of child ASR," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 6, pp. 1242–1252, Oct. 2022.

[11] N. Chen, S. Watanabe, J. Villalba, P. Żelasko, and N. Dehak, "Non-autoregressive transformer for speech recognition," *IEEE Signal Process. Lett.*, vol. 28, pp. 121–125, 2021.

[12] J. Nozaki and T. Komatsu, "Relaxing the conditional independence assumption of CTC-based ASR by conditioning on intermediate predictions," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 3735–3739.

[13] Y. Higuchi et al., "A comparative study on non-autoregressive modelings for speech-to-text generation," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2021, pp. 47–54.

[14] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. 23 rd Int. Conf. Mach. Learn.*, 2006, pp. 369–376.

[15] E. G. Ng, C.-C. Chiu, Y. Zhang, and W. Chan, "Pushing the limits of non-autoregressive speech recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*2021, pp. 3725–3729.

[16] 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. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 3655–3659.

[17] E. A. Chi et al., "Align-refine: Non-autoregressive speech recognition via iterative realignment," in *Proc. Conf. North Amer. Chapter Assoc. Computat. Linguistics: Hum. Lang. Technol.*, 2021, pp. 1920–1927.

[18] Y. Bai, J. Yi, J. Tao, Z. Tian, Z. Wen, and S. Zhang, "Fast end-to-end speech recognition via non-autoregressive models and cross-modal knowledge transferring from BERT," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 1897–1911, 2021.

[19] R. Fan, W. Chu, P. Chang, and J. Xiao, "CASS-NAT: CTC alignment-based single step non-autoregressive transformer for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 5889–5893.

[20] Z. Gao et al., "Paraformer: Fast and accurate parallel transformer for non-autoregressive end-to-end speech recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2022, pp. 2063–2067.

[21] Z. Zhang et al., "SpeechUT: Bridging speech and text with hidden-unit for encoder-decoder based speech-text pre-training," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2022, pp. 1663–1676.

[22] J. Ao et al., "SpeechT5: Unified-modal encoder-decoder pre-training for spoken language processing," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics Vol. 1: Long Papers)*, 2022, pp. 5723–5738.

[23] R. Fan, W. Chu, P. Chang, and A. Alwan, "A CTC alignment-based non-autoregressive transformer for end-to-end automatic speech recognition," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 1436–1448, 2023.

[24] K. Deng et al., "Improving non-autoregressive end-to-end speech recognition with pre-trained acoustic and language models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 8522–8526.

[25] Y. Higuchi, B. Yan, S. Arora, T. Ogawa, T. Kobayashi, and S. Watanabe, "BERT meets CTC: New formulation of end-to-end speech recognition with pre-trained masked language model," in *Proc. Findings Assoc. Computat. Linguistics: EMNLP 2022*, 2022, pp. 5486–5503.

[26] R. Fan, W. Chu, P. Chang, J. Xiao, and A. Alwan, "An improved single step non-autoregressive transformer for automatic speech recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 3715–3719.

[27] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2015, pp. 5206–5210.

[28] W. Ward et al., "My science tutor: A conversational multimedia virtual tutor for elementary school science," *ACM Trans. Speech Lang. Process.*, vol. 7, no. 4, pp. 1–29, 2011.

[29] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "AISHELL-1: An open-source mandarin speech corpus and a speech recognition baseline," in *Proc. 20th Conf. Oriental Chapter Int. Coordinating Committee Speech Databases Speech I/O Syst. Assessment*, 2017, pp. 1–5.

[30] Y. Tang et al., "Unified speech-text pre-training for speech translation and recognition," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics Volume 1: Long Papers)*, 2022, pp. 1488–1499.

[31] G. Wang et al., "Understanding shared speech-text representations," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2023, pp. 1–5.

[32] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proc. Conf. Empirical Methods in Natural Lang. Process.: Syst. Demonstrations*, 2018, pp. 66–71.

[33] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.

[34] R. Fan, Y. Wang, Y. Gaur, and J. Li, "CTCBERT: Advancing hidden-unit bert with CTC objectives," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process*, 2023, pp. 1–5.

[35] G. Zheng, Y. Xiao, K. Gong, P. Zhou, X. Liang, and L. Lin, "Wav-bert: Cooperative acoustic and linguistic representation learning for low-resource speech recognition," in *Proc. Findings Assoc. Comput. Linguistics: EMNLP 2021*, 2021, pp. 2765–2777.

[36] F. Wu et al., "WAV2SEQ: Pre-training speech-to-text encoder-decoder models using pseudo languages," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2023, pp. 1–5.